Salt Documentation

Release 3000.3

SaltStack, Inc.

Oct 21, 2020

CONTENTS

1 **Introduction to Salt** **1**
- 1.1 The 30 second summary 1
- 1.2 Simplicity 1
- 1.3 Parallel execution 1
- 1.4 Builds on proven technology 2
- 1.5 Python client interface 2
- 1.6 Fast, flexible, scalable 2
- 1.7 Open 2
- 1.8 Salt Community 2
- 1.9 Mailing List 2
- 1.10 IRC 3
- 1.11 Follow on Github 3
- 1.12 Blogs 3
- 1.13 Example Salt States 3
- 1.14 Follow on Open Hub 3
- 1.15 Other community links 4
- 1.16 Hack the Source 4

2 **Installation** **5**
- 2.1 Quick Install 5
- 2.2 Platform-specific Installation Instructions 5
- 2.3 Initial Configuration 29
- 2.4 Additional Installation Guides 32
- 2.5 Dependencies 48
- 2.6 Optional Dependencies 49
- 2.7 Upgrading Salt 49
- 2.8 Building Packages using Salt Pack 49

3 **Configuring Salt** **51**
- 3.1 Configuring the Salt Master 51
- 3.2 Configuring the Salt Minion 123
- 3.3 Configuring the Salt Proxy Minion 168
- 3.4 Configuration file examples 170
- 3.5 Minion Blackout Configuration 223
- 3.6 Access Control System 223
- 3.7 Job Management 234
- 3.8 Managing the Job Cache 241
- 3.9 Storing Job Results in an External System 242
- 3.10 Logging 245
- 3.11 External Logging Handlers 248

3.12 Salt File Server . . . 253
3.13 Git Fileserver Backend Walkthrough . . . 259
3.14 MinionFS Backend Walkthrough . . . 275
3.15 Salt Package Manager . . . 277
3.16 Storing Data in Other Databases . . . 295
3.17 Running the Salt Master/Minion as an Unprivileged User . . . 298
3.18 Using cron with Salt . . . 298
3.19 Use cron to initiate a highstate . . . 298
3.20 Hardening Salt . . . 299
3.21 Security disclosure policy . . . 300
3.22 Salt Transport . . . 302
3.23 Master Tops System . . . 304
3.24 Returners . . . 305
3.25 Renderers . . . 357

4 Using Salt **387**
4.1 Grains . . . 387
4.2 Storing Static Data in the Pillar . . . 391
4.3 Targeting Minions . . . 408
4.4 The Salt Mine . . . 417
4.5 Runners . . . 421
4.6 Salt Engines . . . 423
4.7 Understanding YAML . . . 424
4.8 Understanding Jinja . . . 426
4.9 Tutorials Index . . . 453
4.10 Troubleshooting . . . 553
4.11 Frequently Asked Questions . . . 568
4.12 Salt Best Practices . . . 576

5 Remote Execution **585**
5.1 Running Commands on Salt Minions . . . 585
5.2 Writing Execution Modules . . . 587
5.3 Executors . . . 597

6 Configuration Management **599**
6.1 State System Reference . . . 600

7 Return Codes **665**
7.1 Retcode Passthrough . . . 665

8 Utility Modules - Code Reuse in Custom Modules **667**

9 Events & Reactor **671**
9.1 Event System . . . 671
9.2 Beacons . . . 679
9.3 Reactor System . . . 684

10 Orchestration **697**
10.1 Orchestrate Runner . . . 697

11 Solaris **707**
11.1 Solaris-specific Behaviour . . . 707

12 Salt SSH **709**
12.1 Getting Started . . . 709

12.2 Salt SSH Roster . . . 709
12.3 Deploy ssh key for salt-ssh . . . 710
12.4 Calling Salt SSH . . . 710
12.5 States Via Salt SSH . . . 711
12.6 Targeting with Salt SSH . . . 711
12.7 Configuring Salt SSH . . . 711
12.8 Running Salt SSH as non-root user . . . 712
12.9 Define CLI Options with Saltfile . . . 712
12.10 Debugging salt-ssh . . . 712

13 Thorium Complex Reactor 715
13.1 Starting the Thorium Engine . . . 715
13.2 Thorium Modules . . . 715
13.3 Writing Thorium Formulas . . . 715
13.4 The Thorium Register . . . 718

14 Salt Cloud 721
14.1 Configuration . . . 721
14.2 Configuration Inheritance . . . 721
14.3 QuickStart . . . 722
14.4 Using Salt Cloud . . . 722
14.5 Core Configuration . . . 732
14.6 Windows Configuration . . . 742
14.7 Cloud Provider Specifics . . . 745
14.8 Miscellaneous Options . . . 899
14.9 Troubleshooting Steps . . . 905
14.10 Extending Salt Cloud . . . 908
14.11 Using Salt Cloud from Salt . . . 917
14.12 Feature Comparison . . . 922
14.13 Tutorials . . . 925

15 Salt Proxy Minion 933
15.1 New in 2017.7.0 . . . 933
15.2 New in 2016.11.0 . . . 933
15.3 New in 2016.3 . . . 934
15.4 New in 2015.8.2 . . . 935
15.5 New in 2015.8 . . . 935
15.6 Getting Started . . . 935
15.7 The __proxyenabled__ directive . . . 943
15.8 SSH Proxymodules . . . 945

16 Network Automation 953
16.1 New in Carbon (2016.11) . . . 953
16.2 NAPALM . . . 953
16.3 JUNOS . . . 960

17 Salt Virt 963
17.1 Salt Virt Tutorial . . . 963
17.2 The Salt Virt Runner . . . 963
17.3 Based on Live State Data . . . 964
17.4 Deploy from Network or Disk . . . 964

18 Command Line Reference 967
18.1 salt-call . . . 967
18.2 salt . . . 969

18.3 salt-cloud . . . 973
18.4 salt-cp . . . 973
18.5 salt-extend . . . 975
18.6 salt-key . . . 976
18.7 salt-master . . . 979
18.8 salt-minion . . . 980
18.9 salt-proxy . . . 982
18.10 salt-run . . . 983
18.11 salt-ssh . . . 984
18.12 salt-syndic . . . 988
18.13 salt-unity . . . 989
18.14 salt-api . . . 989
18.15 spm . . . 990

19 Pillars **993**

20 Master Tops **995**

21 Salt Module Reference **997**
21.1 auth modules . . . 997
21.2 beacon modules . . . 1005
21.3 cache modules . . . 1025
21.4 cloud modules . . . 1031
21.5 engine modules . . . 1153
21.6 executors modules . . . 1171
21.7 fileserver modules . . . 1172
21.8 grains modules . . . 1176
21.9 execution modules . . . 1185
21.10 netapi modules . . . 2934
21.11 output modules . . . 2979
21.12 pillar modules . . . 2989
21.13 proxy modules . . . 3048
21.14 queue modules . . . 3087
21.15 roster modules . . . 3088
21.16 runner modules . . . 3094
21.17 sdb modules . . . 3161
21.18 serializer modules . . . 3171
21.19 state modules . . . 3175
21.20 thorium modules . . . 3818
21.21 auth modules . . . 3828
21.22 master tops modules . . . 3829
21.23 wheel modules . . . 3837

22 APIs **3843**
22.1 Python client API . . . 3843
22.2 `netapi` modules . . . 3854

23 Architecture **3857**
23.1 High Availability Features in Salt . . . 3857
23.2 Salt Syndic . . . 3858

24 Minion Data Cache **3863**
24.1 Pluggable Data Cache . . . 3863
24.2 Configuring the Minion Data Cache . . . 3863

25 Slots **3865**

25.1 Execution functions . . . 3865

26 Windows **3867**

26.1 Windows Software Repository . . . 3867
26.2 Windows-specific Behaviour . . . 3880

27 Developing Salt **3883**

27.1 Overview . . . 3883
27.2 Salt Client . . . 3883
27.3 Salt Master . . . 3883
27.4 Salt Minion . . . 3885
27.5 A Note on ClearFuncs vs. AESFuncs . . . 3886
27.6 Contributing . . . 3886
27.7 Deprecating Code . . . 3892
27.8 Installing Salt for development . . . 3893
27.9 GitHub Labels and Milestones . . . 3898
27.10 Logging Internals . . . 3902
27.11 Package Providers . . . 3902
27.12 Pull Requests . . . 3906
27.13 Reporting Bugs . . . 3908
27.14 Salt Topology . . . 3909
27.15 Developing Salt Tutorial . . . 3909
27.16 Modular Systems . . . 3912
27.17 Salt Extend . . . 3927
27.18 Salt's Test Suite . . . 3929
27.19 Integration Tests . . . 3935
27.20 Writing Unit Tests . . . 3946
27.21 SaltStack Git Policy . . . 3960
27.22 Salt Conventions . . . 3961
27.23 Salt code and internals . . . 3996
27.24 Salt Community Projects . . . 4004

28 Release Notes **4007**

28.1 Latest Branch Release . . . 4007
28.2 Previous Releases . . . 4007

29 Venafi Tools for Salt **5397**

29.1 Introduction . . . 5397
29.2 Runner Functions . . . 5397
29.3 Transfer certificate to a minion . . . 5398

30 Glossary **5401**

Salt Module Index **5405**

Index **5417**

CHAPTER

ONE

INTRODUCTION TO SALT

We're not just talking about NaCl.

1.1 The 30 second summary

Salt is:

- **A configuration management system.** Salt is capable of maintaining remote nodes in defined states. For example, it can ensure that specific packages are installed and that specific services are running.
- **A distributed remote execution system used to execute commands and query data on remote nodes.** Salt can query and execute commands either on individual nodes or by using an arbitrary selection criteria.

It was developed in order to bring the best solutions found in the world of remote execution together and make them better, faster, and more malleable. Salt accomplishes this through its ability to handle large loads of information, and not just dozens but hundreds and even thousands of individual servers quickly through a simple and manageable interface.

1.2 Simplicity

Providing versatility between massive scale deployments and smaller systems may seem daunting, but Salt is very simple to set up and maintain, regardless of the size of the project. The architecture of Salt is designed to work with any number of servers, from a handful of local network systems to international deployments across different data centers. The topology is a simple server/client model with the needed functionality built into a single set of daemons. While the default configuration will work with little to no modification, Salt can be fine tuned to meet specific needs.

1.3 Parallel execution

The core functions of Salt:

- enable commands to remote systems to be called in parallel rather than serially
- use a secure and encrypted protocol
- use the smallest and fastest network payloads possible
- provide a simple programming interface

Salt also introduces more granular controls to the realm of remote execution, allowing systems to be targeted not just by hostname, but also by system properties.

1.4 Builds on proven technology

Salt takes advantage of a number of technologies and techniques. The networking layer is built with the excellent ZeroMQ networking library, so the Salt daemon includes a viable and transparent AMQ broker. Salt uses public keys for authentication with the master daemon, then uses faster AES encryption for payload communication; authentication and encryption are integral to Salt. Salt takes advantage of communication via msgpack, enabling fast and light network traffic.

1.5 Python client interface

In order to allow for simple expansion, Salt execution routines can be written as plain Python modules. The data collected from Salt executions can be sent back to the master server, or to any arbitrary program. Salt can be called from a simple Python API, or from the command line, so that Salt can be used to execute one-off commands as well as operate as an integral part of a larger application.

1.6 Fast, flexible, scalable

The result is a system that can execute commands at high speed on target server groups ranging from one to very many servers. Salt is very fast, easy to set up, amazingly malleable and provides a single remote execution architecture that can manage the diverse requirements of any number of servers. The Salt infrastructure brings together the best of the remote execution world, amplifies its capabilities and expands its range, resulting in a system that is as versatile as it is practical, suitable for any network.

1.7 Open

Salt is developed under the Apache 2.0 license, and can be used for open and proprietary projects. Please submit your expansions back to the Salt project so that we can all benefit together as Salt grows. Please feel free to sprinkle Salt around your systems and let the deliciousness come forth.

1.8 Salt Community

Join the Salt!

There are many ways to participate in and communicate with the Salt community.

Salt has an active IRC channel and a mailing list.

1.9 Mailing List

Join the salt-users mailing list. It is the best place to ask questions about Salt and see whats going on with Salt development! The Salt mailing list is hosted by Google Groups. It is open to new members.

Additionally, all users of Salt should be subscribed to the Announcements mailing list which contains important updates about Salt, such as new releaes and security-related announcements. This list is low-traffic.

1.10 IRC

The `#salt` IRC channel is hosted on the popular Freenode network. You can use the Freenode webchat client right from your browser.

Logs of the IRC channel activity are being collected courtesy of Moritz Lenz.

If you wish to discuss the development of Salt itself join us in `#salt-devel`.

1.11 Follow on Github

The Salt code is developed via Github. Follow Salt for constant updates on what is happening in Salt development:

https://github.com/saltstack/salt

Long-term planning and strategic decisions are handled via Salt Enhancement Proposals and can be found on GitHub.

1.12 Blogs

SaltStack Inc. keeps a blog with recent news and advancements:

http://www.saltstack.com/blog/

1.13 Example Salt States

The official `salt-states` repository is: https://github.com/saltstack/salt-states

A few examples of salt states from the community:

- https://github.com/blast-hardcheese/blast-salt-states
- https://github.com/kevingranade/kevingranade-salt-state
- https://github.com/uggedal/states
- https://github.com/mattmcclean/salt-openstack/tree/master/salt
- https://github.com/rentalita/ubuntu-setup/
- https://github.com/brutasse/states
- https://github.com/bclermont/states
- https://github.com/pcrews/salt-data

1.14 Follow on Open Hub

https://www.openhub.net/p/salt

1.15 Other community links

- Salt Stack Inc.
- Subreddit
- YouTube
- Facebook
- Twitter
- Wikipedia page
- Stack Overflow

1.16 Hack the Source

If you want to get involved with the development of source code or the documentation efforts, please review the *contributing documentation*!

CHAPTER

TWO

INSTALLATION

This section contains instructions to install Salt. If you are setting up your environment for the first time, you should install a Salt master on a dedicated management server or VM, and then install a Salt minion on each system that you want to manage using Salt. For now you don't need to worry about your *architecture*, you can easily add components and modify your configuration later without needing to reinstall anything.

The general installation process is as follows:

1. Install a Salt master using the instructions for your platform or by running the Salt bootstrap script. If you use the bootstrap script, be sure to include the `-M` option to install the Salt master.
2. Make sure that your Salt minions can *find the Salt master*.
3. Install the Salt minion on each system that you want to manage.
4. Accept the Salt *minion keys* after the Salt minion connects.

After this, you should be able to run a simple command and receive salt version returns from all connected Salt minions.

```
salt '*' test.version
```

2.1 Quick Install

On most distributions, you can set up a **Salt Minion** with the *Salt bootstrap*.

2.2 Platform-specific Installation Instructions

These guides go into detail how to install Salt on a given platform.

2.2.1 Arch Linux

Installation

Salt (stable) is currently available via the Arch Linux Official repositories. There are currently -git packages available in the Arch User repositories (AUR) as well.

Stable Release

Install Salt stable releases from the Arch Linux Official repositories as follows:

```
pacman -S salt
```

Tracking develop

To install the bleeding edge version of Salt (**may include bugs!**), use the -git package. Installing the -git package as follows:

```
wget https://aur.archlinux.org/packages/sa/salt-git/salt-git.tar.gz
tar xf salt-git.tar.gz
cd salt-git/
makepkg -is
```

Note: yaourt

If a tool such as Yaourt is used, the dependencies will be gathered and built automatically.

The command to install salt using the yaourt tool is:

```
yaourt salt-git
```

Post-installation tasks

systemd

Activate the Salt Master and/or Minion via `systemctl` as follows:

```
systemctl enable salt-master.service
systemctl enable salt-minion.service
```

Start the Master

Once you've completed all of these steps you're ready to start your Salt Master. You should be able to start your Salt Master now using the command seen here:

```
systemctl start salt-master
```

Now go to the *Configuring Salt* page.

2.2.2 Debian GNU/Linux / Raspbian

Debian GNU/Linux distribution and some derivatives such as Raspbian already have included Salt packages to their repositories. However, current stable Debian release contains old outdated Salt releases. It is recommended to use SaltStack repository for Debian as described *below*.

Installation from official Debian and Raspbian repositories is described *here*.

Installation from the Official SaltStack Repository

Packages for Debian 9 (Stretch) and Debian 8 (Jessie) are available in the Official SaltStack repository.

Instructions are at https://repo.saltstack.com/#debian.

Note: Regular security support for Debian 7 ended on April 25th 2016. As a result, 2016.3.1 and 2015.8.10 will be the last Salt releases for which Debian 7 packages are created.

Installation from the Debian / Raspbian Official Repository

The Debian distributions contain mostly old Salt packages built by the Debian Salt Team. You can install Salt components directly from Debian but it is recommended to use the instructions above for the packages from the official Salt repository.

On Jessie there is an option to install Salt minion from Stretch with *python-tornado* dependency from *jessie-backports* repositories.

To install fresh release of Salt minion on Jessie:

1. Add *jessie-backports* and *stretch* repositories:

   Debian:

   ```
   echo 'deb http://httpredir.debian.org/debian jessie-backports main' >> /etc/apt/
   ↪sources.list
   echo 'deb http://httpredir.debian.org/debian stretch main' >> /etc/apt/sources.
   ↪list
   ```

   Raspbian:

   ```
   echo 'deb http://archive.raspbian.org/raspbian/ stretch main' >> /etc/apt/sources.
   ↪list
   ```

2. Make Jessie a default release:

   ```
   echo 'APT::Default-Release "jessie";' > /etc/apt/apt.conf.d/10apt
   ```

3. Install Salt dependencies:

   Debian:

   ```
   apt-get update
   apt-get install python-zmq python-systemd/jessie-backports python-tornado/jessie-
   ↪backports salt-common/stretch
   ```

   Raspbian:

   ```
   apt-get update
   apt-get install python-zmq python-tornado/stretch salt-common/stretch
   ```

4. Install Salt minion package from Latest Debian Release:

   ```
   apt-get install salt-minion/stretch
   ```

Install Packages

Install the Salt master, minion or other packages from the repository with the *apt-get* command. These examples each install one of Salt components, but more than one package name may be given at a time:

- `apt-get install salt-api`
- `apt-get install salt-cloud`
- `apt-get install salt-master`
- `apt-get install salt-minion`
- `apt-get install salt-ssh`
- `apt-get install salt-syndic`

Post-installation tasks

Now, go to the *Configuring Salt* page.

2.2.3 Arista EOS Salt minion installation guide

The Salt minion for Arista EOS is distributed as a SWIX extension and can be installed directly on the switch. The EOS network operating system is based on old Fedora distributions and the installation of the `salt-minion` requires backports. This SWIX extension contains the necessary backports, together with the Salt basecode.

Note: This SWIX extension has been tested on Arista DCS-7280SE-68-R, running EOS 4.17.5M and vEOS 4.18.3F.

Important Notes

This package is in beta, make sure to test it carefully before running it in production.

If confirmed working correctly, please report and add a note on this page with the platform model and EOS version.

If you want to uninstall this package, please refer to the *uninstalling* section.

Installation from the Official SaltStack Repository

Download the swix package and save it to flash.

```
veos#copy https://salt-eos.netops.life/salt-eos-latest.swix flash:
veos#copy https://salt-eos.netops.life/startup.sh flash:
```

Install the Extension

Copy the Salt package to extension

```
veos#copy flash:salt-eos-latest.swix extension:
```

Install the SWIX

```
veos#extension salt-eos-latest.swix force
```

Verify the installation

```
veos#show extensions | include salt-eos
     salt-eos-2017-07-19.swix      1.0.11/1.fc25        A, F                27
```

Change the Salt master IP address or FQDN, by edit the variable (SALT_MASTER)

```
veos#bash vi /mnt/flash/startup.sh
```

Make sure you enable the eAPI with unix-socket

```
veos(config)#management api http-commands
         protocol unix-socket
         no shutdown
```

Post-installation tasks

Generate Keys and host record and start Salt minion

```
veos#bash
#sudo /mnt/flash/startup.sh
```

`salt-minion` should be running

Copy the installed extensions to boot-extensions

```
veos#copy installed-extensions boot-extensions
```

Apply event-handler to let EOS start salt-minion during boot-up

```
veos(config)#event-handler boot-up-script
   trigger on-boot
   action bash sudo /mnt/flash/startup.sh
```

For more specific installation details of the `salt-minion`, please refer to *Configuring Salt*.

Uninstalling

If you decide to uninstall this package, the following steps are recommended for safety:

1. Remove the extension from boot-extensions

```
veos#bash rm /mnt/flash/boot-extensions
```

2. Remove the extension from extensions folder

```
veos#bash rm /mnt/flash/.extensions/salt-eos-latest.swix
```

2. Remove boot-up script

```
veos(config)#no event-handler boot-up-script
```

Additional Information

This SWIX extension contains the following RPM packages:

```
libsodium-1.0.11-1.fc25.i686.rpm
libstdc++-6.2.1-2.fc25.i686.rpm
openpgm-5.2.122-6.fc24.i686.rpm
python-Jinja2-2.8-0.i686.rpm
python-PyYAML-3.12-0.i686.rpm
python-babel-0.9.6-5.fc18.noarch.rpm
python-backports-1.0-3.fc18.i686.rpm
python-backports-ssl_match_hostname-3.4.0.2-1.fc18.noarch.rpm
python-backports_abc-0.5-0.i686.rpm
python-certifi-2016.9.26-0.i686.rpm
python-chardet-2.0.1-5.fc18.noarch.rpm
python-crypto-1.4.1-1.noarch.rpm
python-crypto-2.6.1-1.fc18.i686.rpm
python-futures-3.1.1-1.noarch.rpm
python-jtextfsm-0.3.1-0.noarch.rpm
python-kitchen-1.1.1-2.fc18.noarch.rpm
python-markupsafe-0.18-1.fc18.i686.rpm
python-msgpack-python-0.4.8-0.i686.rpm
python-napalm-base-0.24.3-1.noarch.rpm
python-napalm-eos-0.6.0-1.noarch.rpm
python-netaddr-0.7.18-0.noarch.rpm
python-pyeapi-0.7.0-0.noarch.rpm
python-salt-2017.7.0_1414_g2fb986f-1.noarch.rpm
python-singledispatch-3.4.0.3-0.i686.rpm
python-six-1.10.0-0.i686.rpm
python-tornado-4.4.2-0.i686.rpm
python-urllib3-1.5-7.fc18.noarch.rpm
python2-zmq-15.3.0-2.fc25.i686.rpm
zeromq-4.1.4-5.fc25.i686.rpm
```

2.2.4 Fedora

Beginning with version 0.9.4, Salt has been available in the primary Fedora repositories and EPEL. It is installable using `yum` or `dnf`, depending on your version of Fedora.

Note: Released versions of Salt starting with `2015.5.2` through `2016.3.2` do not have Fedora packages available though EPEL. To install a version of Salt within this release array, please use SaltStack's Bootstrap Script and use the git method of installing Salt using the version's associated release tag.

Release `2016.3.3` and onward will have packaged versions available via EPEL.

WARNING: Fedora 19 comes with systemd 204. Systemd has known bugs fixed in later revisions that prevent the salt-master from starting reliably or opening the network connections that it needs to. It's not likely that a salt-master will start or run reliably on any distribution that uses systemd version 204 or earlier. Running salt-minions should be OK.

Installation

Salt can be installed using `yum` and is available in the standard Fedora repositories.

Stable Release

Salt is packaged separately for the minion and the master. It is necessary only to install the appropriate package for the role the machine will play. Typically, there will be one master and multiple minions.

```
yum install salt-master
yum install salt-minion
```

Installing from `updates-testing`

When a new Salt release is packaged, it is first admitted into the `updates-testing` repository, before being moved to the stable repo.

To install from `updates-testing`, use the `enablerepo` argument for yum:

```
yum --enablerepo=updates-testing install salt-master
yum --enablerepo=updates-testing install salt-minion
```

Installation Using pip

Since Salt is on PyPI, it can be installed using pip, though most users prefer to install using a package manager.

Installing from pip has a few additional requirements:

- Install the group 'Development Tools', `dnf groupinstall 'Development Tools'`
- Install the 'zeromq-devel' package if it fails on linking against that afterwards as well.

A pip install does not make the init scripts or the /etc/salt directory, and you will need to provide your own systemd service unit.

Installation from pip:

```
pip install salt
```

> **Warning:** If installing from pip (or from source using `setup.py install`), be advised that the `yum-utils` package is needed for Salt to manage packages. Also, if the Python dependencies are not already installed, then you will need additional libraries/tools installed to build some of them. More information on this can be found *here*.

Post-installation tasks

Master

To have the Master start automatically at boot time:

```
systemctl enable salt-master.service
```

To start the Master:

```
systemctl start salt-master.service
```

Minion

To have the Minion start automatically at boot time:

```
systemctl enable salt-minion.service
```

To start the Minion:

```
systemctl start salt-minion.service
```

Now go to the *Configuring Salt* page.

2.2.5 FreeBSD

Installation

Salt is available in the FreeBSD ports tree at sysutils/py-salt.

FreeBSD binary repo

For Python 2.7 use:

Install Salt via the official package repository. Salt is packaged both as a Python 2.7 or 3.6 version.

```
pkg install py27-salt
```

For Python 3.6 use:

```
pkg install py36-salt
```

FreeBSD ports

Installation from ports:

```
cd /usr/ports/sysutils/py-salt
make install
```

Python 3.6 can be used by setting default Python version to 3.6:

```
echo "DEFAULT_VERSIONS+= python=3.6" >> /etc/make.conf
```

Post-installation tasks

rc.conf

Activate the Salt Master in `/etc/rc.conf`:

```
sysrc salt_master_enable="YES"
```

Start the Master

Start the Salt Master as follows:

```
service salt_master start
```

rc.conf

Activate the Salt Minion in `/etc/rc.conf`:

```
sysrc salt_minion_enable="YES"
```

Start the Minion

Start the Salt Minion as follows:

```
service salt_minion start
```

Now go to the *Configuring Salt* page.

2.2.6 Gentoo

Salt can be easily installed on Gentoo via Portage:

```
emerge app-admin/salt
```

Post-installation tasks

Now go to the *Configuring Salt* page.

2.2.7 OpenBSD

Salt was added to the OpenBSD ports tree on Aug 10th 2013. It has been tested on OpenBSD 5.5 onwards.

Salt is dependent on the following additional ports. These will be installed as dependencies of the `sysutils/salt` port:

```
devel/py-futures
devel/py-progressbar
net/py-msgpack
net/py-zmq
security/py-crypto
security/py-M2Crypto
textproc/py-MarkupSafe
textproc/py-yaml
www/py-jinja2
www/py-requests
www/py-tornado
```

Installation

To install Salt from the OpenBSD pkg repo, use the command:

```
pkg_add salt
```

Post-installation tasks

Master

To have the Master start automatically at boot time:

```
rcctl enable salt_master
```

To start the Master:

```
rcctl start salt_master
```

Minion

To have the Minion start automatically at boot time:

```
rcctl enable salt_minion
```

To start the Minion:

```
rcctl start salt_minion
```

Now go to the *Configuring Salt* page.

2.2.8 macOS

Installation from the Official SaltStack Repository

Latest stable build from the selected branch:

The output of `md5 <salt pkg>` should match the contents of the corresponding md5 file.

Earlier builds from supported branches

Archived builds from unsupported branches

Installation from Homebrew

```
brew install saltstack
```

It should be noted that Homebrew explicitly discourages the use of sudo:

> Homebrew is designed to work without using sudo. You can decide to use it but we strongly recommend not to do so. If you have used sudo and run into a bug then it is likely to be the cause. Please don't file a bug report unless you can reproduce it after reinstalling Homebrew from scratch without using sudo

Installation from MacPorts

Macports isolates its dependencies from the OS, and installs salt in /opt/local by default, with config files under /opt/local/etc/salt. For best results, add /opt/local/bin to your PATH.

```
sudo port install salt
```

Variants allow selection of python version used to run salt, defaulting to python27, but also supporting python34, python35, and python36. To install salt with Python 3.6, use the python36 variant, for example:

```
sudo port install salt @python36
```

Startup items (for master, minion, and rest-cherrypy API gateway, respectively) are installed by subport targets. These will register launchd LaunchDaemons as org.macports.salt-minion, for example, to trigger automatic startup of the salt-minion through launchd. LaunchDaemons for salt can be started and stopped without reboot using the macprots load and unload commands.

```
sudo port install salt-master salt-minion salt-api
sudo port load salt-master salt-minion salt-api
```

Installation from Pip

When only using the macOS system's pip, install this way:

```
sudo pip install salt
```

Salt-Master Customizations

Note: Salt master on macOS is not tested or supported by SaltStack. See SaltStack Platform Support for more information.

To run salt-master on macOS, sudo add this configuration option to the /etc/salt/master file:

```
max_open_files: 8192
```

On versions previous to macOS 10.10 (Yosemite), increase the root user maxfiles limit:

```
sudo launchctl limit maxfiles 4096 8192
```

Note: On macOS 10.10 (Yosemite) and higher, maxfiles should not be adjusted. The default limits are sufficient in all but the most extreme scenarios. Overriding these values with the setting below will cause system instability!

Now the salt-master should run without errors:

```
sudo salt-master --log-level=all
```

Post-installation tasks

Now go to the *Configuring Salt* page.

2.2.9 RHEL / CentOS / Scientific Linux / Amazon Linux / Oracle Linux

Salt should work properly with all mainstream derivatives of Red Hat Enterprise Linux, including CentOS, Scientific Linux, Oracle Linux, and Amazon Linux. Report any bugs or issues on the issue tracker.

Installation from the Official SaltStack Repository

Packages for Redhat, CentOS, and Amazon Linux are available in the SaltStack Repository.

- Red Hat / CentOS
- Amazon Linux

Note: As of 2015.8.0, EPEL repository is no longer required for installing on RHEL systems. SaltStack repository provides all needed dependencies.

Warning: If installing on Red Hat Enterprise Linux 7 with disabled (not subscribed on) 'RHEL Server Releases' or 'RHEL Server Optional Channel' repositories, append CentOS 7 GPG key URL to SaltStack yum repository configuration to install required base packages:

```
[saltstack-repo]
name=SaltStack repo for Red Hat Enterprise Linux $releasever
baseurl=https://repo.saltstack.com/yum/redhat/$releasever/$basearch/latest
enabled=1
gpgcheck=1
gpgkey=https://repo.saltstack.com/yum/redhat/$releasever/$basearch/latest/SALTSTACK-
↪GPG-KEY.pub
       https://repo.saltstack.com/yum/redhat/$releasever/$basearch/latest/base/RPM-
↪GPG-KEY-CentOS-7
```

Note: `systemd` and `systemd-python` are required by Salt, but are not installed by the Red Hat 7 `@base` installation or by the Salt installation. These dependencies might need to be installed before Salt.

Installation from the Community-Maintained Repository

Beginning with version 0.9.4, Salt has been available in EPEL.

Note: Packages in this repository are built by community, and it can take a little while until the latest stable SaltStack release become available.

RHEL/CentOS 6 and 7, Scientific Linux, etc.

Warning: Salt 2015.8 is currently not available in EPEL due to unsatisfied dependencies: `python-crypto` 2.6.1 or higher, and `python-tornado` version 4.2.1 or higher. These packages are not currently available in EPEL for Red Hat Enterprise Linux 6 and 7.

Enabling EPEL

If the EPEL repository is not installed on your system, you can download the RPM for RHEL/CentOS 6 or for RHEL/CentOS 7 and install it using the following command:

```
rpm -Uvh epel-release-X-Y.rpm
```

Replace `epel-release-X-Y.rpm` with the appropriate filename.

Installing Stable Release

Salt is packaged separately for the minion and the master. It is necessary to install only the appropriate package for the role the machine will play. Typically, there will be one master and multiple minions.

- `yum install salt-master`
- `yum install salt-minion`
- `yum install salt-ssh`
- `yum install salt-syndic`
- `yum install salt-cloud`

Installing from `epel-testing`

When a new Salt release is packaged, it is first admitted into the `epel-testing` repository, before being moved to the stable EPEL repository.

To install from `epel-testing`, use the `enablerepo` argument for `yum`:

```
yum --enablerepo=epel-testing install salt-minion
```

Installation Using pip

Since Salt is on PyPI, it can be installed using pip, though most users prefer to install using RPM packages (which can be installed from EPEL).

Installing from pip has a few additional requirements:

- Install the group 'Development Tools', `yum groupinstall 'Development Tools'`
- Install the 'zeromq-devel' package if it fails on linking against that afterwards as well.

A pip install does not make the init scripts or the /etc/salt directory, and you will need to provide your own systemd service unit.

Installation from pip:

```
pip install salt
```

Warning: If installing from pip (or from source using `setup.py install`), be advised that the `yum-utils` package is needed for Salt to manage packages. Also, if the Python dependencies are not already installed, then you will need additional libraries/tools installed to build some of them. More information on this can be found *here*.

ZeroMQ 4

We recommend using ZeroMQ 4 where available. SaltStack provides ZeroMQ 4.0.5 and `pyzmq` 14.5.0 in the *SaltStack Repository*.

If this repository is added *before* Salt is installed, then installing either `salt-master` or `salt-minion` will automatically pull in ZeroMQ 4.0.5, and additional steps to upgrade ZeroMQ and pyzmq are unnecessary.

Package Management

Salt's interface to *`yum`* makes heavy use of the **repoquery** utility, from the yum-utils package. This package will be installed as a dependency if salt is installed via EPEL. However, if salt has been installed using pip, or a host is being managed using salt-ssh, then as of version 2014.7.0 yum-utils will be installed automatically to satisfy this dependency.

Post-installation tasks

Master

To have the Master start automatically at boot time:

RHEL/CentOS 5 and 6

```
chkconfig salt-master on
```

RHEL/CentOS 7

```
systemctl enable salt-master.service
```

To start the Master:

RHEL/CentOS 5 and 6

```
service salt-master start
```

RHEL/CentOS 7

```
systemctl start salt-master.service
```

Minion

To have the Minion start automatically at boot time:

RHEL/CentOS 5 and 6

```
chkconfig salt-minion on
```

RHEL/CentOS 7

```
systemctl enable salt-minion.service
```

To start the Minion:

RHEL/CentOS 5 and 6

```
service salt-minion start
```

RHEL/CentOS 7

```
systemctl start salt-minion.service
```

Now go to the *Configuring Salt* page.

2.2.10 Solaris

Salt is known to work on Solaris but community packages are unmaintained.

It is possible to install Salt on Solaris by using *setuptools*.

For example, to install the develop version of salt:

```
git clone https://github.com/saltstack/salt
cd salt
sudo python setup.py install --force
```

Note: SaltStack does offer commercial support for Solaris which includes packages.

2.2.11 Ubuntu

Installation from the Official SaltStack Repository

Packages for Ubuntu 16 (Xenial), Ubuntu 14 (Trusty), and Ubuntu 12 (Precise) are available in the SaltStack repository.

Instructions are at https://repo.saltstack.com/#ubuntu.

Install Packages

Install the Salt master, minion or other packages from the repository with the *apt-get* command. These examples each install one of Salt components, but more than one package name may be given at a time:

- `apt-get install salt-api`
- `apt-get install salt-cloud`
- `apt-get install salt-master`
- `apt-get install salt-minion`
- `apt-get install salt-ssh`
- `apt-get install salt-syndic`

Post-installation tasks

Now go to the *Configuring Salt* page.

2.2.12 Windows

Salt has full support for running the Salt minion on Windows. You must connect Windows Salt minions to a Salt master on a supported operating system to control your Salt Minions.

Many of the standard Salt modules have been ported to work on Windows and many of the Salt States currently work on Windows as well.

Installation from the Official SaltStack Repository

Latest stable build from the selected branch:

The output of `md5sum <salt minion exe>` should match the contents of the corresponding md5 file.

Earlier builds from supported branches

Archived builds from unsupported branches

Note: The installation executable installs dependencies that the Salt minion requires.

The 64bit installer has been tested on Windows 7 64bit and Windows Server 2008R2 64bit. The 32bit installer has been tested on Windows 2008 Server 32bit. Please file a bug report on our GitHub repo if issues for other platforms are found.

There are installers available for Python 2 and Python 3.

The installer will detect previous installations of Salt and ask if you would like to remove them. Clicking OK will remove the Salt binaries and related files but leave any existing config, cache, and PKI information.

Salt Minion Installation

If the system is missing the appropriate version of the Visual C++ Redistributable (vcredist) the user will be prompted to install it. Click `OK` to install the vcredist. Click `Cancel` to abort the installation without making modifications to the system.

If Salt is already installed on the system the user will be prompted to remove the previous installation. Click `OK` to uninstall Salt without removing the configuration, PKI information, or cached files. Click `Cancel` to abort the installation before making any modifications to the system.

After the Welcome and the License Agreement, the installer asks for two bits of information to configure the minion; the master hostname and the minion name. The installer will update the minion config with these options.

If the installer finds an existing minion config file, these fields will be populated with values from the existing config, but they will be grayed out. There will also be a checkbox to use the existing config. If you continue, the existing config will be used. If the checkbox is unchecked, default values are displayed and can be changed. If you continue, the existing config file in `c:\salt\conf` will be removed along with the `c:\salt\conf\minion.d` directory. The values entered will be used with the default config.

The final page allows you to start the minion service and optionally change its startup type. By default, the minion is set to `Automatic`. You can change the minion start type to `Automatic (Delayed Start)` by checking the 'Delayed Start' checkbox.

Note: Highstates that require a reboot may fail after reboot because salt continues the highstate before Windows has finished the booting process. This can be fixed by changing the startup type to 'Automatic (Delayed Start)'. The drawback is that it may increase the time it takes for the 'salt-minion' service to actually start.

The `salt-minion` service will appear in the Windows Service Manager and can be managed there or from the command line like any other Windows service.

```
sc start salt-minion
net start salt-minion
```

Installation Prerequisites

Most Salt functionality should work just fine right out of the box. A few Salt modules rely on PowerShell. The minimum version of PowerShell required for Salt is version 3. If you intend to work with DSC then Powershell version 5 is the minimum.

Silent Installer Options

The installer can be run silently by providing the `/S` option at the command line. The installer also accepts the following options for configuring the Salt Minion silently:

Option	Description
`/master=`	A string value to set the IP address or hostname of the master. Default value is 'salt'. You can pass a single master or a comma-separated list of masters. Setting the master will cause the installer to use the default config or a custom config if defined.
`/minion-name=`	A string value to set the minion name. Default value is 'hostname'. Setting the minion name causes the installer to use the default config or a custom config if defined.
`/start-minion=`	Either a 1 or 0. '1' will start the salt-minion service, '0' will not. Default is to start the service after installation.
`/start-minion-delayed`	Set the minion start type to `Automatic (Delayed Start)`.
`/default-config`	Overwrite the existing config if present with the default config for salt. Default is to use the existing config if present. If `/master` and/or `/minion-name` is passed, those values will be used to update the new default config.
`/custom-config=`	A string value specifying the name of a custom config file in the same path as the installer or the full path to a custom config file. If `/master` and/or `/minion-name` is passed, those values will be used to update the new custom config.
`/S`	Runs the installation silently. Uses the above settings or the defaults.
`/?`	Displays command line help.

Note: `/start-service` has been deprecated but will continue to function as expected for the time being.

Note: `/default-config` and `/custom-config=` will backup an existing config if found. A timestamp and a `.bak` extension will be added. That includes the `minion` file and the `minion.d` directory.

Here are some examples of using the silent installer:

```
# Install the Salt Minion
# Configure the minion and start the service

Salt-Minion-2017.7.1-Py2-AMD64-Setup.exe /S /master=yoursaltmaster /minion-
↪name=yourminionname
```

```
# Install the Salt Minion
# Configure the minion but don't start the minion service

Salt-Minion-2017.7.1-Py3-AMD64-Setup.exe /S /master=yoursaltmaster /minion-
↪name=yourminionname /start-minion=0
```

```
# Install the Salt Minion
# Configure the minion using a custom config and configuring multimaster

Salt-Minion-2017.7.1-Py3-AMD64-Setup.exe /S /custom-config=windows_minion /
↪master=prod_master1,prod_master2
```

Running the Salt Minion on Windows as an Unprivileged User

Notes:

- These instructions were tested with Windows Server 2008 R2
- They are generalizable to any version of Windows that supports a salt-minion

Create the Unprivileged User that the Salt Minion will Run As

1. Click `Start` > `Control Panel` > `User Accounts`.
2. Click `Add or remove user accounts`.
3. Click `Create new account`.
4. Enter `salt-user` (or a name of your preference) in the `New account name` field.
5. Select the `Standard user` radio button.
6. Click the `Create Account` button.
7. Click on the newly created user account.
8. Click the `Create a password` link.
9. In the `New password` and `Confirm new password` fields, provide a password (e.g "SuperSecretMinionPassword4Me!").
10. In the `Type a password hint` field, provide appropriate text (e.g. "My Salt Password").
11. Click the `Create password` button.
12. Close the `Change an Account` window.

Add the New User to the Access Control List for the Salt Folder

1. In a File Explorer window, browse to the path where Salt is installed (the default path is `C:\Salt`).
2. Right-click on the `Salt` folder and select `Properties`.
3. Click on the `Security` tab.
4. Click the `Edit` button.
5. Click the `Add` button.
6. Type the name of your designated Salt user and click the `OK` button.

7. Check the box to `Allow` the `Modify` permission.
8. Click the `OK` button.
9. Click the `OK` button to close the `Salt Properties` window.

Update the Windows Service User for the `salt-minion` Service

1. Click `Start` > `Administrative Tools` > `Services`.
2. In the Services list, right-click on `salt-minion` and select `Properties`.
3. Click the `Log On` tab.
4. Click the `This account` radio button.
5. Provide the account credentials created in section A.
6. Click the `OK` button.
7. Click the `OK` button to the prompt confirming that the user `has been granted the Log On As A Service right`.
8. Click the `OK` button to the prompt confirming that `The new logon name will not take effect until you stop and restart the service`.
9. Right-Click on `salt-minion` and select `Stop`.
10. Right-Click on `salt-minion` and select `Start`.

Building and Developing on Windows

This document will explain how to set up a development environment for Salt on Windows. The development environment allows you to work with the source code to customize or fix bugs. It will also allow you to build your own installation.

There are several scripts to automate creating a Windows installer as well as setting up an environment that facilitates developing and troubleshooting Salt code. They are located in the `pkg\windows` directory in the Salt repo (here).

Scripts:

Script	Description
`build_env_2.ps1`	A PowerShell script that sets up a Python 2 build environment
`build_env_3.ps1`	A PowerShell script that sets up a Python 3 build environment
`build_pkg.bat`	A batch file that builds a Windows installer based on the contents of the `C:\Python27` directory
`build.bat`	A batch file that fully automates the building of the Windows installer using the above two scripts

Note: The `build.bat` and `build_pkg.bat` scripts both accept a parameter to specify the version of Salt that will be displayed in the Windows installer. If no version is passed, the version will be determined using git.

Both scripts also accept an additional parameter to specify the version of Python to use. The default is 2.

Prerequisite Software

The only prerequisite is Git for Windows.

Create a Build Environment

1. Working Directory

Create a `Salt-Dev` directory on the root of `C:`. This will be our working directory. Navigate to `Salt-Dev` and clone the Salt repo from GitHub.

Open a command line and type:

```
cd \
md Salt-Dev
cd Salt-Dev
git clone https://github.com/saltstack/salt
```

Go into the `salt` directory and checkout the version of salt to work with (2016.3 or higher).

```
cd salt
git checkout 2017.7.2
```

2. Setup the Python Environment

Navigate to the `pkg\windows` directory and execute the **build_env.ps1** PowerShell script.

```
cd pkg\windows
powershell -file build_env_2.ps1
```

Note: You can also do this from Explorer by navigating to the `pkg\windows` directory, right clicking the **build_env_2.ps1** powershell script and selecting **Run with PowerShell**

This will download and install Python 2 with all the dependencies needed to develop and build Salt.

Note: If you get an error or the script fails to run you may need to change the execution policy. Open a powershell window and type the following command:

```
Set-ExecutionPolicy RemoteSigned
```

3. Salt in Editable Mode

Editable mode allows you to more easily modify and test the source code. For more information see the Pip documentation.

Navigate to the root of the `salt` directory and install Salt in editable mode with `pip`

```
cd \Salt-Dev\salt
pip install -e .
```

Note: The `.` is important

Note: If `pip` is not recognized, you may need to restart your shell to get the updated path

Note: If `pip` is still not recognized make sure that the Python Scripts folder is in the System `%PATH%`. (`C:\Python2\Scripts`)

4. Setup Salt Configuration

Salt requires a minion configuration file and a few other directories. The default config file is named `minion` located in `C:\salt\conf`. The easiest way to set this up is to copy the contents of the `salt\pkg\windows\buildenv` directory to `C:\salt`.

```
cd \
md salt
xcopy /s /e \Salt-Dev\salt\pkg\windows\buildenv\* \salt\
```

Now go into the `C:\salt\conf` directory and edit the minion config file named `minion` (no extension). You need to configure the master and id parameters in this file. Edit the following lines:

```
master: <ip or name of your master>
id: <name of your minion>
```

Create a Windows Installer

To create a Windows installer, follow steps 1 and 2 from *Create a Build Environment* above. Then proceed to 3 below:

3. Install Salt

To create the installer for Window we install Salt using Python instead of pip. Navigate to the root `salt` directory and install Salt.

```
cd \Salt-Dev\salt
python setup.py install
```

4. Create the Windows Installer

Navigate to the `pkg\windows` directory and run the `build_pkg.bat` with the build version (2017.7.2) and the Python version as parameters.

```
cd pkg\windows
build_pkg.bat 2017.7.2 2
              ^^^^^^^^ ^
                  |    |
# build version --     |
# python version ------
```

Note: If no version is passed, the `build_pkg.bat` will guess the version number using git. If the python version is not passed, the default is 2.

Creating a Windows Installer: Alternate Method (Easier)

Clone the Salt repo from GitHub into the directory of your choice. We're going to use `Salt-Dev`.

```
cd \
md Salt-Dev
cd Salt-Dev
git clone https://github.com/saltstack/salt
```

Go into the `salt` directory and checkout the version of Salt you want to build.

```
cd salt
git checkout 2017.7.2
```

Then navigate to `pkg\windows` and run the `build.bat` script with the version you're building.

```
cd pkg\windows
build.bat 2017.7.2 3
          ^^^^^^^^ ^
              |    |
# build version    |
# python version --
```

This will install everything needed to build a Windows installer for Salt using Python 3. The binary will be in the `salt\pkg\windows\installer` directory.

Testing the Salt minion

1. Create the directory `C:\salt` (if it doesn't exist already)
2. **Copy the example `conf` and `var` directories from** `pkg\windows\buildenv` into `C:\salt`
3. Edit `C:\salt\conf\minion`

   ```
   master: ipaddress or hostname of your salt-master
   ```

4. Start the salt-minion

   ```
   cd C:\Python27\Scripts
   python salt-minion -l debug
   ```

5. On the salt-master accept the new minion's key

   ```
   sudo salt-key -A
   ```

   This accepts all unaccepted keys. If you're concerned about security just accept the key for this specific minion.

6. Test that your minion is responding

   On the salt-master run:

```
sudo salt '*' test.version
```

You should get the following response: `{'your minion hostname': True}`

Packages Management Under Windows 2003

Windows Server 2003 and Windows XP have both reached End of Support. Though Salt is not officially supported on operating systems that are EoL, some functionality may continue to work.

On Windows Server 2003, you need to install optional component "WMI Windows Installer Provider" to get a full list of installed packages. If you don't have this, salt-minion can't report some installed software.

2.2.13 SUSE

Installation from the Official SaltStack Repository

Packages for SUSE 12 SP1, SUSE 12, SUSE 11, openSUSE 13 and openSUSE Leap 42.1 are available in the SaltStack Repository.

Instructions are at https://repo.saltstack.com/#suse.

Installation from the SUSE Repository

Since openSUSE 13.2, Salt 2014.1.11 is available in the primary repositories. With the release of SUSE manager 3 a new repository setup has been created. The new repo will by systemsmanagement:saltstack, which is the source for newer stable packages. For backward compatibility a linkpackage will be created to the old devel:language:python repo. All development of suse packages will be done in systemsmanagement:saltstack:testing. This will ensure that salt will be in mainline suse repo's, a stable release repo and a testing repo for further enhancements.

Installation

Salt can be installed using `zypper` and is available in the standard openSUSE/SLES repositories.

Stable Release

Salt is packaged separately for the minion and the master. It is necessary only to install the appropriate package for the role the machine will play. Typically, there will be one master and multiple minions.

```
zypper install salt-master
zypper install salt-minion
```

Post-installation tasks openSUSE

Master

To have the Master start automatically at boot time:

```
systemctl enable salt-master.service
```

To start the Master:

```
systemctl start salt-master.service
```

Minion

To have the Minion start automatically at boot time:

```
systemctl enable salt-minion.service
```

To start the Minion:

```
systemctl start salt-minion.service
```

Post-installation tasks SLES

Master

To have the Master start automatically at boot time:

```
chkconfig salt-master on
```

To start the Master:

```
rcsalt-master start
```

Minion

To have the Minion start automatically at boot time:

```
chkconfig salt-minion on
```

To start the Minion:

```
rcsalt-minion start
```

Unstable Release

openSUSE

For openSUSE Tumbleweed run the following as root:

```
zypper install salt salt-minion salt-master
```

For openSUSE 15.0 Leap run the following as root:

```
zypper install salt salt-minion salt-master
```

For openSUSE 15.1 run the following as root:

```
zypper install salt salt-minion salt-master
```

SUSE Linux Enterprise

For SLE 12 run the following as root:

```
zypper addrepo http://download.opensuse.org/repositories/systemsmanagement:/saltstack/
↪SLE_12/systemsmanagement:saltstack.repo
zypper refresh
zypper install salt salt-minion salt-master
```

For SLE 11 SP4 run the following as root:

```
zypper addrepo http://download.opensuse.org/repositories/systemsmanagement:/saltstack/
↪SLE_11_SP4/systemsmanagement:saltstack.repo
zypper refresh
zypper install salt salt-minion salt-master
```

Now go to the *Configuring Salt* page.

2.3 Initial Configuration

2.3.1 Configuring Salt

Salt configuration is very simple. The default configuration for the *master* will work for most installations and the only requirement for setting up a *minion* is to set the location of the master in the minion configuration file.

The configuration files will be installed to `/etc/salt` and are named after the respective components, `/etc/salt/master`, and `/etc/salt/minion`.

Master Configuration

By default the Salt master listens on ports 4505 and 4506 on all interfaces (0.0.0.0). To bind Salt to a specific IP, redefine the "interface" directive in the master configuration file, typically `/etc/salt/master`, as follows:

```
- #interface: 0.0.0.0
+ interface: 10.0.0.1
```

After updating the configuration file, restart the Salt master. See the *master configuration reference* for more details about other configurable options.

Minion Configuration

Although there are many Salt Minion configuration options, configuring a Salt Minion is very simple. By default a Salt Minion will try to connect to the DNS name "salt"; if the Minion is able to resolve that name correctly, no configuration is needed.

If the DNS name "salt" does not resolve to point to the correct location of the Master, redefine the "master" directive in the minion configuration file, typically `/etc/salt/minion`, as follows:

```
- #master: salt
+ master: 10.0.0.1
```

After updating the configuration file, restart the Salt minion. See the *minion configuration reference* for more details about other configurable options.

Proxy Minion Configuration

A proxy minion emulates the behaviour of a regular minion and inherits their options.

Similarly, the configuration file is `/etc/salt/proxy` and the proxy tries to connect to the DNS name "salt".

In addition to the regular minion options, there are several proxy-specific - see the *proxy minion configuration reference*.

Running Salt

1. Start the master in the foreground (to daemonize the process, pass the `-d flag`):

   ```
   salt-master
   ```

2. Start the minion in the foreground (to daemonize the process, pass the `-d flag`):

   ```
   salt-minion
   ```

Having trouble?

The simplest way to troubleshoot Salt is to run the master and minion in the foreground with `log level` set to `debug`:

```
salt-master --log-level=debug
```

For information on salt's logging system please see the *logging document*.

Run as an unprivileged (non-root) user

To run Salt as another user, set the `user` parameter in the master config file.

Additionally, ownership, and permissions need to be set such that the desired user can read from and write to the following directories (and their subdirectories, where applicable):

- /etc/salt
- /var/cache/salt
- /var/log/salt
- /var/run/salt

More information about running salt as a non-privileged user can be found *here*.

There is also a full *troubleshooting guide* available.

Key Identity

Salt provides commands to validate the identity of your Salt master and Salt minions before the initial key exchange. Validating key identity helps avoid inadvertently connecting to the wrong Salt master, and helps prevent a potential MiTM attack when establishing the initial connection.

Master Key Fingerprint

Print the master key fingerprint by running the following command on the Salt master:

```
salt-key -F master
```

Copy the `master.pub` fingerprint from the *Local Keys* section, and then set this value as the *`master_finger`* in the minion configuration file. Save the configuration file and then restart the Salt minion.

Minion Key Fingerprint

Run the following command on each Salt minion to view the minion key fingerprint:

```
salt-call --local key.finger
```

Compare this value to the value that is displayed when you run the `salt-key --finger <MINION_ID>` command on the Salt master.

Key Management

Salt uses AES encryption for all communication between the Master and the Minion. This ensures that the commands sent to the Minions cannot be tampered with, and that communication between Master and Minion is authenticated through trusted, accepted keys.

Before commands can be sent to a Minion, its key must be accepted on the Master. Run the `salt-key` command to list the keys known to the Salt Master:

```
[root@master ~]# salt-key -L
Unaccepted Keys:
alpha
bravo
charlie
delta
Accepted Keys:
```

This example shows that the Salt Master is aware of four Minions, but none of the keys has been accepted. To accept the keys and allow the Minions to be controlled by the Master, again use the `salt-key` command:

```
[root@master ~]# salt-key -A
[root@master ~]# salt-key -L
Unaccepted Keys:
Accepted Keys:
alpha
bravo
charlie
delta
```

The `salt-key` command allows for signing keys individually or in bulk. The example above, using `-A` bulk-accepts all pending keys. To accept keys individually use the lowercase of the same option, `-a keyname`.

See also:

salt-key manpage

Sending Commands

Communication between the Master and a Minion may be verified by running the `test.version` command:

```
[root@master ~]# salt alpha test.version
alpha:
    2018.3.4
```

Communication between the Master and all Minions may be tested in a similar way:

```
[root@master ~]# salt '*' test.version
alpha:
    2018.3.4
bravo:
    2018.3.4
charlie:
    2018.3.4
delta:
    2018.3.4
```

Each of the Minions should send a `2018.3.4` response as shown above, or any other salt version installed.

What's Next?

Understanding *targeting* is important. From there, depending on the way you wish to use Salt, you should also proceed to learn about *Remote Execution* and *Configuration Management*.

2.4 Additional Installation Guides

2.4.1 Salt Bootstrap

The Salt Bootstrap Script allows a user to install the Salt Minion or Master on a variety of system distributions and versions.

The Salt Bootstrap Script is a shell script is known as `bootstrap-salt.sh`. It runs through a series of checks to determine the operating system type and version. It then installs the Salt binaries using the appropriate methods.

The Salt Bootstrap Script installs the minimum number of packages required to run Salt. This means that in the event you run the bootstrap to install via package, Git will not be installed. Installing the minimum number of packages helps ensure the script stays as lightweight as possible, assuming the user will install any other required packages after the Salt binaries are present on the system.

The Salt Bootstrap Script is maintained in a separate repo from Salt, complete with its own issues, pull requests, contributing guidelines, release protocol, etc.

To learn more, please see the Salt Bootstrap repo links:

- Salt Bootstrap repo
- README: includes supported operating systems, example usage, and more.
- Contributing Guidelines
- Release Process

Note: The Salt Bootstrap script can be found in the Salt repo under the `salt/cloud/deploy/bootstrap-salt.sh` path. Any changes to this file will be overwritten! Bug fixes and feature additions must be submitted via the Salt Bootstrap repo. Please see the Salt Bootstrap Script's Release Process for more information.

2.4.2 Opening the Firewall up for Salt

The Salt master communicates with the minions using an AES-encrypted ZeroMQ connection. These communications are done over TCP ports **4505** and **4506**, which need to be accessible on the master only. This document outlines suggested firewall rules for allowing these incoming connections to the master.

Note: No firewall configuration needs to be done on Salt minions. These changes refer to the master only.

Fedora 18 and beyond / RHEL 7 / CentOS 7

Starting with Fedora 18 FirewallD is the tool that is used to dynamically manage the firewall rules on a host. It has support for IPv4/6 settings and the separation of runtime and permanent configurations. To interact with FirewallD use the command line client `firewall-cmd`.

firewall-cmd example:

```
firewall-cmd --permanent --zone=<zone> --add-port=4505-4506/tcp
```

A network zone defines the security level of trust for the the network. The user should choose an appropriate zone value for their setup. Possible values include: drop, block, public, external, dmz, work, home, internal, trusted.

Don't forget to reload after you made your changes.

```
firewall-cmd --reload
```

RHEL 6 / CentOS 6

The `lokkit` command packaged with some Linux distributions makes opening iptables firewall ports very simple via the command line. Just be careful to not lock out access to the server by neglecting to open the ssh port.

lokkit example:

```
lokkit -p 22:tcp -p 4505:tcp -p 4506:tcp
```

The `system-config-firewall-tui` command provides a text-based interface to modifying the firewall.

system-config-firewall-tui:

```
system-config-firewall-tui
```

openSUSE

Salt installs firewall rules in /etc/sysconfig/SuSEfirewall2.d/services/salt. Enable with:

```
SuSEfirewall2 open
SuSEfirewall2 start
```

If you have an older package of Salt where the above configuration file is not included, the `SuSEfirewall2` command makes opening iptables firewall ports very simple via the command line.

SuSEfirewall example:

```
SuSEfirewall2 open EXT TCP 4505
SuSEfirewall2 open EXT TCP 4506
```

The firewall module in YaST2 provides a text-based interface to modifying the firewall.

YaST2:

```
yast2 firewall
```

Windows

Windows Firewall is the default component of Microsoft Windows that provides firewalling and packet filtering. There are many 3rd party firewalls available for Windows, some of which use rules from the Windows Firewall. If you are experiencing problems see the vendor's specific documentation for opening the required ports.

The Windows Firewall can be configured using the Windows Interface or from the command line.

Windows Firewall (interface):

1. Open the Windows Firewall Interface by typing `wf.msc` at the command prompt or in a run dialog (*Windows Key + R*)
2. Navigate to **Inbound Rules** in the console tree
3. Add a new rule by clicking **New Rule…** in the Actions area
4. Change the Rule Type to **Port**. Click **Next**
5. Set the Protocol to **TCP** and specify local ports **4505-4506**. Click **Next**
6. Set the Action to **Allow the connection**. Click **Next**
7. Apply the rule to **Domain**, **Private**, and **Public**. Click **Next**
8. Give the new rule a Name, ie: **Salt**. You may also add a description. Click **Finish**

Windows Firewall (command line):

The Windows Firewall rule can be created by issuing a single command. Run the following command from the command line or a run prompt:

```
netsh advfirewall firewall add rule name="Salt" dir=in action=allow protocol=TCP⊠
↪localport=4505-4506
```

iptables

Different Linux distributions store their *iptables* (also known as netfilter) rules in different places, which makes it difficult to standardize firewall documentation. Included are some of the more common locations, but your mileage may vary.

Fedora / RHEL / CentOS:

```
/etc/sysconfig/iptables
```

Arch Linux:

```
/etc/iptables/iptables.rules
```

Debian

Follow these instructions: https://wiki.debian.org/iptables

Once you've found your firewall rules, you'll need to add the below line to allow traffic on `tcp/4505` and `tcp/4506`:

```
-A INPUT -m state --state new -m tcp -p tcp --dport 4505:4506 -j ACCEPT
```

Ubuntu

Salt installs firewall rules in /etc/ufw/applications.d/salt.ufw. Enable with:

```
ufw allow salt
```

pf.conf

The BSD-family of operating systems uses packet filter (pf). The following example describes the addition to `pf.conf` needed to access the Salt master.

```
pass in on $int_if proto tcp from any to $int_if port 4505:4506
```

Once this addition has been made to the `pf.conf` the rules will need to be reloaded. This can be done using the `pfctl` command.

```
pfctl -vf /etc/pf.conf
```

2.4.3 Whitelist communication to Master

There are situations where you want to selectively allow Minion traffic from specific hosts or networks into your Salt Master. The first scenario which comes to mind is to prevent unwanted traffic to your Master out of security concerns, but another scenario is to handle Minion upgrades when there are backwards incompatible changes between the installed Salt versions in your environment.

Here is an example *Linux iptables* ruleset to be set on the Master:

```
# Allow Minions from these networks
-I INPUT -s 10.1.2.0/24 -p tcp --dports 4505:4506 -j ACCEPT
-I INPUT -s 10.1.3.0/24 -p tcp --dports 4505:4506 -j ACCEPT
# Allow Salt to communicate with Master on the loopback interface
-A INPUT -i lo -p tcp --dports 4505:4506 -j ACCEPT
# Reject everything else
-A INPUT -p tcp --dports 4505:4506 -j REJECT
```

Note: The important thing to note here is that the `salt` command needs to communicate with the listening network socket of `salt-master` on the *loopback* interface. Without this you will see no outgoing Salt traffic from the master, even for a simple `salt '*' test.version`, because the `salt` client never reached the `salt-master` to tell it to carry out the execution.

2.4.4 Preseed Minion with Accepted Key

In some situations, it is not convenient to wait for a minion to start before accepting its key on the master. For instance, you may want the minion to bootstrap itself as soon as it comes online. You may also want to let your developers provision new development machines on the fly.

See also:

Many ways to preseed minion keys

Salt has other ways to generate and pre-accept minion keys in addition to the manual steps outlined below.

salt-cloud performs these same steps automatically when new cloud VMs are created (unless instructed not to).

salt-api exposes an HTTP call to Salt's REST API to *generate and download the new minion keys as a tarball*.

There is a general four step process to do this:

1. Generate the keys on the master:

```
root@saltmaster# salt-key --gen-keys=[key_name]
```

Pick a name for the key, such as the minion's id.

2. Add the public key to the accepted minion folder:

```
root@saltmaster# cp key_name.pub /etc/salt/pki/master/minions/[minion_id]
```

It is necessary that the public key file has the same name as your minion id. This is how Salt matches minions with their keys. Also note that the pki folder could be in a different location, depending on your OS or if specified in the master config file.

3. Distribute the minion keys.

There is no single method to get the keypair to your minion. The difficulty is finding a distribution method which is secure. For Amazon EC2 only, an AWS best practice is to use IAM Roles to pass credentials. (See blog post, http://blogs.aws.amazon.com/security/post/Tx610S2MLVZWEA/Using-IAM-roles-to-distribute-non-AWS-credentials-to-your-EC2-instances)

Security Warning

Since the minion key is already accepted on the master, distributing the private key poses a potential security risk. A malicious party will have access to your entire state tree and other sensitive data if they gain access to a preseeded minion key.

---

4. Preseed the Minion with the keys

You will want to place the minion keys before starting the salt-minion daemon:

```
/etc/salt/pki/minion/minion.pem
/etc/salt/pki/minion/minion.pub
```

Once in place, you should be able to start salt-minion and run `salt-call state.apply` or any other salt commands that require master authentication.

2.4.5 The macOS (Maverick) Developer Step By Step Guide To Salt Installation

This document provides a step-by-step guide to installing a Salt cluster consisting of one master, and one minion running on a local VM hosted on macOS.

Note: This guide is aimed at developers who wish to run Salt in a virtual machine. The official (Linux) walkthrough can be found here.

The 5 Cent Salt Intro

Since you're here you've probably already heard about Salt, so you already know Salt lets you configure and run commands on hordes of servers easily. Here's a brief overview of a Salt cluster:

- Salt works by having a "master" server sending commands to one or multiple "minion" servers. The master server is the "command center". It is going to be the place where you store your configuration files, aka: "which server is the db, which is the web server, and what libraries and software they should have installed". The minions receive orders from the master. Minions are the servers actually performing work for your business.
- Salt has two types of configuration files:

  1. the "salt communication channels" or "meta" or "config" configuration files (not official names): one for the master (usually is /etc/salt/master , **on the master server**), and one for minions (default is /etc/salt/minion or /etc/salt/minion.conf, **on the minion servers**). Those files are used to determine things like the Salt Master IP, port, Salt folder locations, etc.. If these are configured incorrectly, your minions will probably be unable to receive orders from the master, or the master will not know which software a given minion should install.

  2. the "business" or "service" configuration files (once again, not an official name): these are configuration files, ending with ".sls" extension, that describe which software should run on which server, along with particular configuration properties for the software that is being installed. These files should be created in the /srv/salt folder by default, but their location can be changed using … /etc/salt/master configuration file!

Note: This tutorial contains a third important configuration file, not to be confused with the previous two: the virtual machine provisioning configuration file. This in itself is not specifically tied to Salt, but it also contains some Salt configuration. More on that in step 3. Also note that all configuration files are YAML files. So indentation matters.

Note: Salt also works with "masterless" configuration where a minion is autonomous (in which case salt can be seen as a local configuration tool), or in "multiple master" configuration. See the documentation for more on that.

Before Digging In, The Architecture Of The Salt Cluster

Salt Master

The "Salt master" server is going to be the Mac OS machine, directly. Commands will be run from a terminal app, so Salt will need to be installed on the Mac. This is going to be more convenient for toying around with configuration files.

Salt Minion

We'll only have one ”Salt minion” server. It is going to be running on a Virtual Machine running on the Mac, using VirtualBox. It will run an Ubuntu distribution.

Step 1 - Configuring The Salt Master On Your Mac

Official Documentation

Because Salt has a lot of dependencies that are not built in macOS, we will use Homebrew to install Salt. Homebrew is a package manager for Mac, it's great, use it (for this tutorial at least!). Some people spend a lot of time installing libs by hand to better understand dependencies, and then realize how useful a package manager is once they're configuring a brand new machine and have to do it all over again. It also lets you *uninstall* things easily.

Note: Brew is a Ruby program (Ruby is installed by default with your Mac). Brew downloads, compiles, and links software. The linking phase is when compiled software is deployed on your machine. It may conflict with manually installed software, especially in the /usr/local directory. It's ok, remove the manually installed version then refresh the link by typing `brew link 'packageName'`. Brew has a `brew doctor` command that can help you troubleshoot. It's a great command, use it often. Brew requires xcode command line tools. When you run brew the first time it asks you to install them if they're not already on your system. Brew installs software in /usr/local/bin (system bins are in /usr/bin). In order to use those bins you need your $PATH to search there first. Brew tells you if your $PATH needs to be fixed.

Tip: Use the keyboard shortcut `cmd + shift + period` in the ”open” macOS dialog box to display hidden files and folders, such as .profile.

Install Homebrew

Install Homebrew here http://brew.sh/

Or just type

```
ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/
↪install)"
```

Now type the following commands in your terminal (you may want to type `brew doctor` after each to make sure everything's fine):

```
brew install python
brew install swig
brew install zmq
```

Note: zmq is ZeroMQ. It's a fantastic library used for server to server network communication and is at the core of Salt efficiency.

Install Salt

You should now have everything ready to launch this command:

```
pip install salt
```

Note: There should be no need for `sudo pip install salt`. Brew installed Python for your user, so you should have all the access. In case you would like to check, type `which python` to ensure that it's /usr/local/bin/python, and `which pip` which should be /usr/local/bin/pip.

Now type `python` in a terminal then, `import salt`. There should be no errors. Now exit the Python terminal using `exit()`.

Create The Master Configuration

If the default /etc/salt/master configuration file was not created, copy-paste it from here: http://docs.saltstack.com/ref/configuration/examples.html#configuration-examples-master

---

Note: `/etc/salt/master` is a file, not a folder.

Salt Master configuration changes. The Salt master needs a few customization to be able to run on macOS:

```
sudo launchctl limit maxfiles 4096 8192
```

In the /etc/salt/master file, change max_open_files to 8192 (or just add the line: `max_open_files: 8192` (no quote) if it doesn't already exists).

You should now be able to launch the Salt master:

```
sudo salt-master --log-level=all
```

There should be no errors when running the above command.

Note: This command is supposed to be a daemon, but for toying around, we'll keep it running on a terminal to monitor the activity.

Now that the master is set, let's configure a minion on a VM.

Step 2 - Configuring The Minion VM

The Salt minion is going to run on a Virtual Machine. There are a lot of software options that let you run virtual machines on a mac, But for this tutorial we're going to use VirtualBox. In addition to virtualBox, we will use Vagrant, which allows you to create the base VM configuration.

Vagrant lets you build ready to use VM images, starting from an OS image and customizing it using ”provisioners”. In our case, we'll use it to:

- Download the base Ubuntu image
- Install salt on that Ubuntu image (Salt is going to be the ”provisioner” for the VM).
- Launch the VM
- SSH into the VM to debug
- Stop the VM once you're done.

Install VirtualBox

Go get it here: https://www.virtualBox.org/wiki/Downloads (click on VirtualBox for macOS hosts => x86/amd64)

Install Vagrant

Go get it here: http://downloads.vagrantup.com/ and choose the latest version (1.3.5 at time of writing), then the .dmg file. Double-click to install it. Make sure the `vagrant` command is found when run in the terminal. Type `vagrant`. It should display a list of commands.

Create The Minion VM Folder

Create a folder in which you will store your minion's VM. In this tutorial, it's going to be a minion folder in the $home directory.

```
cd $home
mkdir minion
```

Initialize Vagrant

From the minion folder, type

```
vagrant init
```

This command creates a default Vagrantfile configuration file. This configuration file will be used to pass configuration parameters to the Salt provisioner in Step 3.

Import Precise64 Ubuntu Box

```
vagrant box add precise64 http://files.vagrantup.com/precise64.box
```

Note: This box is added at the global Vagrant level. You only need to do it once as each VM will use this same file.

Modify the Vagrantfile

Modify ./minion/Vagrantfile to use th precise64 box. Change the `config.vm.box` line to:

```
config.vm.box = "precise64"
```

Uncomment the line creating a host-only IP. This is the ip of your minion (you can change it to something else if that IP is already in use):

```
config.vm.network :private_network, ip: "192.168.33.10"
```

At this point you should have a VM that can run, although there won't be much in it. Let's check that.

Checking The VM

From the $home/minion folder type:

```
vagrant up
```

A log showing the VM booting should be present. Once it's done you'll be back to the terminal:

```
ping 192.168.33.10
```

The VM should respond to your ping request.

Now log into the VM in ssh using Vagrant again:

```
vagrant ssh
```

You should see the shell prompt change to something similar to `vagrant@precise64:~$` meaning you're inside the VM. From there, enter the following:

```
ping 10.0.2.2
```

Note: That ip is the ip of your VM host (the macOS host). The number is a VirtualBox default and is displayed in the log after the Vagrant ssh command. We'll use that IP to tell the minion where the Salt master is. Once you're done, end the ssh session by typing `exit`.

It's now time to connect the VM to the salt master

Step 3 - Connecting Master and Minion

Creating The Minion Configuration File

Create the `/etc/salt/minion` file. In that file, put the following lines, giving the ID for this minion, and the IP of the master:

```
master: 10.0.2.2
id: 'minion1'
file_client: remote
```

Minions authenticate with the master using keys. Keys are generated automatically if you don't provide one and can accept them later on. However, this requires accepting the minion key every time the minion is destroyed or created (which could be quite often). A better way is to create those keys in advance, feed them to the minion, and authorize them once.

Preseed minion keys

From the minion folder on your Mac run:

```
sudo salt-key --gen-keys=minion1
```

This should create two files: minion1.pem, and minion1.pub. Since those files have been created using sudo, but will be used by vagrant, you need to change ownership:

```
sudo chown youruser:yourgroup minion1.pem
sudo chown youruser:yourgroup minion1.pub
```

Then copy the .pub file into the list of accepted minions:

```
sudo cp minion1.pub /etc/salt/pki/master/minions/minion1
```

Modify Vagrantfile to Use Salt Provisioner

Let's now modify the Vagrantfile used to provision the Salt VM. Add the following section in the Vagrantfile (note: it should be at the same indentation level as the other properties):

```
# salt-vagrant config
config.vm.provision :salt do |salt|
    salt.run_highstate = true
    salt.minion_config = "/etc/salt/minion"
    salt.minion_key = "./minion1.pem"
    salt.minion_pub = "./minion1.pub"
end
```

Now destroy the vm and recreate it from the /minion folder:

```
vagrant destroy
vagrant up
```

If everything is fine you should see the following message:

```
"Bootstrapping Salt... (this may take a while)
Salt successfully configured and installed!"
```

Checking Master-Minion Communication

To make sure the master and minion are talking to each other, enter the following:

```
sudo salt '*' test.version
```

You should see your minion answering with its salt version. It's now time to do some configuration.

Step 4 - Configure Services to Install On the Minion

In this step we'll use the Salt master to instruct our minion to install Nginx.

Checking the system's original state

First, make sure that an HTTP server is not installed on our minion. When opening a browser directed at `http://192.168.33.10/` You should get an error saying the site cannot be reached.

Initialize the top.sls file

System configuration is done in `/srv/salt/top.sls` (and subfiles/folders), and then applied by running the *`state.apply`* function to have the Salt master order its minions to update their instructions and run the associated commands.

First Create an empty file on your Salt master (macOS machine):

```
touch /srv/salt/top.sls
```

When the file is empty, or if no configuration is found for our minion an error is reported:

```
sudo salt 'minion1' state.apply
```

This should return an error stating: **No Top file or external nodes data matches found**.

Create The Nginx Configuration

Now is finally the time to enter the real meat of our server's configuration. For this tutorial our minion will be treated as a web server that needs to have Nginx installed.

Insert the following lines into `/srv/salt/top.sls` (which should current be empty).

```
base:
  'minion1':
    - bin.nginx
```

Now create `/srv/salt/bin/nginx.sls` containing the following:

```
nginx:
  pkg.installed:
    - name: nginx
  service.running:
    - enable: True
    - reload: True
```

Check Minion State

Finally, run the *`state.apply`* function again:

```
sudo salt 'minion1' state.apply
```

You should see a log showing that the Nginx package has been installed and the service configured. To prove it, open your browser and navigate to http://192.168.33.10/, you should see the standard Nginx welcome page.

Congratulations!

Where To Go From Here

A full description of configuration management within Salt (sls files among other things) is available here: http://docs.saltstack.com/en/latest/index.html#configuration-management

2.4.6 running salt as normal user tutorial

Before continuing make sure you have a working Salt installation by following the *Installation* and the *configuration* instructions.

Stuck?

There are many ways to *get help from the Salt community* including our mailing list and our IRC channel #salt.

Running Salt functions as non root user

If you don't want to run salt cloud as root or even install it you can configure it to have a virtual root in your working directory.

The salt system uses the `salt.syspath` module to find the variables

If you run the salt-build, it will generated in:

```
./build/lib.linux-x86_64-2.7/salt/_syspaths.py
```

To generate it, run the command:

```
python setup.py build
```

Copy the generated module into your salt directory

```
cp ./build/lib.linux-x86_64-2.7/salt/_syspaths.py salt/_syspaths.py
```

Edit it to include needed variables and your new paths

```
# you need to edit this
ROOT_DIR = *your current dir* + '/salt/root'

# you need to edit this
INSTALL_DIR = *location of source code*

CONFIG_DIR =  ROOT_DIR + '/etc/salt'
CACHE_DIR = ROOT_DIR + '/var/cache/salt'
SOCK_DIR = ROOT_DIR + '/var/run/salt'
SRV_ROOT_DIR= ROOT_DIR + '/srv'
BASE_FILE_ROOTS_DIR = ROOT_DIR + '/srv/salt'
BASE_PILLAR_ROOTS_DIR = ROOT_DIR + '/srv/pillar'
BASE_MASTER_ROOTS_DIR = ROOT_DIR + '/srv/salt-master'
LOGS_DIR = ROOT_DIR + '/var/log/salt'
PIDFILE_DIR = ROOT_DIR + '/var/run'
CLOUD_DIR = INSTALL_DIR + '/cloud'
BOOTSTRAP = CLOUD_DIR + '/deploy/bootstrap-salt.sh'
```

Create the directory structure

```
mkdir -p root/etc/salt root/var/cache/run root/run/salt root/srv
root/srv/salt root/srv/pillar root/srv/salt-master root/var/log/salt root/var/run
```

Populate the configuration files:

```
cp -r conf/* root/etc/salt/
```

Edit your `root/etc/salt/master` configuration that is used by salt-cloud:

```
user: *your user name*
```

Run like this:

```
PYTHONPATH=`pwd` scripts/salt-cloud
```

2.4.7 Standalone Minion

Since the Salt minion contains such extensive functionality it can be useful to run it standalone. A standalone minion can be used to do a number of things:

- Use salt-call commands on a system without connectivity to a master
- Masterless States, run states entirely from files local to the minion

---

Note: When running Salt in masterless mode, do not run the salt-minion daemon. Otherwise, it will attempt to connect to a master and fail. The salt-call command stands on its own and does not need the salt-minion daemon.

Minion Configuration

Throughout this document there are several references to setting different options to configure a masterless Minion. Salt Minions are easy to configure via a configuration file that is located, by default, in `/etc/salt/minion`. Note, however, that on FreeBSD systems, the minion configuration file is located in `/usr/local/etc/salt/minion`.

You can learn more about minion configuration options in the *Configuring the Salt Minion* docs.

Telling Salt Call to Run Masterless

The salt-call command is used to run module functions locally on a minion instead of executing them from the master. Normally the salt-call command checks into the master to retrieve file server and pillar data, but when running standalone salt-call needs to be instructed to not check the master for this data. To instruct the minion to not look for a master when running salt-call the *`file_client`* configuration option needs to be set. By default the *`file_client`* is set to `remote` so that the minion knows that file server and pillar data are to be gathered from the master. When setting the *`file_client`* option to `local` the minion is configured to not gather this data from the master.

```
file_client: local
```

Now the salt-call command will not look for a master and will assume that the local system has all of the file and pillar resources.

Running States Masterless

The state system can be easily run without a Salt master, with all needed files local to the minion. To do this the minion configuration file needs to be set up to know how to return file_roots information like the master. The file_roots setting defaults to /srv/salt for the base environment just like on the master:

```
file_roots:
  base:
    - /srv/salt
```

Now set up the Salt State Tree, top file, and SLS modules in the same way that they would be set up on a master. Now, with the *`file_client`* option set to `local` and an available state tree then calls to functions in the state module will use the information in the file_roots on the minion instead of checking in with the master.

Remember that when creating a state tree on a minion there are no syntax or path changes needed, SLS modules written to be used from a master do not need to be modified in any way to work with a minion.

This makes it easy to ”script” deployments with Salt states without having to set up a master, and allows for these SLS modules to be easily moved into a Salt master as the deployment grows.

The declared state can now be executed with:

```
salt-call state.apply
```

Or the salt-call command can be executed with the `--local` flag, this makes it unnecessary to change the configuration file:

```
salt-call state.apply --local
```

External Pillars

External pillars are supported when running in masterless mode.

2.4.8 Salt Masterless Quickstart

Running a masterless salt-minion lets you use Salt's configuration management for a single machine without calling out to a Salt master on another machine.

Since the Salt minion contains such extensive functionality it can be useful to run it standalone. A standalone minion can be used to do a number of things:

- Stand up a master server via States (Salting a Salt Master)
- Use salt-call commands on a system without connectivity to a master
- Masterless States, run states entirely from files local to the minion

It is also useful for testing out state trees before deploying to a production setup.

Bootstrap Salt Minion

The salt-bootstrap script makes bootstrapping a server with Salt simple for any OS with a Bourne shell:

```
curl -L https://bootstrap.saltstack.com -o bootstrap_salt.sh
sudo sh bootstrap_salt.sh
```

See the salt-bootstrap documentation for other one liners. When using Vagrant to test out salt, the Vagrant salt provisioner will provision the VM for you.

Telling Salt to Run Masterless

To instruct the minion to not look for a master, the *`file_client`* configuration option needs to be set in the minion configuration file. By default the *`file_client`* is set to `remote` so that the minion gathers file server and pillar data from the salt master. When setting the *`file_client`* option to `local` the minion is configured to not gather this data from the master.

```
file_client: local
```

Now the salt minion will not look for a master and will assume that the local system has all of the file and pillar resources.

Configuration which resided in the *master configuration* (e.g. `/etc/salt/master`) should be moved to the *minion configuration* since the minion does not read the master configuration.

Note: When running Salt in masterless mode, do not run the salt-minion daemon. Otherwise, it will attempt to connect to a master and fail. The salt-call command stands on its own and does not need the salt-minion daemon.

Create State Tree

Following the successful installation of a salt-minion, the next step is to create a state tree, which is where the SLS files that comprise the possible states of the minion are stored.

The following example walks through the steps necessary to create a state tree that ensures that the server has the Apache webserver installed.

Note: For a complete explanation on Salt States, see the tutorial.

---

1. Create the `top.sls` file:

`/srv/salt/top.sls:`

```
base:
  '*':
    - webserver
```

2. Create the webserver state tree:

`/srv/salt/webserver.sls:`

```
apache:                 # ID declaration
  pkg:                  # state declaration
    - installed         # function declaration
```

Note: The apache package has different names on different platforms, for instance on Debian/Ubuntu it is apache2, on Fedora/RHEL it is httpd and on Arch it is apache

---

The only thing left is to provision our minion using `salt-call`.

Salt-call

The salt-call command is used to run remote execution functions locally on a minion instead of executing them from the master. Normally the salt-call command checks into the master to retrieve file server and pillar data, but when running standalone salt-call needs to be instructed to not check the master for this data:

```
salt-call --local state.apply
```

The `--local` flag tells the salt-minion to look for the state tree in the local file system and not to contact a Salt Master for instructions.

To provide verbose output, use `-l debug`:

```
salt-call --local state.apply -l debug
```

The minion first examines the `top.sls` file and determines that it is a part of the group matched by `*` glob and that the `webserver` SLS should be applied.

It then examines the `webserver.sls` file and finds the `apache` state, which installs the Apache package.

The minion should now have Apache installed, and the next step is to begin learning how to write *more complex states*.

2.5 Dependencies

Salt should run on any Unix-like platform so long as the dependencies are met.

- Python - Python2 >= 2.7, Python3 >= 3.4
- msgpack-python - High-performance message interchange format
- YAML - Python YAML bindings
- Jinja2 - parsing Salt States (configurable in the master settings)
- MarkupSafe - Implements a XML/HTML/XHTML Markup safe string for Python
- apache-libcloud - Python lib for interacting with many of the popular cloud service providers using a unified API
- Requests - HTTP library
- Tornado - Web framework and asynchronous networking library
- futures - Python2 only dependency. Backport of the concurrent.futures package from Python 3.2
- ZeroMQ:
  - ZeroMQ >= 3.2.0
  - pyzmq >= 2.2.0 - ZeroMQ Python bindings
  - PyCrypto - The Python cryptography toolkit

Warning: For historical reasons, Salt requires PyCrypto as a "lowest common denominator". However, PyCrypto is unmaintained and best practice is to manually upgrade to use a more maintained library such as PyCryptodome. See Issue #52674 and Issue #54115 for more info

Salt defaults to the ZeroMQ transport. The `--salt-transport` installation option is available, but currently only supports the `szeromq` option. This may be expanded in the future.

```
python setup.py --salt-transport=zeromq install
```

This way, only the required dependencies are pulled by the setup script if need be.

If installing using pip, the `--salt-transport` install option can be provided like:

```
pip install --install-option="--salt-transport=zeromq" salt
```

Note: Salt does not bundle dependencies that are typically distributed as part of the base OS. If you have unmet dependencies and are using a custom or minimal installation, you might need to install some additional packages from your OS vendor.

2.6 Optional Dependencies

- mako - an optional parser for Salt States (configurable in the master settings)
- gcc - dynamic Cython module compiling

2.7 Upgrading Salt

When upgrading Salt, the master(s) should always be upgraded first. Backward compatibility for minions running newer versions of salt than their masters is not guaranteed.

Whenever possible, backward compatibility between new masters and old minions will be preserved. Generally, the only exception to this policy is in case of a security vulnerability.

See also:

Installing Salt for development and contributing to the project.

2.8 Building Packages using Salt Pack

Salt-pack is an open-source package builder for most commonly used Linux platforms, for example: Redhat/CentOS and Debian/Ubuntu families, utilizing SaltStack states and execution modules to build Salt and a specified set of dependencies, from which a platform specific repository can be built.

https://github.com/saltstack/salt-pack

CHAPTER

THREE

CONFIGURING SALT

This section explains how to configure user access, view and store job results, secure and troubleshoot, and how to perform many other administrative tasks.

3.1 Configuring the Salt Master

The Salt system is amazingly simple and easy to configure, the two components of the Salt system each have a respective configuration file. The `salt-master` is configured via the master configuration file, and the `salt-minion` is configured via the minion configuration file.

See also:

Example master configuration file.

The configuration file for the salt-master is located at `/etc/salt/master` by default. Atomic included configuration files can be placed in `/etc/salt/master.d/*.conf`. Warning: files with other suffixes than .conf will not be included. A notable exception is FreeBSD, where the configuration file is located at `/usr/local/etc/salt`. The available options are as follows:

3.1.1 Primary Master Configuration

`interface`

Default: `0.0.0.0` (all interfaces)

The local interface to bind to, must be an IP address.

```
interface: 192.168.0.1
```

`ipv6`

Default: `False`

Whether the master should listen for IPv6 connections. If this is set to True, the interface option must be adjusted too (for example: `interface: '::'`)

```
ipv6: True
```

`publish_port`

Default: `4505`

The network port to set up the publication interface.

```
publish_port: 4505
```

`master_id`

Default: `None`

The id to be passed in the publish job to minions. This is used for MultiSyndics to return the job to the requesting master.

Note: This must be the same string as the syndic is configured with.

```
master_id: MasterOfMaster
```

`user`

Default: `root`

The user to run the Salt processes

```
user: root
```

`enable_ssh_minions`

Default: `False`

Tell the master to also use salt-ssh when running commands against minions.

```
enable_ssh_minions: True
```

Note: Cross-minion communication is still not possible. The Salt mine and publish.publish do not work between minion types.

`ret_port`

Default: `4506`

The port used by the return server, this is the server used by Salt to receive execution returns and command executions.

```
ret_port: 4506
```

`pidfile`

Default: `/var/run/salt-master.pid`

Specify the location of the master pidfile.

```
pidfile: /var/run/salt-master.pid
```

`root_dir`

Default: `/`

The system root directory to operate from, change this to make Salt run from an alternative root.

```
root_dir: /
```

Note: This directory is prepended to the following options: *`pki_dir`*, *`cachedir`*, *`sock_dir`*, *`log_file`*, *`autosign_file`*, *`autoreject_file`*, *`pidfile`*, *`autosign_grains_dir`*.

`conf_file`

Default: `/etc/salt/master`

The path to the master's configuration file.

```
conf_file: /etc/salt/master
```

`pki_dir`

Default: `/etc/salt/pki/master`

The directory to store the pki authentication keys.

```
pki_dir: /etc/salt/pki/master
```

`extension_modules`

Changed in version 2016.3.0: The default location for this directory has been moved. Prior to this version, the location was a directory named `extmods` in the Salt cachedir (on most platforms, `/var/cache/salt/extmods`). It has been moved into the master cachedir (on most platforms, `/var/cache/salt/master/extmods`).

Directory for custom modules. This directory can contain subdirectories for each of Salt's module types such as `runners`, `output`, `wheel`, `modules`, `states`, `returners`, `engines`, `utils`, etc. This path is appended to *`root_dir`*.

```
extension_modules: /root/salt_extmods
```

extmod_whitelist/extmod_blacklist

New in version 2017.7.0.

By using this dictionary, the modules that are synced to the master's extmod cache using *saltutil.sync_** can be limited. If nothing is set to a specific type, then all modules are accepted. To block all modules of a specific type, whitelist an empty list.

```
extmod_whitelist:
  modules:
    - custom_module
  engines:
    - custom_engine
  pillars: []

extmod_blacklist:
  modules:
    - specific_module
```

Valid options:

- modules
- states
- grains
- renderers
- returners
- output
- proxy
- runners
- wheel
- engines
- queues
- pillar
- utils
- sdb
- cache
- clouds
- tops
- roster
- tokens

module_dirs

Default: `[]`

Like `extension_modules`, but a list of extra directories to search for Salt modules.

```
module_dirs:
  - /var/cache/salt/minion/extmods
```

`cachedir`

Default: `/var/cache/salt/master`

The location used to store cache information, particularly the job information for executed salt commands.

This directory may contain sensitive data and should be protected accordingly.

```
cachedir: /var/cache/salt/master
```

`verify_env`

Default: `True`

Verify and set permissions on configuration directories at startup.

```
verify_env: True
```

`keep_jobs`

Default: `24`

Set the number of hours to keep old job information. Note that setting this option to `0` disables the cache cleaner.

```
keep_jobs: 24
```

`gather_job_timeout`

New in version 2014.7.0.

Default: `10`

The number of seconds to wait when the client is requesting information about running jobs.

```
gather_job_timeout: 10
```

`timeout`

Default: `5`

Set the default timeout for the salt command and api.

`loop_interval`

Default: `60`

The loop_interval option controls the seconds for the master's maintenance process check cycle. This process updates file server backends, cleans the job cache and executes the scheduler.

`output`

Default: `nested`

Set the default outputter used by the salt command.

`outputter_dirs`

Default: `[]`

A list of additional directories to search for salt outputters in.

```
outputter_dirs: []
```

`output_file`

Default: None

Set the default output file used by the salt command. Default is to output to the CLI and not to a file. Functions the same way as the "–out-file" CLI option, only sets this to a single file for all salt commands.

```
output_file: /path/output/file
```

`show_timeout`

Default: `True`

Tell the client to show minions that have timed out.

```
show_timeout: True
```

`show_jid`

Default: `False`

Tell the client to display the jid when a job is published.

```
show_jid: False
```

`color`

Default: `True`

By default output is colored, to disable colored output set the color value to False.

```
color: False
```

`color_theme`

Default: `""`

Specifies a path to the color theme to use for colored command line output.

```
color_theme: /etc/salt/color_theme
```

`cli_summary`

Default: `False`

When set to `True`, displays a summary of the number of minions targeted, the number of minions returned, and the number of minions that did not return.

```
cli_summary: False
```

`sock_dir`

Default: `/var/run/salt/master`

Set the location to use for creating Unix sockets for master process communication.

```
sock_dir: /var/run/salt/master
```

`enable_gpu_grains`

Default: `False`

Enable GPU hardware data for your master. Be aware that the master can take a while to start up when lspci and/or dmidecode is used to populate the grains for the master.

```
enable_gpu_grains: True
```

`job_cache`

Default: `True`

The master maintains a temporary job cache. While this is a great addition, it can be a burden on the master for larger deployments (over 5000 minions). Disabling the job cache will make previously executed jobs unavailable to the jobs system and is not generally recommended. Normally it is wise to make sure the master has access to a faster IO system or a tmpfs is mounted to the jobs dir.

```
job_cache: True
```

Note: Setting the `job_cache` to `False` will not cache minion returns, but the JID directory for each job is still created. The creation of the JID directories is necessary because Salt uses those directories to check for JID collisions. By setting this option to `False`, the job cache directory, which is `/var/cache/salt/master/jobs/` by default, will be smaller, but the JID directories will still be present.

Note that the *`keep_jobs`* option can be set to a lower value, such as `1`, to limit the number of hours jobs are stored in the job cache. (The default is 24 hours.)

Please see the *Managing the Job Cache* documentation for more information.

`minion_data_cache`

Default: `True`

The minion data cache is a cache of information about the minions stored on the master, this information is primarily the pillar, grains and mine data. The data is cached via the cache subsystem in the Master cachedir under the name of the minion or in a supported database. The data is used to predetermine what minions are expected to reply from executions.

```
minion_data_cache: True
```

`cache`

Default: `localfs`

Cache subsystem module to use for minion data cache.

```
cache: consul
```

`memcache_expire_seconds`

Default: `0`

Memcache is an additional cache layer that keeps a limited amount of data fetched from the minion data cache for a limited period of time in memory that makes cache operations faster. It doesn't make much sense for the `localfs` cache driver but helps for more complex drivers like `consul`.

This option sets the memcache items expiration time. By default is set to `0` that disables the memcache.

```
memcache_expire_seconds: 30
```

`memcache_max_items`

Default: `1024`

Set memcache limit in items that are bank-key pairs. I.e the list of minion_0/data, minion_0/mine, minion_1/data contains 3 items. This value depends on the count of minions usually targeted in your environment. The best one could be found by analyzing the cache log with `memcache_debug` enabled.

```
memcache_max_items: 1024
```

`memcache_full_cleanup`

Default: `False`

If cache storage got full, i.e. the items count exceeds the `memcache_max_items` value, memcache cleans up it's storage. If this option set to `False` memcache removes the only one oldest value from it's storage. If this set set to `True` memcache removes all the expired items and also removes the oldest one if there are no expired items.

```
memcache_full_cleanup: True
```

memcache_debug

Default: `False`

Enable collecting the memcache stats and log it on *debug* log level. If enabled memcache collect information about how many `fetch` calls has been done and how many of them has been hit by memcache. Also it outputs the rate value that is the result of division of the first two values. This should help to choose right values for the expiration time and the cache size.

```
memcache_debug: True
```

ext_job_cache

Default: `''`

Used to specify a default returner for all minions. When this option is set, the specified returner needs to be properly configured and the minions will always default to sending returns to this returner. This will also disable the local job cache on the master.

```
ext_job_cache: redis
```

event_return

New in version 2015.5.0.

Default: `''`

Specify the returner(s) to use to log events. Each returner may have installation and configuration requirements. Read the returner's documentation.

Note: Not all returners support event returns. Verify that a returner has an `event_return()` function before configuring this option with a returner.

```
event_return:
  - syslog
  - splunk
```

event_return_queue

New in version 2015.5.0.

Default: `0`

On busy systems, enabling event_returns can cause a considerable load on the storage system for returners. Events can be queued on the master and stored in a batched fashion using a single transaction for multiple events. By default, events are not queued.

```
event_return_queue: 0
```

event_return_whitelist

New in version 2015.5.0.

Default: `[]`

Only return events matching tags in a whitelist.

Changed in version 2016.11.0: Supports glob matching patterns.

```
event_return_whitelist:
  - salt/master/a_tag
  - salt/run/*/ret
```

event_return_blacklist

New in version 2015.5.0.

Default: `[]`

Store all event returns _except_ the tags in a blacklist.

Changed in version 2016.11.0: Supports glob matching patterns.

```
event_return_blacklist:
  - salt/master/not_this_tag
  - salt/wheel/*/ret
```

max_event_size

New in version 2014.7.0.

Default: `1048576`

Passing very large events can cause the minion to consume large amounts of memory. This value tunes the maximum size of a message allowed onto the master event bus. The value is expressed in bytes.

```
max_event_size: 1048576
```

master_job_cache

New in version 2014.7.0.

Default: `local_cache`

Specify the returner to use for the job cache. The job cache will only be interacted with from the salt master and therefore does not need to be accessible from the minions.

```
master_job_cache: redis
```

job_cache_store_endtime

New in version 2015.8.0.

Default: `False`

Specify whether the Salt Master should store end times for jobs as returns come in.

```
job_cache_store_endtime: False
```

`enforce_mine_cache`

Default: False

By-default when disabling the minion_data_cache mine will stop working since it is based on cached data, by enabling this option we explicitly enabling only the cache for the mine system.

```
enforce_mine_cache: False
```

`max_minions`

Default: 0

The maximum number of minion connections allowed by the master. Use this to accommodate the number of minions per master if you have different types of hardware serving your minions. The default of `0` means unlimited connections. Please note that this can slow down the authentication process a bit in large setups.

```
max_minions: 100
```

`con_cache`

Default: False

If max_minions is used in large installations, the master might experience high-load situations because of having to check the number of connected minions for every authentication. This cache provides the minion-ids of all connected minions to all MWorker-processes and greatly improves the performance of max_minions.

```
con_cache: True
```

`presence_events`

Default: False

Causes the master to periodically look for actively connected minions. *Presence events* are fired on the event bus on a regular interval with a list of connected minions, as well as events with lists of newly connected or disconnected minions. This is a master-only operation that does not send executions to minions.

```
presence_events: False
```

`ping_on_rotate`

New in version 2014.7.0.

Default: `False`

By default, the master AES key rotates every 24 hours. The next command following a key rotation will trigger a key refresh from the minion which may result in minions which do not respond to the first command after a key refresh.

To tell the master to ping all minions immediately after an AES key refresh, set `ping_on_rotate` to `True`. This should mitigate the issue where a minion does not appear to initially respond after a key is rotated.

Note that enabling this may cause high load on the master immediately after the key rotation event as minions reconnect. Consider this carefully if this salt master is managing a large number of minions.

If disabled, it is recommended to handle this event by listening for the `aes_key_rotate` event with the `key` tag and acting appropriately.

```
ping_on_rotate: False
```

transport

Default: `zeromq`

Changes the underlying transport layer. ZeroMQ is the recommended transport while additional transport layers are under development. Supported values are `zeromq` and `tcp` (experimental). This setting has a significant impact on performance and should not be changed unless you know what you are doing!

```
transport: zeromq
```

transport_opts

Default: `{}`

(experimental) Starts multiple transports and overrides options for each transport with the provided dictionary This setting has a significant impact on performance and should not be changed unless you know what you are doing! The following example shows how to start a TCP transport alongside a ZMQ transport.

```
transport_opts:
  tcp:
    publish_port: 4605
    ret_port: 4606
  zeromq: []
```

master_stats

Default: False

Turning on the master stats enables runtime throughput and statistics events to be fired from the master event bus. These events will report on what functions have been run on the master and how long these runs have, on average, taken over a given period of time.

master_stats_event_iter

Default: 60

The time in seconds to fire master_stats events. This will only fire in conjunction with receiving a request to the master, idle masters will not fire these events.

`sock_pool_size`

Default: 1

To avoid blocking waiting while writing a data to a socket, we support socket pool for Salt applications. For example, a job with a large number of target host list can cause long period blocking waiting. The option is used by ZMQ and TCP transports, and the other transport methods don't need the socket pool by definition. Most of Salt tools, including CLI, are enough to use a single bucket of socket pool. On the other hands, it is highly recommended to set the size of socket pool larger than 1 for other Salt applications, especially Salt API, which must write data to socket concurrently.

```
sock_pool_size: 15
```

`ipc_mode`

Default: `ipc`

The ipc strategy. (i.e., sockets versus tcp, etc.) Windows platforms lack POSIX IPC and must rely on TCP based inter-process communications. `ipc_mode` is set to `tcp` by default on Windows.

```
ipc_mode: ipc
```

`tcp_master_pub_port`

Default: `4512`

The TCP port on which events for the master should be published if `ipc_mode` is TCP.

```
tcp_master_pub_port: 4512
```

`tcp_master_pull_port`

Default: `4513`

The TCP port on which events for the master should be pulled if `ipc_mode` is TCP.

```
tcp_master_pull_port: 4513
```

`tcp_master_publish_pull`

Default: `4514`

The TCP port on which events for the master should be pulled fom and then republished onto the event bus on the master.

```
tcp_master_publish_pull: 4514
```

`tcp_master_workers`

Default: `4515`

The TCP port for `mworkers` to connect to on the master.

```
tcp_master_workers: 4515
```

`auth_events`

New in version 2017.7.3.

Default: `True`

Determines whether the master will fire authentication events. *Authentication events* are fired when a minion performs an authentication check with the master.

```
auth_events: True
```

`minion_data_cache_events`

New in version 2017.7.3.

Default: `True`

Determines whether the master will fire minion data cache events. Minion data cache events are fired when a minion requests a minion data cache refresh.

```
minion_data_cache_events: True
```

`http_connect_timeout`

New in version 2019.2.0.

Default: `20`

HTTP connection timeout in seconds. Applied when fetching files using tornado back-end. Should be greater than overall download time.

```
http_connect_timeout: 20
```

`http_request_timeout`

New in version 2015.8.0.

Default: `3600`

HTTP request timeout in seconds. Applied when fetching files using tornado back-end. Should be greater than overall download time.

```
http_request_timeout: 3600
```

`use_yamlloader_old`

New in version 2019.2.1.

Default: `False`

Use the pre-2019.2 YAML renderer. Uses legacy YAML rendering to support some legacy inline data structures. See the *2019.2.1 release notes* for more details.

```
use_yamlloader_old: False
```

3.1.2 Salt-SSH Configuration

`roster`

Default: `flat`

Define the default salt-ssh roster module to use

```
roster: cache
```

`roster_defaults`

New in version 2017.7.0.

Default settings which will be inherited by all rosters.

```
roster_defaults:
  user: daniel
  sudo: True
  priv: /root/.ssh/id_rsa
  tty: True
```

`roster_file`

Default: `/etc/salt/roster`

Pass in an alternative location for the salt-ssh *`flat`* roster file.

```
roster_file: /root/roster
```

`rosters`

Default: `None`

Define locations for *`flat`* roster files so they can be chosen when using Salt API. An administrator can place roster files into these locations. Then, when calling Salt API, the *`roster_file`* parameter should contain a relative path to these locations. That is, `roster_file=/foo/roster` will be resolved as `/etc/salt/roster.d/foo/roster` etc. This feature prevents passing insecure custom rosters through the Salt API.

```
rosters:
 - /etc/salt/roster.d
 - /opt/salt/some/more/rosters
```

`ssh_passwd`

Default: `''`

The ssh password to log in with.

```
ssh_passwd: ''
```

`ssh_priv_passwd`

Default: `''`

Passphrase for ssh private key file.

```
ssh_priv_passwd: ''
```

`ssh_port`

Default: `22`

The target system's ssh port number.

```
ssh_port: 22
```

`ssh_scan_ports`

Default: `22`

Comma-separated list of ports to scan.

```
ssh_scan_ports: 22
```

`ssh_scan_timeout`

Default: `0.01`

Scanning socket timeout for salt-ssh.

```
ssh_scan_timeout: 0.01
```

`ssh_sudo`

Default: `False`

Boolean to run command via sudo.

```
ssh_sudo: False
```

`ssh_timeout`

Default: `60`

Number of seconds to wait for a response when establishing an SSH connection.

```
ssh_timeout: 60
```

`ssh_user`

Default: `root`

The user to log in as.

```
ssh_user: root
```

`ssh_log_file`

New in version 2016.3.5.

Default: `/var/log/salt/ssh`

Specify the log file of the `salt-ssh` command.

```
ssh_log_file: /var/log/salt/ssh
```

`ssh_minion_opts`

Default: None

Pass in minion option overrides that will be inserted into the SHIM for salt-ssh calls. The local minion config is not used for salt-ssh. Can be overridden on a per-minion basis in the roster (`minion_opts`)

```
ssh_minion_opts:
  gpg_keydir: /root/gpg
```

`ssh_use_home_key`

Default: False

Set this to True to default to using `~/.ssh/id_rsa` for salt-ssh authentication with minions

```
ssh_use_home_key: False
```

`ssh_identities_only`

Default: `False`

Set this to `True` to default salt-ssh to run with `-o IdentitiesOnly=yes`. This option is intended for situations where the ssh-agent offers many different identities and allows ssh to ignore those identities and use the only one specified in options.

```
ssh_identities_only: False
```

`ssh_list_nodegroups`

Default: `{}`

List-only nodegroups for salt-ssh. Each group must be formed as either a comma-separated list, or a YAML list. This option is useful to group minions into easy-to-target groups when using salt-ssh. These groups can then be targeted with the normal -N argument to salt-ssh.

```
ssh_list_nodegroups:
  groupA: minion1,minion2
  groupB: minion1,minion3
```

`thin_extra_mods`

Default: None

List of additional modules, needed to be included into the Salt Thin. Pass a list of importable Python modules that are typically located in the *site-packages* Python directory so they will be also always included into the Salt Thin, once generated.

`min_extra_mods`

Default: None

Identical as *thin_extra_mods*, only applied to the Salt Minimal.

3.1.3 Master Security Settings

`open_mode`

Default: `False`

Open mode is a dangerous security feature. One problem encountered with pki authentication systems is that keys can become ”mixed up” and authentication begins to fail. Open mode turns off authentication and tells the master to accept all authentication. This will clean up the pki keys received from the minions. Open mode should not be turned on for general use. Open mode should only be used for a short period of time to clean up pki keys. To turn on open mode set this value to `True`.

```
open_mode: False
```

`auto_accept`

Default: `False`

Enable auto_accept. This setting will automatically accept all incoming public keys from minions.

```
auto_accept: False
```

`keysize`

Default: `2048`

The size of key that should be generated when creating new keys.

```
keysize: 2048
```

`autosign_timeout`

New in version 2014.7.0.

Default: `120`

Time in minutes that a incoming public key with a matching name found in pki_dir/minion_autosign/keyid is automatically accepted. Expired autosign keys are removed when the master checks the minion_autosign directory. This method to auto accept minions can be safer than an autosign_file because the keyid record can expire and is limited to being an exact name match. This should still be considered a less than secure option, due to the fact that trust is based on just the requesting minion id.

`autosign_file`

Default: `not defined`

If the `autosign_file` is specified incoming keys specified in the autosign_file will be automatically accepted. Matches will be searched for first by string comparison, then by globbing, then by full-string regex matching. This should still be considered a less than secure option, due to the fact that trust is based on just the requesting minion id.

Changed in version 2018.3.0: For security reasons the file must be readonly except for it's owner. If *`permissive_pki_access`* is `True` the owning group can also have write access, but if Salt is running as `root` it must be a member of that group. A less strict requirement also existed in previous version.

`autoreject_file`

New in version 2014.1.0.

Default: `not defined`

Works like *`autosign_file`*, but instead allows you to specify minion IDs for which keys will automatically be rejected. Will override both membership in the *`autosign_file`* and the *`auto_accept`* setting.

`autosign_grains_dir`

New in version 2018.3.0.

Default: `not defined`

If the `autosign_grains_dir` is specified, incoming keys from minions with grain values that match those defined in files in the autosign_grains_dir will be accepted automatically. Grain values that should be accepted automatically can be defined by creating a file named like the corresponding grain in the autosign_grains_dir and writing the values into that file, one value per line. Lines starting with a # will be ignored. Minion must be configured to send the corresponding grains on authentication. This should still be considered a less than secure option, due to the fact that trust is based on just the requesting minion.

Please see the *Autoaccept Minions from Grains* documentation for more information.

```
autosign_grains_dir: /etc/salt/autosign_grains
```

`permissive_pki_access`

Default: `False`

Enable permissive access to the salt keys. This allows you to run the master or minion as root, but have a non-root group be given access to your pki_dir. To make the access explicit, root must belong to the group you've given access to. This is potentially quite insecure. If an autosign_file is specified, enabling permissive_pki_access will allow group access to that specific file.

```
permissive_pki_access: False
```

publisher_acl

Default: `{}`

Enable user accounts on the master to execute specific modules. These modules can be expressed as regular expressions.

```
publisher_acl:
  fred:
    - test.ping
    - pkg.*
```

publisher_acl_blacklist

Default: `{}`

Blacklist users or modules

This example would blacklist all non sudo users, including root from running any commands. It would also blacklist any use of the "cmd" module.

This is completely disabled by default.

```
publisher_acl_blacklist:
  users:
    - root
    - '^(?!sudo_).*$'   #  all non sudo users
  modules:
    - cmd.*
    - test.echo
```

sudo_acl

Default: `False`

Enforce `publisher_acl` and `publisher_acl_blacklist` when users have sudo access to the salt command.

```
sudo_acl: False
```

external_auth

Default: `{}`

The external auth system uses the Salt auth modules to authenticate and validate users to access areas of the Salt system.

```
external_auth:
  pam:
    fred:
      - test.*
```

`token_expire`

Default: `43200`

Time (in seconds) for a newly generated token to live.

Default: 12 hours

```
token_expire: 43200
```

`token_expire_user_override`

Default: `False`

Allow eauth users to specify the expiry time of the tokens they generate.

A boolean applies to all users or a dictionary of whitelisted eauth backends and usernames may be given:

```
token_expire_user_override:
  pam:
    - fred
    - tom
  ldap:
    - gary
```

`keep_acl_in_token`

Default: `False`

Set to True to enable keeping the calculated user's auth list in the token file. This is disabled by default and the auth list is calculated or requested from the eauth driver each time.

```
keep_acl_in_token: False
```

`eauth_acl_module`

Default: `''`

Auth subsystem module to use to get authorized access list for a user. By default it's the same module used for external authentication.

```
eauth_acl_module: django
```

`file_recv`

Default: `False`

Allow minions to push files to the master. This is disabled by default, for security purposes.

```
file_recv: False
```

file_recv_max_size

New in version 2014.7.0.

Default: `100`

Set a hard-limit on the size of the files that can be pushed to the master. It will be interpreted as megabytes.

```
file_recv_max_size: 100
```

master_sign_pubkey

Default: `False`

Sign the master auth-replies with a cryptographic signature of the master's public key. Please see the tutorial how to use these settings in the Multimaster-PKI with Failover Tutorial

```
master_sign_pubkey: True
```

master_sign_key_name

Default: `master_sign`

The customizable name of the signing-key-pair without suffix.

```
master_sign_key_name: <filename_without_suffix>
```

master_pubkey_signature

Default: `master_pubkey_signature`

The name of the file in the master's pki-directory that holds the pre-calculated signature of the master's public-key.

```
master_pubkey_signature: <filename>
```

master_use_pubkey_signature

Default: `False`

Instead of computing the signature for each auth-reply, use a pre-calculated signature. The *`master_pubkey_signature`* must also be set for this.

```
master_use_pubkey_signature: True
```

rotate_aes_key

Default: `True`

Rotate the salt-masters AES-key when a minion-public is deleted with salt-key. This is a very important security-setting. Disabling it will enable deleted minions to still listen in on the messages published by the salt-master. Do not disable this unless it is absolutely clear what this does.

```
rotate_aes_key: True
```

publish_session

Default: `86400`

The number of seconds between AES key rotations on the master.

```
publish_session: Default: 86400
```

ssl

New in version 2016.11.0.

Default: `None`

TLS/SSL connection options. This could be set to a dictionary containing arguments corresponding to python `ssl.wrap_socket` method. For details see Tornado and Python documentation.

Note: to set enum arguments values like `cert_reqs` and `ssl_version` use constant names without ssl module prefix: `CERT_REQUIRED` or `PROTOCOL_SSLv23`.

```
ssl:
    keyfile: <path_to_keyfile>
    certfile: <path_to_certfile>
    ssl_version: PROTOCOL_TLSv1_2
```

preserve_minion_cache

Default: `False`

By default, the master deletes its cache of minion data when the key for that minion is removed. To preserve the cache after key deletion, set `preserve_minion_cache` to True.

WARNING: This may have security implications if compromised minions auth with a previous deleted minion ID.

```
preserve_minion_cache: False
```

allow_minion_key_revoke

Default: `True`

Controls whether a minion can request its own key revocation. When True the master will honor the minion's request and revoke its key. When False, the master will drop the request and the minion's key will remain accepted.

```
allow_minion_key_revoke: False
```

`optimization_order`

Default: `[0, 1, 2]`

In cases where Salt is distributed without .py files, this option determines the priority of optimization level(s) Salt's module loader should prefer.

Note: This option is only supported on Python 3.5+.

```
optimization_order:
  - 2
  - 0
  - 1
```

3.1.4 Master Large Scale Tuning Settings

`max_open_files`

Default: `100000`

Each minion connecting to the master uses AT LEAST one file descriptor, the master subscription connection. If enough minions connect you might start seeing on the console(and then salt-master crashes):

```
Too many open files (tcp_listener.cpp:335)
Aborted (core dumped)
```

```
max_open_files: 100000
```

By default this value will be the one of *ulimit -Hn*, i.e., the hard limit for max open files.

To set a different value than the default one, uncomment, and configure this setting. Remember that this value CANNOT be higher than the hard limit. Raising the hard limit depends on the OS and/or distribution, a good way to find the limit is to search the internet for something like this:

```
raise max open files hard limit debian
```

`worker_threads`

Default: `5`

The number of threads to start for receiving commands and replies from minions. If minions are stalling on replies because you have many minions, raise the worker_threads value.

Worker threads should not be put below 3 when using the peer system, but can drop down to 1 worker otherwise.

Note: When the master daemon starts, it is expected behaviour to see multiple salt-master processes, even if 'worker_threads' is set to '1'. At a minimum, a controlling process will start along with a Publisher, an EventPublisher, and a number of MWorker processes will be started. The number of MWorker processes is tuneable by the 'worker_threads' configuration value while the others are not.

```
worker_threads: 5
```

`pub_hwm`

Default: `1000`

The zeromq high water mark on the publisher interface.

```
pub_hwm: 1000
```

`zmq_backlog`

Default: `1000`

The listen queue size of the ZeroMQ backlog.

```
zmq_backlog: 1000
```

3.1.5 Master Module Management

`runner_dirs`

Default: `[]`

Set additional directories to search for runner modules.

```
runner_dirs:
  - /var/lib/salt/runners
```

`utils_dirs`

New in version 2018.3.0.

Default: `[]`

Set additional directories to search for util modules.

```
utils_dirs:
  - /var/lib/salt/utils
```

`cython_enable`

Default: `False`

Set to true to enable Cython modules (.pyx files) to be compiled on the fly on the Salt master.

```
cython_enable: False
```

3.1.6 Master State System Settings

`state_top`

Default: `top.sls`

The state system uses a "top" file to tell the minions what environment to use and what modules to use. The state_top file is defined relative to the root of the base environment. The value of "state_top" is also used for the pillar top file

```
state_top: top.sls
```

state_top_saltenv

This option has no default value. Set it to an environment name to ensure that *only* the top file from that environment is considered during a *highstate*.

Note: Using this value does not change the merging strategy. For instance, if *`top_file_merging_strategy`* is set to `merge`, and *`state_top_saltenv`* is set to `foo`, then any sections for environments other than `foo` in the top file for the `foo` environment will be ignored. With *`state_top_saltenv`* set to `base`, all states from all environments in the `base` top file will be applied, while all other top files are ignored. The only way to set *`state_top_saltenv`* to something other than `base` and not have the other environments in the targeted top file ignored, would be to set *`top_file_merging_strategy`* to `merge_all`.

```
state_top_saltenv: dev
```

top_file_merging_strategy

Changed in version 2016.11.0: A `merge_all` strategy has been added.

Default: `merge`

When no specific fileserver environment (a.k.a. `saltenv`) has been specified for a *highstate*, all environments' top files are inspected. This config option determines how the SLS targets in those top files are handled.

When set to `merge`, the `base` environment's top file is evaluated first, followed by the other environments' top files. The first target expression (e.g. `'*'`) for a given environment is kept, and when the same target expression is used in a different top file evaluated later, it is ignored. Because `base` is evaluated first, it is authoritative. For example, if there is a target for `'*'` for the `foo` environment in both the `base` and `foo` environment's top files, the one in the `foo` environment would be ignored. The environments will be evaluated in no specific order (aside from `base` coming first). For greater control over the order in which the environments are evaluated, use *`env_order`*. Note that, aside from the `base` environment's top file, any sections in top files that do not match that top file's environment will be ignored. So, for example, a section for the `qa` environment would be ignored if it appears in the `dev` environment's top file. To keep use cases like this from being ignored, use the `merge_all` strategy.

When set to `same`, then for each environment, only that environment's top file is processed, with the others being ignored. For example, only the `dev` environment's top file will be processed for the `dev` environment, and any SLS targets defined for `dev` in the `base` environment's (or any other environment's) top file will be ignored. If an environment does not have a top file, then the top file from the `default_top` config parameter will be used as a fallback.

When set to `merge_all`, then all states in all environments in all top files will be applied. The order in which individual SLS files will be executed will depend on the order in which the top files were evaluated, and the environments will be evaluated in no specific order. For greater control over the order in which the environments are evaluated, use *`env_order`*.

```
top_file_merging_strategy: same
```

`env_order`

Default: `[]`

When *`top_file_merging_strategy`* is set to `merge`, and no environment is specified for a *highstate*, this config option allows for the order in which top files are evaluated to be explicitly defined.

```
env_order:
  - base
  - dev
  - qa
```

`master_tops`

Default: `{}`

The master_tops option replaces the external_nodes option by creating a pluggable system for the generation of external top data. The external_nodes option is deprecated by the master_tops option. To gain the capabilities of the classic external_nodes system, use the following configuration:

```
master_tops:
  ext_nodes: <Shell command which returns yaml>
```

`renderer`

Default: `jinja|yaml`

The renderer to use on the minions to render the state data.

```
renderer: jinja|json
```

`userdata_template`

New in version 2016.11.4.

Default: `None`

The renderer to use for templating userdata files in salt-cloud, if the `userdata_template` is not set in the cloud profile. If no value is set in the cloud profile or master config file, no templating will be performed.

```
userdata_template: jinja
```

`jinja_env`

New in version 2018.3.0.

Default: `{}`

jinja_env overrides the default Jinja environment options for **all templates except sls templates**. To set the options for sls templates use *`jinja_sls_env`*.

Note: The Jinja2 Environment documentation is the official source for the default values. Not all the options listed in the jinja documentation can be overridden using *`jinja_env`* or *`jinja_sls_env`*.

The default options are:

```
jinja_env:
  block_start_string: '{%'
  block_end_string: '%}'
  variable_start_string: '{{'
  variable_end_string: '}}'
  comment_start_string: '{#'
  comment_end_string: '#}'
  line_statement_prefix:
  line_comment_prefix:
  trim_blocks: False
  lstrip_blocks: False
  newline_sequence: '\n'
  keep_trailing_newline: False
```

`jinja_sls_env`

New in version 2018.3.0.

Default: `{}`

jinja_sls_env sets the Jinja environment options for **sls templates**. The defaults and accepted options are exactly the same as they are for *`jinja_env`*.

The default options are:

```
jinja_sls_env:
  block_start_string: '{%'
  block_end_string: '%}'
  variable_start_string: '{{'
  variable_end_string: '}}'
  comment_start_string: '{#'
  comment_end_string: '#}'
  line_statement_prefix:
  line_comment_prefix:
  trim_blocks: False
  lstrip_blocks: False
  newline_sequence: '\n'
  keep_trailing_newline: False
```

Example using line statements and line comments to increase ease of use:

If your configuration options are

```
jinja_sls_env:
  line_statement_prefix: '%'
  line_comment_prefix: '##'
```

With these options jinja will interpret anything after a `%` at the start of a line (ignoreing whitespace) as a jinja statement and will interpret anything after a `##` as a comment.

This allows the following more convenient syntax to be used:

```
## (this comment will not stay once rendered)
# (this comment remains in the rendered template)
## ensure all the formula services are running
% for service in formula_services:
```

(continues on next page)

(continued from previous page)

```
enable_service_{{ service }}:
  service.running:
    name: {{ service }}
% endfor
```

The following less convenient but equivalent syntax would have to be used if you had not set the line_statement and line_comment options:

```
{# (this comment will not stay once rendered) #}
# (this comment remains in the rendered template)
{# ensure all the formula services are running #}
{% for service in formula_services %}
enable_service_{{ service }}:
  service.running:
    name: {{ service }}
{% endfor %}
```

`jinja_trim_blocks`

Deprecated since version 2018.3.0: Replaced by *`jinja_env`* and *`jinja_sls_env`*

New in version 2014.1.0.

Default: `False`

If this is set to `True`, the first newline after a Jinja block is removed (block, not variable tag!). Defaults to `False` and corresponds to the Jinja environment init variable `trim_blocks`.

```
jinja_trim_blocks: False
```

`jinja_lstrip_blocks`

Deprecated since version 2018.3.0: Replaced by *`jinja_env`* and *`jinja_sls_env`*

New in version 2014.1.0.

Default: `False`

If this is set to `True`, leading spaces and tabs are stripped from the start of a line to a block. Defaults to `False` and corresponds to the Jinja environment init variable `lstrip_blocks`.

```
jinja_lstrip_blocks: False
```

`failhard`

Default: `False`

Set the global failhard flag. This informs all states to stop running states at the moment a single state fails.

```
failhard: False
```

state_verbose

Default: `True`

Controls the verbosity of state runs. By default, the results of all states are returned, but setting this value to `False` will cause salt to only display output for states that failed or states that have changes.

```
state_verbose: False
```

state_output

Default: `full`

The state_output setting controls which results will be output full multi line:

- `full`, `terse` - each state will be full/terse
- `mixed` - only states with errors will be full
- `changes` - states with changes and errors will be full

`full_id`, `mixed_id`, `changes_id` and `terse_id` are also allowed; when set, the state ID will be used as name in the output.

```
state_output: full
```

state_output_diff

Default: `False`

The state_output_diff setting changes whether or not the output from successful states is returned. Useful when even the terse output of these states is cluttering the logs. Set it to True to ignore them.

```
state_output_diff: False
```

state_aggregate

Default: `False`

Automatically aggregate all states that have support for mod_aggregate by setting to `True`. Or pass a list of state module names to automatically aggregate just those types.

```
state_aggregate:
  - pkg
```

```
state_aggregate: True
```

state_events

Default: `False`

Send progress events as each function in a state run completes execution by setting to `True`. Progress events are in the format `salt/job/<JID>/prog/<MID>/<RUN NUM>`.

```
state_events: True
```

`yaml_utf8`

Default: `False`

Enable extra routines for YAML renderer used states containing UTF characters.

```
yaml_utf8: False
```

`runner_returns`

Default: `False`

If set to `True`, runner jobs will be saved to job cache (defined by *`master_job_cache`*).

```
runner_returns: True
```

3.1.7 Master File Server Settings

`fileserver_backend`

Default: `['roots']`

Salt supports a modular fileserver backend system, this system allows the salt master to link directly to third party systems to gather and manage the files available to minions. Multiple backends can be configured and will be searched for the requested file in the order in which they are defined here. The default setting only enables the standard backend `roots`, which is configured using the *`file_roots`* option.

Example:

```
fileserver_backend:
  - roots
  - gitfs
```

Note: For masterless Salt, this parameter must be specified in the minion config file.

`fileserver_followsymlinks`

New in version 2014.1.0.

Default: `True`

By default, the file_server follows symlinks when walking the filesystem tree. Currently this only applies to the default roots fileserver_backend.

```
fileserver_followsymlinks: True
```

`fileserver_ignoresymlinks`

New in version 2014.1.0.

Default: `False`

If you do not want symlinks to be treated as the files they are pointing to, set `fileserver_ignoresymlinks` to `True`. By default this is set to False. When set to `True`, any detected symlink while listing files on the Master will not be returned to the Minion.

```
fileserver_ignoresymlinks: False
```

`fileserver_limit_traversal`

New in version 2014.1.0.

Deprecated since version 2018.3.4: This option is now ignored. Firstly, it only traversed *`file_roots`*, which means it did not work for the other fileserver backends. Secondly, since this option was added we have added caching to the code that traverses the file_roots (and gitfs, etc.), which greatly reduces the amount of traversal that is done.

Default: `False`

By default, the Salt fileserver recurses fully into all defined environments to attempt to find files. To limit this behavior so that the fileserver only traverses directories with SLS files and special Salt directories like _modules, set `fileserver_limit_traversal` to `True`. This might be useful for installations where a file root has a very large number of files and performance is impacted.

```
fileserver_limit_traversal: False
```

`fileserver_list_cache_time`

New in version 2014.1.0.

Changed in version 2016.11.0: The default was changed from `30` seconds to `20`.

Default: `20`

Salt caches the list of files/symlinks/directories for each fileserver backend and environment as they are requested, to guard against a performance bottleneck at scale when many minions all ask the fileserver which files are available simultaneously. This configuration parameter allows for the max age of that cache to be altered.

Set this value to `0` to disable use of this cache altogether, but keep in mind that this may increase the CPU load on the master when running a highstate on a large number of minions.

Note: Rather than altering this configuration parameter, it may be advisable to use the *`fileserver.clear_file_list_cache`* runner to clear these caches.

```
fileserver_list_cache_time: 5
```

`fileserver_verify_config`

New in version 2017.7.0.

Default: `True`

By default, as the master starts it performs some sanity checks on the configured fileserver backends. If any of these sanity checks fail (such as when an invalid configuration is used), the master daemon will abort.

To skip these sanity checks, set this option to `False`.

```
fileserver_verify_config: False
```

hash_type

Default: `sha256`

The hash_type is the hash to use when discovering the hash of a file on the master server. The default is sha256, but md5, sha1, sha224, sha384, and sha512 are also supported.

```
hash_type: sha256
```

file_buffer_size

Default: `1048576`

The buffer size in the file server in bytes.

```
file_buffer_size: 1048576
```

file_ignore_regex

Default: `''`

A regular expression (or a list of expressions) that will be matched against the file path before syncing the modules and states to the minions. This includes files affected by the file.recurse state. For example, if you manage your custom modules and states in subversion and don't want all the '.svn' folders and content synced to your minions, you could set this to '/.svn($|/)'. By default nothing is ignored.

```
file_ignore_regex:
  - '/\.svn($|/)'
  - '/\.git($|/)'
```

file_ignore_glob

Default `''`

A file glob (or list of file globs) that will be matched against the file path before syncing the modules and states to the minions. This is similar to file_ignore_regex above, but works on globs instead of regex. By default nothing is ignored.

```
file_ignore_glob:
  - '\*.pyc'
  - '\*/somefolder/\*.bak'
  - '\*.swp'
```

Note: Vim's .swp files are a common cause of Unicode errors in *`file.recurse`* states which use templating. Unless there is a good reason to distribute them via the fileserver, it is good practice to include `'\*.swp'` in the *`file_ignore_glob`*.

`master_roots`

Default: `/srv/salt-master`

A master-only copy of the *`file_roots`* dictionary, used by the state compiler.

```
master_roots: /srv/salt-master
```

roots: Master's Local File Server

`file_roots`

Default:

```
base:
  - /srv/salt
```

Salt runs a lightweight file server written in ZeroMQ to deliver files to minions. This file server is built into the master daemon and does not require a dedicated port.

The file server works on environments passed to the master. Each environment can have multiple root directories. The subdirectories in the multiple file roots cannot match, otherwise the downloaded files will not be able to be reliably ensured. A base environment is required to house the top file.

As of 2018.3.5 and 2019.2.1, it is possible to have __*env*__ as a catch-all environment.

Example:

```
file_roots:
  base:
    - /srv/salt
  dev:
    - /srv/salt/dev/services
    - /srv/salt/dev/states
  prod:
    - /srv/salt/prod/services
    - /srv/salt/prod/states
  __env__:
    - /srv/salt/default
```

Note: For masterless Salt, this parameter must be specified in the minion config file.

`roots_update_interval`

New in version 2018.3.0.

Default: `60`

This option defines the update interval (in seconds) for *`file_roots`*.

Note: Since `file_roots` consists of files local to the minion, the update process for this fileserver backend just reaps the cache for this backend.

```
roots_update_interval: 120
```

gitfs: Git Remote File Server Backend

`gitfs_remotes`

Default: `[]`

When using the `git` fileserver backend at least one git remote needs to be defined. The user running the salt master will need read access to the repo.

The repos will be searched in order to find the file requested by a client and the first repo to have the file will return it. Branches and tags are translated into salt environments.

```
gitfs_remotes:
  - git://github.com/saltstack/salt-states.git
  - file:///var/git/saltmaster
```

Note: `file://` repos will be treated as a remote and copied into the master's gitfs cache, so only the *local* refs for those repos will be exposed as fileserver environments.

As of 2014.7.0, it is possible to have per-repo versions of several of the gitfs configuration parameters. For more information, see the *GitFS Walkthrough*.

`gitfs_provider`

New in version 2014.7.0.

Optional parameter used to specify the provider to be used for gitfs. More information can be found in the *GitFS Walkthrough*.

Must be either `pygit2` or `gitpython`. If unset, then each will be tried in that same order, and the first one with a compatible version installed will be the provider that is used.

```
gitfs_provider: gitpython
```

`gitfs_ssl_verify`

Default: `True`

Specifies whether or not to ignore SSL certificate errors when fetching from the repositories configured in *`gitfs_remotes`*. The `False` setting is useful if you're using a git repo that uses a self-signed certificate. However, keep in mind that setting this to anything other `True` is a considered insecure, and using an SSH-based transport (if available) may be a better option.

```
gitfs_ssl_verify: False
```

Note: pygit2 only supports disabling SSL verification in versions 0.23.2 and newer.

Changed in version 2015.8.0: This option can now be configured on individual repositories as well. See *here* for more info.

Changed in version 2016.11.0: The default config value changed from `False` to `True`.

`gitfs_mountpoint`

New in version 2014.7.0.

Default: `''`

Specifies a path on the salt fileserver which will be prepended to all files served by gitfs. This option can be used in conjunction with *`gitfs_root`*. It can also be configured for an individual repository, see *here* for more info.

```
gitfs_mountpoint: salt://foo/bar
```

Note: The `salt://` protocol designation can be left off (in other words, `foo/bar` and `salt://foo/bar` are equivalent). Assuming a file `baz.sh` in the root of a gitfs remote, and the above example mountpoint, this file would be served up via `salt://foo/bar/baz.sh`.

`gitfs_root`

Default: `''`

Relative path to a subdirectory within the repository from which Salt should begin to serve files. This is useful when there are files in the repository that should not be available to the Salt fileserver. Can be used in conjunction with *`gitfs_mountpoint`*. If used, then from Salt's perspective the directories above the one specified will be ignored and the relative path will (for the purposes of gitfs) be considered as the root of the repo.

```
gitfs_root: somefolder/otherfolder
```

Changed in version 2014.7.0: This option can now be configured on individual repositories as well. See *here* for more info.

`gitfs_base`

Default: `master`

Defines which branch/tag should be used as the `base` environment.

```
gitfs_base: salt
```

Changed in version 2014.7.0: This option can now be configured on individual repositories as well. See *here* for more info.

`gitfs_saltenv`

New in version 2016.11.0.

Default: `[]`

Global settings for *per-saltenv configuration parameters*. Though per-saltenv configuration parameters are typically one-off changes specific to a single gitfs remote, and thus more often configured on a per-remote basis, this parameter can be used to specify per-saltenv changes which should apply to all remotes. For example, the below configuration will map the `develop` branch to the `dev` saltenv for all gitfs remotes.

```
gitfs_saltenv:
  - dev:
    - ref: develop
```

`gitfs_disable_saltenv_mapping`

New in version 2018.3.0.

Default: `False`

When set to `True`, all saltenv mapping logic is disregarded (aside from which branch/tag is mapped to the `base` saltenv). To use any other environments, they must then be defined using *per-saltenv configuration parameters*.

```
gitfs_disable_saltenv_mapping: True
```

Note: This is is a global configuration option, see *here* for examples of configuring it for individual repositories.

`gitfs_ref_types`

New in version 2018.3.0.

Default: `['branch', 'tag', 'sha']`

This option defines what types of refs are mapped to fileserver environments (i.e. saltenvs). It also sets the order of preference when there are ambiguously-named refs (i.e. when a branch and tag both have the same name). The below example disables mapping of both tags and SHAs, so that only branches are mapped as saltenvs:

```
gitfs_ref_types:
  - branch
```

Note: This is is a global configuration option, see *here* for examples of configuring it for individual repositories.

Note: `sha` is special in that it will not show up when listing saltenvs (e.g. with the *`fileserver.envs`* runner), but works within states and with *`cp.cache_file`* to retrieve a file from a specific git SHA.

`gitfs_saltenv_whitelist`

New in version 2014.7.0.

Changed in version 2018.3.0: Renamed from `gitfs_env_whitelist` to `gitfs_saltenv_whitelist`

Default: `[]`

Used to restrict which environments are made available. Can speed up state runs if the repos in *`gitfs_remotes`* contain many branches/tags. More information can be found in the *GitFS Walkthrough*.

```
gitfs_saltenv_whitelist:
  - base
  - v1.*
  - 'mybranch\d+'
```

gitfs_saltenv_blacklist

New in version 2014.7.0.

Changed in version 2018.3.0: Renamed from `gitfs_env_blacklist` to `gitfs_saltenv_blacklist`

Default: `[]`

Used to restrict which environments are made available. Can speed up state runs if the repos in *`gitfs_remotes`* contain many branches/tags. More information can be found in the *GitFS Walkthrough*.

```
gitfs_saltenv_blacklist:
  - base
  - v1.*
  - 'mybranch\d+'
```

gitfs_global_lock

New in version 2015.8.9.

Default: `True`

When set to `False`, if there is an update lock for a gitfs remote and the pid written to it is not running on the master, the lock file will be automatically cleared and a new lock will be obtained. When set to `True`, Salt will simply log a warning when there is an update lock present.

On single-master deployments, disabling this option can help automatically deal with instances where the master was shutdown/restarted during the middle of a gitfs update, leaving a update lock in place.

However, on multi-master deployments with the gitfs cachedir shared via GlusterFS, nfs, or another network filesystem, it is strongly recommended not to disable this option as doing so will cause lock files to be removed if they were created by a different master.

```
# Disable global lock
gitfs_global_lock: False
```

gitfs_update_interval

New in version 2018.3.0.

Default: `60`

This option defines the default update interval (in seconds) for gitfs remotes. The update interval can also be set for a single repository via a *per-remote config option*

```
gitfs_update_interval: 120
```

GitFS Authentication Options

These parameters only currently apply to the pygit2 gitfs provider. Examples of how to use these can be found in the *GitFS Walkthrough*.

`gitfs_user`

New in version 2014.7.0.

Default: `''`

Along with `gitfs_password`, is used to authenticate to HTTPS remotes.

```
gitfs_user: git
```

Note: This is is a global configuration option, see *here* for examples of configuring it for individual repositories.

`gitfs_password`

New in version 2014.7.0.

Default: `''`

Along with `gitfs_user`, is used to authenticate to HTTPS remotes. This parameter is not required if the repository does not use authentication.

```
gitfs_password: mypassword
```

Note: This is is a global configuration option, see *here* for examples of configuring it for individual repositories.

`gitfs_insecure_auth`

New in version 2014.7.0.

Default: `False`

By default, Salt will not authenticate to an HTTP (non-HTTPS) remote. This parameter enables authentication over HTTP. **Enable this at your own risk.**

```
gitfs_insecure_auth: True
```

Note: This is is a global configuration option, see *here* for examples of configuring it for individual repositories.

`gitfs_pubkey`

New in version 2014.7.0.

Default: `''`

Along with *`gitfs_privkey`* (and optionally *`gitfs_passphrase`*), is used to authenticate to SSH remotes. Required for SSH remotes.

```
gitfs_pubkey: /path/to/key.pub
```

Note: This is is a global configuration option, see *here* for examples of configuring it for individual repositories.

`gitfs_privkey`

New in version 2014.7.0.

Default: `''`

Along with *`gitfs_pubkey`* (and optionally *`gitfs_passphrase`*), is used to authenticate to SSH remotes. Required for SSH remotes.

```
gitfs_privkey: /path/to/key
```

Note: This is is a global configuration option, see *here* for examples of configuring it for individual repositories.

`gitfs_passphrase`

New in version 2014.7.0.

Default: `''`

This parameter is optional, required only when the SSH key being used to authenticate is protected by a passphrase.

```
gitfs_passphrase: mypassphrase
```

Note: This is is a global configuration option, see *here* for examples of configuring it for individual repositories.

`gitfs_refspecs`

New in version 2017.7.0.

Default: `['+refs/heads/*:refs/remotes/origin/*', '+refs/tags/*:refs/tags/*']`

When fetching from remote repositories, by default Salt will fetch branches and tags. This parameter can be used to override the default and specify alternate refspecs to be fetched. More information on how this feature works can be found in the *GitFS Walkthrough*.

```
gitfs_refspecs:
  - '+refs/heads/*:refs/remotes/origin/*'
  - '+refs/tags/*:refs/tags/*'
  - '+refs/pull/*/head:refs/remotes/origin/pr/*'
  - '+refs/pull/*/merge:refs/remotes/origin/merge/*'
```

hgfs: Mercurial Remote File Server Backend

`hgfs_remotes`

New in version 0.17.0.

Default: `[]`

When using the `hg` fileserver backend at least one mercurial remote needs to be defined. The user running the salt master will need read access to the repo.

The repos will be searched in order to find the file requested by a client and the first repo to have the file will return it. Branches and/or bookmarks are translated into salt environments, as defined by the *`hgfs_branch_method`* parameter.

```
hgfs_remotes:
  - https://username@bitbucket.org/username/reponame
```

Note: As of 2014.7.0, it is possible to have per-repo versions of the *`hgfs_root`*, *`hgfs_mountpoint`*, *`hgfs_base`*, and *`hgfs_branch_method`* parameters. For example:

```
hgfs_remotes:
  - https://username@bitbucket.org/username/repo1
    - base: saltstates
  - https://username@bitbucket.org/username/repo2:
    - root: salt
    - mountpoint: salt://foo/bar/baz
  - https://username@bitbucket.org/username/repo3:
    - root: salt/states
    - branch_method: mixed
```

`hgfs_branch_method`

New in version 0.17.0.

Default: `branches`

Defines the objects that will be used as fileserver environments.

- `branches` - Only branches and tags will be used
- `bookmarks` - Only bookmarks and tags will be used
- `mixed` - Branches, bookmarks, and tags will be used

```
hgfs_branch_method: mixed
```

Note: Starting in version 2014.1.0, the value of the *hgfs_base* parameter defines which branch is used as the `base` environment, allowing for a `base` environment to be used with an *hgfs_branch_method* of `bookmarks`.

Prior to this release, the `default` branch will be used as the `base` environment.

---

hgfs_mountpoint

New in version 2014.7.0.

Default: `''`

Specifies a path on the salt fileserver which will be prepended to all files served by hgfs. This option can be used in conjunction with *hgfs_root*. It can also be configured on a per-remote basis, see *here* for more info.

```
hgfs_mountpoint: salt://foo/bar
```

Note: The `salt://` protocol designation can be left off (in other words, `foo/bar` and `salt://foo/bar` are equivalent). Assuming a file `baz.sh` in the root of an hgfs remote, this file would be served up via `salt://foo/bar/baz.sh`.

---

hgfs_root

New in version 0.17.0.

Default: `''`

Relative path to a subdirectory within the repository from which Salt should begin to serve files. This is useful when there are files in the repository that should not be available to the Salt fileserver. Can be used in conjunction with *hgfs_mountpoint*. If used, then from Salt's perspective the directories above the one specified will be ignored and the relative path will (for the purposes of hgfs) be considered as the root of the repo.

```
hgfs_root: somefolder/otherfolder
```

Changed in version 2014.7.0: Ability to specify hgfs roots on a per-remote basis was added. See *here* for more info.

hgfs_base

New in version 2014.1.0.

Default: `default`

Defines which branch should be used as the `base` environment. Change this if *hgfs_branch_method* is set to `bookmarks` to specify which bookmark should be used as the `base` environment.

```
hgfs_base: salt
```

hgfs_saltenv_whitelist

New in version 2014.7.0.

Changed in version 2018.3.0: Renamed from `hgfs_env_whitelist` to `hgfs_saltenv_whitelist`

Default: `[]`

Used to restrict which environments are made available. Can speed up state runs if your hgfs remotes contain many branches/bookmarks/tags. Full names, globs, and regular expressions are supported. If using a regular expression, the expression must match the entire minion ID.

If used, only branches/bookmarks/tags which match one of the specified expressions will be exposed as fileserver environments.

If used in conjunction with *hgfs_saltenv_blacklist*, then the subset of branches/bookmarks/tags which match the whitelist but do *not* match the blacklist will be exposed as fileserver environments.

```
hgfs_saltenv_whitelist:
  - base
  - v1.*
  - 'mybranch\d+'
```

hgfs_saltenv_blacklist

New in version 2014.7.0.

Changed in version 2018.3.0: Renamed from `hgfs_env_blacklist` to `hgfs_saltenv_blacklist`

Default: `[]`

Used to restrict which environments are made available. Can speed up state runs if your hgfs remotes contain many branches/bookmarks/tags. Full names, globs, and regular expressions are supported. If using a regular expression, the expression must match the entire minion ID.

If used, branches/bookmarks/tags which match one of the specified expressions will *not* be exposed as fileserver environments.

If used in conjunction with *hgfs_saltenv_whitelist*, then the subset of branches/bookmarks/tags which match the whitelist but do *not* match the blacklist will be exposed as fileserver environments.

```
hgfs_saltenv_blacklist:
  - base
  - v1.*
  - 'mybranch\d+'
```

hgfs_update_interval

New in version 2018.3.0.

Default: `60`

This option defines the update interval (in seconds) for *hgfs_remotes*.

```
hgfs_update_interval: 120
```

svnfs: Subversion Remote File Server Backend

`svnfs_remotes`

New in version 0.17.0.

Default: `[]`

When using the `svn` fileserver backend at least one subversion remote needs to be defined. The user running the salt master will need read access to the repo.

The repos will be searched in order to find the file requested by a client and the first repo to have the file will return it. The trunk, branches, and tags become environments, with the trunk being the `base` environment.

```
svnfs_remotes:
  - svn://foo.com/svn/myproject
```

Note: As of 2014.7.0, it is possible to have per-repo versions of the following configuration parameters:

- *`svnfs_root`*
- *`svnfs_mountpoint`*
- *`svnfs_trunk`*
- *`svnfs_branches`*
- *`svnfs_tags`*

For example:

```
svnfs_remotes:
  - svn://foo.com/svn/project1
  - svn://foo.com/svn/project2:
    - root: salt
    - mountpoint: salt://foo/bar/baz
  - svn//foo.com/svn/project3:
    - root: salt/states
    - branches: branch
    - tags: tag
```

`svnfs_mountpoint`

New in version 2014.7.0.

Default: `''`

Specifies a path on the salt fileserver which will be prepended to all files served by hgfs. This option can be used in conjunction with *`svnfs_root`*. It can also be configured on a per-remote basis, see *`here`* for more info.

```
svnfs_mountpoint: salt://foo/bar
```

Note: The `salt://` protocol designation can be left off (in other words, `foo/bar` and `salt://foo/bar` are equivalent). Assuming a file `baz.sh` in the root of an svnfs remote, this file would be served up via `salt://foo/bar/baz.sh`.

`svnfs_root`

New in version 0.17.0.

Default: `''`

Relative path to a subdirectory within the repository from which Salt should begin to serve files. This is useful when there are files in the repository that should not be available to the Salt fileserver. Can be used in conjunction with *`svnfs_mountpoint`*. If used, then from Salt's perspective the directories above the one specified will be ignored and the relative path will (for the purposes of svnfs) be considered as the root of the repo.

```
svnfs_root: somefolder/otherfolder
```

Changed in version 2014.7.0: Ability to specify svnfs roots on a per-remote basis was added. See *`here`* for more info.

`svnfs_trunk`

New in version 2014.7.0.

Default: `trunk`

Path relative to the root of the repository where the trunk is located. Can also be configured on a per-remote basis, see *`here`* for more info.

```
svnfs_trunk: trunk
```

`svnfs_branches`

New in version 2014.7.0.

Default: `branches`

Path relative to the root of the repository where the branches are located. Can also be configured on a per-remote basis, see *`here`* for more info.

```
svnfs_branches: branches
```

`svnfs_tags`

New in version 2014.7.0.

Default: `tags`

Path relative to the root of the repository where the tags are located. Can also be configured on a per-remote basis, see *`here`* for more info.

```
svnfs_tags: tags
```

`svnfs_saltenv_whitelist`

New in version 2014.7.0.

Changed in version 2018.3.0: Renamed from `svnfs_env_whitelist` to `svnfs_saltenv_whitelist`

Default: `[]`

Used to restrict which environments are made available. Can speed up state runs if your svnfs remotes contain many branches/tags. Full names, globs, and regular expressions are supported. If using a regular expression, the expression must match the entire minion ID.

If used, only branches/tags which match one of the specified expressions will be exposed as fileserver environments.

If used in conjunction with `svnfs_saltenv_blacklist`, then the subset of branches/tags which match the whitelist but do *not* match the blacklist will be exposed as fileserver environments.

```
svnfs_saltenv_whitelist:
  - base
  - v1.*
  - 'mybranch\d+'
```

`svnfs_saltenv_blacklist`

New in version 2014.7.0.

Changed in version 2018.3.0: Renamed from `svnfs_env_blacklist` to `svnfs_saltenv_blacklist`

Default: `[]`

Used to restrict which environments are made available. Can speed up state runs if your svnfs remotes contain many branches/tags. Full names, globs, and regular expressions are supported. If using a regular expression, the expression must match the entire minion ID.

If used, branches/tags which match one of the specified expressions will *not* be exposed as fileserver environments.

If used in conjunction with `svnfs_saltenv_whitelist`, then the subset of branches/tags which match the whitelist but do *not* match the blacklist will be exposed as fileserver environments.

```
svnfs_saltenv_blacklist:
  - base
  - v1.*
  - 'mybranch\d+'
```

`svnfs_update_interval`

New in version 2018.3.0.

Default: `60`

This option defines the update interval (in seconds) for `svnfs_remotes`.

```
svnfs_update_interval: 120
```

minionfs: MinionFS Remote File Server Backend

`minionfs_env`

New in version 2014.7.0.

Default: `base`

Environment from which MinionFS files are made available.

```
minionfs_env: minionfs
```

`minionfs_mountpoint`

New in version 2014.7.0.

Default: `''`

Specifies a path on the salt fileserver from which minionfs files are served.

```
minionfs_mountpoint: salt://foo/bar
```

Note: The `salt://` protocol designation can be left off (in other words, `foo/bar` and `salt://foo/bar` are equivalent).

`minionfs_whitelist`

New in version 2014.7.0.

Default: `[]`

Used to restrict which minions' pushed files are exposed via minionfs. If using a regular expression, the expression must match the entire minion ID.

If used, only the pushed files from minions which match one of the specified expressions will be exposed.

If used in conjunction with *`minionfs_blacklist`*, then the subset of hosts which match the whitelist but do *not* match the blacklist will be exposed.

```
minionfs_whitelist:
  - server01
  - dev*
  - 'mail\d+.mydomain.tld'
```

`minionfs_blacklist`

New in version 2014.7.0.

Default: `[]`

Used to restrict which minions' pushed files are exposed via minionfs. If using a regular expression, the expression must match the entire minion ID.

If used, only the pushed files from minions which match one of the specified expressions will *not* be exposed.

If used in conjunction with *`minionfs_whitelist`*, then the subset of hosts which match the whitelist but do *not* match the blacklist will be exposed.

```
minionfs_blacklist:
  - server01
  - dev*
  - 'mail\d+.mydomain.tld'
```

`minionfs_update_interval`

New in version 2018.3.0.

Default: `60`

This option defines the update interval (in seconds) for *MinionFS*.

Note: Since *MinionFS* consists of files local to the master, the update process for this fileserver backend just reaps the cache for this backend.

```
minionfs_update_interval: 120
```

azurefs: Azure File Server Backend

New in version 2015.8.0.

See the *`azurefs documentation`* for usage examples.

`azurefs_update_interval`

New in version 2018.3.0.

Default: `60`

This option defines the update interval (in seconds) for azurefs.

```
azurefs_update_interval: 120
```

s3fs: S3 File Server Backend

New in version 0.16.0.

See the *`s3fs documentation`* for usage examples.

`s3fs_update_interval`

New in version 2018.3.0.

Default: `60`

This option defines the update interval (in seconds) for s3fs.

```
s3fs_update_interval: 120
```

3.1.8 Pillar Configuration

`pillar_roots`

Default:

```
base:
  - /srv/pillar
```

Set the environments and directories used to hold pillar sls data. This configuration is the same as `file_roots`:

As of 2017.7.5 and 2018.3.1, it is possible to have *__env__* as a catch-all environment.

Example:

```
pillar_roots:
  base:
    - /srv/pillar
  dev:
    - /srv/pillar/dev
  prod:
    - /srv/pillar/prod
  __env__:
    - /srv/pillar/others
```

on_demand_ext_pillar

New in version 2016.3.6,2016.11.3,2017.7.0.

Default: `['libvirt', 'virtkey']`

The external pillars permitted to be used on-demand using `pillar.ext`.

```
on_demand_ext_pillar:
  - libvirt
  - virtkey
  - git
```

Warning: This will allow minions to request specific pillar data via `pillar.ext`, and may be considered a security risk. However, pillar data generated in this way will not affect the *in-memory pillar data*, so this risk is limited to instances in which states/modules/etc. (built-in or custom) rely upon pillar data generated by `pillar.ext`.

decrypt_pillar

New in version 2017.7.0.

Default: `[]`

A list of paths to be recursively decrypted during pillar compilation.

```
decrypt_pillar:
  - 'foo:bar': gpg
  - 'lorem:ipsum:dolor'
```

Entries in this list can be formatted either as a simple string, or as a key/value pair, with the key being the pillar location, and the value being the renderer to use for pillar decryption. If the former is used, the renderer specified by `decrypt_pillar_default` will be used.

`decrypt_pillar_delimiter`

New in version 2017.7.0.

Default: `:`

The delimiter used to distinguish nested data structures in the *decrypt_pillar* option.

```
decrypt_pillar_delimiter: '|'
decrypt_pillar:
  - 'foo|bar': gpg
  - 'lorem|ipsum|dolor'
```

`decrypt_pillar_default`

New in version 2017.7.0.

Default: `gpg`

The default renderer used for decryption, if one is not specified for a given pillar key in *decrypt_pillar*.

```
decrypt_pillar_default: my_custom_renderer
```

`decrypt_pillar_renderers`

New in version 2017.7.0.

Default: `['gpg']`

List of renderers which are permitted to be used for pillar decryption.

```
decrypt_pillar_renderers:
  - gpg
  - my_custom_renderer
```

`pillar_opts`

Default: `False`

The `pillar_opts` option adds the master configuration file data to a dict in the pillar called `master`. This can be used to set simple configurations in the master config file that can then be used on minions.

Note that setting this option to `True` means the master config file will be included in all minion's pillars. While this makes global configuration of services and systems easy, it may not be desired if sensitive data is stored in the master configuration.

```
pillar_opts: False
```

`pillar_safe_render_error`

Default: `True`

The pillar_safe_render_error option prevents the master from passing pillar render errors to the minion. This is set on by default because the error could contain templating data which would give that minion information it shouldn't have, like a password! When set `True` the error message will only show:

```
Rendering SLS 'my.sls' failed. Please see master log for details.
```

```
pillar_safe_render_error: True
```

`ext_pillar`

The ext_pillar option allows for any number of external pillar interfaces to be called when populating pillar data. The configuration is based on ext_pillar functions. The available ext_pillar functions can be found herein:

https://github.com/saltstack/salt/blob/master/salt/pillar

By default, the ext_pillar interface is not configured to run.

Default: `[]`

```
ext_pillar:
  - hiera: /etc/hiera.yaml
  - cmd_yaml: cat /etc/salt/yaml
  - reclass:
      inventory_base_uri: /etc/reclass
```

There are additional details at *Pillars*

`ext_pillar_first`

New in version 2015.5.0.

Default: `False`

This option allows for external pillar sources to be evaluated before *`pillar_roots`*. External pillar data is evaluated separately from *`pillar_roots`* pillar data, and then both sets of pillar data are merged into a single pillar dictionary, so the value of this config option will have an impact on which key "wins" when there is one of the same name in both the external pillar data and *`pillar_roots`* pillar data. By setting this option to `True`, ext_pillar keys will be overridden by *`pillar_roots`*, while leaving it as `False` will allow ext_pillar keys to override those from *`pillar_roots`*.

Note: For a while, this config option did not work as specified above, because of a bug in Pillar compilation. This bug has been resolved in version 2016.3.4 and later.

```
ext_pillar_first: False
```

`pillarenv_from_saltenv`

Default: `False`

When set to `True`, the `pillarenv` value will assume the value of the effective saltenv when running states. This essentially makes `salt-run pillar.show_pillar saltenv=dev` equivalent to `salt-run pillar.show_pillar saltenv=dev pillarenv=dev`. If `pillarenv` is set on the CLI, it will override this option.

```
pillarenv_from_saltenv: True
```

Note: For salt remote execution commands this option should be set in the Minion configuration instead.

`pillar_raise_on_missing`

New in version 2015.5.0.

Default: `False`

Set this option to `True` to force a `KeyError` to be raised whenever an attempt to retrieve a named value from pillar fails. When this option is set to `False`, the failed attempt returns an empty string.

Git External Pillar (git_pillar) Configuration Options

`git_pillar_provider`

New in version 2015.8.0.

Specify the provider to be used for git_pillar. Must be either `pygit2` or `gitpython`. If unset, then both will be tried in that same order, and the first one with a compatible version installed will be the provider that is used.

```
git_pillar_provider: gitpython
```

`git_pillar_base`

New in version 2015.8.0.

Default: `master`

If the desired branch matches this value, and the environment is omitted from the git_pillar configuration, then the environment for that git_pillar remote will be `base`. For example, in the configuration below, the `foo` branch/tag would be assigned to the `base` environment, while `bar` would be mapped to the `bar` environment.

```
git_pillar_base: foo

ext_pillar:
  - git:
    - foo https://mygitserver/git-pillar.git
    - bar https://mygitserver/git-pillar.git
```

`git_pillar_branch`

New in version 2015.8.0.

Default: `master`

If the branch is omitted from a git_pillar remote, then this branch will be used instead. For example, in the configuration below, the first two remotes would use the `pillardata` branch/tag, while the third would use the `foo` branch/tag.

```
git_pillar_branch: pillardata

ext_pillar:
  - git:
    - https://mygitserver/pillar1.git
    - https://mygitserver/pillar2.git:
      - root: pillar
    - foo https://mygitserver/pillar3.git
```

git_pillar_env

New in version 2015.8.0.

Default: `''` (unset)

Environment to use for git_pillar remotes. This is normally derived from the branch/tag (or from a per-remote `env` parameter), but if set this will override the process of deriving the env from the branch/tag name. For example, in the configuration below the `foo` branch would be assigned to the `base` environment, while the `bar` branch would need to explicitly have `bar` configured as it's environment to keep it from also being mapped to the `base` environment.

```
git_pillar_env: base

ext_pillar:
  - git:
    - foo https://mygitserver/git-pillar.git
    - bar https://mygitserver/git-pillar.git:
      - env: bar
```

For this reason, this option is recommended to be left unset, unless the use case calls for all (or almost all) of the git_pillar remotes to use the same environment irrespective of the branch/tag being used.

git_pillar_root

New in version 2015.8.0.

Default: `''`

Path relative to the root of the repository where the git_pillar top file and SLS files are located. In the below configuration, the pillar top file and SLS files would be looked for in a subdirectory called `pillar`.

```
git_pillar_root: pillar

ext_pillar:
  - git:
    - master https://mygitserver/pillar1.git
    - master https://mygitserver/pillar2.git
```

Note: This is a global option. If only one or two repos need to have their files sourced from a subdirectory, then *`git_pillar_root`* can be omitted and the root can be specified on a per-remote basis, like so:

```
ext_pillar:
  - git:
    - master https://mygitserver/pillar1.git
```

(continues on next page)

(continued from previous page)

```
    - master https://mygitserver/pillar2.git:
      - root: pillar
```

In this example, for the first remote the top file and SLS files would be looked for in the root of the repository, while in the second remote the pillar data would be retrieved from the `pillar` subdirectory.

git_pillar_ssl_verify

New in version 2015.8.0.

Changed in version 2016.11.0.

Default: `False`

Specifies whether or not to ignore SSL certificate errors when contacting the remote repository. The `False` setting is useful if you're using a git repo that uses a self-signed certificate. However, keep in mind that setting this to anything other `True` is a considered insecure, and using an SSH-based transport (if available) may be a better option.

In the 2016.11.0 release, the default config value changed from `False` to `True`.

```
git_pillar_ssl_verify: True
```

Note: pygit2 only supports disabling SSL verification in versions 0.23.2 and newer.

git_pillar_global_lock

New in version 2015.8.9.

Default: `True`

When set to `False`, if there is an update/checkout lock for a git_pillar remote and the pid written to it is not running on the master, the lock file will be automatically cleared and a new lock will be obtained. When set to `True`, Salt will simply log a warning when there is an lock present.

On single-master deployments, disabling this option can help automatically deal with instances where the master was shutdown/restarted during the middle of a git_pillar update/checkout, leaving a lock in place.

However, on multi-master deployments with the git_pillar cachedir shared via GlusterFS, nfs, or another network filesystem, it is strongly recommended not to disable this option as doing so will cause lock files to be removed if they were created by a different master.

```
# Disable global lock
git_pillar_global_lock: False
```

git_pillar_includes

New in version 2017.7.0.

Default: `True`

Normally, when processing *git_pillar remotes*, if more than one repo under the same `git` section in the `ext_pillar` configuration refers to the same pillar environment, then each repo in a given environment will have access to the other repos' files to be referenced in their top files. However, it may be desirable to disable this behavior. If so, set this value to `False`.

For a more detailed examination of how includes work, see *this explanation* from the git_pillar documentation.

```
git_pillar_includes: False
```

`git_pillar_update_interval`

New in version 3000.

Default: `60`

This option defines the default update interval (in seconds) for git_pillar remotes. The update is handled within the global loop, hence `git_pillar_update_interval` should be a multiple of `loop_interval`.

```
git_pillar_update_interval: 120
```

Git External Pillar Authentication Options

These parameters only currently apply to the `pygit2` *`git_pillar_provider`*. Authentication works the same as it does in gitfs, as outlined in the *GitFS Walkthrough*, though the global configuration options are named differently to reflect that they are for git_pillar instead of gitfs.

`git_pillar_user`

New in version 2015.8.0.

Default: `''`

Along with *`git_pillar_password`*, is used to authenticate to HTTPS remotes.

```
git_pillar_user: git
```

`git_pillar_password`

New in version 2015.8.0.

Default: `''`

Along with *`git_pillar_user`*, is used to authenticate to HTTPS remotes. This parameter is not required if the repository does not use authentication.

```
git_pillar_password: mypassword
```

`git_pillar_insecure_auth`

New in version 2015.8.0.

Default: `False`

By default, Salt will not authenticate to an HTTP (non-HTTPS) remote. This parameter enables authentication over HTTP. **Enable this at your own risk.**

```
git_pillar_insecure_auth: True
```

git_pillar_pubkey

New in version 2015.8.0.

Default: `''`

Along with *git_pillar_privkey* (and optionally *git_pillar_passphrase*), is used to authenticate to SSH remotes.

```
git_pillar_pubkey: /path/to/key.pub
```

git_pillar_privkey

New in version 2015.8.0.

Default: `''`

Along with *git_pillar_pubkey* (and optionally *git_pillar_passphrase*), is used to authenticate to SSH remotes.

```
git_pillar_privkey: /path/to/key
```

git_pillar_passphrase

New in version 2015.8.0.

Default: `''`

This parameter is optional, required only when the SSH key being used to authenticate is protected by a passphrase.

```
git_pillar_passphrase: mypassphrase
```

git_pillar_refspecs

New in version 2017.7.0.

Default: `['+refs/heads/*:refs/remotes/origin/*', '+refs/tags/*:refs/tags/*']`

When fetching from remote repositories, by default Salt will fetch branches and tags. This parameter can be used to override the default and specify alternate refspecs to be fetched. This parameter works similarly to its *GitFS counterpart*, in that it can be configured both globally and for individual remotes.

```
git_pillar_refspecs:
  - '+refs/heads/*:refs/remotes/origin/*'
  - '+refs/tags/*:refs/tags/*'
  - '+refs/pull/*/head:refs/remotes/origin/pr/*'
  - '+refs/pull/*/merge:refs/remotes/origin/merge/*'
```

`git_pillar_verify_config`

New in version 2017.7.0.

Default: `True`

By default, as the master starts it performs some sanity checks on the configured git_pillar repositories. If any of these sanity checks fail (such as when an invalid configuration is used), the master daemon will abort.

To skip these sanity checks, set this option to `False`.

```
git_pillar_verify_config: False
```

Pillar Merging Options

`pillar_source_merging_strategy`

New in version 2014.7.0.

Default: `smart`

The pillar_source_merging_strategy option allows you to configure merging strategy between different sources. It accepts 5 values:

- `none`:

  It will not do any merging at all and only parse the pillar data from the passed environment and 'base' if no environment was specified.

  New in version 2016.3.4.

- `recurse`:

  It will recursively merge data. For example, theses 2 sources:

  ```
  foo: 42
  bar:
      element1: True
  ```

  ```
  bar:
      element2: True
  baz: quux
  ```

  will be merged as:

  ```
  foo: 42
  bar:
      element1: True
      element2: True
  baz: quux
  ```

- `aggregate`:

  instructs aggregation of elements between sources that use the #!yamlex renderer.

  For example, these two documents:

```
#!yamlex
foo: 42
bar: !aggregate {
  element1: True
}
baz: !aggregate quux
```

```
#!yamlex
bar: !aggregate {
  element2: True
}
baz: !aggregate quux2
```

will be merged as:

```
foo: 42
bar:
  element1: True
  element2: True
baz:
  - quux
  - quux2
```

- `overwrite`:

  Will use the behaviour of the 2014.1 branch and earlier.

  Overwrites elements according the order in which they are processed.

  First pillar processed:

```
A:
  first_key: blah
  second_key: blah
```

  Second pillar processed:

```
A:
  third_key: blah
  fourth_key: blah
```

  will be merged as:

```
A:
  third_key: blah
  fourth_key: blah
```

- `smart` (default):

  Guesses the best strategy based on the "renderer" setting.

Note: In order for yamlex based features such as `!aggregate` to work as expected across documents using the default `smart` merge strategy, the *`renderer`* config option must be set to `jinja|yamlex` or similar.

`pillar_merge_lists`

New in version 2015.8.0.

Default: `False`

Recursively merge lists by aggregating them instead of replacing them.

```
pillar_merge_lists: False
```

`pillar_includes_override_sls`

New in version 2017.7.6,2018.3.1.

Default: `False`

Prior to version 2017.7.3, keys from *pillar includes* would be merged on top of the pillar SLS. Since 2017.7.3, the includes are merged together and then the pillar SLS is merged on top of that.

Set this option to `True` to return to the old behavior.

```
pillar_includes_override_sls: True
```

Pillar Cache Options

`pillar_cache`

New in version 2015.8.8.

Default: `False`

A master can cache pillars locally to bypass the expense of having to render them for each minion on every request. This feature should only be enabled in cases where pillar rendering time is known to be unsatisfactory and any attendant security concerns about storing pillars in a master cache have been addressed.

When enabling this feature, be certain to read through the additional `pillar_cache_*` configuration options to fully understand the tunable parameters and their implications.

```
pillar_cache: False
```

Note: Setting `pillar_cache: True` has no effect on *targeting minions with pillar*.

`pillar_cache_ttl`

New in version 2015.8.8.

Default: `3600`

If and only if a master has set `pillar_cache: True`, the cache TTL controls the amount of time, in seconds, before the cache is considered invalid by a master and a fresh pillar is recompiled and stored.

`pillar_cache_backend`

New in version 2015.8.8.

Default: `disk`

If an only if a master has set `pillar_cache: True`, one of several storage providers can be utilized:

- `disk` (default):

  The default storage backend. This caches rendered pillars to the master cache. Rendered pillars are serialized and deserialized as `msgpack` structures for speed. Note that pillars are stored UNENCRYPTED. Ensure that the master cache has permissions set appropriately (sane defaults are provided).

- `memory` [EXPERIMENTAL]:

  An optional backend for pillar caches which uses a pure-Python in-memory data structure for maximal performance. There are several caveats, however. First, because each master worker contains its own in-memory cache, there is no guarantee of cache consistency between minion requests. This works best in situations where the pillar rarely if ever changes. Secondly, and perhaps more importantly, this means that unencrypted pillars will be accessible to any process which can examine the memory of the `salt-master`! This may represent a substantial security risk.

```
pillar_cache_backend: disk
```

3.1.9 Master Reactor Settings

`reactor`

Default: `[]`

Defines a salt reactor. See the *Reactor* documentation for more information.

```
reactor:
  - 'salt/minion/*/start':
    - salt://reactor/startup_tasks.sls
```

`reactor_refresh_interval`

Default: `60`

The TTL for the cache of the reactor configuration.

```
reactor_refresh_interval: 60
```

`reactor_worker_threads`

Default: `10`

The number of workers for the runner/wheel in the reactor.

```
reactor_worker_threads: 10
```

`reactor_worker_hwm`

Default: `10000`

The queue size for workers in the reactor.

```
reactor_worker_hwm: 10000
```

3.1.10 Salt-API Master Settings

There are some settings for *salt-api* that can be configured on the Salt Master.

`api_logfile`

Default: `/var/log/salt/api`

The logfile location for `salt-api`.

```
api_logfile: /var/log/salt/api
```

`api_pidfile`

Default: /var/run/salt-api.pid

If this master will be running `salt-api`, specify the pidfile of the `salt-api` daemon.

```
api_pidfile: /var/run/salt-api.pid
```

`rest_timeout`

Default: `300`

Used by `salt-api` for the master requests timeout.

```
rest_timeout: 300
```

3.1.11 Syndic Server Settings

A Salt syndic is a Salt master used to pass commands from a higher Salt master to minions below the syndic. Using the syndic is simple. If this is a master that will have syndic servers(s) below it, set the `order_masters` setting to `True`.

If this is a master that will be running a syndic daemon for passthrough the `syndic_master` setting needs to be set to the location of the master server.

Do not forget that, in other words, it means that it shares with the local minion its ID and PKI directory.

order_masters

Default: `False`

Extra data needs to be sent with publications if the master is controlling a lower level master via a syndic minion. If this is the case the order_masters value must be set to True

```
order_masters: False
```

syndic_master

Changed in version 2016.3.5,2016.11.1: Set default higher level master address.

Default: `masterofmasters`

If this master will be running the `salt-syndic` to connect to a higher level master, specify the higher level master with this configuration value.

```
syndic_master: masterofmasters
```

You can optionally connect a syndic to multiple higher level masters by setting the `syndic_master` value to a list:

```
syndic_master:
  - masterofmasters1
  - masterofmasters2
```

Each higher level master must be set up in a multi-master configuration.

syndic_master_port

Default: `4506`

If this master will be running the `salt-syndic` to connect to a higher level master, specify the higher level master port with this configuration value.

```
syndic_master_port: 4506
```

syndic_pidfile

Default: `/var/run/salt-syndic.pid`

If this master will be running the `salt-syndic` to connect to a higher level master, specify the pidfile of the syndic daemon.

```
syndic_pidfile: /var/run/syndic.pid
```

syndic_log_file

Default: `/var/log/salt/syndic`

If this master will be running the `salt-syndic` to connect to a higher level master, specify the log file of the syndic daemon.

```
syndic_log_file: /var/log/salt-syndic.log
```

syndic_failover

New in version 2016.3.0.

Default: `random`

The behaviour of the multi-syndic when connection to a master of masters failed. Can specify `random` (default) or `ordered`. If set to `random`, masters will be iterated in random order. If `ordered` is specified, the configured order will be used.

```
syndic_failover: random
```

syndic_wait

Default: `5`

The number of seconds for the salt client to wait for additional syndics to check in with their lists of expected minions before giving up.

```
syndic_wait: 5
```

syndic_forward_all_events

New in version 2017.7.0.

Default: `False`

Option on multi-syndic or single when connected to multiple masters to be able to send events to all connected masters.

```
syndic_forward_all_events: False
```

3.1.12 Peer Publish Settings

Salt minions can send commands to other minions, but only if the minion is allowed to. By default "Peer Publication" is disabled, and when enabled it is enabled for specific minions and specific commands. This allows secure compartmentalization of commands based on individual minions.

peer

Default: `{}`

The configuration uses regular expressions to match minions and then a list of regular expressions to match functions. The following will allow the minion authenticated as foo.example.com to execute functions from the test and pkg modules.

```
peer:
  foo.example.com:
      - test.*
      - pkg.*
```

This will allow all minions to execute all commands:

```
peer:
  .*:
      - .*
```

This is not recommended, since it would allow anyone who gets root on any single minion to instantly have root on all of the minions!

By adding an additional layer you can limit the target hosts in addition to the accessible commands:

```
peer:
  foo.example.com:
    'db*':
      - test.*
      - pkg.*
```

`peer_run`

Default: `{}`

The peer_run option is used to open up runners on the master to access from the minions. The peer_run configuration matches the format of the peer configuration.

The following example would allow foo.example.com to execute the manage.up runner:

```
peer_run:
  foo.example.com:
      - manage.up
```

3.1.13 Master Logging Settings

`log_file`

Default: `/var/log/salt/master`

The master log can be sent to a regular file, local path name, or network location. See also *`log_file`*.

Examples:

```
log_file: /var/log/salt/master
```

```
log_file: file:///dev/log
```

```
log_file: udp://loghost:10514
```

`log_level`

Default: `warning`

The level of messages to send to the console. See also *`log_level`*.

```
log_level: warning
```

`log_level_logfile`

Default: `warning`

The level of messages to send to the log file. See also *log_level_logfile*. When it is not set explicitly it will inherit the level set by *log_level* option.

```
log_level_logfile: warning
```

`log_datefmt`

Default: `%H:%M:%S`

The date and time format used in console log messages. See also *log_datefmt*.

```
log_datefmt: '%H:%M:%S'
```

`log_datefmt_logfile`

Default: `%Y-%m-%d %H:%M:%S`

The date and time format used in log file messages. See also *log_datefmt_logfile*.

```
log_datefmt_logfile: '%Y-%m-%d %H:%M:%S'
```

`log_fmt_console`

Default: `[%(levelname)-8s] %(message)s`

The format of the console logging messages. See also *log_fmt_console*.

Note: Log colors are enabled in `log_fmt_console` rather than the *color* config since the logging system is loaded before the master config.

Console log colors are specified by these additional formatters:

%(colorlevel)s %(colorname)s %(colorprocess)s %(colormsg)s

Since it is desirable to include the surrounding brackets, '[' and ']', in the coloring of the messages, these color formatters also include padding as well. Color LogRecord attributes are only available for console logging.

```
log_fmt_console: '%(colorlevel)s %(colormsg)s'
log_fmt_console: '[%(levelname)-8s] %(message)s'
```

`log_fmt_logfile`

Default: `%(asctime)s,%(msecs)03d [%(name)-17s][%(levelname)-8s] %(message)s`

The format of the log file logging messages. See also *log_fmt_logfile*.

```
log_fmt_logfile: '%(asctime)s,%(msecs)03d [%(name)-17s][%(levelname)-8s] %(message)s'
```

`log_granular_levels`

Default: `{}`

This can be used to control logging levels more specifically. See also *`log_granular_levels`*.

`log_rotate_max_bytes`

Default: `0`

The maximum number of bytes a single log file may contain before it is rotated. A value of 0 disables this feature. Currently only supported on Windows. On other platforms, use an external tool such as 'logrotate' to manage log files. `log_rotate_max_bytes`

`log_rotate_backup_count`

Default: `0`

The number of backup files to keep when rotating log files. Only used if *`log_rotate_max_bytes`* is greater than 0. Currently only supported on Windows. On other platforms, use an external tool such as 'logrotate' to manage log files. `log_rotate_backup_count`

3.1.14 Node Groups

`nodegroups`

Default: `{}`

Node groups allow for logical groupings of minion nodes. A group consists of a group name and a compound target.

```
nodegroups:
  group1: 'L@foo.domain.com,bar.domain.com,baz.domain.com or bl*.domain.com'
  group2: 'G@os:Debian and foo.domain.com'
  group3: 'G@os:Debian and N@group1'
  group4:
    - 'G@foo:bar'
    - 'or'
    - 'G@foo:baz'
```

More information on using nodegroups can be found *here*.

3.1.15 Range Cluster Settings

`range_server`

Default: `'range:80'`

The range server (and optional port) that serves your cluster information https://github.com/ytoolshed/range/wiki/%22yamlfile%22-module-file-spec

```
range_server: range:80
```

3.1.16 Include Configuration

Configuration can be loaded from multiple files. The order in which this is done is:

1. The master config file itself
2. The files matching the glob in *`default_include`*
3. The files matching the glob in *`include`* (if defined)

Each successive step overrides any values defined in the previous steps. Therefore, any config options defined in one of the *`default_include`* files would override the same value in the master config file, and any options defined in *`include`* would override both.

`default_include`

Default: `master.d/*.conf`

The master can include configuration from other files. Per default the master will automatically include all config files from `master.d/*.conf` where `master.d` is relative to the directory of the master configuration file.

Note: Salt creates files in the `master.d` directory for its own use. These files are prefixed with an underscore. A common example of this is the `_schedule.conf` file.

`include`

Default: `not defined`

The master can include configuration from other files. To enable this, pass a list of paths to this option. The paths can be either relative or absolute; if relative, they are considered to be relative to the directory the main minion configuration file lives in. Paths can make use of shell-style globbing. If no files are matched by a path passed to this option then the master will log a warning message.

```
# Include files from a master.d directory in the same
# directory as the master config file
include: master.d/*

# Include a single extra file into the configuration
include: /etc/roles/webserver

# Include several files and the master.d directory
include:
  - extra_config
  - master.d/*
  - /etc/roles/webserver
```

3.1.17 Keepalive Settings

`tcp_keepalive`

Default: `True`

The tcp keepalive interval to set on TCP ports. This setting can be used to tune Salt connectivity issues in messy network environments with misbehaving firewalls.

```
tcp_keepalive: True
```

tcp_keepalive_cnt

Default: `-1`

Sets the ZeroMQ TCP keepalive count. May be used to tune issues with minion disconnects.

```
tcp_keepalive_cnt: -1
```

tcp_keepalive_idle

Default: `300`

Sets ZeroMQ TCP keepalive idle. May be used to tune issues with minion disconnects.

```
tcp_keepalive_idle: 300
```

tcp_keepalive_intvl

Default: `-1`

Sets ZeroMQ TCP keepalive interval. May be used to tune issues with minion disconnects.

```
tcp_keepalive_intvl': -1
```

3.1.18 Windows Software Repo Settings

winrepo_provider

New in version 2015.8.0.

Specify the provider to be used for winrepo. Must be either `pygit2` or `gitpython`. If unset, then both will be tried in that same order, and the first one with a compatible version installed will be the provider that is used.

```
winrepo_provider: gitpython
```

winrepo_dir

Changed in version 2015.8.0: Renamed from `win_repo` to `winrepo_dir`.

Default: `/srv/salt/win/repo`

Location on the master where the *`winrepo_remotes`* are checked out for pre-2015.8.0 minions. 2015.8.0 and later minions use *`winrepo_remotes_ng`* instead.

```
winrepo_dir: /srv/salt/win/repo
```

`winrepo_dir_ng`

New in version 2015.8.0: A new *ng* repo was added.

Default: `/srv/salt/win/repo-ng`

Location on the master where the *`winrepo_remotes_ng`* are checked out for 2015.8.0 and later minions.

```
winrepo_dir_ng: /srv/salt/win/repo-ng
```

`winrepo_cachefile`

Changed in version 2015.8.0: Renamed from `win_repo_mastercachefile` to `winrepo_cachefile`

Note: 2015.8.0 and later minions do not use this setting since the cachefile is now generated by the minion.

Default: `winrepo.p`

Path relative to *`winrepo_dir`* where the winrepo cache should be created.

```
winrepo_cachefile: winrepo.p
```

`winrepo_remotes`

Changed in version 2015.8.0: Renamed from `win_gitrepos` to `winrepo_remotes`.

Default: `['https://github.com/saltstack/salt-winrepo.git']`

List of git repositories to checkout and include in the winrepo for pre-2015.8.0 minions. 2015.8.0 and later minions use *`winrepo_remotes_ng`* instead.

```
winrepo_remotes:
  - https://github.com/saltstack/salt-winrepo.git
```

To specify a specific revision of the repository, prepend a commit ID to the URL of the repository:

```
winrepo_remotes:
  - '<commit_id> https://github.com/saltstack/salt-winrepo.git'
```

Replace `<commit_id>` with the SHA1 hash of a commit ID. Specifying a commit ID is useful in that it allows one to revert back to a previous version in the event that an error is introduced in the latest revision of the repo.

`winrepo_remotes_ng`

New in version 2015.8.0: A new *ng* repo was added.

Default: `['https://github.com/saltstack/salt-winrepo-ng.git']`

List of git repositories to checkout and include in the winrepo for 2015.8.0 and later minions.

```
winrepo_remotes_ng:
  - https://github.com/saltstack/salt-winrepo-ng.git
```

To specify a specific revision of the repository, prepend a commit ID to the URL of the repository:

```
winrepo_remotes_ng:
  - '<commit_id> https://github.com/saltstack/salt-winrepo-ng.git'
```

Replace `<commit_id>` with the SHA1 hash of a commit ID. Specifying a commit ID is useful in that it allows one to revert back to a previous version in the event that an error is introduced in the latest revision of the repo.

`winrepo_branch`

New in version 2015.8.0.

Default: `master`

If the branch is omitted from a winrepo remote, then this branch will be used instead. For example, in the configuration below, the first two remotes would use the `winrepo` branch/tag, while the third would use the `foo` branch/tag.

```
winrepo_branch: winrepo

winrepo_remotes:
  - https://mygitserver/winrepo1.git
  - https://mygitserver/winrepo2.git:
  - foo https://mygitserver/winrepo3.git
```

`winrepo_ssl_verify`

New in version 2015.8.0.

Changed in version 2016.11.0.

Default: `False`

Specifies whether or not to ignore SSL certificate errors when contacting the remote repository. The `False` setting is useful if you're using a git repo that uses a self-signed certificate. However, keep in mind that setting this to anything other `True` is a considered insecure, and using an SSH-based transport (if available) may be a better option.

In the 2016.11.0 release, the default config value changed from `False` to `True`.

```
winrepo_ssl_verify: True
```

Winrepo Authentication Options

These parameters only currently apply to the `pygit2` *`winrepo_provider`*. Authentication works the same as it does in gitfs, as outlined in the *GitFS Walkthrough*, though the global configuration options are named differently to reflect that they are for winrepo instead of gitfs.

`winrepo_user`

New in version 2015.8.0.

Default: `''`

Along with *`winrepo_password`*, is used to authenticate to HTTPS remotes.

```
winrepo_user: git
```

`winrepo_password`

New in version 2015.8.0.

Default: `''`

Along with *`winrepo_user`*, is used to authenticate to HTTPS remotes. This parameter is not required if the repository does not use authentication.

```
winrepo_password: mypassword
```

`winrepo_insecure_auth`

New in version 2015.8.0.

Default: `False`

By default, Salt will not authenticate to an HTTP (non-HTTPS) remote. This parameter enables authentication over HTTP. **Enable this at your own risk.**

```
winrepo_insecure_auth: True
```

`winrepo_pubkey`

New in version 2015.8.0.

Default: `''`

Along with *`winrepo_privkey`* (and optionally *`winrepo_passphrase`*), is used to authenticate to SSH remotes.

```
winrepo_pubkey: /path/to/key.pub
```

`winrepo_privkey`

New in version 2015.8.0.

Default: `''`

Along with *`winrepo_pubkey`* (and optionally *`winrepo_passphrase`*), is used to authenticate to SSH remotes.

```
winrepo_privkey: /path/to/key
```

`winrepo_passphrase`

New in version 2015.8.0.

Default: `''`

This parameter is optional, required only when the SSH key being used to authenticate is protected by a passphrase.

```
winrepo_passphrase: mypassphrase
```

`winrepo_refspecs`

New in version 2017.7.0.

Default: `['+refs/heads/*:refs/remotes/origin/*', '+refs/tags/*:refs/tags/*']`

When fetching from remote repositories, by default Salt will fetch branches and tags. This parameter can be used to override the default and specify alternate refspecs to be fetched. This parameter works similarly to its *GitFS counterpart*, in that it can be configured both globally and for individual remotes.

```
winrepo_refspecs:
  - '+refs/heads/*:refs/remotes/origin/*'
  - '+refs/tags/*:refs/tags/*'
  - '+refs/pull/*/head:refs/remotes/origin/pr/*'
  - '+refs/pull/*/merge:refs/remotes/origin/merge/*'
```

3.1.19 Configure Master on Windows

The master on Windows requires no additional configuration. You can modify the master configuration by creating/editing the master config file located at `c:\salt\conf\master`. The same configuration options available on Linux are available in Windows, as long as they apply. For example, SSH options wouldn't apply in Windows. The main differences are the file paths. If you are familiar with common salt paths, the following table may be useful:

linux Paths		Windows Paths
`/etc/salt`	`<--->`	`c:\salt\conf`
`/`	`<--->`	`c:\salt`

So, for example, the master config file in Linux is `/etc/salt/master`. In Windows the master config file is `c:\salt\conf\master`. The Linux path `/etc/salt` becomes `c:\salt\conf` in Windows.

Common File Locations

Linux Paths	Windows Paths
`conf_file: /etc/salt/master`	`conf_file: c:\salt\conf\master`
`log_file: /var/log/salt/master`	`log_file: c:\salt\var\log\salt\master`
`pidfile: /var/run/salt-master.pid`	`pidfile: c:\salt\var\run\salt-master.pid`

Common Directories

Linux Paths	Windows Paths
`cachedir: /var/cache/salt/master`	`cachedir: c:\salt\var\cache\salt\master`
`extension_modules: /var/cache/salt/master/extmods`	`c:\salt\var\cache\salt\master\extmods`
`pki_dir: /etc/salt/pki/master`	`pki_dir: c:\salt\conf\pki\master`
`root_dir: /`	`root_dir: c:\salt`
`sock_dir: /var/run/salt/master`	`sock_dir: c:\salt\var\run\salt\master`

Roots

file_roots

Linux Paths	Windows Paths
`/srv/salt`	`c:\salt\srv\salt`
`/srv/spm/salt`	`c:\salt\srv\spm\salt`

pillar_roots

Linux Paths	Windows Paths
`/srv/pillar`	`c:\salt\srv\pillar`
`/srv/spm/pillar`	`c:\salt\srv\spm\pillar`

Win Repo Settings

Linux Paths	Windows Paths
`winrepo_dir: /srv/salt/win/repo`	`winrepo_dir: c:\salt\srv\salt\win\repo`
`winrepo_dir_ng: /srv/salt/win/repo-ng`	`winrepo_dir_ng: c:\salt\srv\salt\win\repo-ng`

3.2 Configuring the Salt Minion

The Salt system is amazingly simple and easy to configure. The two components of the Salt system each have a respective configuration file. The **`salt-master`** is configured via the master configuration file, and the **`salt-minion`** is configured via the minion configuration file.

See also:

example minion configuration file

The Salt Minion configuration is very simple. Typically, the only value that needs to be set is the master value so the minion knows where to locate its master.

By default, the salt-minion configuration will be in `/etc/salt/minion`. A notable exception is FreeBSD, where the configuration will be in `/usr/local/etc/salt/minion`.

3.2.1 Minion Primary Configuration

`master`

Default: `salt`

The hostname or IP address of the master. See `ipv6` for IPv6 connections to the master.

Default: `salt`

```
master: salt
```

master:port Syntax

New in version 2015.8.0.

The `master` config option can also be set to use the master's IP in conjunction with a port number by default.

```
master: localhost:1234
```

For IPv6 formatting with a port, remember to add brackets around the IP address before adding the port and enclose the line in single quotes to make it a string:

```
master: '[2001:db8:85a3:8d3:1319:8a2e:370:7348]:1234'
```

Note: If a port is specified in the `master` as well as *`master_port`*, the `master_port` setting will be overridden by the `master` configuration.

List of Masters Syntax

The option can also be set to a list of masters, enabling *multi-master* mode.

```
master:
  - address1
  - address2
```

Changed in version 2014.7.0: The master can be dynamically configured. The *`master`* value can be set to an module function which will be executed and will assume that the returning value is the ip or hostname of the desired master. If a function is being specified, then the *`master_type`* option must be set to `func`, to tell the minion that the value is a function to be run and not a fully-qualified domain name.

```
master: module.function
master_type: func
```

In addition, instead of using multi-master mode, the minion can be configured to use the list of master addresses as a failover list, trying the first address, then the second, etc. until the minion successfully connects. To enable this behavior, set *`master_type`* to `failover`:

```
master:
  - address1
  - address2
master_type: failover
```

`ipv6`

Default: `None`

Whether the master should be connected over IPv6. By default salt minion will try to automatically detect IPv6 connectivity to master.

```
ipv6: True
```

`master_uri_format`

New in version 2015.8.0.

Specify the format in which the master address will be evaluated. Valid options are `default` or `ip_only`. If `ip_only` is specified, then the master address will not be split into IP and PORT, so be sure that only an IP (or domain name) is set in the *`master`* configuration setting.

```
master_uri_format: ip_only
```

`master_tops_first`

New in version 2018.3.0.

Default: `False`

SLS targets defined using the *Master Tops* system are normally executed *after* any matches defined in the *Top File*. Set this option to `True` to have the minion execute the *Master Tops* states first.

```
master_tops_first: True
```

`master_type`

New in version 2014.7.0.

Default: `str`

The type of the *`master`* variable. Can be `str`, `failover`, `func` or `disable`.

```
master_type: failover
```

If this option is set to `failover`, *`master`* must be a list of master addresses. The minion will then try each master in the order specified in the list until it successfully connects. *`master_alive_interval`* must also be set, this determines how often the minion will verify the presence of the master.

```
master_type: func
```

If the master needs to be dynamically assigned by executing a function instead of reading in the static master value, set this to `func`. This can be used to manage the minion's master setting from an execution module. By simply changing the algorithm in the module to return a new master ip/fqdn, restart the minion and it will connect to the new master.

As of version 2016.11.0 this option can be set to `disable` and the minion will never attempt to talk to the master. This is useful for running a masterless minion daemon.

```
master_type: disable
```

`max_event_size`

New in version 2014.7.0.

Default: `1048576`

Passing very large events can cause the minion to consume large amounts of memory. This value tunes the maximum size of a message allowed onto the minion event bus. The value is expressed in bytes.

```
max_event_size: 1048576
```

`enable_legacy_startup_events`

New in version 2019.2.0.

Default: `True`

When a minion starts up it sends a notification on the event bus with a tag that looks like this: `salt/minion/<minion_id>/start`. For historical reasons the minion also sends a similar event with an event tag like this: `minion_start`. This duplication can cause a lot of clutter on the event bus when there are many minions. Set `enable_legacy_startup_events: False` in the minion config to ensure only the `salt/minion/<minion_id>/start` events are sent. Beginning with the `Sodium` Salt release this option will default to `False`.

```
enable_legacy_startup_events: True
```

`master_failback`

New in version 2016.3.0.

Default: `False`

If the minion is in multi-master mode and the :conf_minion'master_type' configuration option is set to `failover`, this setting can be set to `True` to force the minion to fail back to the first master in the list if the first master is back online.

```
master_failback: False
```

`master_failback_interval`

New in version 2016.3.0.

Default: `0`

If the minion is in multi-master mode, the :conf_minion'master_type' configuration is set to `failover`, and the `master_failback` option is enabled, the master failback interval can be set to ping the top master with this interval, in seconds.

```
master_failback_interval: 0
```

`master_alive_interval`

Default: `0`

Configures how often, in seconds, the minion will verify that the current master is alive and responding. The minion will try to establish a connection to the next master in the list if it finds the existing one is dead.

```
master_alive_interval: 30
```

`master_shuffle`

New in version 2014.7.0.

Deprecated since version 2019.2.0.

Default: `False`

Warning: This option has been deprecated in Salt `2019.2.0`. Please use *`random_master`* instead.

```
master_shuffle: True
```

`random_master`

New in version 2014.7.0.

Changed in version 2019.2.0: The *`master_failback`* option can be used in conjunction with `random_master` to force the minion to fail back to the first master in the list if the first master is back online. Note that *`master_type`* must be set to `failover` in order for the `master_failback` setting to work.

Default: `False`

If *`master`* is a list of addresses, shuffle them before trying to connect to distribute the minions over all available masters. This uses Python's `random.shuffle` method.

If multiple masters are specified in the 'master' setting as a list, the default behavior is to always try to connect to them in the order they are listed. If `random_master` is set to True, the order will be randomized instead upon Minion startup. This can be helpful in distributing the load of many minions executing `salt-call` requests, for example, from a cron job. If only one master is listed, this setting is ignored and a warning is logged.

```
random_master: True
```

Note: When the `failover`, `master_failback`, and `random_master` options are used together, only the "secondary masters" will be shuffled. The first master in the list is ignored in the `random.shuffle` call. See *`master_failback`* for more information.

`retry_dns`

Default: `30`

Set the number of seconds to wait before attempting to resolve the master hostname if name resolution fails. Defaults to 30 seconds. Set to zero if the minion should shutdown and not retry.

```
retry_dns: 30
```

`retry_dns_count`

New in version 2018.3.4.

Default: `None`

Set the number of attempts to perform when resolving the master hostname if name resolution fails. By default the minion will retry indefinitely.

```
retry_dns_count: 3
```

`master_port`

Default: `4506`

The port of the master ret server, this needs to coincide with the ret_port option on the Salt master.

```
master_port: 4506
```

`publish_port`

Default: `4505`

The port of the master publish server, this needs to coincide with the publish_port option on the Salt master.

```
publish_port: 4505
```

`source_interface_name`

New in version 2018.3.0.

The name of the interface to use when establishing the connection to the Master.

Note: If multiple IP addresses are configured on the named interface, the first one will be selected. In that case, for a better selection, consider using the *`source_address`* option.

Note: To use an IPv6 address from the named interface, make sure the option *`ipv6`* is enabled, i.e., `ipv6: true`.

Note: If the interface is down, it will avoid using it, and the Minion will bind to `0.0.0.0` (all interfaces).

Warning: This option requires modern version of the underlying libraries used by the selected transport:

- `zeromq` requires `pyzmq` >= 16.0.1 and `libzmq` >= 4.1.6
- `tcp` requires `tornado` >= 4.5

Configuration example:

```
source_interface_name: bond0.1234
```

`source_address`

New in version 2018.3.0.

The source IP address or the domain name to be used when connecting the Minion to the Master. See *`ipv6`* for IPv6 connections to the Master.

Warning: This option requires modern version of the underlying libraries used by the selected transport:

- `zeromq` requires `pyzmq` >= 16.0.1 and `libzmq` >= 4.1.6
- `tcp` requires `tornado` >= 4.5

Configuration example:

```
source_address: if-bond0-1234.sjc.us-west.internal
```

`source_ret_port`

New in version 2018.3.0.

The source port to be used when connecting the Minion to the Master ret server.

Warning: This option requires modern version of the underlying libraries used by the selected transport:

- `zeromq` requires `pyzmq` >= 16.0.1 and `libzmq` >= 4.1.6
- `tcp` requires `tornado` >= 4.5

Configuration example:

```
source_ret_port: 49017
```

`source_publish_port`

New in version 2018.3.0.

The source port to be used when connecting the Minion to the Master publish server.

Warning: This option requires modern version of the underlying libraries used by the selected transport:

- `zeromq` requires `pyzmq` >= 16.0.1 and `libzmq` >= 4.1.6
- `tcp` requires `tornado` >= 4.5

Configuration example:

```
source_publish_port: 49018
```

`user`

Default: `root`

The user to run the Salt processes

```
user: root
```

`sudo_user`

Default: `''`

The user to run salt remote execution commands as via sudo. If this option is enabled then sudo will be used to change the active user executing the remote command. If enabled the user will need to be allowed access via the sudoers file for the user that the salt minion is configured to run as. The most common option would be to use the root user. If this option is set the `user` option should also be set to a non-root user. If migrating from a root minion to a non root minion the minion cache should be cleared and the minion pki directory will need to be changed to the ownership of the new user.

```
sudo_user: root
```

`pidfile`

Default: `/var/run/salt-minion.pid`

The location of the daemon's process ID file

```
pidfile: /var/run/salt-minion.pid
```

`root_dir`

Default: `/`

This directory is prepended to the following options: *`pki_dir`*, *`cachedir`*, *`log_file`*, *`sock_dir`*, and `pidfile`.

```
root_dir: /
```

`conf_file`

Default: `/etc/salt/minion`

The path to the minion's configuration file.

```
conf_file: /etc/salt/minion
```

`pki_dir`

Default: `/etc/salt/pki/minion`

The directory used to store the minion's public and private keys.

```
pki_dir: /etc/salt/pki/minion
```

`id`

Default: the system's hostname

See also:

Salt Walkthrough

The **Setting up a Salt Minion** section contains detailed information on how the hostname is determined.

Explicitly declare the id for this minion to use. Since Salt uses detached ids it is possible to run multiple minions on the same machine but with different ids.

```
id: foo.bar.com
```

`minion_id_caching`

New in version 0.17.2.

Default: `True`

Caches the minion id to a file when the minion's *id* is not statically defined in the minion config. This setting prevents potential problems when automatic minion id resolution changes, which can cause the minion to lose connection with the master. To turn off minion id caching, set this config to `False`.

For more information, please see Issue #7558 and Pull Request #8488.

```
minion_id_caching: True
```

`append_domain`

Default: `None`

Append a domain to a hostname in the event that it does not exist. This is useful for systems where `socket.getfqdn()` does not actually result in a FQDN (for instance, Solaris).

```
append_domain: foo.org
```

`minion_id_remove_domain`

New in version 3000.

Default: `False`

Remove a domain when the minion id is generated as a fully qualified domain name (either by the user provided `id_function`, or by Salt). This is useful when the minions shall be named like hostnames. Can be a single domain (to prevent name clashes), or True, to remove all domains.

Examples:

- minion_id_remove_domain = foo.org - FQDN = king_bob.foo.org –> minion_id = king_bob - FQDN = king_bob.bar.org –> minion_id = king_bob.bar.org
- minion_id_remove_domain = True - FQDN = king_bob.foo.org –> minion_id = king_bob - FQDN = king_bob.bar.org –> minion_id = king_bob

For more information, please see issue #49212 and PR #49378.

```
minion_id_remove_domain: foo.org
```

`minion_id_lowercase`

Default: `False`

Convert minion id to lowercase when it is being generated. Helpful when some hosts get the minion id in uppercase. Cached ids will remain the same and not converted.

```
minion_id_lowercase: True
```

`cachedir`

Default: `/var/cache/salt/minion`

The location for minion cache data.

This directory may contain sensitive data and should be protected accordingly.

```
cachedir: /var/cache/salt/minion
```

`color_theme`

Default: `""`

Specifies a path to the color theme to use for colored command line output.

```
color_theme: /etc/salt/color_theme
```

`append_minionid_config_dirs`

Default: `[]` (the empty list) for regular minions, `['cachedir']` for proxy minions.

Append minion_id to these configuration directories. Helps with multiple proxies and minions running on the same machine. Allowed elements in the list: `pki_dir`, `cachedir`, `extension_modules`. Normally not needed unless running several proxies and/or minions on the same machine.

```
append_minionid_config_dirs:
  - pki_dir
  - cachedir
```

`verify_env`

Default: `True`

Verify and set permissions on configuration directories at startup.

```
verify_env: True
```

Note: When set to `True` the verify_env option requires WRITE access to the configuration directory (/etc/salt/). In certain situations such as mounting /etc/salt/ as read-only for templating this will create a stack trace when *`state.apply`* is called.

---

cache_jobs

Default: `False`

The minion can locally cache the return data from jobs sent to it, this can be a good way to keep track of the minion side of the jobs the minion has executed. By default this feature is disabled, to enable set cache_jobs to `True`.

```
cache_jobs: False
```

grains

Default: (empty)

See also:

Using grains in a state

Statically assigns grains to the minion.

```
grains:
  roles:
    - webserver
    - memcache
  deployment: datacenter4
  cabinet: 13
  cab_u: 14-15
```

grains_blacklist

Default: `[]`

Each grains key will be compared against each of the expressions in this list. Any keys which match will be filtered from the grains. Exact matches, glob matches, and regular expressions are supported.

Note: Some states and execution modules depend on grains. Filtering may cause them to be unavailable or run unreliably.

New in version 3000.

```
grains_blacklist:
  - cpu_flags
  - zmq*
  - ipv[46]
```

grains_cache

Default: `False`

The minion can locally cache grain data instead of refreshing the data each time the grain is referenced. By default this feature is disabled, to enable set grains_cache to `True`.

```
grains_cache: False
```

grains_deep_merge

New in version 2016.3.0.

Default: `False`

The grains can be merged, instead of overridden, using this option. This allows custom grains to defined different subvalues of a dictionary grain. By default this feature is disabled, to enable set grains_deep_merge to `True`.

```
grains_deep_merge: False
```

For example, with these custom grains functions:

```
def custom1_k1():
    return {'custom1': {'k1': 'v1'}}

def custom1_k2():
    return {'custom1': {'k2': 'v2'}}
```

Without `grains_deep_merge`, the result would be:

```
custom1:
  k1: v1
```

With `grains_deep_merge`, the result will be:

```
custom1:
  k1: v1
  k2: v2
```

grains_refresh_every

Default: `0`

The `grains_refresh_every` setting allows for a minion to periodically check its grains to see if they have changed and, if so, to inform the master of the new grains. This operation is moderately expensive, therefore care should be taken not to set this value too low.

Note: This value is expressed in minutes.

A value of 10 minutes is a reasonable default.

```
grains_refresh_every: 0
```

metadata_server_grains

New in version 2017.7.0.

Default: `False`

Set this option to enable gathering of cloud metadata from `http://169.254.169.254/latest` for use in grains (see *here* for more information).

```
metadata_server_grains: True
```

`fibre_channel_grains`

Default: `False`

The `fibre_channel_grains` setting will enable the `fc_wwn` grain for Fibre Channel WWN's on the minion. Since this grain is expensive, it is disabled by default.

```
fibre_channel_grains: True
```

`iscsi_grains`

Default: `False`

The `iscsi_grains` setting will enable the `iscsi_iqn` grain on the minion. Since this grain is expensive, it is disabled by default.

```
iscsi_grains: True
```

`nvme_grains`

Default: `False`

The `nvme_grains` setting will enable the `nvme_nqn` grain on the minion. Since this grain is expensive, it is disabled by default.

```
nvme_grains: True
```

`mine_enabled`

New in version 2015.8.10.

Default: `True`

Determines whether or not the salt minion should run scheduled mine updates. If this is set to False then the mine update function will not get added to the scheduler for the minion.

```
mine_enabled: True
```

`mine_return_job`

New in version 2015.8.10.

Default: `False`

Determines whether or not scheduled mine updates should be accompanied by a job return for the job cache.

```
mine_return_job: False
```

`mine_functions`

Default: Empty

Designate which functions should be executed at mine_interval intervals on each minion. *See this documentation on the Salt Mine* for more information. Note these can be defined in the pillar for a minion as well.

example minion configuration file

```
mine_functions:
  test.ping: []
  network.ip_addrs:
    interface: eth0
    cidr: '10.0.0.0/8'
```

`mine_interval`

Default: `60`

The number of minutes between mine updates.

```
mine_interval: 60
```

`sock_dir`

Default: `/var/run/salt/minion`

The directory where Unix sockets will be kept.

```
sock_dir: /var/run/salt/minion
```

`enable_gpu_grains`

Default: `True`

Enable GPU hardware data for your master. Be aware that the minion can take a while to start up when lspci and/or dmidecode is used to populate the grains for the minion, so this can be set to `False` if you do not need these grains.

```
enable_gpu_grains: False
```

`outputter_dirs`

Default: `[]`

A list of additional directories to search for salt outputters in.

```
outputter_dirs: []
```

`backup_mode`

Default: `''`

Make backups of files replaced by `file.managed` and `file.recurse` state modules under *`cachedir`* in `file_backup` subdirectory preserving original paths. Refer to *File State Backups documentation* for more details.

```
backup_mode: minion
```

`acceptance_wait_time`

Default: `10`

The number of seconds to wait until attempting to re-authenticate with the master.

```
acceptance_wait_time: 10
```

`acceptance_wait_time_max`

Default: `0`

The maximum number of seconds to wait until attempting to re-authenticate with the master. If set, the wait will increase by *`acceptance_wait_time`* seconds each iteration.

```
acceptance_wait_time_max: 0
```

`rejected_retry`

Default: `False`

If the master rejects the minion's public key, retry instead of exiting. Rejected keys will be handled the same as waiting on acceptance.

```
rejected_retry: False
```

`random_reauth_delay`

Default: `10`

When the master key changes, the minion will try to re-auth itself to receive the new master key. In larger environments this can cause a syn-flood on the master because all minions try to re-auth immediately. To prevent this and have a minion wait for a random amount of time, use this optional parameter. The wait-time will be a random number of seconds between 0 and the defined value.

```
random_reauth_delay: 60
```

`master_tries`

New in version 2016.3.0.

Default: `1`

The number of attempts to connect to a master before giving up. Set this to `-1` for unlimited attempts. This allows for a master to have downtime and the minion to reconnect to it later when it comes back up. In 'failover' mode, which is set in the *`master_type`* configuration, this value is the number of attempts for each set of masters. In this mode, it will cycle through the list of masters for each attempt.

`master_tries` is different than *`auth_tries`* because `auth_tries` attempts to retry auth attempts with a single master. `auth_tries` is under the assumption that you can connect to the master but not gain authorization from it. `master_tries` will still cycle through all of the masters in a given try, so it is appropriate if you expect occasional downtime from the master(s).

```
master_tries: 1
```

`auth_tries`

New in version 2014.7.0.

Default: `7`

The number of attempts to authenticate to a master before giving up. Or, more technically, the number of consecutive SaltReqTimeoutErrors that are acceptable when trying to authenticate to the master.

```
auth_tries: 7
```

`auth_timeout`

New in version 2014.7.0.

Default: `60`

When waiting for a master to accept the minion's public key, salt will continuously attempt to reconnect until successful. This is the timeout value, in seconds, for each individual attempt. After this timeout expires, the minion will wait for *`acceptance_wait_time`* seconds before trying again. Unless your master is under unusually heavy load, this should be left at the default.

```
auth_timeout: 60
```

`auth_safemode`

New in version 2014.7.0.

Default: `False`

If authentication fails due to SaltReqTimeoutError during a ping_interval, this setting, when set to `True`, will cause a sub-minion process to restart.

```
auth_safemode: False
```

`ping_interval`

Default: `0`

Instructs the minion to ping its master(s) every n number of minutes. Used primarily as a mitigation technique against minion disconnects.

```
ping_interval: 0
```

`random_startup_delay`

Default: `0`

The maximum bound for an interval in which a minion will randomly sleep upon starting up prior to attempting to connect to a master. This can be used to splay connection attempts for cases where many minions starting up at once may place undue load on a master.

For example, setting this to `5` will tell a minion to sleep for a value between `0` and `5` seconds.

```
random_startup_delay: 5
```

`recon_default`

Default: `1000`

The interval in milliseconds that the socket should wait before trying to reconnect to the master (1000ms = 1 second).

```
recon_default: 1000
```

`recon_max`

Default: `10000`

The maximum time a socket should wait. Each interval the time to wait is calculated by doubling the previous time. If recon_max is reached, it starts again at the recon_default.

Short example:

- reconnect 1: the socket will wait 'recon_default' milliseconds
- reconnect 2: 'recon_default' * 2
- reconnect 3: ('recon_default' * 2) * 2
- reconnect 4: value from previous interval * 2
- reconnect 5: value from previous interval * 2
- reconnect x: if value >= recon_max, it starts again with recon_default

```
recon_max: 10000
```

`recon_randomize`

Default: `True`

Generate a random wait time on minion start. The wait time will be a random value between recon_default and recon_default + recon_max. Having all minions reconnect with the same recon_default and recon_max value kind of defeats the purpose of being able to change these settings. If all minions have the same values and the setup is quite large (several thousand minions), they will still flood the master. The desired behavior is to have time-frame within all minions try to reconnect.

```
recon_randomize: True
```

`loop_interval`

Default: `1`

The loop_interval sets how long in seconds the minion will wait between evaluating the scheduler and running cleanup tasks. This defaults to 1 second on the minion scheduler.

```
loop_interval: 1
```

`pub_ret`

Default: True

Some installations choose to start all job returns in a cache or a returner and forgo sending the results back to a master. In this workflow, jobs are most often executed with –async from the Salt CLI and then results are evaluated by examining job caches on the minions or any configured returners. WARNING: Setting this to False will **disable** returns back to the master.

```
pub_ret: True
```

`return_retry_timer`

Default: `5`

The default timeout for a minion return attempt.

```
return_retry_timer: 5
```

`return_retry_timer_max`

Default: `10`

The maximum timeout for a minion return attempt. If non-zero the minion return retry timeout will be a random int between `return_retry_timer` and `return_retry_timer_max`

```
return_retry_timer_max: 10
```

`cache_sreqs`

Default: `True`

The connection to the master ret_port is kept open. When set to False, the minion creates a new connection for every return to the master.

```
cache_sreqs: True
```

`ipc_mode`

Default: `ipc`

Windows platforms lack POSIX IPC and must rely on slower TCP based inter- process communications. `ipc_mode` is set to `tcp` on such systems.

```
ipc_mode: ipc
```

`tcp_pub_port`

Default: `4510`

Publish port used when *`ipc_mode`* is set to `tcp`.

```
tcp_pub_port: 4510
```

`tcp_pull_port`

Default: `4511`

Pull port used when *ipc_mode* is set to `tcp`.

```
tcp_pull_port: 4511
```

`transport`

Default: `zeromq`

Changes the underlying transport layer. ZeroMQ is the recommended transport while additional transport layers are under development. Supported values are `zeromq` and `tcp` (experimental). This setting has a significant impact on performance and should not be changed unless you know what you are doing!

```
transport: zeromq
```

`syndic_finger`

Default: `''`

The key fingerprint of the higher-level master for the syndic to verify it is talking to the intended master.

```
syndic_finger: 'ab:30:65:2a:d6:9e:20:4f:d8:b2:f3:a7:d4:65:50:10'
```

`http_connect_timeout`

New in version 2019.2.0.

Default: `20`

HTTP connection timeout in seconds. Applied when fetching files using tornado back-end. Should be greater than overall download time.

```
http_connect_timeout: 20
```

`http_request_timeout`

New in version 2015.8.0.

Default: `3600`

HTTP request timeout in seconds. Applied when fetching files using tornado back-end. Should be greater than overall download time.

```
http_request_timeout: 3600
```

`proxy_host`

Default: `''`

The hostname used for HTTP proxy access.

```
proxy_host: proxy.my-domain
```

`proxy_port`

Default: `0`

The port number used for HTTP proxy access.

```
proxy_port: 31337
```

`proxy_username`

Default: `''`

The username used for HTTP proxy access.

```
proxy_username: charon
```

`proxy_password`

Default: `''`

The password used for HTTP proxy access.

```
proxy_password: obolus
```

`no_proxy`

New in version 2019.2.0.

Default: `[]`

List of hosts to bypass HTTP proxy

Note: This key does nothing unless proxy_host etc is configured, it does not support any kind of wildcards.

```
no_proxy: [ '127.0.0.1', 'foo.tld' ]
```

`use_yamlloader_old`

New in version 2019.2.1.

Default: `False`

Use the pre-2019.2 YAML renderer. Uses legacy YAML rendering to support some legacy inline data structures. See the *2019.2.1 release notes* for more details.

```
use_yamlloader_old: False
```

3.2.2 Docker Configuration

`docker.update_mine`

New in version 2017.7.8,2018.3.3.

Changed in version 2019.2.0: The default value is now `False`

Default: `True`

If enabled, when containers are added, removed, stopped, started, etc., the *mine* will be updated with the results of `docker.ps verbose=True all=True host=True`. This mine data is used by *`mine.get_docker`*. Set this option to `False` to keep Salt from updating the mine with this information.

Note: This option can also be set in Grains or Pillar data, with Grains overriding Pillar and the minion config file overriding Grains.

Note: Disabling this will of course keep *`mine.get_docker`* from returning any information for a given minion.

```
docker.update_mine: False
```

`docker.compare_container_networks`

New in version 2018.3.0.

Default: `{'static': ['Aliases', 'Links', 'IPAMConfig'], 'automatic': ['IPAddress', 'Gateway', 'GlobalIPv6Address', 'IPv6Gateway']}`

Specifies which keys are examined by *`docker.compare_container_networks`*.

Note: This should not need to be modified unless new features added to Docker result in new keys added to the network configuration which must be compared to determine if two containers have different network configs. This config option exists solely as a way to allow users to continue using Salt to manage their containers after an API change, without waiting for a new Salt release to catch up to the changes in the Docker API.

```
docker.compare_container_networks:
  static:
    - Aliases
    - Links
    - IPAMConfig
  automatic:
    - IPAddress
    - Gateway
    - GlobalIPv6Address
    - IPv6Gateway
```

`optimization_order`

Default: `[0, 1, 2]`

In cases where Salt is distributed without .py files, this option determines the priority of optimization level(s) Salt's module loader should prefer.

Note: This option is only supported on Python 3.5+.

```
optimization_order:
  - 2
  - 0
  - 1
```

3.2.3 Minion Execution Module Management

`disable_modules`

Default: `[]` (all execution modules are enabled by default)

The event may occur in which the administrator desires that a minion should not be able to execute a certain module.

However, the `sys` module is built into the minion and cannot be disabled.

This setting can also tune the minion. Because all modules are loaded into system memory, disabling modules will lower the minion's memory footprint.

Modules should be specified according to their file name on the system and not by their virtual name. For example, to disable `cmd`, use the string `cmdmod` which corresponds to `salt.modules.cmdmod`.

```
disable_modules:
  - test
  - solr
```

`disable_returners`

Default: `[]` (all returners are enabled by default)

If certain returners should be disabled, this is the place

```
disable_returners:
  - mongo_return
```

`whitelist_modules`

Default: `[]` (Module whitelisting is disabled. Adding anything to the config option will cause only the listed modules to be enabled. Modules not in the list will not be loaded.)

This option is the reverse of disable_modules. If enabled, only execution modules in this list will be loaded and executed on the minion.

Note that this is a very large hammer and it can be quite difficult to keep the minion working the way you think it should since Salt uses many modules internally itself. At a bare minimum you need the following enabled or else the minion won't start.

```
whitelist_modules:
  - cmdmod
  - test
  - config
```

module_dirs

Default: `[]`

A list of extra directories to search for Salt modules

```
module_dirs:
  - /var/lib/salt/modules
```

returner_dirs

Default: `[]`

A list of extra directories to search for Salt returners

```
returner_dirs:
  - /var/lib/salt/returners
```

states_dirs

Default: `[]`

A list of extra directories to search for Salt states

```
states_dirs:
  - /var/lib/salt/states
```

grains_dirs

Default: `[]`

A list of extra directories to search for Salt grains

```
grains_dirs:
  - /var/lib/salt/grains
```

render_dirs

Default: `[]`

A list of extra directories to search for Salt renderers

```
render_dirs:
  - /var/lib/salt/renderers
```

`utils_dirs`

Default: `[]`

A list of extra directories to search for Salt utilities

```
utils_dirs:
  - /var/lib/salt/utils
```

`cython_enable`

Default: `False`

Set this value to true to enable auto-loading and compiling of `.pyx` modules, This setting requires that `gcc` and `cython` are installed on the minion.

```
cython_enable: False
```

`enable_zip_modules`

New in version 2015.8.0.

Default: `False`

Set this value to true to enable loading of zip archives as extension modules. This allows for packing module code with specific dependencies to avoid conflicts and/or having to install specific modules' dependencies in system libraries.

```
enable_zip_modules: False
```

`providers`

Default: (empty)

A module provider can be statically overwritten or extended for the minion via the `providers` option. This can be done *on an individual basis in an SLS file*, or globally here in the minion config, like below.

```
providers:
  service: systemd
```

`modules_max_memory`

Default: `-1`

Specify a max size (in bytes) for modules on import. This feature is currently only supported on *NIX operating systems and requires psutil.

```
modules_max_memory: -1
```

`extmod_whitelist/extmod_blacklist`

New in version 2017.7.0.

By using this dictionary, the modules that are synced to the minion's extmod cache using *saltutil.sync_** can be limited. If nothing is set to a specific type, then all modules are accepted. To block all modules of a specific type, whitelist an empty list.

```
extmod_whitelist:
  modules:
    - custom_module
  engines:
    - custom_engine
  pillars: []

extmod_blacklist:
  modules:
    - specific_module
```

Valid options:

- beacons
- clouds
- sdb
- modules
- states
- grains
- renderers
- returners
- proxy
- engines
- output
- utils
- pillar

3.2.4 Top File Settings

These parameters only have an effect if running a masterless minion.

`state_top`

Default: `top.sls`

The state system uses a "top" file to tell the minions what environment to use and what modules to use. The state_top file is defined relative to the root of the base environment.

```
state_top: top.sls
```

state_top_saltenv

This option has no default value. Set it to an environment name to ensure that *only* the top file from that environment is considered during a *highstate*.

Note: Using this value does not change the merging strategy. For instance, if *top_file_merging_strategy* is set to `merge`, and *state_top_saltenv* is set to `foo`, then any sections for environments other than `foo` in the top file for the `foo` environment will be ignored. With *state_top_saltenv* set to `base`, all states from all environments in the `base` top file will be applied, while all other top files are ignored. The only way to set *state_top_saltenv* to something other than `base` and not have the other environments in the targeted top file ignored, would be to set *top_file_merging_strategy* to `merge_all`.

```
state_top_saltenv: dev
```

top_file_merging_strategy

Changed in version 2016.11.0: A `merge_all` strategy has been added.

Default: `merge`

When no specific fileserver environment (a.k.a. `saltenv`) has been specified for a *highstate*, all environments' top files are inspected. This config option determines how the SLS targets in those top files are handled.

When set to `merge`, the `base` environment's top file is evaluated first, followed by the other environments' top files. The first target expression (e.g. `'*'`) for a given environment is kept, and when the same target expression is used in a different top file evaluated later, it is ignored. Because `base` is evaluated first, it is authoritative. For example, if there is a target for `'*'` for the `foo` environment in both the `base` and `foo` environment's top files, the one in the `foo` environment would be ignored. The environments will be evaluated in no specific order (aside from `base` coming first). For greater control over the order in which the environments are evaluated, use *env_order*. Note that, aside from the `base` environment's top file, any sections in top files that do not match that top file's environment will be ignored. So, for example, a section for the `qa` environment would be ignored if it appears in the `dev` environment's top file. To keep use cases like this from being ignored, use the `merge_all` strategy.

When set to `same`, then for each environment, only that environment's top file is processed, with the others being ignored. For example, only the `dev` environment's top file will be processed for the `dev` environment, and any SLS targets defined for `dev` in the `base` environment's (or any other environment's) top file will be ignored. If an environment does not have a top file, then the top file from the *default_top* config parameter will be used as a fallback.

When set to `merge_all`, then all states in all environments in all top files will be applied. The order in which individual SLS files will be executed will depend on the order in which the top files were evaluated, and the environments will be evaluated in no specific order. For greater control over the order in which the environments are evaluated, use *env_order*.

```
top_file_merging_strategy: same
```

env_order

Default: `[]`

When *top_file_merging_strategy* is set to `merge`, and no environment is specified for a *highstate*, this config option allows for the order in which top files are evaluated to be explicitly defined.

```
env_order:
  - base
  - dev
  - qa
```

`default_top`

Default: `base`

When *`top_file_merging_strategy`* is set to `same`, and no environment is specified for a *highstate* (i.e. *`environment`* is not set for the minion), this config option specifies a fallback environment in which to look for a top file if an environment lacks one.

```
default_top: dev
```

`startup_states`

Default: `''`

States to run when the minion daemon starts. To enable, set `startup_states` to:

- `highstate`: Execute state.highstate
- `sls`: Read in the sls_list option and execute the named sls files
- `top`: Read top_file option and execute based on that file on the Master

```
startup_states: ''
```

`sls_list`

Default: `[]`

List of states to run when the minion starts up if `startup_states` is set to `sls`.

```
sls_list:
  - edit.vim
  - hyper
```

`start_event_grains`

Default: `[]`

List of grains to pass in start event when minion starts up.

```
start_event_grains:
  - machine_id
  - uuid
```

`top_file`

Default: `''`

Top file to execute if `startup_states` is set to `top`.

```
top_file: ''
```

3.2.5 State Management Settings

`renderer`

Default: `jinja|yaml`

The default renderer used for local state executions

```
renderer: jinja|json
```

`test`

Default: `False`

Set all state calls to only test if they are going to actually make changes or just post what changes are going to be made.

```
test: False
```

`state_verbose`

Default: `True`

Controls the verbosity of state runs. By default, the results of all states are returned, but setting this value to `False` will cause salt to only display output for states that failed or states that have changes.

```
state_verbose: True
```

`state_output`

Default: `full`

The state_output setting controls which results will be output full multi line:

- `full`, `terse` - each state will be full/terse
- `mixed` - only states with errors will be full
- `changes` - states with changes and errors will be full

`full_id`, `mixed_id`, `changes_id` and `terse_id` are also allowed; when set, the state ID will be used as name in the output.

```
state_output: full
```

`state_output_diff`

Default: `False`

The state_output_diff setting changes whether or not the output from successful states is returned. Useful when even the terse output of these states is cluttering the logs. Set it to True to ignore them.

```
state_output_diff: False
```

`autoload_dynamic_modules`

Default: `True`

autoload_dynamic_modules turns on automatic loading of modules found in the environments on the master. This is turned on by default. To turn off auto-loading modules when states run, set this value to `False`.

```
autoload_dynamic_modules: True
```

Default: `True`

clean_dynamic_modules keeps the dynamic modules on the minion in sync with the dynamic modules on the master. This means that if a dynamic module is not on the master it will be deleted from the minion. By default this is enabled and can be disabled by changing this value to `False`.

```
clean_dynamic_modules: True
```

Note: If `extmod_whitelist` is specified, modules which are not whitelisted will also be cleaned here.

`saltenv`

Changed in version 2018.3.0: Renamed from `environment` to `saltenv`. If `environment` is used, `saltenv` will take its value. If both are used, `environment` will be ignored and `saltenv` will be used.

Normally the minion is not isolated to any single environment on the master when running states, but the environment can be isolated on the minion side by statically setting it. Remember that the recommended way to manage environments is to isolate via the top file.

```
saltenv: dev
```

`lock_saltenv`

New in version 2018.3.0.

Default: `False`

For purposes of running states, this option prevents using the `saltenv` argument to manually set the environment. This is useful to keep a minion which has the *`saltenv`* option set to `dev` from running states from an environment other than `dev`.

```
lock_saltenv: True
```

`snapper_states`

Default: False

The *snapper_states* value is used to enable taking snapper snapshots before and after salt state runs. This allows for state runs to be rolled back.

For snapper states to function properly snapper needs to be installed and enabled.

```
snapper_states: True
```

`snapper_states_config`

Default: `root`

Snapper can execute based on a snapper configuration. The configuration needs to be set up before snapper can use it. The default configuration is `root`, this default makes snapper run on SUSE systems using the default configuration set up at install time.

```
snapper_states_config: root
```

3.2.6 File Directory Settings

`file_client`

Default: `remote`

The client defaults to looking on the master server for files, but can be directed to look on the minion by setting this parameter to `local`.

```
file_client: remote
```

`use_master_when_local`

Default: `False`

When using a local *`file_client`*, this parameter is used to allow the client to connect to a master for remote execution.

```
use_master_when_local: False
```

`file_roots`

Default:

```
base:
  - /srv/salt
```

When using a local *`file_client`*, this parameter is used to setup the fileserver's environments. This parameter operates identically to the *`master config parameter`* of the same name.

```
file_roots:
  base:
    - /srv/salt
  dev:
    - /srv/salt/dev/services
    - /srv/salt/dev/states
  prod:
    - /srv/salt/prod/services
    - /srv/salt/prod/states
```

`fileserver_followsymlinks`

New in version 2014.1.0.

Default: `True`

By default, the file_server follows symlinks when walking the filesystem tree. Currently this only applies to the default roots fileserver_backend.

```
fileserver_followsymlinks: True
```

`fileserver_ignoresymlinks`

New in version 2014.1.0.

Default: `False`

If you do not want symlinks to be treated as the files they are pointing to, set `fileserver_ignoresymlinks` to `True`. By default this is set to False. When set to `True`, any detected symlink while listing files on the Master will not be returned to the Minion.

```
fileserver_ignoresymlinks: False
```

`fileserver_limit_traversal`

New in version 2014.1.0.

Default: `False`

By default, the Salt fileserver recurses fully into all defined environments to attempt to find files. To limit this behavior so that the fileserver only traverses directories with SLS files and special Salt directories like _modules, set `fileserver_limit_traversal` to `True`. This might be useful for installations where a file root has a very large number of files and performance is impacted.

```
fileserver_limit_traversal: False
```

`hash_type`

Default: `sha256`

The hash_type is the hash to use when discovering the hash of a file on the local fileserver. The default is sha256, but md5, sha1, sha224, sha384, and sha512 are also supported.

```
hash_type: sha256
```

3.2.7 Pillar Configuration

pillar_roots

Default:

```
base:
  - /srv/pillar
```

When using a local `file_client`, this parameter is used to setup the pillar environments.

```
pillar_roots:
  base:
    - /srv/pillar
  dev:
    - /srv/pillar/dev
  prod:
    - /srv/pillar/prod
```

on_demand_ext_pillar

New in version 2016.3.6,2016.11.3,2017.7.0.

Default: `['libvirt', 'virtkey']`

When using a local `file_client`, this option controls which external pillars are permitted to be used on-demand using `pillar.ext`.

```
on_demand_ext_pillar:
  - libvirt
  - virtkey
  - git
```

Warning: This will allow a masterless minion to request specific pillar data via `pillar.ext`, and may be considered a security risk. However, pillar data generated in this way will not affect the *in-memory pillar data*, so this risk is limited to instances in which states/modules/etc. (built-in or custom) rely upon pillar data generated by `pillar.ext`.

decrypt_pillar

New in version 2017.7.0.

Default: `[]`

A list of paths to be recursively decrypted during pillar compilation.

```
decrypt_pillar:
  - 'foo:bar': gpg
  - 'lorem:ipsum:dolor'
```

Entries in this list can be formatted either as a simple string, or as a key/value pair, with the key being the pillar location, and the value being the renderer to use for pillar decryption. If the former is used, the renderer specified by *decrypt_pillar_default* will be used.

decrypt_pillar_delimiter

New in version 2017.7.0.

Default: `:`

The delimiter used to distinguish nested data structures in the *decrypt_pillar* option.

```
decrypt_pillar_delimiter: '|'
decrypt_pillar:
  - 'foo|bar': gpg
  - 'lorem|ipsum|dolor'
```

decrypt_pillar_default

New in version 2017.7.0.

Default: `gpg`

The default renderer used for decryption, if one is not specified for a given pillar key in *decrypt_pillar*.

```
decrypt_pillar_default: my_custom_renderer
```

decrypt_pillar_renderers

New in version 2017.7.0.

Default: `['gpg']`

List of renderers which are permitted to be used for pillar decryption.

```
decrypt_pillar_renderers:
  - gpg
  - my_custom_renderer
```

pillarenv

Default: `None`

Isolates the pillar environment on the minion side. This functions the same as the environment setting, but for pillar instead of states.

```
pillarenv: dev
```

pillarenv_from_saltenv

New in version 2017.7.0.

Default: `False`

When set to `True`, the *`pillarenv`* value will assume the value of the effective saltenv when running states. This essentially makes `salt '*' state.sls mysls saltenv=dev` equivalent to `salt '*' state.sls mysls saltenv=dev pillarenv=dev`. If *`pillarenv`* is set, either in the minion config file or via the CLI, it will override this option.

```
pillarenv_from_saltenv: True
```

`pillar_raise_on_missing`

New in version 2015.5.0.

Default: `False`

Set this option to `True` to force a `KeyError` to be raised whenever an attempt to retrieve a named value from pillar fails. When this option is set to `False`, the failed attempt returns an empty string.

`minion_pillar_cache`

New in version 2016.3.0.

Default: `False`

The minion can locally cache rendered pillar data under *`cachedir`*/pillar. This allows a temporarily disconnected minion to access previously cached pillar data by invoking salt-call with the –local and –pillar_root=:conf_minion:*cachedir*/pillar options. Before enabling this setting consider that the rendered pillar may contain security sensitive data. Appropriate access restrictions should be in place. By default the saved pillar data will be readable only by the user account running salt. By default this feature is disabled, to enable set minion_pillar_cache to `True`.

```
minion_pillar_cache: False
```

`file_recv_max_size`

New in version 2014.7.0.

Default: `100`

Set a hard-limit on the size of the files that can be pushed to the master. It will be interpreted as megabytes.

```
file_recv_max_size: 100
```

`pass_to_ext_pillars`

Specify a list of configuration keys whose values are to be passed to external pillar functions.

Suboptions can be specified using the ':' notation (i.e. `option:suboption`)

The values are merged and included in the `extra_minion_data` optional parameter of the external pillar function. The `extra_minion_data` parameter is passed only to the external pillar functions that have it explicitly specified in their definition.

If the config contains

```
opt1: value1
opt2:
  subopt1: value2
  subopt2: value3

pass_to_ext_pillars:
  - opt1
  - opt2: subopt1
```

the `extra_minion_data` parameter will be

```
{'opt1': 'value1',
 'opt2': {'subopt1': 'value2'}}
```

3.2.8 Security Settings

`open_mode`

Default: `False`

Open mode can be used to clean out the PKI key received from the Salt master, turn on open mode, restart the minion, then turn off open mode and restart the minion to clean the keys.

```
open_mode: False
```

`master_finger`

Default: `''`

Fingerprint of the master public key to validate the identity of your Salt master before the initial key exchange. The master fingerprint can be found by running "salt-key -F master" on the Salt master.

```
master_finger: 'ba:30:65:2a:d6:9e:20:4f:d8:b2:f3:a7:d4:65:11:13'
```

`keysize`

Default: `2048`

The size of key that should be generated when creating new keys.

```
keysize: 2048
```

`permissive_pki_access`

Default: `False`

Enable permissive access to the salt keys. This allows you to run the master or minion as root, but have a non-root group be given access to your pki_dir. To make the access explicit, root must belong to the group you've given access to. This is potentially quite insecure.

```
permissive_pki_access: False
```

`verify_master_pubkey_sign`

Default: `False`

Enables verification of the master-public-signature returned by the master in auth-replies. Please see the tutorial on how to configure this properly Multimaster-PKI with Failover Tutorial

New in version 2014.7.0.

```
verify_master_pubkey_sign: True
```

If this is set to `True`, *`master_sign_pubkey`* must be also set to `True` in the master configuration file.

`master_sign_key_name`

Default: `master_sign`

The filename without the *.pub* suffix of the public key that should be used for verifying the signature from the master. The file must be located in the minion's pki directory.

New in version 2014.7.0.

```
master_sign_key_name: <filename_without_suffix>
```

`autosign_grains`

New in version 2018.3.0.

Default: `not defined`

The grains that should be sent to the master on authentication to decide if the minion's key should be accepted automatically.

Please see the *Autoaccept Minions from Grains* documentation for more information.

```
autosign_grains:
  - uuid
  - server_id
```

`always_verify_signature`

Default: `False`

If *`verify_master_pubkey_sign`* is enabled, the signature is only verified if the public-key of the master changes. If the signature should always be verified, this can be set to `True`.

New in version 2014.7.0.

```
always_verify_signature: True
```

`cmd_blacklist_glob`

Default: `[]`

If *`cmd_blacklist_glob`* is enabled then any shell command called over remote execution or via salt-call will be checked against the glob matches found in the *cmd_blacklist_glob* list and any matched shell command will be blocked.

Note: This blacklist is only applied to direct executions made by the *salt* and *salt-call* commands. This does NOT blacklist commands called from states or shell commands executed from other modules.

New in version 2016.11.0.

```
cmd_blacklist_glob:
  - 'rm * '
  - 'cat /etc/* '
```

cmd_whitelist_glob

Default: `[]`

If *`cmd_whitelist_glob`* is enabled then any shell command called over remote execution or via salt-call will be checked against the glob matches found in the *cmd_whitelist_glob* list and any shell command NOT found in the list will be blocked. If *cmd_whitelist_glob* is NOT SET, then all shell commands are permitted.

Note: This whitelist is only applied to direct executions made by the *salt* and *salt-call* commands. This does NOT restrict commands called from states or shell commands executed from other modules.

New in version 2016.11.0.

```
cmd_whitelist_glob:
  - 'ls * '
  - 'cat /etc/fstab'
```

ssl

New in version 2016.11.0.

Default: `None`

TLS/SSL connection options. This could be set to a dictionary containing arguments corresponding to python `ssl.wrap_socket` method. For details see Tornado and Python documentation.

Note: to set enum arguments values like `cert_reqs` and `ssl_version` use constant names without ssl module prefix: `CERT_REQUIRED` or `PROTOCOL_SSLv23`.

```
ssl:
    keyfile: <path_to_keyfile>
    certfile: <path_to_certfile>
    ssl_version: PROTOCOL_TLSv1_2
```

3.2.9 Reactor Settings

reactor

Default: `[]`

Defines a salt reactor. See the *Reactor* documentation for more information.

```
reactor: []
```

`reactor_refresh_interval`

Default: `60`

The TTL for the cache of the reactor configuration.

```
reactor_refresh_interval: 60
```

`reactor_worker_threads`

Default: `10`

The number of workers for the runner/wheel in the reactor.

```
reactor_worker_threads: 10
```

`reactor_worker_hwm`

Default: `10000`

The queue size for workers in the reactor.

```
reactor_worker_hwm: 10000
```

3.2.10 Thread Settings

`multiprocessing`

Default: `True`

If `multiprocessing` is enabled when a minion receives a publication a new process is spawned and the command is executed therein. Conversely, if `multiprocessing` is disabled the new publication will be run executed in a thread.

```
multiprocessing: True
```

`process_count_max`

New in version 2018.3.0.

Default: `-1`

Limit the maximum amount of processes or threads created by `salt-minion`. This is useful to avoid resource exhaustion in case the minion receives more publications than it is able to handle, as it limits the number of spawned processes or threads. `-1` is the default and disables the limit.

```
process_count_max: -1
```

3.2.11 Minion Logging Settings

`log_file`

Default: `/var/log/salt/minion`

The minion log can be sent to a regular file, local path name, or network location. See also *log_file*.

Examples:

```
log_file: /var/log/salt/minion
```

```
log_file: file:///dev/log
```

```
log_file: udp://loghost:10514
```

`log_level`

Default: `warning`

The level of messages to send to the console. See also *log_level*.

```
log_level: warning
```

`log_level_logfile`

Default: `warning`

The level of messages to send to the log file. See also *log_level_logfile*. When it is not set explicitly it will inherit the level set by *log_level* option.

```
log_level_logfile: warning
```

`log_datefmt`

Default: `%H:%M:%S`

The date and time format used in console log messages. See also *log_datefmt*.

```
log_datefmt: '%H:%M:%S'
```

`log_datefmt_logfile`

Default: `%Y-%m-%d %H:%M:%S`

The date and time format used in log file messages. See also *log_datefmt_logfile*.

```
log_datefmt_logfile: '%Y-%m-%d %H:%M:%S'
```

`log_fmt_console`

Default: `[%(levelname)-8s] %(message)s`

The format of the console logging messages. See also *`log_fmt_console`*.

Note: Log colors are enabled in `log_fmt_console` rather than the `color` config since the logging system is loaded before the minion config.

Console log colors are specified by these additional formatters:

%(colorlevel)s %(colorname)s %(colorprocess)s %(colormsg)s

Since it is desirable to include the surrounding brackets, '[' and ']', in the coloring of the messages, these color formatters also include padding as well. Color LogRecord attributes are only available for console logging.

```
log_fmt_console: '%(colorlevel)s %(colormsg)s'
log_fmt_console: '[%(levelname)-8s] %(message)s'
```

`log_fmt_logfile`

Default: `%(asctime)s,%(msecs)03d [%(name)-17s][%(levelname)-8s] %(message)s`

The format of the log file logging messages. See also *`log_fmt_logfile`*.

```
log_fmt_logfile: '%(asctime)s,%(msecs)03d [%(name)-17s][%(levelname)-8s] %(message)s'
```

`log_granular_levels`

Default: `{}`

This can be used to control logging levels more specifically. See also *`log_granular_levels`*.

`log_rotate_max_bytes`

Default: `0`

The maximum number of bytes a single log file may contain before it is rotated. A value of 0 disables this feature. Currently only supported on Windows. On other platforms, use an external tool such as 'logrotate' to manage log files. `log_rotate_max_bytes`

`log_rotate_backup_count`

Default: `0`

The number of backup files to keep when rotating log files. Only used if *`log_rotate_max_bytes`* is greater than 0. Currently only supported on Windows. On other platforms, use an external tool such as 'logrotate' to manage log files. `log_rotate_backup_count`

`zmq_monitor`

Default: `False`

To diagnose issues with minions disconnecting or missing returns, ZeroMQ supports the use of monitor sockets to log connection events. This feature requires ZeroMQ 4.0 or higher.

To enable ZeroMQ monitor sockets, set 'zmq_monitor' to 'True' and log at a debug level or higher.

A sample log event is as follows:

```
[DEBUG   ] ZeroMQ event: {'endpoint': 'tcp://127.0.0.1:4505', 'event': 512,
'value': 27, 'description': 'EVENT_DISCONNECTED'}
```

All events logged will include the string `ZeroMQ event`. A connection event should be logged as the minion starts up and initially connects to the master. If not, check for debug log level and that the necessary version of ZeroMQ is installed.

`tcp_authentication_retries`

Default: `5`

The number of times to retry authenticating with the salt master when it comes back online.

Zeromq does a lot to make sure when connections come back online that they reauthenticate. The tcp transport should try to connect with a new connection if the old one times out on reauthenticating.

-1 for infinite tries.

`failhard`

Default: `False`

Set the global failhard flag. This informs all states to stop running states at the moment a single state fails

```
failhard: False
```

3.2.12 Include Configuration

Configuration can be loaded from multiple files. The order in which this is done is:

1. The minion config file itself
2. The files matching the glob in *`default_include`*
3. The files matching the glob in *`include`* (if defined)

Each successive step overrides any values defined in the previous steps. Therefore, any config options defined in one of the *`default_include`* files would override the same value in the minion config file, and any options defined in *`include`* would override both.

`default_include`

Default: `minion.d/*.conf`

The minion can include configuration from other files. Per default the minion will automatically include all config files from *minion.d/*.conf* where minion.d is relative to the directory of the minion configuration file.

Note: Salt creates files in the `minion.d` directory for its own use. These files are prefixed with an underscore. A common example of this is the `_schedule.conf` file.

`include`

Default: `not defined`

The minion can include configuration from other files. To enable this, pass a list of paths to this option. The paths can be either relative or absolute; if relative, they are considered to be relative to the directory the main minion configuration file lives in. Paths can make use of shell-style globbing. If no files are matched by a path passed to this option then the minion will log a warning message.

```
# Include files from a minion.d directory in the same
# directory as the minion config file
include: minion.d/*.conf

# Include a single extra file into the configuration
include: /etc/roles/webserver

# Include several files and the minion.d directory
include:
  - extra_config
  - minion.d/*
  - /etc/roles/webserver
```

3.2.13 Keepalive Settings

`tcp_keepalive`

Default: `True`

The tcp keepalive interval to set on TCP ports. This setting can be used to tune Salt connectivity issues in messy network environments with misbehaving firewalls.

```
tcp_keepalive: True
```

`tcp_keepalive_cnt`

Default: `-1`

Sets the ZeroMQ TCP keepalive count. May be used to tune issues with minion disconnects.

```
tcp_keepalive_cnt: -1
```

`tcp_keepalive_idle`

Default: `300`

Sets ZeroMQ TCP keepalive idle. May be used to tune issues with minion disconnects.

```
tcp_keepalive_idle: 300
```

`tcp_keepalive_intvl`

Default: `-1`

Sets ZeroMQ TCP keepalive interval. May be used to tune issues with minion disconnects.

```
tcp_keepalive_intvl': -1
```

3.2.14 Frozen Build Update Settings

These options control how *`salt.modules.saltutil.update()`* works with esky frozen apps. For more information look at https://github.com/cloudmatrix/esky/.

`update_url`

Default: `False` (Update feature is disabled)

The url to use when looking for application updates. Esky depends on directory listings to search for new versions. A webserver running on your Master is a good starting point for most setups.

```
update_url: 'http://salt.example.com/minion-updates'
```

`update_restart_services`

Default: `[]` (service restarting on update is disabled)

A list of services to restart when the minion software is updated. This would typically just be a list containing the minion's service name, but you may have other services that need to go with it.

```
update_restart_services: ['salt-minion']
```

`winrepo_cache_expire_min`

New in version 2016.11.0.

Default: `1800`

If set to a nonzero integer, then passing `refresh=True` to functions in the *`windows pkg module`* will not refresh the windows repo metadata if the age of the metadata is less than this value. The exception to this is *`pkg.refresh_db`*, which will always refresh the metadata, regardless of age.

```
winrepo_cache_expire_min: 1800
```

`winrepo_cache_expire_max`

New in version 2016.11.0.

Default: `21600`

If the windows repo metadata is older than this value, and the metadata is needed by a function in the *windows pkg module*, the metadata will be refreshed.

```
winrepo_cache_expire_max: 86400
```

3.2.15 Minion Windows Software Repo Settings

Important: To use these config options, the minion can be running in master-minion or masterless mode.

`winrepo_source_dir`

Default: `salt://win/repo-ng/`

The source location for the winrepo sls files.

```
winrepo_source_dir: salt://win/repo-ng/
```

3.2.16 Standalone Minion Windows Software Repo Settings

Important: To use these config options, the minion must be running in masterless mode (set *file_client* to `local`).

`winrepo_dir`

Changed in version 2015.8.0: Renamed from `win_repo` to `winrepo_dir`. Also, this option did not have a default value until this version.

Default: `C:\salt\srv\salt\win\repo`

Location on the minion where the *winrepo_remotes* are checked out.

```
winrepo_dir: 'D:\winrepo'
```

`winrepo_dir_ng`

New in version 2015.8.0: A new *ng* repo was added.

Default: `/srv/salt/win/repo-ng`

Location on the minion where the *winrepo_remotes_ng* are checked out for 2015.8.0 and later minions.

```
winrepo_dir_ng: /srv/salt/win/repo-ng
```

`winrepo_cachefile`

Changed in version 2015.8.0: Renamed from `win_repo_cachefile` to `winrepo_cachefile`. Also, this option did not have a default value until this version.

Default: `winrepo.p`

Path relative to *`winrepo_dir`* where the winrepo cache should be created.

```
winrepo_cachefile: winrepo.p
```

`winrepo_remotes`

Changed in version 2015.8.0: Renamed from `win_gitrepos` to `winrepo_remotes`. Also, this option did not have a default value until this version.

New in version 2015.8.0.

Default: `['https://github.com/saltstack/salt-winrepo.git']`

List of git repositories to checkout and include in the winrepo

```
winrepo_remotes:
  - https://github.com/saltstack/salt-winrepo.git
```

To specify a specific revision of the repository, prepend a commit ID to the URL of the repository:

```
winrepo_remotes:
  - '<commit_id> https://github.com/saltstack/salt-winrepo.git'
```

Replace `<commit_id>` with the SHA1 hash of a commit ID. Specifying a commit ID is useful in that it allows one to revert back to a previous version in the event that an error is introduced in the latest revision of the repo.

`winrepo_remotes_ng`

New in version 2015.8.0: A new *ng* repo was added.

Default: `['https://github.com/saltstack/salt-winrepo-ng.git']`

List of git repositories to checkout and include in the winrepo for 2015.8.0 and later minions.

```
winrepo_remotes_ng:
  - https://github.com/saltstack/salt-winrepo-ng.git
```

To specify a specific revision of the repository, prepend a commit ID to the URL of the repository:

```
winrepo_remotes_ng:
  - '<commit_id> https://github.com/saltstack/salt-winrepo-ng.git'
```

Replace `<commit_id>` with the SHA1 hash of a commit ID. Specifying a commit ID is useful in that it allows one to revert back to a previous version in the event that an error is introduced in the latest revision of the repo.

`ssh_merge_pillar`

New in version 2018.3.2.

Default: `True`

Merges the compiled pillar data with the pillar data already available globally. This is useful when using `salt-ssh` or `salt-call --local` and overriding the pillar data in a state file:

```
apply_showpillar:
  module.run:
    - name: state.apply
    - mods:
      - showpillar
    - kwargs:
          pillar:
              test: "foo bar"
```

If set to `True` the `showpillar` state will have access to the global pillar data.

If set to `False` only the overriding pillar data will be available to the `showpillar` state.

3.3 Configuring the Salt Proxy Minion

The Salt system is amazingly simple and easy to configure. The two components of the Salt system each have a respective configuration file. The **`salt-master`** is configured via the master configuration file, and the **`salt-proxy`** is configured via the proxy configuration file.

See also:

example proxy minion configuration file

The Salt Minion configuration is very simple. Typically, the only value that needs to be set is the master value so the proxy knows where to locate its master.

By default, the salt-proxy configuration will be in `/etc/salt/proxy`. A notable exception is FreeBSD, where the configuration will be in `/usr/local/etc/salt/proxy`.

3.3.1 Proxy-specific Configuration Options

`add_proxymodule_to_opts`

New in version 2015.8.2.

Changed in version 2016.3.0.

Default: `False`

Add the proxymodule LazyLoader object to opts.

```
add_proxymodule_to_opts: True
```

`proxy_merge_grains_in_module`

New in version 2016.3.0.

Changed in version 2017.7.0.

Default: `True`

If a proxymodule has a function called `grains`, then call it during regular grains loading and merge the results with the proxy's grains dictionary. Otherwise it is assumed that the module calls the grains function in a custom way and returns the data elsewhere.

```
proxy_merge_grains_in_module: False
```

`proxy_keep_alive`

New in version 2017.7.0.

Default: `True`

Whether the connection with the remote device should be restarted when dead. The proxy module must implement the `alive` function, otherwise the connection is considered alive.

```
proxy_keep_alive: False
```

`proxy_keep_alive_interval`

New in version 2017.7.0.

Default: `1`

The frequency of keepalive checks, in minutes. It requires the *proxy_keep_alive* option to be enabled (and the proxy module to implement the `alive` function).

```
proxy_keep_alive_interval: 5
```

`proxy_always_alive`

New in version 2017.7.0.

Default: `True`

Whether the proxy should maintain the connection with the remote device. Similarly to *proxy_keep_alive*, this option is very specific to the design of the proxy module. When *proxy_always_alive* is set to `False`, the connection with the remote device is not maintained and has to be closed after every command.

```
proxy_always_alive: False
```

`proxy_merge_pillar_in_opts`

New in version 2017.7.3.

Default: `False`.

Whether the pillar data to be merged into the proxy configuration options. As multiple proxies can run on the same server, we may need different configuration options for each, while there's one single configuration file. The solution is merging the pillar data of each proxy minion into the opts.

```
proxy_merge_pillar_in_opts: True
```

`proxy_deep_merge_pillar_in_opts`

New in version 2017.7.3.

Default: `False`.

Deep merge of pillar data into configuration opts. This option is evaluated only when `proxy_merge_pillar_in_opts` is enabled.

`proxy_merge_pillar_in_opts_strategy`

New in version 2017.7.3.

Default: `smart`.

The strategy used when merging pillar configuration into opts. This option is evaluated only when `proxy_merge_pillar_in_opts` is enabled.

`proxy_mines_pillar`

New in version 2017.7.3.

Default: `True`.

Allow enabling mine details using pillar data. This evaluates the mine configuration under the pillar, for the following regular minion options that are also equally available on the proxy minion: `mine_interval`, and `mine_functions`.

3.4 Configuration file examples

- *Example master configuration file*
- *Example minion configuration file*
- *Example proxy minion configuration file*

3.4.1 Example master configuration file

```
##### Primary configuration settings #####
##########################################
# This configuration file is used to manage the behavior of the Salt Master.
# Values that are commented out but have an empty line after the comment are
# defaults that do not need to be set in the config. If there is no blank line
# after the comment then the value is presented as an example and is not the
# default.

# Per default, the master will automatically include all config files
# from master.d/*.conf (master.d is a directory in the same directory
# as the main master config file).
#default_include: master.d/*.conf

# The address of the interface to bind to:
```

(continues on next page)

(continued from previous page)

```
#interface: 0.0.0.0

# Whether the master should listen for IPv6 connections. If this is set to True,
# the interface option must be adjusted, too. (For example: "interface: '::'")
#ipv6: False

# The tcp port used by the publisher:
#publish_port: 4505

# The user under which the salt master will run. Salt will update all
# permissions to allow the specified user to run the master. The exception is
# the job cache, which must be deleted if this user is changed. If the
# modified files cause conflicts, set verify_env to False.
#user: root

# Tell the master to also use salt-ssh when running commands against minions.
#enable_ssh_minions: False

# The port used by the communication interface. The ret (return) port is the
# interface used for the file server, authentication, job returns, etc.
#ret_port: 4506

# Specify the location of the daemon process ID file:
#pidfile: /var/run/salt-master.pid

# The root directory prepended to these options: pki_dir, cachedir,
# sock_dir, log_file, autosign_file, autoreject_file, extension_modules,
# key_logfile, pidfile, autosign_grains_dir:
#root_dir: /

# The path to the master's configuration file.
#conf_file: /etc/salt/master

# Directory used to store public key data:
#pki_dir: /etc/salt/pki/master

# Key cache. Increases master speed for large numbers of accepted
# keys. Available options: 'sched'. (Updates on a fixed schedule.)
# Note that enabling this feature means that minions will not be
# available to target for up to the length of the maintanence loop
# which by default is 60s.
#key_cache: ''

# Directory to store job and cache data:
# This directory may contain sensitive data and should be protected accordingly.
#
#cachedir: /var/cache/salt/master

# Directory for custom modules. This directory can contain subdirectories for
# each of Salt's module types such as "runners", "output", "wheel", "modules",
# "states", "returners", "engines", "utils", etc.
#extension_modules: /var/cache/salt/master/extmods

# Directory for custom modules. This directory can contain subdirectories for
# each of Salt's module types such as "runners", "output", "wheel", "modules",
# "states", "returners", "engines", "utils", etc.
```

(continues on next page)

(continued from previous page)

```
# Like 'extension_modules' but can take an array of paths
#module_dirs: []

# Verify and set permissions on configuration directories at startup:
#verify_env: True

# Set the number of hours to keep old job information in the job cache:
#keep_jobs: 24

# The number of seconds to wait when the client is requesting information
# about running jobs.
#gather_job_timeout: 10

# Set the default timeout for the salt command and api. The default is 5
# seconds.
#timeout: 5

# The loop_interval option controls the seconds for the master's maintenance
# process check cycle. This process updates file server backends, cleans the
# job cache and executes the scheduler.
#loop_interval: 60

# Set the default outputter used by the salt command. The default is "nested".
#output: nested

# To set a list of additional directories to search for salt outputters, set the
# outputter_dirs option.
#outputter_dirs: []

# Set the default output file used by the salt command. Default is to output
# to the CLI and not to a file. Functions the same way as the "--out-file"
# CLI option, only sets this to a single file for all salt commands.
#output_file: None

# Return minions that timeout when running commands like test.ping
#show_timeout: True

# Tell the client to display the jid when a job is published.
#show_jid: False

# By default, output is colored. To disable colored output, set the color value
# to False.
#color: True

# Do not strip off the colored output from nested results and state outputs
# (true by default).
# strip_colors: False

# To display a summary of the number of minions targeted, the number of
# minions returned, and the number of minions that did not return, set the
# cli_summary value to True. (False by default.)
#
#cli_summary: False

# Set the directory used to hold unix sockets:
#sock_dir: /var/run/salt/master
```

(continues on next page)

(continued from previous page)

```
# The master can take a while to start up when lspci and/or dmidecode is used
# to populate the grains for the master. Enable if you want to see GPU hardware
# data for your master.
# enable_gpu_grains: False

# The master maintains a job cache. While this is a great addition, it can be
# a burden on the master for larger deployments (over 5000 minions).
# Disabling the job cache will make previously executed jobs unavailable to
# the jobs system and is not generally recommended.
#job_cache: True

# Cache minion grains, pillar and mine data via the cache subsystem in the
# cachedir or a database.
#minion_data_cache: True

# Cache subsystem module to use for minion data cache.
#cache: localfs
# Enables a fast in-memory cache booster and sets the expiration time.
#memcache_expire_seconds: 0
# Set a memcache limit in items (bank + key) per cache storage (driver + driver_opts).
#memcache_max_items: 1024
# Each time a cache storage got full cleanup all the expired items not just the
↪oldest one.
#memcache_full_cleanup: False
# Enable collecting the memcache stats and log it on `debug` log level.
#memcache_debug: False

# Store all returns in the given returner.
# Setting this option requires that any returner-specific configuration also
# be set. See various returners in salt/returners for details on required
# configuration values. (See also, event_return_queue, and event_return_queue_max_
↪seconds below.)
#
#event_return: mysql

# On busy systems, enabling event_returns can cause a considerable load on
# the storage system for returners. Events can be queued on the master and
# stored in a batched fashion using a single transaction for multiple events.
# By default, events are not queued.
#event_return_queue: 0

# In some cases enabling event return queueing can be very helpful, but the bus
# may not busy enough to flush the queue consistently.  Setting this to a reasonable
# value (1-30 seconds) will cause the queue to be flushed when the oldest event is
↪older
# than `event_return_queue_max_seconds` regardless of how many events are in the
↪queue.
#event_return_queue_max_seconds: 0

# Only return events matching tags in a whitelist, supports glob matches.
#event_return_whitelist:
#   - salt/master/a_tag
#   - salt/run/*/ret

# Store all event returns **except** the tags in a blacklist, supports globs.
```

(continues on next page)

(continued from previous page)

```
#event_return_blacklist:
#   - salt/master/not_this_tag
#   - salt/wheel/*/ret

# Passing very large events can cause the minion to consume large amounts of
# memory. This value tunes the maximum size of a message allowed onto the
# master event bus. The value is expressed in bytes.
#max_event_size: 1048576

# Windows platforms lack posix IPC and must rely on slower TCP based inter-
# process communications. Set ipc_mode to 'tcp' on such systems
#ipc_mode: ipc

# Overwrite the default tcp ports used by the minion when ipc_mode is set to 'tcp'
#tcp_master_pub_port: 4510
#tcp_master_pull_port: 4511

# By default, the master AES key rotates every 24 hours. The next command
# following a key rotation will trigger a key refresh from the minion which may
# result in minions which do not respond to the first command after a key refresh.
#
# To tell the master to ping all minions immediately after an AES key refresh, set
# ping_on_rotate to True. This should mitigate the issue where a minion does not
# appear to initially respond after a key is rotated.
#
# Note that ping_on_rotate may cause high load on the master immediately after
# the key rotation event as minions reconnect. Consider this carefully if this
# salt master is managing a large number of minions.
#
# If disabled, it is recommended to handle this event by listening for the
# 'aes_key_rotate' event with the 'key' tag and acting appropriately.
# ping_on_rotate: False

# By default, the master deletes its cache of minion data when the key for that
# minion is removed. To preserve the cache after key deletion, set
# 'preserve_minion_cache' to True.
#
# WARNING: This may have security implications if compromised minions auth with
# a previous deleted minion ID.
#preserve_minion_cache: False

# Allow or deny minions from requesting their own key revocation
#allow_minion_key_revoke: True

# If max_minions is used in large installations, the master might experience
# high-load situations because of having to check the number of connected
# minions for every authentication. This cache provides the minion-ids of
# all connected minions to all MWorker-processes and greatly improves the
# performance of max_minions.
# con_cache: False

# The master can include configuration from other files. To enable this,
# pass a list of paths to this option. The paths can be either relative or
# absolute; if relative, they are considered to be relative to the directory
# the main master configuration file lives in (this file). Paths can make use
# of shell-style globbing. If no files are matched by a path passed to this
```

(continues on next page)

(continued from previous page)

```
# option, then the master will log a warning message.
#
# Include a config file from some other path:
# include: /etc/salt/extra_config
#
# Include config from several files and directories:
# include:
#   - /etc/salt/extra_config


#####  Large-scale tuning settings   #####
##########################################
# Max open files
#
# Each minion connecting to the master uses AT LEAST one file descriptor, the
# master subscription connection. If enough minions connect you might start
# seeing on the console (and then salt-master crashes):
#   Too many open files (tcp_listener.cpp:335)
#   Aborted (core dumped)
#
# By default this value will be the one of `ulimit -Hn`, ie, the hard limit for
# max open files.
#
# If you wish to set a different value than the default one, uncomment and
# configure this setting. Remember that this value CANNOT be higher than the
# hard limit. Raising the hard limit depends on your OS and/or distribution,
# a good way to find the limit is to search the internet. For example:
#   raise max open files hard limit debian
#
#max_open_files: 100000

# The number of worker threads to start. These threads are used to manage
# return calls made from minions to the master. If the master seems to be
# running slowly, increase the number of threads. This setting can not be
# set lower than 3.
#worker_threads: 5

# Set the ZeroMQ high water marks
# http://api.zeromq.org/3-2:zmq-setsockopt

# The listen queue size / backlog
#zmq_backlog: 1000

# The publisher interface ZeroMQPubServerChannel
#pub_hwm: 1000

# The master may allocate memory per-event and not
# reclaim it.
# To set a high-water mark for memory allocation, use
# ipc_write_buffer to set a high-water mark for message
# buffering.
# Value: In bytes. Set to 'dynamic' to have Salt select
# a value for you. Default is disabled.
# ipc_write_buffer: 'dynamic'

# These two batch settings, batch_safe_limit and batch_safe_size, are used to
```

(continues on next page)

(continued from previous page)

```
# automatically switch to a batch mode execution. If a command would have been
# sent to more than <batch_safe_limit> minions, then run the command in
# batches of <batch_safe_size>. If no batch_safe_size is specified, a default
# of 8 will be used. If no batch_safe_limit is specified, then no automatic
# batching will occur.
#batch_safe_limit: 100
#batch_safe_size: 8

# Master stats enables stats events to be fired from the master at close
# to the defined interval
#master_stats: False
#master_stats_event_iter: 60


#####        Security settings       #####
##########################################
# Enable passphrase protection of Master private key.  Although a string value
# is acceptable; passwords should be stored in an external vaulting mechanism
# and retrieved via sdb. See https://docs.saltstack.com/en/latest/topics/sdb/.
# Passphrase protection is off by default but an example of an sdb profile and
# query is as follows.
# masterkeyring:
#  driver: keyring
#  service: system
#
# key_pass: sdb://masterkeyring/key_pass

# Enable passphrase protection of the Master signing_key. This only applies if
# master_sign_pubkey is set to True.  This is disabled by default.
# master_sign_pubkey: True
# signing_key_pass: sdb://masterkeyring/signing_pass

# Enable "open mode", this mode still maintains encryption, but turns off
# authentication, this is only intended for highly secure environments or for
# the situation where your keys end up in a bad state. If you run in open mode
# you do so at your own risk!
#open_mode: False

# Enable auto_accept, this setting will automatically accept all incoming
# public keys from the minions. Note that this is insecure.
#auto_accept: False

# The size of key that should be generated when creating new keys.
#keysize: 2048

# Time in minutes that an incoming public key with a matching name found in
# pki_dir/minion_autosign/keyid is automatically accepted. Expired autosign keys
# are removed when the master checks the minion_autosign directory.
# 0 equals no timeout
# autosign_timeout: 120

# If the autosign_file is specified, incoming keys specified in the
# autosign_file will be automatically accepted. This is insecure.  Regular
# expressions as well as globing lines are supported. The file must be readonly
# except for the owner. Use permissive_pki_access to allow the group write access.
#autosign_file: /etc/salt/autosign.conf
```

(continues on next page)

(continued from previous page)

```
# Works like autosign_file, but instead allows you to specify minion IDs for
# which keys will automatically be rejected. Will override both membership in
# the autosign_file and the auto_accept setting.
#autoreject_file: /etc/salt/autoreject.conf

# If the autosign_grains_dir is specified, incoming keys from minions with grain
# values matching those defined in files in this directory will be accepted
# automatically. This is insecure. Minions need to be configured to send the grains.
#autosign_grains_dir: /etc/salt/autosign_grains

# Enable permissive access to the salt keys. This allows you to run the
# master or minion as root, but have a non-root group be given access to
# your pki_dir. To make the access explicit, root must belong to the group
# you've given access to. This is potentially quite insecure. If an autosign_file
# is specified, enabling permissive_pki_access will allow group access to that
# specific file.
#permissive_pki_access: False

# Allow users on the master access to execute specific commands on minions.
# This setting should be treated with care since it opens up execution
# capabilities to non root users. By default this capability is completely
# disabled.
#publisher_acl:
#   larry:
#     - test.ping
#     - network.*
#
# Blacklist any of the following users or modules
#
# This example would blacklist all non sudo users, including root from
# running any commands. It would also blacklist any use of the "cmd"
# module. This is completely disabled by default.
#
#
# Check the list of configured users in client ACL against users on the
# system and throw errors if they do not exist.
#client_acl_verify: True
#
#publisher_acl_blacklist:
#   users:
#     - root
#     - '^(?!sudo_).*$'   #  all non sudo users
#   modules:
#     - cmd

# Enforce publisher_acl & publisher_acl_blacklist when users have sudo
# access to the salt command.
#
#sudo_acl: False

# The external auth system uses the Salt auth modules to authenticate and
# validate users to access areas of the Salt system.
#external_auth:
#   pam:
#     fred:
```

(continues on next page)

(continued from previous page)

```
#      - test.*
#
# Time (in seconds) for a newly generated token to live. Default: 12 hours
#token_expire: 43200
#
# Allow eauth users to specify the expiry time of the tokens they generate.
# A boolean applies to all users or a dictionary of whitelisted eauth backends
# and usernames may be given.
# token_expire_user_override:
#   pam:
#     - fred
#     - tom
#   ldap:
#     - gary
#
#token_expire_user_override: False

# Set to True to enable keeping the calculated user's auth list in the token
# file. This is disabled by default and the auth list is calculated or requested
# from the eauth driver each time.
#keep_acl_in_token: False

# Auth subsystem module to use to get authorized access list for a user. By default it
↪'s
# the same module used for external authentication.
#eauth_acl_module: django

# Allow minions to push files to the master. This is disabled by default, for
# security purposes.
#file_recv: False

# Set a hard-limit on the size of the files that can be pushed to the master.
# It will be interpreted as megabytes. Default: 100
#file_recv_max_size: 100

# Signature verification on messages published from the master.
# This causes the master to cryptographically sign all messages published to its event
# bus, and minions then verify that signature before acting on the message.
#
# This is False by default.
#
# Note that to facilitate interoperability with masters and minions that are different
# versions, if sign_pub_messages is True but a message is received by a minion with
# no signature, it will still be accepted, and a warning message will be logged.
# Conversely, if sign_pub_messages is False, but a minion receives a signed
# message it will be accepted, the signature will not be checked, and a warning
↪message
# will be logged. This behavior went away in Salt 2014.1.0 and these two situations
# will cause minion to throw an exception and drop the message.
# sign_pub_messages: False

# Signature verification on messages published from minions
# This requires that minions cryptographically sign the messages they
# publish to the master.  If minions are not signing, then log this information
# at loglevel 'INFO' and drop the message without acting on it.
# require_minion_sign_messages: False
```

(continues on next page)

(continued from previous page)

```
# The below will drop messages when their signatures do not validate.
# Note that when this option is False but `require_minion_sign_messages` is True
# minions MUST sign their messages but the validity of their signatures
# is ignored.
# These two config options exist so a Salt infrastructure can be moved
# to signing minion messages gradually.
# drop_messages_signature_fail: False

# Use TLS/SSL encrypted connection between master and minion.
# Can be set to a dictionary containing keyword arguments corresponding to Python's
# 'ssl.wrap_socket' method.
# Default is None.
#ssl:
#    keyfile: <path_to_keyfile>
#    certfile: <path_to_certfile>
#    ssl_version: PROTOCOL_TLSv1_2

#####     Salt-SSH Configuration     #####
##########################################
# Define the default salt-ssh roster module to use
#roster: flat

# Pass in an alternative location for the salt-ssh `flat` roster file
#roster_file: /etc/salt/roster

# Define locations for `flat` roster files so they can be chosen when using Salt API.
# An administrator can place roster files into these locations. Then when
# calling Salt API, parameter 'roster_file' should contain a relative path to
# these locations. That is, "roster_file=/foo/roster" will be resolved as
# "/etc/salt/roster.d/foo/roster" etc. This feature prevents passing insecure
# custom rosters through the Salt API.
#
#rosters:
# - /etc/salt/roster.d
# - /opt/salt/some/more/rosters

# The ssh password to log in with.
#ssh_passwd: ''

#The target system's ssh port number.
#ssh_port: 22

# Comma-separated list of ports to scan.
#ssh_scan_ports: 22

# Scanning socket timeout for salt-ssh.
#ssh_scan_timeout: 0.01

# Boolean to run command via sudo.
#ssh_sudo: False

# Number of seconds to wait for a response when establishing an SSH connection.
#ssh_timeout: 60

# The user to log in as.
```

(continues on next page)

(continued from previous page)

```
#ssh_user: root

# The log file of the salt-ssh command:
#ssh_log_file: /var/log/salt/ssh

# Pass in minion option overrides that will be inserted into the SHIM for
# salt-ssh calls. The local minion config is not used for salt-ssh. Can be
# overridden on a per-minion basis in the roster (`minion_opts`)
#ssh_minion_opts:
#  gpg_keydir: /root/gpg

# Set this to True to default to using ~/.ssh/id_rsa for salt-ssh
# authentication with minions
#ssh_use_home_key: False

# Set this to True to default salt-ssh to run with ``-o IdentitiesOnly=yes``.
# This option is intended for situations where the ssh-agent offers many
# different identities and allows ssh to ignore those identities and use the
# only one specified in options.
#ssh_identities_only: False

# List-only nodegroups for salt-ssh. Each group must be formed as either a
# comma-separated list, or a YAML list. This option is useful to group minions
# into easy-to-target groups when using salt-ssh. These groups can then be
# targeted with the normal -N argument to salt-ssh.
#ssh_list_nodegroups: {}

# salt-ssh has the ability to update the flat roster file if a minion is not
# found in the roster.  Set this to True to enable it.
#ssh_update_roster: False

#####    Master Module Management    #####
##########################################
# Manage how master side modules are loaded.

# Add any additional locations to look for master runners:
#runner_dirs: []

# Add any additional locations to look for master utils:
#utils_dirs: []

# Enable Cython for master side modules:
#cython_enable: False


#####      State System settings     #####
##########################################
# The state system uses a "top" file to tell the minions what environment to
# use and what modules to use. The state_top file is defined relative to the
# root of the base environment as defined in "File Server settings" below.
#state_top: top.sls

# The master_tops option replaces the external_nodes option by creating
# a plugable system for the generation of external top data. The external_nodes
# option is deprecated by the master_tops option.
#
```

(continues on next page)

(continued from previous page)

```
# To gain the capabilities of the classic external_nodes system, use the
# following configuration:
# master_tops:
#   ext_nodes: <Shell command which returns yaml>
#
#master_tops: {}

# The renderer to use on the minions to render the state data
#renderer: jinja|yaml

# Default Jinja environment options for all templates except sls templates
#jinja_env:
#  block_start_string: '{%'
#  block_end_string: '%}'
#  variable_start_string: '{{'
#  variable_end_string: '}}'
#  comment_start_string: '{#'
#  comment_end_string: '#}'
#  line_statement_prefix:
#  line_comment_prefix:
#  trim_blocks: False
#  lstrip_blocks: False
#  newline_sequence: '\n'
#  keep_trailing_newline: False

# Jinja environment options for sls templates
#jinja_sls_env:
#  block_start_string: '{%'
#  block_end_string: '%}'
#  variable_start_string: '{{'
#  variable_end_string: '}}'
#  comment_start_string: '{#'
#  comment_end_string: '#}'
#  line_statement_prefix:
#  line_comment_prefix:
#  trim_blocks: False
#  lstrip_blocks: False
#  newline_sequence: '\n'
#  keep_trailing_newline: False

# The failhard option tells the minions to stop immediately after the first
# failure detected in the state execution, defaults to False
#failhard: False

# The state_verbose and state_output settings can be used to change the way
# state system data is printed to the display. By default all data is printed.
# The state_verbose setting can be set to True or False, when set to False
# all data that has a result of True and no changes will be suppressed.
#state_verbose: True

# The state_output setting controls which results will be output full multi line
# full, terse - each state will be full/terse
# mixed - only states with errors will be full
# changes - states with changes and errors will be full
# full_id, mixed_id, changes_id and terse_id are also allowed;
# when set, the state ID will be used as name in the output
```

(continues on next page)

(continued from previous page)

```
#state_output: full

# The state_output_diff setting changes whether or not the output from
# successful states is returned. Useful when even the terse output of these
# states is cluttering the logs. Set it to True to ignore them.
#state_output_diff: False

# Automatically aggregate all states that have support for mod_aggregate by
# setting to 'True'. Or pass a list of state module names to automatically
# aggregate just those types.
#
# state_aggregate:
#   - pkg
#
#state_aggregate: False

# Send progress events as each function in a state run completes execution
# by setting to 'True'. Progress events are in the format
# 'salt/job/<JID>/prog/<MID>/<RUN NUM>'.
#state_events: False

#####      File Server settings      #####
##########################################
# Salt runs a lightweight file server written in zeromq to deliver files to
# minions. This file server is built into the master daemon and does not
# require a dedicated port.

# The file server works on environments passed to the master, each environment
# can have multiple root directories, the subdirectories in the multiple file
# roots cannot match, otherwise the downloaded files will not be able to be
# reliably ensured. A base environment is required to house the top file.
# Example:
# file_roots:
#   base:
#     - /srv/salt/
#   dev:
#     - /srv/salt/dev/services
#     - /srv/salt/dev/states
#   prod:
#     - /srv/salt/prod/services
#     - /srv/salt/prod/states
#
#file_roots:
#  base:
#    - /srv/salt
#

# The master_roots setting configures a master-only copy of the file_roots dictionary,
# used by the state compiler.
#master_roots: /srv/salt-master

# When using multiple environments, each with their own top file, the
# default behaviour is an unordered merge. To prevent top files from
# being merged together and instead to only use the top file from the
# requested environment, set this value to 'same'.
#top_file_merging_strategy: merge
```

(continues on next page)

(continued from previous page)

```
# To specify the order in which environments are merged, set the ordering
# in the env_order option. Given a conflict, the last matching value will
# win.
#env_order: ['base', 'dev', 'prod']

# If top_file_merging_strategy is set to 'same' and an environment does not
# contain a top file, the top file in the environment specified by default_top
# will be used instead.
#default_top: base

# The hash_type is the hash to use when discovering the hash of a file on
# the master server. The default is sha256, but md5, sha1, sha224, sha384 and
# sha512 are also supported.
#
# WARNING: While md5 and sha1 are also supported, do not use them due to the
# high chance of possible collisions and thus security breach.
#
# Prior to changing this value, the master should be stopped and all Salt
# caches should be cleared.
#hash_type: sha256

# The buffer size in the file server can be adjusted here:
#file_buffer_size: 1048576

# A regular expression (or a list of expressions) that will be matched
# against the file path before syncing the modules and states to the minions.
# This includes files affected by the file.recurse state.
# For example, if you manage your custom modules and states in subversion
# and don't want all the '.svn' folders and content synced to your minions,
# you could set this to '/\.svn($|/)'. By default nothing is ignored.
#file_ignore_regex:
#  - '/\.svn($|/)'
#  - '/\.git($|/)'

# A file glob (or list of file globs) that will be matched against the file
# path before syncing the modules and states to the minions. This is similar
# to file_ignore_regex above, but works on globs instead of regex. By default
# nothing is ignored.
# file_ignore_glob:
#  - '*.pyc'
#  - '*/somefolder/*.bak'
#  - '*.swp'

# File Server Backend
#
# Salt supports a modular fileserver backend system, this system allows
# the salt master to link directly to third party systems to gather and
# manage the files available to minions. Multiple backends can be
# configured and will be searched for the requested file in the order in which
# they are defined here. The default setting only enables the standard backend
# "roots" which uses the "file_roots" option.
#fileserver_backend:
#  - roots
#
# To use multiple backends list them in the order they are searched:
```

(continues on next page)

(continued from previous page)

```
#fileserver_backend:
#  - git
#  - roots
#
# Uncomment the line below if you do not want the file_server to follow
# symlinks when walking the filesystem tree. This is set to True
# by default. Currently this only applies to the default roots
# fileserver_backend.
#fileserver_followsymlinks: False
#
# Uncomment the line below if you do not want symlinks to be
# treated as the files they are pointing to. By default this is set to
# False. By uncommenting the line below, any detected symlink while listing
# files on the Master will not be returned to the Minion.
#fileserver_ignoresymlinks: True
#
# By default, the Salt fileserver recurses fully into all defined environments
# to attempt to find files. To limit this behavior so that the fileserver only
# traverses directories with SLS files and special Salt directories like _modules,
# enable the option below. This might be useful for installations where a file root
# has a very large number of files and performance is impacted. Default is False.
# fileserver_limit_traversal: False
#
# The fileserver can fire events off every time the fileserver is updated,
# these are disabled by default, but can be easily turned on by setting this
# flag to True
#fileserver_events: False

# Git File Server Backend Configuration
#
# Optional parameter used to specify the provider to be used for gitfs. Must be
# either pygit2 or gitpython. If unset, then both will be tried (in that
# order), and the first one with a compatible version installed will be the
# provider that is used.
#
#gitfs_provider: pygit2

# Along with gitfs_password, is used to authenticate to HTTPS remotes.
# gitfs_user: ''

# Along with gitfs_user, is used to authenticate to HTTPS remotes.
# This parameter is not required if the repository does not use authentication.
#gitfs_password: ''

# By default, Salt will not authenticate to an HTTP (non-HTTPS) remote.
# This parameter enables authentication over HTTP. Enable this at your own risk.
#gitfs_insecure_auth: False

# Along with gitfs_privkey (and optionally gitfs_passphrase), is used to
# authenticate to SSH remotes. This parameter (or its per-remote counterpart)
# is required for SSH remotes.
#gitfs_pubkey: ''

# Along with gitfs_pubkey (and optionally gitfs_passphrase), is used to
# authenticate to SSH remotes. This parameter (or its per-remote counterpart)
# is required for SSH remotes.
```

(continues on next page)

(continued from previous page)

```
#gitfs_privkey: ''

# This parameter is optional, required only when the SSH key being used to
# authenticate is protected by a passphrase.
#gitfs_passphrase: ''

# When using the git fileserver backend at least one git remote needs to be
# defined. The user running the salt master will need read access to the repo.
#
# The repos will be searched in order to find the file requested by a client
# and the first repo to have the file will return it.
# When using the git backend branches and tags are translated into salt
# environments.
# Note: file:// repos will be treated as a remote, so refs you want used must
# exist in that repo as *local* refs.
#gitfs_remotes:
#  - git://github.com/saltstack/salt-states.git
#  - file:///var/git/saltmaster
#
# The gitfs_ssl_verify option specifies whether to ignore ssl certificate
# errors when contacting the gitfs backend. You might want to set this to
# false if you're using a git backend that uses a self-signed certificate but
# keep in mind that setting this flag to anything other than the default of True
# is a security concern, you may want to try using the ssh transport.
#gitfs_ssl_verify: True
#
# The gitfs_root option gives the ability to serve files from a subdirectory
# within the repository. The path is defined relative to the root of the
# repository and defaults to the repository root.
#gitfs_root: somefolder/otherfolder
#
# The refspecs fetched by gitfs remotes
#gitfs_refspecs:
#  - '+refs/heads/*:refs/remotes/origin/*'
#  - '+refs/tags/*:refs/tags/*'
#
#
#####         Pillar settings        #####
##########################################
# Salt Pillars allow for the building of global data that can be made selectively
# available to different minions based on minion grain filtering. The Salt
# Pillar is laid out in the same fashion as the file server, with environments,
# a top file and sls files. However, pillar data does not need to be in the
# highstate format, and is generally just key/value pairs.
#pillar_roots:
#  base:
#    - /srv/pillar
#
#ext_pillar:
#  - hiera: /etc/hiera.yaml
#  - cmd_yaml: cat /etc/salt/yaml


# A list of paths to be recursively decrypted during pillar compilation.
# Entries in this list can be formatted either as a simple string, or as a
# key/value pair, with the key being the pillar location, and the value being
```

(continues on next page)

(continued from previous page)

```
# the renderer to use for pillar decryption. If the former is used, the
# renderer specified by decrypt_pillar_default will be used.
#decrypt_pillar:
#  - 'foo:bar': gpg
#  - 'lorem:ipsum:dolor'

# The delimiter used to distinguish nested data structures in the
# decrypt_pillar option.
#decrypt_pillar_delimiter: ':'

# The default renderer used for decryption, if one is not specified for a given
# pillar key in decrypt_pillar.
#decrypt_pillar_default: gpg

# List of renderers which are permitted to be used for pillar decryption.
#decrypt_pillar_renderers:
#  - gpg

# The ext_pillar_first option allows for external pillar sources to populate
# before file system pillar. This allows for targeting file system pillar from
# ext_pillar.
#ext_pillar_first: False

# The external pillars permitted to be used on-demand using pillar.ext
#on_demand_ext_pillar:
#  - libvirt
#  - virtkey

# The pillar_gitfs_ssl_verify option specifies whether to ignore ssl certificate
# errors when contacting the pillar gitfs backend. You might want to set this to
# false if you're using a git backend that uses a self-signed certificate but
# keep in mind that setting this flag to anything other than the default of True
# is a security concern, you may want to try using the ssh transport.
#pillar_gitfs_ssl_verify: True

# The pillar_opts option adds the master configuration file data to a dict in
# the pillar called "master". This is used to set simple configurations in the
# master config file that can then be used on minions.
#pillar_opts: False

# The pillar_safe_render_error option prevents the master from passing pillar
# render errors to the minion. This is set on by default because the error could
# contain templating data which would give that minion information it shouldn't
# have, like a password! When set true the error message will only show:
#   Rendering SLS 'my.sls' failed. Please see master log for details.
#pillar_safe_render_error: True

# The pillar_source_merging_strategy option allows you to configure merging strategy
# between different sources. It accepts five values: none, recurse, aggregate,
↪overwrite,
# or smart. None will not do any merging at all. Recurse will merge recursively
↪mapping of data.
# Aggregate instructs aggregation of elements between sources that use the #!yamlex
↪renderer. Overwrite
# will overwrite elements according the order in which they are processed. This is
# behavior of the 2014.1 branch and earlier. Smart guesses the best strategy based
```

(continues on next page)

(continued from previous page)

```
# on the "renderer" setting and is the default value.
#pillar_source_merging_strategy: smart

# Recursively merge lists by aggregating them instead of replacing them.
#pillar_merge_lists: False

# Set this option to True to force the pillarenv to be the same as the effective
# saltenv when running states. If pillarenv is specified this option will be
# ignored.
#pillarenv_from_saltenv: False

# Set this option to 'True' to force a 'KeyError' to be raised whenever an
# attempt to retrieve a named value from pillar fails. When this option is set
# to 'False', the failed attempt returns an empty string. Default is 'False'.
#pillar_raise_on_missing: False

# Git External Pillar (git_pillar) Configuration Options
#
# Specify the provider to be used for git_pillar. Must be either pygit2 or
# gitpython. If unset, then both will be tried in that same order, and the
# first one with a compatible version installed will be the provider that
# is used.
#git_pillar_provider: pygit2

# If the desired branch matches this value, and the environment is omitted
# from the git_pillar configuration, then the environment for that git_pillar
# remote will be base.
#git_pillar_base: master

# If the branch is omitted from a git_pillar remote, then this branch will
# be used instead
#git_pillar_branch: master

# Environment to use for git_pillar remotes. This is normally derived from
# the branch/tag (or from a per-remote env parameter), but if set this will
# override the process of deriving the env from the branch/tag name.
#git_pillar_env: ''

# Path relative to the root of the repository where the git_pillar top file
# and SLS files are located.
#git_pillar_root: ''

# Specifies whether or not to ignore SSL certificate errors when contacting
# the remote repository.
#git_pillar_ssl_verify: False

# When set to False, if there is an update/checkout lock for a git_pillar
# remote and the pid written to it is not running on the master, the lock
# file will be automatically cleared and a new lock will be obtained.
#git_pillar_global_lock: True

# Git External Pillar Authentication Options
#
# Along with git_pillar_password, is used to authenticate to HTTPS remotes.
#git_pillar_user: ''
```

(continues on next page)

(continued from previous page)

```
# Along with git_pillar_user, is used to authenticate to HTTPS remotes.
# This parameter is not required if the repository does not use authentication.
#git_pillar_password: ''

# By default, Salt will not authenticate to an HTTP (non-HTTPS) remote.
# This parameter enables authentication over HTTP.
#git_pillar_insecure_auth: False

# Along with git_pillar_privkey (and optionally git_pillar_passphrase),
# is used to authenticate to SSH remotes.
#git_pillar_pubkey: ''

# Along with git_pillar_pubkey (and optionally git_pillar_passphrase),
# is used to authenticate to SSH remotes.
#git_pillar_privkey: ''

# This parameter is optional, required only when the SSH key being used
# to authenticate is protected by a passphrase.
#git_pillar_passphrase: ''

# The refspecs fetched by git_pillar remotes
#git_pillar_refspecs:
#  - '+refs/heads/*:refs/remotes/origin/*'
#  - '+refs/tags/*:refs/tags/*'

# A master can cache pillars locally to bypass the expense of having to render them
# for each minion on every request. This feature should only be enabled in cases
# where pillar rendering time is known to be unsatisfactory and any attendant security
# concerns about storing pillars in a master cache have been addressed.
#
# When enabling this feature, be certain to read through the additional ``pillar_
↪cache_*``
# configuration options to fully understand the tunable parameters and their↩
↪implications.
#
# Note: setting ``pillar_cache: True`` has no effect on targeting Minions with↩
↪Pillars.
# See https://docs.saltstack.com/en/latest/topics/targeting/pillar.html
#pillar_cache: False

# If and only if a master has set ``pillar_cache: True``, the cache TTL controls the↩
↪amount
# of time, in seconds, before the cache is considered invalid by a master and a fresh
# pillar is recompiled and stored.
#pillar_cache_ttl: 3600

# If and only if a master has set `pillar_cache: True`, one of several storage↩
↪providers
# can be utilized.
#
# `disk`: The default storage backend. This caches rendered pillars to the master↩
↪cache.
#         Rendered pillars are serialized and deserialized as msgpack structures for↩
↪speed.
#         Note that pillars are stored UNENCRYPTED. Ensure that the master cache
#         has permissions set appropriately. (Same defaults are provided.)
```

(continues on next page)

(continued from previous page)

```
#
# memory: [EXPERIMENTAL] An optional backend for pillar caches which uses a pure-
↪Python
#         in-memory data structure for maximal performance. There are several caveats,
#         however. First, because each master worker contains its own in-memory cache,
#         there is no guarantee of cache consistency between minion requests. This
#         works best in situations where the pillar rarely if ever changes. Secondly,
#         and perhaps more importantly, this means that unencrypted pillars will
#         be accessible to any process which can examine the memory of the ``salt-
↪master``!
#         This may represent a substantial security risk.
#
#pillar_cache_backend: disk


######        Reactor Settings        #####
###########################################
# Define a salt reactor. See https://docs.saltstack.com/en/latest/topics/reactor/
#reactor: []

#Set the TTL for the cache of the reactor configuration.
#reactor_refresh_interval: 60

#Configure the number of workers for the runner/wheel in the reactor.
#reactor_worker_threads: 10

#Define the queue size for workers in the reactor.
#reactor_worker_hwm: 10000


#####          Syndic settings       #####
##########################################
# The Salt syndic is used to pass commands through a master from a higher
# master. Using the syndic is simple. If this is a master that will have
# syndic servers(s) below it, then set the "order_masters" setting to True.
#
# If this is a master that will be running a syndic daemon for passthrough, then
# the "syndic_master" setting needs to be set to the location of the master server
# to receive commands from.

# Set the order_masters setting to True if this master will command lower
# masters' syndic interfaces.
#order_masters: False

# If this master will be running a salt syndic daemon, syndic_master tells
# this master where to receive commands from.
#syndic_master: masterofmasters

# This is the 'ret_port' of the MasterOfMaster:
#syndic_master_port: 4506

# PID file of the syndic daemon:
#syndic_pidfile: /var/run/salt-syndic.pid

# The log file of the salt-syndic daemon:
#syndic_log_file: /var/log/salt/syndic
```

(continues on next page)

(continued from previous page)

```
# The behaviour of the multi-syndic when connection to a master of masters failed.
# Can specify ``random`` (default) or ``ordered``. If set to ``random``, masters
# will be iterated in random order. If ``ordered`` is specified, the configured
# order will be used.
#syndic_failover: random

# The number of seconds for the salt client to wait for additional syndics to
# check in with their lists of expected minions before giving up.
#syndic_wait: 5


#####      Peer Publish settings     #####
##########################################
# Salt minions can send commands to other minions, but only if the minion is
# allowed to. By default "Peer Publication" is disabled, and when enabled it
# is enabled for specific minions and specific commands. This allows secure
# compartmentalization of commands based on individual minions.

# The configuration uses regular expressions to match minions and then a list
# of regular expressions to match functions. The following will allow the
# minion authenticated as foo.example.com to execute functions from the test
# and pkg modules.
#peer:
#  foo.example.com:
#    - test.*
#    - pkg.*
#
# This will allow all minions to execute all commands:
#peer:
#  .*:
#    - .*
#
# This is not recommended, since it would allow anyone who gets root on any
# single minion to instantly have root on all of the minions!

# Minions can also be allowed to execute runners from the salt master.
# Since executing a runner from the minion could be considered a security risk,
# it needs to be enabled. This setting functions just like the peer setting
# except that it opens up runners instead of module functions.
#
# All peer runner support is turned off by default and must be enabled before
# using. This will enable all peer runners for all minions:
#peer_run:
#  .*:
#    - .*
#
# To enable just the manage.up runner for the minion foo.example.com:
#peer_run:
#  foo.example.com:
#    - manage.up
#
#
#####         Mine settings     #####
#####################################
# Restrict mine.get access from minions. By default any minion has a full access
```

(continues on next page)

(continued from previous page)

```
# to get all mine data from master cache. In acl definion below, only pcre matches
# are allowed.
# mine_get:
#   .*:
#     - .*
#
# The example below enables minion foo.example.com to get 'network.interfaces' mine
# data only, minions web* to get all network.* and disk.* mine data and all other
# minions won't get any mine data.
# mine_get:
#   foo.example.com:
#     - network.interfaces
#   web.*:
#     - network.*
#     - disk.*


#####         Logging settings       #####
##########################################
# The location of the master log file
# The master log can be sent to a regular file, local path name, or network
# location. Remote logging works best when configured to use rsyslogd(8) (e.g.:
# ``file:///dev/log``), with rsyslogd(8) configured for network logging. The URI
# format is: <file|udp|tcp>://<host|socketpath>:<port-if-required>/<log-facility>
#log_file: /var/log/salt/master
#log_file: file:///dev/log
#log_file: udp://loghost:10514

#log_file: /var/log/salt/master
#key_logfile: /var/log/salt/key

# The level of messages to send to the console.
# One of 'garbage', 'trace', 'debug', info', 'warning', 'error', 'critical'.
#
# The following log levels are considered INSECURE and may log sensitive data:
# ['garbage', 'trace', 'debug']
#
#log_level: warning

# The level of messages to send to the log file.
# One of 'garbage', 'trace', 'debug', 'info', 'warning', 'error', 'critical'.
# If using 'log_granular_levels' this must be set to the highest desired level.
#log_level_logfile: warning

# The date and time format used in log messages. Allowed date/time formatting
# can be seen here: http://docs.python.org/library/time.html#time.strftime
#log_datefmt: '%H:%M:%S'
#log_datefmt_logfile: '%Y-%m-%d %H:%M:%S'

# The format of the console logging messages. Allowed formatting options can
# be seen here: http://docs.python.org/library/logging.html#logrecord-attributes
#
# Console log colors are specified by these additional formatters:
#
# %(colorlevel)s
# %(colorname)s
```

(continues on next page)

(continued from previous page)

```
# %(colorprocess)s
# %(colormsg)s
#
# Since it is desirable to include the surrounding brackets, '[' and ']', in
# the coloring of the messages, these color formatters also include padding as
# well.  Color LogRecord attributes are only available for console logging.
#
#log_fmt_console: '%(colorlevel)s %(colormsg)s'
#log_fmt_console: '[%(levelname)-8s] %(message)s'
#
#log_fmt_logfile: '%(asctime)s,%(msecs)03d [%(name)-17s][%(levelname)-8s] %(message)s'

# This can be used to control logging levels more specificically.  This
# example sets the main salt library at the 'warning' level, but sets
# 'salt.modules' to log at the 'debug' level:
#   log_granular_levels:
#     'salt': 'warning'
#     'salt.modules': 'debug'
#
#log_granular_levels: {}


#####         Node Groups           ######
##########################################
# Node groups allow for logical groupings of minion nodes. A group consists of
# a group name and a compound target. Nodgroups can reference other nodegroups
# with 'N@' classifier. Ensure that you do not have circular references.
#
#nodegroups:
#  group1: 'L@foo.domain.com,bar.domain.com,baz.domain.com or bl*.domain.com'
#  group2: 'G@os:Debian and foo.domain.com'
#  group3: 'G@os:Debian and N@group1'
#  group4:
#    - 'G@foo:bar'
#    - 'or'
#    - 'G@foo:baz'


#####     Range Cluster settings     #####
##########################################
# The range server (and optional port) that serves your cluster information
# https://github.com/ytoolshed/range/wiki/%22yamlfile%22-module-file-spec
#
#range_server: range:80


#####  Windows Software Repo settings #####
###########################################
# Location of the repo on the master:
#winrepo_dir_ng: '/srv/salt/win/repo-ng'
#
# List of git repositories to include with the local repo:
#winrepo_remotes_ng:
#  - 'https://github.com/saltstack/salt-winrepo-ng.git'
```

(continues on next page)

(continued from previous page)

```
#####  Windows Software Repo settings - Pre 2015.8 #####
########################################################
# Legacy repo settings for pre-2015.8 Windows minions.
#
# Location of the repo on the master:
#winrepo_dir: '/srv/salt/win/repo'
#
# Location of the master's repo cache file:
#winrepo_mastercachefile: '/srv/salt/win/repo/winrepo.p'
#
# List of git repositories to include with the local repo:
#winrepo_remotes:
#  - 'https://github.com/saltstack/salt-winrepo.git'

# The refspecs fetched by winrepo remotes
#winrepo_refspecs:
#  - '+refs/heads/*:refs/remotes/origin/*'
#  - '+refs/tags/*:refs/tags/*'
#

#####      Returner settings          ######
############################################
# Which returner(s) will be used for minion's result:
#return: mysql


######    Miscellaneous  settings     ######
############################################
# Default match type for filtering events tags: startswith, endswith, find, regex,
↪fnmatch
#event_match_type: startswith

# Save runner returns to the job cache
#runner_returns: True

# Permanently include any available Python 3rd party modules into thin and minimal
↪Salt
# when they are generated for Salt-SSH or other purposes.
# The modules should be named by the names they are actually imported inside the
↪Python.
# The value of the parameters can be either one module or a comma separated list of
↪them.
#thin_extra_mods: foo,bar
#min_extra_mods: foo,bar,baz


######      Keepalive settings        ######
############################################
# Warning: Failure to set TCP keepalives on the salt-master can result in
# not detecting the loss of a minion when the connection is lost or when
# it's host has been terminated without first closing the socket.
# Salt's Presence System depends on this connection status to know if a minion
# is "present".
# ZeroMQ now includes support for configuring SO_KEEPALIVE if supported by
# the OS. If connections between the minion and the master pass through
# a state tracking device such as a firewall or VPN gateway, there is
```

(continues on next page)

(continued from previous page)

```
# the risk that it could tear down the connection the master and minion
# without informing either party that their connection has been taken away.
# Enabling TCP Keepalives prevents this from happening.

# Overall state of TCP Keepalives, enable (1 or True), disable (0 or False)
# or leave to the OS defaults (-1), on Linux, typically disabled. Default True,␣
↪enabled.
#tcp_keepalive: True

# How long before the first keepalive should be sent in seconds. Default 300
# to send the first keepalive after 5 minutes, OS default (-1) is typically 7200␣
↪seconds
# on Linux see /proc/sys/net/ipv4/tcp_keepalive_time.
#tcp_keepalive_idle: 300

# How many lost probes are needed to consider the connection lost. Default -1
# to use OS defaults, typically 9 on Linux, see /proc/sys/net/ipv4/tcp_keepalive_
↪probes.
#tcp_keepalive_cnt: -1

# How often, in seconds, to send keepalives after the first one. Default -1 to
# use OS defaults, typically 75 seconds on Linux, see
# /proc/sys/net/ipv4/tcp_keepalive_intvl.
#tcp_keepalive_intvl: -1


#####         NetAPI settings          #####
############################################
# Allow the raw_shell parameter to be used when calling Salt SSH client via API
#netapi_allow_raw_shell: True
```

3.4.2 Example minion configuration file

```
##### Primary configuration settings #####
##########################################
# This configuration file is used to manage the behavior of the Salt Minion.
# With the exception of the location of the Salt Master Server, values that are
# commented out but have an empty line after the comment are defaults that need
# not be set in the config. If there is no blank line after the comment, the
# value is presented as an example and is not the default.

# Per default the minion will automatically include all config files
# from minion.d/*.conf (minion.d is a directory in the same directory
# as the main minion config file).
#default_include: minion.d/*.conf

# Set the location of the salt master server. If the master server cannot be
# resolved, then the minion will fail to start.
#master: salt

# Set http proxy information for the minion when doing requests
#proxy_host:
#proxy_port:
#proxy_username:
#proxy_password:
```

(continues on next page)

(continued from previous page)

```
# List of hosts to bypass HTTP proxy. This key does nothing unless proxy_host etc is
# configured, it does not support any kind of wildcards.
#no_proxy: []

# If multiple masters are specified in the 'master' setting, the default behavior
# is to always try to connect to them in the order they are listed. If random_master
# is set to True, the order will be randomized upon Minion startup instead. This can
# be helpful in distributing the load of many minions executing salt-call requests,
# for example, from a cron job. If only one master is listed, this setting is ignored
# and a warning will be logged.
#random_master: False

# NOTE: Deprecated in Salt 2019.2.0. Use 'random_master' instead.
#master_shuffle: False

# Minions can connect to multiple masters simultaneously (all masters
# are "hot"), or can be configured to failover if a master becomes
# unavailable.  Multiple hot masters are configured by setting this
# value to "str".  Failover masters can be requested by setting
# to "failover".  MAKE SURE TO SET master_alive_interval if you are
# using failover.
# Setting master_type to 'disable' let's you have a running minion (with engines and
# beacons) without a master connection
# master_type: str

# Poll interval in seconds for checking if the master is still there.  Only
# respected if master_type above is "failover". To disable the interval entirely,
# set the value to -1. (This may be necessary on machines which have high numbers
# of TCP connections, such as load balancers.)
# master_alive_interval: 30

# If the minion is in multi-master mode and the master_type configuration option
# is set to "failover", this setting can be set to "True" to force the minion
# to fail back to the first master in the list if the first master is back online.
#master_failback: False

# If the minion is in multi-master mode, the "master_type" configuration is set to
# "failover", and the "master_failback" option is enabled, the master failback
# interval can be set to ping the top master with this interval, in seconds.
#master_failback_interval: 0

# Set whether the minion should connect to the master via IPv6:
#ipv6: False

# Set the number of seconds to wait before attempting to resolve
# the master hostname if name resolution fails. Defaults to 30 seconds.
# Set to zero if the minion should shutdown and not retry.
# retry_dns: 30

# Set the number of times to attempt to resolve
# the master hostname if name resolution fails. Defaults to None,
# which will attempt the resolution indefinitely.
# retry_dns_count: 3

# Set the port used by the master reply and authentication server.
```

(continues on next page)

(continued from previous page)

```
#master_port: 4506

# The user to run salt.
#user: root

# The user to run salt remote execution commands as via sudo. If this option is
# enabled then sudo will be used to change the active user executing the remote
# command. If enabled the user will need to be allowed access via the sudoers
# file for the user that the salt minion is configured to run as. The most
# common option would be to use the root user. If this option is set the user
# option should also be set to a non-root user. If migrating from a root minion
# to a non root minion the minion cache should be cleared and the minion pki
# directory will need to be changed to the ownership of the new user.
#sudo_user: root

# Specify the location of the daemon process ID file.
#pidfile: /var/run/salt-minion.pid

# The root directory prepended to these options: pki_dir, cachedir, log_file,
# sock_dir, pidfile.
#root_dir: /

# The path to the minion's configuration file.
#conf_file: /etc/salt/minion

# The directory to store the pki information in
#pki_dir: /etc/salt/pki/minion

# Explicitly declare the id for this minion to use, if left commented the id
# will be the hostname as returned by the python call: socket.getfqdn()
# Since salt uses detached ids it is possible to run multiple minions on the
# same machine but with different ids, this can be useful for salt compute
# clusters.
#id:

# Cache the minion id to a file when the minion's id is not statically defined
# in the minion config. Defaults to "True". This setting prevents potential
# problems when automatic minion id resolution changes, which can cause the
# minion to lose connection with the master. To turn off minion id caching,
# set this config to ``False``.
#minion_id_caching: True

# Append a domain to a hostname in the event that it does not exist.  This is
# useful for systems where socket.getfqdn() does not actually result in a
# FQDN (for instance, Solaris).
#append_domain:

# Custom static grains for this minion can be specified here and used in SLS
# files just like all other grains. This example sets 4 custom grains, with
# the 'roles' grain having two values that can be matched against.
#grains:
#  roles:
#    - webserver
#    - memcache
#  deployment: datacenter4
#  cabinet: 13
```

(continues on next page)

(continued from previous page)

```
#  cab_u: 14-15
#
# Where cache data goes.
# This data may contain sensitive data and should be protected accordingly.
#cachedir: /var/cache/salt/minion

# Append minion_id to these directories.  Helps with
# multiple proxies and minions running on the same machine.
# Allowed elements in the list: pki_dir, cachedir, extension_modules
# Normally not needed unless running several proxies and/or minions on the same
↪machine
# Defaults to ['cachedir'] for proxies, [] (empty list) for regular minions
#append_minionid_config_dirs:

# Verify and set permissions on configuration directories at startup.
#verify_env: True

# The minion can locally cache the return data from jobs sent to it, this
# can be a good way to keep track of jobs the minion has executed
# (on the minion side). By default this feature is disabled, to enable, set
# cache_jobs to True.
#cache_jobs: False

# Set the directory used to hold unix sockets.
#sock_dir: /var/run/salt/minion

# The minion can take a while to start up when lspci and/or dmidecode is used
# to populate the grains for the minion. Set this to False if you do not need
# GPU hardware grains for your minion.
# enable_gpu_grains: True

# Set the default outputter used by the salt-call command. The default is
# "nested".
#output: nested

# To set a list of additional directories to search for salt outputters, set the
# outputter_dirs option.
#outputter_dirs: []

# By default output is colored. To disable colored output, set the color value
# to False.
#color: True

# Do not strip off the colored output from nested results and state outputs
# (true by default).
# strip_colors: False

# Backup files that are replaced by file.managed and file.recurse under
# 'cachedir'/file_backup relative to their original location and appended
# with a timestamp. The only valid setting is "minion". Disabled by default.
#
# Alternatively this can be specified for each file in state files:
# /etc/ssh/sshd_config:
#   file.managed:
#     - source: salt://ssh/sshd_config
#     - backup: minion
```

(continues on next page)

(continued from previous page)

```
#
#backup_mode: minion

# When waiting for a master to accept the minion's public key, salt will
# continuously attempt to reconnect until successful. This is the time, in
# seconds, between those reconnection attempts.
#acceptance_wait_time: 10

# If this is nonzero, the time between reconnection attempts will increase by
# acceptance_wait_time seconds per iteration, up to this maximum. If this is
# set to zero, the time between reconnection attempts will stay constant.
#acceptance_wait_time_max: 0

# If the master rejects the minion's public key, retry instead of exiting.
# Rejected keys will be handled the same as waiting on acceptance.
#rejected_retry: False

# When the master key changes, the minion will try to re-auth itself to receive
# the new master key. In larger environments this can cause a SYN flood on the
# master because all minions try to re-auth immediately. To prevent this and
# have a minion wait for a random amount of time, use this optional parameter.
# The wait-time will be a random number of seconds between 0 and the defined value.
#random_reauth_delay: 60


# To avoid overloading a master when many minions startup at once, a randomized
# delay may be set to tell the minions to wait before connecting to the master.
# This value is the number of seconds to choose from for a random number. For
# example, setting this value to 60 will choose a random number of seconds to delay
# on startup between zero seconds and sixty seconds. Setting to '0' will disable
# this feature.
#random_startup_delay: 0

# When waiting for a master to accept the minion's public key, salt will
# continuously attempt to reconnect until successful. This is the timeout value,
# in seconds, for each individual attempt. After this timeout expires, the minion
# will wait for acceptance_wait_time seconds before trying again. Unless your master
# is under unusually heavy load, this should be left at the default.
#auth_timeout: 60

# Number of consecutive SaltReqTimeoutError that are acceptable when trying to
# authenticate.
#auth_tries: 7

# The number of attempts to connect to a master before giving up.
# Set this to -1 for unlimited attempts. This allows for a master to have
# downtime and the minion to reconnect to it later when it comes back up.
# In 'failover' mode, it is the number of attempts for each set of masters.
# In this mode, it will cycle through the list of masters for each attempt.
#
# This is different than auth_tries because auth_tries attempts to
# retry auth attempts with a single master. auth_tries is under the
# assumption that you can connect to the master but not gain
# authorization from it. master_tries will still cycle through all
# the masters in a given try, so it is appropriate if you expect
# occasional downtime from the master(s).
```

(continues on next page)

(continued from previous page)

```
#master_tries: 1

# If authentication fails due to SaltReqTimeoutError during a ping_interval,
# cause sub minion process to restart.
#auth_safemode: False

# Ping Master to ensure connection is alive (minutes).
#ping_interval: 0

# To auto recover minions if master changes IP address (DDNS)
#    auth_tries: 10
#    auth_safemode: False
#    ping_interval: 2
#
# Minions won't know master is missing until a ping fails. After the ping fail,
# the minion will attempt authentication and likely fails out and cause a restart.
# When the minion restarts it will resolve the masters IP and attempt to reconnect.

# If you don't have any problems with syn-floods, don't bother with the
# three recon_* settings described below, just leave the defaults!
#
# The ZeroMQ pull-socket that binds to the masters publishing interface tries
# to reconnect immediately, if the socket is disconnected (for example if
# the master processes are restarted). In large setups this will have all
# minions reconnect immediately which might flood the master (the ZeroMQ-default
# is usually a 100ms delay). To prevent this, these three recon_* settings
# can be used.
# recon_default: the interval in milliseconds that the socket should wait before
#                trying to reconnect to the master (1000ms = 1 second)
#
# recon_max: the maximum time a socket should wait. each interval the time to wait
#            is calculated by doubling the previous time. if recon_max is reached,
#            it starts again at recon_default. Short example:
#
#            reconnect 1: the socket will wait 'recon_default' milliseconds
#            reconnect 2: 'recon_default' * 2
#            reconnect 3: ('recon_default' * 2) * 2
#            reconnect 4: value from previous interval * 2
#            reconnect 5: value from previous interval * 2
#            reconnect x: if value >= recon_max, it starts again with recon_default
#
# recon_randomize: generate a random wait time on minion start. The wait time will
#                  be a random value between recon_default and recon_default +
#                  recon_max. Having all minions reconnect with the same recon_default
#                  and recon_max value kind of defeats the purpose of being able to
#                  change these settings. If all minions have the same values and your
#                  setup is quite large (several thousand minions), they will still
#                  flood the master. The desired behavior is to have timeframe within
#                  all minions try to reconnect.
#
# Example on how to use these settings. The goal: have all minions reconnect within a
# 60 second timeframe on a disconnect.
# recon_default: 1000
# recon_max: 59000
# recon_randomize: True
#
```

(continues on next page)

(continued from previous page)

```
# Each minion will have a randomized reconnect value between 'recon_default'
# and 'recon_default + recon_max', which in this example means between 1000ms
# 60000ms (or between 1 and 60 seconds). The generated random-value will be
# doubled after each attempt to reconnect. Lets say the generated random
# value is 11 seconds (or 11000ms).
# reconnect 1: wait 11 seconds
# reconnect 2: wait 22 seconds
# reconnect 3: wait 33 seconds
# reconnect 4: wait 44 seconds
# reconnect 5: wait 55 seconds
# reconnect 6: wait time is bigger than 60 seconds (recon_default + recon_max)
# reconnect 7: wait 11 seconds
# reconnect 8: wait 22 seconds
# reconnect 9: wait 33 seconds
# reconnect x: etc.
#
# In a setup with ~6000 thousand hosts these settings would average the reconnects
# to about 100 per second and all hosts would be reconnected within 60 seconds.
# recon_default: 100
# recon_max: 5000
# recon_randomize: False
#
#
# The loop_interval sets how long in seconds the minion will wait between
# evaluating the scheduler and running cleanup tasks.  This defaults to 1
# second on the minion scheduler.
#loop_interval: 1

# Some installations choose to start all job returns in a cache or a returner
# and forgo sending the results back to a master. In this workflow, jobs
# are most often executed with --async from the Salt CLI and then results
# are evaluated by examining job caches on the minions or any configured returners.
# WARNING: Setting this to False will **disable** returns back to the master.
#pub_ret: True


# The grains can be merged, instead of overridden, using this option.
# This allows custom grains to defined different subvalues of a dictionary
# grain. By default this feature is disabled, to enable set grains_deep_merge
# to ``True``.
#grains_deep_merge: False

# The grains_refresh_every setting allows for a minion to periodically check
# its grains to see if they have changed and, if so, to inform the master
# of the new grains. This operation is moderately expensive, therefore
# care should be taken not to set this value too low.
#
# Note: This value is expressed in __minutes__!
#
# A value of 10 minutes is a reasonable default.
#
# If the value is set to zero, this check is disabled.
#grains_refresh_every: 1

# Cache grains on the minion. Default is False.
#grains_cache: False
```

(continues on next page)

(continued from previous page)

```
# Cache rendered pillar data on the minion. Default is False.
# This may cause 'cachedir'/pillar to contain sensitive data that should be
# protected accordingly.
#minion_pillar_cache: False

# Grains cache expiration, in seconds. If the cache file is older than this
# number of seconds then the grains cache will be dumped and fully re-populated
# with fresh data. Defaults to 5 minutes. Will have no effect if 'grains_cache'
# is not enabled.
# grains_cache_expiration: 300

# Determines whether or not the salt minion should run scheduled mine updates.
# Defaults to "True". Set to "False" to disable the scheduled mine updates
# (this essentially just does not add the mine update function to the minion's
# scheduler).
#mine_enabled: True

# Determines whether or not scheduled mine updates should be accompanied by a job
# return for the job cache. Defaults to "False". Set to "True" to include job
# returns in the job cache for mine updates.
#mine_return_job: False

# Example functions that can be run via the mine facility
# NO mine functions are established by default.
# Note these can be defined in the minion's pillar as well.
#mine_functions:
#  test.ping: []
#  network.ip_addrs:
#    interface: eth0
#    cidr: '10.0.0.0/8'

# The number of minutes between mine updates.
#mine_interval: 60

# Windows platforms lack posix IPC and must rely on slower TCP based inter-
# process communications.  ipc_mode is set to 'tcp' on such systems.
#ipc_mode: ipc

# Overwrite the default tcp ports used by the minion when ipc_mode is set to 'tcp'
#tcp_pub_port: 4510
#tcp_pull_port: 4511

# Passing very large events can cause the minion to consume large amounts of
# memory. This value tunes the maximum size of a message allowed onto the
# minion event bus. The value is expressed in bytes.
#max_event_size: 1048576

# When a minion starts up it sends a notification on the event bus with a tag
# that looks like this: `salt/minion/<minion_id>/start`. For historical reasons
# the minion also sends a similar event with an event tag like this:
# `minion_start`. This duplication can cause a lot of clutter on the event bus
# when there are many minions. Set `enable_legacy_startup_events: False` in the
# minion config to ensure only the `salt/minion/<minion_id>/start` events are
# sent. Beginning with the `Sodium` Salt release this option will default to
# `False`
```

(continues on next page)

(continued from previous page)

```
#enable_legacy_startup_events: True

# To detect failed master(s) and fire events on connect/disconnect, set
# master_alive_interval to the number of seconds to poll the masters for
# connection events.
#
#master_alive_interval: 30

# The minion can include configuration from other files. To enable this,
# pass a list of paths to this option. The paths can be either relative or
# absolute; if relative, they are considered to be relative to the directory
# the main minion configuration file lives in (this file). Paths can make use
# of shell-style globbing. If no files are matched by a path passed to this
# option then the minion will log a warning message.
#
# Include a config file from some other path:
# include: /etc/salt/extra_config
#
# Include config from several files and directories:
#include:
#  - /etc/salt/extra_config
#  - /etc/roles/webserver

# The syndic minion can verify that it is talking to the correct master via the
# key fingerprint of the higher-level master with the "syndic_finger" config.
#syndic_finger: ''
#
#
#
#####   Minion module management     #####
##########################################
# Disable specific modules. This allows the admin to limit the level of
# access the master has to the minion.  The default here is the empty list,
# below is an example of how this needs to be formatted in the config file
#disable_modules:
#  - cmdmod
#  - test
#disable_returners: []

# This is the reverse of disable_modules.  The default, like disable_modules, is the
↪empty list,
# but if this option is set to *anything* then *only* those modules will load.
# Note that this is a very large hammer and it can be quite difficult to keep the
↪minion working
# the way you think it should since Salt uses many modules internally itself.  At a
↪bare minimum
# you need the following enabled or else the minion won't start.
#whitelist_modules:
#  - cmdmod
#  - test
#  - config

# Modules can be loaded from arbitrary paths. This enables the easy deployment
# of third party modules. Modules for returners and minions can be loaded.
# Specify a list of extra directories to search for minion modules and
# returners. These paths must be fully qualified!
```

(continues on next page)

(continued from previous page)

```
#module_dirs: []
#returner_dirs: []
#states_dirs: []
#render_dirs: []
#utils_dirs: []
#
# A module provider can be statically overwritten or extended for the minion
# via the providers option, in this case the default module will be
# overwritten by the specified module. In this example the pkg module will
# be provided by the yumpkg5 module instead of the system default.
#providers:
#  pkg: yumpkg5
#
# Enable Cython modules searching and loading. (Default: False)
#cython_enable: False
#
# Specify a max size (in bytes) for modules on import. This feature is currently
# only supported on *nix operating systems and requires psutil.
# modules_max_memory: -1


#####    State Management Settings    #####
###########################################
# The default renderer to use in SLS files. This is configured as a
# pipe-delimited expression. For example, jinja|yaml will first run jinja
# templating on the SLS file, and then load the result as YAML. This syntax is
# documented in further depth at the following URL:
#
# https://docs.saltstack.com/en/latest/ref/renderers/#composing-renderers
#
# NOTE: The "shebang" prefix (e.g. "#!jinja|yaml") described in the
# documentation linked above is for use in an SLS file to override the default
# renderer, it should not be used when configuring the renderer here.
#
#renderer: jinja|yaml
#
# The failhard option tells the minions to stop immediately after the first
# failure detected in the state execution. Defaults to False.
#failhard: False
#
# Reload the modules prior to a highstate run.
#autoload_dynamic_modules: True
#
# clean_dynamic_modules keeps the dynamic modules on the minion in sync with
# the dynamic modules on the master, this means that if a dynamic module is
# not on the master it will be deleted from the minion. By default, this is
# enabled and can be disabled by changing this value to False.
#clean_dynamic_modules: True
#
# Normally, the minion is not isolated to any single environment on the master
# when running states, but the environment can be isolated on the minion side
# by statically setting it. Remember that the recommended way to manage
# environments is to isolate via the top file.
#environment: None
#
# Isolates the pillar environment on the minion side. This functions the same
```

(continues on next page)

(continued from previous page)

```
# as the environment setting, but for pillar instead of states.
#pillarenv: None
#
# Set this option to True to force the pillarenv to be the same as the
# effective saltenv when running states. Note that if pillarenv is specified,
# this option will be ignored.
#pillarenv_from_saltenv: False
#
# Set this option to 'True' to force a 'KeyError' to be raised whenever an
# attempt to retrieve a named value from pillar fails. When this option is set
# to 'False', the failed attempt returns an empty string. Default is 'False'.
#pillar_raise_on_missing: False
#
# If using the local file directory, then the state top file name needs to be
# defined, by default this is top.sls.
#state_top: top.sls
#
# Run states when the minion daemon starts. To enable, set startup_states to:
# 'highstate' -- Execute state.highstate
# 'sls' -- Read in the sls_list option and execute the named sls files
# 'top' -- Read top_file option and execute based on that file on the Master
#startup_states: ''
#
# List of states to run when the minion starts up if startup_states is 'sls':
#sls_list:
#  - edit.vim
#  - hyper
#
# List of grains to pass in start event when minion starts up:
#start_event_grains:
#  - machine_id
#  - uuid
#
# Top file to execute if startup_states is 'top':
#top_file: ''

# Automatically aggregate all states that have support for mod_aggregate by
# setting to True. Or pass a list of state module names to automatically
# aggregate just those types.
#
# state_aggregate:
#   - pkg
#
#state_aggregate: False

#####     File Directory Settings    #####
##########################################
# The Salt Minion can redirect all file server operations to a local directory,
# this allows for the same state tree that is on the master to be used if
# copied completely onto the minion. This is a literal copy of the settings on
# the master but used to reference a local directory on the minion.

# Set the file client. The client defaults to looking on the master server for
# files, but can be directed to look at the local file directory setting
# defined below by setting it to "local". Setting a local file_client runs the
# minion in masterless mode.
```

(continues on next page)

(continued from previous page)

```
#file_client: remote

# The file directory works on environments passed to the minion, each environment
# can have multiple root directories, the subdirectories in the multiple file
# roots cannot match, otherwise the downloaded files will not be able to be
# reliably ensured. A base environment is required to house the top file.
# Example:
# file_roots:
#   base:
#     - /srv/salt/
#   dev:
#     - /srv/salt/dev/services
#     - /srv/salt/dev/states
#   prod:
#     - /srv/salt/prod/services
#     - /srv/salt/prod/states
#
#file_roots:
#  base:
#    - /srv/salt

# Uncomment the line below if you do not want the file_server to follow
# symlinks when walking the filesystem tree. This is set to True
# by default. Currently this only applies to the default roots
# fileserver_backend.
#fileserver_followsymlinks: False
#
# Uncomment the line below if you do not want symlinks to be
# treated as the files they are pointing to. By default this is set to
# False. By uncommenting the line below, any detected symlink while listing
# files on the Master will not be returned to the Minion.
#fileserver_ignoresymlinks: True
#
# By default, the Salt fileserver recurses fully into all defined environments
# to attempt to find files. To limit this behavior so that the fileserver only
# traverses directories with SLS files and special Salt directories like _modules,
# enable the option below. This might be useful for installations where a file root
# has a very large number of files and performance is negatively impacted. Default
# is False.
#fileserver_limit_traversal: False

# The hash_type is the hash to use when discovering the hash of a file on
# the local fileserver. The default is sha256, but md5, sha1, sha224, sha384
# and sha512 are also supported.
#
# WARNING: While md5 and sha1 are also supported, do not use them due to the
# high chance of possible collisions and thus security breach.
#
# Warning: Prior to changing this value, the minion should be stopped and all
# Salt caches should be cleared.
#hash_type: sha256

# The Salt pillar is searched for locally if file_client is set to local. If
# this is the case, and pillar data is defined, then the pillar_roots need to
# also be configured on the minion:
#pillar_roots:
```

(continues on next page)

(continued from previous page)

```
#  base:
#    - /srv/pillar

# Set a hard-limit on the size of the files that can be pushed to the master.
# It will be interpreted as megabytes. Default: 100
#file_recv_max_size: 100
#
#
######        Security settings       #####
############################################
# Enable "open mode", this mode still maintains encryption, but turns off
# authentication, this is only intended for highly secure environments or for
# the situation where your keys end up in a bad state. If you run in open mode
# you do so at your own risk!
#open_mode: False

# The size of key that should be generated when creating new keys.
#keysize: 2048

# Enable permissive access to the salt keys.  This allows you to run the
# master or minion as root, but have a non-root group be given access to
# your pki_dir.  To make the access explicit, root must belong to the group
# you've given access to. This is potentially quite insecure.
#permissive_pki_access: False

# The state_verbose and state_output settings can be used to change the way
# state system data is printed to the display. By default all data is printed.
# The state_verbose setting can be set to True or False, when set to False
# all data that has a result of True and no changes will be suppressed.
#state_verbose: True

# The state_output setting controls which results will be output full multi line
# full, terse - each state will be full/terse
# mixed - only states with errors will be full
# changes - states with changes and errors will be full
# full_id, mixed_id, changes_id and terse_id are also allowed;
# when set, the state ID will be used as name in the output
#state_output: full

# The state_output_diff setting changes whether or not the output from
# successful states is returned. Useful when even the terse output of these
# states is cluttering the logs. Set it to True to ignore them.
#state_output_diff: False

# The state_output_profile setting changes whether profile information
# will be shown for each state run.
#state_output_profile: True

# Fingerprint of the master public key to validate the identity of your Salt master
# before the initial key exchange. The master fingerprint can be found by running
# "salt-key -f master.pub" on the Salt master.
#master_finger: ''

# Use TLS/SSL encrypted connection between master and minion.
# Can be set to a dictionary containing keyword arguments corresponding to Python's
# 'ssl.wrap_socket' method.
```

(continues on next page)

(continued from previous page)

```
# Default is None.
#ssl:
#    keyfile: <path_to_keyfile>
#    certfile: <path_to_certfile>
#    ssl_version: PROTOCOL_TLSv1_2

# Grains to be sent to the master on authentication to check if the minion's key
# will be accepted automatically. Needs to be configured on the master.
#autosign_grains:
#  - uuid
#  - server_id


######        Reactor Settings        #####
###########################################
# Define a salt reactor. See https://docs.saltstack.com/en/latest/topics/reactor/
#reactor: []

#Set the TTL for the cache of the reactor configuration.
#reactor_refresh_interval: 60

#Configure the number of workers for the runner/wheel in the reactor.
#reactor_worker_threads: 10

#Define the queue size for workers in the reactor.
#reactor_worker_hwm: 10000


######         Thread settings        #####
###########################################
# Disable multiprocessing support, by default when a minion receives a
# publication a new process is spawned and the command is executed therein.
#
# WARNING: Disabling multiprocessing may result in substantial slowdowns
# when processing large pillars. See https://github.com/saltstack/salt/issues/38758
# for a full explanation.
#multiprocessing: True

# Limit the maximum amount of processes or threads created by salt-minion.
# This is useful to avoid resource exhaustion in case the minion receives more
# publications than it is able to handle, as it limits the number of spawned
# processes or threads. -1 is the default and disables the limit.
#process_count_max: -1


#####         Logging settings       #####
##########################################
# The location of the minion log file
# The minion log can be sent to a regular file, local path name, or network
# location. Remote logging works best when configured to use rsyslogd(8) (e.g.:
# ``file:///dev/log``), with rsyslogd(8) configured for network logging. The URI
# format is: <file|udp|tcp>://<host|socketpath>:<port-if-required>/<log-facility>
#log_file: /var/log/salt/minion
#log_file: file:///dev/log
#log_file: udp://loghost:10514
#
```

(continues on next page)

(continued from previous page)

```
#log_file: /var/log/salt/minion
#key_logfile: /var/log/salt/key

# The level of messages to send to the console.
# One of 'garbage', 'trace', 'debug', 'info', 'warning', 'error', 'critical'.
#
# The following log levels are considered INSECURE and may log sensitive data:
# ['garbage', 'trace', 'debug']
#
# Default: 'warning'
#log_level: warning

# The level of messages to send to the log file.
# One of 'garbage', 'trace', 'debug', info', 'warning', 'error', 'critical'.
# If using 'log_granular_levels' this must be set to the highest desired level.
# Default: 'warning'
#log_level_logfile:

# The date and time format used in log messages. Allowed date/time formatting
# can be seen here: http://docs.python.org/library/time.html#time.strftime
#log_datefmt: '%H:%M:%S'
#log_datefmt_logfile: '%Y-%m-%d %H:%M:%S'

# The format of the console logging messages. Allowed formatting options can
# be seen here: http://docs.python.org/library/logging.html#logrecord-attributes
#
# Console log colors are specified by these additional formatters:
#
# %(colorlevel)s
# %(colorname)s
# %(colorprocess)s
# %(colormsg)s
#
# Since it is desirable to include the surrounding brackets, '[' and ']', in
# the coloring of the messages, these color formatters also include padding as
# well.  Color LogRecord attributes are only available for console logging.
#
#log_fmt_console: '%(colorlevel)s %(colormsg)s'
#log_fmt_console: '[%(levelname)-8s] %(message)s'
#
#log_fmt_logfile: '%(asctime)s,%(msecs)03d [%(name)-17s][%(levelname)-8s] %(message)s'

# This can be used to control logging levels more specificically.  This
# example sets the main salt library at the 'warning' level, but sets
# 'salt.modules' to log at the 'debug' level:
#   log_granular_levels:
#     'salt': 'warning'
#     'salt.modules': 'debug'
#
#log_granular_levels: {}

# To diagnose issues with minions disconnecting or missing returns, ZeroMQ
# supports the use of monitor sockets to log connection events. This
# feature requires ZeroMQ 4.0 or higher.
#
# To enable ZeroMQ monitor sockets, set 'zmq_monitor' to 'True' and log at a
```

(continues on next page)

(continued from previous page)

```
# debug level or higher.
#
# A sample log event is as follows:
#
# [DEBUG   ] ZeroMQ event: {'endpoint': 'tcp://127.0.0.1:4505', 'event': 512,
# 'value': 27, 'description': 'EVENT_DISCONNECTED'}
#
# All events logged will include the string 'ZeroMQ event'. A connection event
# should be logged as the minion starts up and initially connects to the
# master. If not, check for debug log level and that the necessary version of
# ZeroMQ is installed.
#
#zmq_monitor: False

# Number of times to try to authenticate with the salt master when reconnecting
# to the master
#tcp_authentication_retries: 5

######      Module configuration      #####
###########################################
# Salt allows for modules to be passed arbitrary configuration data, any data
# passed here in valid yaml format will be passed on to the salt minion modules
# for use. It is STRONGLY recommended that a naming convention be used in which
# the module name is followed by a . and then the value. Also, all top level
# data must be applied via the yaml dict construct, some examples:
#
# You can specify that all modules should run in test mode:
#test: True
#
# A simple value for the test module:
#test.foo: foo
#
# A list for the test module:
#test.bar: [baz,quo]
#
# A dict for the test module:
#test.baz: {spam: sausage, cheese: bread}
#
#
######      Update settings          ######
###########################################
# Using the features in Esky, a salt minion can both run as a frozen app and
# be updated on the fly. These options control how the update process
# (saltutil.update()) behaves.
#
# The url for finding and downloading updates. Disabled by default.
#update_url: False
#
# The list of services to restart after a successful update. Empty by default.
#update_restart_services: []


######      Keepalive settings        ######
############################################
# ZeroMQ now includes support for configuring SO_KEEPALIVE if supported by
# the OS. If connections between the minion and the master pass through
```

(continues on next page)

(continued from previous page)

```
# a state tracking device such as a firewall or VPN gateway, there is
# the risk that it could tear down the connection the master and minion
# without informing either party that their connection has been taken away.
# Enabling TCP Keepalives prevents this from happening.

# Overall state of TCP Keepalives, enable (1 or True), disable (0 or False)
# or leave to the OS defaults (-1), on Linux, typically disabled. Default True,
↪enabled.
#tcp_keepalive: True

# How long before the first keepalive should be sent in seconds. Default 300
# to send the first keepalive after 5 minutes, OS default (-1) is typically 7200
↪seconds
# on Linux see /proc/sys/net/ipv4/tcp_keepalive_time.
#tcp_keepalive_idle: 300

# How many lost probes are needed to consider the connection lost. Default -1
# to use OS defaults, typically 9 on Linux, see /proc/sys/net/ipv4/tcp_keepalive_
↪probes.
#tcp_keepalive_cnt: -1

# How often, in seconds, to send keepalives after the first one. Default -1 to
# use OS defaults, typically 75 seconds on Linux, see
# /proc/sys/net/ipv4/tcp_keepalive_intvl.
#tcp_keepalive_intvl: -1


######   Windows Software settings     ######
############################################
# Location of the repository cache file on the master:
#win_repo_cachefile: 'salt://win/repo/winrepo.p'


######      Returner    settings      ######
############################################
# Default Minion returners. Can be a comma delimited string or a list:
#
#return: mysql
#
#return: mysql,slack,redis
#
#return:
#  - mysql
#  - hipchat
#  - slack


######    Miscellaneous  settings     ######
############################################
# Default match type for filtering events tags: startswith, endswith, find, regex,
↪fnmatch
#event_match_type: startswith
```

3.4.3 Example proxy minion configuration file

```
##### Primary configuration settings #####
##########################################
# This configuration file is used to manage the behavior of all Salt Proxy
# Minions on this host.
# With the exception of the location of the Salt Master Server, values that are
# commented out but have an empty line after the comment are defaults that need
# not be set in the config. If there is no blank line after the comment, the
# value is presented as an example and is not the default.

# Per default the minion will automatically include all config files
# from minion.d/*.conf (minion.d is a directory in the same directory
# as the main minion config file).
#default_include: minion.d/*.conf

# Backwards compatibility option for proxymodules created before 2015.8.2
# This setting will default to 'False' in the 2016.3.0 release
# Setting this to True adds proxymodules to the __opts__ dictionary.
# This breaks several Salt features (basically anything that serializes
# __opts__ over the wire) but retains backwards compatibility.
#add_proxymodule_to_opts: True

# Set the location of the salt master server. If the master server cannot be
# resolved, then the minion will fail to start.
#master: salt

# If a proxymodule has a function called 'grains', then call it during
# regular grains loading and merge the results with the proxy's grains
# dictionary.  Otherwise it is assumed that the module calls the grains
# function in a custom way and returns the data elsewhere
#
# Default to False for 2016.3 and 2016.11. Switch to True for 2017.7.0.
# proxy_merge_grains_in_module: True

# If a proxymodule has a function called 'alive' returning a boolean
# flag reflecting the state of the connection with the remove device,
# when this option is set as True, a scheduled job on the proxy will
# try restarting the connection. The polling frequency depends on the
# next option, 'proxy_keep_alive_interval'. Added in 2017.7.0.
# proxy_keep_alive: True

# The polling interval (in minutes) to check if the underlying connection
# with the remote device is still alive. This option requires
# 'proxy_keep_alive' to be configured as True and the proxymodule to
# implement the 'alive' function. Added in 2017.7.0.
# proxy_keep_alive_interval: 1

# By default, any proxy opens the connection with the remote device when
# initialized. Some proxymodules allow through this option to open/close
# the session per command. This requires the proxymodule to have this
# capability. Please consult the documentation to see if the proxy type
# used can be that flexible. Added in 2017.7.0.
# proxy_always_alive: True

# If multiple masters are specified in the 'master' setting, the default behavior
# is to always try to connect to them in the order they are listed. If random_master
↪is
```

(continues on next page)

(continued from previous page)

```
# set to True, the order will be randomized instead. This can be helpful in
↪distributing
# the load of many minions executing salt-call requests, for example, from a cron job.
# If only one master is listed, this setting is ignored and a warning will be logged.
#random_master: False

# Minions can connect to multiple masters simultaneously (all masters
# are "hot"), or can be configured to failover if a master becomes
# unavailable.  Multiple hot masters are configured by setting this
# value to "str".  Failover masters can be requested by setting
# to "failover".  MAKE SURE TO SET master_alive_interval if you are
# using failover.
# master_type: str

# Poll interval in seconds for checking if the master is still there.  Only
# respected if master_type above is "failover".
# master_alive_interval: 30

# Set whether the minion should connect to the master via IPv6:
#ipv6: False

# Set the number of seconds to wait before attempting to resolve
# the master hostname if name resolution fails. Defaults to 30 seconds.
# Set to zero if the minion should shutdown and not retry.
# retry_dns: 30

# Set the port used by the master reply and authentication server.
#master_port: 4506

# The user to run salt.
#user: root

# Setting sudo_user will cause salt to run all execution modules under an sudo
# to the user given in sudo_user.  The user under which the salt minion process
# itself runs will still be that provided in the user config above, but all
# execution modules run by the minion will be rerouted through sudo.
#sudo_user: saltdev

# Specify the location of the daemon process ID file.
#pidfile: /var/run/salt-minion.pid

# The root directory prepended to these options: pki_dir, cachedir, log_file,
# sock_dir, pidfile.
#root_dir: /

# The directory to store the pki information in
#pki_dir: /etc/salt/pki/minion

# Where cache data goes.
# This data may contain sensitive data and should be protected accordingly.
#cachedir: /var/cache/salt/minion

# Append minion_id to these directories.  Helps with
# multiple proxies and minions running on the same machine.
# Allowed elements in the list: pki_dir, cachedir, extension_modules
# Normally not needed unless running several proxies and/or minions on the same
↪machine
```

(continues on next page)

(continued from previous page)

```
# Defaults to ['cachedir'] for proxies, [] (empty list) for regular minions
# append_minionid_config_dirs:
#   - cachedir



# Verify and set permissions on configuration directories at startup.
#verify_env: True

# The minion can locally cache the return data from jobs sent to it, this
# can be a good way to keep track of jobs the minion has executed
# (on the minion side). By default this feature is disabled, to enable, set
# cache_jobs to True.
#cache_jobs: False

# Set the directory used to hold unix sockets.
#sock_dir: /var/run/salt/minion

# Set the default outputter used by the salt-call command. The default is
# "nested".
#output: nested
#
# By default output is colored. To disable colored output, set the color value
# to False.
#color: True

# Do not strip off the colored output from nested results and state outputs
# (true by default).
# strip_colors: False

# Backup files that are replaced by file.managed and file.recurse under
# 'cachedir'/file_backup relative to their original location and appended
# with a timestamp. The only valid setting is "minion". Disabled by default.
#
# Alternatively this can be specified for each file in state files:
# /etc/ssh/sshd_config:
#   file.managed:
#     - source: salt://ssh/sshd_config
#     - backup: minion
#
#backup_mode: minion

# When waiting for a master to accept the minion's public key, salt will
# continuously attempt to reconnect until successful. This is the time, in
# seconds, between those reconnection attempts.
#acceptance_wait_time: 10

# If this is nonzero, the time between reconnection attempts will increase by
# acceptance_wait_time seconds per iteration, up to this maximum. If this is
# set to zero, the time between reconnection attempts will stay constant.
#acceptance_wait_time_max: 0

# If the master rejects the minion's public key, retry instead of exiting.
# Rejected keys will be handled the same as waiting on acceptance.
#rejected_retry: False
```

(continues on next page)

(continued from previous page)

```
# When the master key changes, the minion will try to re-auth itself to receive
# the new master key. In larger environments this can cause a SYN flood on the
# master because all minions try to re-auth immediately. To prevent this and
# have a minion wait for a random amount of time, use this optional parameter.
# The wait-time will be a random number of seconds between 0 and the defined value.
#random_reauth_delay: 60

# When waiting for a master to accept the minion's public key, salt will
# continuously attempt to reconnect until successful. This is the timeout value,
# in seconds, for each individual attempt. After this timeout expires, the minion
# will wait for acceptance_wait_time seconds before trying again. Unless your master
# is under unusually heavy load, this should be left at the default.
#auth_timeout: 60

# Number of consecutive SaltReqTimeoutError that are acceptable when trying to
# authenticate.
#auth_tries: 7

# If authentication fails due to SaltReqTimeoutError during a ping_interval,
# cause sub minion process to restart.
#auth_safemode: False

# Ping Master to ensure connection is alive (minutes).
#ping_interval: 0

# To auto recover minions if master changes IP address (DDNS)
#    auth_tries: 10
#    auth_safemode: False
#    ping_interval: 90
#
# Minions won't know master is missing until a ping fails. After the ping fail,
# the minion will attempt authentication and likely fails out and cause a restart.
# When the minion restarts it will resolve the masters IP and attempt to reconnect.

# If you don't have any problems with syn-floods, don't bother with the
# three recon_* settings described below, just leave the defaults!
#
# The ZeroMQ pull-socket that binds to the masters publishing interface tries
# to reconnect immediately, if the socket is disconnected (for example if
# the master processes are restarted). In large setups this will have all
# minions reconnect immediately which might flood the master (the ZeroMQ-default
# is usually a 100ms delay). To prevent this, these three recon_* settings
# can be used.
# recon_default: the interval in milliseconds that the socket should wait before
#                trying to reconnect to the master (1000ms = 1 second)
#
# recon_max: the maximum time a socket should wait. each interval the time to wait
#            is calculated by doubling the previous time. if recon_max is reached,
#            it starts again at recon_default. Short example:
#
#            reconnect 1: the socket will wait 'recon_default' milliseconds
#            reconnect 2: 'recon_default' * 2
#            reconnect 3: ('recon_default' * 2) * 2
#            reconnect 4: value from previous interval * 2
#            reconnect 5: value from previous interval * 2
#            reconnect x: if value >= recon_max, it starts again with recon_default
```

(continues on next page)

(continued from previous page)

```
#
# recon_randomize: generate a random wait time on minion start. The wait time will
#                  be a random value between recon_default and recon_default +
#                  recon_max. Having all minions reconnect with the same recon_default
#                  and recon_max value kind of defeats the purpose of being able to
#                  change these settings. If all minions have the same values and your
#                  setup is quite large (several thousand minions), they will still
#                  flood the master. The desired behavior is to have timeframe within
#                  all minions try to reconnect.
#
# Example on how to use these settings. The goal: have all minions reconnect within a
# 60 second timeframe on a disconnect.
# recon_default: 1000
# recon_max: 59000
# recon_randomize: True
#
# Each minion will have a randomized reconnect value between 'recon_default'
# and 'recon_default + recon_max', which in this example means between 1000ms
# 60000ms (or between 1 and 60 seconds). The generated random-value will be
# doubled after each attempt to reconnect. Lets say the generated random
# value is 11 seconds (or 11000ms).
# reconnect 1: wait 11 seconds
# reconnect 2: wait 22 seconds
# reconnect 3: wait 33 seconds
# reconnect 4: wait 44 seconds
# reconnect 5: wait 55 seconds
# reconnect 6: wait time is bigger than 60 seconds (recon_default + recon_max)
# reconnect 7: wait 11 seconds
# reconnect 8: wait 22 seconds
# reconnect 9: wait 33 seconds
# reconnect x: etc.
#
# In a setup with ~6000 thousand hosts these settings would average the reconnects
# to about 100 per second and all hosts would be reconnected within 60 seconds.
# recon_default: 100
# recon_max: 5000
# recon_randomize: False
#
#
# The loop_interval sets how long in seconds the minion will wait between
# evaluating the scheduler and running cleanup tasks. This defaults to a
# sane 60 seconds, but if the minion scheduler needs to be evaluated more
# often lower this value
#loop_interval: 60

# The grains_refresh_every setting allows for a minion to periodically check
# its grains to see if they have changed and, if so, to inform the master
# of the new grains. This operation is moderately expensive, therefore
# care should be taken not to set this value too low.
#
# Note: This value is expressed in __minutes__!
#
# A value of 10 minutes is a reasonable default.
#
# If the value is set to zero, this check is disabled.
#grains_refresh_every: 1
```

(continues on next page)

(continued from previous page)

```
# Cache grains on the minion. Default is False.
#grains_cache: False

# Grains cache expiration, in seconds. If the cache file is older than this
# number of seconds then the grains cache will be dumped and fully re-populated
# with fresh data. Defaults to 5 minutes. Will have no effect if 'grains_cache'
# is not enabled.
# grains_cache_expiration: 300

# Windows platforms lack posix IPC and must rely on slower TCP based inter-
# process communications. Set ipc_mode to 'tcp' on such systems
#ipc_mode: ipc

# Overwrite the default tcp ports used by the minion when in tcp mode
#tcp_pub_port: 4510
#tcp_pull_port: 4511

# Passing very large events can cause the minion to consume large amounts of
# memory. This value tunes the maximum size of a message allowed onto the
# minion event bus. The value is expressed in bytes.
#max_event_size: 1048576

# To detect failed master(s) and fire events on connect/disconnect, set
# master_alive_interval to the number of seconds to poll the masters for
# connection events.
#
#master_alive_interval: 30

# The minion can include configuration from other files. To enable this,
# pass a list of paths to this option. The paths can be either relative or
# absolute; if relative, they are considered to be relative to the directory
# the main minion configuration file lives in (this file). Paths can make use
# of shell-style globbing. If no files are matched by a path passed to this
# option then the minion will log a warning message.
#
# Include a config file from some other path:
# include: /etc/salt/extra_config
#
# Include config from several files and directories:
#include:
#  - /etc/salt/extra_config
#  - /etc/roles/webserver
#
#
#
#####   Minion module management     #####
##########################################
# Disable specific modules. This allows the admin to limit the level of
# access the master has to the minion.
#disable_modules: [cmd,test]
#disable_returners: []
#
# Modules can be loaded from arbitrary paths. This enables the easy deployment
# of third party modules. Modules for returners and minions can be loaded.
# Specify a list of extra directories to search for minion modules and
```

(continues on next page)

(continued from previous page)

```
# returners. These paths must be fully qualified!
#module_dirs: []
#returner_dirs: []
#states_dirs: []
#render_dirs: []
#utils_dirs: []
#
# A module provider can be statically overwritten or extended for the minion
# via the providers option, in this case the default module will be
# overwritten by the specified module. In this example the pkg module will
# be provided by the yumpkg5 module instead of the system default.
#providers:
#  pkg: yumpkg5
#
# Enable Cython modules searching and loading. (Default: False)
#cython_enable: False
#
# Specify a max size (in bytes) for modules on import. This feature is currently
# only supported on *nix operating systems and requires psutil.
# modules_max_memory: -1


#####    State Management Settings    #####
###########################################
# The default renderer to use in SLS files. This is configured as a
# pipe-delimited expression. For example, jinja|yaml will first run jinja
# templating on the SLS file, and then load the result as YAML. This syntax is
# documented in further depth at the following URL:
#
# https://docs.saltstack.com/en/latest/ref/renderers/#composing-renderers
#
# NOTE: The "shebang" prefix (e.g. "#!jinja|yaml") described in the
# documentation linked above is for use in an SLS file to override the default
# renderer, it should not be used when configuring the renderer here.
#
#renderer: jinja|yaml
#
# The failhard option tells the minions to stop immediately after the first
# failure detected in the state execution. Defaults to False.
#failhard: False
#
# Reload the modules prior to a highstate run.
#autoload_dynamic_modules: True
#
# clean_dynamic_modules keeps the dynamic modules on the minion in sync with
# the dynamic modules on the master, this means that if a dynamic module is
# not on the master it will be deleted from the minion. By default, this is
# enabled and can be disabled by changing this value to False.
#clean_dynamic_modules: True
#
# Normally, the minion is not isolated to any single environment on the master
# when running states, but the environment can be isolated on the minion side
# by statically setting it. Remember that the recommended way to manage
# environments is to isolate via the top file.
#environment: None
#
```

(continues on next page)

(continued from previous page)

```
# If using the local file directory, then the state top file name needs to be
# defined, by default this is top.sls.
#state_top: top.sls
#
# Run states when the minion daemon starts. To enable, set startup_states to:
# 'highstate' -- Execute state.highstate
# 'sls' -- Read in the sls_list option and execute the named sls files
# 'top' -- Read top_file option and execute based on that file on the Master
#startup_states: ''
#
# List of states to run when the minion starts up if startup_states is 'sls':
#sls_list:
#  - edit.vim
#  - hyper
#
# Top file to execute if startup_states is 'top':
#top_file: ''

# Automatically aggregate all states that have support for mod_aggregate by
# setting to True. Or pass a list of state module names to automatically
# aggregate just those types.
#
# state_aggregate:
#   - pkg
#
#state_aggregate: False

#####     File Directory Settings    #####
##########################################
# The Salt Minion can redirect all file server operations to a local directory,
# this allows for the same state tree that is on the master to be used if
# copied completely onto the minion. This is a literal copy of the settings on
# the master but used to reference a local directory on the minion.

# Set the file client. The client defaults to looking on the master server for
# files, but can be directed to look at the local file directory setting
# defined below by setting it to "local". Setting a local file_client runs the
# minion in masterless mode.
#file_client: remote

# The file directory works on environments passed to the minion, each environment
# can have multiple root directories, the subdirectories in the multiple file
# roots cannot match, otherwise the downloaded files will not be able to be
# reliably ensured. A base environment is required to house the top file.
# Example:
# file_roots:
#   base:
#     - /srv/salt/
#   dev:
#     - /srv/salt/dev/services
#     - /srv/salt/dev/states
#   prod:
#     - /srv/salt/prod/services
#     - /srv/salt/prod/states
#
#file_roots:
```

(continues on next page)

(continued from previous page)

```
#  base:
#    - /srv/salt

# By default, the Salt fileserver recurses fully into all defined environments
# to attempt to find files. To limit this behavior so that the fileserver only
# traverses directories with SLS files and special Salt directories like _modules,
# enable the option below. This might be useful for installations where a file root
# has a very large number of files and performance is negatively impacted. Default
# is False.
#fileserver_limit_traversal: False

# The hash_type is the hash to use when discovering the hash of a file in
# the local fileserver. The default is sha256 but sha224, sha384 and sha512
# are also supported.
#
# WARNING: While md5 and sha1 are also supported, do not use it due to the high chance
# of possible collisions and thus security breach.
#
# WARNING: While md5 is also supported, do not use it due to the high chance
# of possible collisions and thus security breach.
#
# Warning: Prior to changing this value, the minion should be stopped and all
# Salt caches should be cleared.
#hash_type: sha256

# The Salt pillar is searched for locally if file_client is set to local. If
# this is the case, and pillar data is defined, then the pillar_roots need to
# also be configured on the minion:
#pillar_roots:
#  base:
#    - /srv/pillar
#
#
######        Security settings       #####
###########################################
# Enable "open mode", this mode still maintains encryption, but turns off
# authentication, this is only intended for highly secure environments or for
# the situation where your keys end up in a bad state. If you run in open mode
# you do so at your own risk!
#open_mode: False

# Enable permissive access to the salt keys.  This allows you to run the
# master or minion as root, but have a non-root group be given access to
# your pki_dir.  To make the access explicit, root must belong to the group
# you've given access to. This is potentially quite insecure.
#permissive_pki_access: False

# The state_verbose and state_output settings can be used to change the way
# state system data is printed to the display. By default all data is printed.
# The state_verbose setting can be set to True or False, when set to False
# all data that has a result of True and no changes will be suppressed.
#state_verbose: True

# The state_output setting controls which results will be output full multi line
# full, terse - each state will be full/terse
# mixed - only states with errors will be full
```

(continues on next page)

(continued from previous page)

```
# changes - states with changes and errors will be full
# full_id, mixed_id, changes_id and terse_id are also allowed;
# when set, the state ID will be used as name in the output
#state_output: full

# The state_output_diff setting changes whether or not the output from
# successful states is returned. Useful when even the terse output of these
# states is cluttering the logs. Set it to True to ignore them.
#state_output_diff: False

# The state_output_profile setting changes whether profile information
# will be shown for each state run.
#state_output_profile: True

# Fingerprint of the master public key to validate the identity of your Salt master
# before the initial key exchange. The master fingerprint can be found by running
# "salt-key -F master" on the Salt master.
#master_finger: ''


######         Thread settings        #####
###########################################
# Disable multiprocessing support, by default when a minion receives a
# publication a new process is spawned and the command is executed therein.
#multiprocessing: True


#####         Logging settings       #####
##########################################
# The location of the minion log file
# The minion log can be sent to a regular file, local path name, or network
# location. Remote logging works best when configured to use rsyslogd(8) (e.g.:
# ``file:///dev/log``), with rsyslogd(8) configured for network logging. The URI
# format is: <file|udp|tcp>://<host|socketpath>:<port-if-required>/<log-facility>
#log_file: /var/log/salt/minion
#log_file: file:///dev/log
#log_file: udp://loghost:10514
#
#log_file: /var/log/salt/minion
#key_logfile: /var/log/salt/key

# The level of messages to send to the console.
# One of 'garbage', 'trace', 'debug', 'info', 'warning', 'error', 'critical'.
#
# The following log levels are considered INSECURE and may log sensitive data:
# ['garbage', 'trace', 'debug']
#
# Default: 'warning'
#log_level: warning

# The level of messages to send to the log file.
# One of 'garbage', 'trace', 'debug', info', 'warning', 'error', 'critical'.
# If using 'log_granular_levels' this must be set to the highest desired level.
# Default: 'warning'
#log_level_logfile:
```

(continues on next page)

(continued from previous page)

```
# The date and time format used in log messages. Allowed date/time formatting
# can be seen here: http://docs.python.org/library/time.html#time.strftime
#log_datefmt: '%H:%M:%S'
#log_datefmt_logfile: '%Y-%m-%d %H:%M:%S'

# The format of the console logging messages. Allowed formatting options can
# be seen here: http://docs.python.org/library/logging.html#logrecord-attributes
#
# Console log colors are specified by these additional formatters:
#
# %(colorlevel)s
# %(colorname)s
# %(colorprocess)s
# %(colormsg)s
#
# Since it is desirable to include the surrounding brackets, '[' and ']', in
# the coloring of the messages, these color formatters also include padding as
# well.  Color LogRecord attributes are only available for console logging.
#
#log_fmt_console: '%(colorlevel)s %(colormsg)s'
#log_fmt_console: '[%(levelname)-8s] %(message)s'
#
#log_fmt_logfile: '%(asctime)s,%(msecs)03d [%(name)-17s][%(levelname)-8s] %(message)s'

# This can be used to control logging levels more specifically.  This
# example sets the main salt library at the 'warning' level, but sets
# 'salt.modules' to log at the 'debug' level:
#   log_granular_levels:
#     'salt': 'warning'
#     'salt.modules': 'debug'
#
#log_granular_levels: {}

# To diagnose issues with minions disconnecting or missing returns, ZeroMQ
# supports the use of monitor sockets # to log connection events. This
# feature requires ZeroMQ 4.0 or higher.
#
# To enable ZeroMQ monitor sockets, set 'zmq_monitor' to 'True' and log at a
# debug level or higher.
#
# A sample log event is as follows:
#
# [DEBUG   ] ZeroMQ event: {'endpoint': 'tcp://127.0.0.1:4505', 'event': 512,
# 'value': 27, 'description': 'EVENT_DISCONNECTED'}
#
# All events logged will include the string 'ZeroMQ event'. A connection event
# should be logged on the as the minion starts up and initially connects to the
# master. If not, check for debug log level and that the necessary version of
# ZeroMQ is installed.
#
#zmq_monitor: False

######      Module configuration      #####
###########################################
# Salt allows for modules to be passed arbitrary configuration data, any data
# passed here in valid yaml format will be passed on to the salt minion modules
```

(continues on next page)

(continued from previous page)

```
# for use. It is STRONGLY recommended that a naming convention be used in which
# the module name is followed by a . and then the value. Also, all top level
# data must be applied via the yaml dict construct, some examples:
#
# You can specify that all modules should run in test mode:
#test: True
#
# A simple value for the test module:
#test.foo: foo
#
# A list for the test module:
#test.bar: [baz,quo]
#
# A dict for the test module:
#test.baz: {spam: sausage, cheese: bread}
#
#
######      Update settings          ######
###########################################
# Using the features in Esky, a salt minion can both run as a frozen app and
# be updated on the fly. These options control how the update process
# (saltutil.update()) behaves.
#
# The url for finding and downloading updates. Disabled by default.
#update_url: False
#
# The list of services to restart after a successful update. Empty by default.
#update_restart_services: []


######      Keepalive settings        ######
############################################
# ZeroMQ now includes support for configuring SO_KEEPALIVE if supported by
# the OS. If connections between the minion and the master pass through
# a state tracking device such as a firewall or VPN gateway, there is
# the risk that it could tear down the connection the master and minion
# without informing either party that their connection has been taken away.
# Enabling TCP Keepalives prevents this from happening.

# Overall state of TCP Keepalives, enable (1 or True), disable (0 or False)
# or leave to the OS defaults (-1), on Linux, typically disabled. Default True,
↪enabled.
#tcp_keepalive: True

# How long before the first keepalive should be sent in seconds. Default 300
# to send the first keepalive after 5 minutes, OS default (-1) is typically 7200
↪seconds
# on Linux see /proc/sys/net/ipv4/tcp_keepalive_time.
#tcp_keepalive_idle: 300

# How many lost probes are needed to consider the connection lost. Default -1
# to use OS defaults, typically 9 on Linux, see /proc/sys/net/ipv4/tcp_keepalive_
↪probes.
#tcp_keepalive_cnt: -1

# How often, in seconds, to send keepalives after the first one. Default -1 to
```

(continues on next page)

(continued from previous page)

```
# use OS defaults, typically 75 seconds on Linux, see
# /proc/sys/net/ipv4/tcp_keepalive_intvl.
#tcp_keepalive_intvl: -1


######   Windows Software settings    ######
############################################
# Location of the repository cache file on the master:
#win_repo_cachefile: 'salt://win/repo/winrepo.p'


######      Returner  settings        ######
############################################
# Which returner(s) will be used for minion's result:
#return: mysql
```

3.5 Minion Blackout Configuration

New in version 2016.3.0.

Salt supports minion blackouts. When a minion is in blackout mode, all remote execution commands are disabled. This allows production minions to be put ”on hold”, eliminating the risk of an untimely configuration change.

Minion blackouts are configured via a special pillar key, `minion_blackout`. If this key is set to `True`, then the minion will reject all incoming commands, except for `saltutil.refresh_pillar`. (The exception is important, so minions can be brought out of blackout mode)

Salt also supports an explicit whitelist of additional functions that will be allowed during blackout. This is configured with the special pillar key `minion_blackout_whitelist`, which is formed as a list:

```
minion_blackout_whitelist:
 - test.version
 - pillar.get
```

3.6 Access Control System

New in version 0.10.4.

Salt maintains a standard system used to open granular control to non administrative users to execute Salt commands. The access control system has been applied to all systems used to configure access to non administrative control interfaces in Salt.

These interfaces include, the `peer` system, the `external auth` system and the `publisher acl` system.

The access control system mandated a standard configuration syntax used in all of the three aforementioned systems. While this adds functionality to the configuration in 0.10.4, it does not negate the old configuration.

Now specific functions can be opened up to specific minions from specific users in the case of external auth and publisher ACLs, and for specific minions in the case of the peer system.

3.6.1 Publisher ACL system

The salt publisher ACL system is a means to allow system users other than root to have access to execute select salt commands on minions from the master.

The publisher ACL system is configured in the master configuration file via the `publisher_acl` configuration option. Under the `publisher_acl` configuration option the users open to send commands are specified and then a list of the minion functions which will be made available to specified user. Both users and functions could be specified by exact match, shell glob or regular expression. This configuration is much like the *external_auth* configuration:

```
publisher_acl:
  # Allow thatch to execute anything.
  thatch:
    - .*
  # Allow fred to use test and pkg, but only on "web*" minions.
  fred:
    - web*:
      - test.*
      - pkg.*
  # Allow admin and managers to use saltutil module functions
  admin|manager_.*:
    - saltutil.*
  # Allow users to use only my_mod functions on "web*" minions with specific
↪arguments.
  user_.*:
    - web*:
      - 'my_mod.*':
          args:
            - 'a.*'
            - 'b.*'
          kwargs:
            'kwa': 'kwa.*'
            'kwb': 'kwb'
```

Permission Issues

Directories required for `publisher_acl` must be modified to be readable by the users specified:

```
chmod 755 /var/cache/salt /var/cache/salt/master /var/cache/salt/master/jobs /var/run/
↪salt /var/run/salt/master
```

Note: In addition to the changes above you will also need to modify the permissions of /var/log/salt and the existing log file to be writable by the user(s) which will be running the commands. If you do not wish to do this then you must disable logging or Salt will generate errors as it cannot write to the logs as the system users.

If you are upgrading from earlier versions of salt you must also remove any existing user keys and re-start the Salt master:

```
rm /var/cache/salt/.*key
service salt-master restart
```

Whitelist and Blacklist

Salt's authentication systems can be configured by specifying what is allowed using a whitelist, or by specifying what is disallowed using a blacklist. If you specify a whitelist, only specified operations are allowed. If you specify a blacklist, all operations are allowed except those that are blacklisted.

See *publisher_acl* and *publisher_acl_blacklist*.

3.6.2 External Authentication System

Salt's External Authentication System (eAuth) allows for Salt to pass through command authorization to any external authentication system, such as PAM or LDAP.

Note: eAuth using the PAM external auth system requires salt-master to be run as root as this system needs root access to check authentication.

External Authentication System Configuration

The external authentication system allows for specific users to be granted access to execute specific functions on specific minions. Access is configured in the master configuration file and uses the *access control system*:

```
external_auth:
  pam:
    thatch:
      - 'web*':
        - test.*
        - network.*
    steve|admin.*:
      - .*
```

The above configuration allows the user `thatch` to execute functions in the test and network modules on the minions that match the web* target. User `steve` and the users whose logins start with `admin`, are granted unrestricted access to minion commands.

Salt respects the current PAM configuration in place, and uses the 'login' service to authenticate.

Note: The PAM module does not allow authenticating as `root`.

Note: state.sls and state.highstate will return "Failed to authenticate!" if the request timeout is reached. Use -t flag to increase the timeout

---

To allow access to *wheel modules* or *runner modules* the following `@` syntax must be used:

```
external_auth:
  pam:
    thatch:
      - '@wheel'   # to allow access to all wheel modules
      - '@runner'  # to allow access to all runner modules
      - '@jobs'    # to allow access to the jobs runner and/or wheel module
```

Note: The runner/wheel markup is different, since there are no minions to scope the acl to.

Note: Globs will not match wheel or runners! They must be explicitly allowed with @wheel or @runner.

Warning: All users that have external authentication privileges are allowed to run *saltutil.findjob*. Be aware that this could inadvertently expose some data such as minion IDs.

Matching syntax

The structure of the `external_auth` dictionary can take the following shapes. User and function matches are exact matches, shell glob patterns or regular expressions; minion matches are compound targets.

By user:

```
external_auth:
  <eauth backend>:
    <user or group%>:
      - <regex to match function>
```

By user, by minion:

```
external_auth:
  <eauth backend>:
    <user or group%>:
      <minion compound target>:
        - <regex to match function>
```

By user, by runner/wheel:

```
external_auth:
  <eauth backend>:
    <user or group%>:
      <@runner or @wheel>:
        - <regex to match function>
```

By user, by runner+wheel module:

```
external_auth:
  <eauth backend>:
    <user or group%>:
      <@module_name>:
        - <regex to match function without module_name>
```

Groups

To apply permissions to a group of users in an external authentication system, append a `%` to the ID:

```
external_auth:
  pam:
    admins%:
      - '*':
        - 'pkg.*'
```

Limiting by function arguments

Positional arguments or keyword arguments to functions can also be whitelisted.

New in version 2016.3.0.

```
external_auth:
  pam:
    my_user:
      - '*':
        - 'my_mod.*':
            args:
              - 'a.*'
              - 'b.*'
            kwargs:
              'kwa': 'kwa.*'
              'kwb': 'kwb'
      - '@runner':
        - 'runner_mod.*':
            args:
            - 'a.*'
            - 'b.*'
            kwargs:
              'kwa': 'kwa.*'
              'kwb': 'kwb'
```

The rules:

1. The arguments values are matched as regexp.
2. If arguments restrictions are specified the only matched are allowed.
3. If an argument isn't specified any value is allowed.
4. To skip an arg use "everything" regexp `.*`. I.e. if `arg0` and `arg2` should be limited but `arg1` and other arguments could have any value use:

    ```
    args:
      - 'value0'
      - '.*'
      - 'value2'
    ```

Usage

The external authentication system can then be used from the command-line by any user on the same system as the master with the `-a` option:

```
$ salt -a pam web\* test.version
```

The system will ask the user for the credentials required by the authentication system and then publish the command.

Tokens

With external authentication alone, the authentication credentials will be required with every call to Salt. This can be alleviated with Salt tokens.

Tokens are short term authorizations and can be easily created by just adding a `-T` option when authenticating:

```
$ salt -T -a pam web\* test.version
```

Now a token will be created that has an expiration of 12 hours (by default). This token is stored in a file named `salt_token` in the active user's home directory.

Once the token is created, it is sent with all subsequent communications. User authentication does not need to be entered again until the token expires.

Token expiration time can be set in the Salt master config file.

LDAP and Active Directory

Note: LDAP usage requires that you have installed python-ldap.

Salt supports both user and group authentication for LDAP (and Active Directory accessed via its LDAP interface)

OpenLDAP and similar systems

LDAP configuration happens in the Salt master configuration file.

Server configuration values and their defaults:

```
# Server to auth against
auth.ldap.server: localhost

# Port to connect via
auth.ldap.port: 389

# Use TLS when connecting
auth.ldap.tls: False

# Use STARTTLS when connecting
auth.ldap.starttls: False

# LDAP scope level, almost always 2
auth.ldap.scope: 2

# Server specified in URI format
auth.ldap.uri: ''    # Overrides .ldap.server, .ldap.port, .ldap.tls above

# Verify server's TLS certificate
auth.ldap.no_verify: False

# Bind to LDAP anonymously to determine group membership
# Active Directory does not allow anonymous binds without special configuration
# In addition, if auth.ldap.anonymous is True, empty bind passwords are not permitted.
auth.ldap.anonymous: False
```

(continues on next page)

(continued from previous page)

```
# FOR TESTING ONLY, this is a VERY insecure setting.
# If this is True, the LDAP bind password will be ignored and
# access will be determined by group membership alone with
# the group memberships being retrieved via anonymous bind
auth.ldap.auth_by_group_membership_only: False

# Require authenticating user to be part of this Organizational Unit
# This can be blank if your LDAP schema does not use this kind of OU
auth.ldap.groupou: 'Groups'

# Object Class for groups.  An LDAP search will be done to find all groups of this
# class to which the authenticating user belongs.
auth.ldap.groupclass: 'posixGroup'

# Unique ID attribute name for the user
auth.ldap.accountattributename: 'memberUid'

# These are only for Active Directory
auth.ldap.activedirectory: False
auth.ldap.persontype: 'person'

auth.ldap.minion_stripdomains: []

# Redhat Identity Policy Audit
auth.ldap.freeipa: False
```

Authenticating to the LDAP Server

There are two phases to LDAP authentication. First, Salt authenticates to search for a users' Distinguished Name and group membership. The user it authenticates as in this phase is often a special LDAP system user with read-only access to the LDAP directory. After Salt searches the directory to determine the actual user's DN and groups, it re-authenticates as the user running the Salt commands.

If you are already aware of the structure of your DNs and permissions in your LDAP store are set such that users can look up their own group memberships, then the first and second users can be the same. To tell Salt this is the case, omit the `auth.ldap.bindpw` parameter. Note this is not the same thing as using an anonymous bind. Most LDAP servers will not permit anonymous bind, and as mentioned above, if *auth.ldap.anonymous* is False you cannot use an empty password.

You can template the `binddn` like this:

```
auth.ldap.basedn: dc=saltstack,dc=com
auth.ldap.binddn: uid={{ username }},cn=users,cn=accounts,dc=saltstack,dc=com
```

Salt will use the password entered on the salt command line in place of the bindpw.

To use two separate users, specify the LDAP lookup user in the binddn directive, and include a bindpw like so

```
auth.ldap.binddn: uid=ldaplookup,cn=sysaccounts,cn=etc,dc=saltstack,dc=com
auth.ldap.bindpw: mypassword
```

As mentioned before, Salt uses a filter to find the DN associated with a user. Salt substitutes the `{{ username }}` value for the username when querying LDAP

```
auth.ldap.filter: uid={{ username }}
```

Determining Group Memberships (OpenLDAP / non-Active Directory)

For OpenLDAP, to determine group membership, one can specify an OU that contains group data. This is prepended to the basedn to create a search path. Then the results are filtered against `auth.ldap.groupclass`, default `posixGroup`, and the account's 'name' attribute, `memberUid` by default.

```
auth.ldap.groupou: Groups
```

Note that as of 2017.7, auth.ldap.groupclass can refer to either a groupclass or an objectClass. For some LDAP servers (notably OpenLDAP without the `memberOf` overlay enabled) to determine group membership we need to know both the `objectClass` and the `memberUid` attributes. Usually for these servers you will want a `auth.ldap.groupclass` of `posixGroup` and an `auth.ldap.groupattribute` of `memberUid`.

LDAP servers with the `memberOf` overlay will have entries similar to `auth.ldap.groupclass: person` and `auth.ldap.groupattribute: memberOf`.

When using the `ldap('DC=domain,DC=com')` eauth operator, sometimes the records returned from LDAP or Active Directory have fully-qualified domain names attached, while minion IDs instead are simple hostnames. The parameter below allows the administrator to strip off a certain set of domain names so the hostnames looked up in the directory service can match the minion IDs.

```
auth.ldap.minion_stripdomains: ['.external.bigcorp.com', '.internal.bigcorp.com']
```

Determining Group Memberships (Active Directory)

Active Directory handles group membership differently, and does not utilize the `groupou` configuration variable. AD needs the following options in the master config:

```
auth.ldap.activedirectory: True
auth.ldap.filter: sAMAccountName={{username}}
auth.ldap.accountattributename: sAMAccountName
auth.ldap.groupclass: group
auth.ldap.persontype: person
```

To determine group membership in AD, the username and password that is entered when LDAP is requested as the eAuth mechanism on the command line is used to bind to AD's LDAP interface. If this fails, then it doesn't matter what groups the user belongs to, he or she is denied access. Next, the `distinguishedName` of the user is looked up with the following LDAP search:

```
(&(<value of auth.ldap.accountattributename>={{username}})
  (objectClass=<value of auth.ldap.persontype>)
)
```

This should return a distinguishedName that we can use to filter for group membership. Then the following LDAP query is executed:

```
(&(member=<distinguishedName from search above>)
  (objectClass=<value of auth.ldap.groupclass>)
)
```

```
external_auth:
  ldap:
    test_ldap_user:
        - '*':
            - test.ping
```

To configure a LDAP group, append a `%` to the ID:

```
external_auth:
  ldap:
    test_ldap_group%:
      - '*':
        - test.echo
```

In addition, if there are a set of computers in the directory service that should be part of the eAuth definition, they can be specified like this:

```
external_auth:
  ldap:
    test_ldap_group%:
      - ldap('DC=corp,DC=example,DC=com'):
        - test.echo
```

The string inside `ldap()` above is any valid LDAP/AD tree limiter. `OU=` in particular is permitted as long as it would return a list of computer objects.

3.6.3 Peer Communication

Salt 0.9.0 introduced the capability for Salt minions to publish commands. The intent of this feature is not for Salt minions to act as independent brokers one with another, but to allow Salt minions to pass commands to each other.

In Salt 0.10.0 the ability to execute runners from the master was added. This allows for the master to return collective data from runners back to the minions via the peer interface.

The peer interface is configured through two options in the master configuration file. For minions to send commands from the master the `peer` configuration is used. To allow for minions to execute runners from the master the `peer_run` configuration is used.

Since this presents a viable security risk by allowing minions access to the master publisher the capability is turned off by default. The minions can be allowed access to the master publisher on a per minion basis based on regular expressions. Minions with specific ids can be allowed access to certain Salt modules and functions.

Peer Configuration

The configuration is done under the `peer` setting in the Salt master configuration file, here are a number of configuration possibilities.

The simplest approach is to enable all communication for all minions, this is only recommended for very secure environments.

```
peer:
  .*:
    - .*
```

This configuration will allow minions with IDs ending in example.com access to the test, ps, and pkg module functions.

```
peer:
  .*example.com:
    - test.*
    - ps.*
    - pkg.*
```

The configuration logic is simple, a regular expression is passed for matching minion ids, and then a list of expressions matching minion functions is associated with the named minion. For instance, this configuration will also allow minions ending with foo.org access to the publisher.

```
peer:
  .*example.com:
    - test.*
    - ps.*
    - pkg.*
  .*foo.org:
    - test.*
    - ps.*
    - pkg.*
```

Note: Functions are matched using regular expressions.

Peer Runner Communication

Configuration to allow minions to execute runners from the master is done via the `peer_run` option on the master. The `peer_run` configuration follows the same logic as the `peer` option. The only difference is that access is granted to runner modules.

To open up access to all minions to all runners:

```
peer_run:
  .*:
    - .*
```

This configuration will allow minions with IDs ending in example.com access to the manage and jobs runner functions.

```
peer_run:
  .*example.com:
    - manage.*
    - jobs.*
```

Note: Functions are matched using regular expressions.

Using Peer Communication

The publish module was created to manage peer communication. The publish module comes with a number of functions to execute peer communication in different ways. Currently there are three functions in the publish module. These examples will show how to test the peer system via the salt-call command.

To execute test.version on all minions:

```
# salt-call publish.publish \* test.version
```

To execute the manage.up runner:

```
# salt-call publish.runner manage.up
```

To match minions using other matchers, use `tgt_type`:

```
# salt-call publish.publish 'webserv* and not G@os:Ubuntu' test.version tgt_type=
↪'compound'
```

Note: In pre-2017.7.0 releases, use `expr_form` instead of `tgt_type`.

3.6.4 When to Use Each Authentication System

`publisher_acl` is useful for allowing local system users to run Salt commands without giving them root access. If you can log into the Salt master directly, then `publisher_acl` allows you to use Salt without root privileges. If the local system is configured to authenticate against a remote system, like LDAP or Active Directory, then `publisher_acl` will interact with the remote system transparently.

`external_auth` is useful for `salt-api` or for making your own scripts that use Salt's Python API. It can be used at the CLI (with the `-a` flag) but it is more cumbersome as there are more steps involved. The only time it is useful at the CLI is when the local system is *not* configured to authenticate against an external service *but* you still want Salt to authenticate against an external service.

3.6.5 Examples

The access controls are manifested using matchers in these configurations:

```
publisher_acl:
  fred:
    - web\*:
      - pkg.list_pkgs
      - test.*
      - apache.*
```

In the above example, fred is able to send commands only to minions which match the specified glob target. This can be expanded to include other functions for other minions based on standard targets (all matchers are supported except the compound one).

```
external_auth:
  pam:
    dave:
      - test.version
      - mongo\*:
        - network.*
      - log\*:
        - network.*
        - pkg.*
      - 'G@os:RedHat':
        - kmod.*
    steve:
      - .*
```

The above allows for all minions to be hit by test.version by dave, and adds a few functions that dave can execute on other minions. It also allows steve unrestricted access to salt commands.

Note: Functions are matched using regular expressions.

3.7 Job Management

New in version 0.9.7.

Since Salt executes jobs running on many systems, Salt needs to be able to manage jobs running on many systems.

3.7.1 The Minion proc System

Salt Minions maintain a *proc* directory in the Salt `cachedir`. The *proc* directory maintains files named after the executed job ID. These files contain the information about the current running jobs on the minion and allow for jobs to be looked up. This is located in the *proc* directory under the cachedir, with a default configuration it is under `/var/cache/salt/{master|minion}/proc`.

3.7.2 Functions in the saltutil Module

Salt 0.9.7 introduced a few new functions to the *`saltutil`* module for managing jobs. These functions are:

1. `running` Returns the data of all running jobs that are found in the *proc* directory.
2. `find_job` Returns specific data about a certain job based on job id.
3. `signal_job` Allows for a given jid to be sent a signal.
4. `term_job` Sends a termination signal (SIGTERM, 15) to the process controlling the specified job.
5. `kill_job` Sends a kill signal (SIGKILL, 9) to the process controlling the specified job.

These functions make up the core of the back end used to manage jobs at the minion level.

3.7.3 The jobs Runner

A convenience runner front end and reporting system has been added as well. The jobs runner contains functions to make viewing data easier and cleaner.

The jobs runner contains a number of functions...

active

The active function runs saltutil.running on all minions and formats the return data about all running jobs in a much more usable and compact format. The active function will also compare jobs that have returned and jobs that are still running, making it easier to see what systems have completed a job and what systems are still being waited on.

```
# salt-run jobs.active
```

lookup_jid

When jobs are executed the return data is sent back to the master and cached. By default it is cached for 24 hours, but this can be configured via the `keep_jobs` option in the master configuration. Using the lookup_jid runner will display the same return data that the initial job invocation with the salt command would display.

```
# salt-run jobs.lookup_jid <job id number>
```

list_jobs

Before finding a historic job, it may be required to find the job id. `list_jobs` will parse the cached execution data and display all of the job data for jobs that have already, or partially returned.

```
# salt-run jobs.list_jobs
```

3.7.4 Scheduling Jobs

Salt's scheduling system allows incremental executions on minions or the master. The schedule system exposes the execution of any execution function on minions or any runner on the master.

Scheduling can be enabled by multiple methods:

- `schedule` option in either the master or minion config files. These require the master or minion application to be restarted in order for the schedule to be implemented.
- Minion pillar data. Schedule is implemented by refreshing the minion's pillar data, for example by using `saltutil.refresh_pillar`.
- The *`schedule state`* or *`schedule module`*

Note: The scheduler executes different functions on the master and minions. When running on the master the functions reference runner functions, when running on the minion the functions specify execution functions.

A scheduled run has no output on the minion unless the config is set to info level or higher. Refer to `minion-logging-settings`.

States are executed on the minion, as all states are. You can pass positional arguments and provide a YAML dict of named arguments.

```
schedule:
  job1:
    function: state.sls
    seconds: 3600
    args:
      - httpd
    kwargs:
      test: True
```

This will schedule the command: `state.sls httpd test=True` every 3600 seconds (every hour).

```
schedule:
  job1:
    function: state.sls
    seconds: 3600
```

(continues on next page)

(continued from previous page)

```
    args:
      - httpd
    kwargs:
      test: True
    splay: 15
```

This will schedule the command: `state.sls httpd test=True` every 3600 seconds (every hour) splaying the time between 0 and 15 seconds.

```
schedule:
  job1:
    function: state.sls
    seconds: 3600
    args:
      - httpd
    kwargs:
      test: True
    splay:
      start: 10
      end: 15
```

This will schedule the command: `state.sls httpd test=True` every 3600 seconds (every hour) splaying the time between 10 and 15 seconds.

Schedule by Date and Time

New in version 2014.7.0.

Frequency of jobs can also be specified using date strings supported by the Python `dateutil` library. This requires the Python `dateutil` library to be installed.

```
schedule:
  job1:
    function: state.sls
    args:
      - httpd
    kwargs:
      test: True
    when: 5:00pm
```

This will schedule the command: `state.sls httpd test=True` at 5:00 PM minion localtime.

```
schedule:
  job1:
    function: state.sls
    args:
      - httpd
    kwargs:
      test: True
    when:
      - Monday 5:00pm
      - Tuesday 3:00pm
      - Wednesday 5:00pm
      - Thursday 3:00pm
      - Friday 5:00pm
```

This will schedule the command: `state.sls httpd test=True` at 5:00 PM on Monday, Wednesday and Friday, and 3:00 PM on Tuesday and Thursday.

```
schedule:
  job1:
    function: state.sls
    args:
      - httpd
    kwargs:
      test: True
    when:
      - 'tea time'
```

```
whens:
  tea time: 1:40pm
  deployment time: Friday 5:00pm
```

The Salt scheduler also allows custom phrases to be used for the *when* parameter. These *whens* can be stored as either pillar values or grain values.

```
schedule:
  job1:
    function: state.sls
    seconds: 3600
    args:
      - httpd
    kwargs:
      test: True
    range:
      start: 8:00am
      end: 5:00pm
```

This will schedule the command: `state.sls httpd test=True` every 3600 seconds (every hour) between the hours of 8:00 AM and 5:00 PM. The range parameter must be a dictionary with the date strings using the `dateutil` format.

```
schedule:
  job1:
    function: state.sls
    seconds: 3600
    args:
      - httpd
    kwargs:
      test: True
    range:
      invert: True
      start: 8:00am
      end: 5:00pm
```

Using the invert option for range, this will schedule the command `state.sls httpd test=True` every 3600 seconds (every hour) until the current time is between the hours of 8:00 AM and 5:00 PM. The range parameter must be a dictionary with the date strings using the `dateutil` format.

```
schedule:
  job1:
    function: pkg.install
    kwargs:
```

(continues on next page)

(continued from previous page)

```
      pkgs: [{'bar': '>1.2.3'}]
      refresh: true
    once: '2016-01-07T14:30:00'
```

This will schedule the function `pkg.install` to be executed once at the specified time. The schedule entry `job1` will not be removed after the job completes, therefore use `schedule.delete` to manually remove it afterwards.

The default date format is ISO 8601 but can be overridden by also specifying the `once_fmt` option, like this:

```
schedule:
  job1:
    function: test.ping
    once: 2015-04-22T20:21:00
    once_fmt: '%Y-%m-%dT%H:%M:%S'
```

Maximum Parallel Jobs Running

New in version 2014.7.0.

The scheduler also supports ensuring that there are no more than N copies of a particular routine running. Use this for jobs that may be long-running and could step on each other or pile up in case of infrastructure outage.

The default for `maxrunning` is 1.

```
schedule:
  long_running_job:
    function: big_file_transfer
    jid_include: True
    maxrunning: 1
```

Cron-like Schedule

New in version 2014.7.0.

```
schedule:
  job1:
    function: state.sls
    cron: '*/15 * * * *'
    args:
      - httpd
    kwargs:
      test: True
```

The scheduler also supports scheduling jobs using a cron like format. This requires the Python `croniter` library.

Job Data Return

New in version 2015.5.0.

By default, data about jobs runs from the Salt scheduler is returned to the master. Setting the `return_job` parameter to False will prevent the data from being sent back to the Salt master.

```yaml
schedule:
  job1:
    function: scheduled_job_function
    return_job: False
```

Job Metadata

New in version 2015.5.0.

It can be useful to include specific data to differentiate a job from other jobs. Using the metadata parameter special values can be associated with a scheduled job. These values are not used in the execution of the job, but can be used to search for specific jobs later if combined with the `return_job` parameter. The metadata parameter must be specified as a dictionary, othewise it will be ignored.

```yaml
schedule:
  job1:
    function: scheduled_job_function
    metadata:
      foo: bar
```

Run on Start

New in version 2015.5.0.

By default, any job scheduled based on the startup time of the minion will run the scheduled job when the minion starts up. Sometimes this is not the desired situation. Using the `run_on_start` parameter set to `False` will cause the scheduler to skip this first run and wait until the next scheduled run:

```yaml
schedule:
  job1:
    function: state.sls
    seconds: 3600
    run_on_start: False
    args:
      - httpd
    kwargs:
      test: True
```

Until and After

New in version 2015.8.0.

```yaml
schedule:
  job1:
    function: state.sls
    seconds: 15
    until: '12/31/2015 11:59pm'
    args:
      - httpd
    kwargs:
      test: True
```

Using the until argument, the Salt scheduler allows you to specify an end time for a scheduled job. If this argument is specified, jobs will not run once the specified time has passed. Time should be specified in a format supported by the `dateutil` library. This requires the Python `dateutil` library to be installed.

New in version 2015.8.0.

```
schedule:
  job1:
    function: state.sls
    seconds: 15
    after: '12/31/2015 11:59pm'
    args:
      - httpd
    kwargs:
      test: True
```

Using the after argument, the Salt scheduler allows you to specify an start time for a scheduled job. If this argument is specified, jobs will not run until the specified time has passed. Time should be specified in a format supported by the `dateutil` library. This requires the Python `dateutil` library to be installed.

Scheduling States

```
schedule:
  log-loadavg:
    function: cmd.run
    seconds: 3660
    args:
      - 'logger -t salt < /proc/loadavg'
    kwargs:
      stateful: False
      shell: /bin/sh
```

Scheduling Highstates

To set up a highstate to run on a minion every 60 minutes set this in the minion config or pillar:

```
schedule:
  highstate:
    function: state.highstate
    minutes: 60
```

Time intervals can be specified as seconds, minutes, hours, or days.

Scheduling Runners

Runner executions can also be specified on the master within the master configuration file:

```
schedule:
  run_my_orch:
    function: state.orchestrate
    hours: 6
    splay: 600
    args:
      - orchestration.my_orch
```

The above configuration is analogous to running `salt-run state.orch orchestration.my_orch` every 6 hours.

Scheduler With Returner

The scheduler is also useful for tasks like gathering monitoring data about a minion, this schedule option will gather status data and send it to a MySQL returner database:

```
schedule:
  uptime:
    function: status.uptime
    seconds: 60
    returner: mysql
  meminfo:
    function: status.meminfo
    minutes: 5
    returner: mysql
```

Since specifying the returner repeatedly can be tiresome, the `schedule_returner` option is available to specify one or a list of global returners to be used by the minions when scheduling.

3.8 Managing the Job Cache

The Salt Master maintains a job cache of all job executions which can be queried via the jobs runner. This job cache is called the Default Job Cache.

3.8.1 Default Job Cache

A number of options are available when configuring the job cache. The default caching system uses local storage on the Salt Master and can be found in the job cache directory (on Linux systems this is typically `/var/cache/salt/master/jobs`). The default caching system is suitable for most deployments as it does not typically require any further configuration or management.

The default job cache is a temporary cache and jobs will be stored for 24 hours. If the default cache needs to store jobs for a different period the time can be easily adjusted by changing the `keep_jobs` parameter in the Salt Master configuration file. The value passed in is measured via hours:

```
keep_jobs: 24
```

Reducing the Size of the Default Job Cache

The Default Job Cache can sometimes be a burden on larger deployments (over 5000 minions). Disabling the job cache will make previously executed jobs unavailable to the jobs system and is not generally recommended. Normally it is wise to make sure the master has access to a faster IO system or a tmpfs is mounted to the jobs dir.

However, you can disable the *`job_cache`* by setting it to `False` in the Salt Master configuration file. Setting this value to `False` means that the Salt Master will no longer cache minion returns, but a JID directory and `jid` file for each job will still be created. This JID directory is necessary for checking for and preventing JID collisions.

The default location for the job cache is in the `/var/cache/salt/master/jobs/` directory.

Setting the *`job_cache`* to `False` in addition to setting the *`keep_jobs`* option to a smaller value, such as `1`, in the Salt Master configuration file will reduce the size of the Default Job Cache, and thus the burden on the Salt Master.

Note: Changing the `keep_jobs` option sets the number of hours to keep old job information and defaults to `24` hours. Do not set this value to `0` when trying to make the cache cleaner run more frequently, as this means the cache cleaner will never run.

3.8.2 Additional Job Cache Options

Many deployments may wish to use an external database to maintain a long term register of executed jobs. Salt comes with two main mechanisms to do this, the master job cache and the external job cache.

See *Storing Job Results in an External System*.

3.9 Storing Job Results in an External System

After a job executes, job results are returned to the Salt Master by each Salt Minion. These results are stored in the *Default Job Cache*.

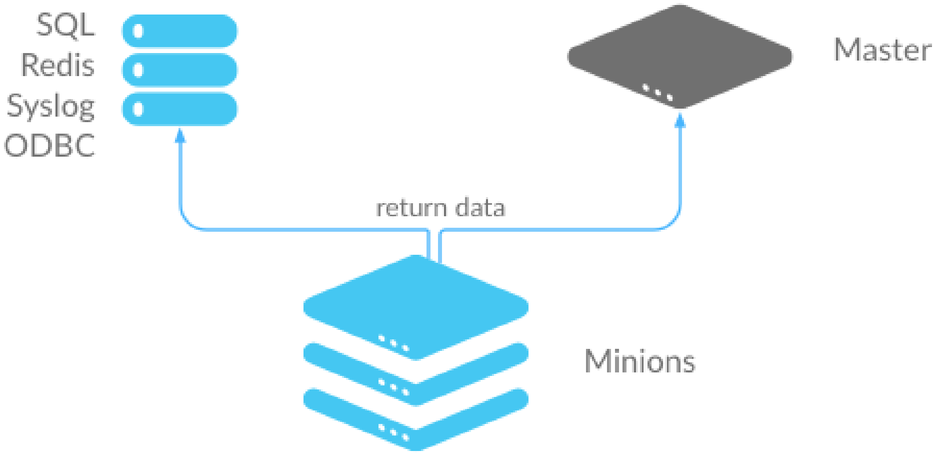
In addition to the Default Job Cache, Salt provides two additional mechanisms to send job results to other systems (databases, local syslog, and others):

- External Job Cache
- Master Job Cache

The major difference between these two mechanism is from where results are returned (from the Salt Master or Salt Minion). Configuring either of these options will also make the `Jobs Runner functions` to automatically query the remote stores for information.

3.9.1 External Job Cache - Minion-Side Returner

When an External Job Cache is configured, data is returned to the Default Job Cache on the Salt Master like usual, and then results are also sent to an External Job Cache using a Salt returner module running on the Salt Minion.



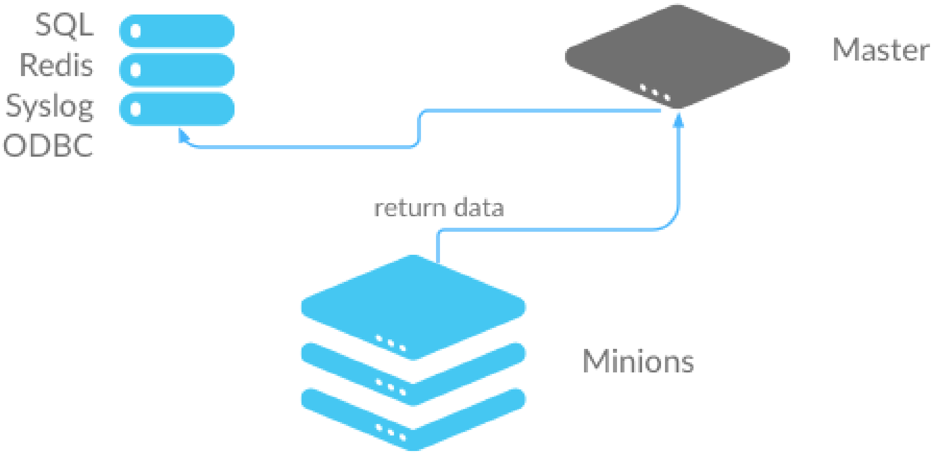
- Advantages: Data is stored without placing additional load on the Salt Master.

- Disadvantages: Each Salt Minion connects to the external job cache, which can result in a large number of connections. Also requires additional configuration to get returner module settings on all Salt Minions.

3.9.2 Master Job Cache - Master-Side Returner

New in version 2014.7.0.

Instead of configuring an External Job Cache on each Salt Minion, you can configure the Master Job Cache to send job results from the Salt Master instead. In this configuration, Salt Minions send data to the Default Job Cache as usual, and then the Salt Master sends the data to the external system using a Salt returner module running on the Salt Master.



- Advantages: A single connection is required to the external system. This is preferred for databases and similar systems.
- Disadvantages: Places additional load on your Salt Master.

3.9.3 Configure an External or Master Job Cache

Step 1: Understand Salt Returners

Before you configure a job cache, it is essential to understand Salt returner modules ("returners"). Returners are pluggable Salt Modules that take the data returned by jobs, and then perform any necessary steps to send the data to an external system. For example, a returner might establish a connection, authenticate, and then format and transfer data.

The Salt Returner system provides the core functionality used by the External and Master Job Cache systems, and the same returners are used by both systems.

Salt currently provides many different returners that let you connect to a wide variety of systems. A complete list is available at *all Salt returners*. Each returner is configured differently, so make sure you read and follow the instructions linked from that page.

For example, the MySQL returner requires:

- A database created using provided schema (structure is available at `MySQL returner`)
- A user created with privileges to the database
- Optional SSL configuration

A simpler returner, such as Slack or HipChat, requires:

- An API key/version
- The target channel/room
- The username that should be used to send the message

Step 2: Configure the Returner

After you understand the configuration and have the external system ready, the configuration requirements must be declared.

External Job Cache

The returner configuration settings can be declared in the Salt Minion configuration file, the Minion's pillar data, or the Minion's grains.

If `external_job_cache` configuration settings are specified in more than one place, the options are retrieved in the following order. The first configuration location that is found is the one that will be used.

- Minion configuration file
- Minion's grains
- Minion's pillar data

Master Job Cache

The returner configuration settings for the Master Job Cache should be declared in the Salt Master's configuration file.

Configuration File Examples

MySQL requires:

```
mysql.host: 'salt'
mysql.user: 'salt'
mysql.pass: 'salt'
mysql.db: 'salt'
mysql.port: 3306
```

Slack requires:

```
slack.channel: 'channel'
slack.api_key: 'key'
slack.from_name: 'name'
```

After you have configured the returner and added settings to the configuration file, you can enable the External or Master Job Cache.

Step 3: Enable the External or Master Job Cache

Configuration is a single line that specifies an already-configured returner to use to send all job data to an external system.

External Job Cache

To enable a returner as the External Job Cache (Minion-side), add the following line to the Salt Master configuration file:

```
ext_job_cache: <returner>
```

For example:

```
ext_job_cache: mysql
```

Note: When configuring an External Job Cache (Minion-side), the returner settings are added to the Minion configuration file, but the External Job Cache setting is configured in the Master configuration file.

Master Job Cache

To enable a returner as a Master Job Cache (Master-side), add the following line to the Salt Master configuration file:

```
master_job_cache: <returner>
```

For example:

```
master_job_cache: mysql
```

Verify that the returner configuration settings are in the Master configuration file, and be sure to restart the salt-master service after you make configuration changes. (`service salt-master restart`).

3.10 Logging

The salt project tries to get the logging to work for you and help us solve any issues you might find along the way.

If you want to get some more information on the nitty-gritty of salt's logging system, please head over to the *logging development document*, if all you're after is salt's logging configurations, please continue reading.

3.10.1 Log Levels

The log levels are ordered numerically such that setting the log level to a specific level will record all log statements at that level and higher. For example, setting `log_level: error` will log statements at `error`, `critical`, and `quiet` levels, although nothing *should* be logged at `quiet` level.

Most of the logging levels are defined by default in Python's logging library and can be found in the official Python documentation. Salt uses some more levels in addition to the standard levels. All levels available in salt are shown in the table below.

Note: Python dependencies used by salt may define and use additional logging levels. For example, the Python 2 version of the `multiprocessing` standard Python library uses the levels `subwarning`, 25 and `subdebug`, 5.

Level	Numeric value	Description
quiet	1000	Nothing should be logged at this level
critical	50	Critical errors
error	40	Errors
warning	30	Warnings
info	20	Normal log information
profile	15	Profiling information on salt performance
debug	10	Information useful for debugging both salt implementations and salt code
trace	5	More detailed code debugging information
garbage	1	Even more debugging information
all	0	Everything

3.10.2 Available Configuration Settings

`log_file`

The log records can be sent to a regular file, local path name, or network location. Remote logging works best when configured to use rsyslogd(8) (e.g.: `file:///dev/log`), with rsyslogd(8) configured for network logging. The format for remote addresses is:

```
<file|udp|tcp>://<host|socketpath>:<port-if-required>/<log-facility>
```

Where `log-facility` is the symbolic name of a syslog facility as defined in the `SysLogHandler documentation`. It defaults to `LOG_USER`.

Default: Dependent of the binary being executed, for example, for `salt-master`, `/var/log/salt/master`.

Examples:

```
log_file: /var/log/salt/master
```

```
log_file: /var/log/salt/minion
```

```
log_file: file:///dev/log
```

```
log_file: file:///dev/log/LOG_DAEMON
```

```
log_file: udp://loghost:10514
```

`log_level`

Default: `warning`

The level of log record messages to send to the console. One of `all`, `garbage`, `trace`, `debug`, `profile`, `info`, `warning`, `error`, `critical`, `quiet`.

```
log_level: warning
```

Note: Add `log_level: quiet` in salt configuration file to completely disable logging. In case of running salt in command line use `--log-level=quiet` instead.

`log_level_logfile`

Default: `info`

The level of messages to send to the log file. One of `all`, `garbage`, `trace`, `debug`, `profile`, `info`, `warning`, `error`, `critical`, `quiet`.

```
log_level_logfile: warning
```

`log_datefmt`

Default: `%H:%M:%S`

The date and time format used in console log messages. Allowed date/time formatting matches those used in `time.strftime()`.

```
log_datefmt: '%H:%M:%S'
```

`log_datefmt_logfile`

Default: `%Y-%m-%d %H:%M:%S`

The date and time format used in log file messages. Allowed date/time formatting matches those used in `time.strftime()`.

```
log_datefmt_logfile: '%Y-%m-%d %H:%M:%S'
```

`log_fmt_console`

Default: `[%(levelname)-8s] %(message)s`

The format of the console logging messages. All standard python logging `LogRecord` attributes can be used. Salt also provides these custom LogRecord attributes to colorize console log output:

```
'%(colorlevel)s'   # log level name colorized by level
'%(colorname)s'    # colorized module name
'%(colorprocess)s' # colorized process number
'%(colormsg)s'     # log message colorized by level
```

Note: The `%(colorlevel)s`, `%(colorname)s`, and `%(colorprocess)` LogRecord attributes also include padding and enclosing brackets, `[` and `]` to match the default values of their collateral non-colorized LogRecord attributes.

```
log_fmt_console: '[%(levelname)-8s] %(message)s'
```

`log_fmt_logfile`

Default: `%(asctime)s,%(msecs)03d [%(name)-17s][%(levelname)-8s] %(message)s`

The format of the log file logging messages. All standard python logging `LogRecord` attributes can be used. Salt also provides these custom LogRecord attributes that include padding and enclosing brackets `[` and `]`:

```
'%(bracketlevel)s'   # equivalent to [%(levelname)-8s]
'%(bracketname)s'    # equivalent to [%(name)-17s]
'%(bracketprocess)s' # equivalent to [%(process)5s]
```

```
log_fmt_logfile: '%(asctime)s,%(msecs)03d [%(name)-17s][%(levelname)-8s] %(message)s'
```

`log_granular_levels`

Default: `{}`

This can be used to control logging levels more specifically, based on log call name. The example sets the main salt library at the 'warning' level, sets `salt.modules` to log at the `debug` level, and sets a custom module to the `all` level:

```
log_granular_levels:
  'salt': 'warning'
  'salt.modules': 'debug'
  'salt.loader.saltmaster.ext.module.custom_module': 'all'
```

`log_fmt_jid`

Default: `[JID: %(jid)s]`

The format of the JID when added to logging messages.

```
log_fmt_jid: '[JID: %(jid)s]'
```

External Logging Handlers

Besides the internal logging handlers used by salt, there are some external which can be used, see the *external logging handlers* document.

3.11 External Logging Handlers

fluent_mod	Fluent Logging Handler
log4mongo_mod	Log4Mongo Logging Handler
logstash_mod	Logstash Logging Handler
sentry_mod	Sentry Logging Handler

3.11.1 salt.log.handlers.fluent_mod

Fluent Logging Handler

New in version 2015.8.0.

This module provides some fluentd logging handlers.

Fluent Logging Handler

In the *fluent* configuration file:

```
<source>
  type forward
  bind localhost
  port 24224
</source>
```

Then, to send logs via fluent in Logstash format, add the following to the salt (master and/or minion) configuration file:

```
fluent_handler:
  host: localhost
  port: 24224
```

To send logs via fluent in the Graylog raw json format, add the following to the salt (master and/or minion) configuration file:

```
fluent_handler:
  host: localhost
  port: 24224
  payload_type: graylog
  tags:
  - salt_master.SALT
```

The above also illustrates the *tags* option, which allows one to set descriptive (or useful) tags on records being sent. If not provided, this defaults to the single tag: 'salt'. Also note that, via Graylog "magic", the 'facility' of the logged message is set to 'SALT' (the portion of the tag after the first period), while the tag itself will be set to simply 'salt_master'. This is a feature, not a bug :)

Note: There is a third emitter, for the GELF format, but it is largely untested, and I don't currently have a setup supporting this config, so while it runs cleanly and outputs what LOOKS to be valid GELF, any real-world feedback on its usefulness, and correctness, will be appreciated.

Log Level

The `fluent_handler` configuration section accepts an additional setting `log_level`. If not set, the logging level used will be the one defined for `log_level` in the global configuration file section.

Inspiration

This work was inspired in fluent-logger-python

3.11.2 salt.log.handlers.log4mongo_mod

Log4Mongo Logging Handler

This module provides a logging handler for sending salt logs to MongoDB

Configuration

In the salt configuration file (e.g. /etc/salt/{master,minion}):

```
log4mongo_handler:
  host: mongodb_host
  port: 27017
  database_name: logs
  collection: salt_logs
  username: logging
  password: reindeerflotilla
  write_concern: 0
  log_level: warning
```

Log Level

If not set, the log_level will be set to the level defined in the global configuration file setting.

Inspiration

This work was inspired by the Salt logging handlers for LogStash and Sentry and by the log4mongo Python implementation.

3.11.3 salt.log.handlers.logstash_mod

Logstash Logging Handler

New in version 0.17.0.

This module provides some Logstash logging handlers.

UDP Logging Handler

For versions of Logstash before 1.2.0:

In the salt configuration file:

```
logstash_udp_handler:
  host: 127.0.0.1
  port: 9999
  version: 0
  msg_type: logstash
```

In the Logstash configuration file:

```
input {
  udp {
    type => "udp-type"
    format => "json_event"
  }
}
```

For version 1.2.0 of Logstash and newer:

In the salt configuration file:

```
logstash_udp_handler:
  host: 127.0.0.1
  port: 9999
  version: 1
  msg_type: logstash
```

In the Logstash configuration file:

```
input {
  udp {
    port => 9999
    codec => json
  }
}
```

Please read the UDP input configuration page for additional information.

ZeroMQ Logging Handler

For versions of Logstash before 1.2.0:

In the salt configuration file:

```
logstash_zmq_handler:
  address: tcp://127.0.0.1:2021
  version: 0
```

In the Logstash configuration file:

```
input {
  zeromq {
    type => "zeromq-type"
    mode => "server"
    topology => "pubsub"
    address => "tcp://0.0.0.0:2021"
    charset => "UTF-8"
    format => "json_event"
  }
}
```

For version 1.2.0 of Logstash and newer:

In the salt configuration file:

```
logstash_zmq_handler:
  address: tcp://127.0.0.1:2021
  version: 1
```

In the Logstash configuration file:

```
input {
  zeromq {
    topology => "pubsub"
```

(continues on next page)

(continued from previous page)

```
    address => "tcp://0.0.0.0:2021"
    codec => json
  }
}
```

Please read the ZeroMQ input configuration page for additional information.

Important Logstash Setting

One of the most important settings that you should not forget on your Logstash configuration file regarding these logging handlers is `format`. Both the *UDP* and *ZeroMQ* inputs need to have `format` as `json_event` which is what we send over the wire.

Log Level

Both the `logstash_udp_handler` and the `logstash_zmq_handler` configuration sections accept an additional setting `log_level`. If not set, the logging level used will be the one defined for `log_level` in the global configuration file section.

HWM

The high water mark for the ZMQ socket setting. Only applicable for the `logstash_zmq_handler`.

Inspiration

This work was inspired in pylogstash, python-logstash, canary and the PyZMQ logging handler.

3.11.4 salt.log.handlers.sentry_mod

Sentry Logging Handler

New in version 0.17.0.

This module provides a Sentry logging handler. Sentry is an open source error tracking platform that provides deep context about exceptions that happen in production. Details about stack traces along with the context variables available at the time of the exception are easily browsable and filterable from the online interface. For more details please see Sentry.

Note

The Raven library needs to be installed on the system for this logging handler to be available.

Configuring the python Sentry client, Raven, should be done under the `sentry_handler` configuration key. Additional *context* may be provided for corresponding grain item(s). At the bare minimum, you need to define the DSN. As an example:

```
sentry_handler:
  dsn: https://pub-key:secret-key@app.getsentry.com/app-id
```

More complex configurations can be achieved, for example:

```
sentry_handler:
  servers:
    - https://sentry.example.com
    - http://192.168.1.1
  project: app-id
  public_key: deadbeefdeadbeefdeadbeefdeadbeef
  secret_key: beefdeadbeefdeadbeefdeadbeefdead
  context:
    - os
    - master
    - saltversion
    - cpuarch
    - ec2.tags.environment
```

Note

The `public_key` and `secret_key` variables are not supported with Sentry > 3.0. The DSN key should be used instead.

All the client configuration keys are supported, please see the Raven client documentation.

The default logging level for the sentry handler is `ERROR`. If you wish to define a different one, define `log_level` under the `sentry_handler` configuration key:

```
sentry_handler:
  dsn: https://pub-key:secret-key@app.getsentry.com/app-id
  log_level: warning
```

The available log levels are those also available for the salt `cli` tools and configuration; `salt --help` should give you the required information.

Threaded Transports

Raven's documents rightly suggest using its threaded transport for critical applications. However, don't forget that if you start having troubles with Salt after enabling the threaded transport, please try switching to a non-threaded transport to see if that fixes your problem.

3.12 Salt File Server

Salt comes with a simple file server suitable for distributing files to the Salt minions. The file server is a stateless ZeroMQ server that is built into the Salt master.

The main intent of the Salt file server is to present files for use in the Salt state system. With this said, the Salt file server can be used for any general file transfer from the master to the minions.

3.12.1 File Server Backends

In Salt 0.12.0, the modular fileserver was introduced. This feature added the ability for the Salt Master to integrate different file server backends. File server backends allow the Salt file server to act as a transparent bridge to external resources. A good example of this is the `git` backend, which allows Salt to serve files sourced from one or more git repositories, but there are several others as well. Click *here* for a full list of Salt's fileserver backends.

Enabling a Fileserver Backend

Fileserver backends can be enabled with the *`fileserver_backend`* option.

```
fileserver_backend:
  - git
```

See the *documentation* for each backend to find the correct value to add to *`fileserver_backend`* in order to enable them.

Using Multiple Backends

If *`fileserver_backend`* is not defined in the Master config file, Salt will use the *`roots`* backend, but the *`fileserver_backend`* option supports multiple backends. When more than one backend is in use, the files from the enabled backends are merged into a single virtual filesystem. When a file is requested, the backends will be searched in order for that file, and the first backend to match will be the one which returns the file.

```
fileserver_backend:
  - roots
  - git
```

With this configuration, the environments and files defined in the *`file_roots`* parameter will be searched first, and if the file is not found then the git repositories defined in *`gitfs_remotes`* will be searched.

Defining Environments

Just as the order of the values in *`fileserver_backend`* matters, so too does the order in which different sources are defined within a fileserver environment. For example, given the below *`file_roots`* configuration, if both `/srv/salt/dev/foo.txt` and `/srv/salt/prod/foo.txt` exist on the Master, then `salt://foo.txt` would point to `/srv/salt/dev/foo.txt` in the `dev` environment, but it would point to `/srv/salt/prod/foo.txt` in the `base` environment.

```
file_roots:
  base:
    - /srv/salt/prod
  qa:
    - /srv/salt/qa
    - /srv/salt/prod
  dev:
    - /srv/salt/dev
    - /srv/salt/qa
    - /srv/salt/prod
```

Similarly, when using the *`git`* backend, if both repositories defined below have a `hotfix23` branch/tag, and both of them also contain the file `bar.txt` in the root of the repository at that branch/tag, then `salt://bar.txt` in the `hotfix23` environment would be served from the `first` repository.

```
gitfs_remotes:
  - https://mydomain.tld/repos/first.git
  - https://mydomain.tld/repos/second.git
```

Note: Environments map differently based on the fileserver backend. For instance, the mappings are explicitly defined in `roots` backend, while in the VCS backends (`git`, `hg`, `svn`) the environments are created from branches/tags/bookmarks/etc. For the `minion` backend, the files are all in a single environment, which is specified by the `minionfs_env` option.

See the documentation for each backend for a more detailed explanation of how environments are mapped.

3.12.2 Requesting Files from Specific Environments

The Salt fileserver supports multiple environments, allowing for SLS files and other files to be isolated for better organization.

For the default backend (called `roots`), environments are defined using the `roots` option. Other backends (such as `gitfs`) define environments in their own ways. For a list of available fileserver backends, see *here*.

Querystring Syntax

Any `salt://` file URL can specify its fileserver environment using a querystring syntax, like so:

```
salt://path/to/file?saltenv=foo
```

In *Reactor* configurations, this method must be used to pull files from an environment other than `base`.

In States

Minions can be instructed which environment to use both globally, and for a single state, and multiple methods for each are available:

Globally

A minion can be pinned to an environment using the `environment` option in the minion config file.

Additionally, the environment can be set for a single call to the following functions:

- `state.apply`
- `state.highstate`
- `state.sls`
- `state.top`

Note: When the `saltenv` parameter is used to trigger a *highstate* using either `state.apply` or `state.highstate`, only states from that environment will be applied.

On a Per-State Basis

Within an individual state, there are two ways of specifying the environment. The first is to add a `saltenv` argument to the state. This example will pull the file from the `config` environment:

```
/etc/foo/bar.conf:
  file.managed:
    - source: salt://foo/bar.conf
    - user: foo
    - mode: 600
    - saltenv: config
```

Another way of doing the same thing is to use the *querystring syntax* described above:

```
/etc/foo/bar.conf:
  file.managed:
    - source: salt://foo/bar.conf?saltenv=config
    - user: foo
    - mode: 600
```

Note: Specifying the environment using either of the above methods is only necessary in cases where a state from one environment needs to access files from another environment. If the SLS file containing this state was in the `config` environment, then it would look in that environment by default.

3.12.3 File Server Configuration

The Salt file server is a high performance file server written in ZeroMQ. It manages large files quickly and with little overhead, and has been optimized to handle small files in an extremely efficient manner.

The Salt file server is an environment aware file server. This means that files can be allocated within many root directories and accessed by specifying both the file path and the environment to search. The individual environments can span across multiple directory roots to create overlays and to allow for files to be organized in many flexible ways.

Environments

The Salt file server defaults to the mandatory `base` environment. This environment **MUST** be defined and is used to download files when no environment is specified.

Environments allow for files and sls data to be logically separated, but environments are not isolated from each other. This allows for logical isolation of environments by the engineer using Salt, but also allows for information to be used in multiple environments.

Directory Overlay

The `environment` setting is a list of directories to publish files from. These directories are searched in order to find the specified file and the first file found is returned.

This means that directory data is prioritized based on the order in which they are listed. In the case of this `file_roots` configuration:

```
file_roots:
  base:
    - /srv/salt/base
    - /srv/salt/failover
```

If a file's URI is `salt://httpd/httpd.conf`, it will first search for the file at `/srv/salt/base/httpd/httpd.conf`. If the file is found there it will be returned. If the file is not found there, then `/srv/salt/failover/httpd/httpd.conf` will be used for the source.

This allows for directories to be overlaid and prioritized based on the order they are defined in the configuration.

It is also possible to have `file_roots` which supports multiple environments:

```
file_roots:
  base:
    - /srv/salt/base
  dev:
    - /srv/salt/dev
    - /srv/salt/base
  prod:
    - /srv/salt/prod
    - /srv/salt/base
```

This example ensures that each environment will check the associated environment directory for files first. If a file is not found in the appropriate directory, the system will default to using the base directory.

Local File Server

New in version 0.9.8.

The file server can be rerouted to run from the minion. This is primarily to enable running Salt states without a Salt master. To use the local file server interface, copy the file server data to the minion and set the file_roots option on the minion to point to the directories copied from the master. Once the minion `file_roots` option has been set, change the `file_client` option to local to make sure that the local file server interface is used.

3.12.4 The cp Module

The cp module is the home of minion side file server operations. The cp module is used by the Salt state system, salt-cp, and can be used to distribute files presented by the Salt file server.

Escaping Special Characters

The `salt://` url format can potentially contain a query string, for example `salt://dir/file.txt?saltenv=base`. You can prevent the fileclient/fileserver from interpreting `?` as the initial token of a query string by referencing the file with `salt://|` rather than `salt://`.

```
/etc/marathon/conf/?checkpoint:
  file.managed:
    - source: salt://|hw/config/?checkpoint
    - makedirs: True
```

Environments

Since the file server is made to work with the Salt state system, it supports environments. The environments are defined in the master config file and when referencing an environment the file specified will be based on the root directory of the environment.

get_file

The cp.get_file function can be used on the minion to download a file from the master, the syntax looks like this:

```
# salt '*' cp.get_file salt://vimrc /etc/vimrc
```

This will instruct all Salt minions to download the vimrc file and copy it to /etc/vimrc

Template rendering can be enabled on both the source and destination file names like so:

```
# salt '*' cp.get_file "salt://{{grains.os}}/vimrc" /etc/vimrc template=jinja
```

This example would instruct all Salt minions to download the vimrc from a directory with the same name as their OS grain and copy it to /etc/vimrc

For larger files, the cp.get_file module also supports gzip compression. Because gzip is CPU-intensive, this should only be used in scenarios where the compression ratio is very high (e.g. pretty-printed JSON or YAML files).

To use compression, use the `gzip` named argument. Valid values are integers from 1 to 9, where 1 is the lightest compression and 9 the heaviest. In other words, 1 uses the least CPU on the master (and minion), while 9 uses the most.

```
# salt '*' cp.get_file salt://vimrc /etc/vimrc gzip=5
```

Finally, note that by default cp.get_file does *not* create new destination directories if they do not exist. To change this, use the `makedirs` argument:

```
# salt '*' cp.get_file salt://vimrc /etc/vim/vimrc makedirs=True
```

In this example, /etc/vim/ would be created if it didn't already exist.

get_dir

The cp.get_dir function can be used on the minion to download an entire directory from the master. The syntax is very similar to get_file:

```
# salt '*' cp.get_dir salt://etc/apache2 /etc
```

cp.get_dir supports template rendering and gzip compression arguments just like get_file:

```
# salt '*' cp.get_dir salt://etc/{{pillar.webserver}} /etc gzip=5 template=jinja
```

3.12.5 File Server Client Instance

A client instance is available which allows for modules and applications to be written which make use of the Salt file server.

The file server uses the same authentication and encryption used by the rest of the Salt system for network communication.

fileclient Module

The `salt/fileclient.py` module is used to set up the communication from the minion to the master. When creating a client instance using the fileclient module, the minion configuration needs to be passed in. When using the fileclient module from within a minion module the built in `__opts__` data can be passed:

```
import salt.minion
import salt.fileclient

def get_file(path, dest, saltenv='base'):
    '''
    Used to get a single file from the Salt master

    CLI Example:
    salt '*' cp.get_file salt://vimrc /etc/vimrc
    '''
    # Get the fileclient object
    client = salt.fileclient.get_file_client(__opts__)
    # Call get_file
    return client.get_file(path, dest, False, saltenv)
```

Creating a fileclient instance outside of a minion module where the `__opts__` data is not available, it needs to be generated:

```
import salt.fileclient
import salt.config

def get_file(path, dest, saltenv='base'):
    '''
    Used to get a single file from the Salt master
    '''
    # Get the configuration data
    opts = salt.config.minion_config('/etc/salt/minion')
    # Get the fileclient object
    client = salt.fileclient.get_file_client(opts)
    # Call get_file
    return client.get_file(path, dest, False, saltenv)
```

3.13 Git Fileserver Backend Walkthrough

Note: This walkthrough assumes basic knowledge of Salt. To get up to speed, check out the *Salt Walkthrough*.

The gitfs backend allows Salt to serve files from git repositories. It can be enabled by adding `git` to the `fileserver_backend` list, and configuring one or more repositories in `gitfs_remotes`.

Branches and tags become Salt fileserver environments.

Note: Branching and tagging can result in a lot of potentially-conflicting *top files*, for this reason it may be useful to set `top_file_merging_strategy` to `same` in the minions' config files if the top files are being managed in a GitFS repo.

3.13.1 Installing Dependencies

Both pygit2 and GitPython are supported Python interfaces to git. If compatible versions of both are installed, pygit2 will be preferred. In these cases, GitPython can be forced using the `gitfs_provider` parameter in the master config file.

Note: It is recommended to always run the most recent version of any the below dependencies. Certain features of GitFS may not be available without the most recent version of the chosen library.

---

pygit2

The minimum supported version of pygit2 is 0.20.3. Availability for this version of pygit2 is still limited, though the SaltStack team is working to get compatible versions available for as many platforms as possible.

For the Fedora/EPEL versions which have a new enough version packaged, the following command would be used to install pygit2:

```
# yum install python-pygit2
```

Provided a valid version is packaged for Debian/Ubuntu (which is not currently the case), the package name would be the same, and the following command would be used to install it:

```
# apt-get install python-pygit2
```

If pygit2 is not packaged for the platform on which the Master is running, the pygit2 website has installation instructions here. Keep in mind however that following these instructions will install libgit2 and pygit2 without system packages. Additionally, keep in mind that *SSH authentication in pygit2* requires libssh2 (*not* libssh) development libraries to be present before libgit2 is built. On some Debian-based distros `pkg-config` is also required to link libgit2 with libssh2.

Note: If you are receiving the error "Unsupported URL Protocol" in the Salt Master log when making a connection using SSH, review the libssh2 details listed above.

Additionally, version 0.21.0 of pygit2 introduced a dependency on python-cffi, which in turn depends on newer releases of libffi. Upgrading libffi is not advisable as several other applications depend on it, so on older LTS linux releases pygit2 0.20.3 and libgit2 0.20.0 is the recommended combination.

Warning: pygit2 is actively developed and frequently makes non-backwards-compatible API changes, even in minor releases. It is not uncommon for pygit2 upgrades to result in errors in Salt. Please take care when upgrading pygit2, and pay close attention to the changelog, keeping an eye out for API changes. Errors can be reported on the SaltStack issue tracker.

RedHat Pygit2 Issues

The release of RedHat/CentOS 7.3 upgraded both `python-cffi` and `http-parser`, both of which are dependencies for pygit2/libgit2. Both `pygit2` and `libgit2` packages (which are from the EPEL repository) should be upgraded to the most recent versions, at least to `0.24.2`.

The below errors will show up in the master log if an incompatible `python-pygit2` package is installed:

```
2017-02-10 09:07:34,892 [salt.utils.gitfs ][ERROR ][11211] Import pygit2 failed:
↪CompileError: command 'gcc' failed with exit status 1
2017-02-10 09:07:34,907 [salt.utils.gitfs ][ERROR ][11211] gitfs is configured but
↪could not be loaded, are pygit2 and libgit2 installed?
```

(continues on next page)

(continued from previous page)

```
2017-02-10 09:07:34,907 [salt.utils.gitfs ][CRITICAL][11211] No suitable gitfs
↪provider module is installed.
2017-02-10 09:07:34,912 [salt.master ][CRITICAL][11211] Master failed pre flight
↪checks, exiting
```

The below errors will show up in the master log if an incompatible `libgit2` package is installed:

```
2017-02-15 18:04:45,211 [salt.utils.gitfs ][ERROR   ][6211] Error occurred fetching
↪gitfs remote 'https://foo.com/bar.git': No Content-Type header in response
```

A restart of the `salt-master` daemon and gitfs cache directory clean up may be required to allow http(s) repositories to continue to be fetched.

GitPython

GitPython 0.3.0 or newer is required to use GitPython for gitfs. For RHEL-based Linux distros, a compatible version is available in EPEL, and can be easily installed on the master using yum:

```
# yum install GitPython
```

Ubuntu 14.04 LTS and Debian Wheezy (7.x) also have a compatible version packaged:

```
# apt-get install python-git
```

GitPython requires the `git` CLI utility to work. If installed from a system package, then git should already be installed, but if installed via pip then it may still be necessary to install git separately. For MacOS users, GitPython comes bundled in with the Salt installer, but git must still be installed for it to work properly. Git can be installed in several ways, including by installing XCode.

Warning: Keep in mind that if GitPython has been previously installed on the master using pip (even if it was subsequently uninstalled), then it may still exist in the build cache (typically `/tmp/pip-build-root/GitPython`) if the cache is not cleared after installation. The package in the build cache will override any requirement specifiers, so if you try upgrading to version 0.3.2.RC1 by running `pip install 'GitPython==0.3.2.RC1'` then it will ignore this and simply install the version from the cache directory. Therefore, it may be necessary to delete the GitPython directory from the build cache in order to ensure that the specified version is installed.

Warning: GitPython 2.0.9 and newer is not compatible with Python 2.6. If installing GitPython using pip on a machine running Python 2.6, make sure that a version earlier than 2.0.9 is installed. This can be done on the CLI by running `pip install 'GitPython<2.0.9'`, or in a *`pip.installed`* state using the following SLS:

```
GitPython:
  pip.installed:
    - name: 'GitPython < 2.0.9'
```

3.13.2 Simple Configuration

To use the gitfs backend, only two configuration changes are required on the master:

1. Include `gitfs` in the *`fileserver_backend`* list in the master config file:

```
fileserver_backend:
  - gitfs
```

Note: `git` also works here. Prior to the 2018.3.0 release, *only* `git` would work.

---

2. Specify one or more `git://`, `https://`, `file://`, or `ssh://` URLs in *`gitfs_remotes`* to configure which repositories to cache and search for requested files:

```
gitfs_remotes:
  - https://github.com/saltstack-formulas/salt-formula.git
```

   SSH remotes can also be configured using scp-like syntax:

```
gitfs_remotes:
  - git@github.com:user/repo.git
  - ssh://user@domain.tld/path/to/repo.git
```

   Information on how to authenticate to SSH remotes can be found *here*.

3. Restart the master to load the new configuration.

Note: In a master/minion setup, files from a gitfs remote are cached once by the master, so minions do not need direct access to the git repository.

3.13.3 Multiple Remotes

The `gitfs_remotes` option accepts an ordered list of git remotes to cache and search, in listed order, for requested files.

A simple scenario illustrates this cascading lookup behavior:

If the `gitfs_remotes` option specifies three remotes:

```
gitfs_remotes:
  - git://github.com/example/first.git
  - https://github.com/example/second.git
  - file:///root/third
```

And each repository contains some files:

```
first.git:
    top.sls
    edit/vim.sls
    edit/vimrc
    nginx/init.sls

second.git:
    edit/dev_vimrc
    haproxy/init.sls

third:
    haproxy/haproxy.conf
    edit/dev_vimrc
```

Salt will attempt to lookup the requested file from each gitfs remote repository in the order in which they are defined in the configuration. The **git://github.com/example/first.git** remote will be searched first. If the requested file is found, then it is served and no further searching is executed. For example:

- A request for the file **salt://haproxy/init.sls** will be served from the **https://github.com/example/second.git** git repo.
- A request for the file **salt://haproxy/haproxy.conf** will be served from the **file:///root/third** repo.

Note: This example is purposefully contrived to illustrate the behavior of the gitfs backend. This example should not be read as a recommended way to lay out files and git repos.

The **file://** prefix denotes a git repository in a local directory. However, it will still use the given **file://** URL as a remote, rather than copying the git repo to the salt cache. This means that any refs you want accessible must exist as *local* refs in the specified repo.

Warning: Salt versions prior to 2014.1.0 are not tolerant of changing the order of remotes or modifying the URI of existing remotes. In those versions, when modifying remotes it is a good idea to remove the gitfs cache directory (`/var/cache/salt/master/gitfs`) before restarting the salt-master service.

3.13.4 Per-remote Configuration Parameters

New in version 2014.7.0.

The following master config parameters are global (that is, they apply to all configured gitfs remotes):

- `gitfs_base`
- `gitfs_root`
- `gitfs_ssl_verify`
- `gitfs_mountpoint` (new in 2014.7.0)
- `gitfs_user` (**pygit2 only**, new in 2014.7.0)
- `gitfs_password` (**pygit2 only**, new in 2014.7.0)
- `gitfs_insecure_auth` (**pygit2 only**, new in 2014.7.0)
- `gitfs_pubkey` (**pygit2 only**, new in 2014.7.0)
- `gitfs_privkey` (**pygit2 only**, new in 2014.7.0)
- `gitfs_passphrase` (**pygit2 only**, new in 2014.7.0)
- `gitfs_refspecs` (new in 2017.7.0)
- `gitfs_disable_saltenv_mapping` (new in 2018.3.0)
- `gitfs_ref_types` (new in 2018.3.0)
- `gitfs_update_interval` (new in 2018.3.0)

Note: pygit2 only supports disabling SSL verification in versions 0.23.2 and newer.

These parameters can now be overridden on a per-remote basis. This allows for a tremendous amount of customization. Here's some example usage:

```yaml
gitfs_provider: pygit2
gitfs_base: develop

gitfs_remotes:
  - https://foo.com/foo.git
  - https://foo.com/bar.git:
    - root: salt
    - mountpoint: salt://bar
    - base: salt-base
    - ssl_verify: False
    - update_interval: 120
  - https://foo.com/bar.git:
    - name: second_bar_repo
    - root: other/salt
    - mountpoint: salt://other/bar
    - base: salt-base
    - ref_types:
      - branch
  - http://foo.com/baz.git:
    - root: salt/states
    - user: joe
    - password: mysupersecretpassword
    - insecure_auth: True
    - disable_saltenv_mapping: True
    - saltenv:
      - foo:
        - ref: foo
  - http://foo.com/quux.git:
    - all_saltenvs: master
```

Important: There are two important distinctions which should be noted for per-remote configuration:

1. The URL of a remote which has per-remote configuration must be suffixed with a colon.
2. Per-remote configuration parameters are named like the global versions, with the `gitfs_` removed from the beginning. The exception being the `name`, `saltenv`, and `all_saltenvs` parameters, which are only available to per-remote configurations.

The `all_saltenvs` parameter is new in the 2018.3.0 release.

In the example configuration above, the following is true:

1. The first and fourth gitfs remotes will use the `develop` branch/tag as the `base` environment, while the second and third will use the `salt-base` branch/tag as the `base` environment.
2. The first remote will serve all files in the repository. The second remote will only serve files from the `salt` directory (and its subdirectories). The third remote will only server files from the `other/salt` directory (and its subdirectories), while the fourth remote will only serve files from the `salt/states` directory (and its subdirectories).
3. The third remote will only serve files from branches, and not from tags or SHAs.
4. The fourth remote will only have two saltenvs available: `base` (pointed at `develop`), and `foo` (pointed at `foo`).
5. The first and fourth remotes will have files located under the root of the Salt fileserver namespace (`salt://`). The files from the second remote will be located under `salt://bar`, while the files from the third remote will be located under `salt://other/bar`.

6. The second and third remotes reference the same repository and unique names need to be declared for duplicate gitfs remotes.
7. The fourth remote overrides the default behavior of *not authenticating to insecure (non-HTTPS) remotes*.
8. Because `all_saltenvs` is configured for the fifth remote, files from the branch/tag `master` will appear in every fileserver environment.

> **Note:** The use of `http://` (instead of `https://`) is permitted here *only* because authentication is not being used. Otherwise, the `insecure_auth` parameter must be used (as in the fourth remote) to force Salt to authenticate to an `http://` remote.

9. The first remote will wait 120 seconds between updates instead of 60.

3.13.5 Per-Saltenv Configuration Parameters

New in version 2016.11.0.

For more granular control, Salt allows the following three things to be overridden for individual saltenvs within a given repo:

- The *mountpoint*
- The *root*
- The branch/tag to be used for a given saltenv

Here is an example:

```
gitfs_root: salt

gitfs_saltenv:
  - dev:
    - mountpoint: salt://gitfs-dev
    - ref: develop

gitfs_remotes:
  - https://foo.com/bar.git:
    - saltenv:
      - staging:
        - ref: qa
        - mountpoint: salt://bar-staging
      - dev:
        - ref: development
  - https://foo.com/baz.git:
    - saltenv:
      - staging:
        - mountpoint: salt://baz-staging
```

Given the above configuration, the following is true:

1. For all gitfs remotes, files for the `dev` saltenv will be located under `salt://gitfs-dev`.
2. For the `dev` saltenv, files from the first remote will be sourced from the `development` branch, while files from the second remote will be sourced from the `develop` branch.
3. For the `staging` saltenv, files from the first remote will be located under `salt://bar-staging`, while files from the second remote will be located under `salt://baz-staging`.

4. For all gitfs remotes, and in all saltenvs, files will be served from the `salt` directory (and its subdirectories).

3.13.6 Custom Refspecs

New in version 2017.7.0.

GitFS will by default fetch remote branches and tags. However, sometimes it can be useful to fetch custom refs (such as those created for GitHub pull requests). To change the refspecs GitFS fetches, use the `gitfs_refspecs` config option:

```
gitfs_refspecs:
  - '+refs/heads/*:refs/remotes/origin/*'
  - '+refs/tags/*:refs/tags/*'
  - '+refs/pull/*/head:refs/remotes/origin/pr/*'
  - '+refs/pull/*/merge:refs/remotes/origin/merge/*'
```

In the above example, in addition to fetching remote branches and tags, GitHub's custom refs for pull requests and merged pull requests will also be fetched. These special `head` refs represent the head of the branch which is requesting to be merged, and the `merge` refs represent the result of the base branch after the merge.

Important: When using custom refspecs, the destination of the fetched refs *must* be under `refs/remotes/origin/`, preferably in a subdirectory like in the example above. These custom refspecs will map as environment names using their relative path underneath `refs/remotes/origin/`. For example, assuming the configuration above, the head branch for pull request 12345 would map to fileserver environment `pr/12345` (slash included).

Refspecs can be configured on a *per-remote basis*. For example, the below configuration would only alter the default refspecs for the *second* GitFS remote. The first remote would only fetch branches and tags (the default).

```
gitfs_remotes:
  - https://domain.tld/foo.git
  - https://domain.tld/bar.git:
    - refspecs:
      - '+refs/heads/*:refs/remotes/origin/*'
      - '+refs/tags/*:refs/tags/*'
      - '+refs/pull/*/head:refs/remotes/origin/pr/*'
      - '+refs/pull/*/merge:refs/remotes/origin/merge/*'
```

3.13.7 Global Remotes

New in version 2018.3.0.

The `all_saltenvs` per-remote configuration parameter overrides the logic Salt uses to map branches/tags to fileserver environments (i.e. saltenvs). This allows a single branch/tag to appear in *all* GitFS saltenvs.

Note: `all_saltenvs` only works *within* GitFS. That is, files in a branch configured using `all_saltenvs` will *not* show up in a fileserver environment defined via some other fileserver backend (e.g. `file_roots`).

This is very useful in particular when working with *salt formulas*. Prior to the addition of this feature, it was necessary to push a branch/tag to the remote repo for each saltenv in which that formula was to be used. If the formula needed to be updated, this update would need to be reflected in all of the other branches/tags. This is both inconvenient and not scalable.

With `all_saltenvs`, it is now possible to define your formula once, in a single branch.

```
gitfs_remotes:
  - http://foo.com/quux.git:
    - all_saltenvs: anything
```

3.13.8 Update Intervals

Prior to the 2018.3.0 release, GitFS would update its fileserver backends as part of a dedicated "maintenance" process, in which various routine maintenance tasks were performed. This tied the update interval to the *`loop_interval`* config option, and also forced all fileservers to update at the same interval.

Now it is possible to make GitFS update at its own interval, using *`gitfs_update_interval`*:

```
gitfs_update_interval: 180

gitfs_remotes:
  - https://foo.com/foo.git
  - https://foo.com/bar.git:
    - update_interval: 120
```

Using the above configuration, the first remote would update every three minutes, while the second remote would update every two minutes.

3.13.9 Configuration Order of Precedence

The order of precedence for GitFS configuration is as follows (each level overrides all levels below it):

1. Per-saltenv configuration (defined under a per-remote `saltenv` param)

```
gitfs_remotes:
  - https://foo.com/bar.git:
    - saltenv:
      - dev:
        - mountpoint: salt://bar
```

2. Global per-saltenv configuration (defined in *`gitfs_saltenv`*)

```
gitfs_saltenv:
  - dev:
    - mountpoint: salt://bar
```

3. Per-remote configuration parameter

```
gitfs_remotes:
  - https://foo.com/bar.git:
    - mountpoint: salt://bar
```

4. Global configuration parameter

```
gitfs_mountpoint: salt://bar
```

Note: The one exception to the above is when *all_saltenvs* is used. This value overrides all logic for mapping branches/tags to fileserver environments. So, even if *`gitfs_saltenv`* is used to globally override the mapping for a given saltenv, *all_saltenvs* would take precedence for any remote which uses it.

It's important to note however that any `root` and `mountpoint` values configured in *`gitfs_saltenv`* (or *per-saltenv configuration*) would be unaffected by this.

3.13.10 Serving from a Subdirectory

The *`gitfs_root`* parameter allows files to be served from a subdirectory within the repository. This allows for only part of a repository to be exposed to the Salt fileserver.

Assume the below layout:

```
.gitignore
README.txt
foo/
foo/bar/
foo/bar/one.txt
foo/bar/two.txt
foo/bar/three.txt
foo/baz/
foo/baz/top.sls
foo/baz/edit/vim.sls
foo/baz/edit/vimrc
foo/baz/nginx/init.sls
```

The below configuration would serve only the files under `foo/baz`, ignoring the other files in the repository:

```
gitfs_remotes:
  - git://mydomain.com/stuff.git

gitfs_root: foo/baz
```

The root can also be configured on a *per-remote basis*.

3.13.11 Mountpoints

New in version 2014.7.0.

The *`gitfs_mountpoint`* parameter will prepend the specified path to the files served from gitfs. This allows an existing repository to be used, rather than needing to reorganize a repository or design it around the layout of the Salt fileserver.

Before the addition of this feature, if a file being served up via gitfs was deeply nested within the root directory (for example, `salt://webapps/foo/files/foo.conf`, it would be necessary to ensure that the file was properly located in the remote repository, and that all of the parent directories were present (for example, the directories `webapps/foo/files/` would need to exist at the root of the repository).

The below example would allow for a file `foo.conf` at the root of the repository to be served up from the Salt fileserver path `salt://webapps/foo/files/foo.conf`.

```
gitfs_remotes:
  - https://mydomain.com/stuff.git

gitfs_mountpoint: salt://webapps/foo/files
```

Mountpoints can also be configured on a *per-remote basis*.

3.13.12 Using gitfs in Masterless Mode

Since 2014.7.0, gitfs can be used in masterless mode. To do so, simply add the gitfs configuration parameters (and set *`fileserver_backend`*) in the _minion_ config file instead of the master config file.

3.13.13 Using gitfs Alongside Other Backends

Sometimes it may make sense to use multiple backends; for instance, if `sls` files are stored in git but larger files are stored directly on the master.

The cascading lookup logic used for multiple remotes is also used with multiple backends. If the *`fileserver_backend`* option contains multiple backends:

```
fileserver_backend:
  - roots
  - git
```

Then the `roots` backend (the default backend of files in `/srv/salt`) will be searched first for the requested file; then, if it is not found on the master, each configured git remote will be searched.

Note: This can be used together with *file_roots* accepting *__env__* as a catch-all environment, since 2018.3.5 and 2019.2.1:

```
file_roots:
  base:
    - /srv/salt
  __env__:
    - /srv/salt
```

3.13.14 Branches, Environments, and Top Files

When using the GitFS backend, branches, and tags will be mapped to environments using the branch/tag name as an identifier.

There is one exception to this rule: the `master` branch is implicitly mapped to the `base` environment.

So, for a typical `base`, `qa`, `dev` setup, the following branches could be used:

```
master
qa
dev
```

`top.sls` files from different branches will be merged into one at runtime. Since this can lead to overly complex configurations, the recommended setup is to have a separate repository, containing only the `top.sls` file with just one single `master` branch.

To map a branch other than `master` as the `base` environment, use the *`gitfs_base`* parameter.

```
gitfs_base: salt-base
```

The base can also be configured on a *per-remote basis*.

3.13.15 Environment Whitelist/Blacklist

New in version 2014.7.0.

The *gitfs_saltenv_whitelist* and *gitfs_saltenv_blacklist* parameters allow for greater control over which branches/tags are exposed as fileserver environments. Exact matches, globs, and regular expressions are supported, and are evaluated in that order. If using a regular expression, ^ and $ must be omitted, and the expression must match the entire branch/tag.

```
gitfs_saltenv_whitelist:
  - base
  - v1.*
  - 'mybranch\d+'
```

Note: `v1.*`, in this example, will match as both a glob and a regular expression (though it will have been matched as a glob, since globs are evaluated before regular expressions).

The behavior of the blacklist/whitelist will differ depending on which combination of the two options is used:

- If only *gitfs_saltenv_whitelist* is used, then **only** branches/tags which match the whitelist will be available as environments
- If only *gitfs_saltenv_blacklist* is used, then the branches/tags which match the blacklist will **not** be available as environments
- If both are used, then the branches/tags which match the whitelist, but do **not** match the blacklist, will be available as environments.

3.13.16 Authentication

pygit2

New in version 2014.7.0.

Both HTTPS and SSH authentication are supported as of version 0.20.3, which is the earliest version of pygit2 supported by Salt for gitfs.

Note: The examples below make use of per-remote configuration parameters, a feature new to Salt 2014.7.0. More information on these can be found *here*.

HTTPS

For HTTPS repositories which require authentication, the username and password can be provided like so:

```
gitfs_remotes:
  - https://domain.tld/myrepo.git:
    - user: git
    - password: mypassword
```

If the repository is served over HTTP instead of HTTPS, then Salt will by default refuse to authenticate to it. This behavior can be overridden by adding an `insecure_auth` parameter:

```
gitfs_remotes:
  - http://domain.tld/insecure_repo.git:
    - user: git
    - password: mypassword
    - insecure_auth: True
```

SSH

SSH repositories can be configured using the `ssh://` protocol designation, or using scp-like syntax. So, the following two configurations are equivalent:

- `ssh://git@github.com/user/repo.git`
- `git@github.com:user/repo.git`

Both *`gitfs_pubkey`* and *`gitfs_privkey`* (or their *per-remote counterparts*) must be configured in order to authenticate to SSH-based repos. If the private key is protected with a passphrase, it can be configured using *`gitfs_passphrase`* (or simply `passphrase` if being configured *per-remote*). For example:

```
gitfs_remotes:
  - git@github.com:user/repo.git:
    - pubkey: /root/.ssh/id_rsa.pub
    - privkey: /root/.ssh/id_rsa
    - passphrase: myawesomepassphrase
```

Finally, the SSH host key must be *added to the known_hosts file*.

Note: There is a known issue with public-key SSH authentication to Microsoft Visual Studio (VSTS) with pygit2. This is due to a bug or lack of support for VSTS in older libssh2 releases. Known working releases include libssh2 1.7.0 and later, and known incompatible releases include 1.5.0 and older. At the time of this writing, 1.6.0 has not been tested.

Since upgrading libssh2 would require rebuilding many other packages (curl, etc.), followed by a rebuild of libgit2 and a reinstall of pygit2, an easier workaround for systems with older libssh2 is to use GitPython with a passphraseless key for authentication.

GitPython

HTTPS

For HTTPS repositories which require authentication, the username and password can be configured in one of two ways. The first way is to include them in the URL using the format `https://<user>:<password>@<url>`, like so:

```
gitfs_remotes:
  - https://git:mypassword@domain.tld/myrepo.git
```

The other way would be to configure the authentication in `~/.netrc`:

```
machine domain.tld
login git
password mypassword
```

If the repository is served over HTTP instead of HTTPS, then Salt will by default refuse to authenticate to it. This behavior can be overridden by adding an `insecure_auth` parameter:

```
gitfs_remotes:
  - http://git:mypassword@domain.tld/insecure_repo.git:
    - insecure_auth: True
```

SSH

Only passphrase-less SSH public key authentication is supported using GitPython. **The auth parameters (pubkey, privkey, etc.) shown in the pygit2 authentication examples above do not work with GitPython**.

```
gitfs_remotes:
  - ssh://git@github.com/example/salt-states.git
```

Since GitPython wraps the git CLI, the private key must be located in `~/.ssh/id_rsa` for the user under which the Master is running, and should have permissions of `0600`. Also, in the absence of a user in the repo URL, GitPython will (just as SSH does) attempt to login as the current user (in other words, the user under which the Master is running, usually `root`).

If a key needs to be used, then `~/.ssh/config` can be configured to use the desired key. Information on how to do this can be found by viewing the manpage for `ssh_config`. Here's an example entry which can be added to the `~/.ssh/config` to use an alternate key for gitfs:

```
Host github.com
    IdentityFile /root/.ssh/id_rsa_gitfs
```

The `Host` parameter should be a hostname (or hostname glob) that matches the domain name of the git repository.

It is also necessary to *add the SSH host key to the known_hosts file*. The exception to this would be if strict host key checking is disabled, which can be done by adding `StrictHostKeyChecking no` to the entry in `~/.ssh/config`

```
Host github.com
    IdentityFile /root/.ssh/id_rsa_gitfs
    StrictHostKeyChecking no
```

However, this is generally regarded as insecure, and is not recommended.

Adding the SSH Host Key to the known_hosts File

To use SSH authentication, it is necessary to have the remote repository's SSH host key in the `~/.ssh/known_hosts` file. If the master is also a minion, this can be done using the `ssh.set_known_host` function:

```
# salt mymaster ssh.set_known_host user=root hostname=github.com
mymaster:
    ----------
    new:
        ----------
        enc:
            ssh-rsa
        fingerprint:
            16:27:ac:a5:76:28:2d:36:63:1b:56:4d:eb:df:a6:48
        hostname:
            |1|OiefWWqOD4kwO3BhoIGa0loR5AA=|BIXVtmcTbPER+68HvXmceodDcfI=
```

(continues on next page)

(continued from previous page)

```
        key:
            ⊠
↪AAAAB3NzaC1yc2EAAAABIwAAAQEAq2A7hRGmdnm9tUDbO9IDSwBK6TbQa+PXYPCPy6rbTrTtw7PHkccKrpp0yVhp5HdEIcKr6p
↪yMf+Se8xhHTvKSCZIFImWwoG6mbUoWf9nzpIoaSjB+weqqUUmpaaasXVal72J+UX2B+2RPW3RcT0eOzQgqlJL3RKrTJvdsjE3J
↪w4yCE6gbODqnTWlg7+wC604ydGXA8VJiS5ap43JXiUFFAaQ==
    old:
        None
    status:
        updated
```

If not, then the easiest way to add the key is to su to the user (usually `root`) under which the salt-master runs and attempt to login to the server via SSH:

```
$ su -
Password:
# ssh github.com
The authenticity of host 'github.com (192.30.252.128)' can't be established.
RSA key fingerprint is 16:27:ac:a5:76:28:2d:36:63:1b:56:4d:eb:df:a6:48.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'github.com,192.30.252.128' (RSA) to the list of known⊠
↪hosts.
Permission denied (publickey).
```

It doesn't matter if the login was successful, as answering `yes` will write the fingerprint to the known_hosts file.

Verifying the Fingerprint

To verify that the correct fingerprint was added, it is a good idea to look it up. One way to do this is to use `nmap`:

```
$ nmap -p 22 github.com --script ssh-hostkey

Starting Nmap 5.51 ( http://nmap.org ) at 2014-08-18 17:47 CDT
Nmap scan report for github.com (192.30.252.129)
Host is up (0.17s latency).
Not shown: 996 filtered ports
PORT     STATE SERVICE
22/tcp   open  ssh
| ssh-hostkey: 1024 ad:1c:08:a4:40:e3:6f:9c:f5:66:26:5d:4b:33:5d:8c (DSA)
|_2048 16:27:ac:a5:76:28:2d:36:63:1b:56:4d:eb:df:a6:48 (RSA)
80/tcp   open  http
443/tcp  open  https
9418/tcp open  git

Nmap done: 1 IP address (1 host up) scanned in 28.78 seconds
```

Another way is to check one's own `known_hosts` file, using this one-liner:

```
$ ssh-keygen -l -f /dev/stdin <<<`ssh-keyscan github.com 2>/dev/null` | awk '{print
↪$2}'
16:27:ac:a5:76:28:2d:36:63:1b:56:4d:eb:df:a6:48
```

Warning: AWS tracks usage of nmap and may flag it as abuse. On AWS hosts, the `ssh-keygen` method is recommended for host key verification.

Note: As of OpenSSH 6.8 the SSH fingerprint is now shown as a base64-encoded SHA256 checksum of the host key. So, instead of the fingerprint looking like `16:27:ac:a5:76:28:2d:36:63:1b:56:4d:eb:df:a6:48`, it would look like `SHA256:nThbg6kXUpJWGl7E1IGOCspRomTxdCARLviKw6E5SY8`.

3.13.17 Refreshing gitfs Upon Push

By default, Salt updates the remote fileserver backends every 60 seconds. However, if it is desirable to refresh quicker than that, the *Reactor System* can be used to signal the master to update the fileserver on each push, provided that the git server is also a Salt minion. There are three steps to this process:

1. On the master, create a file **/srv/reactor/update_fileserver.sls**, with the following contents:

```
update_fileserver:
  runner.fileserver.update
```

2. Add the following reactor configuration to the master config file:

```
reactor:
  - 'salt/fileserver/gitfs/update':
    - /srv/reactor/update_fileserver.sls
```

3. On the git server, add a post-receive hook

   a. If the user executing *git push* is the same as the minion user, use the following hook:

   ```
   #!/usr/bin/env sh
   salt-call event.fire_master update salt/fileserver/gitfs/update
   ```

   b. To enable other git users to run the hook after a *push*, use sudo in the hook script:

   ```
   #!/usr/bin/env sh
   sudo -u root salt-call event.fire_master update salt/fileserver/gitfs/
   ↪update
   ```

4. If using sudo in the git hook (above), the policy must be changed to permit all users to fire the event. Add the following policy to the sudoers file on the git server.

```
Cmnd_Alias SALT_GIT_HOOK = /bin/salt-call event.fire_master update salt/
↪fileserver/gitfs/update
Defaults!SALT_GIT_HOOK !requiretty
ALL ALL=(root) NOPASSWD: SALT_GIT_HOOK
```

The `update` argument right after *`event.fire_master`* in this example can really be anything, as it represents the data being passed in the event, and the passed data is ignored by this reactor.

Similarly, the tag name `salt/fileserver/gitfs/update` can be replaced by anything, so long as the usage is consistent.

The `root` user name in the hook script and sudo policy should be changed to match the user under which the minion is running.

3.13.18 Using Git as an External Pillar Source

The git external pillar (a.k.a. git_pillar) has been rewritten for the 2015.8.0 release. This rewrite brings with it pygit2 support (allowing for access to authenticated repositories), as well as more granular support for per-remote configuration. This configuration schema is detailed *here*.

3.13.19 Why aren't my custom modules/states/etc. syncing to my Minions?

In versions 0.16.3 and older, when using the *git fileserver backend*, certain versions of GitPython may generate errors when fetching, which Salt fails to catch. While not fatal to the fetch process, these interrupt the fileserver update that takes place before custom types are synced, and thus interrupt the sync itself. Try disabling the git fileserver backend in the master config, restarting the master, and attempting the sync again.

This issue is worked around in Salt 0.16.4 and newer.

3.14 MinionFS Backend Walkthrough

New in version 2014.1.0.

Note: This walkthrough assumes basic knowledge of Salt and *cp.push*. To get up to speed, check out the *Salt Walkthrough*.

Sometimes it is desirable to deploy a file located on one minion to one or more other minions. This is supported in Salt, and can be accomplished in two parts:

1. Minion support for pushing files to the master (using *cp.push*)
2. The *minionfs* fileserver backend

This walkthrough will show how to use both of these features.

3.14.1 Enabling File Push

To set the master to accept files pushed from minions, the *file_recv* option in the master config file must be set to `True` (the default is `False`).

```
file_recv: True
```

Note: This change requires a restart of the salt-master service.

3.14.2 Pushing Files

Once this has been done, files can be pushed to the master using the *cp.push* function:

```
salt 'minion-id' cp.push /path/to/the/file
```

This command will store the file in a subdirectory named `minions` under the master's *cachedir*. On most masters, this path will be `/var/cache/salt/master/minions`. Within this directory will be one directory for each minion which has pushed a file to the master, and underneath that the full path to the file on the minion. So, for example, if a minion with an ID of `dev1` pushed a file `/var/log/myapp.log` to the master, it would be saved to `/var/cache/salt/master/minions/dev1/var/log/myapp.log`.

3.14.3 Serving Pushed Files Using MinionFS

While it is certainly possible to add `/var/cache/salt/master/minions` to the master's *file_roots* and serve these files, it may only be desirable to expose files pushed from certain minions. Adding `/var/cache/salt/master/minions/<minion-id>` for each minion that needs to be exposed can be cumbersome and prone to errors.

Enter *minionfs*. This fileserver backend will make files pushed using *cp.push* available to the Salt fileserver, and provides an easy mechanism to restrict which minions' pushed files are made available.

Simple Configuration

To use the *minionfs* backend, add `minionfs` to the list of backends in the *fileserver_backend* configuration option on the master:

```
file_recv: True

fileserver_backend:
  - roots
  - minionfs
```

Note: `minion` also works here. Prior to the 2018.3.0 release, *only* `minion` would work.

Also, as described earlier, `file_recv: True` is needed to enable the master to receive files pushed from minions. As always, changes to the master configuration require a restart of the `salt-master` service.

Files made available via *minionfs* are by default located at `salt://<minion-id>/path/to/file`. Think back to the earlier example, in which `dev1` pushed a file `/var/log/myapp.log` to the master. With *minionfs* enabled, this file would be addressable in Salt at `salt://dev1/var/log/myapp.log`.

If many minions have pushed to the master, this will result in many directories in the root of the Salt fileserver. For this reason, it is recommended to use the *minionfs_mountpoint* config option to organize these files underneath a subdirectory:

```
minionfs_mountpoint: salt://minionfs
```

Using the above mountpoint, the file in the example would be located at `salt://minionfs/dev1/var/log/myapp.log`.

Restricting Certain Minions' Files from Being Available Via MinionFS

A whitelist and blacklist can be used to restrict the minions whose pushed files are available via *minionfs*. These lists can be managed using the *minionfs_whitelist* and *minionfs_blacklist* config options. Click the links for both of them for a detailed explanation of how to use them.

A more complex configuration example, which uses both a whitelist and blacklist, can be found below:

```
file_recv: True

fileserver_backend:
  - roots
  - minionfs

minionfs_mountpoint: salt://minionfs
```

(continues on next page)

(continued from previous page)

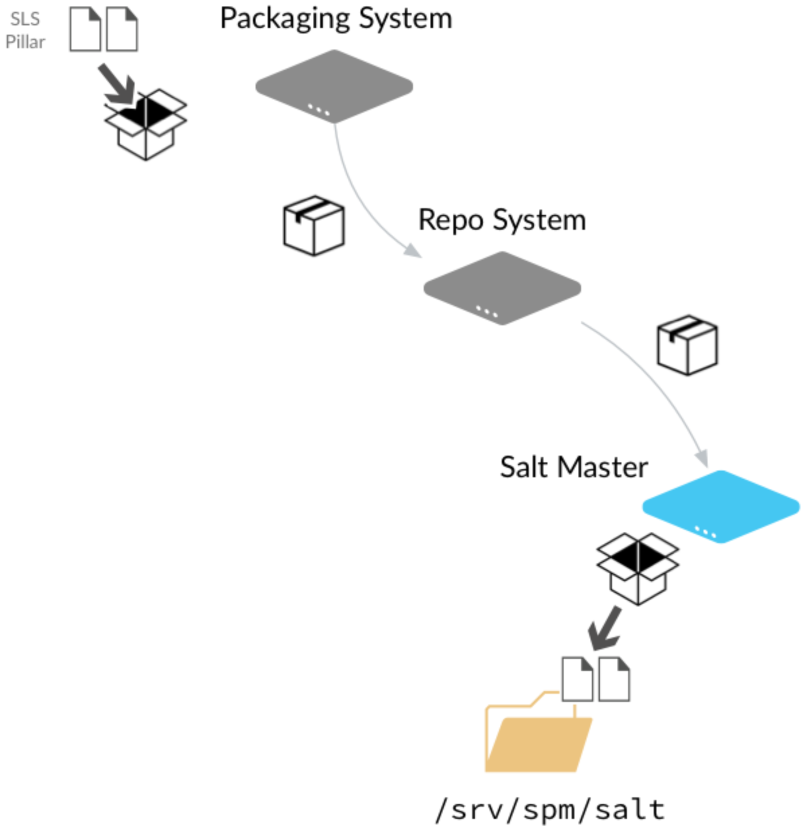
```
minionfs_whitelist:
  - host04
  - web*
  - 'mail\d+\.domain\.tld'

minionfs_blacklist:
  - web21
```

Potential Concerns

- There is no access control in place to restrict which minions have access to files served up by *minionfs*. All minions will have access to these files.
- Unless the *minionfs_whitelist* and/or *minionfs_blacklist* config options are used, all minions which push files to the master will have their files made available via *minionfs*.

3.15 Salt Package Manager

The Salt Package Manager, or *SPM*, enables Salt formulas to be packaged to simplify distribution to Salt masters. The design of SPM was influenced by other existing packaging systems including RPM, Yum, and Pacman.



Note: The previous diagram shows each SPM component as a different system, but this is not required. You can build packages and host the SPM repo on a single Salt master if you'd like.

Packaging System

The packaging system is used to package the state, pillar, file templates, and other files used by your formula into a single file. After a formula package is created, it is copied to the Repository System where it is made available to Salt masters.

See *Building SPM Packages*

Repo System

The Repo system stores the SPM package and metadata files and makes them available to Salt masters via http(s), ftp, or file URLs. SPM repositories can be hosted on a Salt Master, a Salt Minion, or on another system.

See *Distributing SPM Packages*

Salt Master

SPM provides Salt master settings that let you configure the URL of one or more SPM repos. You can then quickly install packages that contain entire formulas to your Salt masters using SPM.

See *Installing SPM Packages*

Contents

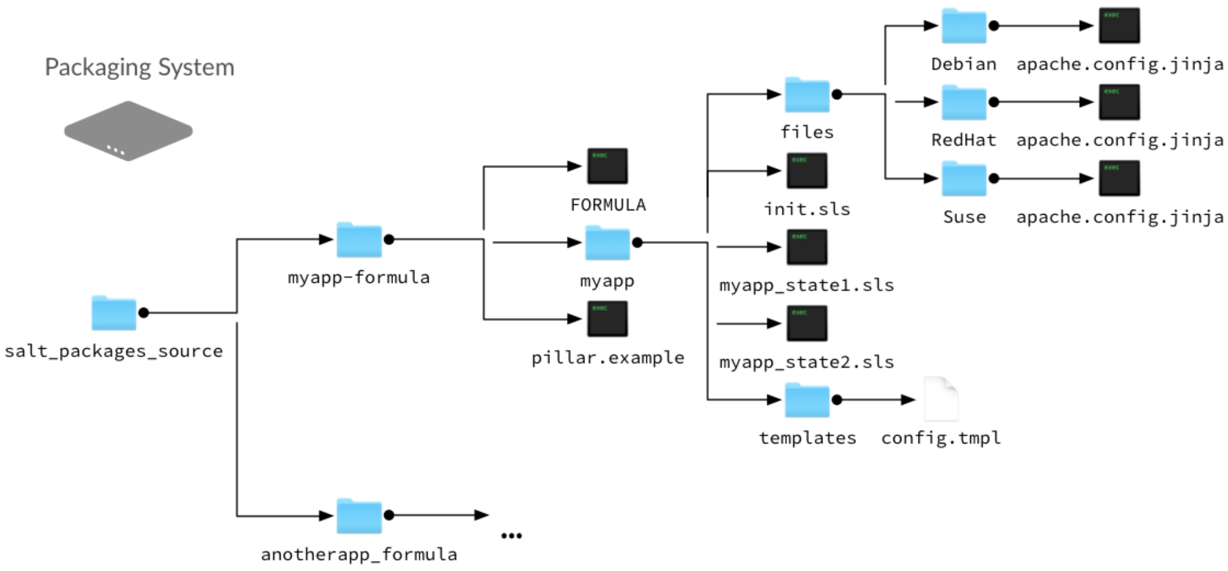
3.15.1 Building SPM Packages

The first step when using Salt Package Manager is to build packages for each of of the formulas that you want to distribute. Packages can be built on any system where you can install Salt.

Package Build Overview

To build a package, all state, pillar, jinja, and file templates used by your formula are assembled into a folder on the build system. These files can be cloned from a Git repository, such as those found at the saltstack-formulas organization on GitHub, or copied directly to the folder.

The following diagram demonstrates a typical formula layout on the build system:


Packaging System
salt_packages_source
myapp-formula
anotherapp_formula
...
FORMULA
myapp
pillar.example
files
init.sls
myapp_state1.sls
myapp_state2.sls
templates
config.tmpl
Debian
RedHat
Suse
apache.config.jinja
apache.config.jinja
apache.config.jinja

In this example, all formula files are placed in a `myapp-formula` folder. This is the folder that is targeted by the `spm build` command when this package is built.

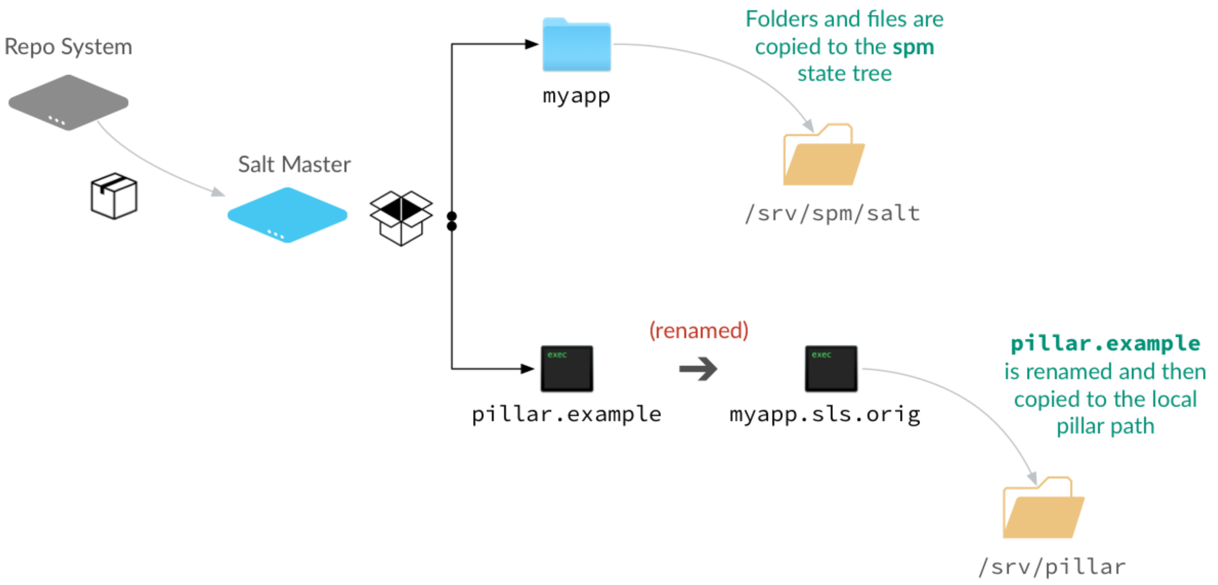
Within this folder, pillar data is placed in a `pillar.example` file at the root, and all state, jinja, and template files are placed within a subfolder that is named after the application being packaged. State files are typically contained within a subfolder, similar to how state files are organized in the state tree. Any non-pillar files in your package that are not contained in a subfolder are placed at the root of the spm state tree.

Additionally, a *FORMULA* file is created and placed in the root of the folder. This file contains package metadata that is used by SPM.

Package Installation Overview

When building packages, it is useful to know where files are installed on the Salt master. During installation, all files except `pillar.example` and `FORMULA` are copied directly to the spm state tree on the Salt master (located at `\srv\spm\salt`).

If a `pillar.example` file is present in the root, it is renamed to `<formula name>.sls.orig` and placed in the `pillar_path`.


Repo System
Salt Master
myapp
Folders and files are copied to the spm state tree
/srv/spm/salt
(renamed)
exec
pillar.example
exec
myapp.sls.orig
pillar.example is renamed and then copied to the local pillar path
/srv/pillar

Note: Even though the pillar data file is copied to the pillar root, you still need to manually assign this pillar data to systems using the pillar top file. This file can also be duplicated and renamed so the `.orig` version is left intact in case you need to restore it later.

Building an SPM Formula Package

1. Assemble formula files in a folder on the build system.
2. Create a *FORMULA* file and place it in the root of the package folder.
3. Run `spm build <folder name>`. The package is built and placed in the `/srv/spm_build` folder.

```
spm build /path/to/salt-packages-source/myapp-formula
```

4. Copy the `.spm` file to a folder on the *repository system*.

Types of Packages

SPM supports different types of packages. The function of each package is denoted by its name. For instance, packages which end in `-formula` are considered to be Salt States (the most common type of formula). Packages which end in `-conf` contain configuration which is to be placed in the `/etc/salt/` directory. Packages which do not contain one of these names are treated as if they have a `-formula` name.

formula

By default, most files from this type of package live in the `/srv/spm/salt/` directory. The exception is the `pillar.example` file, which will be renamed to `<package_name>.sls` and placed in the pillar directory (`/srv/spm/pillar/` by default).

reactor

By default, files from this type of package live in the `/srv/spm/reactor/` directory.

conf

The files in this type of package are configuration files for Salt, which normally live in the `/etc/salt/` directory. Configuration files for packages other than Salt can and should be handled with a Salt State (using a `formula` type of package).

Technical Information

Packages are built using BZ2-compressed tarballs. By default, the package database is stored using the `sqlite3` driver (see Loader Modules below).

Support for these are built into Python, and so no external dependencies are needed.

All other files belonging to SPM use YAML, for portability and ease of use and maintainability.

SPM-Specific Loader Modules

SPM was designed to behave like traditional package managers, which apply files to the filesystem and store package metadata in a local database. However, because modern infrastructures often extend beyond those use cases, certain parts of SPM have been broken out into their own set of modules.

Package Database

By default, the package database is stored using the `sqlite3` module. This module was chosen because support for SQLite3 is built into Python itself.

Please see the SPM Development Guide for information on creating new modules for package database management.

Package Files

By default, package files are installed using the `local` module. This module applies files to the local filesystem, on the machine that the package is installed on.

Please see the *SPM Development Guide* for information on creating new modules for package file management.

3.15.2 Distributing SPM Packages

SPM packages can be distributed to Salt masters over HTTP(S), FTP, or through the file system. The SPM repo can be hosted on any system where you can install Salt. Salt is installed so you can run the `spm create_repo` command when you update or add a package to the repo. SPM repos do not require the salt-master, salt-minion, or any other process running on the system.

Note: If you are hosting the SPM repo on a system where you can not or do not want to install Salt, you can run the `spm create_repo` command on the build system and then copy the packages and the generated `SPM-METADATA` file to the repo. You can also install SPM files *directly on a Salt master*, bypassing the repository completely.

Setting up a Package Repository

After packages are built, the generated SPM files are placed in the `srv/spm_build` folder.

Where you place the built SPM files on your repository server depends on how you plan to make them available to your Salt masters.

You can share the `srv/spm_build` folder on the network, or copy the files to your FTP or Web server.

Adding a Package to the repository

New packages are added by simply copying the SPM file to the repo folder, and then generating repo metadata.

Generate Repo Metadata

Each time you update or add an SPM package to your repository, issue an `spm create_repo` command:

```
spm create_repo /srv/spm_build
```

SPM generates the repository metadata for all of the packages in that directory and places it in an `SPM-METADATA` file at the folder root. This command is used even if repository metadata already exists in that directory.

3.15.3 Installing SPM Packages

SPM packages are installed to your Salt master, where they are available to Salt minions using all of Salt's package management functions.

Configuring Remote Repositories

Before SPM can use a repository, two things need to happen. First, the Salt master needs to know where the repository is through a configuration process. Then it needs to pull down the repository metadata.

Repository Configuration Files

Repositories are configured by adding each of them to the `/etc/salt/spm.repos.d/spm.repo` file on each Salt master. This file contains the name of the repository, and the link to the repository:

```
my_repo:
  url: https://spm.example.com/
```

For HTTP/HTTPS Basic authorization you can define credentials:

```
my_repo:
  url: https://spm.example.com/
  username: user
  password: pass
```

Beware of unauthorized access to this file, please set at least 0640 permissions for this configuration file:

The URL can use `http`, `https`, `ftp`, or `file`.

```
my_repo:
  url: file:///srv/spm_build
```

Updating Local Repository Metadata

After the repository is configured on the Salt master, repository metadata is downloaded using the `spm update_repo` command:

```
spm update_repo
```

Note: A file for each repo is placed in `/var/cache/salt/spm` on the Salt master after you run the *update_repo* command. If you add a repository and it does not seem to be showing up, check this path to verify that the repository was found.

Update File Roots

SPM packages are installed to the `srv/spm/salt` folder on your Salt master. This path needs to be added to the file roots on your Salt master manually.

```
file_roots:
  base:
    1. /srv/salt
    2. /srv/spm/salt
```

Restart the salt-master service after updating the `file_roots` setting.

Installing Packages

To install a package, use the `spm install` command:

```
spm install apache
```

Warning: Currently, SPM does not check to see if files are already in place before installing them. That means that existing files will be overwritten without warning.

Installing directly from an SPM file

You can also install SPM packages using a local SPM file using the `spm local install` command:

```
spm local install /srv/spm/apache-201506-1.spm
```

An SPM repository is not required when using *spm local install*.

Pillars

If an installed package includes Pillar data, be sure to target the installed pillar to the necessary systems using the pillar Top file.

Removing Packages

Packages may be removed after they are installed using the `spm remove` command.

```
spm remove apache
```

If files have been modified, they will not be removed. Empty directories will also be removed.

3.15.4 SPM Configuration

There are a number of options that are specific to SPM. They may be configured in the `master` configuration file, or in SPM's own `spm` configuration file (normally located at `/etc/salt/spm`). If configured in both places, the `spm` file takes precedence. In general, these values will not need to be changed from the defaults.

spm_logfile

Default: `/var/log/salt/spm`

Where SPM logs messages.

spm_repos_config

Default: `/etc/salt/spm.repos`

SPM repositories are configured with this file. There is also a directory which corresponds to it, which ends in `.d`. For instance, if the filename is `/etc/salt/spm.repos`, the directory will be `/etc/salt/spm.repos.d/`.

spm_cache_dir

Default: `/var/cache/salt/spm`

When SPM updates package repository metadata and downloads packaged, they will be placed in this directory. The package database, normally called `packages.db`, also lives in this directory.

spm_db

Default: `/var/cache/salt/spm/packages.db`

The location and name of the package database. This database stores the names of all of the SPM packages installed on the system, the files that belong to them, and the metadata for those files.

spm_build_dir

Default: `/srv/spm_build`

When packages are built, they will be placed in this directory.

spm_build_exclude

Default: `['.git']`

When SPM builds a package, it normally adds all files in the formula directory to the package. Files listed here will be excluded from that package. This option requires a list to be specified.

```
spm_build_exclude:
  - .git
  - .svn
```

Types of Packages

SPM supports different types of formula packages. The function of each package is denoted by its name. For instance, packages which end in `-formula` are considered to be Salt States (the most common type of formula). Packages which end in `-conf` contain configuration which is to be placed in the `/etc/salt/` directory. Packages which do not contain one of these names are treated as if they have a `-formula` name.

formula

By default, most files from this type of package live in the `/srv/spm/salt/` directory. The exception is the `pillar.example` file, which will be renamed to `<package_name>.sls` and placed in the pillar directory (`/srv/spm/pillar/` by default).

reactor

By default, files from this type of package live in the `/srv/spm/reactor/` directory.

conf

The files in this type of package are configuration files for Salt, which normally live in the `/etc/salt/` directory. Configuration files for packages other than Salt can and should be handled with a Salt State (using a `formula` type of package).

3.15.5 FORMULA File

In addition to the formula itself, a `FORMULA` file must exist which describes the package. An example of this file is:

```
name: apache
os: RedHat, Debian, Ubuntu, SUSE, FreeBSD
os_family: RedHat, Debian, Suse, FreeBSD
version: 201506
release: 2
summary: Formula for installing Apache
description: Formula for installing Apache
```

Required Fields

This file must contain at least the following fields:

name

The name of the package, as it will appear in the package filename, in the repository metadata, and the package database. Even if the source formula has `-formula` in its name, this name should probably not include that. For instance, when packaging the `apache-formula`, the name should be set to `apache`.

os

The value of the `os` grain that this formula supports. This is used to help users know which operating systems can support this package.

os_family

The value of the `os_family` grain that this formula supports. This is used to help users know which operating system families can support this package.

version

The version of the package. While it is up to the organization that manages this package, it is suggested that this version is specified in a `YYYYMM` format. For instance, if this version was released in June 2015, the package version should be `201506`. If multiple releases are made in a month, the `release` field should be used.

minimum_version

Minimum recommended version of Salt to use this formula. Not currently enforced.

release

This field refers primarily to a release of a version, but also to multiple versions within a month. In general, if a version has been made public, and immediate updates need to be made to it, this field should also be updated.

summary

A one-line description of the package.

description

A more detailed description of the package which can contain more than one line.

Optional Fields

The following fields may also be present.

top_level_dir

This field is optional, but highly recommended. If it is not specified, the package name will be used.

Formula repositories typically do not store `.sls` files in the root of the repository; instead they are stored in a subdirectory. For instance, an `apache-formula` repository would contain a directory called `apache`, which would contain an `init.sls`, plus a number of other related files. In this instance, the `top_level_dir` should be set to `apache`.

Files outside the `top_level_dir`, such as `README.rst`, `FORMULA`, and `LICENSE` will not be installed. The exceptions to this rule are files that are already treated specially, such as `pillar.example` and `_modules/`.

dependencies

A comma-separated list of packages that must be installed along with this package. When this package is installed, SPM will attempt to discover and install these packages as well. If it is unable to, then it will refuse to install this package.

This is useful for creating packages which tie together other packages. For instance, a package called wordpress-mariadb-apache would depend upon wordpress, mariadb, and apache.

optional

A comma-separated list of packages which are related to this package, but are neither required nor necessarily recommended. This list is displayed in an informational message when the package is installed to SPM.

recommended

A comma-separated list of optional packages that are recommended to be installed with the package. This list is displayed in an informational message when the package is installed to SPM.

files

A files section can be added, to specify a list of files to add to the SPM. Such a section might look like:

```
files:
  - _pillar
  - FORMULA
  - _runners
  - d|mymodule/index.rst
  - r|README.rst
```

When `files` are specified, then only those files will be added to the SPM, regardless of what other files exist in the directory. They will also be added in the order specified, which is useful if you have a need to lay down files in a specific order.

As can be seen in the example above, you may also tag files as being a specific type. This is done by pre-pending a filename with its type, followed by a pipe (`|`) character. The above example contains a document file and a readme. The available file types are:

- `c`: config file
- `d`: documentation file

- `g`: ghost file (i.e. the file contents are not included in the package payload)
- `l`: license file
- `r`: readme file
- `s`: SLS file
- `m`: Salt module

The first 5 of these types (`c`, `d`, `g`, `l`, `r`) will be placed in `/usr/share/salt/spm/` by default. This can be changed by setting an `spm_share_dir` value in your `/etc/salt/spm` configuration file.

The last two types (`s` and `m`) are currently ignored, but they are reserved for future use.

Pre and Post States

It is possible to run Salt states before and after installing a package by using pre and post states. The following sections may be declared in a `FORMULA`:

- `pre_local_state`
- `pre_tgt_state`
- `post_local_state`
- `post_tgt_state`

Sections with `pre` in their name are evaluated before a package is installed and sections with `post` are evaluated after a package is installed. `local` states are evaluated before `tgt` states.

Each of these sections needs to be evaluated as text, rather than as YAML. Consider the following block:

```
pre_local_state: >
  echo test > /tmp/spmtest:
    cmd:
      - run
```

Note that this declaration uses `>` after `pre_local_state`. This is a YAML marker that marks the next multi-line block as text, including newlines. It is important to use this marker whenever declaring `pre` or `post` states, so that the text following it can be evaluated properly.

local States

`local` states are evaluated locally; this is analogous to issuing a state run using a `salt-call --local` command. These commands will be issued on the local machine running the `spm` command, whether that machine is a master or a minion.

`local` states do not require any special arguments, but they must still use the `>` marker to denote that the state is evaluated as text, not a data structure.

```
pre_local_state: >
  echo test > /tmp/spmtest:
    cmd:
      - run
```

tgt States

`tgt` states are issued against a remote target. This is analogous to issuing a state using the `salt` command. As such it requires that the machine that the `spm` command is running on is a master.

Because `tgt` states require that a target be specified, their code blocks are a little different. Consider the following state:

```
pre_tgt_state:
  tgt: '*'
  data: >
    echo test > /tmp/spmtest:
      cmd:
        - run
```

With `tgt` states, the state data is placed under a `data` section, inside the `*_tgt_state` code block. The target is of course specified as a `tgt` and you may also optionally specify a `tgt_type` (the default is `glob`).

You still need to use the > marker, but this time it follows the `data` line, rather than the `*_tgt_state` line.

Templating States

The reason that state data must be evaluated as text rather than a data structure is because that state data is first processed through the rendering engine, as it would be with a standard state run.

This means that you can use Jinja or any other supported renderer inside of Salt. All formula variables are available to the renderer, so you can reference `FORMULA` data inside your state if you need to:

```
pre_tgt_state:
  tgt: '*'
  data: >
     echo {{ name }} > /tmp/spmtest:
      cmd:
        - run
```

You may also declare your own variables inside the `FORMULA`. If SPM doesn't recognize them then it will ignore them, so there are no restrictions on variable names, outside of avoiding reserved words.

By default the renderer is set to `jinja|yaml`. You may change this by changing the `renderer` setting in the `FORMULA` itself.

Building a Package

Once a `FORMULA` file has been created, it is placed into the root of the formula that is to be turned into a package. The `spm build` command is used to turn that formula into a package:

```
spm build /path/to/saltstack-formulas/apache-formula
```

The resulting file will be placed in the build directory. By default this directory is located at `/srv/spm/`.

Loader Modules

When an execution module is placed in `<file_roots>/_modules/` on the master, it will automatically be synced to minions, the next time a sync operation takes place. Other modules are also propagated this way: state modules can be placed in `_states/`, and so on.

When SPM detects a file in a package which resides in one of these directories, that directory will be placed in `<file_roots>` instead of in the formula directory with the rest of the files.

Removing Packages

Packages may be removed once they are installed using the `spm remove` command.

```
spm remove apache
```

If files have been modified, they will not be removed. Empty directories will also be removed.

Technical Information

Packages are built using BZ2-compressed tarballs. By default, the package database is stored using the `sqlite3` driver (see Loader Modules below).

Support for these are built into Python, and so no external dependencies are needed.

All other files belonging to SPM use YAML, for portability and ease of use and maintainability.

SPM-Specific Loader Modules

SPM was designed to behave like traditional package managers, which apply files to the filesystem and store package metadata in a local database. However, because modern infrastructures often extend beyond those use cases, certain parts of SPM have been broken out into their own set of modules.

Package Database

By default, the package database is stored using the `sqlite3` module. This module was chosen because support for SQLite3 is built into Python itself.

Please see the SPM Development Guide for information on creating new modules for package database management.

Package Files

By default, package files are installed using the `local` module. This module applies files to the local filesystem, on the machine that the package is installed on.

Please see the *SPM Development Guide* for information on creating new modules for package file management.

Types of Packages

SPM supports different types of formula packages. The function of each package is denoted by its name. For instance, packages which end in `-formula` are considered to be Salt States (the most common type of formula). Packages which end in `-conf` contain configuration which is to be placed in the `/etc/salt/` directory. Packages which do not contain one of these names are treated as if they have a `-formula` name.

formula

By default, most files from this type of package live in the `/srv/spm/salt/` directory. The exception is the `pillar.example` file, which will be renamed to `<package_name>.sls` and placed in the pillar directory (`/srv/spm/pillar/` by default).

reactor

By default, files from this type of package live in the `/srv/spm/reactor/` directory.

conf

The files in this type of package are configuration files for Salt, which normally live in the `/etc/salt/` directory. Configuration files for packages other than Salt can and should be handled with a Salt State (using a `formula` type of package).

3.15.6 SPM Development Guide

This document discusses developing additional code for SPM.

SPM-Specific Loader Modules

SPM was designed to behave like traditional package managers, which apply files to the filesystem and store package metadata in a local database. However, because modern infrastructures often extend beyond those use cases, certain parts of SPM have been broken out into their own set of modules.

Each function that accepts arguments has a set of required and optional arguments. Take note that SPM will pass all arguments in, and therefore each function must accept each of those arguments. However, arguments that are marked as required are crucial to SPM's core functionality, while arguments that are marked as optional are provided as a benefit to the module, if it needs to use them.

Package Database

By default, the package database is stored using the `sqlite3` module. This module was chosen because support for SQLite3 is built into Python itself.

Modules for managing the package database are stored in the `salt/spm/pkgdb/` directory. A number of functions must exist to support database management.

init()

Get a database connection, and initialize the package database if necessary.

This function accepts no arguments. If a database is used which supports a connection object, then that connection object is returned. For instance, the `sqlite3` module returns a `connect()` object from the `sqlite3` library:

```
conn = sqlite3.connect(__opts__['spm_db'], isolation_level=None)
...
return conn
```

SPM itself will not use this connection object; it will be passed in as-is to the other functions in the module. Therefore, when you set up this object, make sure to do so in a way that is easily usable throughout the module.

info()

Return information for a package. This generally consists of the information that is stored in the `FORMULA` file in the package.

The arguments that are passed in, in order, are `package` (required) and `conn` (optional).

`package` is the name of the package, as specified in the `FORMULA`. `conn` is the connection object returned from `init()`.

list_files()

Return a list of files for an installed package. Only the filename should be returned, and no other information.

The arguments that are passed in, in order, are `package` (required) and `conn` (optional).

`package` is the name of the package, as specified in the `FORMULA`. `conn` is the connection object returned from `init()`.

register_pkg()

Register a package in the package database. Nothing is expected to be returned from this function.

The arguments that are passed in, in order, are `name` (required), `formula_def` (required), and `conn` (optional).

`name` is the name of the package, as specified in the `FORMULA`. `formula_def` is the contents of the `FORMULA` file, as a `dict`. `conn` is the connection object returned from `init()`.

register_file()

Register a file in the package database. Nothing is expected to be returned from this function.

The arguments that are passed in are `name` (required), `member` (required), `path` (required), `digest` (optional), and `conn` (optional).

`name` is the name of the package.

`member` is a `tarfile` object for the package file. It is included, because it contains most of the information for the file.

`path` is the location of the file on the local filesystem.

`digest` is the SHA1 checksum of the file.

`conn` is the connection object returned from `init()`.

unregister_pkg()

Unregister a package from the package database. This usually only involves removing the package's record from the database. Nothing is expected to be returned from this function.

The arguments that are passed in, in order, are `name` (required) and `conn` (optional).

`name` is the name of the package, as specified in the `FORMULA`. `conn` is the connection object returned from `init()`.

unregister_file()

Unregister a package from the package database. This usually only involves removing the package's record from the database. Nothing is expected to be returned from this function.

The arguments that are passed in, in order, are `name` (required), `pkg` (optional) and `conn` (optional).

`name` is the path of the file, as it was installed on the filesystem.

`pkg` is the name of the package that the file belongs to.

`conn` is the connection object returned from `init()`.

db_exists()

Check to see whether the package database already exists. This is the path to the package database file. This function will return `True` or `False`.

The only argument that is expected is `db_`, which is the package database file.

Package Files

By default, package files are installed using the `local` module. This module applies files to the local filesystem, on the machine that the package is installed on.

Modules for managing the package database are stored in the `salt/spm/pkgfiles/` directory. A number of functions must exist to support file management.

init()

Initialize the installation location for the package files. Normally these will be directory paths, but other external destinations such as databases can be used. For this reason, this function will return a connection object, which can be a database object. However, in the default `local` module, this object is a dict containing the paths. This object will be passed into all other functions.

Three directories are used for the destinations: `formula_path`, `pillar_path`, and `reactor_path`.

`formula_path` is the location of most of the files that will be installed. The default is specific to the operating system, but is normally `/srv/salt/`.

`pillar_path` is the location that the `pillar.example` file will be installed to. The default is specific to the operating system, but is normally `/srv/pillar/`.

`reactor_path` is the location that reactor files will be installed to. The default is specific to the operating system, but is normally `/srv/reactor/`.

check_existing()

Check the filesystem for existing files. All files for the package will be checked, and if any are existing, then this function will normally state that SPM will refuse to install the package.

This function returns a list of the files that exist on the system.

The arguments that are passed into this function are, in order: `package` (required), `pkg_files` (required), `formula_def` (formula_def), and `conn` (optional).

`package` is the name of the package that is to be installed.

`pkg_files` is a list of the files to be checked.

`formula_def` is a copy of the information that is stored in the `FORMULA` file.

`conn` is the file connection object.

install_file()

Install a single file to the destination (normally on the filesystem). Nothing is expected to be returned from this function.

This function returns the final location that the file was installed to.

The arguments that are passed into this function are, in order, `package` (required), `formula_tar` (required), `member` (required), `formula_def` (required), and `conn` (optional).

`package` is the name of the package that is to be installed.

`formula_tar` is the tarfile object for the package. This is passed in so that the function can call `formula_tar.extract()` for the file.

`member` is the tarfile object which represents the individual file. This may be modified as necessary, before being passed into `formula_tar.extract()`.

`formula_def` is a copy of the information from the `FORMULA` file.

`conn` is the file connection object.

remove_file()

Remove a single file from file system. Normally this will be little more than an `os.remove()`. Nothing is expected to be returned from this function.

The arguments that are passed into this function are, in order, `path` (required) and `conn` (optional).

`path` is the absolute path to the file to be removed.

`conn` is the file connection object.

hash_file()

Returns the hexdigest hash value of a file.

The arguments that are passed into this function are, in order, `path` (required), `hashobj` (required), and `conn` (optional).

`path` is the absolute path to the file.

`hashobj` is a reference to `hashlib.sha1()`, which is used to pull the `hexdigest()` for the file.

`conn` is the file connection object.

This function will not generally be more complex than:

```
def hash_file(path, hashobj, conn=None):
    with salt.utils.files.fopen(path, 'r') as f:
        hashobj.update(f.read())
        return hashobj.hexdigest()
```

path_exists()

Check to see whether the file already exists on the filesystem. Returns `True` or `False`.

This function expects a `path` argument, which is the absolute path to the file to be checked.

path_isdir()

Check to see whether the path specified is a directory. Returns `True` or `False`.

This function expects a `path` argument, which is the absolute path to be checked.

3.16 Storing Data in Other Databases

The SDB interface is designed to store and retrieve data that, unlike pillars and grains, is not necessarily minion-specific. The initial design goal was to allow passwords to be stored in a secure database, such as one managed by the keyring package, rather than as plain-text files. However, as a generic database interface, it could conceptually be used for a number of other purposes.

SDB was added to Salt in version 2014.7.0.

3.16.1 SDB Configuration

In order to use the SDB interface, a configuration profile must be set up. To be available for master commands, such as runners, it needs to be configured in the master configuration. For modules executed on a minion, it can be set either in the minion configuration file, or as a pillar. The configuration stanza includes the name/ID that the profile will be referred to as, a `driver` setting, and any other arguments that are necessary for the SDB module that will be used. For instance, a profile called `mykeyring`, which uses the `system` service in the `keyring` module would look like:

```
mykeyring:
  driver: keyring
  service: system
```

It is recommended to keep the name of the profile simple, as it is used in the SDB URI as well.

3.16.2 SDB URIs

SDB is designed to make small database queries (hence the name, SDB) using a compact URL. This allows users to reference a database value quickly inside a number of Salt configuration areas, without a lot of overhead. The basic format of an SDB URI is:

```
sdb://<profile>/<args>
```

The profile refers to the configuration profile defined in either the master or the minion configuration file. The args are specific to the module referred to in the profile, but will typically only need to refer to the key of a key/value pair inside the database. This is because the profile itself should define as many other parameters as possible.

For example, a profile might be set up to reference credentials for a specific OpenStack account. The profile might look like:

```
kevinopenstack:
  driver: keyring
  service: salt.cloud.openstack.kevin
```

And the URI used to reference the password might look like:

```
sdb://kevinopenstack/password
```

3.16.3 Getting, Setting and Deleting SDB Values

Once an SDB driver is configured, you can use the `sdb` execution module to get, set and delete values from it. There are two functions that may appear in most SDB modules: `get`, `set` and `delete`.

Getting a value requires only the SDB URI to be specified. To retrieve a value from the `kevinopenstack` profile above, you would use:

```
salt-call sdb.get sdb://kevinopenstack/password
```

Warning: The `vault` driver previously only supported splitting the path and key with a question mark. This has since been deprecated in favor of using the standard / to split the path and key. The use of the questions mark will still be supported to ensure backwards compatibility, but please use the prefered method using /. The deprecated approach required the full path to where the key is stored, followed by a question mark, followed by the key to be retrieved. If you were using a profile called `myvault`, you would use a URI that looks like:

```
salt-call sdb.get 'sdb://myvault/secret/salt?saltstack'
```

Instead of the above please use the prefered URI using / instead:

```
salt-call sdb.get 'sdb://myvault/secret/salt/saltstack'
```

Setting a value uses the same URI as would be used to retrieve it, followed by the value as another argument.

```
salt-call sdb.set 'sdb://myvault/secret/salt/saltstack' 'super awesome'
```

Deleting values (if supported by the driver) is done pretty much the same way as getting them. Provided that you have a profile called `mykvstore` that uses a driver allowing to delete values you would delete a value as shown below:

```
salt-call sdb.delete 'sdb://mykvstore/foobar'
```

The `sdb.get`, `sdb.set` and `sdb.delete` functions are also available in the runner system:

```
salt-run sdb.get 'sdb://myvault/secret/salt/saltstack'
salt-run sdb.set 'sdb://myvault/secret/salt/saltstack' 'super awesome'
salt-run sdb.delete 'sdb://mykvstore/foobar'
```

3.16.4 Using SDB URIs in Files

SDB URIs can be used in both configuration files, and files that are processed by the renderer system (jinja, mako, etc.). In a configuration file (such as `/etc/salt/master`, `/etc/salt/minion`, `/etc/salt/cloud`, etc.), make an entry as usual, and set the value to the SDB URI. For instance:

```
mykey: sdb://myetcd/mykey
```

To retrieve this value using a module, the module in question must use the `config.get` function to retrieve configuration values. This would look something like:

```
mykey = __salt__['config.get']('mykey')
```

Templating renderers use a similar construct. To get the `mykey` value from above in Jinja, you would use:

```
{{ salt['config.get']('mykey') }}
```

When retrieving data from configuration files using `config.get`, the SDB URI need only appear in the configuration file itself.

If you would like to retrieve a key directly from SDB, you would call the `sdb.get` function directly, using the SDB URI. For instance, in Jinja:

```
{{ salt['sdb.get']('sdb://myetcd/mykey') }}
```

When writing Salt modules, it is not recommended to call `sdb.get` directly, as it requires the user to provide values in SDB, using a specific URI. Use `config.get` instead.

3.16.5 Writing SDB Modules

There is currently one function that MUST exist in any SDB module (`get()`), one that SHOULD exist (`set_()`) and one that MAY exist (`delete()`). If using a (`set_()`) function, a `__func_alias__` dictionary MUST be declared in the module as well:

```
__func_alias__ = {
    'set_': 'set',
}
```

This is because `set` is a Python built-in, and therefore functions should not be created which are called `set()`. The `__func_alias__` functionality is provided via Salt's loader interfaces, and allows legally-named functions to be referred to using names that would otherwise be unwise to use.

The `get()` function is required, as it will be called via functions in other areas of the code which make use of the `sdb://` URI. For example, the `config.get` function in the `config` execution module uses this function.

The `set_()` function may be provided, but is not required, as some sources may be read-only, or may be otherwise unwise to access via a URI (for instance, because of SQL injection attacks).

The `delete()` function may be provided as well, but is not required, as many sources may be read-only or restrict such operations.

A simple example of an SDB module is `salt/sdb/keyring_db.py`, as it provides basic examples of most, if not all, of the types of functionality that are available not only for SDB modules, but for Salt modules in general.

3.17 Running the Salt Master/Minion as an Unprivileged User

While the default setup runs the master and minion as the root user, some may consider it an extra measure of security to run the master as a non-root user. Keep in mind that doing so does not change the master's capability to access minions as the user they are running as. Due to this many feel that running the master as a non-root user does not grant any real security advantage which is why the master has remained as root by default.

Note: Some of Salt's operations cannot execute correctly when the master is not running as root, specifically the pam external auth system, as this system needs root access to check authentication.

As of Salt 0.9.10 it is possible to run Salt as a non-root user. This can be done by setting the *`user`* parameter in the master configuration file. and restarting the `salt-master` service.

The minion has it's own *`user`* parameter as well, but running the minion as an unprivileged user will keep it from making changes to things like users, installed packages, etc. unless access controls (sudo, etc.) are setup on the minion to permit the non-root user to make the needed changes.

In order to allow Salt to successfully run as a non-root user, ownership, and permissions need to be set such that the desired user can read from and write to the following directories (and their subdirectories, where applicable):

- /etc/salt
- /var/cache/salt
- /var/log/salt
- /var/run/salt

Ownership can be easily changed with `chown`, like so:

```
# chown -R user /etc/salt /var/cache/salt /var/log/salt /var/run/salt
```

Warning: Running either the master or minion with the *`root_dir`* parameter specified will affect these paths, as will setting options like *`pki_dir`*, *`cachedir`*, *`log_file`*, and other options that normally live in the above directories.

3.18 Using cron with Salt

The Salt Minion can initiate its own *highstate* using the `salt-call` command.

```
$ salt-call state.apply
```

This will cause the minion to check in with the master and ensure it is in the correct "state".

3.19 Use cron to initiate a highstate

If you would like the Salt Minion to regularly check in with the master you can use cron to run the `salt-call` command:

```
0 0 * * * salt-call state.apply
```

The above cron entry will run a *highstate* every day at midnight.

Note: When executing Salt using cron, keep in mind that the default PATH for cron may not include the path for any scripts or commands used by Salt, and it may be necessary to set the PATH accordingly in the crontab:

```
PATH=/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin:/opt/bin

0 0 * * * salt-call state.apply
```

3.20 Hardening Salt

This topic contains tips you can use to secure and harden your Salt environment. How you best secure and harden your Salt environment depends heavily on how you use Salt, where you use Salt, how your team is structured, where you get data from, and what kinds of access (internal and external) you require.

Warning: For historical reasons, Salt requires PyCrypto as a "lowest common denominator". However, PyCrypto is unmaintained and best practice is to manually upgrade to use a more maintained library such as PyCryptodome. See Issue #52674 and Issue #54115 for more info

3.20.1 General hardening tips

- Restrict who can directly log into your Salt master system.
- Use SSH keys secured with a passphrase to gain access to the Salt master system.
- Track and secure SSH keys and any other login credentials you and your team need to gain access to the Salt master system.
- Use a hardened bastion server or a VPN to restrict direct access to the Salt master from the internet.
- Don't expose the Salt master any more than what is required.
- Harden the system as you would with any high-priority target.
- Keep the system patched and up-to-date.
- Use tight firewall rules. Pay particular attention to TCP/4505 and TCP/4506 on the salt master and avoid exposing these ports unnecessarily.

3.20.2 Salt hardening tips

- Subscribe to salt-users or salt-announce so you know when new Salt releases are available. Keep your systems up-to-date with the latest patches.
- Use Salt's Client *ACL system* to avoid having to give out root access in order to run Salt commands.
- Use Salt's Client *ACL system* to restrict which users can run what commands.
- Use *external Pillar* to pull data into Salt from external sources so that non-sysadmins (other teams, junior admins, developers, etc) can provide configuration data without needing access to the Salt master.

- Make heavy use of SLS files that are version-controlled and go through a peer-review/code-review process before they're deployed and run in production. This is good advice even for "one-off" CLI commands because it helps mitigate typos and mistakes.
- Use salt-api, SSL, and restrict authentication with the *external auth* system if you need to expose your Salt master to external services.
- Make use of Salt's event system and *reactor* to allow minions to signal the Salt master without requiring direct access.
- Run the `salt-master` daemon as non-root.
- Disable which modules are loaded onto minions with the *`disable_modules`* setting. (for example, disable the `cmd` module if it makes sense in your environment.)
- Look through the fully-commented sample *master* and *minion* config files. There are many options for securing an installation.
- Run *masterless-mode* minions on particularly sensitive minions. There is also *Salt SSH* or the `modules.sudo` if you need to further restrict a minion.
- Monitor specific security releated log messages. Salt `salt-master` logs attempts to access methods which are not exposed to network clients. These log messages are logged at the `error` log level and start with `Requested method not exposed`.

3.21 Security disclosure policy

email security@saltstack.com

gpg key ID 4EA0793D

gpg key fingerprint `8ABE 4EFC F0F4 B24B FF2A AF90 D570 F2D3 4EA0 793D`

gpg public key:

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBFO15mMBEADa3CfQwk5ED9wAQ8fFDku277CegG3U1hVGdcxqKNvucblwoKCb
hRK6u9ihgaO9V9duV2glwgjytiBI/z6lyWqdaD37YXG/gTL+9Md+qdSDeaOa/9eg
7y+g4P+FvU9HWUlujRVlofUn5Dj/IZgUywbxwEybutuzvvFVTzsn+DFVwTH34Qoh
QIuNzQCSEz3Lhh8zq9LqkNy91ZZQO1ZIUrypafspH6GBHHcE8msBFgYiNBnVcUFH
u0r4j1Rav+621EtD5GZsOt05+NJI8pkaC/dDKjURcuiV6bhmeSpNzLaXUhwx6f29
Vhag5JhVGGNQxlRTxNEM86HEFp+4zJQ8m/wRDrGX5IAHsdESdhP+ljDVlAAX/ttP
/Ucl2fgpTnDKVHOA00E515Q87ZHv6awJ3GL1veqi8zfsLaag7rw1TuuHyGLOPkDt
t5PAjsS9R3KI7pGnhqI6bTOi591odUdgzUhZChWUUX1VStiIDi2jCvyoOOLMOGS5
AEYXuWYP7KgujZCDRaTNqRDdgPd93Mh9JI8UmkzXDUgijdzVpzPjYgFaWtyK8lsc
Fizqe3/Yzf9RCVX/lmRbiEH+ql/zSxcWlBQd17PKaL+TisQFXcmQzccYgAxFbj2r
QHp5ABEu9YjFme2Jzun7Mv9V4qo3JF5dmnUk31yupZeAOGZkirIsaWC3hwARAQAB
tDBTYWx0U3RhY2sgU2VjdXJpdHkgVGVhbSA8c2VjdXJpdHlAc2FsdHN0YWNrLmNv
bT6JAj4EEwECACgFAlO15mMCGwMFCQeGH4AGCwkIBwMCBhUIAgkKCwQWAgMBAh4B
AheAAAoJENVw8tNOoHk9z/MP/2vzY27fmVxU5X8joiiturjlgEqQw41IYEmWv1Bw
4WVXYCHP1yu/1MC1uuvOmOd5BlI8YO2C2oyW7d1B0NorguPtz55b7jabCElekVCh
h/H4ZVThiwqgPpthRv/2npXjIm7SLSs/kuaXo6Qy2JpszwDVFw+xCRVL0tH9KJxz
HuNBeVq7abWD5fzIWkmGM9hicG/R2D0RIlco1Q0VNKy8klG+pOFOW886KnwkSPc7
JUYp1oUlHsSlhTmkLEG54cyVzrTP/XuZuyMTdtyTc3mfgW0adneAL6MARtC5UB/h
q+v9dqMf4iD3wY6ctu8KWE8Vo5MUEsNNO9EA2dUR88LwFZ3ZnnXdQkizgR/Aa515
dm17vlNkSoomYCo84eN7GOTfxWcq+iXYSWcKWT4X+h/ra+LmNndQWQBRebVUtbKE
ZDwKmiQz/5LY5EhlWcuU4lVmMSFpWXt5FR/PtzgTdZAo9QKkBjcv97LYbXvsPI69
El1BLAg+m+1UpE1L7zJT1il6PqVyEFAWBxW46wXCCkGssFsvz2yRp0PDX8A6u4yq
```

(continues on next page)

(continued from previous page)

```
rTkt09uYht1is61joLDJ/kq3+6k8gJWkDOW+2NMrmf+/qcdYCMYXmrtOpg/wF27W
GMNAkbdyzgeX/MbUBCGCMdzhevRuivOI5bu4vT5s3KdshG+yhzV45bapKRd5VN+1
mZRqiQJVBBMBAgA/AhsDBgsJCAcDAgYVCAIJCgsEFgIDAQIeAQIXgBYhBIq+Tvzw
9LJL/yqvkNVw8tNOoHk9BQJb0e5rBQkL3m8IAAoJENVw8tNOoHk9fzMP/ApQtkQD
BmoYEBTF6BH1bywzDw5OHpnBSLbuoYtA3gkhnm/83MzFDcGn22pgo2Fv0MuHltWI
G2oExzje7szmcM6Xg3ZTKapJ3/p2J+P33tkJA1LWpg+DdgdQlqrjlXKwEnikszuB
9IMhbjoPeBzwiUtsBQmcwbVgwMzbscwoV5DJ/gLDCkgF4rp2uKEYAcBi8s9NGX6p
zQsb9Sb0/bKdCrszAcvUn4WYB6WbAPttvutYHtg/nZfXEeX/SgBueXo3lO9vzFlO
r3Zgk7WeucsEqa9Qo0VLOq28HykixM5mEJKsAQrNIqM1DqXgfDch8RJAHzgMBHFH
Qi9hJXk1/6OA2FPXQGcA9Td5Dt0i1Z7wMrAUMj3s9gNMVCD0hQqEKfUtpyV7KBAj
AO5j8Wr8KafnRm6czBCkcV0SRzHQSHdYyncozWwPgWOaRC9AY9fEDz8lBaSoB/C+
dyO/xZMTWoaWqkHozVoHIrCc4CAtZTye/5mxFhq15Q1Iy/NjelrMTCD1kql1dNIP
oOgfOYl1xLMQIBwrrCrgeRIvxEgKRf9KOLbSrS7+3vOKoxf+LD4AQfLci8dFyH+I
t0Z43nk93yTOI82RTdz5GwUXIKcvGhsJ8bgNlGTxM1R/Sl8Sg8diE2PRAp/fk7+g
CwOM8VkeyrDM2k1cy64d8USkbR7YtT3otyFQuQINBFO15mMBEAC5UuLii9ZLz6qH
fIJp35IOW9U8SOf7QFhzXR7NZ3DmJsd3f6Nb/habQFIHjm3K9wbpj+FvaW2oWRlF
VvYdzjUq6c82GUUjW1dnqgUvFwdmM8351n0YQ2TonmyaF882RvsRZrbJ65uvy7SQ
xlouXaAYOdqwLsPxBEOyOnMPSktW5V2UIWyxsNP3sADchWIGq9p5D3Y/loyIMsS1
dj+TjoQZOKSj7CuRT98+8yhGAY8YBEXu9r3I9o6mDkuPpAljuMc8r09Im6az2egt
K/szKt4Hy1bpSSBZU4W/XR7XwQNywmb3wxjmYT6Od3Mwj0jtzc3gQiH8hcEy3+BO
+NNmyzFVyIwOLziwjmEcw62S57wYKUVnHD2nglMsQa8Ve0e6ABBMEY7zGEGStva5
9rfgeh0jUMJiccGiUDTMs0tdkC6knYKbu/fdRqNYFoNuDcSeLEw4DdCuP01l2W4y
Y+fiK6hAcL25amjzc+yYo9eaaqTn6RATbzdhHQZdpAMxY+vNT0+NhP1Zo5gYBMR6
5Zp/VhFsf67ijb03FUtdw9N8dHwiR2m8vVA8kO/gCD6wS2p9RdXqrJ9JhnHYWjiV
uXR+f755ZAndyQfRtowMdQIoiXuJEXYw6XN+/BX81gJaynJYc0uw0MnxWQX+A5m8
HqEsbIFUXBYXPgbwXTm7c4IHGgXXdwARAQABiQI8BBgBAgAmAhsMFiEEir5O/PD0
skv/Kq+Q1XDy006geT0FAlvR7oMFCQvebyAACgkQ1XDy006geT2Hxw//Zha8j8Uc
4B+DmHhZIvPmHp9aFI4DWhC7CBDrYKztBz42H6eX+UsBu4p+uBDKdW9xJH+Qt/zF
nf/zB5Bhc/wFceVRCAkWxPdiIQeo5XQGjZeORjle7E9iunTko+5q1q9I7IgqWYrn
jRmulDvRhO7AoUrqGACDrV6t0F1/XPB8seR2i6axFmFlt1qBHasRq11yksdgNYiD
KXaovf7csDGPGOCWEKMX7BFGpdK/dWdNYfH0Arfom0U5TqNfvGtP4yRPx2bcs7/1
VXPj7IqhBgOtA9pwtMjFki8HGkqj7bB2ErFBOnSwqqNnNcbnhiO6D74SHVGAHhKZ
whaMPDg76EvjAezoLHg7KWYOyUkWJSLa+YoM9r4+PJuEuW/XuaZCNbrAhek+p3pD
ywhElvZe/2UFk619qKzwSbTzk7a90rxLQ2wwtd0vxAW/GyjWl4/kOMZhI5+LAk1l
REucE0fSQxzCTeXu2ObvFR9ic02IYGH3Koz8CrGReEI1J05041Y5IhKxdsvGOD2W
e7ymcblYW4Gz8eYFlLeNJkj/38R7qmNZ028XHzAZDCAWDiTFrnCoglyk+U0JRHfg
HTsdvoc8mBdT/s24LhnfAbpLizlrZZquuOF6NLQSkbuLtmIwf+h9ynEEJxEkGGWg
7JqB1tMjNHLkRpveO/DTYB+iffpba1nCguk=
=OgRY
-----END PGP PUBLIC KEY BLOCK-----
```

The SaltStack Security Team is available at security@saltstack.com for security-related bug reports or questions.

We request the disclosure of any security-related bugs or issues be reported non-publicly until such time as the issue can be resolved and a security-fix release can be prepared. At that time we will release the fix and make a public announcement with upgrade instructions and download locations.

3.21.1 Security response procedure

SaltStack takes security and the trust of our customers and users very seriously. Our disclosure policy is intended to resolve security issues as quickly and safely as is possible.

1. A security report sent to security@saltstack.com is assigned to a team member. This person is the primary contact for questions and will coordinate the fix, release, and announcement.
2. The reported issue is reproduced and confirmed. A list of affected projects and releases is made.

3. Fixes are implemented for all affected projects and releases that are actively supported. Back-ports of the fix are made to any old releases that are actively supported.
4. Packagers are notified via the salt-packagers mailing list that an issue was reported and resolved, and that an announcement is incoming.
5. A new release is created and pushed to all affected repositories. The release documentation provides a full description of the issue, plus any upgrade instructions or other relevant details.
6. An announcement is made to the salt-users and salt-announce mailing lists. The announcement contains a description of the issue and a link to the full release documentation and download locations.

3.21.2 Receiving security announcements

The fastest place to receive security announcements is via the salt-announce mailing list. This list is low-traffic.

3.22 Salt Transport

One of fundamental features of Salt is remote execution. Salt has two basic "channels" for communicating with minions. Each channel requires a client (minion) and a server (master) implementation to work within Salt. These pairs of channels will work together to implement the specific message passing required by the channel interface.

3.22.1 Pub Channel

The pub channel, or publish channel, is how a master sends a job (payload) to a minion. This is a basic pub/sub paradigm, which has specific targeting semantics. All data which goes across the publish system should be encrypted such that only members of the Salt cluster can decrypt the publishes.

3.22.2 Req Channel

The req channel is how the minions send data to the master. This interface is primarily used for fetching files and returning job returns. The req channels have two basic interfaces when talking to the master. `send` is the basic method that guarantees the message is encrypted at least so that only minions attached to the same master can read it– but no guarantee of minion-master confidentiality, whereas the `crypted_transfer_decode_dictentry` method does guarantee minion-master confidentiality.

Zeromq Transport

Note: Zeromq is the current default transport within Salt

Zeromq is a messaging library with bindings into many languages. Zeromq implements a socket interface for message passing, with specific semantics for the socket type.

Pub Channel

The pub channel is implemented using zeromq's pub/sub sockets. By default we don't use zeromq's filtering, which means that all publish jobs are sent to all minions and filtered minion side. Zeromq does have publisher side filtering which can be enabled in salt using `zmq_filtering`.

Req Channel

The req channel is implemented using zeromq's req/rep sockets. These sockets enforce a send/recv pattern, which forces salt to serialize messages through these socket pairs. This means that although the interface is asynchronous on the minion we cannot send a second message until we have received the reply of the first message.

TCP Transport

The tcp transport is an implementation of Salt's channels using raw tcp sockets. Since this isn't using a pre-defined messaging library we will describe the wire protocol, message semantics, etc. in this document.

The tcp transport is enabled by changing the `transport` setting to `tcp` on each Salt minion and Salt master.

```
transport: tcp
```

> **Warning:** We currently recommend that when using Syndics that all Masters and Minions use the same transport. We're investigating a report of an error when using mixed transport types at very heavy loads.

Wire Protocol

This implementation over TCP focuses on flexibility over absolute efficiency. This means we are okay to spend a couple of bytes of wire space for flexibility in the future. That being said, the wire framing is quite efficient and looks like:

```
msgpack({'head': SOMEHEADER, 'body': SOMEBODY})
```

Since msgpack is an iterably parsed serialization, we can simply write the serialized payload to the wire. Within that payload we have two items "head" and "body". Head contains header information (such as "message id"). The Body contains the actual message that we are sending. With this flexible wire protocol we can implement any message semantics that we'd like– including multiplexed message passing on a single socket.

TLS Support

New in version 2016.11.1.

The TCP transport allows for the master/minion communication to be optionally wrapped in a TLS connection. Enabling this is simple, the master and minion need to be using the tcp connection, then the *ssl* option is enabled. The *ssl* option is passed as a dict and corresponds to the options passed to the Python *ssl.wrap_socket <https://docs.python.org/2/library/ssl.html#ssl.wrap_socket>* function.

A simple setup looks like this, on the Salt Master add the *ssl* option to the master configuration file:

```
ssl:
  keyfile: <path_to_keyfile>
  certfile: <path_to_certfile>
  ssl_version: PROTOCOL_TLSv1_2
```

The minimal *ssl* option in the minion configuration file looks like this:

```
ssl: True
# Versions below 2016.11.4:
ssl: {}
```

Specific options can be sent to the minion also, as defined in the Python *ssl.wrap_socket* function.

Note: While setting the ssl_version is not required, we recommend it. Some older versions of python do not support the latest TLS protocol and if this is the case for your version of python we strongly recommend upgrading your version of Python.

Crypto

The current implementation uses the same crypto as the `zeromq` transport.

Pub Channel

For the pub channel we send messages without "message ids" which the remote end interprets as a one-way send.

Note: As of today we send all publishes to all minions and rely on minion-side filtering.

Req Channel

For the req channel we send messages with a "message id". This "message id" allows us to multiplex messages across the socket.

3.23 Master Tops System

In 0.10.4 the *external_nodes* system was upgraded to allow for modular subsystems to be used to generate the top file data for a *highstate* run on the master.

The old *external_nodes* option has been removed. The master tops system provides a pluggable and extendable replacement for it, allowing for multiple different subsystems to provide top file data.

Using the new *master_tops* option is simple:

```
master_tops:
  ext_nodes: cobbler-external-nodes
```

for `Cobbler` or:

```
master_tops:
  reclass:
    inventory_base_uri: /etc/reclass
    classes_uri: roles
```

for `Reclass`.

```
master_tops:
  varstack: /path/to/the/config/file/varstack.yaml
```

for `Varstack`.

It's also possible to create custom master_tops modules. Simply place them into `salt://_tops` in the Salt fileserver and use the *saltutil.sync_tops* runner to sync them. If this runner function is not available, they can manually be placed into `extmods/tops`, relative to the master cachedir (in most cases the full path will be `/var/cache/salt/master/extmods/tops`).

Custom tops modules are written like any other execution module, see the source for the two modules above for examples of fully functional ones. Below is a bare-bones example:

/etc/salt/master:

```yaml
master_tops:
  customtop: True
```

customtop.py: (custom master_tops module)

```python
import logging
import sys
# Define the module's virtual name
__virtualname__ = 'customtop'

log = logging.getLogger(__name__)


def __virtual__():
    return __virtualname__


def top(**kwargs):
    log.debug('Calling top in customtop')
    return {'base': ['test']}
```

salt minion state.show_top should then display something like:

```
$ salt minion state.show_top

minion
    ----------
    base:
      - test
```

Note: If a master_tops module returns *top file* data for a given minion, it will be added to the states configured in the top file. It will *not* replace it altogether. The 2018.3.0 release adds additional functionality allowing a minion to treat master_tops as the single source of truth, irrespective of the top file.

3.24 Returners

By default the return values of the commands sent to the Salt minions are returned to the Salt master, however anything at all can be done with the results data.

By using a Salt returner, results data can be redirected to external data-stores for analysis and archival.

Returners pull their configuration values from the Salt minions. Returners are only configured once, which is generally at load time.

The returner interface allows the return data to be sent to any system that can receive data. This means that return data can be sent to a Redis server, a MongoDB server, a MySQL server, or any system.

See also:

Full list of builtin returners

3.24.1 Using Returners

All Salt commands will return the command data back to the master. Specifying returners will ensure that the data is _also_ sent to the specified returner interfaces.

Specifying what returners to use is done when the command is invoked:

```
salt '*' test.version --return redis_return
```

This command will ensure that the redis_return returner is used.

It is also possible to specify multiple returners:

```
salt '*' test.version --return mongo_return,redis_return,cassandra_return
```

In this scenario all three returners will be called and the data from the test.version command will be sent out to the three named returners.

3.24.2 Writing a Returner

Returners are Salt modules that allow the redirection of results data to targets other than the Salt Master.

Returners Are Easy To Write!

Writing a Salt returner is straightforward.

A returner is a Python module containing at minimum a `returner` function. Other optional functions can be included to add support for *`master_job_cache`*, *Storing Job Results in an External System*, and *Event Returners*.

`returner` The `returner` function must accept a single argument. The argument contains return data from the called minion function. If the minion function `test.version` is called, the value of the argument will be a dictionary. Run the following command from a Salt master to get a sample of the dictionary:

```
salt-call --local --metadata test.version --out=pprint
```

```
import redis
import salt.utils.json

def returner(ret):
    '''
    Return information to a redis server
    '''
    # Get a redis connection
    serv = redis.Redis(
        host='redis-serv.example.com',
        port=6379,
        db='0')
    serv.sadd("%(id)s:jobs" % ret, ret['jid'])
```

(continues on next page)

(continued from previous page)

```
    serv.set("%(jid)s:%(id)s" % ret, salt.utils.json.dumps(ret['return']))
    serv.sadd('jobs', ret['jid'])
    serv.sadd(ret['jid'], ret['id'])
```

The above example of a returner set to send the data to a Redis server serializes the data as JSON and sets it in redis.

Using Custom Returner Modules

Place custom returners in a `_returners/` directory within the *`file_roots`* specified by the master config file.

Custom returners are distributed when any of the following are called:

- *`state.apply`*
- *`saltutil.sync_returners`*
- *`saltutil.sync_all`*

Any custom returners which have been synced to a minion that are named the same as one of Salt's default set of returners will take the place of the default returner with the same name.

Naming the Returner

Note that a returner's default name is its filename (i.e. `foo.py` becomes returner `foo`), but that its name can be overridden by using a *__virtual__ function*. A good example of this can be found in the redis returner, which is named `redis_return.py` but is loaded as simply `redis`:

```
try:
    import redis
    HAS_REDIS = True
except ImportError:
    HAS_REDIS = False

__virtualname__ = 'redis'

def __virtual__():
    if not HAS_REDIS:
        return False
    return __virtualname__
```

Master Job Cache Support

`master_job_cache`, *Storing Job Results in an External System*, and *Event Returners*. Salt's *`master_job_cache`* allows returners to be used as a pluggable replacement for the *Default Job Cache*. In order to do so, a returner must implement the following functions:

Note: The code samples contained in this section were taken from the cassandra_cql returner.

`prep_jid` Ensures that job ids (jid) don't collide, unless passed_jid is provided.

`nocache` is an optional boolean that indicates if return data should be cached. `passed_jid` is a caller provided jid which should be returned unconditionally.

```
def prep_jid(nocache, passed_jid=None):  # pylint: disable=unused-argument
    '''
    Do any work necessary to prepare a JID, including sending a custom id
    '''
    return passed_jid if passed_jid is not None else salt.utils.jid.gen_jid()
```

`save_load` Save job information. The `jid` is generated by `prep_jid` and should be considered a unique identifier for the job. The jid, for example, could be used as the primary/unique key in a database. The `load` is what is returned to a Salt master by a minion. `minions` is a list of minions that the job was run against. The following code example stores the load as a JSON string in the salt.jids table.

```
import salt.utils.json

def save_load(jid, load, minions=None):
    '''
    Save the load to the specified jid id
    '''
    query = '''INSERT INTO salt.jids (
                 jid, load
               ) VALUES (
                 '{0}', '{1}'
               );'''.format(jid, salt.utils.json.dumps(load))

    # cassandra_cql.cql_query may raise a CommandExecutionError
    try:
        __salt__['cassandra_cql.cql_query'](query)
    except CommandExecutionError:
        log.critical('Could not save load in jids table.')
        raise
    except Exception as e:
        log.critical(
            'Unexpected error while inserting into jids: {0}'.format(e)
        )
        raise
```

`get_load` must accept a job id (jid) and return the job load stored by `save_load`, or an empty dictionary when not found.

```
def get_load(jid):
    '''
    Return the load data that marks a specified jid
    '''
    query = '''SELECT load FROM salt.jids WHERE jid = '{0}';'''.format(jid)

    ret = {}

    # cassandra_cql.cql_query may raise a CommandExecutionError
    try:
        data = __salt__['cassandra_cql.cql_query'](query)
        if data:
            load = data[0].get('load')
            if load:
                ret = json.loads(load)
    except CommandExecutionError:
        log.critical('Could not get load from jids table.')
        raise
    except Exception as e:
```

(continues on next page)

(continued from previous page)

```
        log.critical('''Unexpected error while getting load from
         jids: {0}'''.format(str(e)))
        raise

    return ret
```

External Job Cache Support

Salt's *Storing Job Results in an External System* extends the `master_job_cache`. External Job Cache support requires the following functions in addition to what is required for Master Job Cache support:

`get_jid` Return a dictionary containing the information (load) returned by each minion when the specified job id was executed.

Sample:

```
{
    "local": {
        "master_minion": {
            "fun_args": [],
            "jid": "20150330121011408195",
            "return": "2018.3.4",
            "retcode": 0,
            "success": true,
            "cmd": "_return",
            "_stamp": "2015-03-30T12:10:12.708663",
            "fun": "test.version",
            "id": "master_minion"
        }
    }
}
```

`get_fun` Return a dictionary of minions that called a given Salt function as their last function call.

Sample:

```
{
    "local": {
        "minion1": "test.version",
        "minion3": "test.version",
        "minion2": "test.version"
    }
}
```

`get_jids` Return a list of all job ids.

Sample:

```
{
    "local": [
        "20150330121011408195",
        "20150330195922139916"
    ]
}
```

`get_minions` Returns a list of minions

Sample:

```
{
    "local": [
        "minion3",
        "minion2",
        "minion1",
        "master_minion"
    ]
}
```

Please refer to one or more of the existing returners (i.e. mysql, cassandra_cql) if you need further clarification.

Event Support

An `event_return` function must be added to the returner module to allow events to be logged from a master via the returner. A list of events are passed to the function by the master.

The following example was taken from the MySQL returner. In this example, each event is inserted into the salt_events table keyed on the event tag. The tag contains the jid and therefore is guaranteed to be unique.

```
import salt.utils.json

def event_return(events):
 '''
 Return event to mysql server

 Requires that configuration be enabled via 'event_return'
 option in master config.
 '''
 with _get_serv(events, commit=True) as cur:
     for event in events:
         tag = event.get('tag', '')
         data = event.get('data', '')
         sql = '''INSERT INTO `salt_events` (`tag`, `data`, `master_id` )
                  VALUES (%s, %s, %s)'''
         cur.execute(sql, (tag, salt.utils.json.dumps(data), __opts__['id']))
```

Testing the Returner

The `returner`, `prep_jid`, `save_load`, `get_load`, and `event_return` functions can be tested by configuring the *`master_job_cache`* and *Event Returners* in the master config file and submitting a job to `test.version` each minion from the master.

Once you have successfully exercised the Master Job Cache functions, test the External Job Cache functions using the `ret` execution module.

```
salt-call ret.get_jids cassandra_cql --output=json
salt-call ret.get_fun cassandra_cql test.version --output=json
salt-call ret.get_minions cassandra_cql --output=json
salt-call ret.get_jid cassandra_cql 20150330121011408195 --output=json
```

3.24.3 Event Returners

For maximum visibility into the history of events across a Salt infrastructure, all events seen by a salt master may be logged to one or more returners.

To enable event logging, set the `event_return` configuration option in the master config to the returner(s) which should be designated as the handler for event returns.

Note: Not all returners support event returns. Verify a returner has an `event_return()` function before using.

Note: On larger installations, many hundreds of events may be generated on a busy master every second. Be certain to closely monitor the storage of a given returner as Salt can easily overwhelm an underpowered server with thousands of returns.

3.24.4 Full List of Returners

returner modules

carbon_return	Take data from salt and "return" it into a carbon receiver
cassandra_cql_return	Return data to a cassandra server
cassandra_return	Return data to a Cassandra ColumnFamily
couchbase_return	Simple returner for Couchbase.
couchdb_return	Simple returner for CouchDB.
django_return	A returner that will inform a Django system that returns are available using Django's signal system.
elasticsearch_return	Return data to an elasticsearch server for indexing.
etcd_return	Return data to an etcd server or cluster
highstate_return	Return the results of a highstate (or any other state function that returns data in a compatible format) via an HTML email or HTML file.
influxdb_return	Return data to an influxdb server.
kafka_return	Return data to a Kafka topic
librato_return	Salt returner to return highstate stats to Librato
local	The local returner is used to test the returner interface, it just prints the return data to the console to verify that it is being passed properly
local_cache	Return data to local job cache
mattermost_returner	Return salt data via mattermost
memcache_return	Return data to a memcache server
mongo_future_return	Return data to a mongodb server
mongo_return	Return data to a mongodb server
multi_returner	Read/Write multiple returners
mysql	Return data to a mysql server
nagios_nrdp_return	Return salt data to Nagios
odbc	Return data to an ODBC compliant server.
pgjsonb	Return data to a PostgreSQL server with json data stored in Pg's jsonb data type

Continued on next page

Table 2 – continued from previous page

postgres	Return data to a postgresql server
postgres_local_cache	Use a postgresql server for the master job cache.
pushover_returner	Return salt data via pushover (http://www.pushover.net)
rawfile_json	Take data from salt and "return" it into a raw file containing the json, with one line per event.
redis_return	Return data to a redis server
sentry_return	Salt returner that reports execution results back to sentry.
slack_returner	Return salt data via slack
sms_return	Return data by SMS.
smtp_return	Return salt data via email
splunk	Send json response data to Splunk via the HTTP Event Collector Requires the following config values to be specified in config or pillar:
sqlite3_return	Insert minion return data into a sqlite3 database
syslog_return	Return data to the host operating system's syslog facility
telegram_return	Return salt data via Telegram.
xmpp_return	Return salt data via xmpp
zabbix_return	Return salt data to Zabbix

salt.returners.carbon_return

Take data from salt and "return" it into a carbon receiver

Add the following configuration to the minion configuration file:

```
carbon.host: <server ip address>
carbon.port: 2003
```

Errors when trying to convert data to numbers may be ignored by setting `carbon.skip_on_error` to *True*:

```
carbon.skip_on_error: True
```

By default, data will be sent to carbon using the plaintext protocol. To use the pickle protocol, set `carbon.mode` to `pickle`:

```
carbon.mode: pickle
```

You can also specify the pattern used for the metric base path (except for virt modules metrics):
carbon.metric_base_pattern: carbon.[minion_id].[module].[function]

These tokens can used : [module]: salt module [function]: salt function [minion_id]: minion id

Default is : carbon.metric_base_pattern: [module].[function].[minion_id]

Carbon settings may also be configured as:

```
carbon:
  host: <server IP or hostname>
  port: <carbon port>
  skip_on_error: True
  mode: (pickle|text)
  metric_base_pattern: <pattern> | [module].[function].[minion_id]
```

Alternative configuration values can be used by prefacing the configuration. Any values not found in the alternative configuration will be pulled from the default location:

```
alternative.carbon:
  host: <server IP or hostname>
  port: <carbon port>
  skip_on_error: True
  mode: (pickle|text)
```

To use the carbon returner, append '–return carbon' to the salt command.

```
salt '*' test.ping --return carbon
```

To use the alternative configuration, append '–return_config alternative' to the salt command.

New in version 2015.5.0.

```
salt '*' test.ping --return carbon --return_config alternative
```

To override individual configuration items, append –return_kwargs '{"key:": "value"}' to the salt command.

New in version 2016.3.0.

```
salt '*' test.ping --return carbon --return_kwargs '{"skip_on_error": False}'
```

`salt.returners.carbon_return.`**`event_return`**`(`*events*`)`

Return event data to remote carbon server

Provide a list of events to be stored in carbon

`salt.returners.carbon_return.`**`prep_jid`**`(`*nocache=False, passed_jid=None*`)`

Do any work necessary to prepare a JID, including sending a custom id

`salt.returners.carbon_return.`**`returner`**`(`*ret*`)`

Return data to a remote carbon server using the text metric protocol

Each metric will look like:

```
[module].[function].[minion_id].[metric path [...]].[metric name]
```

salt.returners.cassandra_cql_return

Return data to a cassandra server

New in version 2015.5.0.

maintainer Corin Kochenower<ckochenower@saltstack.com>

maturity new as of 2015.2

depends salt.modules.cassandra_cql

depends DataStax Python Driver for Apache Cassandra https://github.com/datastax/python-driver pip install cassandra-driver

platform all

configuration To enable this returner, the minion will need the DataStax Python Driver for Apache Cassandra (https://github.com/datastax/python-driver) installed and the following values configured in the minion or master config. The list of cluster IPs must include at least one cassandra node IP address. No assumption or default will be used for the cluster IPs. The cluster IPs will be tried in the order listed. The port, username, and password values shown below will be the assumed defaults if you do not provide values.:

```
cassandra:
  cluster:
    - 192.168.50.11
    - 192.168.50.12
    - 192.168.50.13
  port: 9042
  username: salt
  password: salt
```

Use the following cassandra database schema:

```
CREATE KEYSPACE IF NOT EXISTS salt
    WITH replication = {'class': 'SimpleStrategy', 'replication_factor'
↪: 1};

CREATE USER IF NOT EXISTS salt WITH PASSWORD 'salt' NOSUPERUSER;

GRANT ALL ON KEYSPACE salt TO salt;

USE salt;

CREATE TABLE IF NOT EXISTS salt.salt_returns (
    jid text,
    minion_id text,
    fun text,
    alter_time timestamp,
    full_ret text,
    return text,
    success boolean,
    PRIMARY KEY (jid, minion_id, fun)
) WITH CLUSTERING ORDER BY (minion_id ASC, fun ASC);
CREATE INDEX IF NOT EXISTS salt_returns_minion_id ON salt.salt_returns
↪(minion_id);
CREATE INDEX IF NOT EXISTS salt_returns_fun ON salt.salt_returns (fun);

CREATE TABLE IF NOT EXISTS salt.jids (
    jid text PRIMARY KEY,
    load text
);

CREATE TABLE IF NOT EXISTS salt.minions (
    minion_id text PRIMARY KEY,
    last_fun text
);
CREATE INDEX IF NOT EXISTS minions_last_fun ON salt.minions (last_fun);

CREATE TABLE IF NOT EXISTS salt.salt_events (
    id timeuuid,
    tag text,
    alter_time timestamp,
    data text,
    master_id text,
    PRIMARY KEY (id, tag)
) WITH CLUSTERING ORDER BY (tag ASC);
CREATE INDEX tag ON salt.salt_events (tag);
```

Required python modules: cassandra-driver

To use the cassandra returner, append '–return cassandra_cql' to the salt command. ex:

```
salt '*' test.ping --return_cql cassandra
```

Note: if your Cassandra instance has not been tuned much you may benefit from altering some timeouts in *cassandra.yaml* like so:

```
# How long the coordinator should wait for read operations to complete
read_request_timeout_in_ms: 5000
# How long the coordinator should wait for seq or index scans to complete
range_request_timeout_in_ms: 20000
# How long the coordinator should wait for writes to complete
write_request_timeout_in_ms: 20000
# How long the coordinator should wait for counter writes to complete
counter_write_request_timeout_in_ms: 10000
# How long a coordinator should continue to retry a CAS operation
# that contends with other proposals for the same row
cas_contention_timeout_in_ms: 5000
# How long the coordinator should wait for truncates to complete
# (This can be much longer, because unless auto_snapshot is disabled
# we need to flush first so we can snapshot before removing the data.)
truncate_request_timeout_in_ms: 60000
# The default timeout for other, miscellaneous operations
request_timeout_in_ms: 20000
```

As always, your mileage may vary and your Cassandra cluster may have different needs. SaltStack has seen situations where these timeouts can resolve some stacktraces that appear to come from the Datastax Python driver.

`salt.returners.cassandra_cql_return.`**`event_return`**(*events*)

Return event to one of potentially many clustered cassandra nodes

Requires that configuration be enabled via 'event_return' option in master config.

Cassandra does not support an auto-increment feature due to the highly inefficient nature of creating a monotonically increasing number across all nodes in a distributed database. Each event will be assigned a uuid by the connecting client.

`salt.returners.cassandra_cql_return.`**`get_fun`**(*fun*)

Return a dict of the last function called for all minions

`salt.returners.cassandra_cql_return.`**`get_jid`**(*jid*)

Return the information returned when the specified job id was executed

`salt.returners.cassandra_cql_return.`**`get_jids`**()

Return a list of all job ids

`salt.returners.cassandra_cql_return.`**`get_load`**(*jid*)

Return the load data that marks a specified jid

`salt.returners.cassandra_cql_return.`**`get_minions`**()

Return a list of minions

`salt.returners.cassandra_cql_return.`**`prep_jid`**(*nocache, passed_jid=None*)

Do any work necessary to prepare a JID, including sending a custom id

`salt.returners.cassandra_cql_return.`**`returner`**(*ret*)

Return data to one of potentially many clustered cassandra nodes

`salt.returners.cassandra_cql_return.`**`save_load`**(*jid, load, minions=None*)

Save the load to the specified jid id

salt.returners.cassandra_return

Return data to a Cassandra ColumnFamily

Here's an example Keyspace / ColumnFamily setup that works with this returner:

```
create keyspace salt;
use salt;
create column family returns
  with key_validation_class='UTF8Type'
  and comparator='UTF8Type'
  and default_validation_class='UTF8Type';
```

Required python modules: pycassa

> To use the cassandra returner, append '–return cassandra' to the salt command. ex:
>
> > salt '*' test.ping –return cassandra

`salt.returners.cassandra_return.`**`prep_jid`**`(`*nocache=False, passed_jid=None*`)`
Do any work necessary to prepare a JID, including sending a custom id

`salt.returners.cassandra_return.`**`returner`**`(`*ret*`)`
Return data to a Cassandra ColumnFamily

salt.returners.couchbase_return

Simple returner for Couchbase. Optional configuration settings are listed below, along with sane defaults.

```
couchbase.host:   'salt'
couchbase.port:   8091
couchbase.bucket: 'salt'
couchbase.ttl: 24
couchbase.password: 'password'
couchbase.skip_verify_views: False
```

To use the couchbase returner, append '–return couchbase' to the salt command. ex:

```
salt '*' test.ping --return couchbase
```

To use the alternative configuration, append '–return_config alternative' to the salt command.

New in version 2015.5.0.

```
salt '*' test.ping --return couchbase --return_config alternative
```

To override individual configuration items, append –return_kwargs '{"key:": "value"}' to the salt command.

New in version 2016.3.0.

```
salt '*' test.ping --return couchbase --return_kwargs '{"bucket": "another-salt"}'
```

All of the return data will be stored in documents as follows:

JID

load: load obj tgt_minions: list of minions targeted nocache: should we not cache the return data

JID/MINION_ID

return: return_data full_ret: full load of job return

`salt.returners.couchbase_return.`**`get_jid`**`(`*jid*`)`
Return the information returned when the specified job id was executed

`salt.returners.couchbase_return.`**`get_jids`**`()`
Return a list of all job ids

`salt.returners.couchbase_return.`**`get_load`**`(`*jid*`)`
Return the load data that marks a specified jid

`salt.returners.couchbase_return.`**`prep_jid`**`(`*nocache=False, passed_jid=None*`)`
Return a job id and prepare the job id directory This is the function responsible for making sure jids don't collide (unless its passed a jid) So do what you have to do to make sure that stays the case

`salt.returners.couchbase_return.`**`returner`**`(`*load*`)`
Return data to couchbase bucket

`salt.returners.couchbase_return.`**`save_load`**`(`*jid, clear_load, minion=None*`)`
Save the load to the specified jid

`salt.returners.couchbase_return.`**`save_minions`**`(`*jid, minions, syndic_id=None*`)`
Save/update the minion list for a given jid. The syndic_id argument is included for API compatibility only.

salt.returners.couchdb_return

Simple returner for CouchDB. Optional configuration settings are listed below, along with sane defaults:

```
couchdb.db: 'salt'
couchdb.url: 'http://salt:5984/'
```

Alternative configuration values can be used by prefacing the configuration. Any values not found in the alternative configuration will be pulled from the default location:

```
alternative.couchdb.db: 'salt'
alternative.couchdb.url: 'http://salt:5984/'
```

To use the couchdb returner, append `--return couchdb` to the salt command. Example:

```
salt '*' test.ping --return couchdb
```

To use the alternative configuration, append `--return_config alternative` to the salt command.

New in version 2015.5.0.

```
salt '*' test.ping --return couchdb --return_config alternative
```

To override individual configuration items, append –return_kwargs '{"key:": "value"}' to the salt command.

New in version 2016.3.0.

```
salt '*' test.ping --return couchdb --return_kwargs '{"db": "another-salt"}'
```

On concurrent database access

As this returner creates a couchdb document with the salt job id as document id and as only one document with a given id can exist in a given couchdb database, it is advised for most setups that every minion be configured to write to it own database (the value of `couchdb.db` may be suffixed with the minion id), otherwise multi-minion targeting can lead to losing output:

- the first returning minion is able to create a document in the database
- other minions fail with `{'error': 'HTTP Error 409: Conflict'}`

`salt.returners.couchdb_return.`**`ensure_views`**`()`
This function makes sure that all the views that should exist in the design document do exist.

`salt.returners.couchdb_return.`**`get_fun`**`(`*fun*`)`
Return a dict with key being minion and value being the job details of the last run of function 'fun'.

`salt.returners.couchdb_return.`**`get_jid`**`(`*jid*`)`
Get the document with a given JID.

`salt.returners.couchdb_return.`**`get_jids`**`()`
List all the jobs that we have..

`salt.returners.couchdb_return.`**`get_minions`**`()`
Return a list of minion identifiers from a request of the view.

`salt.returners.couchdb_return.`**`get_valid_salt_views`**`()`
Returns a dict object of views that should be part of the salt design document.

`salt.returners.couchdb_return.`**`prep_jid`**`(`*nocache=False, passed_jid=None*`)`
Do any work necessary to prepare a JID, including sending a custom id

`salt.returners.couchdb_return.`**`returner`**`(`*ret*`)`
Take in the return and shove it into the couchdb database.

`salt.returners.couchdb_return.`**`set_salt_view`**`()`
Helper function that sets the salt design document. Uses get_valid_salt_views and some hardcoded values.

salt.returners.django_return

A returner that will inform a Django system that returns are available using Django's signal system.

https://docs.djangoproject.com/en/dev/topics/signals/

It is up to the Django developer to register necessary handlers with the signals provided by this returner and process returns as necessary.

The easiest way to use signals is to import them from this returner directly and then use a decorator to register them.

An example Django module that registers a function called 'returner_callback' with this module's 'returner' function:

```
import salt.returners.django_return
from django.dispatch import receiver

@receiver(salt.returners.django_return, sender=returner)
def returner_callback(sender, ret):
    print('I received {0} from {1}'.format(ret, sender))
```

`salt.returners.django_return.`**`prep_jid`**`(`*nocache=False, passed_jid=None*`)`
Do any work necessary to prepare a JID, including sending a custom ID

`salt.returners.django_return.`**`returner`**(*ret*)
Signal a Django server that a return is available

`salt.returners.django_return.`**`save_load`**(*jid*, *load*, *minions=None*)
Save the load to the specified jid

salt.returners.elasticsearch_return

Return data to an elasticsearch server for indexing.

maintainer Jurnell Cockhren <jurnell.cockhren@sophicware.com>, Arnold Bechtoldt <mail@arnoldbechtoldt.com>

maturity New

depends elasticsearch-py

platform all

To enable this returner the elasticsearch python client must be installed on the desired minions (all or some subset).

Please see documentation of *`elasticsearch execution module`* for a valid connection configuration.

Warning: The index that you wish to store documents will be created by Elasticsearch automatically if doesn't exist yet. It is highly recommended to create predefined index templates with appropriate mapping(s) that will be used by Elasticsearch upon index creation. Otherwise you will have problems as described in #20826.

To use the returner per salt call:

```
salt '*' test.ping --return elasticsearch
```

In order to have the returner apply to all minions:

```
ext_job_cache: elasticsearch
```

Minion configuration:

debug_returner_payload': False Output the payload being posted to the log file in debug mode

doc_type: 'default' Document type to use for normal return messages

functions_blacklist Optional list of functions that should not be returned to elasticsearch

index_date: False Use a dated index (e.g. <index>-2016.11.29)

master_event_index: 'salt-master-event-cache' Index to use when returning master events

master_event_doc_type: 'efault' Document type to use got master events

master_job_cache_index: 'salt-master-job-cache' Index to use for master job cache

master_job_cache_doc_type: 'default' Document type to use for master job cache

number_of_shards: 1 Number of shards to use for the indexes

number_of_replicas: 0 Number of replicas to use for the indexes

NOTE: The following options are valid for 'state.apply', 'state.sls' and 'state.highstate' functions only.

states_count: False Count the number of states which succeeded or failed and return it in top-level item called 'counts'. States reporting None (i.e. changes would be made but it ran in test mode) are counted as successes.

states_order_output: False Prefix the state UID (e.g. file_|-yum_configured_|-/etc/yum.conf_|-managed) with a zero-padded version of the '__run_num__' value to allow for easier sorting. Also store the state function (i.e. file.managed) into a new key '_func'. Change the index to be '<index>-ordered' (e.g. salt-state_apply-ordered).

states_single_index: False Store results for state.apply, state.sls and state.highstate in the salt-state_apply index (or -ordered/-<date>) indexes if enabled

```
elasticsearch:
    hosts:
      - "10.10.10.10:9200"
      - "10.10.10.11:9200"
      - "10.10.10.12:9200"
    index_date: True
    number_of_shards: 5
    number_of_replicas: 1
    debug_returner_payload: True
    states_count: True
    states_order_output: True
    states_single_index: True
    functions_blacklist:
      - test.ping
      - saltutil.find_job
```

`salt.returners.elasticsearch_return.`**`event_return`**(*events*)

Return events to Elasticsearch

Requires that the *event_return* configuration be set in master config.

`salt.returners.elasticsearch_return.`**`get_load`**(*jid*)

Return the load data that marks a specified jid

New in version 2015.8.1.

`salt.returners.elasticsearch_return.`**`prep_jid`**(*nocache=False, passed_jid=None*)

Do any work necessary to prepare a JID, including sending a custom id

`salt.returners.elasticsearch_return.`**`returner`**(*ret*)

Process the return from Salt

`salt.returners.elasticsearch_return.`**`save_load`**(*jid, load, minions=None*)

Save the load to the specified jid id

New in version 2015.8.1.

salt.returners.etcd_return

Return data to an etcd server or cluster

depends

- python-etcd

In order to return to an etcd server, a profile should be created in the master configuration file:

```
my_etcd_config:
  etcd.host: 127.0.0.1
  etcd.port: 2379
```

It is technically possible to configure etcd without using a profile, but this is not considered to be a best practice, especially when multiple etcd servers or clusters are available.

```
etcd.host: 127.0.0.1
etcd.port: 2379
```

Additionally, two more options must be specified in the top-level configuration in order to use the etcd returner:

```
etcd.returner: my_etcd_config
etcd.returner_root: /salt/return
```

The `etcd.returner` option specifies which configuration profile to use. The `etcd.returner_root` option specifies the path inside etcd to use as the root of the returner system.

Once the etcd options are configured, the returner may be used:

CLI Example:

> salt '*' test.ping –return etcd

A username and password can be set:

```
etcd.username: larry  # Optional; requires etcd.password to be set
etcd.password: 123pass  # Optional; requires etcd.username to be set
```

You can also set a TTL (time to live) value for the returner:

```
etcd.ttl: 5
```

Authentication with username and password, and ttl, currently requires the `master` branch of `python-etcd`.

You may also specify different roles for read and write operations. First, create the profiles as specified above. Then add:

```
etcd.returner_read_profile: my_etcd_read
etcd.returner_write_profile: my_etcd_write
```

`salt.returners.etcd_return.`**`clean_old_jobs`**`()`
: Included for API consistency

`salt.returners.etcd_return.`**`get_fun`**`(`*fun*`)`
: Return a dict of the last function called for all minions

`salt.returners.etcd_return.`**`get_jid`**`(`*jid*`)`
: Return the information returned when the specified job id was executed

`salt.returners.etcd_return.`**`get_jids`**`()`
: Return a list of all job ids

`salt.returners.etcd_return.`**`get_load`**`(`*jid*`)`
: Return the load data that marks a specified jid

`salt.returners.etcd_return.`**`get_minions`**`()`
: Return a list of minions

`salt.returners.etcd_return.`**`prep_jid`**`(`*nocache=False, passed_jid=None*`)`
: Do any work necessary to prepare a JID, including sending a custom id

`salt.returners.etcd_return.`**`returner`**`(`*ret*`)`
: Return data to an etcd server or cluster

`salt.returners.etcd_return.`**`save_load`**(*jid*, *load*, *minions=None*)
Save the load to the specified jid

salt.returners.highstate_return module

Return the results of a highstate (or any other state function that returns data in a compatible format) via an HTML email or HTML file.

New in version 2017.7.0.

Similar results can be achieved by using the smtp returner with a custom template, except an attempt at writing such a template for the complex data structure returned by highstate function had proven to be a challenge, not to mention that the smtp module doesn't support sending HTML mail at the moment.

The main goal of this returner was to produce an easy to read email similar to the output of highstate outputter used by the CLI.

This returner could be very useful during scheduled executions, but could also be useful for communicating the results of a manual execution.

Returner configuration is controlled in a standard fashion either via highstate group or an alternatively named group.

```
salt '*' state.highstate --return highstate
```

To use the alternative configuration, append '–return_config config-name'

```
salt '*' state.highstate --return highstate --return_config simple
```

Here is an example of what the configuration might look like:

```
simple.highstate:
  report_failures: True
  report_changes: True
  report_everything: False
  failure_function: pillar.items
  success_function: pillar.items
  report_format: html
  report_delivery: smtp
  smtp_success_subject: 'success minion {id} on host {host}'
  smtp_failure_subject: 'failure minion {id} on host {host}'
  smtp_server: smtp.example.com
  smtp_recipients: saltusers@example.com, devops@example.com
  smtp_sender: salt@example.com
```

The *report_failures*, *report_changes*, and *report_everything* flags provide filtering of the results. If you want an email to be sent every time, then *report_everything* is your choice. If you want to be notified only when changes were successfully made use *report_changes*. And *report_failures* will generate an email if there were failures.

The configuration allows you to run a salt module function in case of success (*success_function*) or failure (*failure_function*).

Any salt function, including ones defined in the _module folder of your salt repo, could be used here and its output will be displayed under the 'extra' heading of the email.

Supported values for *report_format* are html, json, and yaml. The latter two are typically used for debugging purposes, but could be used for applying a template at some later stage.

The values for *report_delivery* are smtp or file. In case of file delivery the only other applicable option is *file_output*.

In case of smtp delivery, smtp_* options demonstrated by the example above could be used to customize the email.

As you might have noticed, the success and failure subjects contain {id} and {host} values. Any other grain name could be used. As opposed to using {{grains['id']}}, which will be rendered by the master and contain master's values at the time of pillar generation, these will contain minion values at the time of execution.

salt.returners.highstate_return.**returner**(*ret*)
Check highstate return information and possibly fire off an email or save a file.

salt.returners.influxdb_return

Return data to an influxdb server.

New in version 2015.8.0.

To enable this returner the minion will need the python client for influxdb installed and the following values configured in the minion or master config, these are the defaults:

```
influxdb.db: 'salt'
influxdb.user: 'salt'
influxdb.password: 'salt'
influxdb.host: 'localhost'
influxdb.port: 8086
```

Alternative configuration values can be used by prefacing the configuration. Any values not found in the alternative configuration will be pulled from the default location:

```
alternative.influxdb.db: 'salt'
alternative.influxdb.user: 'salt'
alternative.influxdb.password: 'salt'
alternative.influxdb.host: 'localhost'
alternative.influxdb.port: 6379
```

To use the influxdb returner, append '–return influxdb' to the salt command.

```
salt '*' test.ping --return influxdb
```

To use the alternative configuration, append '–return_config alternative' to the salt command.

```
salt '*' test.ping --return influxdb --return_config alternative
```

To override individual configuration items, append –return_kwargs '{"key:": "value"}' to the salt command.

New in version 2016.3.0.

```
salt '*' test.ping --return influxdb --return_kwargs '{"db": "another-salt"}'
```

salt.returners.influxdb_return.**get_fun**(*fun*)
Return a dict of the last function called for all minions

salt.returners.influxdb_return.**get_jid**(*jid*)
Return the information returned when the specified job id was executed

salt.returners.influxdb_return.**get_jids**()
Return a list of all job ids

salt.returners.influxdb_return.**get_load**(*jid*)
Return the load data that marks a specified jid

salt.returners.influxdb_return.**get_minions**()
Return a list of minions

`salt.returners.influxdb_return.`**`prep_jid`**(*nocache=False*, *passed_jid=None*)
Do any work necessary to prepare a JID, including sending a custom id

`salt.returners.influxdb_return.`**`returner`**(*ret*)
Return data to a influxdb data store

`salt.returners.influxdb_return.`**`save_load`**(*jid*, *load*, *minions=None*)
Save the load to the specified jid

salt.returners.kafka_return

Return data to a Kafka topic

maintainer Justin Desilets (justin.desilets@gmail.com)

maturity 20181119

depends confluent-kafka

platform all

To enable this returner install confluent-kafka and enable the following settings in the minion config:

returner.kafka.bootstrap:

- ”server1:9092”
- ”server2:9092”
- ”server3:9092”

returner.kafka.topic: ’topic’

To use the kafka returner, append *–return kafka* to the Salt command, eg;

salt ’*’ test.ping –return kafka

`salt.returners.kafka_return.`**`returner`**(*ret*)
Return information to a Kafka server

salt.returners.librato_return

Salt returner to return highstate stats to Librato

To enable this returner the minion will need the Librato client importable on the Python path and the following values configured in the minion or master config.

The Librato python client can be found at: https://github.com/librato/python-librato

```
librato.email: example@librato.com
librato.api_token: abc12345def
```

This return supports multi-dimension metrics for Librato. To enable support for more metrics, the tags JSON object can be modified to include other tags.

Adding EC2 Tags example: If ec2_tags:region were desired within the tags for multi-dimension. The tags could be modified to include the ec2 tags. Multiple dimensions are added simply by adding more tags to the submission.

```
pillar_data = __salt__['pillar.raw']()
q.add(metric.name, value, tags={'Name': ret['id'],'Region': pillar_data['ec2_tags'][
↪'Name']})
```

`salt.returners.librato_return.`**`returner`**`(`*ret*`)`
Parse the return data and return metrics to Librato.

salt.returners.local

The local returner is used to test the returner interface, it just prints the return data to the console to verify that it is being passed properly

To use the local returner, append '–return local' to the salt command. ex:

```
salt '*' test.ping --return local
```

`salt.returners.local.`**`event_return`**`(`*event*`)`
Print event return data to the terminal to verify functionality

`salt.returners.local.`**`returner`**`(`*ret*`)`
Print the return data to the terminal to verify functionality

salt.returners.local_cache

Return data to local job cache

`salt.returners.local_cache.`**`clean_old_jobs`**`()`
Clean out the old jobs from the job cache

`salt.returners.local_cache.`**`get_endtime`**`(`*jid*`)`
Retrieve the stored endtime for a given job

Returns False if no endtime is present

`salt.returners.local_cache.`**`get_jid`**`(`*jid*`)`
Return the information returned when the specified job id was executed

`salt.returners.local_cache.`**`get_jids`**`()`
Return a dict mapping all job ids to job information

`salt.returners.local_cache.`**`get_jids_filter`**`(`*count, filter_find_job=True*`)`
Return a list of all jobs information filtered by the given criteria. :param int count: show not more than the count of most recent jobs :param bool filter_find_jobs: filter out 'saltutil.find_job' jobs

`salt.returners.local_cache.`**`get_load`**`(`*jid*`)`
Return the load data that marks a specified jid

`salt.returners.local_cache.`**`load_reg`**`()`
Load the register from msgpack files

`salt.returners.local_cache.`**`prep_jid`**`(`*nocache=False, passed_jid=None, recurse_count=0*`)`
Return a job id and prepare the job id directory.

This is the function responsible for making sure jids don't collide (unless it is passed a jid). So do what you have to do to make sure that stays the case

`salt.returners.local_cache.`**`returner`**`(`*load*`)`
Return data to the local job cache

`salt.returners.local_cache.`**`save_load`**`(`*jid, clear_load, minions=None, recurse_count=0*`)`
Save the load to the specified jid

minions argument is to provide a pre-computed list of matched minions for the job, for cases when this function can't compute that list itself (such as for salt-ssh)

`salt.returners.local_cache.`**`save_minions`**(*jid*, *minions*, *syndic_id=None*)
Save/update the serialized list of minions for a given job

`salt.returners.local_cache.`**`save_reg`**(*data*)
Save the register to msgpack files

`salt.returners.local_cache.`**`update_endtime`**(*jid*, *time*)
Update (or store) the end time for a given job

Endtime is stored as a plain text string

salt.returners.mattermost_returner module

Return salt data via mattermost

New in version 2017.7.0.

The following fields can be set in the minion conf file:

```
mattermost.hook (required)
mattermost.username (optional)
mattermost.channel (optional)
```

Alternative configuration values can be used by prefacing the configuration. Any values not found in the alternative configuration will be pulled from the default location:

```
mattermost.channel
mattermost.hook
mattermost.username
```

mattermost settings may also be configured as:

```
mattermost:
  channel: RoomName
  hook: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
  username: user
```

To use the mattermost returner, append '–return mattermost' to the salt command.

```
salt '*' test.ping --return mattermost
```

To override individual configuration items, append –return_kwargs '{'key:': 'value'}' to the salt command.

```
salt '*' test.ping --return mattermost --return_kwargs '{'channel': '#random'}'
```

`salt.returners.mattermost_returner.`**`event_return`**(*events*)
Send the events to a mattermost room.

Parameters **`events`** – List of events

Returns Boolean if messages were sent successfully.

`salt.returners.mattermost_returner.`**`post_message`**(*channel*, *message*, *username*, *api_url*, *hook*)
Send a message to a mattermost room.

Parameters

- **`channel`** – The room name.
- **`message`** – The message to send to the mattermost room.

- **username** – Specify who the message is from.
- **hook** – The mattermost hook, if not specified in the configuration.

Returns Boolean if message was sent successfully.

`salt.returners.mattermost_returner.`**`returner`**`(`*ret*`)`
Send an mattermost message with the data

salt.returners.memcache_return

Return data to a memcache server

To enable this returner the minion will need the python client for memcache installed and the following values configured in the minion or master config, these are the defaults.

```
memcache.host: 'localhost'
memcache.port: '11211'
```

Alternative configuration values can be used by prefacing the configuration. Any values not found in the alternative configuration will be pulled from the default location.

```
alternative.memcache.host: 'localhost'
alternative.memcache.port: '11211'
```

python2-memcache uses 'localhost' and '11211' as syntax on connection.

To use the memcache returner, append '–return memcache' to the salt command.

```
salt '*' test.ping --return memcache
```

To use the alternative configuration, append '–return_config alternative' to the salt command.

New in version 2015.5.0.

```
salt '*' test.ping --return memcache --return_config alternative
```

To override individual configuration items, append –return_kwargs '{"key:": "value"}' to the salt command.

New in version 2016.3.0.

```
salt '*' test.ping --return memcache --return_kwargs '{"host": "hostname.domain.com"}'
```

`salt.returners.memcache_return.`**`get_fun`**`(`*fun*`)`
Return a dict of the last function called for all minions

`salt.returners.memcache_return.`**`get_jid`**`(`*jid*`)`
Return the information returned when the specified job id was executed

`salt.returners.memcache_return.`**`get_jids`**`()`
Return a list of all job ids

`salt.returners.memcache_return.`**`get_load`**`(`*jid*`)`
Return the load data that marks a specified jid

`salt.returners.memcache_return.`**`get_minions`**`()`
Return a list of minions

`salt.returners.memcache_return.`**`prep_jid`**`(`*nocache=False, passed_jid=None*`)`
Do any work necessary to prepare a JID, including sending a custom id

`salt.returners.memcache_return.`**`returner`**(*ret*)
Return data to a memcache data store

`salt.returners.memcache_return.`**`save_load`**(*jid*, *load*, *minions=None*)
Save the load to the specified jid

salt.returners.mongo_future_return

Return data to a mongodb server

Required python modules: pymongo

This returner will send data from the minions to a MongoDB server. MongoDB server can be configured by using host, port, db, user and password settings or by connection string URI (for pymongo > 2.3). To configure the settings for your MongoDB server, add the following lines to the minion config files:

```
mongo.db: <database name>
mongo.host: <server ip address>
mongo.user: <MongoDB username>
mongo.password: <MongoDB user password>
mongo.port: 27017
```

Or single URI:

```
mongo.uri: URI
```

where uri is in the format:

```
mongodb://[username:password@]host1[:port1][,host2[:port2],...[,hostN[:portN]]][/
↪[database][?options]]
```

Example:

```
mongodb://db1.example.net:27017/mydatabase
mongodb://db1.example.net:27017,db2.example.net:2500/?replicaSet=test
mongodb://db1.example.net:27017,db2.example.net:2500/?replicaSet=test&
↪connectTimeoutMS=300000
```

More information on URI format can be found in https://docs.mongodb.com/manual/reference/connection-string/

You can also ask for indexes creation on the most common used fields, which should greatly improve performance. Indexes are not created by default.

```
mongo.indexes: true
```

Alternative configuration values can be used by prefacing the configuration. Any values not found in the alternative configuration will be pulled from the default location:

```
alternative.mongo.db: <database name>
alternative.mongo.host: <server ip address>
alternative.mongo.user: <MongoDB username>
alternative.mongo.password: <MongoDB user password>
alternative.mongo.port: 27017
```

Or single URI:

```
alternative.mongo.uri: URI
```

This mongo returner is being developed to replace the default mongodb returner in the future and should not be considered API stable yet.

To use the mongo returner, append '–return mongo' to the salt command.

```
salt '*' test.ping --return mongo
```

To use the alternative configuration, append '–return_config alternative' to the salt command.

New in version 2015.5.0.

```
salt '*' test.ping --return mongo --return_config alternative
```

To override individual configuration items, append –return_kwargs '{"key:": "value"}' to the salt command.

New in version 2016.3.0.

```
salt '*' test.ping --return mongo --return_kwargs '{"db": "another-salt"}'
```

`salt.returners.mongo_future_return.`**`event_return`**(*events*)
Return events to Mongodb server

`salt.returners.mongo_future_return.`**`get_fun`**(*fun*)
Return the most recent jobs that have executed the named function

`salt.returners.mongo_future_return.`**`get_jid`**(*jid*)
Return the return information associated with a jid

`salt.returners.mongo_future_return.`**`get_jids`**()
Return a list of job ids

`salt.returners.mongo_future_return.`**`get_load`**(*jid*)
Return the load associated with a given job id

`salt.returners.mongo_future_return.`**`get_minions`**()
Return a list of minions

`salt.returners.mongo_future_return.`**`prep_jid`**(*nocache=False, passed_jid=None*)
Do any work necessary to prepare a JID, including sending a custom id

`salt.returners.mongo_future_return.`**`returner`**(*ret*)
Return data to a mongodb server

`salt.returners.mongo_future_return.`**`save_load`**(*jid, load, minions=None*)
Save the load for a given job id

salt.returners.mongo_return

Return data to a mongodb server

Required python modules: pymongo

This returner will send data from the minions to a MongoDB server. To configure the settings for your MongoDB server, add the following lines to the minion config files.

```
mongo.db: <database name>
mongo.host: <server ip address>
mongo.user: <MongoDB username>
mongo.password: <MongoDB user password>
mongo.port: 27017
```

Alternative configuration values can be used by prefacing the configuration. Any values not found in the alternative configuration will be pulled from the default location.

```
alternative.mongo.db: <database name>
alternative.mongo.host: <server ip address>
alternative.mongo.user: <MongoDB username>
alternative.mongo.password: <MongoDB user password>
alternative.mongo.port: 27017
```

To use the mongo returner, append '–return mongo' to the salt command.

```
salt '*' test.ping --return mongo_return
```

To use the alternative configuration, append '–return_config alternative' to the salt command.

New in version 2015.5.0.

```
salt '*' test.ping --return mongo_return --return_config alternative
```

To override individual configuration items, append –return_kwargs '{"key:": "value"}' to the salt command.

New in version 2016.3.0.

```
salt '*' test.ping --return mongo --return_kwargs '{"db": "another-salt"}'
```

To override individual configuration items, append –return_kwargs '{"key:": "value"}' to the salt command.

New in version 2016.3.0.

```
salt '*' test.ping --return mongo --return_kwargs '{"db": "another-salt"}'
```

`salt.returners.mongo_return.`**`get_fun`**(*fun*)
Return the most recent jobs that have executed the named function

`salt.returners.mongo_return.`**`get_jid`**(*jid*)
Return the return information associated with a jid

`salt.returners.mongo_return.`**`prep_jid`**(*nocache=False, passed_jid=None*)
Do any work necessary to prepare a JID, including sending a custom id

`salt.returners.mongo_return.`**`returner`**(*ret*)
Return data to a mongodb server

salt.returners.multi_returner

Read/Write multiple returners

`salt.returners.multi_returner.`**`clean_old_jobs`**()
Clean out the old jobs from all returners (if you have it)

`salt.returners.multi_returner.`**`get_jid`**(*jid*)
Merge the return data from all returners

`salt.returners.multi_returner.`**`get_jids`**()
Return all job data from all returners

`salt.returners.multi_returner.`**`get_load`**(*jid*)
Merge the load data from all returners

`salt.returners.multi_returner.`**`prep_jid`**(*nocache=False*, *passed_jid=None*)

Call both with prep_jid on all returners in multi_returner

TODO: finish this, what do do when you get different jids from 2 returners... since our jids are time based, this make this problem hard, because they aren't unique, meaning that we have to make sure that no one else got the jid and if they did we spin to get a new one, which means "locking" the jid in 2 returners is non-trivial

`salt.returners.multi_returner.`**`returner`**(*load*)

Write return to all returners in multi_returner

`salt.returners.multi_returner.`**`save_load`**(*jid*, *clear_load*, *minions=None*)

Write load to all returners in multi_returner

salt.returners.mysql

Return data to a mysql server

maintainer Dave Boucha <dave@saltstack.com>, Seth House <shouse@saltstack.com>

maturity mature

depends python-mysqldb

platform all

To enable this returner, the minion will need the python client for mysql installed and the following values configured in the minion or master config. These are the defaults:

```
mysql.host: 'salt'
mysql.user: 'salt'
mysql.pass: 'salt'
mysql.db: 'salt'
mysql.port: 3306
```

SSL is optional. The defaults are set to None. If you do not want to use SSL, either exclude these options or set them to None.

```
mysql.ssl_ca: None
mysql.ssl_cert: None
mysql.ssl_key: None
```

Alternative configuration values can be used by prefacing the configuration with *alternative.*. Any values not found in the alternative configuration will be pulled from the default location. As stated above, SSL configuration is optional. The following ssl options are simply for illustration purposes:

```
alternative.mysql.host: 'salt'
alternative.mysql.user: 'salt'
alternative.mysql.pass: 'salt'
alternative.mysql.db: 'salt'
alternative.mysql.port: 3306
alternative.mysql.ssl_ca: '/etc/pki/mysql/certs/localhost.pem'
alternative.mysql.ssl_cert: '/etc/pki/mysql/certs/localhost.crt'
alternative.mysql.ssl_key: '/etc/pki/mysql/certs/localhost.key'
```

Should you wish the returner data to be cleaned out every so often, set *keep_jobs* to the number of hours for the jobs to live in the tables. Setting it to *0* will cause the data to stay in the tables. The default setting for *keep_jobs* is set to *24*.

Should you wish to archive jobs in a different table for later processing, set *archive_jobs* to True. Salt will create 3 archive tables

- *jids_archive*
- *salt_returns_archive*
- *salt_events_archive*

and move the contents of *jids*, *salt_returns*, and *salt_events* that are more than *keep_jobs* hours old to these tables.

Use the following mysql database schema:

```
CREATE DATABASE  `salt`
  DEFAULT CHARACTER SET utf8
  DEFAULT COLLATE utf8_general_ci;

USE `salt`;

--
-- Table structure for table `jids`
--

DROP TABLE IF EXISTS `jids`;
CREATE TABLE `jids` (
  `jid` varchar(255) NOT NULL,
  `load` mediumtext NOT NULL,
  UNIQUE KEY `jid` (`jid`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Table structure for table `salt_returns`
--

DROP TABLE IF EXISTS `salt_returns`;
CREATE TABLE `salt_returns` (
  `fun` varchar(50) NOT NULL,
  `jid` varchar(255) NOT NULL,
  `return` mediumtext NOT NULL,
  `id` varchar(255) NOT NULL,
  `success` varchar(10) NOT NULL,
  `full_ret` mediumtext NOT NULL,
  `alter_time` TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  KEY `id` (`id`),
  KEY `jid` (`jid`),
  KEY `fun` (`fun`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

--
-- Table structure for table `salt_events`
--

DROP TABLE IF EXISTS `salt_events`;
CREATE TABLE `salt_events` (
`id` BIGINT NOT NULL AUTO_INCREMENT,
`tag` varchar(255) NOT NULL,
`data` mediumtext NOT NULL,
`alter_time` TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
`master_id` varchar(255) NOT NULL,
PRIMARY KEY (`id`),
KEY `tag` (`tag`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Required python modules: MySQLdb

To use the mysql returner, append '–return mysql' to the salt command.

```
salt '*' test.ping --return mysql
```

To use the alternative configuration, append '–return_config alternative' to the salt command.

New in version 2015.5.0.

```
salt '*' test.ping --return mysql --return_config alternative
```

To override individual configuration items, append –return_kwargs '{"key:": "value"}' to the salt command.

New in version 2016.3.0.

```
salt '*' test.ping --return mysql --return_kwargs '{"db": "another-salt"}'
```

`salt.returners.mysql.`**`clean_old_jobs`**`()`
Called in the master's event loop every loop_interval. Archives and/or deletes the events and job details from the database. :return:

`salt.returners.mysql.`**`event_return`**`(`*events*`)`
Return event to mysql server

Requires that configuration be enabled via 'event_return' option in master config.

`salt.returners.mysql.`**`get_fun`**`(`*fun*`)`
Return a dict of the last function called for all minions

`salt.returners.mysql.`**`get_jid`**`(`*jid*`)`
Return the information returned when the specified job id was executed

`salt.returners.mysql.`**`get_jids`**`()`
Return a list of all job ids

`salt.returners.mysql.`**`get_jids_filter`**`(`*count, filter_find_job=True*`)`
Return a list of all job ids :param int count: show not more than the count of most recent jobs :param bool filter_find_jobs: filter out 'saltutil.find_job' jobs

`salt.returners.mysql.`**`get_load`**`(`*jid*`)`
Return the load data that marks a specified jid

`salt.returners.mysql.`**`get_minions`**`()`
Return a list of minions

`salt.returners.mysql.`**`prep_jid`**`(`*nocache=False, passed_jid=None*`)`
Do any work necessary to prepare a JID, including sending a custom id

`salt.returners.mysql.`**`returner`**`(`*ret*`)`
Return data to a mysql server

`salt.returners.mysql.`**`save_load`**`(`*jid, load, minions=None*`)`
Save the load to the specified jid id

salt.returners.nagios_nrdp_return

Return salt data to Nagios

The following fields can be set in the minion conf file:

```
nagios.url (required)
nagios.token (required)
nagios.service (optional)
nagios.check_type (optional)
```

Alternative configuration values can be used by prefacing the configuration. Any values not found in the alternative configuration will be pulled from the default location:

```
nagios.url
nagios.token
nagios.service
```

Nagios settings may also be configured as:

```
  nagios:
      url: http://localhost/nrdp
      token: r4nd0mt0k3n
      service: service-check

  alternative.nagios:
      url: http://localhost/nrdp
      token: r4nd0mt0k3n
      service: another-service-check

To use the Nagios returner, append '--return nagios' to the salt command. ex:

.. code-block:: bash

  salt '*' test.ping --return nagios

To use the alternative configuration, append '--return_config alternative' to the
↪salt command. ex:

  salt '*' test.ping --return nagios --return_config alternative
```

To override individual configuration items, append –return_kwargs '{"key:": "value"}' to the salt command.

New in version 2016.3.0.

```
salt '*' test.ping --return nagios --return_kwargs '{"service": "service-name"}'
```

`salt.returners.nagios_nrdp_return.`**`returner`**(*ret*)
Send a message to Nagios with the data

salt.returners.odbc

Return data to an ODBC compliant server. This driver was developed with Microsoft SQL Server in mind, but theoretically could be used to return data to any compliant ODBC database as long as there is a working ODBC driver for it on your minion platform.

maintainer

C. R. Oldham (cr@saltstack.com)

maturity New

depends unixodbc, pyodbc, freetds (for SQL Server)

platform all

To enable this returner the minion will need

On Linux:

unixodbc (http://www.unixodbc.org) pyodbc (*pip install pyodbc*) The FreeTDS ODBC driver for SQL Server (http://www.freetds.org) or another compatible ODBC driver

On Windows:

TBD

unixODBC and FreeTDS need to be configured via /etc/odbcinst.ini and /etc/odbc.ini.

/etc/odbcinst.ini:

```
[TDS]
Description=TDS
Driver=/usr/lib/x86_64-linux-gnu/odbc/libtdsodbc.so
```

(Note the above Driver line needs to point to the location of the FreeTDS shared library. This example is for Ubuntu 14.04.)

/etc/odbc.ini:

```
[TS]
Description = "Salt Returner"
Driver=TDS
Server = <your server ip or fqdn>
Port = 1433
Database = salt
Trace = No
```

Also you need the following values configured in the minion or master config. Configure as you see fit:

```
returner.odbc.dsn: 'TS'
returner.odbc.user: 'salt'
returner.odbc.passwd: 'salt'
```

Alternative configuration values can be used by prefacing the configuration. Any values not found in the alternative configuration will be pulled from the default location:

```
alternative.returner.odbc.dsn: 'TS'
alternative.returner.odbc.user: 'salt'
alternative.returner.odbc.passwd: 'salt'
```

Running the following commands against Microsoft SQL Server in the desired database as the appropriate user should create the database tables correctly. Replace with equivalent SQL for other ODBC-compliant servers

```
  --
  -- Table structure for table 'jids'
  --

  if OBJECT_ID('dbo.jids', 'U') is not null
      DROP TABLE dbo.jids

  CREATE TABLE dbo.jids (
     jid   varchar(255) PRIMARY KEY,
     load  varchar(MAX) NOT NULL
```

(continues on next page)

(continued from previous page)

```
  );

  --
  -- Table structure for table 'salt_returns'
  --
  IF OBJECT_ID('dbo.salt_returns', 'U') IS NOT NULL
      DROP TABLE dbo.salt_returns;

  CREATE TABLE dbo.salt_returns (
     added     datetime not null default (getdate()),
     fun       varchar(100) NOT NULL,
     jid       varchar(255) NOT NULL,
     retval    varchar(MAX) NOT NULL,
     id        varchar(255) NOT NULL,
     success   bit default(0) NOT NULL,
     full_ret  varchar(MAX)
   );

  CREATE INDEX salt_returns_added on dbo.salt_returns(added);
  CREATE INDEX salt_returns_id on dbo.salt_returns(id);
  CREATE INDEX salt_returns_jid on dbo.salt_returns(jid);
  CREATE INDEX salt_returns_fun on dbo.salt_returns(fun);

To use this returner, append '--return odbc' to the salt command.

.. code-block:: bash

  salt '*' status.diskusage --return odbc

To use the alternative configuration, append '--return_config alternative' to the
↪salt command.

.. versionadded:: 2015.5.0

.. code-block:: bash

  salt '*' test.ping --return odbc --return_config alternative
```

To override individual configuration items, append –return_kwargs '{"key:": "value"}' to the salt command.

New in version 2016.3.0.

```
salt '*' test.ping --return odbc --return_kwargs '{"dsn": "dsn-name"}'
```

`salt.returners.odbc.`**`get_fun`**`(`*fun*`)`
: Return a dict of the last function called for all minions

`salt.returners.odbc.`**`get_jid`**`(`*jid*`)`
: Return the information returned when the specified job id was executed

`salt.returners.odbc.`**`get_jids`**`()`
: Return a list of all job ids

`salt.returners.odbc.`**`get_load`**`(`*jid*`)`
: Return the load data that marks a specified jid

`salt.returners.odbc.`**`get_minions`**`()`
: Return a list of minions

`salt.returners.odbc.`**`prep_jid`**(*nocache=False*, *passed_jid=None*)
Do any work necessary to prepare a JID, including sending a custom id

`salt.returners.odbc.`**`returner`**(*ret*)
Return data to an odbc server

`salt.returners.odbc.`**`save_load`**(*jid*, *load*, *minions=None*)
Save the load to the specified jid id

salt.returners.pgjsonb

Return data to a PostgreSQL server with json data stored in Pg's jsonb data type

maintainer Dave Boucha <dave@saltstack.com>, Seth House <shouse@saltstack.com>, C. R. Oldham <cr@saltstack.com>

maturity Stable

depends python-psycopg2

platform all

Note: There are three PostgreSQL returners. Any can function as an external *master job cache*. but each has different features. SaltStack recommends *`returners.pgjsonb`* if you are working with a version of PostgreSQL that has the appropriate native binary JSON types. Otherwise, review *`returners.postgres`* and *`returners.postgres_local_cache`* to see which module best suits your particular needs.

To enable this returner, the minion will need the python client for PostgreSQL installed and the following values configured in the minion or master config. These are the defaults:

```
returner.pgjsonb.host: 'salt'
returner.pgjsonb.user: 'salt'
returner.pgjsonb.pass: 'salt'
returner.pgjsonb.db: 'salt'
returner.pgjsonb.port: 5432
```

SSL is optional. The defaults are set to None. If you do not want to use SSL, either exclude these options or set them to None.

```
returner.pgjsonb.sslmode: None
returner.pgjsonb.sslcert: None
returner.pgjsonb.sslkey: None
returner.pgjsonb.sslrootcert: None
returner.pgjsonb.sslcrl: None
```

New in version 2017.5.0.

Alternative configuration values can be used by prefacing the configuration with *alternative.*. Any values not found in the alternative configuration will be pulled from the default location. As stated above, SSL configuration is optional. The following ssl options are simply for illustration purposes:

```
alternative.pgjsonb.host: 'salt'
alternative.pgjsonb.user: 'salt'
alternative.pgjsonb.pass: 'salt'
alternative.pgjsonb.db: 'salt'
alternative.pgjsonb.port: 5432
```

(continues on next page)

(continued from previous page)

```
alternative.pgjsonb.ssl_ca: '/etc/pki/mysql/certs/localhost.pem'
alternative.pgjsonb.ssl_cert: '/etc/pki/mysql/certs/localhost.crt'
alternative.pgjsonb.ssl_key: '/etc/pki/mysql/certs/localhost.key'
```

Should you wish the returner data to be cleaned out every so often, set `keep_jobs` to the number of hours for the jobs to live in the tables. Setting it to `0` or leaving it unset will cause the data to stay in the tables.

Should you wish to archive jobs in a different table for later processing, set `archive_jobs` to True. Salt will create 3 archive tables;

- `jids_archive`
- `salt_returns_archive`
- `salt_events_archive`

and move the contents of `jids`, `salt_returns`, and `salt_events` that are more than `keep_jobs` hours old to these tables.

New in version 2019.2.0.

Use the following Pg database schema:

```
CREATE DATABASE  salt
  WITH ENCODING 'utf-8';

--
-- Table structure for table `jids`
--
DROP TABLE IF EXISTS jids;
CREATE TABLE jids (
   jid varchar(255) NOT NULL primary key,
   load jsonb NOT NULL
);
CREATE INDEX idx_jids_jsonb on jids
       USING gin (load)
       WITH (fastupdate=on);

--
-- Table structure for table `salt_returns`
--

DROP TABLE IF EXISTS salt_returns;
CREATE TABLE salt_returns (
  fun varchar(50) NOT NULL,
  jid varchar(255) NOT NULL,
  return jsonb NOT NULL,
  id varchar(255) NOT NULL,
  success varchar(10) NOT NULL,
  full_ret jsonb NOT NULL,
  alter_time TIMESTAMP WITH TIME ZONE DEFAULT NOW());

CREATE INDEX idx_salt_returns_id ON salt_returns (id);
CREATE INDEX idx_salt_returns_jid ON salt_returns (jid);
CREATE INDEX idx_salt_returns_fun ON salt_returns (fun);
CREATE INDEX idx_salt_returns_return ON salt_returns
    USING gin (return) with (fastupdate=on);
CREATE INDEX idx_salt_returns_full_ret ON salt_returns
    USING gin (full_ret) with (fastupdate=on);
```

(continues on next page)

(continued from previous page)

```
--
-- Table structure for table `salt_events`
--

DROP TABLE IF EXISTS salt_events;
DROP SEQUENCE IF EXISTS seq_salt_events_id;
CREATE SEQUENCE seq_salt_events_id;
CREATE TABLE salt_events (
    id BIGINT NOT NULL UNIQUE DEFAULT nextval('seq_salt_events_id'),
    tag varchar(255) NOT NULL,
    data jsonb NOT NULL,
    alter_time TIMESTAMP WITH TIME ZONE DEFAULT NOW(),
    master_id varchar(255) NOT NULL);

CREATE INDEX idx_salt_events_tag on
    salt_events (tag);
CREATE INDEX idx_salt_events_data ON salt_events
    USING gin (data) with (fastupdate=on);
```

Required python modules: Psycopg2

To use this returner, append '–return pgjsonb' to the salt command.

```
salt '*' test.ping --return pgjsonb
```

To use the alternative configuration, append '–return_config alternative' to the salt command.

New in version 2015.5.0.

```
salt '*' test.ping --return pgjsonb --return_config alternative
```

To override individual configuration items, append –return_kwargs '{"key:": "value"}' to the salt command.

New in version 2016.3.0.

```
salt '*' test.ping --return pgjsonb --return_kwargs '{"db": "another-salt"}'
```

`salt.returners.pgjsonb.`**`clean_old_jobs`**`()`
: Called in the master's event loop every loop_interval. Archives and/or deletes the events and job details from the database. :return:

`salt.returners.pgjsonb.`**`event_return`**`(`*events*`)`
: Return event to Pg server

Requires that configuration be enabled via 'event_return' option in master config.

`salt.returners.pgjsonb.`**`get_fun`**`(`*fun*`)`
: Return a dict of the last function called for all minions

`salt.returners.pgjsonb.`**`get_jid`**`(`*jid*`)`
: Return the information returned when the specified job id was executed

`salt.returners.pgjsonb.`**`get_jids`**`()`
: Return a list of all job ids

`salt.returners.pgjsonb.`**`get_load`**`(`*jid*`)`
: Return the load data that marks a specified jid

`salt.returners.pgjsonb.`**`get_minions`**`()`
Return a list of minions

`salt.returners.pgjsonb.`**`prep_jid`**`(`*nocache=False, passed_jid=None*`)`
Do any work necessary to prepare a JID, including sending a custom id

`salt.returners.pgjsonb.`**`returner`**`(`*ret*`)`
Return data to a Pg server

`salt.returners.pgjsonb.`**`save_load`**`(`*jid, load, minions=None*`)`
Save the load to the specified jid id

salt.returners.postgres

Return data to a postgresql server

Note: There are three PostgreSQL returners. Any can function as an external *master job cache*. but each has different features. SaltStack recommends `returners.pgjsonb` if you are working with a version of PostgreSQL that has the appropriate native binary JSON types. Otherwise, review `returners.postgres` and `returners.postgres_local_cache` to see which module best suits your particular needs.

maintainer None

maturity New

depends psycopg2

platform all

To enable this returner the minion will need the psycopg2 installed and the following values configured in the minion or master config:

```
returner.postgres.host: 'salt'
returner.postgres.user: 'salt'
returner.postgres.passwd: 'salt'
returner.postgres.db: 'salt'
returner.postgres.port: 5432
```

Alternative configuration values can be used by prefacing the configuration. Any values not found in the alternative configuration will be pulled from the default location:

```
alternative.returner.postgres.host: 'salt'
alternative.returner.postgres.user: 'salt'
alternative.returner.postgres.passwd: 'salt'
alternative.returner.postgres.db: 'salt'
alternative.returner.postgres.port: 5432
```

Running the following commands as the postgres user should create the database correctly:

```
psql << EOF
CREATE ROLE salt WITH PASSWORD 'salt';
CREATE DATABASE salt WITH OWNER salt;
EOF

psql -h localhost -U salt << EOF
--
-- Table structure for table 'jids'
```

(continues on next page)

(continued from previous page)

```
--

DROP TABLE IF EXISTS jids;
CREATE TABLE jids (
  jid   varchar(20) PRIMARY KEY,
  load  text NOT NULL
);

--
-- Table structure for table 'salt_returns'
--

DROP TABLE IF EXISTS salt_returns;
CREATE TABLE salt_returns (
  fun       varchar(50) NOT NULL,
  jid       varchar(255) NOT NULL,
  return    text NOT NULL,
  full_ret  text,
  id        varchar(255) NOT NULL,
  success   varchar(10) NOT NULL,
  alter_time   TIMESTAMP WITH TIME ZONE DEFAULT now()
);

CREATE INDEX idx_salt_returns_id ON salt_returns (id);
CREATE INDEX idx_salt_returns_jid ON salt_returns (jid);
CREATE INDEX idx_salt_returns_fun ON salt_returns (fun);
CREATE INDEX idx_salt_returns_updated ON salt_returns (alter_time);

--
-- Table structure for table `salt_events`
--

DROP TABLE IF EXISTS salt_events;
DROP SEQUENCE IF EXISTS seq_salt_events_id;
CREATE SEQUENCE seq_salt_events_id;
CREATE TABLE salt_events (
    id BIGINT NOT NULL UNIQUE DEFAULT nextval('seq_salt_events_id'),
    tag varchar(255) NOT NULL,
    data text NOT NULL,
    alter_time TIMESTAMP WITH TIME ZONE DEFAULT NOW(),
    master_id varchar(255) NOT NULL
);

CREATE INDEX idx_salt_events_tag on salt_events (tag);

EOF
```

Required python modules: psycopg2

To use the postgres returner, append '–return postgres' to the salt command.

```
salt '*' test.ping --return postgres
```

To use the alternative configuration, append '–return_config alternative' to the salt command.

New in version 2015.5.0.

```
salt '*' test.ping --return postgres --return_config alternative
```

To override individual configuration items, append –return_kwargs '{"key:": "value"}' to the salt command.

New in version 2016.3.0.

```
salt '*' test.ping --return postgres --return_kwargs '{"db": "another-salt"}'
```

`salt.returners.postgres.`**`event_return`**`(`*events*`)`
Return event to Pg server

Requires that configuration be enabled via 'event_return' option in master config.

`salt.returners.postgres.`**`get_fun`**`(`*fun*`)`
Return a dict of the last function called for all minions

`salt.returners.postgres.`**`get_jid`**`(`*jid*`)`
Return the information returned when the specified job id was executed

`salt.returners.postgres.`**`get_jids`**`()`
Return a list of all job ids

`salt.returners.postgres.`**`get_load`**`(`*jid*`)`
Return the load data that marks a specified jid

`salt.returners.postgres.`**`get_minions`**`()`
Return a list of minions

`salt.returners.postgres.`**`prep_jid`**`(`*nocache=False, passed_jid=None*`)`
Do any work necessary to prepare a JID, including sending a custom id

`salt.returners.postgres.`**`returner`**`(`*ret*`)`
Return data to a postgres server

`salt.returners.postgres.`**`save_load`**`(`*jid, load, minions=None*`)`
Save the load to the specified jid id

salt.returners.postgres_local_cache

Use a postgresql server for the master job cache. This helps the job cache to cope with scale.

Note: There are three PostgreSQL returners. Any can function as an external *master job cache*. but each has different features. SaltStack recommends `returners.pgjsonb` if you are working with a version of PostgreSQL that has the appropriate native binary JSON types. Otherwise, review `returners.postgres` and `returners.postgres_local_cache` to see which module best suits your particular needs.

---

maintainer gjredelinghuys@gmail.com

maturity Stable

depends psycopg2

platform all

To enable this returner the minion will need the psycopg2 installed and the following values configured in the master config:

```
master_job_cache: postgres_local_cache
master_job_cache.postgres.host: 'salt'
master_job_cache.postgres.user: 'salt'
master_job_cache.postgres.passwd: 'salt'
master_job_cache.postgres.db: 'salt'
master_job_cache.postgres.port: 5432
```

Running the following command as the postgres user should create the database correctly:

```
psql << EOF
CREATE ROLE salt WITH PASSWORD 'salt';
CREATE DATABASE salt WITH OWNER salt;
EOF
```

In case the postgres database is a remote host, you'll need this command also:

```
ALTER ROLE salt WITH LOGIN;
```

and then:

```
psql -h localhost -U salt << EOF
--
-- Table structure for table 'jids'
--

DROP TABLE IF EXISTS jids;
CREATE TABLE jids (
  jid   varchar(20) PRIMARY KEY,
  started TIMESTAMP WITH TIME ZONE DEFAULT now(),
  tgt_type text NOT NULL,
  cmd text NOT NULL,
  tgt text NOT NULL,
  kwargs text NOT NULL,
  ret text NOT NULL,
  username text NOT NULL,
  arg text NOT NULL,
  fun text NOT NULL
);

--
-- Table structure for table 'salt_returns'
--
-- note that 'success' must not have NOT NULL constraint, since
-- some functions don't provide it.

DROP TABLE IF EXISTS salt_returns;
CREATE TABLE salt_returns (
  added     TIMESTAMP WITH TIME ZONE DEFAULT now(),
  fun       text NOT NULL,
  jid       varchar(20) NOT NULL,
  return    text NOT NULL,
  id        text NOT NULL,
  success   boolean
);
CREATE INDEX ON salt_returns (added);
CREATE INDEX ON salt_returns (id);
CREATE INDEX ON salt_returns (jid);
```

(continues on next page)

(continued from previous page)

```
CREATE INDEX ON salt_returns (fun);

DROP TABLE IF EXISTS salt_events;
CREATE TABLE salt_events (
  id SERIAL,
  tag text NOT NULL,
  data text NOT NULL,
  alter_time TIMESTAMP WITH TIME ZONE DEFAULT now(),
  master_id text NOT NULL
);
CREATE INDEX ON salt_events (tag);
CREATE INDEX ON salt_events (data);
CREATE INDEX ON salt_events (id);
CREATE INDEX ON salt_events (master_id);
EOF
```

Required python modules: psycopg2

`salt.returners.postgres_local_cache.`**`clean_old_jobs`**`()`
Clean out the old jobs from the job cache

`salt.returners.postgres_local_cache.`**`event_return`**`(`*events*`)`
Return event to a postgres server

Require that configuration be enabled via 'event_return' option in master config.

`salt.returners.postgres_local_cache.`**`get_jid`**`(`*jid*`)`
Return the information returned when the specified job id was executed

`salt.returners.postgres_local_cache.`**`get_jids`**`()`
Return a list of all job ids For master job cache this also formats the output and returns a string

`salt.returners.postgres_local_cache.`**`get_load`**`(`*jid*`)`
Return the load data that marks a specified jid

`salt.returners.postgres_local_cache.`**`prep_jid`**`(`*nocache=False, passed_jid=None*`)`
Return a job id and prepare the job id directory This is the function responsible for making sure jids don't collide (unless its passed a jid). So do what you have to do to make sure that stays the case

`salt.returners.postgres_local_cache.`**`returner`**`(`*load*`)`
Return data to a postgres server

`salt.returners.postgres_local_cache.`**`save_load`**`(`*jid, clear_load, minions=None*`)`
Save the load to the specified jid id

salt.returners.pushover_returner

Return salt data via pushover (http://www.pushover.net)

New in version 2016.3.0.

The following fields can be set in the minion conf file:

```
pushover.user (required)
pushover.token (required)
pushover.title (optional)
pushover.device (optional)
pushover.priority (optional)
```

(continues on next page)

(continued from previous page)

```
pushover.expire (optional)
pushover.retry (optional)
pushover.profile (optional)
```

Alternative configuration values can be used by prefacing the configuration. Any values not found in the alternative configuration will be pulled from the default location:

```
alternative.pushover.user
alternative.pushover.token
alternative.pushover.title
alternative.pushover.device
alternative.pushover.priority
alternative.pushover.expire
alternative.pushover.retry
```

PushOver settings may also be configured as:

```
  pushover:
      user: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
      token: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
      title: Salt Returner
      device: phone
      priority: -1
      expire: 3600
      retry: 5

  alternative.pushover:
      user: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
      token: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
      title: Salt Returner
      device: phone
      priority: 1
      expire: 4800
      retry: 2

  pushover_profile:
      pushover.token: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

  pushover:
      user: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
      profile: pushover_profile

  alternative.pushover:
      user: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
      profile: pushover_profile

To use the PushOver returner, append '--return pushover' to the salt command. ex:

.. code-block:: bash

  salt '*' test.ping --return pushover

To use the alternative configuration, append '--return_config alternative' to the
↪salt command. ex:

  salt '*' test.ping --return pushover --return_config alternative
```

To override individual configuration items, append –return_kwargs '{"key:": "value"}' to the salt command.

```
salt '*' test.ping --return pushover --return_kwargs '{"title": "Salt is awesome!"}'
```

`salt.returners.pushover_returner.`**`returner`**`(`*ret*`)`
Send an PushOver message with the data

salt.returners.rawfile_json

Take data from salt and "return" it into a raw file containing the json, with one line per event.

Add the following to the minion or master configuration file.

```
rawfile_json.filename: <path_to_output_file>
```

Default is `/var/log/salt/events`.

Common use is to log all events on the master. This can generate a lot of noise, so you may wish to configure batch processing and/or configure the *`event_return_whitelist`* or *`event_return_blacklist`* to restrict the events that are written.

`salt.returners.rawfile_json.`**`event_return`**`(`*events*`)`
Write event data (return data and non-return data) to file on the master.

`salt.returners.rawfile_json.`**`returner`**`(`*ret*`)`
Write the return data to a file on the minion.

salt.returners.redis_return

Return data to a redis server

To enable this returner the minion will need the python client for redis installed and the following values configured in the minion or master config, these are the defaults:

```
redis.db: '0'
redis.host: 'salt'
redis.port: 6379
```

New in version 2018.3.1: Alternatively a UNIX socket can be specified by *unix_socket_path*:

```
redis.db: '0'
redis.unix_socket_path: /var/run/redis/redis.sock
```

Cluster Mode Example:

```
redis.db: '0'
redis.cluster_mode: true
redis.cluster.skip_full_coverage_check: true
redis.cluster.startup_nodes:
  - host: redis-member-1
    port: 6379
  - host: redis-member-2
    port: 6379
```

Alternative configuration values can be used by prefacing the configuration. Any values not found in the alternative configuration will be pulled from the default location:

```
alternative.redis.db: '0'
alternative.redis.host: 'salt'
alternative.redis.port: 6379
```

To use the redis returner, append '–return redis' to the salt command.

```
salt '*' test.ping --return redis
```

To use the alternative configuration, append '–return_config alternative' to the salt command.

New in version 2015.5.0.

```
salt '*' test.ping --return redis --return_config alternative
```

To override individual configuration items, append –return_kwargs '{"key:": "value"}' to the salt command.

New in version 2016.3.0.

```
salt '*' test.ping --return redis --return_kwargs '{"db": "another-salt"}'
```

Redis Cluster Mode Options:

cluster_mode: `False` Whether cluster_mode is enabled or not

cluster.startup_nodes: A list of host, port dictionaries pointing to cluster members. At least one is required but multiple nodes are better

```
cache.redis.cluster.startup_nodes
  - host: redis-member-1
    port: 6379
  - host: redis-member-2
    port: 6379
```

cluster.skip_full_coverage_check: `False` Some cluster providers restrict certain redis commands such as CONFIG for enhanced security. Set this option to true to skip checks that required advanced privileges.

> **Note**: Most cloud hosted redis clusters will require this to be set to `True`

`salt.returners.redis_return.`**`clean_old_jobs`**`()`

Clean out minions's return data for old jobs.

Normally, hset 'ret:<jid>' are saved with a TTL, and will eventually get cleaned by redis.But for jobs with some very late minion return, the corresponding hset's TTL will be refreshed to a too late timestamp, we'll do manually cleaning here.

`salt.returners.redis_return.`**`get_fun`**`(`*fun*`)`

Return a dict of the last function called for all minions

`salt.returners.redis_return.`**`get_jid`**`(`*jid*`)`

Return the information returned when the specified job id was executed

`salt.returners.redis_return.`**`get_jids`**`()`

Return a dict mapping all job ids to job information

`salt.returners.redis_return.`**`get_load`**`(`*jid*`)`

Return the load data that marks a specified jid

`salt.returners.redis_return.`**`get_minions`**`()`

Return a list of minions

`salt.returners.redis_return.`**`prep_jid`**(*nocache=False, passed_jid=None*)
Do any work necessary to prepare a JID, including sending a custom id

`salt.returners.redis_return.`**`returner`**(*ret*)
Return data to a redis data store

`salt.returners.redis_return.`**`save_load`**(*jid, load, minions=None*)
Save the load to the specified jid

salt.returners.sentry_return

Salt returner that reports execution results back to sentry. The returner will inspect the payload to identify errors and flag them as such.

Pillar needs something like:

```
raven:
  servers:
    - http://192.168.1.1
    - https://sentry.example.com
  public_key: deadbeefdeadbeefdeadbeefdeadbeef
  secret_key: beefdeadbeefdeadbeefdeadbeefdead
  project: 1
  tags:
    - os
    - master
    - saltversion
    - cpuarch
```

or using a dsn:

```
raven:
  dsn: https://aaaa:bbbb@app.getsentry.com/12345
  tags:
    - os
    - master
    - saltversion
    - cpuarch
```

https://pypi.python.org/pypi/raven must be installed.

The pillar can be hidden on sentry return by setting hide_pillar: true.

The tags list (optional) specifies grains items that will be used as sentry tags, allowing tagging of events in the sentry ui.

To report only errors to sentry, set report_errors_only: true.

`salt.returners.sentry_return.`**`prep_jid`**(*nocache=False, passed_jid=None*)
Do any work necessary to prepare a JID, including sending a custom id

`salt.returners.sentry_return.`**`returner`**(*ret*)
Log outcome to sentry. The returner tries to identify errors and report them as such. All other messages will be reported at info level. Failed states will be appended as separate list for convenience.

salt.returners.slack_returner

Return salt data via slack

New in version 2015.5.0.

The following fields can be set in the minion conf file:

```
slack.channel (required)
slack.api_key (required)
slack.username (required)
slack.as_user (required to see the profile picture of your bot)
slack.profile (optional)
slack.changes(optional, only show changes and failed states)
slack.only_show_failed(optional, only show failed states)
slack.yaml_format(optional, format the json in yaml format)
```

Alternative configuration values can be used by prefacing the configuration. Any values not found in the alternative configuration will be pulled from the default location:

```
slack.channel
slack.api_key
slack.username
slack.as_user
```

Slack settings may also be configured as:

```
slack:
    channel: RoomName
    api_key: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
    username: user
    as_user: true

alternative.slack:
    room_id: RoomName
    api_key: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
    from_name: user@email.com

slack_profile:
    slack.api_key: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
    slack.from_name: user@email.com

slack:
    profile: slack_profile
    channel: RoomName

alternative.slack:
    profile: slack_profile
    channel: RoomName
```

To use the Slack returner, append '–return slack' to the salt command.

```
salt '*' test.ping --return slack
```

To use the alternative configuration, append '–return_config alternative' to the salt command.

```
salt '*' test.ping --return slack --return_config alternative
```

To override individual configuration items, append –return_kwargs '{"key:": "value"}' to the salt command.

New in version 2016.3.0.

```
salt '*' test.ping --return slack --return_kwargs '{"channel": "#random"}'
```

`salt.returners.slack_returner.`**`returner`**`(`*ret*`)`
Send an slack message with the data

salt.returners.sms_return

Return data by SMS.

New in version 2015.5.0.

maintainer Damian Myerscough

maturity new

depends twilio

platform all

To enable this returner the minion will need the python twilio library installed and the following values configured in the minion or master config:

```
twilio.sid: 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
twilio.token: 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'
twilio.to: '+1415XXXXXXX'
twilio.from: '+1650XXXXXXX'
```

To use the sms returner, append '–return sms' to the salt command.

```
salt '*' test.ping --return sms
```

`salt.returners.sms_return.`**`returner`**`(`*ret*`)`
Return a response in an SMS message

salt.returners.smtp_return

Return salt data via email

The following fields can be set in the minion conf file. Fields are optional unless noted otherwise.

- `from` (required) The name/address of the email sender.
- **`to` (required) The names/addresses of the email recipients**; comma-delimited. For example: `you@example.com,someoneelse@example.com`.
- `host` (required) The SMTP server hostname or address.
- `port` The SMTP server port; defaults to `25`.
- **`username` The username used to authenticate to the server. If specified a** password is also required. It is recommended but not required to also use TLS with this option.
- `password` The password used to authenticate to the server.
- `tls` Whether to secure the connection using TLS; defaults to `False`
- `subject` The email subject line.
- **`fields` Which fields from the returned data to include in the subject line** of the email; comma-delimited. For example: `id,fun`. Please note, *the subject line is not encrypted*.

- **gpgowner A user's `~/.gpg` directory. This must contain a gpg** public key matching the address the mail is sent to. If left unset, no encryption will be used. Requires **`python-gnupg`** to be installed.
- `template` The path to a file to be used as a template for the email body.
- **`renderer` A Salt renderer, or render-pipe, to use to render the email** template. Default `jinja`.

Below is an example of the above settings in a Salt Minion configuration file:

```
smtp.from: me@example.net
smtp.to: you@example.com
smtp.host: localhost
smtp.port: 1025
```

Alternative configuration values can be used by prefacing the configuration. Any values not found in the alternative configuration will be pulled from the default location. For example:

```
alternative.smtp.username: saltdev
alternative.smtp.password: saltdev
alternative.smtp.tls: True
```

To use the SMTP returner, append '–return smtp' to the `salt` command.

```
salt '*' test.ping --return smtp
```

To use the alternative configuration, append '–return_config alternative' to the `salt` command.

New in version 2015.5.0.

```
salt '*' test.ping --return smtp --return_config alternative
```

To override individual configuration items, append –return_kwargs '{"key:": "value"}' to the `salt` command.

New in version 2016.3.0.

```
salt '*' test.ping --return smtp --return_kwargs '{"to": "user@domain.com"}'
```

An easy way to test the SMTP returner is to use the development SMTP server built into Python. The command below will start a single-threaded SMTP server that prints any email it receives to the console.

```
python -m smtpd -n -c DebuggingServer localhost:1025
```

New in version 2016.11.0.

It is possible to send emails with selected Salt events by configuring `event_return` option for Salt Master. For example:

```
event_return: smtp

event_return_whitelist:
  - salt/key

smtp.from: me@example.net
smtp.to: you@example.com
smtp.host: localhost
smtp.subject: 'Salt Master {{act}}ed key from Minion ID: {{id}}'
smtp.template: /srv/salt/templates/email.j2
```

Also you need to create additional file `/srv/salt/templates/email.j2` with email body template:

```
act: {{act}}
id: {{id}}
result: {{result}}
```

This configuration enables Salt Master to send an email when accepting or rejecting minions keys.

`salt.returners.smtp_return.`**`event_return`**(*events*)
Return event data via SMTP

`salt.returners.smtp_return.`**`prep_jid`**(*nocache=False, passed_jid=None*)
Do any work necessary to prepare a JID, including sending a custom id

`salt.returners.smtp_return.`**`returner`**(*ret*)
Send an email with the data

salt.returners.splunk module

Send json response data to Splunk via the HTTP Event Collector Requires the following config values to be specified in config or pillar:

```
splunk_http_forwarder:
  token: <splunk_http_forwarder_token>
  indexer: <hostname/IP of Splunk indexer>
  sourcetype: <Destination sourcetype for data>
  index: <Destination index for data>
```

Run a test by using `salt-call test.ping --return splunk`

Written by Scott Pack (github.com/scottjpack)

`class` `salt.returners.splunk.`**`http_event_collector`**(*token, http_event_server, host='', http_event_port='8088', http_event_server_ssl=True, max_bytes=100000*)

`batchEvent`(*payload, eventtime=''*)

`flushBatch`()

`sendEvent`(*payload, eventtime=''*)

`salt.returners.splunk.`**`returner`**(*ret*)
Send a message to Splunk via the HTTP Event Collector

salt.returners.sqlite3

Insert minion return data into a sqlite3 database

maintainer Mickey Malone <mickey.malone@gmail.com>

maturity New

depends None

platform All

Sqlite3 is a serverless database that lives in a single file. In order to use this returner the database file must exist, have the appropriate schema defined, and be accessible to the user whom the minion process is running as. This returner requires the following values configured in the master or minion config:

```
sqlite3.database: /usr/lib/salt/salt.db
sqlite3.timeout: 5.0
```

Alternative configuration values can be used by prefacing the configuration. Any values not found in the alternative configuration will be pulled from the default location:

```
alternative.sqlite3.database: /usr/lib/salt/salt.db
alternative.sqlite3.timeout: 5.0
```

Use the commands to create the sqlite3 database and tables:

```
sqlite3 /usr/lib/salt/salt.db << EOF
--
-- Table structure for table 'jids'
--

CREATE TABLE jids (
  jid TEXT PRIMARY KEY,
  load TEXT NOT NULL
  );

--
-- Table structure for table 'salt_returns'
--

CREATE TABLE salt_returns (
  fun TEXT KEY,
  jid TEXT KEY,
  id TEXT KEY,
  fun_args TEXT,
  date TEXT NOT NULL,
  full_ret TEXT NOT NULL,
  success TEXT NOT NULL
  );
EOF
```

To use the sqlite returner, append '–return sqlite3' to the salt command.

```
salt '*' test.ping --return sqlite3
```

To use the alternative configuration, append '–return_config alternative' to the salt command.

New in version 2015.5.0.

```
salt '*' test.ping --return sqlite3 --return_config alternative
```

To override individual configuration items, append –return_kwargs '{"key:": "value"}' to the salt command.

New in version 2016.3.0.

```
salt '*' test.ping --return sqlite3 --return_kwargs '{"db": "/var/lib/salt/another-
↪salt.db"}'
```

`salt.returners.sqlite3_return.`**`get_fun`**(*fun*)
: Return a dict of the last function called for all minions

`salt.returners.sqlite3_return.`**`get_jid`**(*jid*)
: Return the information returned from a specified jid

`salt.returners.sqlite3_return.`**`get_jids`**`()`
Return a list of all job ids

`salt.returners.sqlite3_return.`**`get_load`**`(`*jid*`)`
Return the load from a specified jid

`salt.returners.sqlite3_return.`**`get_minions`**`()`
Return a list of minions

`salt.returners.sqlite3_return.`**`prep_jid`**`(`*nocache=False, passed_jid=None*`)`
Do any work necessary to prepare a JID, including sending a custom id

`salt.returners.sqlite3_return.`**`returner`**`(`*ret*`)`
Insert minion return data into the sqlite3 database

`salt.returners.sqlite3_return.`**`save_load`**`(`*jid, load, minions=None*`)`
Save the load to the specified jid

salt.returners.syslog_return

Return data to the host operating system's syslog facility

To use the syslog returner, append '–return syslog' to the salt command.

```
salt '*' test.ping --return syslog
```

The following fields can be set in the minion conf file:

```
syslog.level (optional, Default: LOG_INFO)
syslog.facility (optional, Default: LOG_USER)
syslog.tag (optional, Default: salt-minion)
syslog.options (list, optional, Default: [])
```

Available levels, facilities, and options can be found in the `syslog` docs for your python version.

Note: The default tag comes from `sys.argv[0]` which is usually "salt-minion" but could be different based on the specific environment.

Configuration example:

```
syslog.level: 'LOG_ERR'
syslog.facility: 'LOG_DAEMON'
syslog.tag: 'mysalt'
syslog.options:
  - LOG_PID
```

Of course you can also nest the options:

```
syslog:
  level: 'LOG_ERR'
  facility: 'LOG_DAEMON'
  tag: 'mysalt'
  options:
    - LOG_PID
```

Alternative configuration values can be used by prefacing the configuration. Any values not found in the alternative configuration will be pulled from the default location:

```
alternative.syslog.level: 'LOG_WARN'
alternative.syslog.facility: 'LOG_NEWS'
```

To use the alternative configuration, append `--return_config alternative` to the salt command.

New in version 2015.5.0.

```
salt '*' test.ping --return syslog --return_config alternative
```

To override individual configuration items, append –return_kwargs '{"key:": "value"}' to the salt command.

New in version 2016.3.0.

```
salt '*' test.ping --return syslog --return_kwargs '{"level": "LOG_DEBUG"}'
```

Note: Syslog server implementations may have limits on the maximum record size received by the client. This may lead to job return data being truncated in the syslog server's logs. For example, for rsyslog on RHEL-based systems, the default maximum record size is approximately 2KB (which return data can easily exceed). This is configurable in rsyslog.conf via the $MaxMessageSize config parameter. Please consult your syslog implmentation's documentation to determine how to adjust this limit.

`salt.returners.syslog_return.`**`prep_jid`**`(`*nocache=False, passed_jid=None*`)`
Do any work necessary to prepare a JID, including sending a custom id

`salt.returners.syslog_return.`**`returner`**`(`*ret*`)`
Return data to the local syslog

salt.returners.telegram_return

Return salt data via Telegram.

The following fields can be set in the minion conf file:

```
telegram.chat_id (required)
telegram.token (required)
```

Telegram settings may also be configured as:

```
telegram:
  chat_id: 000000000
  token: 000000000:xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

To use the Telegram return, append '–return telegram' to the salt command.

```
salt '*' test.ping --return telegram
```

`salt.returners.telegram_return.`**`returner`**`(`*ret*`)`
Send a Telegram message with the data.

Parameters **`ret`** – The data to be sent.

Returns Boolean if message was sent successfully.

salt.returners.xmpp_return

Return salt data via xmpp

depends sleekxmpp >= 1.3.1

The following fields can be set in the minion conf file:

```
xmpp.jid (required)
xmpp.password (required)
xmpp.recipient (required)
xmpp.profile (optional)
```

Alternative configuration values can be used by prefacing the configuration. Any values not found in the alternative configuration will be pulled from the default location:

```
xmpp.jid
xmpp.password
xmpp.recipient
xmpp.profile
```

XMPP settings may also be configured as:

```
xmpp:
    jid: user@xmpp.domain.com/resource
    password: password
    recipient: user@xmpp.example.com

alternative.xmpp:
    jid: user@xmpp.domain.com/resource
    password: password
    recipient: someone@xmpp.example.com

xmpp_profile:
    xmpp.jid: user@xmpp.domain.com/resource
    xmpp.password: password

xmpp:
    profile: xmpp_profile
    recipient: user@xmpp.example.com

alternative.xmpp:
    profile: xmpp_profile
    recipient: someone-else@xmpp.example.com
```

To use the XMPP returner, append '–return xmpp' to the salt command.

```
salt '*' test.ping --return xmpp
```

To use the alternative configuration, append '–return_config alternative' to the salt command.

New in version 2015.5.0.

```
salt '*' test.ping --return xmpp --return_config alternative
```

To override individual configuration items, append –return_kwargs '{"key:": "value"}' to the salt command.

New in version 2016.3.0.

```
salt '*' test.ping --return xmpp --return_kwargs '{"recipient": "someone-else@xmpp.
↪example.com"}'
```

class `salt.returners.xmpp_return.`**`SendMsgBot`**(*jid, password, recipient, msg*)

`start`(*event*)

`salt.returners.xmpp_return.`**`returner`**(*ret*)
Send an xmpp message with the data

salt.returners.zabbix_return module

Return salt data to Zabbix

The following Type: "Zabbix trapper" with "Type of information" Text items are required:

```
Key: salt.trap.info
Key: salt.trap.warning
Key: salt.trap.high
```

To use the Zabbix returner, append '–return zabbix' to the salt command. ex:

```
salt '*' test.ping --return zabbix
```

`salt.returners.zabbix_return.`**`returner`**(*ret*)

`salt.returners.zabbix_return.`**`save_load`**(*jid, load, minions=None*)
Included for API consistency

`salt.returners.zabbix_return.`**`zabbix_send`**(*key, output*)

`salt.returners.zabbix_return.`**`zbx`**()

3.25 Renderers

The Salt state system operates by gathering information from common data types such as lists, dictionaries, and strings that would be familiar to any developer.

Salt Renderers translate input from the format in which it is written into Python data structures.

The default renderer is set in the master/minion configuration file using the *`renderer`* config option, which defaults to `jinja|yaml`.

3.25.1 Two Kinds of Renderers

Renderers fall into one of two categories, based on what they output: text or data. The one exception to this would be the *`pure python`* renderer, which can be used in either capacity.

Text Renderers

A text renderer returns text. These include templating engines such as *`jinja`*, *`mako`*, and *`genshi`*, as well as the *`gpg`* renderer. The following are all text renderers:

- `aws_kms`

- *cheetah*
- *genshi*
- *gpg*
- *jinja*
- *mako*
- nacl
- *pass*
- *py*
- *wempy*

Data Renderers

A data renderer returns a Python data structure (typically a dictionary). The following are all data renderers:

- *dson*
- *hjson*
- *json5*
- *json*
- *pydsl*
- *pyobjects*
- *py*
- *stateconf*
- *yamlex*
- *yaml*

3.25.2 Overriding the Default Renderer

It can sometimes be beneficial to write an SLS file using a renderer other than the default one. This can be done by using a "shebang"-like syntax on the first line of the SLS file:

Here is an example of using the *pure python* renderer to install a package:

```
#!py

def run():
    '''
    Install version 1.5-1.el7 of package "python-foo"
    '''
    return {
        'include': ['python'],
        'python-foo': {
            'pkg.installed': [
                {'version': '1.5-1.el7'},
            ]
        }
    }
```

This would be equivalent to the following:

```
include:
  - python

python-foo:
  pkg.installed:
    - version: '1.5-1.el7'
```

3.25.3 Composing Renderers (a.k.a. The "Render Pipeline")

A render pipeline can be composed from other renderers by connecting them in a series of "pipes" (i.e. `|`). The renderers will be evaluated from left to right, with each renderer receiving the result of the previous renderer's execution.

Take for example the default renderer (`jinja|yaml`). The file is evaluated first a jinja template, and the result of that template is evaluated as a YAML document.

Other render pipeline combinations include:

`yaml` Just YAML, no templating.

`mako|yaml` This passes the input to the `mako` renderer, with its output fed into the `yaml` renderer.

`jinja|mako|yaml` This one allows you to use both jinja and mako templating syntax in the input and then parse the final rendered output as YAML.

The following is a contrived example SLS file using the `jinja|mako|yaml` render pipeline:

```
#!jinja|mako|yaml

An_Example:
  cmd.run:
    - name: |
        echo "Using Salt ${grains['saltversion']}" \
             "from path {{grains['saltpath']}}."
    - cwd: /

<%doc> ${...} is Mako's notation, and so is this comment. </%doc>
{#     Similarly, {{...}} is Jinja's notation, and so is this comment. #}
```

Important: Keep in mind that not all renderers can be used alone or with any other renderers. For example, text renderers shouldn't be used alone as their outputs are just strings, which still need to be parsed by another renderer to turn them into Python data structures.

For example, it would not make sense to use `yaml|jinja` because the output of the *`yaml`* renderer is a Python data structure, and the *`jinja`* renderer only accepts text as input.

Therefore, when combining renderers, you should know what each renderer accepts as input and what it returns as output. One way of thinking about it is that you can chain together multiple text renderers, but the pipeline *must* end in a data renderer. Similarly, since the text renderers in Salt don't accept data structures as input, a text renderer should usually not come after a data renderer. It's technically *possible* to write a renderer that takes a data structure as input and returns a string, but no such renderer is distributed with Salt.

3.25.4 Writing Renderers

A custom renderer must be a Python module which implements a `render` function. This function must implement three positional arguments:

1. `data` - Can be called whatever you like. This is the input to be rendered.
2. `saltenv`
3. `sls`

The first is the important one, and the 2nd and 3rd must be included since Salt needs to pass this info to each render, even though it is only used by template renderers.

Renderers should be written so that the `data` argument can accept either strings or file-like objects as input. For example:

```
import mycoolmodule
from salt.ext import six

def render(data, saltenv='base', sls='', **kwargs):
    if not isinstance(data, six.string_types):
        # Read from file-like object
        data = data.read()

    return mycoolmodule.do_something(data)
```

Custom renderers should be placed within `salt://_renderers/`, so that they can be synced to minions. They are synced when any of the following are run:

- *state.apply*
- *saltutil.sync_renderers*
- *saltutil.sync_all*

Any custom renderers which have been synced to a minion, that are named the same as one of Salt's default set of renderers, will take the place of the default renderer with the same name.

Note: Renderers can also be synced from `salt://_renderers/` to the Master using either the *saltutil.sync_renderers* or *saltutil.sync_all* runner function.

3.25.5 Examples

The best place to find examples of renderers is in the Salt source code.

Documentation for renderers included with Salt can be found here:

https://github.com/saltstack/salt/blob/master/salt/renderers

Here is a simple YAML renderer example:

```
import salt.utils.yaml
from salt.utils.yamlloader import SaltYamlSafeLoader
from salt.ext import six

def render(yaml_data, saltenv='', sls='', **kws):
    if not isinstance(yaml_data, six.string_types):
```

(continues on next page)

(continued from previous page)

```
        yaml_data = yaml_data.read()
    data = salt.utils.yaml.safe_load(yaml_data)
    return data if data else {}
```

3.25.6 Full List of Renderers

renderer modules

cheetah	Cheetah Renderer for Salt
dson	DSON Renderer for Salt
genshi	Genshi Renderer for Salt
gpg	Renderer that will decrypt GPG ciphers
hjson	hjson renderer for Salt
jinja	Jinja loading utils to enable a more powerful backend for jinja templates
json	JSON Renderer for Salt
json5	JSON5 Renderer for Salt
mako	Mako Renderer for Salt
msgpack	
pass	Pass Renderer for Salt
py	Pure python state renderer
pydsl	A Python-based DSL
pyobjects	Python renderer that includes a Pythonic Object based interface
stateconf	A flexible renderer that takes a templating engine and a data format
wempy	
yaml	YAML Renderer for Salt
yamlex	

salt.renderers.cheetah

Cheetah Renderer for Salt

`salt.renderers.cheetah.`**`render`**(*cheetah_data*, *saltenv='base'*, *sls=''*, *method='xml'*, ***kws*)

Render a Cheetah template.

Return type A Python data structure

salt.renderers.dson

DSON Renderer for Salt

This renderer is intended for demonstration purposes. Information on the DSON spec can be found here.

This renderer requires Dogeon (installable via pip)

`salt.renderers.dson.`**`render`**(*dson_input*, *saltenv='base'*, *sls=''*, ***kwargs*)

Accepts DSON data as a string or as a file object and runs it through the JSON parser.

Return type A Python data structure

salt.renderers.genshi

Genshi Renderer for Salt

`salt.renderers.genshi.`**`render`**(*genshi_data, saltenv='base', sls='', method='xml', **kws*)

Render a Genshi template. A method should be passed in as part of the kwargs. If no method is passed in, xml is assumed. Valid methods are:

Note that the `text` method will call `NewTextTemplate`. If `oldtext` is desired, it must be called explicitly

Return type A Python data structure

salt.renderers.gpg

Renderer that will decrypt GPG ciphers

Any key in the SLS file can be a GPG cipher, and this renderer will decrypt it before passing it off to Salt. This allows you to safely store secrets in source control, in such a way that only your Salt master can decrypt them and distribute them only to the minions that need them.

The typical use-case would be to use ciphers in your pillar data, and keep a secret key on your master. You can put the public key in source control so that developers can add new secrets quickly and easily.

This renderer requires the gpg binary. No python libraries are required as of the 2015.8.0 release.

Setup

To set things up, first generate a keypair. On the master, run the following:

```
# mkdir -p /etc/salt/gpgkeys
# chmod 0700 /etc/salt/gpgkeys
# gpg --gen-key --homedir /etc/salt/gpgkeys
```

Do not supply a password for the keypair, and use a name that makes sense for your application. Be sure to back up the `gpgkeys` directory someplace safe!

Note: Unfortunately, there are some scenarios - for example, on virtual machines which don't have real hardware - where insufficient entropy causes key generation to be extremely slow. In these cases, there are usually means of increasing the system entropy. On virtualised Linux systems, this can often be achieved by installing the `rng-tools` package.

Export the Public Key

```
# gpg --homedir /etc/salt/gpgkeys --armor --export <KEY-NAME> > exported_pubkey.gpg
```

Import the Public Key

To encrypt secrets, copy the public key to your local machine and run:

```
$ gpg --import exported_pubkey.gpg
```

To generate a cipher from a secret:

```
$ echo -n "supersecret" | gpg --armor --batch --trust-model always --encrypt -r <KEY-
↪name>
```

To apply the renderer on a file-by-file basis add the following line to the top of any pillar with gpg data in it:

```
#!yaml|gpg
```

Now with your renderer configured, you can include your ciphers in your pillar data like so:

```
#!yaml|gpg

a-secret: |
  -----BEGIN PGP MESSAGE-----
  Version: GnuPG v1

  hQEMAweRHKaPCfNeAQf9GLTN16hCfXAbPwU6BbBK0unOc7i9/etGuVc5CyU9Q6um
  QuetdvQVLFO/HkrC4lgeNQdM6D9E8PKonMlgJPyUvC8ggxhj0/IPFEKmrsnv2k6+
  cnEfmVexS7o/U1VOVjoyUeliMCJlAz/30RXaME49Cpi6No2+vKD8a4q4nZN1UZcG
  RhkhC0S22zNxOXQ38TBkmtJcqxnqT6YWKTUsjVubW3bVC+u2HGqJHu79wmwuN8tz
  m4wBkfCAd8Eyo2jEnWQcM4TcXiF01XPL4z4g1/9AAxh+Q4d8RIRP4fbw7ct4nCJv
  Gr9v2DTF7HNigIMl4ivMIn9fp+EZurJNiQskLgNbktJGAeEKYkqX5iCuB1b693hJ
  FKlwHiJt5yA8X2dDtfk8/Ph1Jx2TwGS+lGjlZaNqp3R1xuAZzXzZMLyZDe5+i3RJ
  skqmFTbOiA===Eqsm
  -----END PGP MESSAGE-----
```

Encrypted CLI Pillar Data

New in version 2016.3.0.

Functions like *state.highstate* and *state.sls* allow for pillar data to be passed on the CLI.

```
salt myminion state.highstate pillar="{'mypillar': 'foo'}"
```

Starting with the 2016.3.0 release of Salt, it is now possible for this pillar data to be GPG-encrypted, and to use the GPG renderer to decrypt it.

Replacing Newlines

To pass encrypted pillar data on the CLI, the ciphertext must have its newlines replaced with a literal backslash-n (`\n`), as newlines are not supported within Salt CLI arguments. There are a number of ways to do this:

With awk or Perl:

```
# awk
ciphertext=`echo -n "supersecret" | gpg --armor --batch --trust-model always --
↪encrypt -r user@domain.com | awk '{printf "%s\\n",$0} END {print ""}'`
# Perl
ciphertext=`echo -n "supersecret" | gpg --armor --batch --trust-model always --
↪encrypt -r user@domain.com | perl -pe 's/\n/\\n/g'`
```

With Python:

```
import subprocess

secret, stderr = subprocess.Popen(
    ['gpg', '--armor', '--batch', '--trust-model', 'always', '--encrypt',
     '-r', 'user@domain.com'],
    stdin=subprocess.PIPE,
    stdout=subprocess.PIPE,
    stderr=subprocess.PIPE).communicate(input='supersecret')

if secret:
    print(secret.replace('\n', r'\n'))
else:
    raise ValueError('No ciphertext found: {0}'.format(stderr))
```

```
ciphertext=`python /path/to/script.py`
```

The ciphertext can be included in the CLI pillar data like so:

```
salt myminion state.sls secretstuff pillar_enc=gpg pillar="{secret_pillar: '
↪$ciphertext'}"
```

The `pillar_enc=gpg` argument tells Salt that there is GPG-encrypted pillar data, so that the CLI pillar data is passed through the GPG renderer, which will iterate recursively though the CLI pillar dictionary to decrypt any encrypted values.

Encrypting the Entire CLI Pillar Dictionary

If several values need to be encrypted, it may be more convenient to encrypt the entire CLI pillar dictionary. Again, this can be done in several ways:

With awk or Perl:

```
# awk
ciphertext=`echo -n "{'secret_a': 'CorrectHorseBatteryStaple', 'secret_b': 'GPG is
↪fun!'}" | gpg --armor --batch --trust-model always --encrypt -r user@domain.com |
↪awk '{printf "%s\\n",$0} END {print ""}'`
# Perl
ciphertext=`echo -n "{'secret_a': 'CorrectHorseBatteryStaple', 'secret_b': 'GPG is
↪fun!'}" | gpg --armor --batch --trust-model always --encrypt -r user@domain.com |
↪perl -pe 's/\n/\\n/g'`
```

With Python:

```
import subprocess

pillar_data = {'secret_a': 'CorrectHorseBatteryStaple',
               'secret_b': 'GPG is fun!'}

secret, stderr = subprocess.Popen(
    ['gpg', '--armor', '--batch', '--trust-model', 'always', '--encrypt',
     '-r', 'user@domain.com'],
    stdin=subprocess.PIPE,
    stdout=subprocess.PIPE,
    stderr=subprocess.PIPE).communicate(input=repr(pillar_data))

```

(continues on next page)

(continued from previous page)

```
if secret:
    print(secret.replace('\n', r'\n'))
else:
    raise ValueError('No ciphertext found: {0}'.format(stderr))
```

```
ciphertext=`python /path/to/script.py`
```

With the entire pillar dictionary now encrypted, it can be included in the CLI pillar data like so:

```
salt myminion state.sls secretstuff pillar_enc=gpg pillar="$ciphertext"
```

`salt.renderers.gpg.`**`render`**(*gpg_data*, *saltenv='base'*, *sls=''*, *argline=''*, ***kwargs*)

Create a gpg object given a gpg_keydir, and then use it to try to decrypt the data to be rendered.

salt.renderers.hjson

hjson renderer for Salt

See the hjson documentation for more information

`salt.renderers.hjson.`**`render`**(*hjson_data*, *saltenv='base'*, *sls=''*, ***kws*)

Accepts HJSON as a string or as a file object and runs it through the HJSON parser.

Return type A Python data structure

salt.renderers.jinja

Jinja loading utils to enable a more powerful backend for jinja templates

For Jinja usage information see *Understanding Jinja*.

`salt.renderers.jinja.`**`render`**(*template_file*, *saltenv='base'*, *sls=''*, *argline=''*, *context=None*, *tmplpath=None*, ***kws*)

Render the template_file, passing the functions and grains into the Jinja rendering system.

Return type string

`class` `salt.utils.jinja.`**`SerializerExtension`**(*environment*)

Yaml and Json manipulation.

Format filters

Allows jsonifying or yamlifying any data structure. For example, this dataset:

```
data = {
    'foo': True,
    'bar': 42,
    'baz': [1, 2, 3],
    'qux': 2.0
}
```

```
yaml = {{ data|yaml }}
json = {{ data|json }}
python = {{ data|python }}
xml  = {{ {'root_node': data}|xml }}
```

will be rendered as:

```
yaml = {bar: 42, baz: [1, 2, 3], foo: true, qux: 2.0}
json = {"baz": [1, 2, 3], "foo": true, "bar": 42, "qux": 2.0}
python = {'bar': 42, 'baz': [1, 2, 3], 'foo': True, 'qux': 2.0}
xml = """<<?xml version="1.0" ?>
         <root_node bar="42" foo="True" qux="2.0">
          <baz>1</baz>
          <baz>2</baz>
          <baz>3</baz>
         </root_node>"""
```

The yaml filter takes an optional flow_style parameter to control the default-flow-style parameter of the YAML dumper.

```
{{ data|yaml(False) }}
```

will be rendered as:

```
bar: 42
baz:
  - 1
  - 2
  - 3
foo: true
qux: 2.0
```

Load filters

Strings and variables can be deserialized with **load_yaml** and **load_json** tags and filters. It allows one to manipulate data directly in templates, easily:

```
{%- set yaml_src = "{foo: it works}"|load_yaml %}
{%- set json_src = "{'bar': 'for real'}"|load_json %}
Dude, {{ yaml_src.foo }} {{ json_src.bar }}!
```

will be rendered as:

```
Dude, it works for real!
```

Load tags

Salt implements `load_yaml` and `load_json` tags. They work like the import tag, except that the document is also deserialized.

Syntaxes are `{% load_yaml as [VARIABLE] %}[YOUR DATA]{% endload %}` and `{% load_json as [VARIABLE] %}[YOUR DATA]{% endload %}`

For example:

```
{% load_yaml as yaml_src %}
    foo: it works
{% endload %}
{% load_json as json_src %}
    {
        "bar": "for real"
    }
{% endload %}
Dude, {{ yaml_src.foo }} {{ json_src.bar }}!
```

will be rendered as:

```
Dude, it works for real!
```

Import tags

External files can be imported and made available as a Jinja variable.

```
{% import_yaml "myfile.yml" as myfile %}
{% import_json "defaults.json" as defaults %}
{% import_text "completeworksofshakespeare.txt" as poems %}
```

Catalog

`import_*` and `load_*` tags will automatically expose their target variable to import. This feature makes catalog of data to handle.

for example:

```
# doc1.sls
{% load_yaml as var1 %}
    foo: it works
{% endload %}
{% load_yaml as var2 %}
    bar: for real
{% endload %}
```

```
# doc2.sls
{% from "doc1.sls" import var1, var2 as local2 %}
{{ var1.foo }} {{ local2.bar }}
```

** Escape Filters **

New in version 2017.7.0.

Allows escaping of strings so they can be interpreted literally by another function.

For example:

```
regex_escape = {{ 'https://example.com?foo=bar%20baz' | regex_escape }}
```

will be rendered as:

```
regex_escape = https\:\/\/example\.com\?foo\=bar\%20baz
```

** Set Theory Filters **

New in version 2017.7.0.

Performs set math using Jinja filters.

For example:

```
unique = {{ ['foo', 'foo', 'bar'] | unique }}
```

will be rendered as:

```
unique = ['foo', 'bar']
```

salt.renderers.json

JSON Renderer for Salt

`salt.renderers.json.`**`render`**(*json_data*, *saltenv='base'*, *sls=''*, ***kws*)
Accepts JSON as a string or as a file object and runs it through the JSON parser.

Return type A Python data structure

salt.renderers.json5

JSON5 Renderer for Salt

New in version 2016.3.0.

JSON5 is an unofficial extension to JSON. See http://json5.org/ for more information.

This renderer requires the json5 python bindings, installable via pip.

`salt.renderers.json5.`**`render`**(*json_data*, *saltenv='base'*, *sls=''*, ***kws*)
Accepts JSON as a string or as a file object and runs it through the JSON parser.

Return type A Python data structure

salt.renderers.mako

Mako Renderer for Salt

`salt.renderers.mako.`**`render`**(*template_file*, *saltenv='base'*, *sls=''*, *context=None*, *tmplpath=None*, ***kws*)
Render the template_file, passing the functions and grains into the Mako rendering system.

Return type string

salt.renderers.msgpack

`salt.renderers.msgpack.`**`render`**(*msgpack_data*, *saltenv='base'*, *sls=''*, ***kws*)
Accepts a message pack string or a file object, renders said data back to a python dict.

Return type A Python data structure

salt.renderers.pass module

Pass Renderer for Salt

pass is an encrypted on-disk password store.

New in version 2017.7.0.

Setup

Note: `<user>` needs to be replaced with the user salt-master will be running as.

Have private gpg loaded into `user`'s gpg keyring

```
load_private_gpg_key:
  cmd.run:
    - name: gpg --import <location_of_private_gpg_key>
    - unless: gpg --list-keys '<gpg_name>'
```

Said private key's public key should have been used when encrypting pass entries that are of interest for pillar data.

Fetch and keep local pass git repo up-to-date

```
update_pass:
  git.latest:
    - force_reset: True
    - name: <git_repo>
    - target: /<user>/.password-store
    - identity: <location_of_ssh_private_key>
    - require:
      - cmd: load_private_gpg_key
```

Install pass binary

```
pass:
  pkg.installed
```

`salt.renderers.pass.`**`render`**(*pass_info*, *saltenv='base'*, *sls=''*, *argline=''*, ***kwargs*)
Fetch secret from pass based on pass_path

salt.renderers.py

Pure python state renderer

To use this renderer, the SLS file should contain a function called `run` which returns highstate data.

The highstate data is a dictionary containing identifiers as keys, and execution dictionaries as values. For example the following state declaration in YAML:

```
common_packages:
  pkg.installed:
   - pkgs:
      - curl
      - vim
```

tranlastes to:

```
{'common_packages': {'pkg.installed': [{'pkgs': ['curl', 'vim']}]}}
```

In this module, a few objects are defined for you, giving access to Salt's execution functions, grains, pillar, etc. They are:

- `__salt__` - *Execution functions* (i.e. `__salt__['test.echo']('foo')`)
- `__grains__` - *Grains* (i.e. `__grains__['os']`)
- `__pillar__` - *Pillar data* (i.e. `__pillar__['foo']`)
- `__opts__` - Minion configuration options

- `__env__` - The effective salt fileserver environment (i.e. `base`). Also referred to as a "saltenv". `__env__` should not be modified in a pure python SLS file. To use a different environment, the environment should be set when executing the state. This can be done in a couple different ways:
  - Using the `saltenv` argument on the salt CLI (i.e. `salt '*' state.sls foo.bar.baz saltenv=env_name`).
  - By adding a `saltenv` argument to an individual state within the SLS file. In other words, adding a line like this to the state's data structure: `{'saltenv': 'env_name'}`
- `__sls__` - The SLS path of the file. For example, if the root of the base environment is `/srv/salt`, and the SLS file is `/srv/salt/foo/bar/baz.sls`, then `__sls__` in that file will be `foo.bar.baz`.

When writing a reactor SLS file the global context `data` (same as context `{{ data }}` for states written with Jinja + YAML) is available. The following YAML + Jinja state declaration:

```
{% if data['id'] == 'mysql1' %}
highstate_run:
  local.state.apply:
    - tgt: mysql1
{% endif %}
```

translates to:

```
if data['id'] == 'mysql1':
    return {'highstate_run': {'local.state.apply': [{'tgt': 'mysql1'}]}}
```

Full Example

```
#!py

def run():
    config = {}

    if __grains__['os'] == 'Ubuntu':
        user = 'ubuntu'
        group = 'ubuntu'
        home = '/home/{0}'.format(user)
    else:
        user = 'root'
        group = 'root'
        home = '/root/'

    config['s3cmd'] = {
        'pkg': [
            'installed',
            {'name': 's3cmd'},
        ],
    }

    config[home + '/.s3cfg'] = {
        'file.managed': [
            {'source': 'salt://s3cfg/templates/s3cfg'},
            {'template': 'jinja'},
            {'user': user},
            {'group': group},
            {'mode': 600},
```

(continues on next page)

(continued from previous page)

```
                {'context': {
                    'aws_key': __pillar__['AWS_ACCESS_KEY_ID'],
                    'aws_secret_key': __pillar__['AWS_SECRET_ACCESS_KEY'],
                    },
                },
            ],
        }

        return config
```

`salt.renderers.py.`**`render`**(*template*, *saltenv='base'*, *sls="*, *tmplpath=None*, ***kws*)

Render the python module's components

Return type string

salt.renderers.pydsl

A Python-based DSL

maintainer Jack Kuan <kjkuan@gmail.com>

maturity new

platform all

The *pydsl* renderer allows one to author salt formulas (.sls files) in pure Python using a DSL that's easy to write and easy to read. Here's an example:

```
#!pydsl

apache = state('apache')
apache.pkg.installed()
apache.service.running()
state('/var/www/index.html') \
    .file('managed',
          source='salt://webserver/index.html') \
    .require(pkg='apache')
```

Notice that any Python code is allow in the file as it's really a Python module, so you have the full power of Python at your disposal. In this module, a few objects are defined for you, including the usual (with `__` added) `__salt__` dictionary, `__grains__`, `__pillar__`, `__opts__`, `__env__`, and `__sls__`, plus a few more:

`__file__`

local file system path to the sls module.

`__pydsl__`

Salt PyDSL object, useful for configuring DSL behavior per sls rendering.

`include`

Salt PyDSL function for creating *Include declaration*'s.

`extend`

Salt PyDSL function for creating *Extend declaration*'s.

`state`

Salt PyDSL function for creating *ID declaration*'s.

A state *ID declaration* is created with a `state(id)` function call. Subsequent `state(id)` call with the same id returns the same object. This singleton access pattern applies to all declaration objects created with the DSL.

```
state('example')
assert state('example') is state('example')
assert state('example').cmd is state('example').cmd
assert state('example').cmd.running is state('example').cmd.running
```

The *id* argument is optional. If omitted, an UUID will be generated and used as the *id*.

`state(id)` returns an object under which you can create a *State declaration* object by accessing an attribute named after *any* state module available in Salt.

```
state('example').cmd
state('example').file
state('example').pkg
...
```

Then, a *Function declaration* object can be created from a *State declaration* object by one of the following two ways:

1. by calling a method named after the state function on the *State declaration* object.

```
state('example').file.managed(...)
```

2. by directly calling the attribute named for the *State declaration*, and supplying the state function name as the first argument.

```
state('example').file('managed', ...)
```

With either way of creating a *Function declaration* object, any *Function arg declaration*'s can be passed as keyword arguments to the call. Subsequent calls of a *Function declaration* will update the arg declarations.

```
state('example').file('managed', source='salt://webserver/index.html')
state('example').file.managed(source='salt://webserver/index.html')
```

As a shortcut, the special *name* argument can also be passed as the first or second positional argument depending on the first or second way of calling the *State declaration* object. In the following two examples *ls -la* is the *name* argument.

```
state('example').cmd.run('ls -la', cwd='/')
state('example').cmd('run', 'ls -la', cwd='/')
```

Finally, a *Requisite declaration* object with its *Requisite reference*'s can be created by invoking one of the requisite methods (see *State Requisites*) on either a *Function declaration* object or a *State declaration* object. The return value of a requisite call is also a *Function declaration* object, so you can chain several requisite calls together.

Arguments to a requisite call can be a list of *State declaration* objects and/or a set of keyword arguments whose names are state modules and values are IDs of *ID declaration*'s or names of *Name declaration*'s.

```
apache2 = state('apache2')
apache2.pkg.installed()
state('libapache2-mod-wsgi').pkg.installed()

# you can call requisites on function declaration
apache2.service.running() \
               .require(apache2.pkg,
                        pkg='libapache2-mod-wsgi') \
               .watch(file='/etc/apache2/httpd.conf')
```

(continues on next page)

(continued from previous page)

```
# or you can call requisites on state declaration.
# this actually creates an anonymous function declaration object
# to add the requisites.
apache2.service.require(state('libapache2-mod-wsgi').pkg,
                        pkg='apache2') \
               .watch(file='/etc/apache2/httpd.conf')

# we still need to set the name of the function declaration.
apache2.service.running()
```

Include declaration objects can be created with the `include` function, while *Extend declaration* objects can be created with the `extend` function, whose arguments are just *Function declaration* objects.

```
include('edit.vim', 'http.server')
extend(state('apache2').service.watch(file='/etc/httpd/httpd.conf')
```

The `include` function, by default, causes the included sls file to be rendered as soon as the `include` function is called. It returns a list of rendered module objects; sls files not rendered with the pydsl renderer return `None`'s. This behavior creates no *Include declaration*'s in the resulting high state data structure.

```
import types

# including multiple sls returns a list.
_, mod = include('a-non-pydsl-sls', 'a-pydsl-sls')

assert _ is None
assert isinstance(slsmods[1], types.ModuleType)

# including a single sls returns a single object
mod = include('a-pydsl-sls')

# myfunc is a function that calls state(...) to create more states.
mod.myfunc(1, 2, "three")
```

Notice how you can define a reusable function in your pydsl sls module and then call it via the module returned by `include`.

It's still possible to do late includes by passing the `delayed=True` keyword argument to `include`.

```
include('edit.vim', 'http.server', delayed=True)
```

Above will just create a *Include declaration* in the rendered result, and such call always returns `None`.

Special integration with the *cmd* state

Taking advantage of rendering a Python module, PyDSL allows you to declare a state that calls a pre-defined Python function when the state is executed.

```
greeting = "hello world"
def helper(something, *args, **kws):
    print greeting                # hello world
    print something, args, kws    # test123 ['a', 'b', 'c'] {'x': 1, 'y': 2}

state().cmd.call(helper, "test123", 'a', 'b', 'c', x=1, y=2)
```

The *cmd.call* state function takes care of calling our `helper` function with the arguments we specified in the states, and translates the return value of our function into a structure expected by the state system. See *`salt.states.cmd.call()`* for more information.

Implicit ordering of states

Salt states are explicitly ordered via *Requisite declaration*'s. However, with *pydsl* it's possible to let the renderer track the order of creation for *Function declaration* objects, and implicitly add `require` requisites for your states to enforce the ordering. This feature is enabled by setting the `ordered` option on `__pydsl__`.

Note: this feature is only available if your minions are using Python >= 2.7.

```
include('some.sls.file')

A = state('A').cmd.run(cwd='/var/tmp')
extend(A)

__pydsl__.set(ordered=True)

for i in range(10):
    i = six.text_type(i)
    state(i).cmd.run('echo '+i, cwd='/')
state('1').cmd.run('echo one')
state('2').cmd.run(name='echo two')
```

Notice that the `ordered` option needs to be set after any `extend` calls. This is to prevent *pydsl* from tracking the creation of a state function that's passed to an `extend` call.

Above example should create states from `0` to `9` that will output `0`, `one`, `two`, `3`, … `9`, in that order.

It's important to know that *pydsl* tracks the *creations* of *Function declaration* objects, and automatically adds a `require` requisite to a *Function declaration* object that requires the last *Function declaration* object created before it in the sls file.

This means later calls (perhaps to update the function's *Function arg declaration*) to a previously created function declaration will not change the order.

Render time state execution

When Salt processes a salt formula file, the file is rendered to salt's high state data representation by a renderer before the states can be executed. In the case of the *pydsl* renderer, the .sls file is executed as a python module as it is being rendered which makes it easy to execute a state at render time. In *pydsl*, executing one or more states at render time can be done by calling a configured *ID declaration* object.

```
#!pydsl

s = state() # save for later invocation

# configure it
s.cmd.run('echo at render time', cwd='/')
s.file.managed('target.txt', source='salt://source.txt')

s() # execute the two states now
```

Once an *ID declaration* is called at render time it is detached from the sls module as if it was never defined.

Note: If *implicit ordering* is enabled (i.e., via `__pydsl__.set(ordered=True)`) then the *first* invocation of a *ID declaration* object must be done before a new *Function declaration* is created.

Integration with the stateconf renderer

The `salt.renderers.stateconf` renderer offers a few interesting features that can be leveraged by the *pydsl* renderer. In particular, when using with the *pydsl* renderer, we are interested in *stateconf*'s sls namespacing feature (via dot-prefixed id declarations), as well as, the automatic *start* and *goal* states generation.

Now you can use *pydsl* with *stateconf* like this:

```
#!pydsl|stateconf -ps

include('xxx', 'yyy')

# ensure that states in xxx run BEFORE states in this file.
extend(state('.start').stateconf.require(stateconf='xxx::goal'))

# ensure that states in yyy run AFTER states in this file.
extend(state('.goal').stateconf.require_in(stateconf='yyy::start'))

__pydsl__.set(ordered=True)

...
```

`-s` enables the generation of a stateconf *start* state, and `-p` lets us pipe high state data rendered by *pydsl* to *stateconf*. This example shows that by `require`-ing or `require_in`-ing the included sls' *start* or *goal* states, it's possible to ensure that the included sls files can be made to execute before or after a state in the including sls file.

Importing custom Python modules

To use a custom Python module inside a PyDSL state, place the module somewhere that it can be loaded by the Salt loader, such as *_modules* in the */srv/salt* directory.

Then, copy it to any minions as necessary by using *saltutil.sync_modules*.

To import into a PyDSL SLS, one must bypass the Python importer and insert it manually by getting a reference from Python's *sys.modules* dictionary.

For example:

```
#!pydsl|stateconf -ps

def main():
    my_mod = sys.modules['salt.loaded.ext.module.my_mod']
```

`salt.renderers.pydsl.`**`render`**(*template*, *saltenv='base'*, *sls="*, *tmplpath=None*, *rendered_sls=None*, ***kws*)

salt.renderers.pyobjects

Python renderer that includes a Pythonic Object based interface

maintainer Evan Borgstrom <evan@borgstrom.ca>

Let's take a look at how you use pyobjects in a state file. Here's a quick example that ensures the `/tmp` directory is in the correct state.

```
#!pyobjects

File.managed("/tmp", user='root', group='root', mode='1777')
```

Nice and Pythonic!

By using the "shebang" syntax to switch to the pyobjects renderer we can now write our state data using an object based interface that should feel at home to python developers. You can import any module and do anything that you'd like (with caution, importing sqlalchemy, django or other large frameworks has not been tested yet). Using the pyobjects renderer is exactly the same as using the built-in Python renderer with the exception that pyobjects provides you with an object based interface for generating state data.

Creating state data

Pyobjects takes care of creating an object for each of the available states on the minion. Each state is represented by an object that is the CamelCase version of its name (i.e. `File`, `Service`, `User`, etc), and these objects expose all of their available state functions (i.e. `File.managed`, `Service.running`, etc).

The name of the state is split based upon underscores (_), then each part is capitalized and finally the parts are joined back together.

Some examples:

- `postgres_user` becomes `PostgresUser`
- `ssh_known_hosts` becomes `SshKnownHosts`

Context Managers and requisites

How about something a little more complex. Here we're going to get into the core of how to use pyobjects to write states.

```
#!pyobjects

with Pkg.installed("nginx"):
    Service.running("nginx", enable=True)

    with Service("nginx", "watch_in"):
        File.managed("/etc/nginx/conf.d/mysite.conf",
                     owner='root', group='root', mode='0444',
                     source='salt://nginx/mysite.conf')
```

The objects that are returned from each of the magic method calls are setup to be used a Python context managers (`with`) and when you use them as such all declarations made within the scope will **automatically** use the enclosing state as a requisite!

The above could have also been written use direct requisite statements as.

```
#!pyobjects

Pkg.installed("nginx")
```

(continues on next page)

(continued from previous page)

```
Service.running("nginx", enable=True, require=Pkg("nginx"))
File.managed("/etc/nginx/conf.d/mysite.conf",
             owner='root', group='root', mode='0444',
             source='salt://nginx/mysite.conf',
             watch_in=Service("nginx"))
```

You can use the direct requisite statement for referencing states that are generated outside of the current file.

```
#!pyobjects

# some-other-package is defined in some other state file
Pkg.installed("nginx", require=Pkg("some-other-package"))
```

The last thing that direct requisites provide is the ability to select which of the SaltStack requisites you want to use (require, require_in, watch, watch_in, use & use_in) when using the requisite as a context manager.

```
#!pyobjects

with Service("my-service", "watch_in"):
    ...
```

The above example would cause all declarations inside the scope of the context manager to automatically have their `watch_in` set to `Service("my-service")`.

Including and Extending

To include other states use the `include()` function. It takes one name per state to include.

To extend another state use the `extend()` function on the name when creating a state.

```
#!pyobjects

include('http', 'ssh')

Service.running(extend('apache'),
                watch=[File('/etc/httpd/extra/httpd-vhosts.conf')])
```

Importing from other state files

Like any Python project that grows you will likely reach a point where you want to create reusability in your state tree and share objects between state files, Map Data (described below) is a perfect example of this.

To facilitate this Python's `import` statement has been augmented to allow for a special case when working with a Salt state tree. If you specify a Salt url (`salt://...`) as the target for importing from then the pyobjects renderer will take care of fetching the file for you, parsing it with all of the pyobjects features available and then place the requested objects in the global scope of the template being rendered.

This works for all types of import statements; `import X`, `from X import Y`, and `from X import Y as Z`.

```
#!pyobjects

import salt://myfile.sls
```

(continues on next page)

(continued from previous page)

```
from salt://something/data.sls import Object
from salt://something/data.sls import Object as Other
```

See the Map Data section for a more practical use.

Caveats:

- Imported objects are ALWAYS put into the global scope of your template, regardless of where your import statement is.

Salt object

In the spirit of the object interface for creating state data pyobjects also provides a simple object interface to the `__salt__` object.

A function named `salt` exists in scope for your sls files and will dispatch its attributes to the `__salt__` dictionary.

The following lines are functionally equivalent:

```
#!pyobjects

ret = salt.cmd.run(bar)
ret = __salt__['cmd.run'](bar)
```

Pillar, grain, mine & config data

Pyobjects provides shortcut functions for calling `pillar.get`, `grains.get`, `mine.get` & `config.get` on the `__salt__` object. This helps maintain the readability of your state files.

Each type of data can be access by a function of the same name: `pillar()`, `grains()`, `mine()` and `config()`.

The following pairs of lines are functionally equivalent:

```
#!pyobjects

value = pillar('foo:bar:baz', 'qux')
value = __salt__['pillar.get']('foo:bar:baz', 'qux')

value = grains('pkg:apache')
value = __salt__['grains.get']('pkg:apache')

value = mine('os:Fedora', 'network.interfaces', 'grain')
value = __salt__['mine.get']('os:Fedora', 'network.interfaces', 'grain')

value = config('foo:bar:baz', 'qux')
value = __salt__['config.get']('foo:bar:baz', 'qux')
```

Map Data

When building complex states or formulas you often need a way of building up a map of data based on grain data. The most common use of this is tracking the package and service name differences between distributions.

To build map data using pyobjects we provide a class named Map that you use to build your own classes with inner classes for each set of values for the different grain matches.

```
#!pyobjects

class Samba(Map):
    merge = 'samba:lookup'
    # NOTE: priority is new to 2017.7.0
    priority = ('os_family', 'os')

    class Ubuntu:
        __grain__ = 'os'
        service = 'smbd'

    class Debian:
        server = 'samba'
        client = 'samba-client'
        service = 'samba'

    class RHEL:
        __match__ = 'RedHat'
        server = 'samba'
        client = 'samba'
        service = 'smb'
```

Note: By default, the `os_family` grain will be used as the target for matching. This can be overridden by specifying a `__grain__` attribute.

If a `__match__` attribute is defined for a given class, then that value will be matched against the targeted grain, otherwise the class name's value will be be matched.

Given the above example, the following is true:

1. Minions with an `os_family` of **Debian** will be assigned the attributes defined in the **Debian** class.
2. Minions with an `os` grain of **Ubuntu** will be assigned the attributes defined in the **Ubuntu** class.
3. Minions with an `os_family` grain of **RedHat** will be assigned the attributes defined in the **RHEL** class.

That said, sometimes a minion may match more than one class. For instance, in the above example, Ubuntu minions will match both the **Debian** and **Ubuntu** classes, since Ubuntu has an `os_family` grain of **Debian** and an `os` grain of **Ubuntu**. As of the 2017.7.0 release, the order is dictated by the order of declaration, with classes defined later overriding earlier ones. Additionally, 2017.7.0 adds support for explicitly defining the ordering using an optional attribute called `priority`.

Given the above example, `os_family` matches will be processed first, with `os` matches processed after. This would have the effect of assigning `smbd` as the `service` attribute on Ubuntu minions. If the `priority` item was not defined, or if the order of the items in the `priority` tuple were reversed, Ubuntu minions would have a `service` attribute of `samba`, since `os_family` matches would have been processed second.

To use this new data you can import it into your state file and then access your attributes. To access the data in the map you simply access the attribute name on the base class that is extending Map. Assuming the above Map was in the file `samba/map.sls`, you could do the following.

```
#!pyobjects

from salt://samba/map.sls import Samba

with Pkg.installed("samba", names=[Samba.server, Samba.client]):
    Service.running("samba", name=Samba.service)
```

class `salt.renderers.pyobjects.`**`PyobjectsModule`**(*name*, *attrs*)

This provides a wrapper for bare imports.

`salt.renderers.pyobjects.`**`load_states`**()

This loads our states into the salt __context__

`salt.renderers.pyobjects.`**`render`**(*template*, *saltenv='base'*, *sls=''*, *salt_data=True*, ***kwargs*)

salt.renderers.stateconf

maintainer Jack Kuan <kjkuan@gmail.com>

maturity new

platform all

This module provides a custom renderer that processes a salt file with a specified templating engine (e.g. Jinja) and a chosen data renderer (e.g. YAML), extracts arguments for any `stateconf.set` state, and provides the extracted arguments (including Salt-specific args, such as `require`, etc) as template context. The goal is to make writing reusable/configurable/parameterized salt files easier and cleaner.

To use this renderer, either set it as the default renderer via the `renderer` option in master/minion's config, or use the shebang line in each individual sls file, like so: `#!stateconf`. Note, due to the way this renderer works, it must be specified as the first renderer in a render pipeline. That is, you cannot specify `#!mako|yaml|stateconf`, for example. Instead, you specify them as renderer arguments: `#!stateconf mako . yaml`.

Here's a list of features enabled by this renderer.

- Prefixes any state id (declaration or reference) that starts with a dot (`.`) to avoid duplicated state ids when the salt file is included by other salt files.

  For example, in the *salt://some/file.sls*, a state id such as `.sls_params` will be turned into `some.file::sls_params`. Example:

  ```
  #!stateconf yaml . jinja

  .vim:
    pkg.installed
  ```

  Above will be translated into:

  ```
  some.file::vim:
    pkg.installed:
      - name: vim
  ```

  Notice how that if a state under a dot-prefixed state id has no `name` argument then one will be added automatically by using the state id with the leading dot stripped off.

  The leading dot trick can be used with extending state ids as well, so you can include relatively and extend relatively. For example, when extending a state in *salt://some/other_file.sls*, e.g.:

  ```
  #!stateconf yaml . jinja

  include:
    - .file

  extend:
    .file::sls_params:
      stateconf.set:
        - name1: something
  ```

Above will be pre-processed into:

```
include:
  - some.file

extend:
  some.file::sls_params:
    stateconf.set:
      - name1: something
```

- Adds a `sls_dir` context variable that expands to the directory containing the rendering salt file. So, you can write `salt://{{sls_dir}}/...` to reference templates files used by your salt file.
- Recognizes the special state function, `stateconf.set`, that configures a default list of named arguments usable within the template context of the salt file. Example:

```
#!stateconf yaml . jinja

.sls_params:
  stateconf.set:
    - name1: value1
    - name2: value2
    - name3:
      - value1
      - value2
      - value3
    - require_in:
      - cmd: output

# --- end of state config ---

.output:
  cmd.run:
    - name: |
        echo 'name1={{sls_params.name1}}
              name2={{sls_params.name2}}
              name3[1]={{sls_params.name3[1]}}
        '
```

This even works with `include` + `extend` so that you can override the default configured arguments by including the salt file and then `extend` the `stateconf.set` states that come from the included salt file. (*IMPORTANT: Both the included and the extending sls files must use the stateconf renderer for this ``extend`` to work!*)

Notice that the end of configuration marker (`# --- end of state config --`) is needed to separate the use of 'stateconf.set' form the rest of your salt file. The regex that matches such marker can be configured via the `stateconf_end_marker` option in your master or minion config file.

Sometimes, it is desirable to set a default argument value that's based on earlier arguments in the same `stateconf.set`. For example, it may be tempting to do something like this:

```
#!stateconf yaml . jinja

.apache:
  stateconf.set:
    - host: localhost
    - port: 1234
    - url: 'http://{{host}}:{{port}}/'
```

(continues on next page)

(continued from previous page)

```
# --- end of state config ---

.test:
  cmd.run:
    - name: echo '{{apache.url}}'
    - cwd: /
```

However, this won't work. It can however be worked around like so:

```
#!stateconf yaml . jinja

.apache:
  stateconf.set:
    - host: localhost
    - port: 1234
{#  - url: 'http://{{host}}:{{port}}/' #}

# --- end of state config ---
# {{ apache.setdefault('url', "http://%(host)s:%(port)s/" % apache) }}

.test:
  cmd.run:
    - name: echo '{{apache.url}}'
    - cwd: /
```

- Adds support for relative include and exclude of .sls files. Example:

```
#!stateconf yaml . jinja

include:
  - .apache
  - .db.mysql
  - ..app.django

exclude:
  - sls: .users
```

If the above is written in a salt file at *salt://some/where.sls* then it will include *salt://some/apache.sls*, *salt://some/db/mysql.sls* and *salt://app/django.sls*, and exclude *salt://some/users.ssl*. Actually, it does that by rewriting the above `include` and `exclude` into:

```
include:
  - some.apache
  - some.db.mysql
  - app.django

exclude:
  - sls: some.users
```

- Optionally (enabled by default, *disable* via the *-G* renderer option, e.g. in the shebang line: `#!stateconf -G`), generates a `stateconf.set` goal state (state id named as `.goal` by default, configurable via the master/minion config option, `stateconf_goal_state`) that requires all other states in the salt file. Note, the `.goal` state id is subject to dot-prefix rename rule mentioned earlier.

  Such goal state is intended to be required by some state in an including salt file. For example, in your webapp salt file, if you include a sls file that is supposed to setup Tomcat, you might want to make sure that all states in the Tomcat sls file will be executed before some state in the webapp sls file.

- Optionally (enable via the *-o* renderer option, e.g. in the shebang line: `#!stateconf -o`), orders the states in a sls file by adding a `require` requisite to each state such that every state requires the state defined just before it. The order of the states here is the order they are defined in the sls file. (Note: this feature is only available if your minions are using Python >= 2.7. For Python2.6, it should also work if you install the *ordereddict* module from PyPI)

  By enabling this feature, you are basically agreeing to author your sls files in a way that gives up the explicit (or implicit?) ordering imposed by the use of `require`, `watch`, `require_in` or `watch_in` requisites, and instead, you rely on the order of states you define in the sls files. This may or may not be a better way for you. However, if there are many states defined in a sls file, then it tends to be easier to see the order they will be executed with this feature.

  You are still allowed to use all the requisites, with a few restrictions. You cannot `require` or `watch` a state defined *after* the current state. Similarly, in a state, you cannot `require_in` or `watch_in` a state defined *before* it. Breaking any of the two restrictions above will result in a state loop. The renderer will check for such incorrect uses if this feature is enabled.

  Additionally, `names` declarations cannot be used with this feature because the way they are compiled into low states make it impossible to guarantee the order in which they will be executed. This is also checked by the renderer. As a workaround for not being able to use `names`, you can achieve the same effect, by generate your states with the template engine available within your sls file.

  Finally, with the use of this feature, it becomes possible to easily make an included sls file execute all its states *after* some state (say, with id `X`) in the including sls file. All you have to do is to make state, `X`, `require_in` the first state defined in the included sls file.

When writing sls files with this renderer, one should avoid using what can be defined in a `name` argument of a state as the state's id. That is, avoid writing states like this:

```
/path/to/some/file:
  file.managed:
    - source: salt://some/file

cp /path/to/some/file file2:
  cmd.run:
    - cwd: /
    - require:
      - file: /path/to/some/file
```

Instead, define the state id and the `name` argument separately for each state. Also, the ID should be something meaningful and easy to reference within a requisite (which is a good habit anyway, and such extra indirection would also makes the sls file easier to modify later). Thus, the above states should be written like this:

```
add-some-file:
  file.managed:
    - name: /path/to/some/file
    - source: salt://some/file

copy-files:
  cmd.run:
    - name: cp /path/to/some/file file2
    - cwd: /
    - require:
      - file: add-some-file
```

Moreover, when referencing a state from a requisite, you should reference the state's id plus the state name rather than the state name plus its `name` argument. (Yes, in the above example, you can actually `require` the `file:`

`/path/to/some/file`, instead of the `file: add-some-file`). The reason is that this renderer will re-write or rename state id's and their references for state id's prefixed with `..`. So, if you reference `name` then there's no way to reliably rewrite such reference.

salt.renderers.wempy

`salt.renderers.wempy.`**`render`**(*template_file*, *saltenv='base'*, *sls=''*, *argline=''*, *context=None*, ***kws*)
Render the data passing the functions and grains into the rendering system

Return type string

salt.renderers.yaml

Understanding YAML

The default renderer for SLS files is the YAML renderer. YAML is a markup language with many powerful features. However, Salt uses a small subset of YAML that maps over very commonly used data structures, like lists and dictionaries. It is the job of the YAML renderer to take the YAML data structure and compile it into a Python data structure for use by Salt.

Though YAML syntax may seem daunting and terse at first, there are only three very simple rules to remember when writing YAML for SLS files.

Rule One: Indentation

YAML uses a fixed indentation scheme to represent relationships between data layers. Salt requires that the indentation for each level consists of exactly two spaces. Do not use tabs.

Rule Two: Colons

Python dictionaries are, of course, simply key-value pairs. Users from other languages may recognize this data type as hashes or associative arrays.

Dictionary keys are represented in YAML as strings terminated by a trailing colon. Values are represented by either a string following the colon, separated by a space:

```
my_key: my_value
```

In Python, the above maps to:

```
{'my_key': 'my_value'}
```

Dictionaries can be nested:

```
first_level_dict_key:
  second_level_dict_key: value_in_second_level_dict
```

And in Python:

```
{'first_level_dict_key': {'second_level_dict_key': 'value_in_second_level_dict' } }
```

Rule Three: Dashes

To represent lists of items, a single dash followed by a space is used. Multiple items are a part of the same list as a function of their having the same level of indentation.

```
- list_value_one
- list_value_two
- list_value_three
```

Lists can be the value of a key-value pair. This is quite common in Salt:

```
my_dictionary:
  - list_value_one
  - list_value_two
  - list_value_three
```

Reference

YAML Renderer for Salt

For YAML usage information see *Understanding YAML*.

`salt.renderers.yaml.`**`get_yaml_loader`**(*argline*)
Return the ordered dict yaml loader

`salt.renderers.yaml.`**`render`**(*yaml_data, saltenv='base', sls='', argline='', **kws*)
Accepts YAML as a string or as a file object and runs it through the YAML parser.

Return type A Python data structure

salt.renderers.yamlex

YAMLEX renderer is a replacement of the YAML renderer. It's 100% YAML with a pinch of Salt magic:

- All mappings are automatically OrderedDict
- All strings are automatically str obj
- data aggregation with !aggregation yaml tag, based on the `salt.utils.aggregation` module.
- data aggregation over documents for pillar

Instructed aggregation within the `!aggregation` and the `!reset` tags:

```
#!yamlex
foo: !aggregate first
foo: !aggregate second
bar: !aggregate {first: foo}
bar: !aggregate {second: bar}
baz: !aggregate 42
qux: !aggregate default
!reset qux: !aggregate my custom data
```

is roughly equivalent to

```
foo: [first, second]
bar: {first: foo, second: bar}
baz: [42]
qux: [my custom data]
```

Reference

`salt.renderers.yamlex.`**`render`**(*sls_data*, *saltenv='base'*, *sls="*, ***kws*)

Accepts YAML_EX as a string or as a file object and runs it through the YAML_EX parser.

Return type A Python data structure

CHAPTER

FOUR

USING SALT

This section describes the fundamental components and concepts that you need to understand to use Salt.

4.1 Grains

Salt comes with an interface to derive information about the underlying system. This is called the grains interface, because it presents salt with grains of information. Grains are collected for the operating system, domain name, IP address, kernel, OS type, memory, and many other system properties.

The grains interface is made available to Salt modules and components so that the right salt minion commands are automatically available on the right systems.

Grain data is relatively static, though if system information changes (for example, if network settings are changed), or if a new value is assigned to a custom grain, grain data is refreshed.

Note: Grains resolve to lowercase letters. For example, `FOO`, and `foo` target the same grain.

4.1.1 Listing Grains

Available grains can be listed by using the 'grains.ls' module:

```
salt '*' grains.ls
```

Grains data can be listed by using the 'grains.items' module:

```
salt '*' grains.items
```

4.1.2 Using grains in a state

To use a grain in a state you can access it via *{{ grains['key'] }}*.

4.1.3 Grains in the Minion Config

Grains can also be statically assigned within the minion configuration file. Just add the option *`grains`* and pass options to it:

```
grains:
  roles:
    - webserver
    - memcache
  deployment: datacenter4
  cabinet: 13
  cab_u: 14-15
```

Then status data specific to your servers can be retrieved via Salt, or used inside of the State system for matching. It also makes it possible to target based on specific data about your deployment, as in the example above.

4.1.4 Grains in /etc/salt/grains

If you do not want to place your custom static grains in the minion config file, you can also put them in `/etc/salt/grains` on the minion. They are configured in the same way as in the above example, only without a top-level `grains:` key:

```
roles:
  - webserver
  - memcache
deployment: datacenter4
cabinet: 13
cab_u: 14-15
```

Note: Grains in `/etc/salt/grains` are ignored if you specify the same grains in the minion config.

Note: Grains are static, and since they are not often changed, they will need a grains refresh when they are updated. You can do this by calling: `salt minion saltutil.refresh_modules`

Note: You can equally configure static grains for Proxy Minions. As multiple Proxy Minion processes can run on the same machine, you need to index the files using the Minion ID, under `/etc/salt/proxy.d/<minion ID>/grains`. For example, the grains for the Proxy Minion `router1` can be defined under `/etc/salt/proxy.d/router1/grains`, while the grains for the Proxy Minion `switch7` can be put in `/etc/salt/proxy.d/switch7/grains`.

4.1.5 Matching Grains in the Top File

With correctly configured grains on the Minion, the *top file* used in Pillar or during Highstate can be made very efficient. For example, consider the following configuration:

```
'roles:webserver':
  - match: grain
  - state0

'roles:memcache':
  - match: grain
  - state1
  - state2
```

For this example to work, you would need to have defined the grain `role` for the minions you wish to match.

4.1.6 Writing Grains

The grains are derived by executing all of the "public" functions (i.e. those which do not begin with an underscore) found in the modules located in the Salt's core grains code, followed by those in any custom grains modules. The functions in a grains module must return a Python dictionary, where the dictionary keys are the names of grains, and each key's value is that value for that grain.

Custom grains modules should be placed in a subdirectory named `_grains` located under the *`file_roots`* specified by the master config file. The default path would be `/srv/salt/_grains`. Custom grains modules will be distributed to the minions when *`state.highstate`* is run, or by executing the *`saltutil.sync_grains`* or *`saltutil.sync_all`* functions.

Grains modules are easy to write, and (as noted above) only need to return a dictionary. For example:

```
def yourfunction():
     # initialize a grains dictionary
     grains = {}
     # Some code for logic that sets grains like
     grains['yourcustomgrain'] = True
     grains['anothergrain'] = 'somevalue'
     return grains
```

The name of the function does not matter and will not factor into the grains data at all; only the keys/values returned become part of the grains.

When to Use a Custom Grain

Before adding new grains, consider what the data is and remember that grains should (for the most part) be static data.

If the data is something that is likely to change, consider using *Pillar* or an execution module instead. If it's a simple set of key/value pairs, pillar is a good match. If compiling the information requires that system commands be run, then putting this information in an execution module is likely a better idea.

Good candidates for grains are data that is useful for targeting minions in the *top file* or the Salt CLI. The name and data structure of the grain should be designed to support many platforms, operating systems or applications. Also, keep in mind that Jinja templating in Salt supports referencing pillar data as well as invoking functions from execution modules, so there's no need to place information in grains to make it available to Jinja templates. For example:

```
...
...
{{ salt['module.function_name']('argument_1', 'argument_2') }}
{{ pillar['my_pillar_key'] }}
...
...
```

> **Warning:** Custom grains will not be available in the top file until after the first *highstate*. To make custom grains available on a minion's first highstate, it is recommended to use *this example* to ensure that the custom grains are synced when the minion starts.

Loading Custom Grains

If you have multiple functions specifying grains that are called from a `main` function, be sure to prepend grain function names with an underscore. This prevents Salt from including the loaded grains from the grain functions in the final grain data structure. For example, consider this custom grain file:

```
#!/usr/bin/env python
def _my_custom_grain():
    my_grain = {'foo': 'bar', 'hello': 'world'}
    return my_grain


def main():
    # initialize a grains dictionary
    grains = {}
    grains['my_grains'] = _my_custom_grain()
    return grains
```

The output of this example renders like so:

```
# salt-call --local grains.items
local:
    ----------
    <Snipped for brevity>
    my_grains:
        ----------
        foo:
            bar
        hello:
            world
```

However, if you don't prepend the `my_custom_grain` function with an underscore, the function will be rendered twice by Salt in the items output: once for the `my_custom_grain` call itself, and again when it is called in the `main` function:

```
# salt-call --local grains.items
local:
----------
    <Snipped for brevity>
    foo:
        bar
    <Snipped for brevity>
    hello:
        world
    <Snipped for brevity>
    my_grains:
        ----------
        foo:
            bar
        hello:
            world
```

4.1.7 Precedence

Core grains can be overridden by custom grains. As there are several ways of defining custom grains, there is an order of precedence which should be kept in mind when defining them. The order of evaluation is as follows:

1. Core grains.
2. Custom grains in `/etc/salt/grains`.
3. Custom grains in `/etc/salt/minion`.
4. Custom grain modules in `_grains` directory, synced to minions.

Each successive evaluation overrides the previous ones, so any grains defined by custom grains modules synced to minions that have the same name as a core grain will override that core grain. Similarly, grains from `/etc/salt/minion` override both core grains and custom grain modules, and grains in `_grains` will override *any* grains of the same name.

For custom grains, if the function takes an argument `grains`, then the previously rendered grains will be passed in. Because the rest of the grains could be rendered in any order, the only grains that can be relied upon to be passed in are `core` grains. This was added in the 2019.2.0 release.

4.1.8 Examples of Grains

The core module in the grains package is where the main grains are loaded by the Salt minion and provides the principal example of how to write grains:

https://github.com/saltstack/salt/blob/master/salt/grains/core.py

4.1.9 Syncing Grains

Syncing grains can be done a number of ways. They are automatically synced when `state.highstate` is called, or (as noted above) the grains can be manually synced and reloaded by calling the `saltutil.sync_grains` or `saltutil.sync_all` functions.

Note: When the `grains_cache` is set to False, the grains dictionary is built and stored in memory on the minion. Every time the minion restarts or `saltutil.refresh_grains` is run, the grain dictionary is rebuilt from scratch.

4.2 Storing Static Data in the Pillar

Pillar is an interface for Salt designed to offer global values that can be distributed to minions. Pillar data is managed in a similar way as the Salt State Tree.

Pillar was added to Salt in version 0.9.8

Note: Storing sensitive data

Pillar data is compiled on the master. Additionally, pillar data for a given minion is only accessible by the minion for which it is targeted in the pillar configuration. This makes pillar useful for storing sensitive data specific to a particular minion.

4.2.1 Declaring the Master Pillar

The Salt Master server maintains a `pillar_roots` setup that matches the structure of the `file_roots` used in the Salt file server. Like `file_roots`, the `pillar_roots` option maps environments to directories. The pillar data is then mapped to minions based on matchers in a top file which is laid out in the same way as the state top file. Salt pillars can use the same matcher types as the standard *top file*.

conf_master:*pillar_roots* is configured just like `file_roots`. For example:

```
pillar_roots:
  base:
    - /srv/pillar
```

This example configuration declares that the base environment will be located in the `/srv/pillar` directory. It must not be in a subdirectory of the state tree.

The top file used matches the name of the top file used for States, and has the same structure:

`/srv/pillar/top.sls`

```
base:
  '*':
    - packages
```

In the above top file, it is declared that in the `base` environment, the glob matching all minions will have the pillar data found in the `packages` pillar available to it. Assuming the `pillar_roots` value of `/srv/pillar` taken from above, the `packages` pillar would be located at `/srv/pillar/packages.sls`.

Any number of matchers can be added to the base environment. For example, here is an expanded version of the Pillar top file stated above:

/srv/pillar/top.sls:

```
base:
  '*':
    - packages
  'web*':
    - vim
```

In this expanded top file, minions that match `web*` will have access to the `/srv/pillar/packages.sls` file, as well as the `/srv/pillar/vim.sls` file.

Another example shows how to use other standard top matching types to deliver specific salt pillar data to minions with different properties.

Here is an example using the `grains` matcher to target pillars to minions by their `os` grain:

```
dev:
  'os:Debian':
    - match: grain
    - servers
```

Pillar definitions can also take a keyword argument `ignore_missing`. When the value of `ignore_missing` is `True`, all errors for missing pillar files are ignored. The default value for `ignore_missing` is `False`.

Here is an example using the `ignore_missing` keyword parameter to ignore errors for missing pillar files:

```
base:
  '*':
    - servers
    - systems
    - ignore_missing: True
```

Assuming that the pillar `servers` exists in the fileserver backend and the pillar `systems` doesn't, all pillar data from `servers` pillar is delivered to minions and no error for the missing pillar `systems` is noted under the key `_errors` in the pillar data delivered to minions.

Should the `ignore_missing` keyword parameter have the value `False`, an error for the missing pillar `systems` would produce the value `Specified SLS 'servers' in environment 'base' is not available on the salt master` under the key `_errors` in the pillar data delivered to minions.

`/srv/pillar/packages.sls`

```
{% if grains['os'] == 'RedHat' %}
apache: httpd
git: git
{% elif grains['os'] == 'Debian' %}
apache: apache2
git: git-core
{% endif %}

company: Foo Industries
```

Important: See *Is Targeting using Grain Data Secure?* for important security information.

The above pillar sets two key/value pairs. If a minion is running RedHat, then the `apache` key is set to `httpd` and the `git` key is set to the value of `git`. If the minion is running Debian, those values are changed to `apache2` and `git-core` respectively. All minions that have this pillar targeting to them via a top file will have the key of `company` with a value of `Foo Industries`.

Consequently this data can be used from within modules, renderers, State SLS files, and more via the shared pillar dictionary:

```
apache:
  pkg.installed:
    - name: {{ pillar['apache'] }}
```

```
git:
  pkg.installed:
    - name: {{ pillar['git'] }}
```

Finally, the above states can utilize the values provided to them via Pillar. All pillar values targeted to a minion are available via the 'pillar' dictionary. As seen in the above example, Jinja substitution can then be utilized to access the keys and values in the Pillar dictionary.

Note that you cannot just list key/value-information in `top.sls`. Instead, target a minion to a pillar file and then list the keys and values in the pillar. Here is an example top file that illustrates this point:

```
base:
  '*':
     - common_pillar
```

And the actual pillar file at '/srv/pillar/common_pillar.sls':

```
foo: bar
boo: baz
```

Note: When working with multiple pillar environments, assuming that each pillar environment has its own top file, the jinja placeholder `{{ saltenv }}` can be used in place of the environment name:

```
{{ saltenv }}:
  '*':
     - common_pillar
```

Yes, this is `{{ saltenv }}`, and not `{{ pillarenv }}`. The reason for this is because the Pillar top files are parsed using some of the same code which parses top files when *running states*, so the pillar environment takes the place of `{{ saltenv }}` in the jinja context.

4.2.2 Dynamic Pillar Environments

If environment `__env__` is specified in *`pillar_roots`*, all environments that are not explicitly specified in *`pillar_roots`* will map to the directories from `__env__`. This allows one to use dynamic git branch based environments for state/pillar files with the same file-based pillar applying to all environments. For example:

```
pillar_roots:
  __env__:
    - /srv/pillar

ext_pillar:
  - git:
    - __env__ https://example.com/git-pillar.git
```

New in version 2017.7.5,2018.3.1.

4.2.3 Pillar Namespace Flattening

The separate pillar SLS files all merge down into a single dictionary of key-value pairs. When the same key is defined in multiple SLS files, this can result in unexpected behavior if care is not taken to how the pillar SLS files are laid out.

For example, given a `top.sls` containing the following:

```
base:
  '*':
    - packages
    - services
```

with `packages.sls` containing:

```
bind: bind9
```

and `services.sls` containing:

```
bind: named
```

Then a request for the `bind` pillar key will only return `named`. The `bind9` value will be lost, because `services.sls` was evaluated later.

Note: Pillar files are applied in the order they are listed in the top file. Therefore conflicting keys will be overwritten in a 'last one wins' manner! For example, in the above scenario conflicting key values in `services` will overwrite those in `packages` because it's at the bottom of the list.

It can be better to structure your pillar files with more hierarchy. For example the `package.sls` file could be configured like so:

```
packages:
  bind: bind9
```

This would make the `packages` pillar key a nested dictionary containing a `bind` key.

4.2.4 Pillar Dictionary Merging

If the same pillar key is defined in multiple pillar SLS files, and the keys in both files refer to nested dictionaries, then the content from these dictionaries will be recursively merged.

For example, keeping the `top.sls` the same, assume the following modifications to the pillar SLS files:

`packages.sls`:

```
bind:
  package-name: bind9
  version: 9.9.5
```

`services.sls`:

```
bind:
  port: 53
  listen-on: any
```

The resulting pillar dictionary will be:

```
$ salt-call pillar.get bind
local:
    ----------
    listen-on:
        any
    package-name:
        bind9
    port:
        53
    version:
        9.9.5
```

Since both pillar SLS files contained a `bind` key which contained a nested dictionary, the pillar dictionary's `bind` key contains the combined contents of both SLS files' `bind` keys.

4.2.5 Including Other Pillars

New in version 0.16.0.

Pillar SLS files may include other pillar files, similar to State files. Two syntaxes are available for this purpose. The simple form simply includes the additional pillar as if it were part of the same file:

```
include:
  - users
```

The full include form allows two additional options – passing default values to the templating engine for the included pillar file as well as an optional key under which to nest the results of the included pillar:

```
include:
  - users:
      defaults:
          sudo: ['bob', 'paul']
      key: users
```

With this form, the included file (users.sls) will be nested within the 'users' key of the compiled pillar. Additionally, the 'sudo' value will be available as a template variable to users.sls.

4.2.6 In-Memory Pillar Data vs. On-Demand Pillar Data

Since compiling pillar data is computationally expensive, the minion will maintain a copy of the pillar data in memory to avoid needing to ask the master to recompile and send it a copy of the pillar data each time pillar data is requested. This in-memory pillar data is what is returned by the *pillar.item*, *pillar.get*, and *pillar.raw* functions.

Also, for those writing custom execution modules, or contributing to Salt's existing execution modules, the in-memory pillar data is available as the `__pillar__` dunder dictionary.

The in-memory pillar data is generated on minion start, and can be refreshed using the *saltutil.refresh_pillar* function:

```
salt '*' saltutil.refresh_pillar
```

This function triggers the minion to asynchronously refresh the in-memory pillar data and will always return `None`.

In contrast to in-memory pillar data, certain actions trigger pillar data to be compiled to ensure that the most up-to-date pillar data is available. These actions include:

- Running states
- Running *pillar.items*

Performing these actions will *not* refresh the in-memory pillar data. So, if pillar data is modified, and then states are run, the states will see the updated pillar data, but *pillar.item*, *pillar.get*, and *pillar.raw* will not see this data unless refreshed using *saltutil.refresh_pillar*.

4.2.7 How Pillar Environments Are Handled

When multiple pillar environments are used, the default behavior is for the pillar data from all environments to be merged together. The pillar dictionary will therefore contain keys from all configured environments.

The *pillarenv* minion config option can be used to force the minion to only consider pillar configuration from a single environment. This can be useful in cases where one needs to run states with alternate pillar data, either in a testing/QA environment or to test changes to the pillar data before pushing them live.

For example, assume that the following is set in the minion config file:

```
pillarenv: base
```

This would cause that minion to ignore all other pillar environments besides `base` when compiling the in-memory pillar data. Then, when running states, the `pillarenv` CLI argument can be used to override the minion's *pillarenv* config value:

```
salt '*' state.apply mystates pillarenv=testing
```

The above command will run the states with pillar data sourced exclusively from the `testing` environment, without modifying the in-memory pillar data.

Note: When running states, the `pillarenv` CLI option does not require a *`pillarenv`* option to be set in the minion config file. When *`pillarenv`* is left unset, as mentioned above all configured environments will be combined. Running states with `pillarenv=testing` in this case would still restrict the states' pillar data to just that of the `testing` pillar environment.

Starting in the 2017.7.0 release, it is possible to pin the pillarenv to the effective saltenv, using the *`pillarenv_from_saltenv`* minion config option. When this is set to `True`, if a specific saltenv is specified when running states, the `pillarenv` will be the same. This essentially makes the following two commands equivalent:

```
salt '*' state.apply mystates saltenv=dev
salt '*' state.apply mystates saltenv=dev pillarenv=dev
```

However, if a pillarenv is specified, it will override this behavior. So, the following command will use the `qa` pillar environment but source the SLS files from the `dev` saltenv:

```
salt '*' state.apply mystates saltenv=dev pillarenv=qa
```

So, if a `pillarenv` is set in the minion config file, *`pillarenv_from_saltenv`* will be ignored, and passing a `pillarenv` on the CLI will temporarily override *`pillarenv_from_saltenv`*.

4.2.8 Viewing Pillar Data

To view pillar data, use the *`pillar`* execution module. This module includes several functions, each of them with their own use. These functions include:

- *`pillar.item`* - Retrieves the value of one or more keys from the *in-memory pillar data*.
- *`pillar.items`* - Compiles a fresh pillar dictionary and returns it, leaving the *in-memory pillar data* untouched. If pillar keys are passed to this function however, this function acts like *`pillar.item`* and returns their values from the *in-memory pillar data*.
- *`pillar.raw`* - Like *`pillar.items`*, it returns the entire pillar dictionary, but from the *in-memory pillar data* instead of compiling fresh pillar data.
- *`pillar.get`* - Described in detail below.

4.2.9 The `pillar.get` Function

New in version 0.14.0.

The *`pillar.get`* function works much in the same way as the `get` method in a python dict, but with an enhancement: nested dictonaries can be traversed using a colon as a delimiter.

If a structure like this is in pillar:

```
foo:
  bar:
    baz: qux
```

Extracting it from the raw pillar in an sls formula or file template is done this way:

```
{{ pillar['foo']['bar']['baz'] }}
```

Now, with the new *pillar.get* function the data can be safely gathered and a default can be set, allowing the template to fall back if the value is not available:

```
{{ salt['pillar.get']('foo:bar:baz', 'qux') }}
```

This makes handling nested structures much easier.

Note: `pillar.get()` vs `salt['pillar.get']()`

It should be noted that within templating, the `pillar` variable is just a dictionary. This means that calling `pillar.get()` inside of a template will just use the default dictionary `.get()` function which does not include the extra `:` delimiter functionality. It must be called using the above syntax (`salt['pillar.get']('foo:bar:baz', 'qux')`) to get the salt function, instead of the default dictionary behavior.

4.2.10 Setting Pillar Data at the Command Line

Pillar data can be set at the command line like the following example:

```
salt '*' state.apply pillar='{"cheese": "spam"}'
```

This will add a pillar key of `cheese` with its value set to `spam`.

Note: Be aware that when sending sensitive data via pillar on the command-line that the publication containing that data will be received by all minions and will not be restricted to the targeted minions. This may represent a security concern in some cases.

4.2.11 Pillar Encryption

Salt's renderer system can be used to decrypt pillar data. This allows for pillar items to be stored in an encrypted state, and decrypted during pillar compilation.

Encrypted Pillar SLS

New in version 2017.7.0.

Consider the following pillar SLS file:

```
secrets:
  vault:
    foo: |
      -----BEGIN PGP MESSAGE-----

      hQEMAw2B674HRhwSAQgAhTrN8NizwUv/VunVrqa4/X8t6EUulrnhKcSeb8sZS4th
      W1Qz3K2NjL4lkUHCQHKZVx/VoZY7zsddBIFvvoGGfj8+2wjkEDwFmFjGE4DEsS74
      ZLRFIFJC1iB/O0AiQ+oU745skQkU6OEKxqavmKMrKo3rvJ8ZCXDC470+i2/Hqrp7
      +KWGmaDOO422JaSKRm5D9bQZr9oX7KqnrPG9I1+UbJyQSJdsdtquPWmeIpamEVHb
      VMDNQRjSezZ1yKC4kCWm3YQbBF76qTHzG1VlLF5qOzuGI9VkyvlMaLfMibriqY73
      zBbPzf6Bkp2+Y9qyzuveYMmwS4sEOuZL/PetqisWe9JGAWD/O+slQ2KRu9hNww06
```

(continues on next page)

(continued from previous page)

```
      KMDPJRdyj5bRuBVE4hHkkP23KrYr7SuhW2vpe7O/MvWEJ9uDNegpMLhTWruGngJh
      iFndxegN9w==
      =bAuo
      -----END PGP MESSAGE-----
    bar: this was unencrypted already
    baz: |
      -----BEGIN PGP MESSAGE-----

      hQEMAw2B674HRhwSAQf+Ne+IfsP2IcPDrUWct8sTJrga47jQvlPCmO+7zJjOVcqz
      gLjUKvMajrbI/jorBWxyAbF+5E7WdG9WHHVnuoywsyTB9rbmzuPqYCJCe+ZVyqWf
      9qgJ+oUjcvYIFmH3h7H68ldqbxaAUkAOQbTRHdr253wwaTIC91ZeX0SCj64HfTg7
      Izwk383CRWonEktXJpientApQFSUWNeLUWagEr/YPNFA3vzpPF5/Ia9X8/z/6oO2
      q+D5W5mVsns3i2HHbg2A8Y+pm4TWnH6mTSh/gdxPqssi9qIrzGQ6H1tEoFFOEq1V
      kJBe0izlfudqMq62XswzuRB4CYT5Iqw1c97T+1RqENJCASG0Wz8AGhinTdlU5iQl
      JkLKqBxcBz4L70LYWyHhYwYROJWjHgKAywX5T67ftq0wi8APuZl9olnOkwSK+wrY
      1OZi
      =7epf
      -----END PGP MESSAGE-----
    qux:
      - foo
      - bar
      - |
        -----BEGIN PGP MESSAGE-----

        hQEMAw2B674HRhwSAQgAg1YCmokrweoOI1c9HO0BLamWBaFPTMblOaTo0WJLZoTS
        ksbQ3OJAMkrkn3BnnM/djJc5C7vNs86ZfSJ+pvE8Sp1Rhtuxh25EKMqGOn/SBedI
        gR6N5vGUNiIpG5Tf3DuYAMNFDUqw8uY0MyDJI+ZW3o3xrMUABzTH0ew+Piz85FDA
        YrVgwZfqyL+9OQuu6T66jOIdwQNRX2NPFZqvon8liZUPus5VzD8E5cAL9OPxQ3sF
        f7/zE91YIXUTimrv3L7eCgU1dSxKhhfvA2bEUi+AskMWFXFuETYVrIhFJAKnkFmE
        uZx+O9R9hADW3hM5hWHKH9/CRtb0/cC84I9oCWIQPdI+AaPtICxtsD2N8Q98hhhd
        4M7I0sLZhV+4ZJqzpUsOnSpaGyfh1Zy/1d3ijJi99/l+uVHuvmMllsNmgR+ZTj0=
        =LrCQ
        -----END PGP MESSAGE-----
```

When the pillar data is compiled, the results will be decrypted:

```
# salt myminion pillar.items
myminion:
    ----------
    secrets:
        ----------
        vault:
            ----------
            bar:
                this was unencrypted already
            baz:
                rosebud
            foo:
                supersecret
            qux:
                - foo
                - bar
                - baz
```

Salt must be told what portions of the pillar data to decrypt. This is done using the *decrypt_pillar* config option:

```
decrypt_pillar:
  - 'secrets:vault': gpg
```

The notation used to specify the pillar item(s) to be decrypted is the same as the one used in *pillar.get* function.

If a different delimiter is needed, it can be specified using the *decrypt_pillar_delimiter* config option:

```
decrypt_pillar:
  - 'secrets|vault': gpg

decrypt_pillar_delimiter: '|'
```

The name of the renderer used to decrypt a given pillar item can be omitted, and if so it will fall back to the value specified by the *decrypt_pillar_default* config option, which defaults to `gpg`. So, the first example above could be rewritten as:

```
decrypt_pillar:
  - 'secrets:vault'
```

Encrypted Pillar Data on the CLI

New in version 2016.3.0.

The following functions support passing pillar data on the CLI via the `pillar` argument:

- *pillar.items*
- *state.apply*
- *state.highstate*
- *state.sls*

Triggerring decryption of this CLI pillar data can be done in one of two ways:

1. Using the `pillar_enc` argument:

   ```
   # salt myminion pillar.items pillar_enc=gpg pillar='{foo: "-----BEGIN PGP MESSAGE-
   ↪----\n\nhQEMAw2B674HRhwSAQf+OvPqEdDoA2fk15I5dYUTDoj1yf/
   ↪pVolAma6iU4v8Zixn\nRDgWsaAnFz99FEiFACsAGDEFdZaVOxG80T0Lj+PnW4pVy0OXmXHnY2KjV9zx8FLS\nQxfvmhRR4
   ↪OvZHhxH7cnIiGQIHc7N9nQH7ibyoKQzQMSZeilSMGr2abAHun\nmLzscr4wKMb+81Z0/
   ↪fdBfP6g3bLWMJga3hSzSldU9ovu7KR8rDJI1qOlENj3Wm8C\nwTpDOB33kWIKMqiAjY3JFtb5MCHrafyggwQL7cX1+tI+A
   ↪FbjZ9CTWrQ=\n=0hO/\n-----END PGP MESSAGE-----"}'
   ```

   The newlines in this example are specified using a literal `\n`. Newlines can be replaced with a literal `\n` using `sed`:

   ```
   $ echo -n bar | gpg --armor --trust-model always --encrypt -r user@domain.tld |
   ↪sed ':a;N;$!ba;s/\n/\\n/g'
   ```

   Note: Using `pillar_enc` will perform the decryption minion-side, so for this to work it will be necessary to set up the keyring in `/etc/salt/gpgkeys` on the minion just as one would typically do on the master. The easiest way to do this is to first export the keys from the master:

   ```
   # gpg --homedir /etc/salt/gpgkeys --export-secret-key -a user@domain.tld >/tmp/
   ↪keypair.gpg
   ```

   Then, copy the file to the minion, setup the keyring, and import:

```
# mkdir -p /etc/salt/gpgkeys
# chmod 0700 /etc/salt/gpgkeys
# gpg --homedir /etc/salt/gpgkeys --list-keys
# gpg --homedir /etc/salt/gpgkeys --import --allow-secret-key-import keypair.gpg
```

The `--list-keys` command is run create a keyring in the newly-created directory.

Pillar data which is decrypted minion-side will still be securely transferred to the master, since the data sent between minion and master is encrypted with the master's public key.

2. Use the `decrypt_pillar` option. This is less flexible in that the pillar key passed on the CLI must be pre-configured on the master, but it doesn't require a keyring to be setup on the minion. One other caveat to this method is that pillar decryption on the master happens at the end of pillar compilation, so if the encrypted pillar data being passed on the CLI needs to be referenced by pillar or ext_pillar *during pillar compilation*, it *must* be decrypted minion-side.

Adding New Renderers for Decryption

Those looking to add new renderers for decryption should look at the `gpg` renderer for an example of how to do so. The function that performs the decryption should be recursive and be able to traverse a mutable type such as a dictionary, and modify the values in-place.

Once the renderer has been written, `decrypt_pillar_renderers` should be modified so that Salt allows it to be used for decryption.

If the renderer is being submitted upstream to the Salt project, the renderer should be added in salt/renderers/. Additionally, the following should be done:

- Both occurrences of `decrypt_pillar_renderers` in salt/config/__init__.py should be updated to include the name of the new renderer so that it is included in the default value for this config option.
- The documentation for the `decrypt_pillar_renderers` config option in the master config file and minion config file should be updated to show the correct new default value.
- The commented example for the `decrypt_pillar_renderers` config option in the master config template should be updated to show the correct new default value.

4.2.12 Binary Data in the Pillar

Salt has partial support for binary pillar data.

Note: There are some situations (such as salt-ssh) where only text (ASCII or Unicode) is allowed.

The simplest way to embed binary data in your pillar is to make use of YAML's built-in binary data type, which requires base64 encoded data.

```
salt_pic: !!binary
    iVBORw0KGgoAAAANSUhEUgAAAAoAAAAKCAMAAAC67D+PAAAABGdBTUEAALGPC/xhBQAAACBjSFJNAA
```

Then you can use it as a `contents_pillar` in a state:

```
/tmp/salt.png:
  file.managed:
    - contents_pillar: salt_pic
```

It is also possible to add ASCII-armored encrypted data to pillars, as mentioned in the Pillar Encryption section.

4.2.13 Master Config in Pillar

For convenience the data stored in the master configuration file can be made available in all minion's pillars. This makes global configuration of services and systems very easy but may not be desired if sensitive data is stored in the master configuration. This option is disabled by default.

To enable the master config from being added to the pillar set `pillar_opts` to `True` in the minion config file:

```
pillar_opts: True
```

4.2.14 Minion Config in Pillar

Minion configuration options can be set on pillars. Any option that you want to modify, should be in the first level of the pillars, in the same way you set the options in the config file. For example, to configure the MySQL root password to be used by MySQL Salt execution module, set the following pillar variable:

```
mysql.pass: hardtoguesspassword
```

4.2.15 Master Provided Pillar Error

By default if there is an error rendering a pillar, the detailed error is hidden and replaced with:

```
Rendering SLS 'my.sls' failed. Please see master log for details.
```

The error is protected because it's possible to contain templating data which would give that minion information it shouldn't know, like a password!

To have the master provide the detailed error that could potentially carry protected data set `pillar_safe_render_error` to `False`:

```
pillar_safe_render_error: False
```

Pillar Walkthrough

Note: This walkthrough assumes that the reader has already completed the initial Salt *walkthrough*.

Pillars are tree-like structures of data defined on the Salt Master and passed through to minions. They allow confidential, targeted data to be securely sent only to the relevant minion.

Note: Grains and Pillar are sometimes confused, just remember that Grains are data about a minion which is stored or generated from the minion. This is why information like the OS and CPU type are found in Grains. Pillar is information about a minion or many minions stored or generated on the Salt Master.

Pillar data is useful for:

Highly Sensitive Data: Information transferred via pillar is guaranteed to only be presented to the minions that are targeted, making Pillar suitable for managing security information, such as cryptographic keys and passwords.

Minion Configuration: Minion modules such as the execution modules, states, and returners can often be configured via data stored in pillar.

Variables: Variables which need to be assigned to specific minions or groups of minions can be defined in pillar and then accessed inside sls formulas and template files.

Arbitrary Data: Pillar can contain any basic data structure in dictionary format, so a key/value store can be defined making it easy to iterate over a group of values in sls formulas.

Pillar is therefore one of the most important systems when using Salt. This walkthrough is designed to get a simple Pillar up and running in a few minutes and then to dive into the capabilities of Pillar and where the data is available.

Setting Up Pillar

The pillar is already running in Salt by default. To see the minion's pillar data:

```
salt '*' pillar.items
```

Note: Prior to version 0.16.2, this function is named `pillar.data`. This function name is still supported for backwards compatibility.

By default, the contents of the master configuration file are not loaded into pillar for all minions. This default is stored in the `pillar_opts` setting, which defaults to `False`.

The contents of the master configuration file can be made available to minion pillar files. This makes global configuration of services and systems very easy, but note that this may not be desired or appropriate if sensitive data is stored in the master's configuration file. To enable the master configuration file to be available to a minion's pillar files, set `pillar_opts` to `True` in the minion configuration file.

Similar to the state tree, the pillar is comprised of sls files and has a top file. The default location for the pillar is in /srv/pillar.

Note: The pillar location can be configured via the `pillar_roots` option inside the master configuration file. It must not be in a subdirectory of the state tree or file_roots. If the pillar is under file_roots, any pillar targeting can be bypassed by minions.

To start setting up the pillar, the /srv/pillar directory needs to be present:

```
mkdir /srv/pillar
```

Now create a simple top file, following the same format as the top file used for states:

`/srv/pillar/top.sls`:

```
base:
  '*':
    - data
```

This top file associates the data.sls file to all minions. Now the `/srv/pillar/data.sls` file needs to be populated:

`/srv/pillar/data.sls`:

```
info: some data
```

To ensure that the minions have the new pillar data, issue a command to them asking that they fetch their pillars from the master:

```
salt '*' saltutil.refresh_pillar
```

Now that the minions have the new pillar, it can be retrieved:

```
salt '*' pillar.items
```

The key `info` should now appear in the returned pillar data.

More Complex Data

Unlike states, pillar files do not need to define **formulas**. This example sets up user data with a UID:

`/srv/pillar/users/init.sls`:

```
users:
  thatch: 1000
  shouse: 1001
  utahdave: 1002
  redbeard: 1003
```

Note: The same directory lookups that exist in states exist in pillar, so the file `users/init.sls` can be referenced with `users` in the *top file*.

The top file will need to be updated to include this sls file:

`/srv/pillar/top.sls`:

```
base:
  '*':
    - data
    - users
```

Now the data will be available to the minions. To use the pillar data in a state, you can use Jinja:

`/srv/salt/users/init.sls`

```
{% for user, uid in pillar.get('users', {}).items() %}
{{user}}:
  user.present:
    - uid: {{uid}}
{% endfor %}
```

This approach allows for users to be safely defined in a pillar and then the user data is applied in an sls file.

Parameterizing States With Pillar

Pillar data can be accessed in state files to customise behavior for each minion. All pillar (and grain) data applicable to each minion is substituted into the state files through templating before being run. Typical uses include setting directories appropriate for the minion and skipping states that don't apply.

A simple example is to set up a mapping of package names in pillar for separate Linux distributions:

`/srv/pillar/pkg/init.sls`:

```
pkgs:
  {% if grains['os_family'] == 'RedHat' %}
  apache: httpd
  vim: vim-enhanced
  {% elif grains['os_family'] == 'Debian' %}
  apache: apache2
  vim: vim
  {% elif grains['os'] == 'Arch' %}
  apache: apache
  vim: vim
  {% endif %}
```

The new `pkg` sls needs to be added to the top file:

`/srv/pillar/top.sls`:

```
base:
  '*':
    - data
    - users
    - pkg
```

Now the minions will auto map values based on respective operating systems inside of the pillar, so sls files can be safely parameterized:

`/srv/salt/apache/init.sls`:

```
apache:
  pkg.installed:
    - name: {{ pillar['pkgs']['apache'] }}
```

Or, if no pillar is available a default can be set as well:

Note: The function `pillar.get` used in this example was added to Salt in version 0.14.0

`/srv/salt/apache/init.sls`:

```
apache:
  pkg.installed:
    - name: {{ salt['pillar.get']('pkgs:apache', 'httpd') }}
```

In the above example, if the pillar value `pillar['pkgs']['apache']` is not set in the minion's pillar, then the default of `httpd` will be used.

Note: Under the hood, pillar is just a Python dict, so Python dict methods such as `get` and `items` can be used.

Pillar Makes Simple States Grow Easily

One of the design goals of pillar is to make simple sls formulas easily grow into more flexible formulas without refactoring or complicating the states.

A simple formula:

`/srv/salt/edit/vim.sls`:

```
vim:
  pkg.installed: []

/etc/vimrc:
  file.managed:
    - source: salt://edit/vimrc
    - mode: 644
    - user: root
    - group: root
    - require:
      - pkg: vim
```

Can be easily transformed into a powerful, parameterized formula:

`/srv/salt/edit/vim.sls`:

```
vim:
  pkg.installed:
    - name: {{ pillar['pkgs']['vim'] }}

/etc/vimrc:
  file.managed:
    - source: {{ pillar['vimrc'] }}
    - mode: 644
    - user: root
    - group: root
    - require:
      - pkg: vim
```

Where the vimrc source location can now be changed via pillar:

`/srv/pillar/edit/vim.sls`:

```
{% if grains['id'].startswith('dev') %}
vimrc: salt://edit/dev_vimrc
{% elif grains['id'].startswith('qa') %}
vimrc: salt://edit/qa_vimrc
{% else %}
vimrc: salt://edit/vimrc
{% endif %}
```

Ensuring that the right vimrc is sent out to the correct minions.

The pillar top file must include a reference to the new sls pillar file:

`/srv/pillar/top.sls`:

```
base:
  '*':
    - pkg
    - edit.vim
```

Setting Pillar Data on the Command Line

Pillar data can be set on the command line when running `state.apply <salt.modules.state.apply_()` like so:

```
salt '*' state.apply pillar='{"foo": "bar"}'
salt '*' state.apply my_sls_file pillar='{"hello": "world"}'
```

Nested pillar values can also be set via the command line:

```
salt '*' state.sls my_sls_file pillar='{"foo": {"bar": "baz"}}'
```

Lists can be passed via command line pillar data as follows:

```
salt '*' state.sls my_sls_file pillar='{"some_list": ["foo", "bar", "baz"]}'
```

Note: If a key is passed on the command line that already exists on the minion, the key that is passed in will overwrite the entire value of that key, rather than merging only the specified value set via the command line.

The example below will swap the value for vim with telnet in the previously specified list, notice the nested pillar dict:

```
salt '*' state.apply edit.vim pillar='{"pkgs": {"vim": "telnet"}}'
```

This will attempt to install telnet on your minions, feel free to uninstall the package or replace telnet value with anything else.

Note: Be aware that when sending sensitive data via pillar on the command-line that the publication containing that data will be received by all minions and will not be restricted to the targeted minions. This may represent a security concern in some cases.

More On Pillar

Pillar data is generated on the Salt master and securely distributed to minions. Salt is not restricted to the pillar sls files when defining the pillar but can retrieve data from external sources. This can be useful when information about an infrastructure is stored in a separate location.

Reference information on pillar and the external pillar interface can be found in the Salt documentation:

Pillar

Minion Config in Pillar

Minion configuration options can be set on pillars. Any option that you want to modify, should be in the first level of the pillars, in the same way you set the options in the config file. For example, to configure the MySQL root password to be used by MySQL Salt execution module:

```
mysql.pass: hardtoguesspassword
```

This is very convenient when you need some dynamic configuration change that you want to be applied on the fly. For example, there is a chicken and the egg problem if you do this:

```
mysql-admin-passwd:
  mysql_user.present:
    - name: root
```

(continues on next page)

(continued from previous page)

```
    - password: somepasswd

mydb:
  mysql_db.present
```

The second state will fail, because you changed the root password and the minion didn't notice it. Setting mysql.pass in the pillar, will help to sort out the issue. But always change the root admin password in the first place.

This is very helpful for any module that needs credentials to apply state changes: mysql, keystone, etc.

4.3 Targeting Minions

Targeting minions is specifying which minions should run a command or execute a state by matching against hostnames, or system information, or defined groups, or even combinations thereof.

For example the command `salt web1 apache.signal restart` to restart the Apache httpd server specifies the machine `web1` as the target and the command will only be run on that one minion.

Similarly when using States, the following *top file* specifies that only the `web1` minion should execute the contents of `webserver.sls`:

```
base:
  'web1':
    - webserver
```

The simple target specifications, glob, regex, and list will cover many use cases, and for some will cover all use cases, but more powerful options exist.

4.3.1 Targeting with Grains

The Grains interface was built into Salt to allow minions to be targeted by system properties. So minions running on a particular operating system can be called to execute a function, or a specific kernel.

Calling via a grain is done by passing the -G option to salt, specifying a grain and a glob expression to match the value of the grain. The syntax for the target is the grain key followed by a glob expression: "os:Arch*".

```
salt -G 'os:Fedora' test.version
```

Will return True from all of the minions running Fedora.

To discover what grains are available and what the values are, execute the grains.item salt function:

```
salt '*' grains.items
```

More info on using targeting with grains can be found *here*.

4.3.2 Compound Targeting

New in version 0.9.5.

Multiple target interfaces can be used in conjunction to determine the command targets. These targets can then be combined using `and` or `or` statements. This is well defined with an example:

```
salt -C 'G@os:Debian and webser* or E@db.*' test.version
```

In this example any minion who's id starts with `webser` and is running Debian, or any minion who's id starts with db will be matched.

The type of matcher defaults to glob, but can be specified with the corresponding letter followed by the `@` symbol. In the above example a grain is used with `G@` as well as a regular expression with `E@`. The `webser*` target does not need to be prefaced with a target type specifier because it is a glob.

More info on using compound targeting can be found *here*.

4.3.3 Node Group Targeting

New in version 0.9.5.

For certain cases, it can be convenient to have a predefined group of minions on which to execute commands. This can be accomplished using what are called *nodegroups*. Nodegroups allow for predefined compound targets to be declared in the master configuration file, as a sort of shorthand for having to type out complicated compound expressions.

```
nodegroups:
  group1: 'L@foo.domain.com,bar.domain.com,baz.domain.com and bl*.domain.com'
  group2: 'G@os:Debian and foo.domain.com'
  group3: 'G@os:Debian and N@group1'
```

4.3.4 Advanced Targeting Methods

There are many ways to target individual minions or groups of minions in Salt:

Matching the `minion id`

Each minion needs a unique identifier. By default when a minion starts for the first time it chooses its FQDN (fully qualified domain name) as that identifier. The minion id can be overridden via the minion's *`id`* configuration setting.

Tip: minion id and minion keys

The *minion id* is used to generate the minion's public/private keys and if it ever changes the master must then accept the new key as though the minion was a new host.

Globbing

The default matching that Salt utilizes is `shell-style globbing` around the *minion id*. This also works for states in the *top file*.

Note: You must wrap **`salt`** calls that use globbing in single-quotes to prevent the shell from expanding the globs before Salt is invoked.

Match all minions:

```
salt '*' test.version
```

Match all minions in the example.net domain or any of the example domains:

```
salt '*.example.net' test.version
salt '*.example.*' test.version
```

Match all the `webN` minions in the example.net domain (`web1.example.net`, `web2.example.net` … `webN.example.net`):

```
salt 'web?.example.net' test.version
```

Match the `web1` through `web5` minions:

```
salt 'web[1-5]' test.version
```

Match the `web1` and `web3` minions:

```
salt 'web[1,3]' test.version
```

Match the `web-x`, `web-y`, and `web-z` minions:

```
salt 'web-[x-z]' test.version
```

Note: For additional targeting methods please review the *compound matchers* documentation.

Regular Expressions

Minions can be matched using Perl-compatible `regular expressions` (which is globbing on steroids and a ton of caffeine).

Match both `web1-prod` and `web1-devel` minions:

```
salt -E 'web1-(prod|devel)' test.version
```

When using regular expressions in a State's *top file*, you must specify the matcher as the first option. The following example executes the contents of `webserver.sls` on the above-mentioned minions.

```
base:
  'web1-(prod|devel)':
  - match: pcre
  - webserver
```

Lists

At the most basic level, you can specify a flat list of minion IDs:

```
salt -L 'web1,web2,web3' test.version
```

Targeting using Grains

Grain data can be used when targeting minions.

For example, the following matches all CentOS minions:

```
salt -G 'os:CentOS' test.version
```

Match all minions with 64-bit CPUs, and return number of CPU cores for each matching minion:

```
salt -G 'cpuarch:x86_64' grains.item num_cpus
```

Additionally, globs can be used in grain matches, and grains that are nested in a dictionary can be matched by adding a colon for each level that is traversed. For example, the following will match hosts that have a grain called `ec2_tags`, which itself is a dictionary with a key named `environment`, which has a value that contains the word `production`:

```
salt -G 'ec2_tags:environment:*production*'
```

Important: See *Is Targeting using Grain Data Secure?* for important security information.

Targeting using Pillar

Pillar data can be used when targeting minions. This allows for ultimate control and flexibility when targeting minions.

Note: To start using Pillar targeting it is required to make a Pillar data cache on Salt Master for each Minion via following commands: `salt '*' saltutil.refresh_pillar` or `salt '*' saltutil.sync_all`. Also Pillar data cache will be populated during the *highstate* run. Once Pillar data changes, you must refresh the cache by running above commands for this targeting method to work correctly.

Example:

```
salt -I 'somekey:specialvalue' test.version
```

Like with *Grains*, it is possible to use globbing as well as match nested values in Pillar, by adding colons for each level that is being traversed. The below example would match minions with a pillar named `foo`, which is a dict containing a key `bar`, with a value beginning with `baz`:

```
salt -I 'foo:bar:baz*' test.version
```

Subnet/IP Address Matching

Minions can easily be matched based on IP address, or by subnet (using CIDR notation).

```
salt -S 192.168.40.20 test.version
salt -S 2001:db8::/64 test.version
```

Ipcidr matching can also be used in compound matches

```
salt -C 'S@10.0.0.0/24 and G@os:Debian' test.version
```

It is also possible to use in both pillar and state-matching

```
'172.16.0.0/12':
   - match: ipcidr
   - internal
```

Compound matchers

Compound matchers allow very granular minion targeting using any of Salt's matchers. The default matcher is a `glob` match, just as with CLI and *top file* matching. To match using anything other than a glob, prefix the match string with the appropriate letter from the table below, followed by an @ sign.

Letter	Match Type	Example	*Alt Delimiter?*
G	Grains glob	`G@os:Ubuntu`	Yes
E	PCRE Minion ID	`E@web\d+\.(dev|qa|prod)\.loc`	No
P	Grains PCRE	`P@os:(RedHat|Fedora|CentOS)`	Yes
L	List of minions	`L@minion1.example.com,minion3.domain.com or bl*.domain.com`	No
I	Pillar glob	`I@pdata:foobar`	Yes
J	Pillar PCRE	`J@pdata:^(foo|bar)$`	Yes
S	Subnet/IP address	`S@192.168.1.0/24` or `S@192.168.1.100`	No
R	Range cluster	`R@%foo.bar`	No
N	Nodegroups	`N@group1`	No

Matchers can be joined using boolean `and`, `or`, and `not` operators.

For example, the following string matches all Debian minions with a hostname that begins with `webserv`, as well as any minions that have a hostname which matches the `regular expression` `web-dc1-srv.*`:

```
salt -C 'webserv* and G@os:Debian or E@web-dc1-srv.*' test.version
```

That same example expressed in a *top file* looks like the following:

```
base:
  'webserv* and G@os:Debian or E@web-dc1-srv.*':
    - match: compound
    - webserver
```

New in version 2015.8.0.

Excluding a minion based on its ID is also possible:

```
salt -C 'not web-dc1-srv' test.version
```

Versions prior to 2015.8.0 a leading `not` was not supported in compound matches. Instead, something like the following was required:

```
salt -C '* and not G@kernel:Darwin' test.version
```

Excluding a minion based on its ID was also possible:

```
salt -C '* and not web-dc1-srv' test.version
```

Precedence Matching

Matchers can be grouped together with parentheses to explicitly declare precedence amongst groups.

```
salt -C '( ms-1 or G@id:ms-3 ) and G@id:ms-3' test.version
```

Note: Be certain to note that spaces are required between the parentheses and targets. Failing to obey this rule may result in incorrect targeting!

Alternate Delimiters

New in version 2015.8.0.

Matchers that target based on a key value pair use a colon (`:`) as a delimiter. Matchers with a `Yes` in the `Alt Delimiters` column in the previous table support specifying an alternate delimiter character.

This is done by specifying an alternate delimiter character between the leading matcher character and the `@` pattern separator character. This avoids incorrect interpretation of the pattern in the case that `:` is part of the grain or pillar data structure traversal.

```
salt -C 'J|@foo|bar|^foo:bar$ or J!@gitrepo!https://github.com:example/project.git'
↪test.ping
```

Node groups

Nodegroups are declared using a compound target specification. The compound target documentation can be found *here*.

The *`nodegroups`* master config file parameter is used to define nodegroups. Here's an example nodegroup configuration within `/etc/salt/master`:

```
nodegroups:
  group1: 'L@foo.domain.com,bar.domain.com,baz.domain.com or bl*.domain.com'
  group2: 'G@os:Debian and foo.domain.com'
  group3: 'G@os:Debian and N@group1'
  group4:
    - 'G@foo:bar'
    - 'or'
    - 'G@foo:baz'
```

Note: The `L` within group1 is matching a list of minions, while the `G` in group2 is matching specific grains. See the *compound matchers* documentation for more details.

As of the 2017.7.0 release of Salt, group names can also be prepended with a dash. This brings the usage in line with many other areas of Salt. For example:

```
nodegroups:
  - group1: 'L@foo.domain.com,bar.domain.com,baz.domain.com or bl*.domain.com'
```

New in version 2015.8.0.

Note: Nodegroups can reference other nodegroups as seen in `group3`. Ensure that you do not have circular references. Circular references will be detected and cause partial expansion with a logged error message.

New in version 2015.8.0.

Compound nodegroups can be either string values or lists of string values. When the nodegroup is A string value will be tokenized by splitting on whitespace. This may be a problem if whitespace is necessary as part of a pattern. When a nodegroup is a list of strings then tokenization will happen for each list element as a whole.

To match a nodegroup on the CLI, use the `-N` command-line option:

```
salt -N group1 test.version
```

New in version 2019.2.0.

Note: The `N@` classifier historically could not be used in compound matches within the CLI or *top file*, it was only recognized in the `nodegroups` master config file parameter. As of the 2019.2.0 release, this limitation no longer exists.

To match a nodegroup in your *top file*, make sure to put `- match: nodegroup` on the line directly following the nodegroup name.

```
base:
  group1:
    - match: nodegroup
    - webserver
```

Note: When adding or modifying nodegroups to a master configuration file, the master must be restarted for those changes to be fully recognized.

A limited amount of functionality, such as targeting with -N from the command-line may be available without a restart.

Defining Nodegroups as Lists of Minion IDs

A simple list of minion IDs would traditionally be defined like this:

```
nodegroups:
  group1: L@host1,host2,host3
```

They can now also be defined as a YAML list, like this:

```
nodegroups:
  group1:
    - host1
```

(continues on next page)

(continued from previous page)

```
    - host2
    - host3
```

New in version 2016.11.0.

Batch Size

The `-b` (or `--batch-size`) option allows commands to be executed on only a specified number of minions at a time. Both percentages and finite numbers are supported.

```
salt '*' -b 10 test.version

salt -G 'os:RedHat' --batch-size 25% apache.signal restart
```

This will only run test.version on 10 of the targeted minions at a time and then restart apache on 25% of the minions matching `os:RedHat` at a time and work through them all until the task is complete. This makes jobs like rolling web server restarts behind a load balancer or doing maintenance on BSD firewalls using carp much easier with salt.

The batch system maintains a window of running minions, so, if there are a total of 150 minions targeted and the batch size is 10, then the command is sent to 10 minions, when one minion returns then the command is sent to one additional minion, so that the job is constantly running on 10 minions.

New in version 2016.3.

The `--batch-wait` argument can be used to specify a number of seconds to wait after a minion returns, before sending the command to a new minion.

SECO Range

SECO range is a cluster-based metadata store developed and maintained by Yahoo!

The Range project is hosted here:

https://github.com/ytoolshed/range

Learn more about range here:

https://github.com/ytoolshed/range/wiki/

Prerequisites

To utilize range support in Salt, a range server is required. Setting up a range server is outside the scope of this document. Apache modules are included in the range distribution.

With a working range server, cluster files must be defined. These files are written in YAML and define hosts contained inside a cluster. Full documentation on writing YAML range files is here:

https://github.com/ytoolshed/range/wiki/%22yamlfile%22-module-file-spec

Additionally, the Python seco range libraries must be installed on the salt master. One can verify that they have been installed correctly via the following command:

```
python -c 'import seco.range'
```

If no errors are returned, range is installed successfully on the salt master.

Preparing Salt

Range support must be enabled on the salt master by setting the hostname and port of the range server inside the master configuration file:

```
range_server: my.range.server.com:80
```

Following this, the master must be restarted for the change to have an effect.

Targeting with Range

Once a cluster has been defined, it can be targeted with a salt command by using the `-R` or `--range` flags.

For example, given the following range YAML file being served from a range server:

```
$ cat /etc/range/test.yaml
CLUSTER: host1..100.test.com
APPS:
  - frontend
  - backend
  - mysql
```

One might target host1 through host100 in the test.com domain with Salt as follows:

```
salt --range %test:CLUSTER test.version
```

The following salt command would target three hosts: `frontend`, `backend`, and `mysql`:

```
salt --range %test:APPS test.version
```

4.3.5 Loadable Matchers

New in version 2019.2.0.

Internally targeting is implemented with chunks of code called Matchers. As of the 2019.2.0 release, matchers can be loaded dynamically. Currently new matchers cannot be created, but existing matchers can have their functionality altered or extended. For more information on Matchers see

Matchers

New in version 3000.

Matchers are modules that provide Salt's targeting abilities. As of the 3000 release, matchers can be dynamically loaded. Currently new matchers cannot be created because the required plumbing for the CLI does not exist yet. Existing matchers may have their functionality altered or extended.

For details of targeting methods, see the *Targeting* topic.

A matcher module must have a function called `match()`. This function ends up becoming a method on the Matcher class. All matcher functions require at least two arguments, `self` (because the function will be turned into a method), and `tgt`, which is the actual target string. The grains and pillar matchers also take a `delimiter` argument and should default to `DEFAULT_TARGET_DELIM`.

Like other Salt loadable modules, modules that override built-in functionality can be placed in `file_roots` in a special directory and then copied to the minion through the normal sync process. *`saltutil.sync_all`* will transfer all loadable modules, and the 3000 release introduces *saltutil.sync_matchers*. For matchers, the directory is `/srv/salt/_matchers` (assuming your `file_roots` is set to the default `/srv/salt`).

As an example, let's modify the `list` matcher to have the separator be a '`/`' instead of the default '`,`'.

```
from __future__ import absolute_import, print_function, unicode_literals
from salt.ext import six  # pylint: disable=3rd-party-module-not-gated

def match(self, tgt):
    '''
    Determines if this host is on the list
    '''
    if isinstance(tgt, six.string_types):
        # The stock matcher splits on `,`.  Change to `/` below.
        tgt = tgt.split('/')
    return bool(self.opts['id'] in tgt)
```

Place this code in a file called `list_matcher.py` in `_matchers` in your `file_roots`. Sync this down to your minions with *saltutil.sync_matchers*. Then attempt to match with the following, replacing `minionX` with three of your minions.

```
salt -L 'minion1/minion2/minion3' test.ping
```

Three of your minions should respond.

The current supported matchers and associated filenames are

Salt CLI Switch	Match Type	Filename
<none>	Glob	glob_match.py
-C	Compound	compound_match.py
-E	Perl-Compatible Regular Expressions	pcre_match.py
-L	List	list_match.py
-G	Grain	grain_match.py
-P	Grain Perl-Compatible Regular Expressions	grain_pcre_match.py
-N	Nodegroup	nodegroup_match.py
-R	Range	range_match.py
-I	Pillar	pillar_match.py
-J	Pillar Perl-Compatible Regular Expressions	pillar_pcre.py
-S	IP-Classless Internet Domain Routing	ipcidr_match.py

4.4 The Salt Mine

The Salt Mine is used to collect arbitrary data from Minions and store it on the Master. This data is then made available to all Minions via the *salt.modules.mine* module.

Mine data is gathered on the Minion and sent back to the Master where only the most recent data is maintained (if long term data is required use returners or the external job cache).

4.4.1 Mine vs Grains

Mine data is designed to be much more up-to-date than grain data. Grains are refreshed on a very limited basis and are largely static data. Mines are designed to replace slow peer publishing calls when Minions need data from other Minions. Rather than having a Minion reach out to all the other Minions for a piece of data, the Salt Mine, running on the Master, can collect it from all the Minions every *Mine Interval*, resulting in almost fresh data at any given time, with much less overhead.

4.4.2 Mine Functions

To enable the Salt Mine the `mine_functions` option needs to be applied to a Minion. This option can be applied via the Minion's configuration file, or the Minion's Pillar. The `mine_functions` option dictates what functions are being executed and allows for arguments to be passed in. The list of functions are available in the `salt.module`. If no arguments are passed, an empty list must be added like in the `test.ping` function in the example below:

```
mine_functions:
  test.ping: []
  network.ip_addrs:
    interface: eth0
    cidr: 10.0.0.0/8
```

In the example above `salt.modules.network.ip_addrs` has additional filters to help narrow down the results. In the above example IP addresses are only returned if they are on a eth0 interface and in the 10.0.0.0/8 IP range.

Changed in version 3000.

The format to define mine_functions has been changed to allow the same format as used for module.run. The old format (above) will still be supported.

```
mine_functions:
  test.ping: []
  network.ip_addrs:
    - interface: eth0
    - cidr: 10.0.0.0/8
  test.arg:
    - isn't
    - this
    - fun
    - this: that
    - salt: stack
```

Minion-side Access Control

New in version 3000.

Mine functions can be targeted to only be available to specific minions. This uses the same targeting parameters as *Targeting Minions* but with keywords `allow_tgt` and `allow_tgt_type`. When a minion requests a function from the salt mine that is not allowed to be requested by that minion (i.e. when looking up the combination of `allow_tgt` and `allow_tgt_type` and the requesting minion is not in the list) it will get no data, just as if the requested function is not present in the salt mine.

```
mine_functions:
  network.ip_addrs:
    - interface: eth0
    - cidr: 10.0.0.0/8
    - allow_tgt: 'G@role:master'
    - allow_tgt_type: 'compound'
```

Mine Functions Aliases

Function aliases can be used to provide friendly names, usage intentions or to allow multiple calls of the same function with different arguments. There is a different syntax for passing positional and key-value arguments. Mixing positional and key-value arguments is not supported.

New in version 2014.7.0.

```
mine_functions:
  network.ip_addrs: [eth0]
  networkplus.internal_ip_addrs: []
  internal_ip_addrs:
    mine_function: network.ip_addrs
    cidr: 192.168.0.0/16
  ip_list:
    - mine_function: grains.get
    - ip_interfaces
```

Changed in version 3000.

With the addition of the module.run-like format for defining mine_functions, the method of adding aliases remains similar. Just add a `mine_function` kwarg with the name of the real function to call, making the key below `mine_functions` the alias:

```
mine_functions:
  alias_name:
    - mine_function: network.ip_addrs
    - eth0
  internal_ip_addrs:
    - mine_function: network.ip_addrs
    - cidr: 192.168.0.0/16
  ip_list:
    - mine_function: grains.get
    - ip_interfaces
```

4.4.3 Mine Interval

The Salt Mine functions are executed when the Minion starts and at a given interval by the scheduler. The default interval is every 60 minutes and can be adjusted for the Minion via the `mine_interval` option in the minion config:

```
mine_interval: 60
```

4.4.4 Mine in Salt-SSH

As of the 2015.5.0 release of salt, salt-ssh supports `mine.get`.

Because the Minions cannot provide their own `mine_functions` configuration, we retrieve the args for specified mine functions in one of three places, searched in the following order:

1. Roster data
2. Pillar
3. Master config

The `mine_functions` are formatted exactly the same as in normal salt, just stored in a different location. Here is an example of a flat roster containing `mine_functions`:

```
test:
  host: 104.237.131.248
  user: root
  mine_functions:
    cmd.run: ['echo "hello!"']
    network.ip_addrs:
      interface: eth0
```

Note: Because of the differences in the architecture of salt-ssh, `mine.get` calls are somewhat inefficient. Salt must make a new salt-ssh call to each of the Minions in question to retrieve the requested data, much like a publish call. However, unlike publish, it must run the requested function as a wrapper function, so we can retrieve the function args from the pillar of the Minion in question. This results in a non-trivial delay in retrieving the requested data.

4.4.5 Minions Targeting with Mine

The `mine.get` function supports various methods of *Minions targeting* to fetch Mine data from particular hosts, such as glob or regular expression matching on Minion id (name), grains, pillars and *compound matches*. See the `salt.modules.mine` module documentation for the reference.

Note: Pillar data needs to be cached on Master for pillar targeting to work with Mine. Read the note in *relevant section*.

4.4.6 Example

One way to use data from Salt Mine is in a State. The values can be retrieved via Jinja and used in the SLS file. The following example is a partial HAProxy configuration file and pulls IP addresses from all Minions with the "web" grain to add them to the pool of load balanced servers.

`/srv/pillar/top.sls`:

```
base:
  'G@roles:web':
    - web
```

`/srv/pillar/web.sls`:

```
mine_functions:
  network.ip_addrs: [eth0]
```

Then trigger the minions to refresh their pillar data by running:

```
salt '*' saltutil.refresh_pillar
```

Verify that the results are showing up in the pillar on the minions by executing the following and checking for `network.ip_addrs` in the output:

```
salt '*' pillar.items
```

Which should show that the function is present on the minion, but not include the output:

```
minion1.example.com:
    ----------
    mine_functions:
        ----------
        network.ip_addrs:
            - eth0
```

Mine data is typically only updated on the master every 60 minutes, this can be modified by setting:

`/etc/salt/minion.d/mine.conf`:

```
mine_interval: 5
```

To force the mine data to update immediately run:

```
salt '*' mine.update
```

Setup the *`salt.states.file.managed`* state in `/srv/salt/haproxy.sls`:

```
haproxy_config:
  file.managed:
    - name: /etc/haproxy/config
    - source: salt://haproxy_config
    - template: jinja
```

Create the Jinja template in `/srv/salt/haproxy_config`:

```
<...file contents snipped...>

{% for server, addrs in salt['mine.get']('roles:web', 'network.ip_addrs', tgt_type=
↪'grain') | dictsort() %}
server {{ server }} {{ addrs[0] }}:80 check
{% endfor %}

<...file contents snipped...>
```

In the above example, `server` will be expanded to the `minion_id`.

Note: The expr_form argument will be renamed to `tgt_type` in the 2017.7.0 release of Salt.

4.5 Runners

Salt runners are convenience applications executed with the salt-run command.

Salt runners work similarly to Salt execution modules however they execute on the Salt master itself instead of remote Salt minions.

A Salt runner can be a simple client call or a complex application.

See also:

The full list of runners

4.5.1 Writing Salt Runners

A Salt runner is written in a similar manner to a Salt execution module. Both are Python modules which contain functions and each public function is a runner which may be executed via the *salt-run* command.

For example, if a Python module named `test.py` is created in the runners directory and contains a function called `foo`, the `test` runner could be invoked with the following command:

```
# salt-run test.foo
```

Runners have several options for controlling output.

Any `print` statement in a runner is automatically also fired onto the master event bus where. For example:

```
def a_runner(outputter=None, display_progress=False):
    print('Hello world')
    ...
```

The above would result in an event fired as follows:

```
Event fired at Tue Jan 13 15:26:45 2015
*************************
Tag: salt/run/20150113152644070246/print
Data:
{'_stamp': '2015-01-13T15:26:45.078707',
 'data': 'hello',
  'outputter': 'pprint'}
```

A runner may also send a progress event, which is displayed to the user during runner execution and is also passed across the event bus if the `display_progress` argument to a runner is set to True.

A custom runner may send its own progress event by using the `__jid_event_.fire_event()` method as shown here:

```
if display_progress:
    __jid_event__.fire_event({'message': 'A progress message'}, 'progress')
```

The above would produce output on the console reading: `A progress message` as well as an event on the event similar to:

```
Event fired at Tue Jan 13 15:21:20 2015
*************************
Tag: salt/run/20150113152118341421/progress
Data:
{'_stamp': '2015-01-13T15:21:20.390053',
 'message': "A progress message"}
```

A runner could use the same approach to send an event with a customized tag onto the event bus by replacing the second argument (`progress`) with whatever tag is desired. However, this will not be shown on the command-line and will only be fired onto the event bus.

4.5.2 Synchronous vs. Asynchronous

A runner may be fired asynchronously which will immediately return control. In this case, no output will be display to the user if `salt-run` is being used from the command-line. If used programmatically, no results will be returned. If results are desired, they must be gathered either by firing events on the bus from the runner and then watching for them or by some other means.

Note: When running a runner in asynchronous mode, the `--progress` flag will not deliver output to the salt-run CLI. However, progress events will still be fired on the bus.

In synchronous mode, which is the default, control will not be returned until the runner has finished executing.

To add custom runners, put them in a directory and add it to `runner_dirs` in the master configuration file.

4.5.3 Examples

Examples of runners can be found in the Salt distribution:

https://github.com/saltstack/salt/blob/master/salt/runners

A simple runner that returns a well-formatted list of the minions that are responding to Salt calls could look like this:

```python
# Import salt modules
import salt.client

def up():
    '''
    Print a list of all of the minions that are up
    '''
    client = salt.client.LocalClient(__opts__['conf_file'])
    minions = client.cmd('*', 'test.version', timeout=1)
    for minion in sorted(minions):
        print minion
```

4.6 Salt Engines

New in version 2015.8.0.

Salt Engines are long-running, external system processes that leverage Salt.

- Engines have access to Salt configuration, execution modules, and runners (`__opts__`, `__salt__`, and `__runners__`).
- Engines are executed in a separate process that is monitored by Salt. If a Salt engine stops, it is restarted automatically.
- Engines can run on the Salt master and on Salt minions.

Salt engines enhance and replace the external processes functionality.

4.6.1 Configuration

Salt engines are configured under an `engines` top-level section in your Salt master or Salt minion configuration. Provide a list of engines and parameters under this section.

```yaml
engines:
  - logstash:
      host: log.my_network.com
      port: 5959
      proto: tcp
```

New in version 3000.

Multiple copies of a particular Salt engine can be configured by including the `engine_module` parameter in the engine configuration.

```
engines:
  - production_logstash:
      host: production_log.my_network.com
      port: 5959
      proto: tcp
      engine_module: logstash
  - develop_logstash:
      host: develop_log.my_network.com
      port: 5959
      proto: tcp
      engine_module: logstash
```

Salt engines must be in the Salt path, or you can add the `engines_dirs` option in your Salt master configuration with a list of directories under which Salt attempts to find Salt engines. This option should be formatted as a list of directories to search, such as:

```
engines_dirs:
  - /home/bob/engines
```

4.6.2 Writing an Engine

An example Salt engine, https://github.com/saltstack/salt/blob/master/salt/engines/test.py, is available in the Salt source. To develop an engine, the only requirement is that your module implement the `start()` function.

4.7 Understanding YAML

The default renderer for SLS files is the YAML renderer. YAML is a markup language with many powerful features. However, Salt uses a small subset of YAML that maps over very commonly used data structures, like lists and dictionaries. It is the job of the YAML renderer to take the YAML data structure and compile it into a Python data structure for use by Salt.

Though YAML syntax may seem daunting and terse at first, there are only three very simple rules to remember when writing YAML for SLS files.

4.7.1 Rule One: Indentation

YAML uses a fixed indentation scheme to represent relationships between data layers. Salt requires that the indentation for each level consists of exactly two spaces. Do not use tabs.

4.7.2 Rule Two: Colons

Python dictionaries are, of course, simply key-value pairs. Users from other languages may recognize this data type as hashes or associative arrays.

Dictionary keys are represented in YAML as strings terminated by a trailing colon. Values are represented by either a string following the colon, separated by a space:

```
my_key: my_value
```

In Python, the above maps to:

```
{'my_key': 'my_value'}
```

Alternatively, a value can be associated with a key through indentation.

```
my_key:
  my_value
```

Note: The above syntax is valid YAML but is uncommon in SLS files because most often, the value for a key is not singular but instead is a *list* of values.

In Python, the above maps to:

```
{'my_key': 'my_value'}
```

Dictionaries can be nested:

```
first_level_dict_key:
  second_level_dict_key: value_in_second_level_dict
```

And in Python:

```
{
    'first_level_dict_key': {
        'second_level_dict_key': 'value_in_second_level_dict'
    }
}
```

4.7.3 Rule Three: Dashes

To represent lists of items, a single dash followed by a space is used. Multiple items are a part of the same list as a function of their having the same level of indentation.

```
- list_value_one
- list_value_two
- list_value_three
```

Lists can be the value of a key-value pair. This is quite common in Salt:

```
my_dictionary:
  - list_value_one
  - list_value_two
  - list_value_three
```

In Python, the above maps to:

```
{'my_dictionary': ['list_value_one', 'list_value_two', 'list_value_three']}
```

4.7.4 Learning More

One easy way to learn more about how YAML gets rendered into Python data structures is to use an online YAML parser to see the Python output.

One excellent choice for experimenting with YAML parsing is: http://yaml-online-parser.appspot.com/

4.7.5 Templating

Jinja statements and expressions are allowed by default in SLS files. See *Understanding Jinja*.

4.8 Understanding Jinja

Jinja is the default templating language in SLS files.

4.8.1 Jinja in States

Jinja is evaluated before YAML, which means it is evaluated before the States are run.

The most basic usage of Jinja in state files is using control structures to wrap conditional or redundant state elements:

```
{% if grains['os'] != 'FreeBSD' %}
tcsh:
    pkg:
        - installed
{% endif %}

motd:
  file.managed:
    {% if grains['os'] == 'FreeBSD' %}
    - name: /etc/motd
    {% elif grains['os'] == 'Debian' %}
    - name: /etc/motd.tail
    {% endif %}
    - source: salt://motd
```

In this example, the first **if** block will only be evaluated on minions that aren't running FreeBSD, and the second block changes the file name based on the *os* grain.

Writing **if-else** blocks can lead to very redundant state files however. In this case, using *pillars*, or using a previously defined variable might be easier:

```
{% set motd = ['/etc/motd'] %}
{% if grains['os'] == 'Debian' %}
  {% set motd = ['/etc/motd.tail', '/var/run/motd'] %}
{% endif %}

{% for motdfile in motd %}
{{ motdfile }}:
  file.managed:
    - source: salt://motd
{% endfor %}
```

Using a variable set by the template, the for loop will iterate over the list of MOTD files to update, adding a state block for each file.

The filter_by function can also be used to set variables based on grains:

```
{% set auditd = salt['grains.filter_by']({
'RedHat': { 'package': 'audit' },
'Debian': { 'package': 'auditd' },
}) %}
```

4.8.2 Include and Import

Includes and imports can be used to share common, reusable state configuration between state files and between files.

```
{% from 'lib.sls' import test %}
```

This would import the `test` template variable or macro, not the `test` state element, from the file `lib.sls`. In the case that the included file performs checks against grains, or something else that requires context, passing the context into the included file is required:

```
{% from 'lib.sls' import test with context %}
```

Includes must use full paths, like so:

Listing 1: spam/eggs.jinja

```
 {% include 'spam/foobar.jinja' %}
```

Including Context During Include/Import

By adding `with context` to the include/import directive, the current context can be passed to an included/imported template.

```
{% import 'openssl/vars.sls' as ssl with context %}
```

4.8.3 Macros

Macros are helpful for eliminating redundant code. Macros are most useful as mini-templates to repeat blocks of strings with a few parameterized variables. Be aware that stripping whitespace from the template block, as well as contained blocks, may be necessary to emulate a variable return from the macro.

```
# init.sls
{% from 'lib.sls' import pythonpkg with context %}

python-virtualenv:
  pkg.installed:
    - name: {{ pythonpkg('virtualenv') }}

python-fabric:
  pkg.installed:
    - name: {{ pythonpkg('fabric') }}
```

```
# lib.sls
{% macro pythonpkg(pkg) -%}
  {%- if grains['os'] == 'FreeBSD' -%}
    py27-{{ pkg }}
  {%- elif grains['os'] == 'Debian' -%}
    python-{{ pkg }}
  {%- endif -%}
{%- endmacro %}
```

This would define a macro that would return a string of the full package name, depending on the packaging system's naming convention. The whitespace of the macro was eliminated, so that the macro would return a string without line breaks, using whitespace control.

4.8.4 Template Inheritance

Template inheritance works fine from state files and files. The search path starts at the root of the state tree or pillar.

4.8.5 Errors

Saltstack allows raising custom errors using the `raise` jinja function.

```
{{ raise('Custom Error') }}
```

When rendering the template containing the above statement, a `TemplateError` exception is raised, causing the rendering to fail with the following message:

```
TemplateError: Custom Error
```

4.8.6 Filters

Saltstack extends builtin filters with these custom filters:

strftime

Converts any time related object into a time based string. It requires valid strftime directives. An exhaustive list can be found here in the Python documentation.

```
{% set curtime = None | strftime() %}
```

Fuzzy dates require the timelib Python module is installed.

```
{{ "2002/12/25"|strftime("%y") }}
{{ "1040814000"|strftime("%Y-%m-%d") }}
{{ datetime|strftime("%u") }}
{{ "tomorrow"|strftime }}
```

sequence

Ensure that parsed data is a sequence.

`yaml_encode`

Serializes a single object into a YAML scalar with any necessary handling for escaping special characters. This will work for any scalar YAML data type: ints, floats, timestamps, booleans, strings, unicode. It will *not* work for multi-objects such as sequences or maps.

```
{%- set bar = 7 %}
{%- set baz = none %}
{%- set zip = true %}
{%- set zap = 'The word of the day is "salty"' %}

{%- load_yaml as foo %}
bar: {{ bar|yaml_encode }}
baz: {{ baz|yaml_encode }}
zip: {{ zip|yaml_encode }}
zap: {{ zap|yaml_encode }}
{%- endload %}
```

In the above case `{{ bar }}` and `{{ foo.bar }}` should be identical and `{{ baz }}` and `{{ foo.baz }}` should be identical.

`yaml_dquote`

Serializes a string into a properly-escaped YAML double-quoted string. This is useful when the contents of a string are unknown and may contain quotes or unicode that needs to be preserved. The resulting string will be emitted with opening and closing double quotes.

```
{%- set bar = '"The quick brown fox . . ."' %}
{%- set baz = 'The word of the day is "salty".' %}

{%- load_yaml as foo %}
bar: {{ bar|yaml_dquote }}
baz: {{ baz|yaml_dquote }}
{%- endload %}
```

In the above case `{{ bar }}` and `{{ foo.bar }}` should be identical and `{{ baz }}` and `{{ foo.baz }}` should be identical. If variable contents are not guaranteed to be a string then it is better to use `yaml_encode` which handles all YAML scalar types.

`yaml_squote`

Similar to the `yaml_dquote` filter but with single quotes. Note that YAML only allows special escapes inside double quotes so `yaml_squote` is not nearly as useful (viz. you likely want to use `yaml_encode` or `yaml_dquote`).

`to_bool`

New in version 2017.7.0.

Returns the logical value of an element.

Example:

```
{{ 'yes' | to_bool }}
{{ 'true' | to_bool }}
{{ 1 | to_bool }}
{{ 'no' | to_bool }}
```

Will be rendered as:

```
True
True
True
False
```

exactly_n_true

New in version 2017.7.0.

Tests that exactly N items in an iterable are ”truthy” (neither None, False, nor 0).

Example:

```
{{ ['yes', 0, False, 'True'] | exactly_n_true(2) }}
```

Returns:

```
True
```

exactly_one_true

New in version 2017.7.0.

Tests that exactly one item in an iterable is ”truthy” (neither None, False, nor 0).

Example:

```
{{ ['yes', False, 0, None] | exactly_one_true }}
```

Returns:

```
True
```

quote

New in version 2017.7.0.

This text will be wrapped in quotes.

regex_search

New in version 2017.7.0.

Scan through string looking for a location where this regular expression produces a match. Returns `None` in case there were no matches found

Example:

```
{{ 'abcdefabcdef' | regex_search('BC(.*)', ignorecase=True) }}
```

Returns:

```
('defabcdef',)
```

`regex_match`

New in version 2017.7.0.

If zero or more characters at the beginning of string match this regular expression, otherwise returns `None`.

Example:

```
{{ 'abcdefabcdef' | regex_match('BC(.*)', ignorecase=True) }}
```

Returns:

```
None
```

`regex_replace`

New in version 2017.7.0.

Searches for a pattern and replaces with a sequence of characters.

Example:

```
{% set my_text = 'yes, this is a TEST' %}
{{ my_text | regex_replace(' ([a-z])', '__\\1', ignorecase=True) }}
```

Returns:

```
yes,__this__is__a__TEST
```

`uuid`

New in version 2017.7.0.

Return a UUID.

Example:

```
{{ 'random' | uuid }}
```

Returns:

```
3652b285-26ad-588e-a5dc-c2ee65edc804
```

`is_list`

New in version 2017.7.0.

Return if an object is list.

Example:

```
{{ [1, 2, 3] | is_list }}
```

Returns:

```
True
```

`is_iter`

New in version 2017.7.0.

Return if an object is iterable.

Example:

```
{{ [1, 2, 3] | is_iter }}
```

Returns:

```
True
```

`min`

New in version 2017.7.0.

Return the minimum value from a list.

Example:

```
{{ [1, 2, 3] | min }}
```

Returns:

```
1
```

`max`

New in version 2017.7.0.

Returns the maximum value from a list.

Example:

```
{{ [1, 2, 3] | max }}
```

Returns:

```
3
```

`avg`

New in version 2017.7.0.

Returns the average value of the elements of a list

Example:

```
{{ [1, 2, 3] | avg }}
```

Returns:

```
2
```

`union`

New in version 2017.7.0.

Return the union of two lists.

Example:

```
{{ [1, 2, 3] | union([2, 3, 4]) | join(', ') }}
```

Returns:

```
1, 2, 3, 4
```

`intersect`

New in version 2017.7.0.

Return the intersection of two lists.

Example:

```
{{ [1, 2, 3] | intersect([2, 3, 4]) | join(', ') }}
```

Returns:

```
2, 3
```

`difference`

New in version 2017.7.0.

Return the difference of two lists.

Example:

```
{{ [1, 2, 3] | difference([2, 3, 4]) | join(', ') }}
```

Returns:

```
1
```

`symmetric_difference`

New in version 2017.7.0.

Return the symmetric difference of two lists.

Example:

```
{{ [1, 2, 3] | symmetric_difference([2, 3, 4]) | join(', ') }}
```

Returns:

```
1, 4
```

`is_sorted`

New in version 2017.7.0.

Return `True` if an iterable object is already sorted.

Example:

```
{{ [1, 2, 3] | is_sorted }}
```

Returns:

```
True
```

`compare_lists`

New in version 2017.7.0.

Compare two lists and return a dictionary with the changes.

Example:

```
{{ [1, 2, 3] | compare_lists([1, 2, 4]) }}
```

Returns:

```
{'new': [4], 'old': [3]}
```

`compare_dicts`

New in version 2017.7.0.

Compare two dictionaries and return a dictionary with the changes.

Example:

```
{{ {'a': 'b'} | compare_dicts({'a': 'c'}) }}
```

Returns:

```
{'a': {'new': 'c', 'old': 'b'}}
```

`is_hex`

New in version 2017.7.0.

Return `True` if the value is hexadecimal.

Example:

```
{{ '0xabcd' | is_hex }}
{{ 'xyzt' | is_hex }}
```

Returns:

```
True
False
```

`contains_whitespace`

New in version 2017.7.0.

Return `True` if a text contains whitespaces.

Example:

```
{{ 'abcd' | contains_whitespace }}
{{ 'ab cd' | contains_whitespace }}
```

Returns:

```
False
True
```

`substring_in_list`

New in version 2017.7.0.

Return `True` if a substring is found in a list of string values.

Example:

```
{{ 'abcd' | substring_in_list(['this', 'is', 'an abcd example']) }}
```

Returns:

```
True
```

`check_whitelist_blacklist`

New in version 2017.7.0.

Check a whitelist and/or blacklist to see if the value matches it.

This filter can be used with either a whitelist or a blacklist individually, or a whitelist and a blacklist can be passed simultaneously.

If whitelist is used alone, value membership is checked against the whitelist only. If the value is found, the function returns `True`. Otherwise, it returns `False`.

If blacklist is used alone, value membership is checked against the blacklist only. If the value is found, the function returns `False`. Otherwise, it returns `True`.

If both a whitelist and a blacklist are provided, value membership in the blacklist will be examined first. If the value is not found in the blacklist, then the whitelist is checked. If the value isn't found in the whitelist, the function returns `False`.

Whitelist Example:

```
{{ 5 | check_whitelist_blacklist(whitelist=[5, 6, 7]) }}
```

Returns:

```
True
```

Blacklist Example:

```
{{ 5 | check_whitelist_blacklist(blacklist=[5, 6, 7]) }}
```

```
False
```

date_format

New in version 2017.7.0.

Converts unix timestamp into human-readable string.

Example:

```
{{ 1457456400 | date_format }}
{{ 1457456400 | date_format('%d.%m.%Y %H:%M') }}
```

Returns:

```
2017-03-08
08.03.2017 17:00
```

to_num

New in version 2017.7.0.

New in version 2018.3.0: Renamed from `str_to_num` to `to_num`.

Converts a string to its numerical value.

Example:

```
{{ '5' | to_num }}
```

Returns:

```
5
```

`to_bytes`

New in version 2017.7.0.

Converts string-type object to bytes.

Example:

```
{{ 'wall of text' | to_bytes }}
```

Note: This option may have adverse effects when using the default renderer, `jinja|yaml`. This is due to the fact that YAML requires proper handling in regard to special characters. Please see the section on *YAML ASCII support* in the *YAML Idiosyncracies* documentation for more information.

`json_encode_list`

New in version 2017.7.0.

New in version 2018.3.0: Renamed from `json_decode_list` to `json_encode_list`. When you encode something you get bytes, and when you decode, you get your locale's encoding (usually a `unicode` type). This filter was incorrectly-named when it was added. `json_decode_list` will be supported until the Aluminium release.

Deprecated since version 2018.3.3,2019.2.0: The *`tojson`* filter accomplishes what this filter was designed to do, making this filter redundant.

Recursively encodes all string elements of the list to bytes.

Example:

```
{{ [1, 2, 3] | json_encode_list }}
```

Returns:

```
[1, 2, 3]
```

`json_encode_dict`

New in version 2017.7.0.

New in version 2018.3.0: Renamed from `json_decode_dict` to `json_encode_dict`. When you encode something you get bytes, and when you decode, you get your locale's encoding (usually a `unicode` type). This filter was incorrectly-named when it was added. `json_decode_dict` will be supported until the Aluminium release.

Deprecated since version 2018.3.3,2019.2.0: The *`tojson`* filter accomplishes what this filter was designed to do, making this filter redundant.

Recursively encodes all string items in the dictionary to bytes.

Example:

Assuming that `pillar['foo']` contains `{u'a': u'\u0414'}`, and your locale is `en_US.UTF-8`:

```
{{ pillar['foo'] | json_encode_dict }}
```

Returns:

```
{'a': '\xd0\x94'}
```

tojson

New in version 2018.3.3,2019.2.0.

Dumps a data structure to JSON.

This filter was added to provide this functionality to hosts which have a Jinja release older than version 2.9 installed. If Jinja 2.9 or newer is installed, then the upstream version of the filter will be used. See the upstream docs for more information.

random_hash

New in version 2017.7.0.

New in version 2018.3.0: Renamed from `rand_str` to `random_hash` to more accurately describe what the filter does. `rand_str` will be supported to ensure backwards compatibility but please use the preferred `random_hash`.

Generates a random number between 1 and the number passed to the filter, and then hashes it. The default hash type is the one specified by the minion's *hash_type* config option, but an alternate hash type can be passed to the filter as an argument.

Example:

```
{% set num_range = 99999999 %}
{{ num_range | random_hash }}
{{ num_range | random_hash('sha512') }}
```

Returns:

```
43ec517d68b6edd3015b3edc9a11367b
d94a45acd81f8e3107d237dbc0d5d195f6a52a0d188bc0284c0763ece1eac9f9496fb6a531a296074c87b3540398dace1222b
```

set_dict_key_value

..versionadded:: 3000

Allows you to set a value in a nested dictionary without having to worry if all the nested keys actually exist. Missing keys will be automatically created if they do not exist. The default delimiter for the keys is ':', however, with the *delimiter*-parameter, a different delimiter can be specified.

Examples:

Example 1: {%- set foo = {} %} {{ foo | set_dict_key_value('bar:baz', 42) }}

Example 2: {{ {} | set_dict_key_value('bar.baz.qux', 42, delimiter='.') }}

Returns:

Example 1: {'bar': {'baz': 42}}

Example 2: {'bar': {'baz': {'qux': 42}}}

`append_dict_key_value`

..versionadded:: 3000

Allows you to append to a list nested (deep) in a dictionary without having to worry if all the nested keys (or the list itself) actually exist. Missing keys will automatically be created if they do not exist. The default delimiter for the keys is ':', however, with the *delimiter*-parameter, a different delimiter can be specified.

Examples:

Example 1: {%- set foo = {'bar': {'baz': [1, 2]}} %} {{ foo | append_dict_key_value('bar:baz', 42) }}

Example 2: {%- set foo = {} %} {{ foo | append_dict_key_value('bar:baz:qux', 42) }}

Returns:

Example 1: {'bar': {'baz': [1, 2, 42]}}

Example 2: {'bar': {'baz': {'qux': [42]}}}

`extend_dict_key_value`

..versionadded:: 3000

Allows you to extend a list nested (deep) in a dictionary without having to worry if all the nested keys (or the list itself) actually exist. Missing keys will automatically be created if they do not exist. The default delimiter for the keys is ':', however, with the *delimiter*-parameter, a different delimiter can be specified.

Examples:

Example 1: {%- set foo = {'bar': {'baz': [1, 2]}} %} {{ foo | extend_dict_key_value('bar:baz', [42, 42]) }}

Example 2: {{ {} | extend_dict_key_value('bar:baz:qux', [42]) }}

Returns:

Example 1: {'bar': {'baz': [1, 2, 42, 42]}}

Example 2: {'bar': {'baz': {'qux': [42]}}}

`update_dict_key_value`

..versionadded:: 3000

Allows you to update a dictionary nested (deep) in another dictionary without having to worry if all the nested keys actually exist. Missing keys will automatically be created if they do not exist. The default delimiter for the keys is ':', however, with the *delimiter*-parameter, a different delimiter can be specified.

Examples:

Example 1: {%- set foo = {'bar': {'baz': {'qux': 1}}} %} {{ foo | update_dict_key_value('bar:baz', {'quux': 3}) }}

Example 2: {{ {} | update_dict_key_value('bar:baz:qux', {'quux': 3}) }}

```

```

Example 1: {'bar': {'baz': {'qux': 1, 'quux': 3}}}

Example 2: {'bar': {'baz': {'qux': {'quux': 3}}}}

md5

New in version 2017.7.0.

Return the md5 digest of a string.

Example:

```
{{ 'random' | md5 }}
```

Returns:

```
7ddf32e17a6ac5ce04a8ecbf782ca509
```

sha256

New in version 2017.7.0.

Return the sha256 digest of a string.

Example:

```
{{ 'random' | sha256 }}
```

Returns:

```
a441b15fe9a3cf56661190a0b93b9dec7d04127288cc87250967cf3b52894d11
```

sha512

New in version 2017.7.0.

Return the sha512 digest of a string.

Example:

```
{{ 'random' | sha512 }}
```

Returns:

```
811a90e1c8e86c7b4c0eef5b2c0bf0ec1b19c4b1b5a242e6455be93787cb473cb7bc9b0fdeb960d00d5c6881c2094dd63c5c
```

base64_encode

New in version 2017.7.0.

Encode a string as base64.

Example:

```
{{ 'random' | base64_encode }}
```

Returns:

```
cmFuZG9t
```

base64_decode

New in version 2017.7.0.

Decode a base64-encoded string.

```
{{ 'Z2V0IHNhbHRlZA==' | base64_decode }}
```

Returns:

```
get salted
```

hmac

New in version 2017.7.0.

Verify a challenging hmac signature against a string / shared-secret. Returns a boolean value.

Example:

```
{{ 'get salted' | hmac('shared secret', 'eBWf9bstXg+NiP5AOwppB5HMvZiYMPzEM9W5YMm/AmQ=
↪') }}
```

Returns:

```
True
```

http_query

New in version 2017.7.0.

Return the HTTP reply object from a URL.

Example:

```
{{ 'http://jsonplaceholder.typicode.com/posts/1' | http_query }}
```

Returns:

```
{
  'body': '{
    "userId": 1,
    "id": 1,
    "title": "sunt aut facere repellat provident occaecati excepturi option
↪reprehenderit",
    "body": "quia et suscipit\\nsuscipit recusandae consequuntur expedita et
↪cum\\nreprehenderit molestiae ut ut quas totam\\nnostrum rerum est autem sunt rem
↪eveniet architecto"
  }'
}
```

`traverse`

New in version 2018.3.3.

Traverse a dict or list using a colon-delimited target string. The target 'foo:bar:0' will return data['foo']['bar'][0] if this value exists, and will otherwise return the provided default value.

Example:

```
{{ {'a1': {'b1': {'c1': 'foo'}}, 'a2': 'bar'} | traverse('a1:b1', 'default') }}
```

Returns:

```
{'c1': 'foo'}
```

```
{{ {'a1': {'b1': {'c1': 'foo'}}, 'a2': 'bar'} | traverse('a2:b2', 'default') }}
```

Returns:

```
'default'
```

`json_query`

New in version 3000.

A port of Ansible `json_query` Jinja filter to make queries against JSON data using JMESPath language. Could be used to filter `pillar` data, `yaml` maps, and together with *`http_query`*. Depends on the jmespath Python module.

Examples:

```
Example 1: {{ [1, 2, 3, 4, [5, 6]] | json_query('[]') }}

Example 2: {{
{"machines": [
  {"name": "a", "state": "running"},
  {"name": "b", "state": "stopped"},
  {"name": "c", "state": "running"}
]} | json_query("machines[?state=='running'].name") }}

Example 3: {{
{"services": [
  {"name": "http", "host": "1.2.3.4", "port": 80},
```

(continues on next page)

(continued from previous page)

```
  {"name": "smtp", "host": "1.2.3.5", "port": 25},
  {"name": "ssh",  "host": "1.2.3.6", "port": 22},
]} | json_query("services[].port") }}
```

Returns:

```
Example 1: [1, 2, 3, 4, 5, 6]

Example 2: ['a', 'c']

Example 3: [80, 25, 22]
```

`to_snake_case`

New in version 3000.

Converts a string from camelCase (or CamelCase) to snake_case.

```
Example: {{ camelsWillLoveThis | to_snake_case }}
```

Returns:

```
Example: camels_will_love_this
```

`to_camelcase`

New in version 3000.

Converts a string from snake_case to camelCase (or UpperCamelCase if so indicated).

```
Example 1: {{ snake_case_for_the_win | to_camelcase }}

Example 2: {{ snake_case_for_the_win | to_camelcase(uppercamel=True) }}
```

Returns:

```
Example 1: snakeCaseForTheWin
Example 2: SnakeCaseForTheWin
```

Networking Filters

The following networking-related filters are supported:

`is_ip`

New in version 2017.7.0.

Return if a string is a valid IP Address.

```
{{ '192.168.0.1' | is_ip }}
```

Additionally accepts the following options:

- global
- link-local
- loopback
- multicast
- private
- public
- reserved
- site-local
- unspecified

Example - test if a string is a valid loopback IP address.

```
{{ '192.168.0.1' | is_ip(options='loopback') }}
```

`is_ipv4`

New in version 2017.7.0.

Returns if a string is a valid IPv4 address. Supports the same options as `is_ip`.

```
{{ '192.168.0.1' | is_ipv4 }}
```

`is_ipv6`

New in version 2017.7.0.

Returns if a string is a valid IPv6 address. Supports the same options as `is_ip`.

```
{{ 'fe80::' | is_ipv6 }}
```

`ipaddr`

New in version 2017.7.0.

From a list, returns only valid IP entries. Supports the same options as `is_ip`. The list can contains also IP interfaces/networks.

Example:

```
{{ ['192.168.0.1', 'foo', 'bar', 'fe80::'] | ipaddr }}
```

Returns:

```
['192.168.0.1', 'fe80::']
```

`ipv4`

New in version 2017.7.0.

From a list, returns only valid IPv4 entries. Supports the same options as `is_ip`. The list can contains also IP interfaces/networks.

Example:

```
{{ ['192.168.0.1', 'foo', 'bar', 'fe80::'] | ipv4 }}
```

Returns:

```
['192.168.0.1']
```

`ipv6`

New in version 2017.7.0.

From a list, returns only valid IPv6 entries. Supports the same options as `is_ip`. The list can contains also IP interfaces/networks.

Example:

```
{{ ['192.168.0.1', 'foo', 'bar', 'fe80::'] | ipv6 }}
```

Returns:

```
['fe80::']
```

`network_hosts`

New in version 2017.7.0.

Return the list of hosts within a networks. This utility works for both IPv4 and IPv6.

Note: When running this command with a large IPv6 network, the command will take a long time to gather all of the hosts.

Example:

```
{{ '192.168.0.1/30' | network_hosts }}
```

Returns:

```
['192.168.0.1', '192.168.0.2']
```

`network_size`

New in version 2017.7.0.

Return the size of the network. This utility works for both IPv4 and IPv6.

Example:

```
{{ '192.168.0.1/8' | network_size }}
```

Returns:

```
16777216
```

gen_mac

New in version 2017.7.0.

Generates a MAC address with the defined OUI prefix.

Common prefixes:

- `00:16:3E` – Xen
- `00:18:51` – OpenVZ
- `00:50:56` – VMware (manually generated)
- `52:54:00` – QEMU/KVM
- `AC:DE:48` – PRIVATE

Example:

```
{{ '00:50' | gen_mac }}
```

Returns:

```
00:50:71:52:1C
```

mac_str_to_bytes

New in version 2017.7.0.

Converts a string representing a valid MAC address to bytes.

Example:

```
{{ '00:11:22:33:44:55' | mac_str_to_bytes }}
```

Note: This option may have adverse effects when using the default renderer, `jinja|yaml`. This is due to the fact that YAML requires proper handling in regard to special characters. Please see the section on *YAML ASCII support* in the *YAML Idiosyncracies* documentation for more information.

---

dns_check

New in version 2017.7.0.

Return the ip resolved by dns, but do not exit on failure, only raise an exception. Obeys system preference for IPv4/6 address resolution.

Example:

```
{{ 'www.google.com' | dns_check(port=443) }}
```

Returns:

```
'172.217.3.196'
```

File filters

`is_text_file`

New in version 2017.7.0.

Return if a file is text.

Uses heuristics to guess whether the given file is text or binary, by reading a single block of bytes from the file. If more than 30% of the chars in the block are non-text, or there are NUL ('x00') bytes in the block, assume this is a binary file.

Example:

```
{{ '/etc/salt/master' | is_text_file }}
```

Returns:

```
True
```

`is_binary_file`

New in version 2017.7.0.

Return if a file is binary.

Detects if the file is a binary, returns bool. Returns True if the file is a bin, False if the file is not and None if the file is not available.

Example:

```
{{ '/etc/salt/master' | is_binary_file }}
```

Returns:

```
False
```

`is_empty_file`

New in version 2017.7.0.

Return if a file is empty.

Example:

```
{{ '/etc/salt/master' | is_empty_file }}
```

Returns:

```
False
```

`file_hashsum`

New in version 2017.7.0.

Return the hashsum of a file.

Example:

```
{{ '/etc/salt/master' | file_hashsum }}
```

Returns:

```
02d4ef135514934759634f10079653252c7ad594ea97bd385480c532bca0fdda
```

`list_files`

New in version 2017.7.0.

Return a recursive list of files under a specific path.

Example:

```
{{ '/etc/salt/' | list_files | join('\n') }}
```

Returns:

```
/etc/salt/master
/etc/salt/proxy
/etc/salt/minion
/etc/salt/pillar/top.sls
/etc/salt/pillar/device1.sls
```

`path_join`

New in version 2017.7.0.

Joins absolute paths.

Example:

```
{{ '/etc/salt/' | path_join('pillar', 'device1.sls') }}
```

Returns:

```
/etc/salt/pillar/device1.sls
```

`which`

New in version 2017.7.0.

Python clone of /usr/bin/which.

Example:

```
{{ 'salt-master' | which }}
```

Returns:

```
/usr/local/salt/virtualenv/bin/salt-master
```

4.8.7 Tests

Saltstack extends builtin tests with these custom tests:

`equalto`

Tests the equality between two values.

Can be used in an `if` statement directly:

```
{% if 1 is equalto(1) %}
    < statements >
{% endif %}
```

If clause evaluates to `True`

or with the `selectattr` filter:

```
{{ [{'value': 1}, {'value': 2} , {'value': 3}] | selectattr('value', 'equalto', 3) |
↪list }}
```

Returns:

```
[{'value': 3}]
```

`match`

Tests that a string matches the regex passed as an argument.

Can be used in a `if` statement directly:

```
{% if 'a' is match('[a-b]') %}
    < statements >
{% endif %}
```

If clause evaluates to `True`

or with the `selectattr` filter:

```
{{ [{'value': 'a'}, {'value': 'b'}, {'value': 'c'}] | selectattr('value', 'match',
↪'[b-e]') | list }}
```

Returns:

```
[{'value': 'b'}, {'value': 'c'}]
```

Test supports additional optional arguments: `ignorecase`, `multiline`

Escape filters

`regex_escape`

New in version 2017.7.0.

Allows escaping of strings so they can be interpreted literally by another function.

Example:

```
regex_escape = {{ 'https://example.com?foo=bar%20baz' | regex_escape }}
```

will be rendered as:

```
regex_escape = https\:\/\/example\.com\?foo\=bar\%20baz
```

Set Theory Filters

`unique`

New in version 2017.7.0.

Performs set math using Jinja filters.

Example:

```
unique = {{ ['foo', 'foo', 'bar'] | unique }}
```

will be rendered as:

```
unique = ['foo', 'bar']
```

4.8.8 Jinja in Files

Jinja can be used in the same way in managed files:

```
# redis.sls
/etc/redis/redis.conf:
    file.managed:
        - source: salt://redis.conf
        - template: jinja
        - context:
          bind: 127.0.0.1
```

```
# lib.sls
{% set port = 6379 %}
```

```
# redis.conf
{% from 'lib.sls' import port with context %}
port {{ port }}
bind {{ bind }}
```

As an example, configuration was pulled from the file context and from an external template file.

Note: Macros and variables can be shared across templates. They should not be starting with one or more underscores, and should be managed by one of the following tags: *macro*, *set*, *load_yaml*, *load_json*, *import_yaml* and *import_json*.

4.8.9 Escaping Jinja

Occasionally, it may be necessary to escape Jinja syntax. There are two ways to do this in Jinja. One is escaping individual variables or strings and the other is to escape entire blocks.

To escape a string commonly used in Jinja syntax such as `{{`, you can use the following syntax:

```
{{ '{{' }}
```

For larger blocks that contain Jinja syntax that needs to be escaped, you can use raw blocks:

```
{% raw %}
    some text that contains jinja characters that need to be escaped
{% endraw %}
```

See the Escaping section of Jinja's documentation to learn more.

A real-word example of needing to use raw tags to escape a larger block of code is when using `file.managed` with the `contents_pillar` option to manage files that contain something like consul-template, which shares a syntax subset with Jinja. Raw blocks are necessary here because the Jinja in the pillar would be rendered before the file.managed is ever called, so the Jinja syntax must be escaped:

```
{% raw %}
- contents_pillar: |
    job "example-job" {
      <snipped>
      task "example" {
          driver = "docker"

          config {
              image = "docker-registry.service.consul:5000/example-job:{{key "nomad/
↪jobs/example-job/version"}}"
      <snipped>
{% endraw %}
```

4.8.10 Calling Salt Functions

The Jinja renderer provides a shorthand lookup syntax for the `salt` dictionary of *execution function*.

New in version 2014.7.0.

```
# The following two function calls are equivalent.
{{ salt['cmd.run']('whoami') }}
{{ salt.cmd.run('whoami') }}
```

4.8.11 Debugging

The `show_full_context` function can be used to output all variables present in the current Jinja context.

New in version 2014.7.0.

```
Context is: {{ show_full_context()|yaml(False) }}
```

Logs

New in version 2017.7.0.

Yes, in Salt, one is able to debug a complex Jinja template using the logs. For example, making the call:

```
{%- do salt.log.error('testing jinja logging') -%}
```

Will insert the following message in the minion logs:

```
2017-02-01 01:24:40,728 [salt.module.logmod][ERROR   ][3779] testing jinja logging
```

4.8.12 Python Methods

A powerful feature of jinja that is only hinted at in the official jinja documentation is that you can use the native python methods of the variable type. Here is the python documentation for string methods.

```
{% set hostname,domain = grains.id.partition('.')[::2] %}{{ hostname }}
```

```
{% set strings = grains.id.split('-') %}{{ strings[0] }}
```

4.8.13 Custom Execution Modules

Custom execution modules can be used to supplement or replace complex Jinja. Many tasks that require complex looping and logic are trivial when using Python in a Salt execution module. Salt execution modules are easy to write and distribute to Salt minions.

Functions in custom execution modules are available in the Salt execution module dictionary just like the built-in execution modules:

```
{{ salt['my_custom_module.my_custom_function']() }}
```

- *How to Convert Jinja Logic to an Execution Module*
- *Writing Execution Modules*

4.8.14 Custom Jinja filters

Given that all execution modules are available in the Jinja template, one can easily define a custom module as in the previous paragraph and use it as a Jinja filter. However, please note that it will not be accessible through the pipe.

For example, instead of:

```
{{ my_variable | my_jinja_filter }}
```

The user will need to define `my_jinja_filter` function under an extension module, say `my_filters` and use as:

```
{{ salt.my_filters.my_jinja_filter(my_variable) }}
```

The greatest benefit is that you are able to access thousands of existing functions, e.g.:

- get the DNS AAAA records for a specific address using the *dnsutil*:

  ```
  {{ salt.dnsutil.AAAA('www.google.com') }}
  ```

- retrieve a specific field value from a `Redis` hash:

  ```
  {{ salt.redis.hget('foo_hash', 'bar_field') }}
  ```

- get the routes to `0.0.0.0/0` using the *NAPALM route*:

  ```
  {{ salt.route.show('0.0.0.0/0') }}
  ```

4.9 Tutorials Index

4.9.1 Autoaccept minions from Grains

New in version 2018.3.0.

To automatically accept minions based on certain characteristics, e.g. the `uuid` you can specify certain grain values on the salt master. Minions with matching grains will have their keys automatically accepted.

1. Configure the autosign_grains_dir in the master config file:

```
autosign_grains_dir: /etc/salt/autosign_grains
```

2. Configure the grain values to be accepted

Place a file named like the grain in the autosign_grains_dir and write the values that should be accepted automatically inside that file. For example to automatically accept minions based on their `uuid` create a file named `/etc/salt/autosign_grains/uuid`:

```
8f7d68e2-30c5-40c6-b84a-df7e978a03ee
1d3c5473-1fbc-479e-b0c7-877705a0730f
```

The master is now setup to accept minions with either of the two specified uuids. Multiple values must always be written into separate lines. Lines starting with a # are ignored.

3. Configure the minion to send the specific grains to the master in the minion config file:

```
autosign_grains:
  - uuid
```

Now you should be able to start salt-minion and run `salt-call state.apply` or any other salt commands that require master authentication.

4.9.2 Salt as a Cloud Controller

In Salt 0.14.0, an advanced cloud control system were introduced, allow private cloud vms to be managed directly with Salt. This system is generally referred to as **Salt Virt**.

The Salt Virt system already exists and is installed within Salt itself, this means that besides setting up Salt, no additional salt code needs to be deployed.

Note: The `libvirt` python module and the `certtool` binary are required.

The main goal of Salt Virt is to facilitate a very fast and simple cloud. The cloud that can scale and is fully featured. Salt Virt comes with the ability to set up and manage complex virtual machine networking, powerful image and disk management, as well as virtual machine migration with and without shared storage.

This means that Salt Virt can be used to create a cloud from a blade center and a SAN, but can also create a cloud out of a swarm of Linux Desktops without a single shared storage system. Salt Virt can make clouds from truly commodity hardware, but can also stand up the power of specialized hardware as well.

Setting up Hypervisors

The first step to set up the hypervisors involves getting the correct software installed and setting up the hypervisor network interfaces.

Installing Hypervisor Software

Salt Virt is made to be hypervisor agnostic but currently the only fully implemented hypervisor is KVM via libvirt.

The required software for a hypervisor is libvirt and kvm. For advanced features install libguestfs or qemu-nbd.

Note: Libguestfs and qemu-nbd allow for virtual machine images to be mounted before startup and get pre-seeded with configurations and a salt minion

---

This sls will set up the needed software for a hypervisor, and run the routines to set up the libvirt pki keys.

Note: Package names and setup used is Red Hat specific, different package names will be required for different platforms

---

```yaml
libvirt:
  pkg.installed: []
  file.managed:
    - name: /etc/sysconfig/libvirtd
    - contents: 'LIBVIRTD_ARGS="--listen"'
    - require:
      - pkg: libvirt
  virt.keys:
    - require:
      - pkg: libvirt
  service.running:
    - name: libvirtd
    - require:
      - pkg: libvirt
      - network: br0
      - libvirt: libvirt
    - watch:
      - file: libvirt
```

(continues on next page)

(continued from previous page)

```
libvirt-python:
  pkg.installed: []

libguestfs:
  pkg.installed:
    - pkgs:
      - libguestfs
      - libguestfs-tools
```

Hypervisor Network Setup

The hypervisors will need to be running a network bridge to serve up network devices for virtual machines, this formula will set up a standard bridge on a hypervisor connecting the bridge to eth0:

```
eth0:
  network.managed:
    - enabled: True
    - type: eth
    - bridge: br0

br0:
  network.managed:
    - enabled: True
    - type: bridge
    - proto: dhcp
    - require:
      - network: eth0
```

Virtual Machine Network Setup

Salt Virt comes with a system to model the network interfaces used by the deployed virtual machines; by default a single interface is created for the deployed virtual machine and is bridged to `br0`. To get going with the default networking setup, ensure that the bridge interface named `br0` exists on the hypervisor and is bridged to an active network device.

Note: To use more advanced networking in Salt Virt, read the *Salt Virt Networking* document:

Salt Virt Networking

Libvirt State

One of the challenges of deploying a libvirt based cloud is the distribution of libvirt certificates. These certificates allow for virtual machine migration. Salt comes with a system used to auto deploy these certificates. Salt manages the signing authority key and generates keys for libvirt clients on the master, signs them with the certificate authority and uses pillar to distribute them. This is managed via the `libvirt` state. Simply execute this formula on the minion to ensure that the certificate is in place and up to date:

Note: The above formula includes the calls needed to set up libvirt keys.

```
libvirt_keys:
  virt.keys
```

Getting Virtual Machine Images Ready

Salt Virt, requires that virtual machine images be provided as these are not generated on the fly. Generating these virtual machine images differs greatly based on the underlying platform.

Virtual machine images can be manually created using KVM and running through the installer, but this process is not recommended since it is very manual and prone to errors.

Virtual Machine generation applications are available for many platforms:

kiwi: (openSUSE, SLES, RHEL, CentOS) https://suse.github.io/kiwi/

vm-builder: https://wiki.debian.org/VMBuilder

> **See also:**
>
> vmbuilder-formula

Once virtual machine images are available, the easiest way to make them available to Salt Virt is to place them in the Salt file server. Just copy an image into `/srv/salt` and it can now be used by Salt Virt.

For purposes of this demo, the file name `centos.img` will be used.

Existing Virtual Machine Images

Many existing Linux distributions distribute virtual machine images which can be used with Salt Virt. Please be advised that NONE OF THESE IMAGES ARE SUPPORTED BY SALTSTACK.

CentOS

These images have been prepared for OpenNebula but should work without issue with Salt Virt, only the raw qcow image file is needed: http://wiki.centos.org/Cloud/OpenNebula

Fedora Linux

Images for Fedora Linux can be found here: https://alt.fedoraproject.org/cloud

openSUSE

https://download.opensuse.org/distribution/leap/15.1/jeos/openSUSE-Leap-15.1-JeOS.x86_64-15.1.0-kvm-and-xen-Current.qcow2.meta4

SUSE

https://www.suse.com/products/server/jeos

Ubuntu Linux

Images for Ubuntu Linux can be found here: http://cloud-images.ubuntu.com/

Using Salt Virt

With hypervisors set up and virtual machine images ready, Salt can start issuing cloud commands using the *virt runner*.

Start by running a Salt Virt hypervisor info command:

```
salt-run virt.host_info
```

This will query the running hypervisor(s) for stats and display useful information such as the number of cpus and amount of memory.

You can also list all VMs and their current states on all hypervisor nodes:

```
salt-run virt.list
```

Now that hypervisors are available a virtual machine can be provisioned. The `virt.init` routine will create a new virtual machine:

```
salt-run virt.init centos1 2 512 salt://centos.img
```

The Salt Virt runner will now automatically select a hypervisor to deploy the new virtual machine on. Using `salt://` assumes that the CentOS virtual machine image is located in the root of the *Salt File Server* on the master. When images are cloned (i.e. copied locatlly after retrieval from the file server) the destination directory on the hypervisor minion is determined by the `virt:images` config option; by default this is `/srv/salt-images/`.

When a VM is initialized using `virt.init` the image is copied to the hypervisor using `cp.cache_file` and will be mounted and seeded with a minion. Seeding includes setting pre-authenticated keys on the new machine. A minion will only be installed if one can not be found on the image using the default arguments to `seed.apply`.

Note: The biggest bottleneck in starting VMs is when the Salt Minion needs to be installed. Making sure that the source VM images already have Salt installed will GREATLY speed up virtual machine deployment.

You can also deploy an image on a particular minion by directly calling the *virt* execution module with an absolute image path. This can be quite handy for testing:

```
salt 'hypervisor*' virt.init centos1 2 512 image=/var/lib/libvirt/images/centos.img
```

Now that the new VM has been prepared, it can be seen via the `virt.query` command:

```
salt-run virt.query
```

This command will return data about all of the hypervisors and respective virtual machines.

Now that the new VM is booted it should have contacted the Salt Master, a `test.version` will reveal if the new VM is running.

QEMU copy on write support

For fast image cloning you can use the qcow disk image format. Pass the `enable_qcow` flag and a *.qcow2* image path to *virt.init*:

```
salt 'hypervisor*' virt.init centos1 2 512 image=/var/lib/libvirt/images/centos.qcow2␣
↪enable_qcow=True start=False
```

Note: Beware that attempting to boot a qcow image too quickly after cloning can result in a race condition where libvirt may try to boot the machine before image seeding has completed. For that reason it is recommended to also pass `start=False` to `virt.init`.

Also know that you **must not** modify the original base image without first making a copy and then *rebasing* all overlay images onto it. See the `qemu-img rebase` usage docs.

Migrating Virtual Machines

Salt Virt comes with full support for virtual machine migration, and using the libvirt state in the above formula makes migration possible.

A few things need to be available to support migration. Many operating systems turn on firewalls when originally set up, the firewall needs to be opened up to allow for libvirt and kvm to cross communicate and execution migration routines. On Red Hat based hypervisors in particular port 16514 needs to be opened on hypervisors:

```
iptables -A INPUT -m state --state NEW -m tcp -p tcp --dport 16514 -j ACCEPT
```

Note: More in-depth information regarding distribution specific firewall settings can read in:

Opening the Firewall up for Salt

Salt also needs the `virt:tunnel` option to be turned on. This flag tells Salt to run migrations securely via the libvirt TLS tunnel and to use port 16514. Without `virt:tunnel` libvirt tries to bind to random ports when running migrations.

To turn on `virt:tunnel` simply apply it to the master config file:

```
virt:
    tunnel: True
```

Once the master config has been updated, restart the master and send out a call to the minions to refresh the pillar to pick up on the change:

```
salt \* saltutil.refresh_modules
```

Now, migration routines can be run! To migrate a VM, simply run the Salt Virt migrate routine:

```
salt-run virt.migrate centos <new hypervisor>
```

VNC Consoles

Although not enabled by default, Salt Virt can also set up VNC consoles allowing for remote visual consoles to be opened up. When creating a new VM using `virt.init` pass the `enable_vnc=True` parameter to have a console configured for the new VM.

The information from a `virt.query` routine will display the vnc console port for the specific vms:

```
centos
  CPU: 2
  Memory: 524288
  State: running
  Graphics: vnc - hyper6:5900
  Disk - vda:
    Size: 2.0G
    File: /srv/salt-images/ubuntu2/system.qcow2
    File Format: qcow2
  Nic - ac:de:48:98:08:77:
    Source: br0
    Type: bridge
```

The line *Graphics: vnc - hyper6:5900* holds the key. First the port named, in this case 5900, will need to be available in the hypervisor's firewall. Once the port is open, then the console can be easily opened via vncviewer:

```
vncviewer hyper6:5900
```

By default there is no VNC security set up on these ports, which suggests that keeping them firewalled and mandating that SSH tunnels be used to access these VNC interfaces. Keep in mind that activity on a VNC interface that is accessed can be viewed by any other user that accesses that same VNC interface, and any other user logging in can also operate with the logged in user on the virtual machine.

Conclusion

Now with Salt Virt running, new hypervisors can be seamlessly added just by running the above states on new bare metal machines, and these machines will be instantly available to Salt Virt.

4.9.3 Running Salt States and Commands in Docker Containers

The 2016.11.0 release of Salt introduces the ability to execute Salt States and Salt remote execution commands directly inside of Docker containers.

This addition makes it possible to not only deploy fresh containers using Salt States. This also allows for running containers to be audited and modified using Salt, but without running a Salt Minion inside the container. Some of the applications include security audits of running containers as well as gathering operating data from containers.

This new feature is simple and straightforward, and can be used via a running Salt Minion, the Salt Call command, or via Salt SSH. For this tutorial we will use the *salt-call* command, but like all salt commands these calls are directly translatable to *salt* and *salt-ssh*.

Step 1 - Install Docker

Since setting up Docker is well covered in the Docker documentation we will make no such effort to describe it here. Please see the Docker Installation Documentation for installing and setting up Docker: https://docs.docker.com/engine/installation/

The Docker integration also requires that the *docker-py* library is installed. This can easily be done using pip or via your system package manager:

```
pip install docker-py
```

Step 2 - Install Salt

For this tutorial we will be using Salt Call, which is available in the *salt-minion* package, please follow the Salt Installation docs found here: https://repo.saltstack.com/

Step 3 - Create With Salt States

Next some Salt States are needed, for this example a very basic state which installs *vim* is used, but anything Salt States can do can be done here, please see the Salt States Introduction Tutorial to learn more about Salt States: https://docs.saltstack.com/en/stage/getstarted/config/

For this tutorial, simply create a small state file in */srv/salt/vim.sls*:

```
vim:
  pkg.installed
```

Note: The base image you choose will need to have python 2.6 or 2.7 installed. We are hoping to resolve this constraint in a future release.

If *base* is omitted the default image used is a minimal openSUSE image with Python support, maintained by SUSE

Next run the *docker.sls_build* command:

```
salt-call --local dockerng.sls_build test base=my_base_image mods=vim
```

Now we have a fresh image called *test* to work with and vim has been installed.

Step 4 - Running Commands Inside the Container

Salt can now run remote execution functions inside the container with another simple *salt-call* command:

```
salt-call --local dockerng.call test test.version
salt-call --local dockerng.call test network.interfaces
salt-call --local dockerng.call test disk.usage
salt-call --local dockerng.call test pkg.list_pkgs
salt-call --local dockerng.call test service.running httpd
salt-call --local dockerng.call test cmd.run 'ls -l /etc'
```

4.9.4 Automatic Updates / Frozen Deployments

New in version 0.10.3.d.

Salt has support for the Esky application freezing and update tool. This tool allows one to build a complete zipfile out of the salt scripts and all their dependencies - including shared objects / DLLs.

Getting Started

To build frozen applications, suitable build environment will be needed for each platform. You should probably set up a virtualenv in order to limit the scope of Q/A.

This process does work on Windows. Directions are available at https://github.com/saltstack/salt-windows-install for details on installing Salt in Windows. Only the 32-bit Python and dependencies have been tested, but they have been tested on 64-bit Windows.

Install `bbfreeze`, and then `esky` from PyPI in order to enable the `bdist_esky` command in `setup.py`. Salt itself must also be installed, in addition to its dependencies.

Building and Freezing

Once you have your tools installed and the environment configured, use `setup.py` to prepare the distribution files.

```
python setup.py sdist
python setup.py bdist
```

Once the distribution files are in place, Esky can be used traverse the module tree and pack all the scripts up into a redistributable.

```
python setup.py bdist_esky
```

There will be an appropriately versioned `salt-VERSION.zip` in `dist/` if everything went smoothly.

Windows

`C:\Python27\lib\site-packages\zmq` will need to be added to the PATH variable. This helps bbfreeze find the zmq DLL so it can pack it up.

Using the Frozen Build

Unpack the zip file in the desired install location. Scripts like `salt-minion` and `salt-call` will be in the root of the zip file. The associated libraries and bootstrapping will be in the directories at the same level. (Check the Esky documentation for more information)

To support updating your minions in the wild, put the builds on a web server that the minions can reach. *`salt.modules.saltutil.update()`* will trigger an update and (optionally) a restart of the minion service under the new version.

Troubleshooting

A Windows minion isn't responding

The process dispatch on Windows is slower than it is on *nix. It may be necessary to add '-t 15' to salt commands to give minions plenty of time to return.

Windows and the Visual Studio Redist

The Visual C++ 2008 32-bit redistributable will need to be installed on all Windows minions. Esky has an option to pack the library into the zipfile, but OpenSSL does not seem to acknowledge the new location. If a `no OPENSSL_Applink` error appears on the console when trying to start a frozen minion, the redistributable is not installed.

Mixed Linux environments and Yum

The Yum Python module doesn't appear to be available on any of the standard Python package mirrors. If RHEL/CentOS systems need to be supported, the frozen build should created on that platform to support all the Linux nodes. Remember to build the virtualenv with `--system-site-packages` so that the `yum` module is included.

Automatic (Python) module discovery

Automatic (Python) module discovery does not work with the late-loaded scheme that Salt uses for (Salt) modules. Any misbehaving modules will need to be explicitly added to the `freezer_includes` in Salt's `setup.py`. Always check the zipped application to make sure that the necessary modules were included.

4.9.5 ESXi Proxy Minion

New in version 2015.8.4.

Note: This tutorial assumes basic knowledge of Salt. To get up to speed, check out the *Salt Walkthrough*.

This tutorial also assumes a basic understanding of Salt Proxy Minions. If you're unfamiliar with Salt's Proxy Minion system, please read the *Salt Proxy Minion* documentation and the *Salt Proxy Minion End-to-End Example* tutorial.

The third assumption that this tutorial makes is that you also have a basic understanding of ESXi hosts. You can learn more about ESXi hosts on VMware's various resources.

Salt's ESXi Proxy Minion allows a VMware ESXi host to be treated as an individual Salt Minion, without installing a Salt Minion on the ESXi host.

Since an ESXi host may not necessarily run on an OS capable of hosting a Python stack, the ESXi host can't run a regular Salt Minion directly. Therefore, Salt's Proxy Minion functionality enables you to designate another machine to host a proxy process that "proxies" communication from the Salt Master to the ESXi host. The master does not know or care that the ESXi target is not a "real" Salt Minion.

More in-depth conceptual reading on Proxy Minions can be found in the *Proxy Minion* section of Salt's documentation.

Salt's ESXi Proxy Minion was added in the 2015.8.4 release of Salt.

Note: Be aware that some functionality for the ESXi Proxy Minion may depend on the type of license attached the ESXi host(s).

For example, certain services are only available to manipulate service state or policies with a VMware vSphere Enterprise or Enterprise Plus license, while others are available with a Standard license. The `ntpd` service is restricted to an Enterprise Plus license, while `ssh` is available via the Standard license.

Please see the vSphere Comparison page for more information.

Dependencies

Manipulation of the ESXi host via a Proxy Minion requires the machine running the Proxy Minion process to have the ESXCLI package (and all of it's dependencies) and the pyVmomi Python Library to be installed.

ESXi Password

The ESXi Proxy Minion uses VMware's API to perform tasks on the host as if it was a regular Salt Minion. In order to access the API that is already running on the ESXi host, the ESXi host must have a username and password that is used to log into the host. The username is usually `root`. Before Salt can access the ESXi host via VMware's API, a default password *must* be set on the host.

pyVmomi

The pyVmomi Python library must be installed on the machine that is running the proxy process. pyVmomi can be installed via pip:

```
pip install pyVmomi
```

Note: Version 6.0 of pyVmomi has some problems with SSL error handling on certain versions of Python. If using version 6.0 of pyVmomi, the machine that you are running the proxy minion process from must have either Python 2.6, Python 2.7.9, or newer. This is due to an upstream dependency in pyVmomi 6.0 that is not supported in Python version 2.7 to 2.7.8. If the version of Python running the proxy process is not in the supported range, you will need to install an earlier version of pyVmomi. See Issue #29537 for more information.

Based on the note above, to install an earlier version of pyVmomi than the version currently listed in PyPi, run the following:

```
pip install pyVmomi==5.5.0.2014.1.1
```

The 5.5.0.2014.1.1 is a known stable version that the original ESXi Proxy Minion was developed against.

ESXCLI

Currently, about a third of the functions used for the ESXi Proxy Minion require the ESXCLI package be installed on the machine running the Proxy Minion process.

The ESXCLI package is also referred to as the VMware vSphere CLI, or vCLI. VMware provides vCLI package installation instructions for vSphere 5.5 and vSphere 6.0.

Once all of the required dependencies are in place and the vCLI package is installed, you can check to see if you can connect to your ESXi host by running the following command:

```
esxcli -s <host-location> -u <username> -p <password> system syslog config get
```

If the connection was successful, ESXCLI was successfully installed on your system. You should see output related to the ESXi host's syslog configuration.

Configuration

There are several places where various configuration values need to be set in order for the ESXi Proxy Minion to run and connect properly.

Proxy Config File

On the machine that will be running the Proxy Minon process(es), a proxy config file must be in place. This file should be located in the `/etc/salt/` directory and should be named `proxy`. If the file is not there by default, create it.

This file should contain the location of your Salt Master that the Salt Proxy will connect to.

Example Proxy Config File:

```
# /etc/salt/proxy

master: <salt-master-location>
```

Pillar Profiles

Proxy minions get their configuration from Salt's Pillar. Every proxy must have a stanza in Pillar and a reference in the Pillar top-file that matches the Proxy ID. At a minimum for communication with the ESXi host, the pillar should look like this:

```
proxy:
  proxytype: esxi
  host: <ip or dns name of esxi host>
  username: <ESXi username>
  passwords:
    - first_password
    - second_password
    - third_password
```

Some other optional settings are `protocol` and `port`. These can be added to the pillar configuration.

proxytype

The `proxytype` key and value pair is critical, as it tells Salt which interface to load from the `proxy` directory in Salt's install hierarchy, or from `/srv/salt/_proxy` on the Salt Master (if you have created your own proxy module, for example). To use this ESXi Proxy Module, set this to `esxi`.

host

The location, or ip/dns, of the ESXi host. Required.

username

The username used to login to the ESXi host, such as `root`. Required.

passwords

A list of passwords to be used to try and login to the ESXi host. At least one password in this list is required.

The proxy integration will try the passwords listed in order. It is configured this way so you can have a regular password and the password you may be updating for an ESXi host either via the *`vsphere.update_host_password`* execution module function or via the *esxi.password_present* state function. This way, after the password is changed, you should not need to restart the proxy minion–it should just pick up the new password provided in the list. You can then change pillar at will to move that password to the front and retire the unused ones.

Use-case/reasoning for using a list of passwords: You are setting up an ESXi host for the first time, and the host comes with a default password. You know that you'll be changing this password during your initial setup from the default to a new password. If you only have one password option, and if you have a state changing the password, any remote execution commands or states that run after the password change will not be able to run on the host until the password is updated in Pillar and the Proxy Minion process is restarted.

This allows you to use any number of potential fallback passwords.

Note: When a password is changed on the host to one in the list of possible passwords, the further down on the list the password is, the longer individual commands will take to return. This is due to the nature of pyVmomi's login system. We have to wait for the first attempt to fail before trying the next password on the list.

This scenario is especially true, and even slower, when the proxy minion first starts. If the correct password is not the first password on the list, it may take up to a minute for `test.version` to respond with salt's version installed (Example: `2018.3.4`. Once the initial authorization is complete, the responses for commands will be a little faster.

To avoid these longer waiting periods, SaltStack recommends moving the correct password to the top of the list and restarting the proxy minion at your earliest convenience.

---

protocol

If the ESXi host is not using the default protocol, set this value to an alternate protocol. Default is `https`. For example:

port

If the ESXi host is not using the default port, set this value to an alternate port. Default is `443`.

Example Configuration Files

An example of all of the basic configurations that need to be in place before starting the Proxy Minion processes includes the Proxy Config File, Pillar Top File, and any individual Proxy Minion Pillar files.

In this example, we'll assuming there are two ESXi hosts to connect to. Therefore, we'll be creating two Proxy Minion config files, one config for each ESXi host.

Proxy Config File:

```
# /etc/salt/proxy

master: <salt-master-location>
```

Pillar Top File:

```
# /srv/pillar/top.sls

base:
  'esxi-1':
    - esxi-1
```

(continues on next page)

(continued from previous page)

```
  'esxi-2':
    - esxi-2
```

Pillar Config File for the first ESXi host, esxi-1:

```
# /srv/pillar/esxi-1.sls

proxy:
  proxytype: esxi
  host: esxi-1.example.com
  username: 'root'
  passwords:
    - bad-password-1
    - backup-bad-password-1
```

Pillar Config File for the second ESXi host, esxi-2:

```
# /srv/pillar/esxi-2.sls

proxy:
  proxytype: esxi
  host: esxi-2.example.com
  username: 'root'
  passwords:
    - bad-password-2
    - backup-bad-password-2
```

Starting the Proxy Minion

Once all of the correct configuration files are in place, it is time to start the proxy processes!

1. First, make sure your Salt Master is running.
2. Start the first Salt Proxy, in debug mode, by giving the Proxy Minion process and ID that matches the config file name created in the *Configuration* section.

```
salt-proxy --proxyid='esxi-1' -l debug
```

1. Accept the `esxi-1` Proxy Minion's key on the Salt Master:

```
# salt-key -L
Accepted Keys:
Denied Keys:
Unaccepted Keys:
esxi-1
Rejected Keys:
#
# salt-key -a esxi-1
The following keys are going to be accepted:
Unaccepted Keys:
esxi-1
Proceed? [n/Y] y
Key for minion esxi-1 accepted.
```

1. Repeat for the second Salt Proxy, this time we'll run the proxy process as a daemon, as an example.

```
salt-proxy --proxyid='esxi-2' -d
```

1. Accept the `esxi-2` Proxy Minion's key on the Salt Master:

```
# salt-key -L
Accepted Keys:
esxi-1
Denied Keys:
Unaccepted Keys:
esxi-2
Rejected Keys:
#
# salt-key -a esxi-1
The following keys are going to be accepted:
Unaccepted Keys:
esxi-2
Proceed? [n/Y] y
Key for minion esxi-1 accepted.
```

1. Check and see if your Proxy Minions are responding:

```
# salt 'esxi-*' test.version
esxi-1:
    True
esxi-3:
    True
```

Executing Commands

Now that you've configured your Proxy Minions and have them responding successfully to a `test.version`, we can start executing commands against the ESXi hosts via Salt.

It's important to understand how this particular proxy works, and there are a couple of important pieces to be aware of in order to start running remote execution and state commands against the ESXi host via a Proxy Minion: the *vSphere Execution Module*, the *ESXi Execution Module*, and the *ESXi State Module*.

vSphere Execution Module

The *`Salt.modules.vsphere`* is a standard Salt execution module that does the bulk of the work for the ESXi Proxy Minion. If you pull up the docs for it you'll see that almost every function in the module takes credentials (`username` and `password`) and a target `host` argument. When credentials and a host aren't passed, Salt runs commands through `pyVmomi` or `ESXCLI` against the local machine. If you wanted, you could run functions from this module on any machine where an appropriate version of `pyVmomi` and `ESXCLI` are installed, and that machine would reach out over the network and communicate with the ESXi host.

You'll notice that most of the functions in the vSphere module require a `host`, `username`, and `password`. These parameters are contained in the Pillar files and passed through to the function via the proxy process that is already running. You don't need to provide these parameters when you execute the commands. See the *Running Remote Execution Commands* section below for an example.

ESXi Execution Module

In order for the Pillar information set up in the *Configuration* section above to be passed to the function call in the vSphere Execution Module, the *`salt.modules.esxi`* execution module acts as a "shim" between the vSphere execution module functions and the proxy process.

The "shim" takes the authentication credentials specified in the Pillar files and passes them through to the `host`, `username`, `password`, and optional `protocol` and `port` options required by the vSphere Execution Module functions.

If the function takes more positional, or keyword, arguments you can append them to the call. It's this shim that speaks to the ESXi host through the proxy, arranging for the credentials and hostname to be pulled from the Pillar section for the ESXi Proxy Minion.

Because of the presence of the shim, to lookup documentation for what functions you can use to interface with the ESXi host, you'll want to look in `salt.modules.vsphere` instead of `salt.modules.esxi`.

Running Remote Execution Commands

To run commands from the Salt Master to execute, via the ESXi Proxy Minion, against the ESXi host, you use the `esxi.cmd <vsphere-function-name>` syntax to call functions located in the vSphere Execution Module. Both args and kwargs needed for various vsphere execution module functions must be passed through in a kwarg-type manor. For example:

```
salt 'esxi-*' esxi.cmd system_info
salt 'exsi-*' esxi.cmd get_service_running service_name='ssh'
```

ESXi State Module

The ESXi State Module functions similarly to other state modules. The "shim" provided by the *ESXi Execution Module* passes the necessary `host`, `username`, and `password` credentials through, so those options don't need to be provided in the state. Other than that, state files are written and executed just like any other Salt state. See the `salt.modules.esxi` state for ESXi state functions.

The follow state file is an example of how to configure various pieces of an ESXi host including enabling SSH, uploading and SSH key, configuring a coredump network config, syslog, ntp, enabling VMotion, resetting a host password, and more.

```
# /srv/salt/configure-esxi.sls

configure-host-ssh:
  esxi.ssh_configured:
    - service_running: True
    - ssh_key_file: /etc/salt/ssh_keys/my_key.pub
    - service_policy: 'automatic'
    - service_restart: True
    - certificate_verify: True

configure-host-coredump:
  esxi.coredump_configured:
    - enabled: True
    - dump_ip: 'my-coredump-ip.example.com'

configure-host-syslog:
  esxi.syslog_configured:
    - syslog_configs:
        loghost: ssl://localhost:5432,tcp://10.1.0.1:1514
        default-timeout: 120
    - firewall: True
```

(continues on next page)

(continued from previous page)

```
    - reset_service: True
    - reset_syslog_config: True
    - reset_configs: loghost,default-timeout

configure-host-ntp:
  esxi.ntp_configured:
    - service_running: True
    - ntp_servers:
      - 192.174.1.100
      - 192.174.1.200
    - service_policy: 'automatic'
    - service_restart: True

configure-vmotion:
  esxi.vmotion_configured:
    - enabled: True

configure-host-vsan:
  esxi.vsan_configured:
    - enabled: True
    - add_disks_to_vsan: True

configure-host-password:
  esxi.password_present:
    - password: 'new-bad-password'
```

States are called via the ESXi Proxy Minion just as they would on a regular minion. For example:

```
salt 'esxi-*' state.sls configure-esxi test=true
salt 'esxi-*' state.sls configure-esxi
```

Relevant Salt Files and Resources

- *ESXi Proxy Minion*
- *ESXi Execution Module*
- *ESXi State Module*
- *Salt Proxy Minion Docs*
- *Salt Proxy Minion End-to-End Example*
- *vSphere Execution Module*

4.9.6 Installing and Configuring Halite

> **Warning:** Halite is deprecated
>
> The Halite project is retired. The code will remain available on GitHub.

In this tutorial, we'll walk through installing and setting up Halite. The current version of Halite is considered pre-alpha and is supported only in Salt `v2014.1.0` or greater. Additional information is available on GitHub: https://github.com/saltstack/halite

Before beginning this tutorial, ensure that the salt-master is installed. To install the salt-master, please review the installation documentation: http://docs.saltstack.com/topics/installation/index.html

Note: Halite only works with Salt versions greater than 2014.1.0.

Installing Halite Via Package

On CentOS, RHEL, or Fedora:

```
$ yum install python-halite
```

Note: By default python-halite only installs CherryPy. If you would like to use a different webserver please review the instructions below to install pip and your server of choice. The package does not modify the master configuration with `/etc/salt/master`.

Installing Halite Using pip

To begin the installation of Halite from PyPI, you'll need to install pip. The Salt package, as well as the bootstrap, do not install pip by default.

On CentOS, RHEL, or Fedora:

```
$ yum install python-pip
```

On Debian:

```
$ apt-get install python-pip
```

Once you have pip installed, use it to install halite:

```
$ pip install -U halite
```

Depending on the webserver you want to run halite through, you'll need to install that piece as well. On RHEL based distros, use one of the following:

```
$ pip install cherrypy
```

```
$ pip install paste
```

```
$ yum install python-devel
$ yum install gcc
$ pip install gevent
$ pip install pyopenssl
```

On Debian based distributions:

```
$ pip install CherryPy
```

```
$ pip install paste
$ apt-get install gcc
$ apt-get install python-dev
$ apt-get install libevent-dev
$ pip install gevent
$ pip install pyopenssl
```

Configuring Halite Permissions

Configuring Halite access permissions is easy. By default, you only need to ensure that the @runner group is configured. In the `/etc/salt/master` file, uncomment and modify the following lines:

```
external_auth:
  pam:
    testuser:
      - .*
      - '@runner'
```

Note: You cannot use the root user for pam login; it will fail to authenticate.

Halite uses the runner manage.present to get the status of minions, so runner permissions are required. For example:

```
external_auth:
  pam:
    mytestuser:
      - .*
      - '@runner'
      - '@wheel'
```

Currently Halite allows, but does not require, any wheel modules.

Configuring Halite Settings

Once you've configured the permissions for Halite, you'll need to set up the Halite settings in the /etc/salt/master file. Halite supports CherryPy, Paste, and Gevent out of the box.

To configure cherrypy, add the following to the bottom of your /etc/salt/master file:

```
halite:
  level: 'debug'
  server: 'cherrypy'
  host: '0.0.0.0'
  port: '8080'
  cors: False
  tls: True
  certpath: '/etc/pki/tls/certs/localhost.crt'
  keypath: '/etc/pki/tls/certs/localhost.key'
  pempath: '/etc/pki/tls/certs/localhost.pem'
```

If you wish to use paste:

```
halite:
  level: 'debug'
  server: 'paste'
```

(continues on next page)

(continued from previous page)

```
  host: '0.0.0.0'
  port: '8080'
  cors: False
  tls: True
  certpath: '/etc/pki/tls/certs/localhost.crt'
  keypath: '/etc/pki/tls/certs/localhost.key'
  pempath: '/etc/pki/tls/certs/localhost.pem'
```

To use gevent:

```
halite:
  level: 'debug'
  server: 'gevent'
  host: '0.0.0.0'
  port: '8080'
  cors: False
  tls: True
  certpath: '/etc/pki/tls/certs/localhost.crt'
  keypath: '/etc/pki/tls/certs/localhost.key'
  pempath: '/etc/pki/tls/certs/localhost.pem'
```

The ”cherrypy” and ”gevent” servers require the certpath and keypath files to run tls/ssl. The .crt file holds the public cert and the .key file holds the private key. Whereas the ”paste” server requires a single .pem file that contains both the cert and key. This can be created simply by concatenating the .crt and .key files.

If you want to use a self-signed cert, you can create one using the Salt.tls module:

Note: The following command needs to be run on your salt master.

```
salt-call tls.create_self_signed_cert tls
```

Note that certs generated by the above command can be found under the `/etc/pki/tls/certs/` directory. When using self-signed certs, browsers will need approval before accepting the cert. If the web application page has been cached with a non-HTTPS version of the app, then the browser cache will have to be cleared before it will recognize and prompt to accept the self-signed certificate.

Starting Halite

Once you've configured the halite section of your /etc/salt/master, you can restart the salt-master service, and your halite instance will be available. Depending on your configuration, the instance will be available either at https://localhost:8080/app, https://domain:8080/app, or https://123.456.789.012:8080/app .

Note: halite requires an HTML 5 compliant browser.

All logs relating to halite are logged to the default /var/log/salt/master file.

4.9.7 HTTP Modules

This tutorial demonstrates using the various HTTP modules available in Salt. These modules wrap the Python `tornado`, `urllib2`, and `requests` libraries, extending them in a manner that is more consistent with Salt workflows.

The `salt.utils.http` Library

This library forms the core of the HTTP modules. Since it is designed to be used from the minion as an execution module, in addition to the master as a runner, it was abstracted into this multi-use library. This library can also be imported by 3rd-party programs wishing to take advantage of its extended functionality.

Core functionality of the execution, state, and runner modules is derived from this library, so common usages between them are described here. Documentation specific to each module is described below.

This library can be imported with:

```
import salt.utils.http
```

Configuring Libraries

This library can make use of either `tornado`, which is required by Salt, `urllib2`, which ships with Python, or `requests`, which can be installed separately. By default, `tornado` will be used. In order to switch to `urllib2`, set the following variable:

```
backend: urllib2
```

In order to switch to `requests`, set the following variable:

```
backend: requests
```

This can be set in the master or minion configuration file, or passed as an option directly to any `http.query()` functions.

`salt.utils.http.query()`

This function forms a basic query, but with some add-ons not present in the `tornado`, `urllib2`, and `requests` libraries. Not all functionality currently available in these libraries has been added, but can be in future iterations.

HTTPS Request Methods

A basic query can be performed by calling this function with no more than a single URL:

```
salt.utils.http.query('http://example.com')
```

By default the query will be performed with a `GET` method. The method can be overridden with the `method` argument:

```
salt.utils.http.query('http://example.com/delete/url', 'DELETE')
```

When using the `POST` method (and others, such as `PUT`), extra data is usually sent as well. This data can be sent directly (would be URL encoded when necessary), or in whatever format is required by the remote server (XML, JSON, plain text, etc).

```
salt.utils.http.query(
    'http://example.com/post/url',
    method='POST',
    data=json.dumps(mydict)
)
```

Data Formatting and Templating

Bear in mind that the data must be sent pre-formatted; this function will not format it for you. However, a templated file stored on the local system may be passed through, along with variables to populate it with. To pass through only the file (untemplated):

```
salt.utils.http.query(
    'http://example.com/post/url',
    method='POST',
    data_file='/srv/salt/somefile.xml'
)
```

To pass through a file that contains jinja + yaml templating (the default):

```
salt.utils.http.query(
    'http://example.com/post/url',
    method='POST',
    data_file='/srv/salt/somefile.jinja',
    data_render=True,
    template_dict={'key1': 'value1', 'key2': 'value2'}
)
```

To pass through a file that contains mako templating:

```
salt.utils.http.query(
    'http://example.com/post/url',
    method='POST',
    data_file='/srv/salt/somefile.mako',
    data_render=True,
    data_renderer='mako',
    template_dict={'key1': 'value1', 'key2': 'value2'}
)
```

Because this function uses Salt's own rendering system, any Salt renderer can be used. Because Salt's renderer requires `__opts__` to be set, an `opts` dictionary should be passed in. If it is not, then the default `__opts__` values for the node type (master or minion) will be used. Because this library is intended primarily for use by minions, the default node type is `minion`. However, this can be changed to `master` if necessary.

```
salt.utils.http.query(
    'http://example.com/post/url',
    method='POST',
    data_file='/srv/salt/somefile.jinja',
    data_render=True,
    template_dict={'key1': 'value1', 'key2': 'value2'},
    opts=__opts__
)

salt.utils.http.query(
    'http://example.com/post/url',
    method='POST',
    data_file='/srv/salt/somefile.jinja',
    data_render=True,
    template_dict={'key1': 'value1', 'key2': 'value2'},
    node='master'
)
```

Headers

Headers may also be passed through, either as a `header_list`, a `header_dict`, or as a `header_file`. As with the `data_file`, the `header_file` may also be templated. Take note that because HTTP headers are normally syntactically-correct YAML, they will automatically be imported as an a Python dict.

```
salt.utils.http.query(
    'http://example.com/delete/url',
    method='POST',
    header_file='/srv/salt/headers.jinja',
    header_render=True,
    header_renderer='jinja',
    template_dict={'key1': 'value1', 'key2': 'value2'}
)
```

Because much of the data that would be templated between headers and data may be the same, the `template_dict` is the same for both. Correcting possible variable name collisions is up to the user.

Authentication

The `query()` function supports basic HTTP authentication. A username and password may be passed in as `username` and `password`, respectively.

```
salt.utils.http.query(
    'http://example.com',
    username='larry',
    password=`5700g3543v4r`,
)
```

Cookies and Sessions

Cookies are also supported, using Python's built-in `cookielib`. However, they are turned off by default. To turn cookies on, set `cookies` to True.

```
salt.utils.http.query(
    'http://example.com',
    cookies=True
)
```

By default cookies are stored in Salt's cache directory, normally `/var/cache/salt`, as a file called `cookies.txt`. However, this location may be changed with the `cookie_jar` argument:

```
salt.utils.http.query(
    'http://example.com',
    cookies=True,
    cookie_jar='/path/to/cookie_jar.txt'
)
```

By default, the format of the cookie jar is LWP (aka, lib-www-perl). This default was chosen because it is a human-readable text file. If desired, the format of the cookie jar can be set to Mozilla:

```
salt.utils.http.query(
    'http://example.com',
```

(continues on next page)

(continued from previous page)

```
    cookies=True,
    cookie_jar='/path/to/cookie_jar.txt',
    cookie_format='mozilla'
)
```

Because Salt commands are normally one-off commands that are piped together, this library cannot normally behave as a normal browser, with session cookies that persist across multiple HTTP requests. However, the session can be persisted in a separate cookie jar. The default filename for this file, inside Salt's cache directory, is `cookies.session.p`. This can also be changed.

```
salt.utils.http.query(
    'http://example.com',
    persist_session=True,
    session_cookie_jar='/path/to/jar.p'
)
```

The format of this file is msgpack, which is consistent with much of the rest of Salt's internal structure. Historically, the extension for this file is `.p`. There are no current plans to make this configurable.

Proxy

If the `tornado` backend is used (`tornado` is the default), proxy information configured in `proxy_host`, `proxy_port`, `proxy_username`, `proxy_password` and `no_proxy` from the `__opts__` dictionary will be used. Normally these are set in the minion configuration file.

```
proxy_host: proxy.my-domain
proxy_port: 31337
proxy_username: charon
proxy_password: obolus
no_proxy: ['127.0.0.1', 'localhost']
```

```
salt.utils.http.query(
    'http://example.com',
    opts=__opts__,
    backend='tornado'
)
```

Return Data

Note: Return data encoding

If `decode` is set to `True`, `query()` will attempt to decode the return data. `decode_type` defaults to `auto`. Set it to a specific encoding, `xml`, for example, to override autodetection.

Because Salt's http library was designed to be used with REST interfaces, `query()` will attempt to decode the data received from the remote server when `decode` is set to `True`. First it will check the `Content-type` header to try and find references to XML. If it does not find any, it will look for references to JSON. If it does not find any, it will fall back to plain text, which will not be decoded.

JSON data is translated into a dict using Python's built-in `json` library. XML is translated using `salt.utils.xml_util`, which will use Python's built-in XML libraries to attempt to convert the XML into a dict. In order to force either JSON or XML decoding, the `decode_type` may be set:

```
salt.utils.http.query(
    'http://example.com',
    decode_type='xml'
)
```

Once translated, the return dict from `query()` will include a dict called `dict`.

If the data is not to be translated using one of these methods, decoding may be turned off.

```
salt.utils.http.query(
    'http://example.com',
    decode=False
)
```

If decoding is turned on, and references to JSON or XML cannot be found, then this module will default to plain text, and return the undecoded data as `text` (even if text is set to `False`; see below).

The `query()` function can return the HTTP status code, headers, and/or text as required. However, each must individually be turned on.

```
salt.utils.http.query(
    'http://example.com',
    status=True,
    headers=True,
    text=True
)
```

The return from these will be found in the return dict as `status`, `headers` and `text`, respectively.

Writing Return Data to Files

It is possible to write either the return data or headers to files, as soon as the response is received from the server, but specifying file locations via the `text_out` or `headers_out` arguments. `text` and `headers` do not need to be returned to the user in order to do this.

```
salt.utils.http.query(
    'http://example.com',
    text=False,
    headers=False,
    text_out='/path/to/url_download.txt',
    headers_out='/path/to/headers_download.txt',
)
```

SSL Verification

By default, this function will verify SSL certificates. However, for testing or debugging purposes, SSL verification can be turned off.

```
salt.utils.http.query(
    'https://example.com',
    verify_ssl=False,
)
```

CA Bundles

The `requests` library has its own method of detecting which CA (certificate authority) bundle file to use. Usually this is implemented by the packager for the specific operating system distribution that you are using. However, `urllib2` requires a little more work under the hood. By default, Salt will try to auto-detect the location of this file. However, if it is not in an expected location, or a different path needs to be specified, it may be done so using the `ca_bundle` variable.

```
salt.utils.http.query(
    'https://example.com',
    ca_bundle='/path/to/ca_bundle.pem',
)
```

Updating CA Bundles

The `update_ca_bundle()` function can be used to update the bundle file at a specified location. If the target location is not specified, then it will attempt to auto-detect the location of the bundle file. If the URL to download the bundle from does not exist, a bundle will be downloaded from the cURL website.

CAUTION: The `target` and the `source` should always be specified! Failure to specify the `target` may result in the file being written to the wrong location on the local system. Failure to specify the `source` may cause the upstream URL to receive excess unnecessary traffic, and may cause a file to be download which is hazardous or does not meet the needs of the user.

```
salt.utils.http.update_ca_bundle(
    target='/path/to/ca-bundle.crt',
    source='https://example.com/path/to/ca-bundle.crt',
    opts=__opts__,
)
```

The `opts` parameter should also always be specified. If it is, then the `target` and the `source` may be specified in the relevant configuration file (master or minion) as `ca_bundle` and `ca_bundle_url`, respectively.

```
ca_bundle: /path/to/ca-bundle.crt
ca_bundle_url: https://example.com/path/to/ca-bundle.crt
```

If Salt is unable to auto-detect the location of the CA bundle, it will raise an error.

The `update_ca_bundle()` function can also be passed a string or a list of strings which represent files on the local system, which should be appended (in the specified order) to the end of the CA bundle file. This is useful in environments where private certs need to be made available, and are not otherwise reasonable to add to the bundle file.

```
salt.utils.http.update_ca_bundle(
    opts=__opts__,
    merge_files=[
        '/etc/ssl/private_cert_1.pem',
        '/etc/ssl/private_cert_2.pem',
        '/etc/ssl/private_cert_3.pem',
    ]
)
```

Test Mode

This function may be run in test mode. This mode will perform all work up until the actual HTTP request. By default, instead of performing the request, an empty dict will be returned. Using this function with `TRACE` logging turned on will reveal the contents of the headers and POST data to be sent.

Rather than returning an empty dict, an alternate `test_url` may be passed in. If this is detected, then test mode will replace the `url` with the `test_url`, set `test` to `True` in the return data, and perform the rest of the requested operations as usual. This allows a custom, non-destructive URL to be used for testing when necessary.

Execution Module

The `http` execution module is a very thin wrapper around the `salt.utils.http` library. The `opts` can be passed through as well, but if they are not specified, the minion defaults will be used as necessary.

Because passing complete data structures from the command line can be tricky at best and dangerous (in terms of execution injection attacks) at worse, the `data_file`, and `header_file` are likely to see more use here.

All methods for the library are available in the execution module, as kwargs.

```
salt myminion http.query http://example.com/restapi method=POST \
    username='larry' password='5700g3543v4r' headers=True text=True \
    status=True decode_type=xml data_render=True \
    header_file=/tmp/headers.txt data_file=/tmp/data.txt \
    header_render=True cookies=True persist_session=True
```

Runner Module

Like the execution module, the `http` runner module is a very thin wrapper around the `salt.utils.http` library. The only significant difference is that because runners execute on the master instead of a minion, a target is not required, and default opts will be derived from the master config, rather than the minion config.

All methods for the library are available in the runner module, as kwargs.

```
salt-run http.query http://example.com/restapi method=POST \
    username='larry' password='5700g3543v4r' headers=True text=True \
    status=True decode_type=xml data_render=True \
    header_file=/tmp/headers.txt data_file=/tmp/data.txt \
    header_render=True cookies=True persist_session=True
```

State Module

The state module is a wrapper around the runner module, which applies stateful logic to a query. All kwargs as listed above are specified as usual in state files, but two more kwargs are available to apply stateful logic. A required parameter is `match`, which specifies a pattern to look for in the return text. By default, this will perform a string comparison of looking for the value of match in the return text. In Python terms this looks like:

```
if match in html_text:
    return True
```

If more complex pattern matching is required, a regular expression can be used by specifying a `match_type`. By default this is set to `string`, but it can be manually set to `pcre` instead. Please note that despite the name, this will use Python's `re.search()` rather than `re.match()`.

Therefore, the following states are valid:

```
http://example.com/restapi:
  http.query:
    - match: 'SUCCESS'
    - username: 'larry'
    - password: '5700g3543v4r'
    - data_render: True
    - header_file: /tmp/headers.txt
    - data_file: /tmp/data.txt
    - header_render: True
    - cookies: True
    - persist_session: True

http://example.com/restapi:
  http.query:
    - match_type: pcre
    - match: '(?i)succe[ss|ed]'
    - username: 'larry'
    - password: '5700g3543v4r'
    - data_render: True
    - header_file: /tmp/headers.txt
    - data_file: /tmp/data.txt
    - header_render: True
    - cookies: True
    - persist_session: True
```

In addition to, or instead of a match pattern, the status code for a URL can be checked. This is done using the `status` argument:

```
http://example.com/:
  http.query:
    - status: 200
```

If both are specified, both will be checked, but if only one is `True` and the other is `False`, then `False` will be returned. In this case, the comments in the return data will contain information for troubleshooting.

Because this is a monitoring state, it will return extra data to code that expects it. This data will always include `text` and `status`. Optionally, `headers` and `dict` may also be requested by setting the `headers` and `decode` arguments to True, respectively.

4.9.8 Using Salt at scale

The focus of this tutorial will be building a Salt infrastructure for handling large numbers of minions. This will include tuning, topology, and best practices.

For how to install the Salt Master please go here: Installing saltstack

---

Note: This tutorial is intended for large installations, although these same settings won't hurt, it may not be worth the complexity to smaller installations.

When used with minions, the term 'many' refers to at least a thousand and 'a few' always means 500.

For simplicity reasons, this tutorial will default to the standard ports used by Salt.

The Master

The most common problems on the Salt Master are:

1. too many minions authing at once
2. too many minions re-authing at once
3. too many minions re-connecting at once
4. too many minions returning at once
5. too few resources (CPU/HDD)

The first three are all ”thundering herd” problems. To mitigate these issues we must configure the minions to back-off appropriately when the Master is under heavy load.

The fourth is caused by masters with little hardware resources in combination with a possible bug in ZeroMQ. At least that’s what it looks like till today (Issue 118651, Issue 5948, Mail thread)

To fully understand each problem, it is important to understand, how Salt works.

Very briefly, the Salt Master offers two services to the minions.

- a job publisher on port 4505
- an open port 4506 to receive the minions returns

All minions are always connected to the publisher on port 4505 and only connect to the open return port 4506 if necessary. On an idle Master, there will only be connections on port 4505.

Too many minions authing

When the Minion service is first started up, it will connect to its Master’s publisher on port 4505. If too many minions are started at once, this can cause a ”thundering herd”. This can be avoided by not starting too many minions at once.

The connection itself usually isn’t the culprit, the more likely cause of master-side issues is the authentication that the Minion must do with the Master. If the Master is too heavily loaded to handle the auth request it will time it out. The Minion will then wait *acceptance_wait_time* to retry. If *acceptance_wait_time_max* is set then the Minion will increase its wait time by the *acceptance_wait_time* each subsequent retry until reaching *acceptance_wait_time_max*.

Too many minions re-authing

This is most likely to happen in the testing phase of a Salt deployment, when all Minion keys have already been accepted, but the framework is being tested and parameters are frequently changed in the Salt Master’s configuration file(s).

The Salt Master generates a new AES key to encrypt its publications at certain events such as a Master restart or the removal of a Minion key. If you are encountering this problem of too many minions re-authing against the Master, you will need to recalibrate your setup to reduce the rate of events like a Master restart or Minion key removal (`salt-key -d`).

When the Master generates a new AES key, the minions aren’t notified of this but will discover it on the next pub job they receive. When the Minion receives such a job it will then re-auth with the Master. Since Salt does minion-side filtering this means that all the minions will re-auth on the next command published on the master– causing another ”thundering herd”. This can be avoided by setting the

```
random_reauth_delay: 60
```

in the minions configuration file to a higher value and stagger the amount of re-auth attempts. Increasing this value will of course increase the time it takes until all minions are reachable via Salt commands.

Too many minions re-connecting

By default the zmq socket will re-connect every 100ms which for some larger installations may be too quick. This will control how quickly the TCP session is re-established, but has no bearing on the auth load.

To tune the minions sockets reconnect attempts, there are a few values in the sample configuration file (default values)

```
recon_default: 1000
recon_max: 5000
recon_randomize: True
```

- recon_default: the default value the socket should use, i.e. 1000. This value is in milliseconds. (1000ms = 1 second)
- recon_max: the max value that the socket should use as a delay before trying to reconnect This value is in milliseconds. (5000ms = 5 seconds)
- recon_randomize: enables randomization between recon_default and recon_max

To tune this values to an existing environment, a few decision have to be made.

1. How long can one wait, before the minions should be online and reachable via Salt?
2. How many reconnects can the Master handle without a syn flood?

These questions can not be answered generally. Their answers depend on the hardware and the administrators requirements.

Here is an example scenario with the goal, to have all minions reconnect within a 60 second time-frame on a Salt Master service restart.

```
recon_default: 1000
recon_max: 59000
recon_randomize: True
```

Each Minion will have a randomized reconnect value between 'recon_default' and 'recon_default + recon_max', which in this example means between 1000ms and 60000ms (or between 1 and 60 seconds). The generated random-value will be doubled after each attempt to reconnect (ZeroMQ default behavior).

Lets say the generated random value is 11 seconds (or 11000ms).

```
reconnect 1: wait 11 seconds
reconnect 2: wait 22 seconds
reconnect 3: wait 33 seconds
reconnect 4: wait 44 seconds
reconnect 5: wait 55 seconds
reconnect 6: wait time is bigger than 60 seconds (recon_default + recon_max)
reconnect 7: wait 11 seconds
reconnect 8: wait 22 seconds
reconnect 9: wait 33 seconds
reconnect x: etc.
```

With a thousand minions this will mean

```
1000/60 = ~16
```

round about 16 connection attempts a second. These values should be altered to values that match your environment. Keep in mind though, that it may grow over time and that more minions might raise the problem again.

Too many minions returning at once

This can also happen during the testing phase, if all minions are addressed at once with

```
$ salt * disk.usage
```

it may cause thousands of minions trying to return their data to the Salt Master open port 4506. Also causing a flood of syn-flood if the Master can't handle that many returns at once.

This can be easily avoided with Salt's batch mode:

```
$ salt * disk.usage -b 50
```

This will only address 50 minions at once while looping through all addressed minions.

Too few resources

The masters resources always have to match the environment. There is no way to give good advise without knowing the environment the Master is supposed to run in. But here are some general tuning tips for different situations:

The Master is CPU bound

Salt uses RSA-Key-Pairs on the masters and minions end. Both generate 4096 bit key-pairs on first start. While the key-size for the Master is currently not configurable, the minions keysize can be configured with different key-sizes. For example with a 2048 bit key:

```
keysize: 2048
```

With thousands of decryptions, the amount of time that can be saved on the masters end should not be neglected. See here for reference: Pull Request 9235 how much influence the key-size can have.

Downsizing the Salt Master's key is not that important, because the minions do not encrypt as many messages as the Master does.

In installations with large or with complex pillar files, it is possible for the master to exhibit poor performance as a result of having to render many pillar files at once. This exhibit itself in a number of ways, both as high load on the master and on minions which block on waiting for their pillar to be delivered to them.

To reduce pillar rendering times, it is possible to cache pillars on the master. To do this, see the set of master configuration options which are prefixed with *pillar_cache*.

Note: Caching pillars on the master may introduce security considerations. Be certain to read caveats outlined in the master configuration file to understand how pillar caching may affect a master's ability to protect sensitive data!

The Master is disk IO bound

By default, the Master saves every Minion's return for every job in its job-cache. The cache can then be used later, to lookup results for previous jobs. The default directory for this is:

```
cachedir: /var/cache/salt
```

and then in the `/proc` directory.

Each job return for every Minion is saved in a single file. Over time this directory can grow quite large, depending on the number of published jobs. The amount of files and directories will scale with the number of jobs published and the retention time defined by

```
keep_jobs: 24
```

```
250 jobs/day * 2000 minions returns = 500,000 files a day
```

Use and External Job Cache

An external job cache allows for job storage to be placed on an external system, such as a database.

- ext_job_cache: this will have the minions store their return data directly into a returner (not sent through the Master)
- master_job_cache (New in *2014.7.0*): this will make the Master store the job data using a returner (instead of the local job cache on disk).

If a master has many accepted keys, it may take a long time to publish a job because the master much first determine the matching minions and deliver that information back to the waiting client before the job can be published.

To mitigate this, a key cache may be enabled. This will reduce the load on the master to a single file open instead of thousands or tens of thousands.

This cache is updated by the maintanence process, however, which means that minions with keys that are accepted may not be targeted by the master for up to sixty seconds by default.

To enable the master key cache, set *key_cache: 'sched'* in the master configuration file.

Disable The Job Cache

The job cache is a central component of the Salt Master and many aspects of the Salt Master will not function correctly without a running job cache.

Disabling the job cache is **STRONGLY DISCOURAGED** and should not be done unless the master is being used to execute routines that require no history or reliable feedback!

The job cache can be disabled:

```
job_cache: False
```

4.9.9 How to Convert Jinja Logic to an Execution Module

Note: This tutorial assumes a basic knowledge of Salt states and specifically experience using the *maps.jinja* idiom.

This tutorial was written by a salt user who was told "if your maps.jinja is too complicated, write an execution module!". If you are experiencing over-complicated jinja, read on.

The Problem: Jinja Gone Wild

It is often said in the Salt community that "Jinja is not a Programming Language". There's an even older saying known as Maslow's hammer. It goes something like "if all you have is a hammer, everything looks like a nail". Jinja is a reliable hammer, and so is the *maps.jinja* idiom. Unfortunately, it can lead to code that looks like the following.

```
# storage/maps.yaml

{% import_yaml 'storage/defaults.yaml' as default_settings %}
{% set storage = default_settings.storage %}
{% do storage.update(salt['grains.filter_by']({
    'Debian': {
    },
    'RedHat': {
    }
}, merge=salt['pillar.get']('storage:lookup'))) %}

{% if 'VirtualBox' == grains.get('virtual', None) or 'oracle' == grains.get('virtual',
↪ None) %}
{%   do storage.update({'depot_ip': '192.168.33.81', 'server_ip':  '192.168.33.51'})
↪%}
{% else %}
{%   set colo = pillar.get('inventory', {}).get('colo', 'Unknown') %}
{%   set servers_list = pillar.get('storage_servers', {}).get(colo, [storage.depot_ip,
↪ ]) %}
{%   if opts.id.startswith('foo') %}
{%     set modulus = servers_list | count %}
{%     set integer_id = opts.id | replace('foo', '') | int %}
{%     set server_index = integer_id % modulus %}
{%   else %}
{%     set server_index = 0 %}
{%   endif %}
{%   do storage.update({'server_ip': servers_list[server_index]}) %}
{% endif %}

{% for network, _ in salt.pillar.get('inventory:networks', {}) | dictsort %}
{%   do storage.ipsets.hash_net.foo_networks.append(network) %}
{% endfor %}
```

This is an example from the author's salt formulae demonstrating misuse of jinja. Aside from being difficult to read and maintain, accessing the logic it contains from a non-jinja renderer while probably possible is a significant barrier!

Refactor

The first step is to reduce the maps.jinja file to something reasonable. This gives us an idea of what the module we are writing needs to do. There is a lot of logic around selecting a storage server ip. Let's move that to an execution module.

```
# storage/maps.yaml

{% import_yaml 'storage/defaults.yaml' as default_settings %}
{% set storage = default_settings.storage %}
{% do storage.update(salt['grains.filter_by']({
    'Debian': {
    },
    'RedHat': {
```

(continues on next page)

(continued from previous page)

```
    }
}, merge=salt['pillar.get']('storage:lookup'))) %}

{% if 'VirtualBox' == grains.get('virtual', None) or 'oracle' == grains.get('virtual',
↪ None) %}
{%   do storage.update({'depot_ip': '192.168.33.81'}) %}
{% endif %}

{% do storage.update({'server_ip': salt['storage.ip']()}) %}

{% for network, _ in salt.pillar.get('inventory:networks', {}) | dictsort %}
{%   do storage.ipsets.hash_net.af_networks.append(network) %}
{% endfor %}
```

And then, write the module. Note how the module encapsulates all of the logic around finding the storage server IP.

```
# _modules/storage.py
#!python

'''
Functions related to storage servers.
'''

import re


def ips():
    '''
    Provide a list of all local storage server IPs.

    CLI Example::

        salt \* storage.ips
    '''

    if  __grains__.get('virtual', None) in ['VirtualBox', 'oracle']:
        return ['192.168.33.51', ]

    colo = __pillar__.get('inventory', {}).get('colo', 'Unknown')
    return __pillar__.get('storage_servers', {}).get(colo, ['unknown', ])


def ip():
    '''
    Select and return a local storage server IP.

    This loadbalances across storage servers by using the modulus of the client's id
↪number.

    :maintainer:    Andrew Hammond <ahammond@anchorfree.com>
    :maturity:      new
    :depends:       None
    :platform:      all

    CLI Example::
```

(continues on next page)

(continued from previous page)

```
        salt \* storage.ip

    '''

    numerical_suffix = re.compile(r'^.*(\d+)$')
    servers_list = ips()

    m = numerical_suffix.match(__grains__['id'])
    if m:
        modulus = len(servers_list)
        server_number = int(m.group(1))
        server_index = server_number % modulus
    else:
        server_index = 0

    return servers_list[server_index]
```

Conclusion

That was… surprisingly straight-forward. Now the logic is available in every renderer, instead of just Jinja. Best of all, it can be maintained in Python, which is a whole lot easier than Jinja.

4.9.10 Using Apache Libcloud for declarative and procedural multi-cloud orchestration

New in version 2018.3.0.

Note: This walkthrough assumes basic knowledge of Salt and Salt States. To get up to speed, check out the *Salt Walkthrough*.

Apache Libcloud is a Python library which hides differences between different cloud provider APIs and allows you to manage different cloud resources through a unified and easy to use API. Apache Libcloud supports over 60 cloud platforms, including Amazon, Microsoft Azure, DigitalOcean, Google Cloud Platform and OpenStack.

Execution and state modules are available for Compute, DNS, Storage and Load Balancer drivers from Apache Libcloud in SaltStack.

- `libcloud_compute` - **Compute** - services such as OpenStack Nova, Amazon EC2, Microsoft Azure VMs
- `libcloud_dns` - **DNS as a Service** - services such as Amazon Route 53 and Zerigo
- `libcloud_loadbalancer` - **Load Balancers as a Service** - services such as Amazon Elastic Load Balancer and GoGrid LoadBalancers
- `libcloud_storage` - **Cloud Object Storage and CDN** - services such as Amazon S3 and Rackspace CloudFiles, OpenStack Swift

These modules are designed as a way of having a multi-cloud deployment and abstracting simple differences between platform to design a high-availability architecture.

The Apache Libcloud functionality is available through both execution modules and Salt states.

Configuring Drivers

Drivers can be configured in the Salt Configuration/Minion settings. All libcloud modules expect a list of "profiles" to be configured with authentication details for each driver.

Each driver will have a string identifier, these can be found in the libcloud.<api>.types.Provider class for each API, http://libcloud.readthedocs.io/en/latest/supported_providers.html

Some drivers require additional parameters, which are documented in the Apache Libcloud documentation. For example, GoDaddy DNS expects "*shopper_id*", which is the customer ID. These additional parameters can be added to the profile settings and will be passed directly to the driver instantiation method.

```
libcloud_dns:
    godaddy:
        driver: godaddy
        shopper_id: 90425123
        key: AFDDJFGIjDFVNSDIFNASMC
        secret: FG(#f8vdfgjlkm)

libcloud_storage:
    google:
        driver: google_storage
        key: GOOG4ASDIDFNVIdfnIVW
        secret: R+qYE9hkfdhv89h4invhdfvird4Pq3an8rnK
```

You can have multiple profiles for a single driver, for example if you wanted 2 DNS profiles for Amazon Route53, naming them "route53_prod" and "route54_test" would help your administrators distinguish their purpose.

```
libcloud_dns:
    route53_prod:
        driver: route53
        key: AFDDJFGIjDFVNSDIFNASMC
        secret: FG(#f8vdfgjlkm)
    route53_test:
        driver: route53
        key: AFDDJFGIjdfgdfgdf
        secret: FG(#f8vdfgjlkm)
```

Using the execution modules

Amongst over 60 clouds that Apache Libcloud supports, you can add profiles to your Salt configuration to access and control these clouds. Each of the libcloud execution modules exposes the common API methods for controlling Compute, DNS, Load Balancers and Object Storage. To see which functions are supported across specific clouds, see the Libcloud supported methods documentation.

The module documentation explains each of the API methods and how to leverage them.

- `libcloud_compute` - **Compute** - services such as OpenStack Nova, Amazon EC2, Microsoft Azure VMs
- `libcloud_dns` - **DNS as a Service** - services such as Amazon Route 53 and Zerigo
- `libcloud_loadbalancer` - **Load Balancers as a Service** - services such as Amazon Elastic Load Balancer and GoGrid LoadBalancers
- `libcloud_storage` - **Cloud Object Storage and CDN** - services such as Amazon S3 and Rackspace CloudFiles, OpenStack Swift

For example, listing buckets in the Google Storage platform:

```
$ salt-call libcloud_storage.list_containers google

    local:
        |_
        ----------
        extra:
            ----------
            creation_date:
                2017-01-05T05:44:56.324Z
        name:
            anthonypjshaw
```

The Apache Libcloud storage module can be used to syncronize files between multiple storage clouds, such as Google Storage, S3 and OpenStack Swift

```
$ salt '*' libcloud_storage.download_object DeploymentTools test.sh /tmp/test.sh␣
↪google_storage
```

Using the state modules

For each configured profile, the assets available in the API (e.g. storage objects, containers, DNS records and load balancers) can be deployed via Salt's state system.

The state module documentation explains the specific states that each module supports

- *`libcloud_storage`* - **Cloud Object Storage and CDN**
    - services such as Amazon S3 and Rackspace CloudFiles, OpenStack Swift
- *`libcloud_loadbalancer`* - **Load Balancers as a Service**
    - services such as Amazon Elastic Load Balancer and GoGrid LoadBalancers
- *`libcloud_dns`* - **DNS as a Service**
    - services such as Amazon Route 53 and Zerigo

For DNS, the state modules can be used to provide DNS resilience for multiple nameservers, for example:

```
libcloud_dns:
    godaddy:
        driver: godaddy
        shopper_id: 12345
        key: 2orgk34kgk34g
        secret: fjgoidhjgoim
    amazon:
        driver: route53
        key: blah
        secret: blah
```

And then in a state file:

```
webserver:
  libcloud_dns.zone_present:
    name: mywebsite.com
    profile: godaddy
  libcloud_dns.record_present:
    name: www
```

(continues on next page)

(continued from previous page)

```
    zone: mywebsite.com
    type: A
    data: 12.34.32.3
    profile: godaddy
  libcloud_dns.zone_present:
    name: mywebsite.com
    profile: amazon
  libcloud_dns.record_present:
    name: www
    zone: mywebsite.com
    type: A
    data: 12.34.32.3
    profile: amazon
```

This could be combined with a multi-cloud load balancer deployment,

```
webserver:
  libcloud_dns.zone_present:
    - name: mywebsite.com
    - profile: godaddy
    ...
  libcloud_loadbalancer.balancer_present:
    - name: web_main
    - port: 80
    - protocol: http
    - members:
        - ip: 1.2.4.5
          port: 80
        - ip: 2.4.5.6
          port: 80
    - profile: google_gce
  libcloud_loadbalancer.balancer_present:
    - name: web_main
    - port: 80
    - protocol: http
    - members:
        - ip: 1.2.4.5
          port: 80
        - ip: 2.4.5.6
          port: 80
    - profile: amazon_elb
```

Extended parameters can be passed to the specific cloud, for example you can specify the region with the Google Cloud API, because *create_balancer* can accept a *ex_region* argument. Adding this argument to the state will pass the additional command to the driver.

```
lb_test:
    libcloud_loadbalancer.balancer_absent:
        - name: example
        - port: 80
        - protocol: http
        - profile: google
        - ex_region: us-east1
```

Accessing custom arguments in execution modules

Some cloud providers have additional functionality that can be accessed on top of the base API, for example the Google Cloud Engine load balancer service offers the ability to provision load balancers into a specific region.

Looking at the API documentation, we can see that it expects an *ex_region* in the *create_balancer* method, so when we execute the salt command, we can add this additional parameter like this:

```
$ salt myminion libcloud_storage.create_balancer my_balancer 80 http profile1 ex_
↪region=us-east1
$ salt myminion libcloud_storage.list_container_objects my_bucket profile1 ex_
↪prefix=me
```

Accessing custom methods in Libcloud drivers

Some cloud APIs have additional methods that are prefixed with *ex_* in Apache Libcloud, these methods are part of the non-standard API but can still be accessed from the Salt modules for *libcloud_storage*, *libcloud_loadbalancer* and *libcloud_dns*. The extra methods are available via the *extra* command, which expects the name of the method as the first argument, the profile as the second and then accepts a list of keyword arguments to pass onto the driver method, for example, accessing permissions in Google Storage objects:

```
$ salt myminion libcloud_storage.extra ex_get_permissions google container_name=my_
↪container object_name=me.jpg --out=yaml
```

Example profiles

Google Cloud

Using Service Accounts with GCE, you can provide a path to the JSON file and the project name in the parameters.

```
google:
    driver: gce
    user_id: 234234-compute@developer.gserviceaccount.com
    key: /path/to/service_account_download.json
    auth_type: SA
    project: project-name
```

4.9.11 LXC Management with Salt

Note: This walkthrough assumes basic knowledge of Salt. To get up to speed, check out the *Salt Walkthrough*.

Dependencies

Manipulation of LXC containers in Salt requires the minion to have an LXC version of at least 1.0 (an alpha or beta release of LXC 1.0 is acceptable). The following distributions are known to have new enough versions of LXC packaged:

- RHEL/CentOS 6 and later (via EPEL)
- Fedora (All non-EOL releases)

- Debian 8.0 (Jessie)
- Ubuntu 14.04 LTS and later (LXC templates are packaged separately as **lxc-templates**, it is recommended to also install this package)
- openSUSE 13.2 and later

Profiles

Profiles allow for a sort of shorthand for commonly-used configurations to be defined in the minion config file, *grains*, *pillar*, or the master config file. The profile is retrieved by Salt using the `config.get` function, which looks in those locations, in that order. This allows for profiles to be defined centrally in the master config file, with several options for overriding them (if necessary) on groups of minions or individual minions.

There are two types of profiles:

- One for defining the parameters used in container creation/clone.
- One for defining the container's network interface(s) settings.

Container Profiles

LXC container profiles are defined underneath the `lxc.container_profile` config option:

```
lxc.container_profile:
  centos:
    template: centos
    backing: lvm
    vgname: vg1
    lvname: lxclv
    size: 10G
  centos_big:
    template: centos
    backing: lvm
    vgname: vg1
    lvname: lxclv
    size: 20G
```

Profiles are retrieved using the `config.get` function, with the **recurse** merge strategy. This means that a profile can be defined at a lower level (for example, the master config file) and then parts of it can be overridden at a higher level (for example, in pillar data). Consider the following container profile data:

In the Master config file:

```
lxc.container_profile:
  centos:
    template: centos
    backing: lvm
    vgname: vg1
    lvname: lxclv
    size: 10G
```

In the Pillar data

```
lxc.container_profile:
  centos:
    size: 20G
```

Any minion with the above Pillar data would have the **size** parameter in the **centos** profile overridden to 20G, while those minions without the above Pillar data would have the 10G **size** value. This is another way of achieving the same result as the **centos_big** profile above, without having to define another whole profile that differs in just one value.

Note: In the 2014.7.x release cycle and earlier, container profiles are defined under `lxc.profile`. This parameter will still work in version 2015.5.0, but is deprecated and will be removed in a future release. Please note however that the profile merging feature described above will only work with profiles defined under `lxc.container_profile`, and only in versions 2015.5.0 and later.

Additionally, in version 2015.5.0 container profiles have been expanded to support passing template-specific CLI options to `lxc.create`. Below is a table describing the parameters which can be configured in container profiles:

Parameter	2015.5.0 and Newer	2014.7.x and Earlier
template[1]	Yes	Yes
options[1]	Yes	No
image[1]	Yes	Yes
backing	Yes	Yes
snapshot[2]	Yes	Yes
lvname[1]	Yes	Yes
fstype[1]	Yes	Yes
size	Yes	Yes

1. Parameter is only supported for container creation, and will be ignored if the profile is used when cloning a container.
2. Parameter is only supported for container cloning, and will be ignored if the profile is used when not cloning a container.

Network Profiles

LXC network profiles are defined defined underneath the `lxc.network_profile` config option. By default, the module uses a DHCP based configuration and try to guess a bridge to get connectivity.

Warning: on pre **2015.5.2**, you need to specify explicitly the network bridge

```
lxc.network_profile:
  centos:
    eth0:
      link: br0
      type: veth
      flags: up
  ubuntu:
    eth0:
      link: lxcbr0
      type: veth
      flags: up
```

As with container profiles, network profiles are retrieved using the `config.get` function, with the **recurse** merge strategy. Consider the following network profile data:

In the Master config file:

```
lxc.network_profile:
  centos:
    eth0:
      link: br0
      type: veth
      flags: up
```

In the Pillar data

```
lxc.network_profile:
  centos:
    eth0:
      link: lxcbr0
```

Any minion with the above Pillar data would use the **lxcbr0** interface as the bridge interface for any container configured using the **centos** network profile, while those minions without the above Pillar data would use the **br0** interface for the same.

Note: In the 2014.7.x release cycle and earlier, network profiles are defined under `lxc.nic`. This parameter will still work in version 2015.5.0, but is deprecated and will be removed in a future release. Please note however that the profile merging feature described above will only work with profiles defined under `lxc.network_profile`, and only in versions 2015.5.0 and later.

The following are parameters which can be configured in network profiles. These will directly correspond to a parameter in an LXC configuration file (see `man 5 lxc.container.conf`).

- **type** - Corresponds to **lxc.network.type**
- **link** - Corresponds to **lxc.network.link**
- **flags** - Corresponds to **lxc.network.flags**

Interface-specific options (MAC address, IPv4/IPv6, etc.) must be passed on a container-by-container basis, for instance using the `nic_opts` argument to *`lxc.create`*:

```
salt myminion lxc.create container1 profile=centos network_profile=centos nic_opts='
↪{eth0: {ipv4: 10.0.0.20/24, gateway: 10.0.0.1}}'
```

Warning: The `ipv4`, `ipv6`, `gateway`, and `link` (bridge) settings in network profiles / nic_opts will only work if the container doesn't redefine the network configuration (for example in `/etc/sysconfig/network-scripts/ifcfg-<interface_name>` on RHEL/CentOS, or `/etc/network/interfaces` on Debian/Ubuntu/etc.). Use these with caution. The container images installed using the `download` template, for instance, typically are configured for eth0 to use DHCP, which will conflict with static IP addresses set at the container level.

Note: For LXC < 1.0.7 and DHCP support, set `ipv4.gateway: 'auto'` is your network profile, ie.:

```
lxc.network_profile.nic:
  debian:
    eth0:
      link: lxcbr0
      ipv4.gateway: 'auto'
```

Old lxc support (<1.0.7)

With saltstack **2015.5.2** and above, normally the setting is autoselected, but before, you'll need to teach your network profile to set **lxc.network.ipv4.gateway** to **auto** when using a classic ipv4 configuration.

Thus you'll need

```
lxc.network_profile.foo:
  etho:
    link: lxcbr0
    ipv4.gateway: auto
```

Tricky network setups Examples

This example covers how to make a container with both an internal ip and a public routable ip, wired on two veth pairs.

The another interface which receives directly a public routable ip can't be on the first interface that we reserve for private inter LXC networking.

```
lxc.network_profile.foo:
  eth0: {gateway: null, bridge: lxcbr0}
  eth1:
    # replace that by your main interface
    'link': 'br0'
    'mac': '00:16:5b:01:24:e1'
    'gateway': '2.20.9.14'
    'ipv4': '2.20.9.1'
```

Creating a Container on the CLI

From a Template

LXC is commonly distributed with several template scripts in /usr/share/lxc/templates. Some distros may package these separately in an **lxc-templates** package, so make sure to check if this is the case.

There are LXC template scripts for several different operating systems, but some of them are designed to use tools specific to a given distribution. For instance, the `ubuntu` template uses deb_bootstrap, the `centos` template uses yum, etc., making these templates impractical when a container from a different OS is desired.

The *`lxc.create`* function is used to create containers using a template script. To create a CentOS container named `container1` on a CentOS minion named `mycentosminion`, using the `centos` LXC template, one can simply run the following command:

```
salt mycentosminion lxc.create container1 template=centos
```

For these instances, there is a `download` template which retrieves minimal container images for several different operating systems. To use this template, it is necessary to provide an `options` parameter when creating the container, with three values:

1. **dist** - the Linux distribution (i.e. `ubuntu` or `centos`)
2. **release** - the release name/version (i.e. `trusty` or `6`)
3. **arch** - CPU architecture (i.e. `amd64` or `i386`)

The *lxc.images* function (new in version 2015.5.0) can be used to list the available images. Alternatively, the releases can be viewed on http://images.linuxcontainers.org/images/. The images are organized in such a way that the **dist**, **release**, and **arch** can be determined using the following URL format: `http://images.linuxcontainers.org/images/dist/release/arch`. For example, `http://images.linuxcontainers.org/images/centos/6/amd64` would correspond to a **dist** of `centos`, a **release** of `6`, and an **arch** of `amd64`.

Therefore, to use the `download` template to create a new 64-bit CentOS 6 container, the following command can be used:

```
salt myminion lxc.create container1 template=download options='{dist: centos,
↪release: 6, arch: amd64}'
```

Note: These command-line options can be placed into a *container profile*, like so:

```
lxc.container_profile.cent6:
  template: download
  options:
    dist: centos
    release: 6
    arch: amd64
```

The `options` parameter is not supported in profiles for the 2014.7.x release cycle and earlier, so it would still need to be provided on the command-line.

Cloning an Existing Container

To clone a container, use the *lxc.clone* function:

```
salt myminion lxc.clone container2 orig=container1
```

Using a Container Image

While cloning is a good way to create new containers from a common base container, the source container that is being cloned needs to already exist on the minion. This makes deploying a common container across minions difficult. For this reason, Salt's *lxc.create* is capable of installing a container from a tar archive of another container's rootfs. To create an image of a container named `cent6`, run the following command as root:

```
tar czf cent6.tar.gz -C /var/lib/lxc/cent6 rootfs
```

Note: Before doing this, it is recommended that the container is stopped.

The resulting tarball can then be placed alongside the files in the salt fileserver and referenced using a `salt://` URL. To create a container using an image, use the `image` parameter with *lxc.create*:

```
salt myminion lxc.create new-cent6 image=salt://path/to/cent6.tar.gz
```

Note: Making images of containers with LVM backing

For containers with LVM backing, the rootfs is not mounted, so it is necessary to mount it first before creating the tar archive. When a container is created using LVM backing, an empty `rootfs` dir is handily created within `/var/lib/lxc/container_name`, so this can be used as the mountpoint. The location of the logical volume for the container will be `/dev/vgname/lvname`, where `vgname` is the name of the volume group, and `lvname` is the name of the logical volume. Therefore, assuming a volume group of `vg1`, a logical volume of `lxc-cent6`, and a container name of `cent6`, the following commands can be used to create a tar archive of the rootfs:

```
mount /dev/vg1/lxc-cent6 /var/lib/lxc/cent6/rootfs
tar czf cent6.tar.gz -C /var/lib/lxc/cent6 rootfs
umount /var/lib/lxc/cent6/rootfs
```

Warning: One caveat of using this method of container creation is that `/etc/hosts` is left unmodified. This could cause confusion for some distros if salt-minion is later installed on the container, as the functions that determine the hostname take `/etc/hosts` into account.

Additionally, when creating an rootfs image, be sure to remove `/etc/salt/minion_id` and make sure that `id` is not defined in `/etc/salt/minion`, as this will cause similar issues.

Initializing a New Container as a Salt Minion

The above examples illustrate a few ways to create containers on the CLI, but often it is desirable to also have the new container run as a Minion. To do this, the *`lxc.init`* function can be used. This function will do the following:

1. Create a new container
2. Optionally set password and/or DNS
3. Bootstrap the minion (using either salt-bootstrap or a custom command)

By default, the new container will be pointed at the same Salt Master as the host machine on which the container was created. It will then request to authenticate with the Master like any other bootstrapped Minion, at which point it can be accepted.

```
salt myminion lxc.init test1 profile=centos
salt-key -a test1
```

For even greater convenience, the *`LXC runner`* contains a runner function of the same name (*`lxc.init`*), which creates a keypair, seeds the new minion with it, and pre-accepts the key, allowing for the new Minion to be created and authorized in a single step:

```
salt-run lxc.init test1 host=myminion profile=centos
```

Running Commands Within a Container

For containers which are not running their own Minion, commands can be run within the container in a manner similar to using (`cmd.run <salt.modules.cmdmod.run`). The means of doing this have been changed significantly in version 2015.5.0 (though the deprecated behavior will still be supported for a few releases). Both the old and new usage are documented below.

2015.5.0 and Newer

New functions have been added to mimic the behavior of the functions in the *cmd* module. Below is a table with the *cmd* functions and their *lxc* module equivalents:

Description	*cmd* module	*lxc* module
Run a command and get all output	*cmd.run*	*lxc.run*
Run a command and get just stdout	*cmd.run_stdout*	*lxc.run_stdout*
Run a command and get just stderr	*cmd.run_stderr*	*lxc.run_stderr*
Run a command and get just the retcode	*cmd.retcode*	*lxc.retcode*
Run a command and get all information	*cmd.run_all*	*lxc.run_all*

2014.7.x and Earlier

Earlier Salt releases use a single function (`lxc.run_cmd`) to run commands within containers. Whether stdout, stderr, etc. are returned depends on how the function is invoked.

To run a command and return the stdout:

```
salt myminion lxc.run_cmd web1 'tail /var/log/messages'
```

To run a command and return the stderr:

```
salt myminion lxc.run_cmd web1 'tail /var/log/messages' stdout=False stderr=True
```

To run a command and return the retcode:

```
salt myminion lxc.run_cmd web1 'tail /var/log/messages' stdout=False stderr=False
```

To run a command and return all information:

```
salt myminion lxc.run_cmd web1 'tail /var/log/messages' stdout=True stderr=True
```

Container Management Using salt-cloud

Salt cloud uses under the hood the salt runner and module to manage containers, Please look at *this chapter*

Container Management Using States

Several states are being renamed or otherwise modified in version 2015.5.0. The information in this tutorial refers to the new states. For 2014.7.x and earlier, please refer to the *documentation for the LXC states*.

Ensuring a Container Is Present

To ensure the existence of a named container, use the *lxc.present* state. Here are some examples:

```
# Using a template
web1:
  lxc.present:
    - template: download
```

(continues on next page)

(continued from previous page)

```
    - options:
        dist: centos
        release: 6
        arch: amd64

# Cloning
web2:
  lxc.present:
    - clone_from: web-base

# Using a rootfs image
web3:
  lxc.present:
    - image: salt://path/to/cent6.tar.gz

# Using profiles
web4:
  lxc.present:
    - profile: centos_web
    - network_profile: centos
```

> **Warning:** The *lxc.present* state will not modify an existing container (in other words, it will not re-create the container). If an *lxc.present* state is run on an existing container, there will be no change and the state will return a `True` result.

The *lxc.present* state also includes an optional `running` parameter which can be used to ensure that a container is running/stopped. Note that there are standalone *lxc.running* and *lxc.stopped* states which can be used for this purpose.

Ensuring a Container Does Not Exist

To ensure that a named container is not present, use the *lxc.absent* state. For example:

```
web1:
  lxc.absent
```

Ensuring a Container is Running/Stopped/Frozen

Containers can be in one of three states:

- **running** - Container is running and active
- **frozen** - Container is running, but all process are blocked and the container is essentially non-active until the container is "unfrozen"
- **stopped** - Container is not running

Salt has three states (*lxc.running*, *lxc.frozen*, and *lxc.stopped*) which can be used to ensure a container is in one of these states:

```
web1:
  lxc.running
```

(continues on next page)

(continued from previous page)

```
# Restart the container if it was already running
web2:
  lxc.running:
    - restart: True

web3:
  lxc.stopped

# Explicitly kill all tasks in container instead of gracefully stopping
web4:
  lxc.stopped:
    - kill: True

web5:
  lxc.frozen

# If container is stopped, do not start it (in which case the state will fail)
web6:
  lxc.frozen:
    - start: False
```

4.9.12 Remote execution tutorial

Before continuing make sure you have a working Salt installation by following the *Installation* and the *configuration* instructions.

Stuck?

There are many ways to *get help from the Salt community* including our mailing list and our IRC channel #salt.

Order your minions around

Now that you have a *master* and at least one *minion* communicating with each other you can perform commands on the minion via the **salt** command. Salt calls are comprised of three main components:

```
salt '<target>' <function> [arguments]
```

See also:

salt manpage

target

The target component allows you to filter which minions should run the following function. The default filter is a glob on the minion id. For example:

```
salt '*' test.version
salt '*.example.org' test.version
```

Targets can be based on minion system information using the Grains system:

```
salt -G 'os:Ubuntu' test.version
```

See also:

Grains system

Targets can be filtered by regular expression:

```
salt -E 'virtmach[0-9]' test.version
```

Targets can be explicitly specified in a list:

```
salt -L 'foo,bar,baz,quo' test.version
```

Or Multiple target types can be combined in one command:

```
salt -C 'G@os:Ubuntu and webser* or E@database.*' test.version
```

function

A function is some functionality provided by a module. Salt ships with a large collection of available functions. List all available functions on your minions:

```
salt '*' sys.doc
```

Here are some examples:

Show all currently available minions:

```
salt '*' test.version
```

Run an arbitrary shell command:

```
salt '*' cmd.run 'uname -a'
```

See also:

the full list of modules

arguments

Space-delimited arguments to the function:

```
salt '*' cmd.exec_code python 'import sys; print sys.version'
```

Optional, keyword arguments are also supported:

```
salt '*' pip.install salt timeout=5 upgrade=True
```

They are always in the form of `kwarg=argument`.

4.9.13 Multi Master Tutorial

As of Salt 0.16.0, the ability to connect minions to multiple masters has been made available. The multi-master system allows for redundancy of Salt masters and facilitates multiple points of communication out to minions. When using a multi-master setup, all masters are running hot, and any active master can be used to send commands out to the minions.

Note: If you need failover capabilities with multiple masters, there is also a MultiMaster-PKI setup available, that uses a different topology MultiMaster-PKI with Failover Tutorial

In 0.16.0, the masters do not share any information, keys need to be accepted on both masters, and shared files need to be shared manually or use tools like the git fileserver backend to ensure that the `file_roots` are kept consistent.

Beginning with Salt 2016.11.0, the *Pluggable Minion Data Cache* was introduced. The minion data cache contains the Salt Mine data, minion grains, and minion pillar information cached on the Salt Master. By default, Salt uses the `localfs` cache module, but other external data stores can be used instead.

Using a pluggable minion cache modules allows for the data stored on a Salt Master about Salt Minions to be replicated on other Salt Masters the Minion is connected to. Please see the *Minion Data Cache* documentation for more information and configuration examples.

Summary of Steps

1. Create a redundant master server
2. Copy primary master key to redundant master
3. Start redundant master
4. Configure minions to connect to redundant master
5. Restart minions
6. Accept keys on redundant master

Prepping a Redundant Master

The first task is to prepare the redundant master. If the redundant master is already running, stop it. There is only one requirement when preparing a redundant master, which is that masters share the same private key. When the first master was created, the master's identifying key pair was generated and placed in the master's `pki_dir`. The default location of the master's key pair is `/etc/salt/pki/master/`. Take the private key, `master.pem`, and copy it to the same location on the redundant master. Do the same for the master's public key, `master.pub`. Assuming that no minions have yet been connected to the new redundant master, it is safe to delete any existing key in this location and replace it.

Note: There is no logical limit to the number of redundant masters that can be used.

Once the new key is in place, the redundant master can be safely started.

Configure Minions

Since minions need to be master-aware, the new master needs to be added to the minion configurations. Simply update the minion configurations to list all connected masters:

```
master:
  - saltmaster1.example.com
  - saltmaster2.example.com
```

Now the minion can be safely restarted.

Note: If the ipc_mode for the minion is set to TCP (default in Windows), then each minion in the multi-minion setup (one per master) needs its own tcp_pub_port and tcp_pull_port.

If these settings are left as the default 4510/4511, each minion object will receive a port 2 higher than the previous. Thus the first minion will get 4510/4511, the second will get 4512/4513, and so on. If these port decisions are unacceptable, you must configure tcp_pub_port and tcp_pull_port with lists of ports for each master. The length of these lists should match the number of masters, and there should not be overlap in the lists.

Now the minions will check into the original master and also check into the new redundant master. Both masters are first-class and have rights to the minions.

Note: Minions can automatically detect failed masters and attempt to reconnect to them quickly. To enable this functionality, set *master_alive_interval* in the minion config and specify a number of seconds to poll the masters for connection status.

If this option is not set, minions will still reconnect to failed masters but the first command sent after a master comes back up may be lost while the minion authenticates.

Sharing Files Between Masters

Salt does not automatically share files between multiple masters. A number of files should be shared or sharing of these files should be strongly considered.

Minion Keys

Minion keys can be accepted the normal way using **salt-key** on both masters. Keys accepted, deleted, or rejected on one master will NOT be automatically managed on redundant masters; this needs to be taken care of by running salt-key on both masters or sharing the `/etc/salt/pki/master/{minions,minions_pre, minions_rejected}` directories between masters.

Note: While sharing the **/etc/salt/pki/master** directory will work, it is strongly discouraged, since allowing access to the **master.pem** key outside of Salt creates a *SERIOUS* security risk.

File_Roots

The *`file_roots`* contents should be kept consistent between masters. Otherwise state runs will not always be consistent on minions since instructions managed by one master will not agree with other masters.

The recommended way to sync these is to use a fileserver backend like gitfs or to keep these files on shared storage.

Important: If using gitfs/git_pillar with the cachedir shared between masters using GlusterFS, nfs, or another network filesystem, and the masters are running Salt 2015.5.9 or later, it is strongly recommended not to turn off `gitfs_global_lock`/`git_pillar_global_lock` as doing so will cause lock files to be removed if they were created by a different master.

Pillar_Roots

Pillar roots should be given the same considerations as `file_roots`.

Master Configurations

While reasons may exist to maintain separate master configurations, it is wise to remember that each master maintains independent control over minions. Therefore, access controls should be in sync between masters unless a valid reason otherwise exists to keep them inconsistent.

These access control options include but are not limited to:

- external_auth
- publisher_acl
- peer
- peer_run

4.9.14 Multi-Master-PKI Tutorial With Failover

This tutorial will explain, how to run a salt-environment where a single minion can have multiple masters and fail-over between them if its current master fails.

The individual steps are

- setup the master(s) to sign its auth-replies
- setup minion(s) to verify master-public-keys
- enable multiple masters on minion(s)
- enable master-check on minion(s)

> Please note, that it is advised to have good knowledge of the salt- authentication and communication-process to understand this tutorial. All of the settings described here, go on top of the default authentication/communication process.

Motivation

The default behaviour of a salt-minion is to connect to a master and accept the masters public key. With each publication, the master sends his public-key for the minion to check and if this public-key ever changes, the minion complains and exits. Practically this means, that there can only be a single master at any given time.

Would it not be much nicer, if the minion could have any number of masters (1:n) and jump to the next master if its current master died because of a network or hardware failure?

Note: There is also a MultiMaster-Tutorial with a different approach and topology than this one, that might also suite your needs or might even be better suited Multi-Master Tutorial

---

It is also desirable, to add some sort of authenticity-check to the very first public key a minion receives from a master. Currently a minions takes the first masters public key for granted.

The Goal

Setup the master to sign the public key it sends to the minions and enable the minions to verify this signature for authenticity.

Prepping the master to sign its public key

For signing to work, both master and minion must have the signing and/or verification settings enabled. If the master signs the public key but the minion does not verify it, the minion will complain and exit. The same happens, when the master does not sign but the minion tries to verify.

The easiest way to have the master sign its public key is to set

```
master_sign_pubkey: True
```

After restarting the salt-master service, the master will automatically generate a new key-pair

```
master_sign.pem
master_sign.pub
```

A custom name can be set for the signing key-pair by setting

```
master_sign_key_name: <name_without_suffix>
```

The master will then generate that key-pair upon restart and use it for creating the public keys signature attached to the auth-reply.

The computation is done for every auth-request of a minion. If many minions auth very often, it is advised to use conf_master:*master_pubkey_signature* and conf_master:*master_use_pubkey_signature* settings described below.

If multiple masters are in use and should sign their auth-replies, the signing key-pair master_sign.* has to be copied to each master. Otherwise a minion will fail to verify the masters public when connecting to a different master than it did initially. That is because the public keys signature was created with a different signing key-pair.

Prepping the minion to verify received public keys

The minion must have the public key (and only that one!) available to be able to verify a signature it receives. That public key (defaults to master_sign.pub) must be copied from the master to the minions pki-directory.

```
/etc/salt/pki/minion/master_sign.pub
```

Important: DO NOT COPY THE master_sign.pem FILE. IT MUST STAY ON THE MASTER AND ONLY THERE!

When that is done, enable the signature checking in the minions configuration

```
verify_master_pubkey_sign: True
```

and restart the minion. For the first try, the minion should be run in manual debug mode.

```
salt-minion -l debug
```

Upon connecting to the master, the following lines should appear on the output:

```
[DEBUG   ] Attempting to authenticate with the Salt Master at 172.16.0.10
[DEBUG   ] Loaded minion key: /etc/salt/pki/minion/minion.pem
[DEBUG   ] salt.crypt.verify_signature: Loading public key
[DEBUG   ] salt.crypt.verify_signature: Verifying signature
[DEBUG   ] Successfully verified signature of master public key with verification
↪public key master_sign.pub
[INFO    ] Received signed and verified master pubkey from master 172.16.0.10
[DEBUG   ] Decrypting the current master AES key
```

If the signature verification fails, something went wrong and it will look like this

```
[DEBUG   ] Attempting to authenticate with the Salt Master at 172.16.0.10
[DEBUG   ] Loaded minion key: /etc/salt/pki/minion/minion.pem
[DEBUG   ] salt.crypt.verify_signature: Loading public key
[DEBUG   ] salt.crypt.verify_signature: Verifying signature
[DEBUG   ] Failed to verify signature of public key
[CRITICAL] The Salt Master server's public key did not authenticate!
```

In a case like this, it should be checked, that the verification pubkey (master_sign.pub) on the minion is the same as the one on the master.

Once the verification is successful, the minion can be started in daemon mode again.

For the paranoid among us, its also possible to verify the publication whenever it is received from the master. That is, for every single auth-attempt which can be quite frequent. For example just the start of the minion will force the signature to be checked 6 times for various things like auth, mine, *highstate*, etc.

If that is desired, enable the setting

```
always_verify_signature: True
```

Multiple Masters For A Minion

Configuring multiple masters on a minion is done by specifying two settings:

- a list of masters addresses
- what type of master is defined

```
master:
    - 172.16.0.10
    - 172.16.0.11
    - 172.16.0.12
```

```
master_type: failover
```

This tells the minion that all the master above are available for it to connect to. When started with this configuration, it will try the master in the order they are defined. To randomize that order, set

```
master_shuffle: True
```

The master-list will then be shuffled before the first connection attempt.

The first master that accepts the minion, is used by the minion. If the master does not yet know the minion, that counts as accepted and the minion stays on that master.

For the minion to be able to detect if its still connected to its current master enable the check for it

```
master_alive_interval: <seconds>
```

If the loss of the connection is detected, the minion will temporarily remove the failed master from the list and try one of the other masters defined (again shuffled if that is enabled).

Testing the setup

At least two running masters are needed to test the failover setup.

Both masters should be running and the minion should be running on the command line in debug mode

```
salt-minion -l debug
```

The minion will connect to the first master from its master list

```
[DEBUG   ] Attempting to authenticate with the Salt Master at 172.16.0.10
[DEBUG   ] Loaded minion key: /etc/salt/pki/minion/minion.pem
[DEBUG   ] salt.crypt.verify_signature: Loading public key
[DEBUG   ] salt.crypt.verify_signature: Verifying signature
[DEBUG   ] Successfully verified signature of master public key with verification
↪public key master_sign.pub
[INFO    ] Received signed and verified master pubkey from master 172.16.0.10
[DEBUG   ] Decrypting the current master AES key
```

A test.version on the master the minion is currently connected to should be run to test connectivity.

If successful, that master should be turned off. A firewall-rule denying the minions packets will also do the trick.

Depending on the configured conf_minion:*master_alive_interval*, the minion will notice the loss of the connection and log it to its logfile.

```
[INFO    ] Connection to master 172.16.0.10 lost
[INFO    ] Trying to tune in to next master from master-list
```

The minion will then remove the current master from the list and try connecting to the next master

```
[INFO    ] Removing possibly failed master 172.16.0.10 from list of masters
[WARNING ] Master ip address changed from 172.16.0.10 to 172.16.0.11
[DEBUG   ] Attempting to authenticate with the Salt Master at 172.16.0.11
```

If everything is configured correctly, the new masters public key will be verified successfully

```
[DEBUG   ] Loaded minion key: /etc/salt/pki/minion/minion.pem
[DEBUG   ] salt.crypt.verify_signature: Loading public key
[DEBUG   ] salt.crypt.verify_signature: Verifying signature
[DEBUG   ] Successfully verified signature of master public key with verification
↪public key master_sign.pub
```

the authentication with the new master is successful

```
[INFO    ] Received signed and verified master pubkey from master 172.16.0.11
[DEBUG   ] Decrypting the current master AES key
```

(continues on next page)

(continued from previous page)

```
[DEBUG   ] Loaded minion key: /etc/salt/pki/minion/minion.pem
[INFO    ] Authentication with master successful!
```

and the minion can be pinged again from its new master.

Performance Tuning

With the setup described above, the master computes a signature for every auth-request of a minion. With many minions and many auth-requests, that can chew up quite a bit of CPU-Power.

To avoid that, the master can use a pre-created signature of its public-key. The signature is saved as a base64 encoded string which the master reads once when starting and attaches only that string to auth-replies.

Enabling this also gives paranoid users the possibility, to have the signing key-pair on a different system than the actual salt-master and create the public keys signature there. Probably on a system with more restrictive firewall rules, without internet access, less users, etc.

That signature can be created with

```
salt-key --gen-signature
```

This will create a default signature file in the master pki-directory

```
/etc/salt/pki/master/master_pubkey_signature
```

It is a simple text-file with the binary-signature converted to base64.

If no signing-pair is present yet, this will auto-create the signing pair and the signature file in one call

```
salt-key --gen-signature --auto-create
```

Telling the master to use the pre-created signature is done with

```yaml
master_use_pubkey_signature: True
```

That requires the file 'master_pubkey_signature' to be present in the masters pki-directory with the correct signature.

If the signature file is named differently, its name can be set with

```yaml
master_pubkey_signature: <filename>
```

With many masters and many public-keys (default and signing), it is advised to use the salt-masters hostname for the signature-files name. Signatures can be easily confused because they do not provide any information about the key the signature was created from.

Verifying that everything works is done the same way as above.

How the signing and verification works

The default key-pair of the salt-master is

```
/etc/salt/pki/master/master.pem
/etc/salt/pki/master/master.pub
```

To be able to create a signature of a message (in this case a public-key), another key-pair has to be added to the setup. Its default name is:

```
master_sign.pem
master_sign.pub
```

The combination of the master.* and master_sign.* key-pairs give the possibility of generating signatures. The signature of a given message is unique and can be verified, if the public-key of the signing-key-pair is available to the recipient (the minion).

The signature of the masters public-key in master.pub is computed with

```
master_sign.pem
master.pub
M2Crypto.EVP.sign_update()
```

This results in a binary signature which is converted to base64 and attached to the auth-reply send to the minion.

With the signing-pairs public-key available to the minion, the attached signature can be verified with

```
master_sign.pub
master.pub
M2Cryptos EVP.verify_update().
```

When running multiple masters, either the signing key-pair has to be present on all of them, or the master_pubkey_signature has to be pre-computed for each master individually (because they all have different public-keys).

> DO NOT PUT THE SAME master.pub ON ALL MASTERS FOR EASE OF USE.

4.9.15 Packaging External Modules for Salt

External Modules Setuptools Entry-Points Support

The salt loader was enhanced to look for external modules by looking at the *salt.loader* entry-point:

> https://setuptools.readthedocs.io/en/latest/pkg_resources.html#entry-points

pkg_resources should be installed, which is normally included in setuptools.

> https://setuptools.readthedocs.io/en/latest/pkg_resources.html

The package which has custom engines, minion modules, outputters, etc, should require setuptools and should define the following entry points in its setup function:

```
from setuptools import setup, find_packages

setup(name=<NAME>,
      version=<VERSION>,
      description=<DESC>,
      author=<AUTHOR>,
      author_email=<AUTHOR-EMAIL>,
      url=' ... ',
      packages=find_packages(),
      entry_points='''
        [salt.loader]
        engines_dirs = <package>.<loader-module>:engines_dirs
        fileserver_dirs = <package>.<loader-module>:fileserver_dirs
        pillar_dirs = <package>.<loader-module>:pillar_dirs
        returner_dirs = <package>.<loader-module>:returner_dirs
```

(continues on next page)

(continued from previous page)

```
        roster_dirs = <package>.<loader-module>:roster_dirs
      ''')
```

The above setup script example mentions a loader module. here's an example of how *<package>/<loader-module>.py* it should look:

```
# -*- coding: utf-8 -*-

# Import python libs
import os

PKG_DIR = os.path.abspath(os.path.dirname(__file__))


def engines_dirs():
    '''
    yield one path per parent directory of where engines can be found
    '''
    yield os.path.join(PKG_DIR, 'engines_1')
    yield os.path.join(PKG_DIR, 'engines_2')


def fileserver_dirs():
    '''
    yield one path per parent directory of where fileserver modules can be found
    '''
    yield os.path.join(PKG_DIR, 'fileserver')


def pillar_dirs():
    '''
    yield one path per parent directory of where external pillar modules can be found
    '''
    yield os.path.join(PKG_DIR, 'pillar')


def returner_dirs():
    '''
    yield one path per parent directory of where returner modules can be found
    '''
    yield os.path.join(PKG_DIR, 'returners')


def roster_dirs():
    '''
    yield one path per parent directory of where roster modules can be found
    '''
    yield os.path.join(PKG_DIR, 'roster')
```

4.9.16 How Do I Use Salt States?

Simplicity, Simplicity, Simplicity

Many of the most powerful and useful engineering solutions are founded on simple principles. Salt States strive to do just that: K.I.S.S. (Keep It Stupidly Simple)

The core of the Salt State system is the SLS, or **S**a**L**t **S**tate file. The SLS is a representation of the state in which a system should be in, and is set up to contain this data in a simple format. This is often called configuration management.

Note: This is just the beginning of using states, make sure to read up on pillar *Pillar* next.

It is All Just Data

Before delving into the particulars, it will help to understand that the SLS file is just a data structure under the hood. While understanding that the SLS is just a data structure isn't critical for understanding and making use of Salt States, it should help bolster knowledge of where the real power is.

SLS files are therefore, in reality, just dictionaries, lists, strings, and numbers. By using this approach Salt can be much more flexible. As one writes more state files, it becomes clearer exactly what is being written. The result is a system that is easy to understand, yet grows with the needs of the admin or developer.

The Top File

The example SLS files in the below sections can be assigned to hosts using a file called **top.sls**. This file is described in-depth *here*.

Default Data - YAML

By default Salt represents the SLS data in what is one of the simplest serialization formats available - YAML.

A typical SLS file will often look like this in YAML:

Note: These demos use some generic service and package names, different distributions often use different names for packages and services. For instance *apache* should be replaced with *httpd* on a Red Hat system. Salt uses the name of the init script, systemd name, upstart name etc. based on what the underlying service management for the platform. To get a list of the available service names on a platform execute the service.get_all salt function.

Information on how to make states work with multiple distributions is later in the tutorial.

```yaml
apache:
  pkg.installed: []
  service.running:
    - require:
      - pkg: apache
```

This SLS data will ensure that the package named apache is installed, and that the apache service is running. The components can be explained in a simple way.

The first line is the ID for a set of data, and it is called the ID Declaration. This ID sets the name of the thing that needs to be manipulated.

The second and third lines contain the state module function to be run, in the format `<state_module>.<function>`. The `pkg.installed` state module function ensures that a software package is installed via the system's native package manager. The `service.running` state module function ensures that a given system daemon is running.

Finally, on line five, is the word `require`. This is called a Requisite Statement, and it makes sure that the Apache service is only started after a successful installation of the apache package.

Adding Configs and Users

When setting up a service like an Apache web server, many more components may need to be added. The Apache configuration file will most likely be managed, and a user and group may need to be set up.

```
apache:
  pkg.installed: []
  service.running:
    - watch:
      - pkg: apache
      - file: /etc/httpd/conf/httpd.conf
      - user: apache
  user.present:
    - uid: 87
    - gid: 87
    - home: /var/www/html
    - shell: /bin/nologin
    - require:
      - group: apache
  group.present:
    - gid: 87
    - require:
      - pkg: apache

/etc/httpd/conf/httpd.conf:
  file.managed:
    - source: salt://apache/httpd.conf
    - user: root
    - group: root
    - mode: 644
```

This SLS data greatly extends the first example, and includes a config file, a user, a group and new requisite statement: `watch`.

Adding more states is easy, since the new user and group states are under the Apache ID, the user and group will be the Apache user and group. The `require` statements will make sure that the user will only be made after the group, and that the group will be made only after the Apache package is installed.

Next, the `require` statement under service was changed to watch, and is now watching 3 states instead of just one. The watch statement does the same thing as require, making sure that the other states run before running the state with a watch, but it adds an extra component. The `watch` statement will run the state's watcher function for any changes to the watched states. So if the package was updated, the config file changed, or the user uid modified, then the service state's watcher will be run. The service state's watcher just restarts the service, so in this case, a change in the config file will also trigger a restart of the respective service.

Moving Beyond a Single SLS

When setting up Salt States in a scalable manner, more than one SLS will need to be used. The above examples were in a single SLS file, but two or more SLS files can be combined to build out a State Tree. The above example also references a file with a strange source - `salt://apache/httpd.conf`. That file will need to be available as well.

The SLS files are laid out in a directory structure on the Salt master; an SLS is just a file and files to download are just files.

The Apache example would be laid out in the root of the Salt file server like this:

```
apache/init.sls
apache/httpd.conf
```

So the httpd.conf is just a file in the apache directory, and is referenced directly.

Do not use dots in SLS file names or their directories

The initial implementation of `top.sls` and *Include declaration* followed the python import model where a slash is represented as a period. This means that a SLS file with a period in the name (besides the suffix period) can not be referenced. For example, webserver_1.0.sls is not referenceable because webserver_1.0 would refer to the directory/file webserver_1/0.sls

The same applies for any subdirectories, this is especially 'tricky' when git repos are created. Another command that typically can't render it's output is ``state.show_sls`` of a file in a path that contains a dot.

But when using more than one single SLS file, more components can be added to the toolkit. Consider this SSH example:

`ssh/init.sls:`

```
openssh-client:
  pkg.installed

/etc/ssh/ssh_config:
  file.managed:
    - user: root
    - group: root
    - mode: 644
    - source: salt://ssh/ssh_config
    - require:
      - pkg: openssh-client
```

`ssh/server.sls:`

```
include:
  - ssh

openssh-server:
  pkg.installed

sshd:
  service.running:
    - require:
      - pkg: openssh-client
      - pkg: openssh-server
      - file: /etc/ssh/banner
      - file: /etc/ssh/sshd_config

/etc/ssh/sshd_config:
  file.managed:
    - user: root
    - group: root
    - mode: 644
    - source: salt://ssh/sshd_config
    - require:
      - pkg: openssh-server
```

(continues on next page)

(continued from previous page)

```

/etc/ssh/banner:
  file:
    - managed
    - user: root
    - group: root
    - mode: 644
    - source: salt://ssh/banner
    - require:
      - pkg: openssh-server
```

Note: Notice that we use two similar ways of denoting that a file is managed by Salt. In the */etc/ssh/sshd_config* state section above, we use the *file.managed* state declaration whereas with the */etc/ssh/banner* state section, we use the *file* state declaration and add a *managed* attribute to that state declaration. Both ways produce an identical result; the first way – using *file.managed* – is merely a shortcut.

Now our State Tree looks like this:

```
apache/init.sls
apache/httpd.conf
ssh/init.sls
ssh/server.sls
ssh/banner
ssh/ssh_config
ssh/sshd_config
```

This example now introduces the `include` statement. The include statement includes another SLS file so that components found in it can be required, watched or as will soon be demonstrated - extended.

The include statement allows for states to be cross linked. When an SLS has an include statement it is literally extended to include the contents of the included SLS files.

Note that some of the SLS files are called init.sls, while others are not. More info on what this means can be found in the *States Tutorial*.

Extending Included SLS Data

Sometimes SLS data needs to be extended. Perhaps the apache service needs to watch additional resources, or under certain circumstances a different file needs to be placed.

In these examples, the first will add a custom banner to ssh and the second will add more watchers to apache to include mod_python.

`ssh/custom-server.sls:`

```
include:
  - ssh.server

extend:
  /etc/ssh/banner:
    file:
      - source: salt://ssh/custom-banner
```

`python/mod_python.sls:`

```
include:
  - apache

extend:
  apache:
    service:
      - watch:
        - pkg: mod_python

mod_python:
  pkg.installed
```

The `custom-server.sls` file uses the extend statement to overwrite where the banner is being downloaded from, and therefore changing what file is being used to configure the banner.

In the new mod_python SLS the mod_python package is added, but more importantly the apache service was extended to also watch the mod_python package.

Using extend with require or watch

The `extend` statement works differently for `require` or `watch`. It appends to, rather than replacing the requisite component.

Understanding the Render System

Since SLS data is simply that (data), it does not need to be represented with YAML. Salt defaults to YAML because it is very straightforward and easy to learn and use. But the SLS files can be rendered from almost any imaginable medium, so long as a renderer module is provided.

The default rendering system is the `jinja|yaml` renderer. The `jinja|yaml` renderer will first pass the template through the Jinja2 templating system, and then through the YAML parser. The benefit here is that full programming constructs are available when creating SLS files.

Other renderers available are `yaml_mako` and `yaml_wempy` which each use the Mako or Wempy templating system respectively rather than the jinja templating system, and more notably, the pure Python or `py`, `pydsl` & `pyobjects` renderers. The `py` renderer allows for SLS files to be written in pure Python, allowing for the utmost level of flexibility and power when preparing SLS data; while the *`pydsl`* renderer provides a flexible, domain-specific language for authoring SLS data in Python; and the *`pyobjects`* renderer gives you a "Pythonic" interface to building state data.

Note: The templating engines described above aren't just available in SLS files. They can also be used in *`file.managed`* states, making file management much more dynamic and flexible. Some examples for using templates in managed files can be found in the documentation for the *`file state`*, as well as the *MooseFS example* below.

Getting to Know the Default - jinja|yaml

The default renderer - `jinja|yaml`, allows for use of the jinja templating system. A guide to the Jinja templating system can be found here: http://jinja.pocoo.org/docs

When working with renderers a few very useful bits of data are passed in. In the case of templating engine based renderers, three critical components are available, `salt`, `grains`, and `pillar`. The `salt` object allows for any

Salt function to be called from within the template, and `grains` allows for the Grains to be accessed from within the template. A few examples:

`apache/init.sls:`

```
apache:
  pkg.installed:
    {% if grains['os'] == 'RedHat'%}
    - name: httpd
    {% endif %}
  service.running:
    {% if grains['os'] == 'RedHat'%}
    - name: httpd
    {% endif %}
    - watch:
      - pkg: apache
      - file: /etc/httpd/conf/httpd.conf
      - user: apache
  user.present:
    - uid: 87
    - gid: 87
    - home: /var/www/html
    - shell: /bin/nologin
    - require:
      - group: apache
  group.present:
    - gid: 87
    - require:
      - pkg: apache

/etc/httpd/conf/httpd.conf:
  file.managed:
    - source: salt://apache/httpd.conf
    - user: root
    - group: root
    - mode: 644
```

This example is simple. If the `os` grain states that the operating system is Red Hat, then the name of the Apache package and service needs to be httpd.

A more aggressive way to use Jinja can be found here, in a module to set up a MooseFS distributed filesystem chunkserver:

`moosefs/chunk.sls:`

```
include:
  - moosefs

{% for mnt in salt['cmd.run']('ls /dev/data/moose*').split() %}
/mnt/moose{{ mnt[-1] }}:
  mount.mounted:
    - device: {{ mnt }}
    - fstype: xfs
    - mkmnt: True
  file.directory:
    - user: mfs
    - group: mfs
    - require:
      - user: mfs
```

(continues on next page)

(continued from previous page)

```
      - group: mfs
{% endfor %}

/etc/mfshdd.cfg:
  file.managed:
    - source: salt://moosefs/mfshdd.cfg
    - user: root
    - group: root
    - mode: 644
    - template: jinja
    - require:
      - pkg: mfs-chunkserver

/etc/mfschunkserver.cfg:
  file.managed:
    - source: salt://moosefs/mfschunkserver.cfg
    - user: root
    - group: root
    - mode: 644
    - template: jinja
    - require:
      - pkg: mfs-chunkserver

mfs-chunkserver:
  pkg.installed: []
mfschunkserver:
  service.running:
    - require:
{% for mnt in salt['cmd.run']('ls /dev/data/moose*') %}
      - mount: /mnt/moose{{ mnt[-1] }}
      - file: /mnt/moose{{ mnt[-1] }}
{% endfor %}
      - file: /etc/mfschunkserver.cfg
      - file: /etc/mfshdd.cfg
      - file: /var/lib/mfs
```

This example shows much more of the available power of Jinja. Multiple for loops are used to dynamically detect available hard drives and set them up to be mounted, and the `salt` object is used multiple times to call shell commands to gather data.

Introducing the Python, PyDSL, and the Pyobjects Renderers

Sometimes the chosen default renderer might not have enough logical power to accomplish the needed task. When this happens, the Python renderer can be used. Normally a YAML renderer should be used for the majority of SLS files, but an SLS file set to use another renderer can be easily added to the tree.

This example shows a very basic Python SLS file:

`python/django.sls:`

```
#!py

def run():
    '''
    Install the django package
```

(continues on next page)

(continued from previous page)

```
    '''
    return {'include': ['python'],
            'django': {'pkg': ['installed']}}
```

This is a very simple example; the first line has an SLS shebang that tells Salt to not use the default renderer, but to use the `py` renderer. Then the run function is defined, the return value from the run function must be a Salt friendly data structure, or better known as a Salt *HighState data structure*.

Alternatively, using the *`pydsl`* renderer, the above example can be written more succinctly as:

```
#!pydsl

include('python', delayed=True)
state('django').pkg.installed()
```

The *`pyobjects`* renderer provides an "Pythonic" object based approach for building the state data. The above example could be written as:

```
#!pyobjects

include('python')
Pkg.installed("django")
```

These Python examples would look like this if they were written in YAML:

```
include:
  - python

django:
  pkg.installed
```

This example clearly illustrates that; one, using the YAML renderer by default is a wise decision and two, unbridled power can be obtained where needed by using a pure Python SLS.

Running and Debugging Salt States

Once the rules in an SLS are ready, they should be tested to ensure they work properly. To invoke these rules, simply execute `salt '*' state.apply` on the command line. If you get back only hostnames with a `:` after, but no return, chances are there is a problem with one or more of the sls files. On the minion, use the `salt-call` command to examine the output for errors:

```
salt-call state.apply -l debug
```

This should help troubleshoot the issue. The minion can also be started in the foreground in debug mode by running `salt-minion -l debug`.

Next Reading

With an understanding of states, the next recommendation is to become familiar with Salt's pillar interface:

> *Pillar Walkthrough*

4.9.17 States tutorial, part 1 - Basic Usage

The purpose of this tutorial is to demonstrate how quickly you can configure a system to be managed by Salt States. For detailed information about the state system please refer to the full *states reference*.

This tutorial will walk you through using Salt to configure a minion to run the Apache HTTP server and to ensure the server is running.

Before continuing make sure you have a working Salt installation by following the *Installation* and the *configuration* instructions.

Stuck?

There are many ways to *get help from the Salt community* including our mailing list and our IRC channel #salt.

Setting up the Salt State Tree

States are stored in text files on the master and transferred to the minions on demand via the master's File Server. The collection of state files make up the `State Tree`.

To start using a central state system in Salt, the Salt File Server must first be set up. Edit the master config file (*`file_roots`*) and uncomment the following lines:

```
file_roots:
  base:
    - /srv/salt
```

Note: If you are deploying on FreeBSD via ports, the `file_roots` path defaults to `/usr/local/etc/salt/states`.

Restart the Salt master in order to pick up this change:

```
pkill salt-master
salt-master -d
```

Preparing the Top File

On the master, in the directory uncommented in the previous step, (`/srv/salt` by default), create a new file called *`top.sls`* and add the following:

```
base:
  '*':
    - webserver
```

The *top file* is separated into environments (discussed later). The default environment is `base`. Under the `base` environment a collection of minion matches is defined; for now simply specify all hosts (`*`).

Targeting minions

The expressions can use any of the targeting mechanisms used by Salt — minions can be matched by glob, PCRE regular expression, or by *grains*. For example:

```
base:
  'os:Fedora':
    - match: grain
    - webserver
```

Create an `sls` file

In the same directory as the *top file*, create a file named `webserver.sls`, containing the following:

```
apache:                 # ID declaration
  pkg:                  # state declaration
    - installed         # function declaration
```

The first line, called the *ID declaration*, is an arbitrary identifier. In this case it defines the name of the package to be installed.

Note: The package name for the Apache httpd web server may differ depending on OS or distro — for example, on Fedora it is `httpd` but on Debian/Ubuntu it is `apache2`.

The second line, called the *State declaration*, defines which of the Salt States we are using. In this example, we are using the *`pkg state`* to ensure that a given package is installed.

The third line, called the *Function declaration*, defines which function in the *`pkg state`* module to call.

Renderers

States `sls` files can be written in many formats. Salt requires only a simple data structure and is not concerned with how that data structure is built. Templating languages and DSLs are a dime-a-dozen and everyone has a favorite.

Building the expected data structure is the job of Salt *Renderers* and they are dead-simple to write.

In this tutorial we will be using YAML in Jinja2 templates, which is the default format. The default can be changed by editing *`renderer`* in the master configuration file.

Install the package

Next, let's run the state we created. Open a terminal on the master and run:

```
salt '*' state.apply
```

Our master is instructing all targeted minions to run `state.apply`. When this function is executed without any SLS targets, a minion will download the *top file* and attempt to match the expressions within it. When the minion does match an expression the modules listed for it will be downloaded, compiled, and executed.

Note: This action is referred to as a "highstate", and can be run using the *`state.highstate`* function. However, to make the usage easier to understand ("highstate" is not necessarily an intuitive name), a *`state.apply`* function was added in version 2015.5.0, which when invoked without any SLS names will trigger a highstate. *`state.highstate`* still exists and can be used, but the documentation (as can be seen above) has been updated to reference *`state.apply`*, so keep the following in mind as you read the documentation:

- *`state.apply`* invoked without any SLS names will run *`state.highstate`*

- `state.apply` invoked with SLS names will run `state.sls`

Once completed, the minion will report back with a summary of all actions taken and all changes made.

Warning: If you have created *custom grain modules*, they will not be available in the top file until after the first *highstate*. To make custom grains available on a minion's first *highstate*, it is recommended to use *this example* to ensure that the custom grains are synced when the minion starts.

SLS File Namespace

Note that in the *example* above, the SLS file `webserver.sls` was referred to simply as `webserver`. The namespace for SLS files when referenced in `top.sls` or an *Include declaration* follows a few simple rules:

1. The `.sls` is discarded (i.e. `webserver.sls` becomes `webserver`).
2. **Subdirectories can be used for better organization**.
   a. Each subdirectory is represented with a dot (following the Python import model) in Salt states and on the command line . `webserver/dev.sls` on the filesystem is referred to as `webserver.dev` in Salt
   b. Because slashes are represented as dots, SLS files can not contain dots in the name (other than the dot for the SLS suffix). The SLS file `webserver_1.0.sls` can not be matched, and `webserver_1.0` would match the directory/file `webserver_1/0.sls`
3. A file called `init.sls` in a subdirectory is referred to by the path of the directory. So, `webserver/init.sls` is referred to as `webserver`.
4. If both `webserver.sls` and `webserver/init.sls` happen to exist, `webserver/init.sls` will be ignored and `webserver.sls` will be the file referred to as `webserver`.

Troubleshooting Salt

If the expected output isn't seen, the following tips can help to narrow down the problem.

Turn up logging Salt can be quite chatty when you change the logging setting to `debug`:

```
salt-minion -l debug
```

Run the minion in the foreground By not starting the minion in daemon mode (*-d*) one can view any output from the minion as it works:

```
salt-minion
```

Increase the default timeout value when running **`salt`**. For example, to change the default timeout to 60 seconds:

```
salt -t 60
```

For best results, combine all three:

```
salt-minion -l debug        # On the minion
salt '*' state.apply -t 60  # On the master
```

Next steps

This tutorial focused on getting a simple Salt States configuration working. *Part 2* will build on this example to cover more advanced `sls` syntax and will explore more of the states that ship with Salt.

4.9.18 States tutorial, part 2 - More Complex States, Requisites

---

Note: This tutorial builds on topics covered in *part 1*. It is recommended that you begin there.

In the *last part* of the Salt States tutorial we covered the basics of installing a package. We will now modify our `webserver.sls` file to have requirements, and use even more Salt States.

Call multiple States

You can specify multiple *State declaration* under an *ID declaration*. For example, a quick modification to our `webserver.sls` to also start Apache if it is not running:

```yaml
apache:
  pkg.installed: []
  service.running:
    - require:
      - pkg: apache
```

Try stopping Apache before running *`state.apply`* once again and observe the output.

Note: For those running RedhatOS derivatives (Centos, AWS), you will want to specify the service name to be httpd. More on state service here, *`service state`*. With the example above, just add "- name: httpd" above the require line and with the same spacing.

Require other states

We now have a working installation of Apache so let's add an HTML file to customize our website. It isn't exactly useful to have a website without a webserver so we don't want Salt to install our HTML file until Apache is installed and running. Include the following at the bottom of your `webserver/init.sls` file:

```yaml
 1 apache:
 2   pkg.installed: []
 3   service.running:
 4     - require:
 5       - pkg: apache
 6 
 7 /var/www/index.html:                        # ID declaration
 8   file:                                     # state declaration
 9     - managed                               # function
10     - source: salt://webserver/index.html   # function arg
11     - require:                              # requisite declaration
12       - pkg: apache                         # requisite reference
```

line 7 is the *ID declaration*. In this example it is the location we want to install our custom HTML file. (**Note**: the default location that Apache serves may differ from the above on your OS or distro. `/srv/www` could also be a likely place to look.)

Line 8 the *State declaration*. This example uses the Salt `file state`.

Line 9 is the *Function declaration*. The `managed function` will download a file from the master and install it in the location specified.

Line 10 is a *Function arg declaration* which, in this example, passes the `source` argument to the `managed function`.

Line 11 is a *Requisite declaration*.

Line 12 is a *Requisite reference* which refers to a state and an ID. In this example, it is referring to the `ID declaration` from our example in *part 1*. This declaration tells Salt not to install the HTML file until Apache is installed.

Next, create the `index.html` file and save it in the `webserver` directory:

```
<!DOCTYPE html>
<html>
    <head><title>Salt rocks</title></head>
    <body>
        <h1>This file brought to you by Salt</h1>
    </body>
</html>
```

Last, call `state.apply` again and the minion will fetch and execute the *highstate* as well as our HTML file from the master using Salt's File Server:

```
salt '*' state.apply
```

Verify that Apache is now serving your custom HTML.

`require` vs. `watch`

There are two *Requisite declaration*, "require", and "watch". Not every state supports "watch". The `service state` does support "watch" and will restart a service based on the watch condition.

For example, if you use Salt to install an Apache virtual host configuration file and want to restart Apache whenever that file is changed you could modify our Apache example from earlier as follows:

```
/etc/httpd/extra/httpd-vhosts.conf:
  file.managed:
    - source: salt://webserver/httpd-vhosts.conf

apache:
  pkg.installed: []
  service.running:
    - watch:
      - file: /etc/httpd/extra/httpd-vhosts.conf
    - require:
      - pkg: apache
```

If the pkg and service names differ on your OS or distro of choice you can specify each one separately using a *Name declaration* which explained in *Part 3*.

Next steps

In *part 3* we will discuss how to use includes, extends, and templating to make a more complete State Tree configuration.

4.9.19 States tutorial, part 3 - Templating, Includes, Extends

Note: This tutorial builds on topics covered in *part 1* and *part 2*. It is recommended that you begin there.

This part of the tutorial will cover more advanced templating and configuration techniques for `sls` files.

Templating SLS modules

SLS modules may require programming logic or inline execution. This is accomplished with module templating. The default module templating system used is Jinja2 and may be configured by changing the *`renderer`* value in the master config.

All states are passed through a templating system when they are initially read. To make use of the templating system, simply add some templating markup. An example of an sls module with templating markup may look like this:

```
{% for usr in ['moe','larry','curly'] %}
{{ usr }}:
  user.present
{% endfor %}
```

This templated sls file once generated will look like this:

```
moe:
  user.present
larry:
  user.present
curly:
  user.present
```

Here's a more complex example:

```
# Comments in yaml start with a hash symbol.
# Since jinja rendering occurs before yaml parsing, if you want to include jinja
# in the comments you may need to escape them using 'jinja' comments to prevent
# jinja from trying to render something which is not well-defined jinja.
# e.g.
# {# iterate over the Three Stooges using a {% for %}..{% endfor %} loop
# with the iterator variable {{ usr }} becoming the state ID. #}
{% for usr in 'moe','larry','curly' %}
{{ usr }}:
  group:
    - present
  user:
    - present
    - gid_from_name: True
    - require:
      - group: {{ usr }}
{% endfor %}
```

Using Grains in SLS modules

Often times a state will need to behave differently on different systems. *Salt grains* objects are made available in the template context. The *grains* can be used from within sls modules:

```
apache:
  pkg.installed:
    {% if grains['os'] == 'RedHat' %}
    - name: httpd
    {% elif grains['os'] == 'Ubuntu' %}
    - name: apache2
    {% endif %}
```

Using Environment Variables in SLS modules

You can use `salt['environ.get']('VARNAME')` to use an environment variable in a Salt state.

```
MYENVVAR="world" salt-call state.template test.sls
```

```
Create a file with contents from an environment variable:
  file.managed:
    - name: /tmp/hello
    - contents: {{ salt['environ.get']('MYENVVAR') }}
```

Error checking:

```
{% set myenvvar = salt['environ.get']('MYENVVAR') %}
{% if myenvvar %}

Create a file with contents from an environment variable:
  file.managed:
    - name: /tmp/hello
    - contents: {{ salt['environ.get']('MYENVVAR') }}

{% else %}

Fail - no environment passed in:
  test.fail_without_changes

{% endif %}
```

Calling Salt modules from templates

All of the Salt modules loaded by the minion are available within the templating system. This allows data to be gathered in real time on the target system. It also allows for shell commands to be run easily from within the sls modules.

The Salt module functions are also made available in the template context as `salt:`

The following example illustrates calling the `group_to_gid` function in the `file` execution module with a single positional argument called `some_group_that_exists`.

```
moe:
  user.present:
    - gid: {{ salt['file.group_to_gid']('some_group_that_exists') }}
```

One way to think about this might be that the `gid` key is being assigned a value equivalent to the following python pseudo-code:

```
import salt.modules.file
file.group_to_gid('some_group_that_exists')
```

Note that for the above example to work, `some_group_that_exists` must exist before the state file is processed by the templating engine.

Below is an example that uses the `network.hw_addr` function to retrieve the MAC address for eth0:

```
salt['network.hw_addr']('eth0')
```

To examine the possible arguments to each execution module function, one can examine the module reference documentation:

Advanced SLS module syntax

Lastly, we will cover some incredibly useful techniques for more complex State trees.

Include declaration

A previous example showed how to spread a Salt tree across several files. Similarly, *Requisites and Other Global State Arguments* span multiple files by using an *Include declaration*. For example:

`python/python-libs.sls:`

```
python-dateutil:
  pkg.installed
```

`python/django.sls:`

```
include:
  - python.python-libs

django:
  pkg.installed:
    - require:
      - pkg: python-dateutil
```

Extend declaration

You can modify previous declarations by using an *Extend declaration*. For example the following modifies the Apache tree to also restart Apache when the vhosts file is changed:

`apache/apache.sls:`

```
apache:
  pkg.installed
```

`apache/mywebsite.sls:`

```
include:
  - apache.apache

extend:
```

(continues on next page)

(continued from previous page)

```
  apache:
    service:
      - running
      - watch:
        - file: /etc/httpd/extra/httpd-vhosts.conf

/etc/httpd/extra/httpd-vhosts.conf:
  file.managed:
    - source: salt://apache/httpd-vhosts.conf
```

Using extend with require or watch

The `extend` statement works differently for `require` or `watch`. It appends to, rather than replacing the requisite component.

Name declaration

You can override the *ID declaration* by using a *Name declaration*. For example, the previous example is a bit more maintainable if rewritten as follows:

`apache/mywebsite.sls`:

```
include:
  - apache.apache

extend:
  apache:
    service:
      - running
      - watch:
        - file: mywebsite

mywebsite:
  file.managed:
    - name: /etc/httpd/extra/httpd-vhosts.conf
    - source: salt://apache/httpd-vhosts.conf
```

Names declaration

Even more powerful is using a *Names declaration* to override the *ID declaration* for multiple states at once. This often can remove the need for looping in a template. For example, the first example in this tutorial can be rewritten without the loop:

```
stooges:
  user.present:
    - names:
      - moe
      - larry
      - curly
```

Next steps

In *part 4* we will discuss how to use salt's `file_roots` to set up a workflow in which states can be ”promoted” from dev, to QA, to production.

4.9.20 States tutorial, part 4

Note: This tutorial builds on topics covered in *part 1*, *part 2*, and *part 3*. It is recommended that you begin there.

This part of the tutorial will show how to use salt's `file_roots` to set up a workflow in which states can be ”promoted” from dev, to QA, to production.

Salt fileserver path inheritance

Salt's fileserver allows for more than one root directory per environment, like in the below example, which uses both a local directory and a secondary location shared to the salt master via NFS:

```
# In the master config file (/etc/salt/master)
file_roots:
  base:
    - /srv/salt
    - /mnt/salt-nfs/base
```

Salt's fileserver collapses the list of root directories into a single virtual environment containing all files from each root. If the same file exists at the same relative path in more than one root, then the top-most match ”wins”. For example, if `/srv/salt/foo.txt` and `/mnt/salt-nfs/base/foo.txt` both exist, then `salt://foo.txt` will point to `/srv/salt/foo.txt`.

Note: When using multiple fileserver backends, the order in which they are listed in the `fileserver_backend` parameter also matters. If both `roots` and `git` backends contain a file with the same relative path, and `roots` appears before `git` in the `fileserver_backend` list, then the file in `roots` will ”win”, and the file in gitfs will be ignored.

A more thorough explanation of how Salt's modular fileserver works can be found *here*. We recommend reading this.

Environment configuration

Configure a multiple-environment setup like so:

```
file_roots:
  base:
    - /srv/salt/prod
  qa:
    - /srv/salt/qa
    - /srv/salt/prod
  dev:
    - /srv/salt/dev
    - /srv/salt/qa
    - /srv/salt/prod
```

Given the path inheritance described above, files within `/srv/salt/prod` would be available in all environments. Files within `/srv/salt/qa` would be available in both `qa`, and `dev`. Finally, the files within `/srv/salt/dev` would only be available within the `dev` environment.

Based on the order in which the roots are defined, new files/states can be placed within `/srv/salt/dev`, and pushed out to the dev hosts for testing.

Those files/states can then be moved to the same relative path within `/srv/salt/qa`, and they are now available only in the `dev` and `qa` environments, allowing them to be pushed to QA hosts and tested.

Finally, if moved to the same relative path within `/srv/salt/prod`, the files are now available in all three environments.

Requesting files from specific fileserver environments

See *here* for documentation on how to request files from specific environments.

Practical Example

As an example, consider a simple website, installed to `/var/www/foobarcom`. Below is a top.sls that can be used to deploy the website:

`/srv/salt/prod/top.sls:`

```
base:
  'web*prod*':
    - webserver.foobarcom
qa:
  'web*qa*':
    - webserver.foobarcom
dev:
  'web*dev*':
    - webserver.foobarcom
```

Using pillar, roles can be assigned to the hosts:

`/srv/pillar/top.sls:`

```
base:
  'web*prod*':
    - webserver.prod
  'web*qa*':
    - webserver.qa
  'web*dev*':
    - webserver.dev
```

`/srv/pillar/webserver/prod.sls:`

```
webserver_role: prod
```

`/srv/pillar/webserver/qa.sls:`

```
webserver_role: qa
```

`/srv/pillar/webserver/dev.sls:`

```
webserver_role: dev
```

And finally, the SLS to deploy the website:

`/srv/salt/prod/webserver/foobarcom.sls:`

```
{% if pillar.get('webserver_role', '') %}
/var/www/foobarcom:
  file.recurse:
    - source: salt://webserver/src/foobarcom
    - env: {{ pillar['webserver_role'] }}
    - user: www
    - group: www
    - dir_mode: 755
    - file_mode: 644
{% endif %}
```

Given the above SLS, the source for the website should initially be placed in `/srv/salt/dev/webserver/src/foobarcom`.

First, let's deploy to dev. Given the configuration in the top file, this can be done using *`state.apply`*:

```
salt --pillar 'webserver_role:dev' state.apply
```

However, in the event that it is not desirable to apply all states configured in the top file (which could be likely in more complex setups), it is possible to apply just the states for the `foobarcom` website, by invoking *`state.apply`* with the desired SLS target as an argument:

```
salt --pillar 'webserver_role:dev' state.apply webserver.foobarcom
```

Once the site has been tested in dev, then the files can be moved from `/srv/salt/dev/webserver/src/foobarcom` to `/srv/salt/qa/webserver/src/foobarcom`, and deployed using the following:

```
salt --pillar 'webserver_role:qa' state.apply webserver.foobarcom
```

Finally, once the site has been tested in qa, then the files can be moved from `/srv/salt/qa/webserver/src/foobarcom` to `/srv/salt/prod/webserver/src/foobarcom`, and deployed using the following:

```
salt --pillar 'webserver_role:prod' state.apply webserver.foobarcom
```

Thanks to Salt's fileserver inheritance, even though the files have been moved to within `/srv/salt/prod`, they are still available from the same `salt://` URI in both the qa and dev environments.

Continue Learning

The best way to continue learning about Salt States is to read through the *reference documentation* and to look through examples of existing state trees. Many pre-configured state trees can be found on GitHub in the saltstack-formulas collection of repositories.

If you have any questions, suggestions, or just want to chat with other people who are using Salt, we have a very *active community* and we'd love to hear from you.

In addition, by continuing to the *Orchestrate Runner* docs, you can learn about the powerful orchestration of which Salt is capable.

4.9.21 States Tutorial, Part 5 - Orchestration with Salt

This was moved to *Orchestrate Runner*.

4.9.22 Syslog-ng usage

Overview

Syslog_ng state module is for generating syslog-ng configurations. You can do the following things:

- generate syslog-ng configuration from YAML,
- use non-YAML configuration,
- start, stop or reload syslog-ng.

There is also an execution module, which can check the syntax of the configuration, get the version and other information about syslog-ng.

Configuration

Users can create syslog-ng configuration statements with the `syslog_ng.config` function. It requires a *name* and a *config* parameter. The *name* parameter determines the name of the generated statement and the *config* parameter holds a parsed YAML structure.

A statement can be declared in the following forms (both are equivalent):

```
source.s_localhost:
  syslog_ng.config:
    - config:
        - tcp:
          - ip: "127.0.0.1"
          - port: 1233
```

```
s_localhost:
  syslog_ng.config:
    - config:
        source:
          - tcp:
            - ip: "127.0.0.1"
            - port: 1233
```

The first one is called short form, because it needs less typing. Users can use lists and dictionaries to specify their configuration. The format is quite self describing and there are more examples [at the end](#examples) of this document.

Quotation

The quotation can be tricky sometimes but here are some rules to follow:

- when a string meant to be `"string"` in the generated configuration, it should be like `'"string"'` in the YAML document
- similarly, users should write `"'string'"` to get `'string'` in the generated configuration

Full example

The following configuration is an example, how a complete syslog-ng configuration looks like:

```
# Set the location of the configuration file
set_location:
  module.run:
    - name: syslog_ng.set_config_file
    - m_name: "/home/tibi/install/syslog-ng/etc/syslog-ng.conf"

# The syslog-ng and syslog-ng-ctl binaries are here. You needn't use
# this method if these binaries can be found in a directory in your PATH.
set_bin_path:
  module.run:
    - name: syslog_ng.set_binary_path
    - m_name: "/home/tibi/install/syslog-ng/sbin"

# Writes the first lines into the config file, also erases its previous
# content
write_version:
  module.run:
    - name: syslog_ng.write_version
    - m_name: "3.6"

# There is a shorter form to set the above variables
set_variables:
  module.run:
    - name: syslog_ng.set_parameters
    - version: "3.6"
    - binary_path: "/home/tibi/install/syslog-ng/sbin"
    - config_file: "/home/tibi/install/syslog-ng/etc/syslog-ng.conf"


# Some global options
options.global_options:
  syslog_ng.config:
    - config:
        - time_reap: 30
        - mark_freq: 10
        - keep_hostname: "yes"

source.s_localhost:
  syslog_ng.config:
    - config:
        - tcp:
          - ip: "127.0.0.1"
          - port: 1233

destination.d_log_server:
  syslog_ng.config:
    - config:
        - tcp:
          - "127.0.0.1"
          - port: 1234

log.l_log_to_central_server:
  syslog_ng.config:
    - config:
```

(continues on next page)

(continued from previous page)

```
        - source: s_localhost
        - destination: d_log_server

some_comment:
  module.run:
    - name: syslog_ng.write_config
    - config: |
        # Multi line
        # comment

# Another mode to use comments or existing configuration snippets
config.other_comment_form:
  syslog_ng.config:
    - config: |
        # Multi line
        # comment
```

The `syslog_ng.reloaded` function can generate syslog-ng configuration from YAML. If the statement (source, destination, parser, etc.) has a name, this function uses the id as the name, otherwise (log statement) it's purpose is like a mandatory comment.

After execution this example the syslog_ng state will generate this file:

```
#Generated by Salt on 2014-08-18 00:11:11
@version: 3.6

options {
    time_reap(
        30
    );
    mark_freq(
        10
    );
    keep_hostname(
        yes
    );
};


source s_localhost {
    tcp(
        ip(
            127.0.0.1
        ),
        port(
            1233
        )
    );
};


destination d_log_server {
    tcp(
        127.0.0.1,
        port(
            1234
        )
```

(continues on next page)

(continued from previous page)

```
    );
};


log {
    source(
        s_localhost
    );
    destination(
        d_log_server
    );
};


# Multi line
# comment


# Multi line
# comment
```

Users can include arbitrary texts in the generated configuration with using the `config` statement (see the example above).

Syslog_ng module functions

You can use *syslog_ng.set_binary_path* to set the directory which contains the syslog-ng and syslog-ng-ctl binaries. If this directory is in your PATH, you don't need to use this function. There is also a *syslog_ng.set_config_file* function to set the location of the configuration file.

Examples

Simple source

```
source s_tail {
 file(
   "/var/log/apache/access.log",
   follow_freq(1),
   flags(no-parse, validate-utf8)
 );
};
```

```
s_tail:
  # Salt will call the source function of syslog_ng module
  syslog_ng.config:
    - config:
        source:
          - file:
            - file: ''"/var/log/apache/access.log"''
            - follow_freq : 1
            - flags:
              - no-parse
              - validate-utf8
```

OR

```
s_tail:
  syslog_ng.config:
    - config:
        source:
            - file:
              - '''"/var/log/apache/access.log"'''
              - follow_freq : 1
              - flags:
                - no-parse
                - validate-utf8
```

OR

```
source.s_tail:
  syslog_ng.config:
    - config:
        - file:
          - '''"/var/log/apache/access.log"'''
          - follow_freq : 1
          - flags:
            - no-parse
            - validate-utf8
```

Complex source

```
source s_gsoc2014 {
 tcp(
   ip("0.0.0.0"),
   port(1234),
   flags(no-parse)
 );
};
```

```
s_gsoc2014:
  syslog_ng.config:
    - config:
        source:
          - tcp:
            - ip: 0.0.0.0
            - port: 1234
            - flags: no-parse
```

Filter

```
filter f_json {
 match(
   "@json:"
 );
};
```

```
f_json:
  syslog_ng.config:
    - config:
        filter:
          - match:
            - '''"@json:"'''
```

Template

```
template t_demo_filetemplate {
 template(
   "$ISODATE $HOST $MSG "
 );
 template_escape(
   no
 );
};
```

```
t_demo_filetemplate:
  syslog_ng.config:
    -config:
        template:
          - template:
            - '"$ISODATE $HOST $MSG\n"'
          - template_escape:
            - "no"
```

Rewrite

```
rewrite r_set_message_to_MESSAGE {
 set(
   "${.json.message}",
   value("$MESSAGE")
 );
};
```

```
r_set_message_to_MESSAGE:
  syslog_ng.config:
    - config:
        rewrite:
          - set:
            - '"${.json.message}"'
            - value : '"$MESSAGE"'
```

Global options

```
options {
   time_reap(30);
   mark_freq(10);
   keep_hostname(yes);
};
```

```
global_options:
  syslog_ng.config:
    - config:
        options:
          - time_reap: 30
          - mark_freq: 10
          - keep_hostname: "yes"
```

Log

```
log {
 source(s_gsoc2014);
 junction {
  channel {
   filter(f_json);
   parser(p_json);
   rewrite(r_set_json_tag);
   rewrite(r_set_message_to_MESSAGE);
   destination {
    file(
      "/tmp/json-input.log",
      template(t_gsoc2014)
    );
   };
   flags(final);
  };
  channel {
   filter(f_not_json);
   parser {
    syslog-parser(

    );
   };
   rewrite(r_set_syslog_tag);
   flags(final);
  };
 };
 destination {
  file(
    "/tmp/all.log",
    template(t_gsoc2014)
  );
 };
};
```

```
l_gsoc2014:
  syslog_ng.config:
    - config:
        log:
          - source: s_gsoc2014
          - junction:
            - channel:
              - filter: f_json
              - parser: p_json
              - rewrite: r_set_json_tag
```

(continues on next page)

(continued from previous page)

```
              - rewrite: r_set_message_to_MESSAGE
              - destination:
                - file:
                  - '"/tmp/json-input.log"'
                  - template: t_gsoc2014
              - flags: final
            - channel:
              - filter: f_not_json
              - parser:
                - syslog-parser: []
              - rewrite: r_set_syslog_tag
              - flags: final
          - destination:
            - file:
              - "/tmp/all.log"
              - template: t_gsoc2014
```

4.9.23 Salt in 10 Minutes

Note: Welcome to SaltStack! I am excited that you are interested in Salt and starting down the path to better infrastructure management. I developed (and am continuing to develop) Salt with the goal of making the best software available to manage computers of almost any kind. I hope you enjoy working with Salt and that the software can solve your real world needs!

- Thomas S Hatch
- Salt creator and Chief Developer
- CTO of SaltStack, Inc.

Getting Started

What is Salt?

Salt is a different approach to infrastructure management, founded on the idea that high-speed communication with large numbers of systems can open up new capabilities. This approach makes Salt a powerful multitasking system that can solve many specific problems in an infrastructure.

The backbone of Salt is the remote execution engine, which creates a high-speed, secure and bi-directional communication net for groups of systems. On top of this communication system, Salt provides an extremely fast, flexible, and easy-to-use configuration management system called `Salt States`.

Installing Salt

SaltStack has been made to be very easy to install and get started. The *installation documents* contain instructions for all supported platforms.

Starting Salt

Salt functions on a master/minion topology. A master server acts as a central control bus for the clients, which are called `minions`. The minions connect back to the master.

Setting Up the Salt Master

Turning on the Salt Master is easy – just turn it on! The default configuration is suitable for the vast majority of installations. The Salt Master can be controlled by the local Linux/Unix service manager:

On Systemd based platforms (newer Debian, openSUSE, Fedora):

```
systemctl start salt-master
```

On Upstart based systems (Ubuntu, older Fedora/RHEL):

```
service salt-master start
```

On SysV Init systems (Gentoo, older Debian etc.):

```
/etc/init.d/salt-master start
```

Alternatively, the Master can be started directly on the command-line:

```
salt-master -d
```

The Salt Master can also be started in the foreground in debug mode, thus greatly increasing the command output:

```
salt-master -l debug
```

The Salt Master needs to bind to two TCP network ports on the system. These ports are `4505` and `4506`. For more in depth information on firewalling these ports, the firewall tutorial is available *here*.

Finding the Salt Master

When a minion starts, by default it searches for a system that resolves to the `salt` hostname on the network. If found, the minion initiates the handshake and key authentication process with the Salt master. This means that the easiest configuration approach is to set internal DNS to resolve the name `salt` back to the Salt Master IP.

Otherwise, the minion configuration file will need to be edited so that the configuration option `master` points to the DNS name or the IP of the Salt Master:

Note: The default location of the configuration files is `/etc/salt`. Most platforms adhere to this convention, but platforms such as FreeBSD and Microsoft Windows place this file in different locations.

`/etc/salt/minion:`

```
master: saltmaster.example.com
```

Setting up a Salt Minion

Note: The Salt Minion can operate with or without a Salt Master. This walk-through assumes that the minion will be connected to the master, for information on how to run a master-less minion please see the master-less quick-start guide:

Masterless Minion Quickstart

Now that the master can be found, start the minion in the same way as the master; with the platform init system or via the command line directly:

As a daemon:

```
salt-minion -d
```

In the foreground in debug mode:

```
salt-minion -l debug
```

When the minion is started, it will generate an `id` value, unless it has been generated on a previous run and cached (in `/etc/salt/minion_id` by default). This is the name by which the minion will attempt to authenticate to the master. The following steps are attempted, in order to try to find a value that is not `localhost`:

1. The Python function `socket.getfqdn()` is run
2. `/etc/hostname` is checked (non-Windows only)
3. `/etc/hosts` (`%WINDIR%\system32\drivers\etc\hosts` on Windows hosts) is checked for hostnames that map to anything within **127.0.0.0/8**.

If none of the above are able to produce an id which is not `localhost`, then a sorted list of IP addresses on the minion (excluding any within **127.0.0.0/8**) is inspected. The first publicly-routable IP address is used, if there is one. Otherwise, the first privately-routable IP address is used.

If all else fails, then `localhost` is used as a fallback.

Note: Overriding the `id`

The minion id can be manually specified using the *`id`* parameter in the minion config file. If this configuration value is specified, it will override all other sources for the `id`.

Now that the minion is started, it will generate cryptographic keys and attempt to connect to the master. The next step is to venture back to the master server and accept the new minion's public key.

Using salt-key

Salt authenticates minions using public-key encryption and authentication. For a minion to start accepting commands from the master, the minion keys need to be accepted by the master.

The `salt-key` command is used to manage all of the keys on the master. To list the keys that are on the master:

```
salt-key -L
```

The keys that have been rejected, accepted, and pending acceptance are listed. The easiest way to accept the minion key is to accept all pending keys:

```
salt-key -A
```

Note: Keys should be verified! Print the master key fingerprint by running `salt-key -F master` on the Salt master. Copy the `master.pub` fingerprint from the Local Keys section, and then set this value as the *master_finger* in the minion configuration file. Restart the Salt minion.

On the master, run `salt-key -f minion-id` to print the fingerprint of the minion's public key that was received by the master. On the minion, run `salt-call key.finger --local` to print the fingerprint of the minion key.

On the master:

```
# salt-key -f foo.domain.com
Unaccepted Keys:
foo.domain.com:  39:f9:e4:8a:aa:74:8d:52:1a:ec:92:03:82:09:c8:f9
```

On the minion:

```
# salt-call key.finger --local
local:
    39:f9:e4:8a:aa:74:8d:52:1a:ec:92:03:82:09:c8:f9
```

If they match, approve the key with `salt-key -a foo.domain.com`.

Sending the First Commands

Now that the minion is connected to the master and authenticated, the master can start to command the minion.

Salt commands allow for a vast set of functions to be executed and for specific minions and groups of minions to be targeted for execution.

The `salt` command is comprised of command options, target specification, the function to execute, and arguments to the function.

A simple command to start with looks like this:

```
salt '*' test.version
```

The `*` is the target, which specifies all minions.

`test.version` tells the minion to run the *test.version* function.

In the case of `test.version`, `test` refers to a *execution module*. `version` refers to the *version* function contained in the aforementioned `test` module.

Note: Execution modules are the workhorses of Salt. They do the work on the system to perform various tasks, such as manipulating files and restarting services.

The result of running this command will be the master instructing all of the minions to execute *test.version* in parallel and return the result. Using *test.version* is a good way of confirming that a minion is connected, and reaffirm to the user the salt version(s) they have installed on the minions.

Note: Each minion registers itself with a unique minion ID. This ID defaults to the minion's hostname, but can be explicitly defined in the minion config as well by using the `id` parameter.

Of course, there are hundreds of other modules that can be called just as `test.version` can. For example, the following would return disk usage on all targeted minions:

```
salt '*' disk.usage
```

Getting to Know the Functions

Salt comes with a vast library of functions available for execution, and Salt functions are self-documenting. To see what functions are available on the minions execute the `sys.doc` function:

```
salt '*' sys.doc
```

This will display a very large list of available functions and documentation on them.

Note: Module documentation is also available *on the web*.

These functions cover everything from shelling out to package management to manipulating database servers. They comprise a powerful system management API which is the backbone to Salt configuration management and many other aspects of Salt.

Note: Salt comes with many plugin systems. The functions that are available via the `salt` command are called *Execution Modules*.

Helpful Functions to Know

The `cmd` module contains functions to shell out on minions, such as `cmd.run` and `cmd.run_all`:

```
salt '*' cmd.run 'ls -l /etc'
```

The `pkg` functions automatically map local system package managers to the same salt functions. This means that `pkg.install` will install packages via `yum` on Red Hat based systems, `apt` on Debian systems, etc.:

```
salt '*' pkg.install vim
```

Note: Some custom Linux spins and derivatives of other distributions are not properly detected by Salt. If the above command returns an error message saying that `pkg.install` is not available, then you may need to override the pkg provider. This process is explained *here*.

The `network.interfaces` function will list all interfaces on a minion, along with their IP addresses, netmasks, MAC addresses, etc:

```
salt '*' network.interfaces
```

Changing the Output Format

The default output format used for most Salt commands is called the `nested` outputter, but there are several other outputters that can be used to change the way the output is displayed. For instance, the `pprint` outputter can be used to display the return data using Python's `pprint` module:

```
root@saltmaster:~# salt myminion grains.item pythonpath --out=pprint
{'myminion': {'pythonpath': ['/usr/lib64/python2.7',
                             '/usr/lib/python2.7/plat-linux2',
                             '/usr/lib64/python2.7/lib-tk',
                             '/usr/lib/python2.7/lib-tk',
                             '/usr/lib/python2.7/site-packages',
                             '/usr/lib/python2.7/site-packages/gst-0.10',
                             '/usr/lib/python2.7/site-packages/gtk-2.0']}}
```

The full list of Salt outputters, as well as example output, can be found *here*.

`salt-call`

The examples so far have described running commands from the Master using the `salt` command, but when troubleshooting it can be more beneficial to login to the minion directly and use `salt-call`.

Doing so allows you to see the minion log messages specific to the command you are running (which are *not* part of the return data you see when running the command from the Master using `salt`), making it unnecessary to tail the minion log. More information on `salt-call` and how to use it can be found *here*.

Grains

Salt uses a system called *Grains* to build up static data about minions. This data includes information about the operating system that is running, CPU architecture and much more. The grains system is used throughout Salt to deliver platform data to many components and to users.

Grains can also be statically set, this makes it easy to assign values to minions for grouping and managing.

A common practice is to assign grains to minions to specify what the role or roles a minion might be. These static grains can be set in the minion configuration file or via the `grains.setval` function.

Targeting

Salt allows for minions to be targeted based on a wide range of criteria. The default targeting system uses globular expressions to match minions, hence if there are minions named `larry1`, `larry2`, `curly1`, and `curly2`, a glob of `larry*` will match `larry1` and `larry2`, and a glob of `*1` will match `larry1` and `curly1`.

Many other targeting systems can be used other than globs, these systems include:

Regular Expressions Target using PCRE-compliant regular expressions

Grains Target based on grains data: *Targeting with Grains*

Pillar Target based on pillar data: *Targeting with Pillar*

IP Target based on IP address/subnet/range

Compound Create logic to target based on multiple targets: *Targeting with Compound*

Nodegroup Target with nodegroups: *Targeting with Nodegroup*

The concepts of targets are used on the command line with Salt, but also function in many other areas as well, including the state system and the systems used for ACLs and user permissions.

Passing in Arguments

Many of the functions available accept arguments which can be passed in on the command line:

```
salt '*' pkg.install vim
```

This example passes the argument `vim` to the pkg.install function. Since many functions can accept more complex input than just a string, the arguments are parsed through YAML, allowing for more complex data to be sent on the command line:

```
salt '*' test.echo 'foo: bar'
```

In this case Salt translates the string 'foo: bar' into the dictionary "{'foo': 'bar'}"

Note: Any line that contains a newline will not be parsed by YAML.

Salt States

Now that the basics are covered the time has come to evaluate `States`. Salt `States`, or the `State System` is the component of Salt made for configuration management.

The state system is already available with a basic Salt setup, no additional configuration is required. States can be set up immediately.

Note: Before diving into the state system, a brief overview of how states are constructed will make many of the concepts clearer. Salt states are based on data modeling and build on a low level data structure that is used to execute each state function. Then more logical layers are built on top of each other.

The high layers of the state system which this tutorial will cover consists of everything that needs to be known to use states, the two high layers covered here are the *sls* layer and the highest layer *highstate*.

Understanding the layers of data management in the State System will help with understanding states, but they never need to be used. Just as understanding how a compiler functions assists when learning a programming language, understanding what is going on under the hood of a configuration management system will also prove to be a valuable asset.

The First SLS Formula

The state system is built on SLS (SaLt State) formulas. These formulas are built out in files on Salt's file server. To make a very basic SLS formula open up a file under /srv/salt named vim.sls. The following state ensures that vim is installed on a system to which that state has been applied.

`/srv/salt/vim.sls:`

```
vim:
  pkg.installed
```

Now install vim on the minions by calling the SLS directly:

```
salt '*' state.apply vim
```

This command will invoke the state system and run the `vim` SLS.

Now, to beef up the vim SLS formula, a `vimrc` can be added:

`/srv/salt/vim.sls:`

```
vim:
  pkg.installed: []

/etc/vimrc:
  file.managed:
    - source: salt://vimrc
    - mode: 644
    - user: root
    - group: root
```

Now the desired `vimrc` needs to be copied into the Salt file server to `/srv/salt/vimrc`. In Salt, everything is a file, so no path redirection needs to be accounted for. The `vimrc` file is placed right next to the `vim.sls` file. The same command as above can be executed to all the vim SLS formulas and now include managing the file.

Note: Salt does not need to be restarted/reloaded or have the master manipulated in any way when changing SLS formulas. They are instantly available.

Adding Some Depth

Obviously maintaining SLS formulas right in a single directory at the root of the file server will not scale out to reasonably sized deployments. This is why more depth is required. Start by making an nginx formula a better way, make an nginx subdirectory and add an init.sls file:

`/srv/salt/nginx/init.sls:`

```
nginx:
  pkg.installed: []
  service.running:
    - require:
      - pkg: nginx
```

A few concepts are introduced in this SLS formula.

First is the service statement which ensures that the `nginx` service is running.

Of course, the nginx service can't be started unless the package is installed – hence the `require` statement which sets up a dependency between the two.

The `require` statement makes sure that the required component is executed before and that it results in success.

Note: The *require* option belongs to a family of options called *requisites*. Requisites are a powerful component of Salt States, for more information on how requisites work and what is available see: *Requisites*

Also evaluation ordering is available in Salt as well: *Ordering States*

This new sls formula has a special name – `init.sls`. When an SLS formula is named `init.sls` it inherits the name of the directory path that contains it. This formula can be referenced via the following command:

```
salt '*' state.apply nginx
```

Note: *`state.apply`* is just another remote execution function, just like *`test.version`* or *`disk.usage`*. It simply takes the name of an SLS file as an argument.

Now that subdirectories can be used, the `vim.sls` formula can be cleaned up. To make things more flexible, move the `vim.sls` and vimrc into a new subdirectory called `edit` and change the `vim.sls` file to reflect the change:

`/srv/salt/edit/vim.sls`:

```
vim:
  pkg.installed

/etc/vimrc:
  file.managed:
    - source: salt://edit/vimrc
    - mode: 644
    - user: root
    - group: root
```

Only the source path to the vimrc file has changed. Now the formula is referenced as `edit.vim` because it resides in the edit subdirectory. Now the edit subdirectory can contain formulas for emacs, nano, joe or any other editor that may need to be deployed.

Next Reading

Two walk-throughs are specifically recommended at this point. First, a deeper run through States, followed by an explanation of Pillar.

1. *Starting States*
2. *Pillar Walkthrough*

An understanding of Pillar is extremely helpful in using States.

Getting Deeper Into States

Two more in-depth States tutorials exist, which delve much more deeply into States functionality.

1. *How Do I Use Salt States?*, covers much more to get off the ground with States.
2. The *States Tutorial* also provides a fantastic introduction.

These tutorials include much more in-depth information including templating SLS formulas etc.

So Much More!

This concludes the initial Salt walk-through, but there are many more things still to learn! These documents will cover important core aspects of Salt:

- *Pillar*
- *Job Management*

A few more tutorials are also available:

- *Remote Execution Tutorial*
- *Standalone Minion*

This still is only scratching the surface, many components such as the reactor and event systems, extending Salt, modular components and more are not covered here. For an overview of all Salt features and documentation, look at the *Table of Contents*.

4.9.24 Salt's Test Suite: An Introduction

Note: This tutorial makes a couple of assumptions. The first assumption is that you have a basic knowledge of Salt. To get up to speed, check out the *Salt Walkthrough*.

The second assumption is that your Salt development environment is already configured and that you have a basic understanding of contributing to the Salt codebase. If you're unfamiliar with either of these topics, please refer to the *Installing Salt for Development* and the *Contributing* pages, respectively.

Salt comes with a powerful integration and unit test suite. The test suite allows for the fully automated run of integration and/or unit tests from a single interface.

Salt's test suite is located under the `tests` directory in the root of Salt's code base and is divided into two main types of tests: *unit tests and integration tests*. The `unit` and `integration` sub-test-suites are located in the `tests` directory, which is where the majority of Salt's test cases are housed.

Getting Set Up For Tests

There are a couple of requirements, in addition to Salt's requirements, that need to be installed in order to run Salt's test suite. You can install these additional requirements using the files located in the `salt/requirements` directory, depending on your relevant version of Python:

```
pip install -r requirements/dev_python27.txt
pip install -r requirements/dev_python34.txt
```

To be able to run integration tests which utilizes ZeroMQ transport, you also need to install additional requirements for it. Make sure you have installed the C/C++ compiler and development libraries and header files needed for your Python version.

This is an example for RedHat-based operating systems:

```
yum install gcc gcc-c++ python-devel
pip install -r requirements/zeromq.txt
```

On Debian, Ubuntu or their derivatives run the following commands:

```
apt-get install build-essential python-dev
pip install -r requirements/zeromq.txt
```

This will install the latest `pycrypto` and `pyzmq` (with bundled `libzmq`) Python modules required for running integration tests suite.

Test Directory Structure

As noted in the introduction to this tutorial, Salt's test suite is located in the `tests` directory in the root of Salt's code base. From there, the tests are divided into two groups `integration` and `unit`. Within each of these directories, the directory structure roughly mirrors the directory structure of Salt's own codebase. For example, the files inside `tests/integration/modules` contains tests for the files located within `salt/modules`.

Note: `tests/integration` and `tests/unit` are the only directories discussed in this tutorial. With the exception of the `tests/runtests.py` file, which is used below in the *Running the Test Suite* section, the other directories and files located in `tests` are outside the scope of this tutorial.

Integration vs. Unit

Given that Salt's test suite contains two powerful, though very different, testing approaches, when should you write integration tests and when should you write unit tests?

Integration tests use Salt masters, minions, and a syndic to test salt functionality directly and focus on testing the interaction of these components. Salt's integration test runner includes functionality to run Salt execution modules, runners, states, shell commands, salt-ssh commands, salt-api commands, and more. This provides a tremendous ability to use Salt to test itself and makes writing such tests a breeze. Integration tests are the preferred method of testing Salt functionality when possible.

Unit tests do not spin up any Salt daemons, but instead find their value in testing singular implementations of individual functions. Instead of testing against specific interactions, unit tests should be used to test a function's logic. Unit tests should be used to test a function's exit point(s) such as any `return` or `raises` statements.

Unit tests are also useful in cases where writing an integration test might not be possible. While the integration test suite is extremely powerful, unfortunately at this time, it does not cover all functional areas of Salt's ecosystem. For example, at the time of this writing, there is not a way to write integration tests for Proxy Minions. Since the test runner will need to be adjusted to account for Proxy Minion processes, unit tests can still provide some testing support in the interim by testing the logic contained inside Proxy Minion functions.

Running the Test Suite

Once all of the *requirements* are installed, the `runtests.py` file in the `salt/tests` directory is used to instantiate Salt's test suite:

```
python tests/runtests.py [OPTIONS]
```

The command above, if executed without any options, will run the entire suite of integration and unit tests. Some tests require certain flags to run, such as destructive tests. If these flags are not included, then the test suite will only perform the tests that don't require special attention.

At the end of the test run, you will see a summary output of the tests that passed, failed, or were skipped.

The test runner also includes a `--help` option that lists all of the various command line options:

```
python tests/runtests.py --help
```

You can also call the test runner as an executable:

```
./tests/runtests.py --help
```

Running Integration Tests

Salt's set of integration tests use Salt to test itself. The integration portion of the test suite includes some built-in Salt daemons that will spin up in preparation of the test run. This list of Salt daemon processes includes:

- 2 Salt Masters
- 2 Salt Minions
- 1 Salt Syndic

These various daemons are used to execute Salt commands and functionality within the test suite, allowing you to write tests to assert against expected or unexpected behaviors.

A simple example of a test utilizing a typical master/minion execution module command is the test for the `test_ping` function in the `tests/integration/modules/test_test.py` file:

```
def test_ping(self):
    '''
    test.ping
    '''
    self.assertTrue(self.run_function('test.ping'))
```

The test above is a very simple example where the `test.ping` function is executed by Salt's test suite runner and is asserting that the minion returned with a `True` response.

Test Selection Options

If you look in the output of the `--help` command of the test runner, you will see a section called `Tests Selection Options`. The options under this section contain various subsections of the integration test suite such as `--modules`, `--ssh`, or `--states`. By selecting any one of these options, the test daemons will spin up and the integration tests in the named subsection will run.

```
./tests/runtests.py --modules
```

Note: The testing subsections listed in the `Tests Selection Options` of the `--help` output *only* apply to the integration tests. They do not run unit tests.

Running Unit Tests

While `./tests/runtests.py` executes the *entire* test suite (barring any tests requiring special flags), the `--unit` flag can be used to run *only* Salt's unit tests. Salt's unit tests include the tests located in the `tests/unit` directory.

The unit tests do not spin up any Salt testing daemons as the integration tests do and execute very quickly compared to the integration tests.

```
./tests/runtests.py --unit
```

Running Specific Tests

There are times when a specific test file, test class, or even a single, individual test need to be executed, such as when writing new tests. In these situations, the `--name` option should be used.

For running a single test file, such as the pillar module test file in the integration test directory, you must provide the file path using `.` instead of `/` as separators and no file extension:

```
./tests/runtests.py --name=integration.modules.test_pillar
./tests/runtests.py -n integration.modules.test_pillar
```

Some test files contain only one test class while other test files contain multiple test classes. To run a specific test class within the file, append the name of the test class to the end of the file path:

```
./tests/runtests.py --name=integration.modules.test_pillar.PillarModuleTest
./tests/runtests.py -n integration.modules.test_pillar.PillarModuleTest
```

To run a single test within a file, append both the name of the test class the individual test belongs to, as well as the name of the test itself:

```
./tests/runtests.py \
  --name=integration.modules.test_pillar.PillarModuleTest.test_data
./tests/runtests.py \
  -n integration.modules.test_pillar.PillarModuleTest.test_data
```

The `--name` and `-n` options can be used for unit tests as well as integration tests. The following command is an example of how to execute a single test found in the `tests/unit/modules/test_cp.py` file:

```
./tests/runtests.py \
  -n unit.modules.test_cp.CpTestCase.test_get_template_success
```

Writing Tests for Salt

Once you're comfortable running tests, you can now start writing them! Be sure to review the *Integration vs. Unit* section of this tutorial to determine what type of test makes the most sense for the code you're testing.

Note: There are many decorators, naming conventions, and code specifications required for Salt test files. We will not be covering all of the these specifics in this tutorial. Please refer to the testing documentation links listed below in the *Additional Testing Documentation* section to learn more about these requirements.

In the following sections, the test examples assume the "new" test is added to a test file that is already present and regularly running in the test suite and is written with the correct requirements.

Writing Integration Tests

Since integration tests validate against a running environment, as explained in the *Running Integration Tests* section of this tutorial, integration tests are very easy to write and are generally the preferred method of writing Salt tests.

The following integration test is an example taken from the `test.py` file in the `tests/integration/modules` directory. This test uses the `run_function` method to test the functionality of a traditional execution module command.

The `run_function` method uses the integration test daemons to execute a `module.function` command as you would with Salt. The minion runs the function and returns. The test also uses Python's Assert Functions to test that the minion's return is expected.

```
def test_ping(self):
    '''
    test.ping
    '''
    self.assertTrue(self.run_function('test.ping'))
```

Args can be passed in to the `run_function` method as well:

```
def test_echo(self):
    '''
    test.echo
    '''
    self.assertEqual(self.run_function('test.echo', ['text']), 'text')
```

The next example is taken from the `tests/integration/modules/test_aliases.py` file and demonstrates how to pass kwargs to the `run_function` call. Also note that this test uses another salt function to ensure the correct data is present (via the `aliases.set_target` call) before attempting to assert what the `aliases.get_target` call should return.

```
def test_set_target(self):
    '''
    aliases.set_target and aliases.get_target
    '''
    set_ret = self.run_function(
            'aliases.set_target',
            alias='fred',
            target='bob')
    self.assertTrue(set_ret)
    tgt_ret = self.run_function(
            'aliases.get_target',
            alias='fred')
    self.assertEqual(tgt_ret, 'bob')
```

Using multiple Salt commands in this manner provides two useful benefits. The first is that it provides some additional coverage for the `aliases.set_target` function. The second benefit is the call to `aliases.get_target` is not dependent on the presence of any aliases set outside of this test. Tests should not be dependent on the previous execution, success, or failure of other tests. They should be isolated from other tests as much as possible.

While it might be tempting to build out a test file where tests depend on one another before running, this should be avoided. SaltStack recommends that each test should test a single functionality and not rely on other tests. Therefore, when possible, individual tests should also be broken up into singular pieces. These are not hard-and-fast rules, but serve more as recommendations to keep the test suite simple. This helps with debugging code and related tests when failures occur and problems are exposed. There may be instances where large tests use many asserts to set up a use case that protects against potential regressions.

Note: The examples above all use the `run_function` option to test execution module functions in a traditional master/minion environment. To see examples of how to test other common Salt components such as runners, salt-api, and more, please refer to the *Integration Test Class Examples* documentation.

Destructive vs Non-destructive Tests

Since Salt is used to change the settings and behavior of systems, often, the best approach to run tests is to make actual changes to an underlying system. This is where the concept of destructive integration tests comes into play. Tests can be written to alter the system they are running on. This capability is what fills in the gap needed to properly test aspects of system management like package installation.

To write a destructive test, import and use the `destructiveTest` decorator for the test method:

```
import integration
from tests.support.helpers import destructiveTest

class PkgTest(integration.ModuleCase):
    @destructiveTest
    def test_pkg_install(self):
        ret = self.run_function('pkg.install', name='finch')
        self.assertSaltTrueReturn(ret)
        ret = self.run_function('pkg.purge', name='finch')
        self.assertSaltTrueReturn(ret)
```

Writing Unit Tests

As explained in the *Integration vs. Unit* section above, unit tests should be written to test the *logic* of a function. This includes focusing on testing `return` and `raises` statements. Substantial effort should be made to mock external resources that are used in the code being tested.

External resources that should be mocked include, but are not limited to, APIs, function calls, external data either globally available or passed in through function arguments, file data, etc. This practice helps to isolate unit tests to test Salt logic. One handy way to think about writing unit tests is to ”block all of the exits”. More information about how to properly mock external resources can be found in Salt's *Unit Test* documentation.

Salt's unit tests utilize Python's mock class as well as MagicMock. The `@patch` decorator is also heavily used when ”blocking all the exits”.

A simple example of a unit test currently in use in Salt is the `test_get_file_not_found` test in the `tests/unit/modules/test_cp.py` file. This test uses the `@patch` decorator and `MagicMock` to mock the return of the call to Salt's `cp.hash_file` execution module function. This ensures that we're testing the `cp.get_file` function directly, instead of inadvertently testing the call to `cp.hash_file`, which is used in `cp.get_file`.

```
def test_get_file_not_found(self):
    '''
    Test if get_file can't find the file.
    '''
    with patch('salt.modules.cp.hash_file', MagicMock(return_value=False)):
        path = 'salt://saltines'
        dest = '/srv/salt/cheese'
        ret = ''
        self.assertEqual(cp.get_file(path, dest), ret)
```

Note that Salt's `cp` module is imported at the top of the file, along with all of the other necessary testing imports. The `get_file` function is then called directed in the testing function, instead of using the `run_function` method as the integration test examples do above.

The call to `cp.get_file` returns an empty string when a `hash_file` isn't found. Therefore, the example above is a good illustration of a unit test ”blocking the exits” via the `@patch` decorator, as well as testing logic via asserting against the `return` statement in the `if` clause.

There are more examples of writing unit tests of varying complexities available in the following docs:

- *Simple Unit Test Example*
- *Complete Unit Test Example*
- *Complex Unit Test Example*

Note: Considerable care should be made to ensure that you're testing something useful in your test functions. It is very easy to fall into a situation where you have mocked so much of the original function that the test results in only asserting against the data you have provided. This results in a poor and fragile unit test.

Checking for Log Messages

To test to see if a given log message has been emitted, the following pattern can be used

```
# Import logging handler
from tests.support.helpers import TstSuiteLoggingHandler

# .. inside test
with TstSuiteLoggingHandler() as handler:
    for message in handler.messages:
        if message.startswith('ERROR: This is the error message we seek'):
            break
        else:
            raise AssertionError('Did not find error message')
```

Automated Test Runs

SaltStack maintains a Jenkins server which can be viewed at https://jenkins.saltstack.com. The tests executed from this Jenkins server create fresh virtual machines for each test run, then execute the destructive tests on the new, clean virtual machine. This allows for the execution of tests across supported platforms.

Additional Testing Documentation

In addition to this tutorial, there are some other helpful resources and documentation that go into more depth on Salt's test runner, writing tests for Salt code, and general Python testing documentation. Please see the follow references for more information:

- *Salt's Test Suite Documentation*
- *Integration Tests*
- *Unit Tests*
- MagicMock
- Python Unittest
- Python's Assert Functions

4.10 Troubleshooting

The intent of the troubleshooting section is to introduce solutions to a number of common issues encountered by users and the tools that are available to aid in developing States and Salt code.

4.10.1 Troubleshooting the Salt Master

If your Salt master is having issues such as minions not returning data, slow execution times, or a variety of other issues, the following links contain details on troubleshooting the most common issues encountered:

Troubleshooting the Salt Master

Running in the Foreground

A great deal of information is available via the debug logging system, if you are having issues with minions connecting or not starting run the master in the foreground:

```
# salt-master -l debug
```

Anyone wanting to run Salt daemons via a process supervisor such as monit, runit, or supervisord, should omit the `-d` argument to the daemons and run them in the foreground.

What Ports does the Master Need Open?

For the master, TCP ports 4505 and 4506 need to be open. If you've put both your Salt master and minion in debug mode and don't see an acknowledgment that your minion has connected, it could very well be a firewall interfering with the connection. See our *firewall configuration* page for help opening the firewall on various platforms.

If you've opened the correct TCP ports and still aren't seeing connections, check that no additional access control system such as SELinux or AppArmor is blocking Salt.

Too many open files

The salt-master needs at least 2 sockets per host that connects to it, one for the Publisher and one for response port. Thus, large installations may, upon scaling up the number of minions accessing a given master, encounter:

```
12:45:29,289 [salt.master    ][INFO    ] Starting Salt worker process 38
Too many open files
sock != -1 (tcp_listener.cpp:335)
```

The solution to this would be to check the number of files allowed to be opened by the user running salt-master (root by default):

```
[root@salt-master ~]# ulimit -n
1024
```

If this value is not equal to at least twice the number of minions, then it will need to be raised. For example, in an environment with 1800 minions, the `nofile` limit should be set to no less than 3600. This can be done by creating the file `/etc/security/limits.d/99-salt.conf`, with the following contents:

```
root        hard    nofile        4096
root        soft    nofile        4096
```

Replace `root` with the user under which the master runs, if different.

If your master does not have an `/etc/security/limits.d` directory, the lines can simply be appended to `/etc/security/limits.conf`.

As with any change to resource limits, it is best to stay logged into your current shell and open another shell to run `ulimit -n` again and verify that the changes were applied correctly. Additionally, if your master is running upstart, it may be necessary to specify the `nofile` limit in `/etc/default/salt-master` if upstart isn't respecting your resource limits:

```
limit nofile 4096 4096
```

Note: The above is simply an example of how to set these values, and you may wish to increase them even further if your Salt master is doing more than just running Salt.

Salt Master Stops Responding

There are known bugs with ZeroMQ versions less than 2.1.11 which can cause the Salt master to not respond properly. If you're running a ZeroMQ version greater than or equal to 2.1.9, you can work around the bug by setting the sysctls `net.core.rmem_max` and `net.core.wmem_max` to 16777216. Next, set the third field in `net.ipv4.tcp_rmem` and `net.ipv4.tcp_wmem` to at least 16777216.

You can do it manually with something like:

```
# echo 16777216 > /proc/sys/net/core/rmem_max
# echo 16777216 > /proc/sys/net/core/wmem_max
# echo "4096 87380 16777216" > /proc/sys/net/ipv4/tcp_rmem
# echo "4096 87380 16777216" > /proc/sys/net/ipv4/tcp_wmem
```

Or with the following Salt state:

```
net.core.rmem_max:
  sysctl:
    - present
    - value: 16777216

net.core.wmem_max:
  sysctl:
    - present
    - value: 16777216

net.ipv4.tcp_rmem:
  sysctl:
    - present
    - value: 4096 87380 16777216

net.ipv4.tcp_wmem:
  sysctl:
    - present
    - value: 4096 87380 16777216
```

Live Python Debug Output

If the master seems to be unresponsive, a SIGUSR1 can be passed to the salt-master threads to display what piece of code is executing. This debug information can be invaluable in tracking down bugs.

To pass a SIGUSR1 to the master, first make sure the master is running in the foreground. Stop the service if it is running as a daemon, and start it in the foreground like so:

```
# salt-master -l debug
```

Then pass the signal to the master when it seems to be unresponsive:

```
# killall -SIGUSR1 salt-master
```

When filing an issue or sending questions to the mailing list for a problem with an unresponsive daemon, be sure to include this information if possible.

Live Salt-Master Profiling

When faced with performance problems one can turn on master process profiling by sending it SIGUSR2.

```
# killall -SIGUSR2 salt-master
```

This will activate `yappi` profiler inside salt-master code, then after some time one must send SIGUSR2 again to stop profiling and save results to file. If run in foreground salt-master will report filename for the results, which are usually located under `/tmp` on Unix-based OSes and `c:\temp` on windows.

Make sure you have yappi installed.

Results can then be analyzed with kcachegrind or similar tool.

On Windows, in the absense of kcachegrind, a simple file-based workflow to create profiling graphs could use gprof2dot, graphviz and this batch file:

```
::
:: Converts callgrind* profiler output to *.pdf, via *.dot
::
@echo off
del *.dot.pdf
for /r %%f in (callgrind*) do (
echo "%%f"
    gprof2dot.exe -f callgrind --show-samples  "%%f" -o "%%f.dot"
    dot.exe "%%f.dot" -Tpdf -O
    del "%%f.dot"
)
```

Commands Time Out or Do Not Return Output

Depending on your OS (this is most common on Ubuntu due to apt-get) you may sometimes encounter times where a *`state.apply`*, or other long running commands do not return output.

By default the timeout is set to 5 seconds. The timeout value can easily be increased by modifying the `timeout` line within your `/etc/salt/master` configuration file.

Having keys accepted for Salt minions that no longer exist or are not reachable also increases the possibility of timeouts, since the Salt master waits for those systems to return command results.

Passing the -c Option to Salt Returns a Permissions Error

Using the `-c` option with the Salt command modifies the configuration directory. When the configuration file is read it will still base data off of the `root_dir` setting. This can result in unintended behavior if you are expecting files such as `/etc/salt/pki` to be pulled from the location specified with `-c`. Modify the `root_dir` setting to address this behavior.

Salt Master Doesn't Return Anything While Running jobs

When a command being run via Salt takes a very long time to return (package installations, certain scripts, etc.) the master may drop you back to the shell. In most situations the job is still running but Salt has exceeded the set timeout before returning. Querying the job queue will provide the data of the job but is inconvenient. This can be resolved by either manually using the `-t` option to set a longer timeout when running commands (by default it is 5 seconds) or by modifying the master configuration file: `/etc/salt/master` and setting the `timeout` value to change the default timeout for all commands, and then restarting the salt-master service.

If a `state.apply` run takes too long, you can find a bottleneck by adding the *`--out=profile`* option.

Salt Master Auth Flooding

In large installations, care must be taken not to overwhealm the master with authentication requests. Several options can be set on the master which mitigate the chances of an authentication flood from causing an interruption in service.

Note: recon_default:

The average number of seconds to wait between reconnection attempts.

recon_max: The maximum number of seconds to wait between reconnection attempts.

recon_randomize: A flag to indicate whether the recon_default value should be randomized.

acceptance_wait_time: The number of seconds to wait for a reply to each authentication request.

random_reauth_delay: The range of seconds across which the minions should attempt to randomize authentication attempts.

auth_timeout: The total time to wait for the authentication process to complete, regardless of the number of attempts.

Running states locally

To debug the states, you can use call locally.

```
salt-call -l trace --local state.highstate
```

The top.sls file is used to map what SLS modules get loaded onto what minions via the state system.

It is located in the file defined in the `file_roots` variable of the salt master configuration file which is defined by found in `CONFIG_DIR/master`, normally `/etc/salt/master`

The default configuration for the `file_roots` is:

```
file_roots:
  base:
    - /srv/salt
```

So the top file is defaulted to the location `/srv/salt/top.sls`

Salt Master Umask

The salt master uses a cache to track jobs as they are published and returns come back. The recommended umask for a salt-master is *022*, which is the default for most users on a system. Incorrect umasks can result in permission-denied errors when the master tries to access files in its cache.

4.10.2 Troubleshooting the Salt Minion

In the event that your Salt minion is having issues, a variety of solutions and suggestions are available. Please refer to the following links for more information:

Troubleshooting the Salt Minion

Running in the Foreground

A great deal of information is available via the debug logging system, if you are having issues with minions connecting or not starting run the minion in the foreground:

```
# salt-minion -l debug
```

Anyone wanting to run Salt daemons via a process supervisor such as monit, runit, or supervisord, should omit the `-d` argument to the daemons and run them in the foreground.

What Ports does the Minion Need Open?

No ports need to be opened on the minion, as it makes outbound connections to the master. If you've put both your Salt master and minion in debug mode and don't see an acknowledgment that your minion has connected, it could very well be a firewall interfering with the connection. See our *firewall configuration* page for help opening the firewall on various platforms.

If you have netcat installed, you can check port connectivity from the minion with the `nc` command:

```
$ nc -v -z salt.master.ip.addr 4505
Connection to salt.master.ip.addr 4505 port [tcp/unknown] succeeded!
$ nc -v -z salt.master.ip.addr 4506
Connection to salt.master.ip.addr 4506 port [tcp/unknown] succeeded!
```

The Nmap utility can also be used to check if these ports are open:

```
# nmap -sS -q -p 4505-4506 salt.master.ip.addr

Starting Nmap 6.40 ( http://nmap.org ) at 2013-12-29 19:44 CST
Nmap scan report for salt.master.ip.addr (10.0.0.10)
Host is up (0.0026s latency).
PORT     STATE  SERVICE
4505/tcp open   unknown
4506/tcp open   unknown
MAC Address: 00:11:22:AA:BB:CC (Intel)

Nmap done: 1 IP address (1 host up) scanned in 1.64 seconds
```

If you've opened the correct TCP ports and still aren't seeing connections, check that no additional access control system such as SELinux or AppArmor is blocking Salt. Tools like tcptraceroute can also be used to determine if an intermediate device or firewall is blocking the needed TCP ports.

Using salt-call

The `salt-call` command was originally developed for aiding in the development of new Salt modules. Since then, many applications have been developed for running any Salt module locally on a minion. These range from the original intent of salt-call (development assistance), to gathering more verbose output from calls like *`state.apply`*.

When initially creating your state tree, it is generally recommended to invoke highstates by running *`state.apply`* directly from the minion with `salt-call`, rather than remotely from the master. This displays far more information about the execution than calling it remotely. For even more verbosity, increase the loglevel using the `-l` argument:

```
# salt-call -l debug state.apply
```

The main difference between using `salt` and using `salt-call` is that `salt-call` is run from the minion, and it only runs the selected function on that minion. By contrast, `salt` is run from the master, and requires you to specify the minions on which to run the command using salt's *targeting system*.

Live Python Debug Output

If the minion seems to be unresponsive, a SIGUSR1 can be passed to the process to display what piece of code is executing. This debug information can be invaluable in tracking down bugs.

To pass a SIGUSR1 to the minion, first make sure the minion is running in the foreground. Stop the service if it is running as a daemon, and start it in the foreground like so:

```
# salt-minion -l debug
```

Then pass the signal to the minion when it seems to be unresponsive:

```
# killall -SIGUSR1 salt-minion
```

When filing an issue or sending questions to the mailing list for a problem with an unresponsive daemon, be sure to include this information if possible.

Multiprocessing in Execution Modules

As is outlined in github issue #6300, Salt cannot use python's multiprocessing pipes and queues from execution modules. Multiprocessing from the execution modules is perfectly viable, it is just necessary to use Salt's event system to communicate back with the process.

The reason for this difficulty is that python attempts to pickle all objects in memory when communicating, and it cannot pickle function objects. Since the Salt loader system creates and manages function objects this causes the pickle operation to fail.

Salt Minion Doesn't Return Anything While Running Jobs Locally

When a command being run via Salt takes a very long time to return (package installations, certain scripts, etc.) the minion may drop you back to the shell. In most situations the job is still running but Salt has exceeded the set timeout before returning. Querying the job queue will provide the data of the job but is inconvenient. This can be resolved by either manually using the `-t` option to set a longer timeout when running commands (by default it is 5 seconds) or by modifying the minion configuration file: `/etc/salt/minion` and setting the `timeout` value to change the default timeout for all commands, and then restarting the salt-minion service.

Note: Modifying the minion timeout value is not required when running commands from a Salt Master. It is only required when running commands locally on the minion.

If a `state.apply` run takes too long, you can find a bottleneck by adding the *`--out=profile`* option.

4.10.3 Running in the Foreground

A great deal of information is available via the debug logging system, if you are having issues with minions connecting or not starting run the minion and/or master in the foreground:

```
salt-master -l debug
salt-minion -l debug
```

Anyone wanting to run Salt daemons via a process supervisor such as monit, runit, or supervisord, should omit the `-d` argument to the daemons and run them in the foreground.

4.10.4 What Ports do the Master and Minion Need Open?

No ports need to be opened up on each minion. For the master, TCP ports 4505 and 4506 need to be open. If you've put both your Salt master and minion in debug mode and don't see an acknowledgment that your minion has connected, it could very well be a firewall.

You can check port connectivity from the minion with the nc command:

```
nc -v -z salt.master.ip 4505
nc -v -z salt.master.ip 4506
```

There is also a *firewall configuration* document that might help as well.

If you've enabled the right TCP ports on your operating system or Linux distribution's firewall and still aren't seeing connections, check that no additional access control system such as SELinux or AppArmor is blocking Salt.

4.10.5 Using salt-call

The `salt-call` command was originally developed for aiding in the development of new Salt modules. Since then, many applications have been developed for running any Salt module locally on a minion. These range from the original intent of salt-call, development assistance, to gathering more verbose output from calls like *`state.apply`*.

When initially creating your state tree, it is generally recommended to invoke *`state.apply`* directly from the minion with `salt-call`, rather than remotely from the master. This displays far more information about the execution than calling it remotely. For even more verbosity, increase the loglevel using the `-l` argument:

```
salt-call -l debug state.apply
```

The main difference between using `salt` and using `salt-call` is that `salt-call` is run from the minion, and it only runs the selected function on that minion. By contrast, `salt` is run from the master, and requires you to specify the minions on which to run the command using salt's *targeting system*.

4.10.6 Too many open files

The salt-master needs at least 2 sockets per host that connects to it, one for the Publisher and one for response port. Thus, large installations may, upon scaling up the number of minions accessing a given master, encounter:

```
12:45:29,289 [salt.master    ][INFO    ] Starting Salt worker process 38
Too many open files
sock != -1 (tcp_listener.cpp:335)
```

The solution to this would be to check the number of files allowed to be opened by the user running salt-master (root by default):

```
[root@salt-master ~]# ulimit -n
1024
```

And modify that value to be at least equal to the number of minions x 2. This setting can be changed in limits.conf as the nofile value(s), and activated upon new a login of the specified user.

So, an environment with 1800 minions, would need 1800 x 2 = 3600 as a minimum.

4.10.7 Salt Master Stops Responding

There are known bugs with ZeroMQ versions less than 2.1.11 which can cause the Salt master to not respond properly. If you're running a ZeroMQ version greater than or equal to 2.1.9, you can work around the bug by setting the sysctls `net.core.rmem_max` and `net.core.wmem_max` to 16777216. Next, set the third field in `net.ipv4.tcp_rmem` and `net.ipv4.tcp_wmem` to at least 16777216.

You can do it manually with something like:

```
# echo 16777216 > /proc/sys/net/core/rmem_max
# echo 16777216 > /proc/sys/net/core/wmem_max
# echo "4096 87380 16777216" > /proc/sys/net/ipv4/tcp_rmem
# echo "4096 87380 16777216" > /proc/sys/net/ipv4/tcp_wmem
```

Or with the following Salt state:

```
net.core.rmem_max:
  sysctl:
    - present
    - value: 16777216

net.core.wmem_max:
  sysctl:
    - present
    - value: 16777216

net.ipv4.tcp_rmem:
  sysctl:
    - present
    - value: 4096 87380 16777216

net.ipv4.tcp_wmem:
  sysctl:
    - present
    - value: 4096 87380 16777216
```

4.10.8 Salt and SELinux

Currently there are no SELinux policies for Salt. For the most part Salt runs without issue when SELinux is running in Enforcing mode. This is because when the minion executes as a daemon the type context is changed to `initrc_t`. The problem with SELinux arises when using salt-call or running the minion in the foreground, since the type context stays `unconfined_t`.

This problem is generally manifest in the rpm install scripts when using the pkg module. Until a full SELinux Policy is available for Salt the solution to this issue is to set the execution context of `salt-call` and `salt-minion` to rpm_exec_t:

```
# CentOS 5 and RHEL 5:
chcon -t system_u:system_r:rpm_exec_t:s0 /usr/bin/salt-minion
chcon -t system_u:system_r:rpm_exec_t:s0 /usr/bin/salt-call

# CentOS 6 and RHEL 6:
chcon system_u:object_r:rpm_exec_t:s0 /usr/bin/salt-minion
chcon system_u:object_r:rpm_exec_t:s0 /usr/bin/salt-call
```

This works well, because the `rpm_exec_t` context has very broad control over other types.

4.10.9 Red Hat Enterprise Linux 5

Salt requires Python 2.6 or 2.7. Red Hat Enterprise Linux 5 and its variants come with Python 2.4 installed by default. When installing on RHEL 5 from the EPEL repository this is handled for you. But, if you run Salt from git, be advised that its dependencies need to be installed from EPEL and that Salt needs to be run with the `python26` executable.

4.10.10 Common YAML Gotchas

An extensive list of YAML idiosyncrasies has been compiled:

YAML Idiosyncrasies

One of Salt's strengths, the use of existing serialization systems for representing SLS data, can also backfire. YAML is a general purpose system and there are a number of things that would seem to make sense in an sls file that cause YAML issues. It is wise to be aware of these issues. While reports or running into them are generally rare they can still crop up at unexpected times.

Spaces vs Tabs

YAML uses spaces, period. Do not use tabs in your SLS files! If strange errors are coming up in rendering SLS files, make sure to check that no tabs have crept in! In Vim, after enabling search highlighting with: `:set hlsearch`, you can check with the following key sequence in normal mode(you can hit *ESC* twice to be sure): `/`, *Ctrl-v*, *Tab*, then hit *Enter*. Also, you can convert tabs to 2 spaces by these commands in Vim: `:set tabstop=2 expandtab` and then `:retab`.

Indentation

The suggested syntax for YAML files is to use 2 spaces for indentation, but YAML will follow whatever indentation system that the individual file uses. Indentation of two spaces works very well for SLS files given the fact that the data is uniform and not deeply nested.

Nested Dictionaries

When dictionaries are nested within other data structures (particularly lists), the indentation logic sometimes changes. Examples of where this might happen include `context` and `default` options from the *`file.managed`* state:

```
/etc/http/conf/http.conf:
  file:
    - managed
    - source: salt://apache/http.conf
    - user: root
    - group: root
    - mode: 644
    - template: jinja
    - context:
        custom_var: "override"
    - defaults:
        custom_var: "default value"
        other_var: 123
```

Notice that while the indentation is two spaces per level, for the values under the `context` and `defaults` options there is a four-space indent. If only two spaces are used to indent, then those keys will be considered part of the same dictionary that contains the `context` key, and so the data will not be loaded correctly. If using a double indent is not desirable, then a deeply-nested dict can be declared with curly braces:

```
/etc/http/conf/http.conf:
  file:
    - managed
    - source: salt://apache/http.conf
    - user: root
    - group: root
    - mode: 644
    - template: jinja
    - context: {
      custom_var: "override" }
    - defaults: {
      custom_var: "default value",
      other_var: 123 }
```

Here is a more concrete example of how YAML actually handles these indentations, using the Python interpreter on the command line:

```
>>> import yaml
>>> yaml.safe_load('''mystate:
...   file.managed:
...     - context:
...         some: var''')
{'mystate': {'file.managed': [{'context': {'some': 'var'}}]}}
>>> yaml.safe_load('''mystate:
...   file.managed:
...     - context:
...       some: var''')
{'mystate': {'file.managed': [{'some': 'var', 'context': None}]}}
```

Note that in the second example, `some` is added as another key in the same dictionary, whereas in the first example, it's the start of a new dictionary. That's the distinction. `context` is a common example because it is a keyword arg for many functions, and should contain a dictionary.

True/False, Yes/No, On/Off

PyYAML will load these values as boolean `True` or `False`. Un-capitalized versions will also be loaded as booleans (`true`, `false`, `yes`, `no`, `on`, and `off`). This can be especially problematic when constructing Pillar data. Make sure that your Pillars which need to use the string versions of these values are enclosed in quotes. Pillars will be parsed twice by salt, so you'll need to wrap your values in multiple quotes, including double quotation marks (`" "`) and single quotation marks (`' '`). Note that spaces are included in the quotation type examples for clarity.

Multiple quoting examples looks like this:

```
- '"false"'
- "'True'"
- "'YES'"
- '"No"'
```

Note: When using multiple quotes in this manner, they must be different. Using `"" ""` or `'' ''` won't work in this case (spaces are included in examples for clarity).

The '%' Sign

The `%` symbol has a special meaning in YAML, it needs to be passed as a string literal:

```
cheese:
  ssh_auth.present:
    - user: tbortels
    - source: salt://ssh_keys/chease.pub
    - config: '%h/.ssh/authorized_keys'
```

Time Expressions

PyYAML will load a time expression as the integer value of that, assuming `HH:MM`. So for example, `12:00` is loaded by PyYAML as `720`. An excellent explanation for why can be found here.

To keep time expressions like this from being loaded as integers, always quote them.

Note: When using a jinja `load_yaml` map, items must be quoted twice. For example:

```
{% load_yaml as wsus_schedule %}

FRI_10:
  time: '"23:00"'
  day: 6 - Every Friday
SAT_10:
  time: '"06:00"'
  day: 7 - Every Saturday
SAT_20:
  time: '"14:00"'
  day: 7 - Every Saturday
SAT_30:
  time: '"22:00"'
  day: 7 - Every Saturday
SUN_10:
  time: '"06:00"'
  day: 1 - Every Sunday
{% endload %}
```

YAML does not like ”Double Short Decs”

If I can find a way to make YAML accept ”Double Short Decs” then I will, since I think that double short decs would be awesome. So what is a ”Double Short Dec”? It is when you declare a multiple short decs in one ID. Here is a standard short dec, it works great:

```
vim:
  pkg.installed
```

The short dec means that there are no arguments to pass, so it is not required to add any arguments, and it can save space.

YAML though, gets upset when declaring multiple short decs, for the record…

THIS DOES NOT WORK:

```
vim:
  pkg.installed
  user.present
```

Similarly declaring a short dec in the same ID dec as a standard dec does not work either…

ALSO DOES NOT WORK:

```
fred:
  user.present
  ssh_auth.present:
    - name: AAAAB3NzaC...
    - user: fred
    - enc: ssh-dss
    - require:
      - user: fred
```

The correct way is to define them like this:

```
vim:
  pkg.installed: []
  user.present: []

fred:
  user.present: []
  ssh_auth.present:
    - name: AAAAB3NzaC...
    - user: fred
    - enc: ssh-dss
    - require:
      - user: fred
```

Alternatively, they can be defined the ”old way”, or with multiple ”full decs”:

```
vim:
  pkg:
    - installed
  user:
    - present

fred:
  user:
```

(continues on next page)

(continued from previous page)

```
    - present
  ssh_auth:
    - present
    - name: AAAAB3NzaC...
    - user: fred
    - enc: ssh-dss
    - require:
      - user: fred
```

YAML supports only plain ASCII

According to YAML specification, only ASCII characters can be used.

Within double-quotes, special characters may be represented with C-style escape sequences starting with a backslash (\ ).

Examples:

```
- micro: "\u00b5"
- copyright: "\u00A9"
- A: "\x41"
- alpha: "\u0251"
- Alef: "\u05d0"
```

List of usable Unicode characters will help you to identify correct numbers.

Python can also be used to discover the Unicode number for a character:

```
repr(u"Text with wrong characters i need to figure out")
```

This shell command can find wrong characters in your SLS files:

```
find . -name '*.sls'  -exec  grep --color='auto' -P -n '[^\x00-\x7F]' \{} \;
```

Alternatively you can toggle the *yaml_utf8* setting in your master configuration file. This is still an experimental setting but it should manage the right encoding conversion in salt after yaml states compilations.

Underscores stripped in Integer Definitions

If a definition only includes numbers and underscores, it is parsed by YAML as an integer and all underscores are stripped. To ensure the object becomes a string, it should be surrounded by quotes. More information here.

Here's an example:

```
>>> import yaml
>>> yaml.safe_load('2013_05_10')
20130510
>>> yaml.safe_load('"2013_05_10"')
'2013_05_10'
```

Automatic `datetime` conversion

If there is a value in a YAML file formatted `2014-01-20 14:23:23` or similar, YAML will automatically convert this to a Python `datetime` object. These objects are not msgpack serializable, and so may break core salt functionality. If values such as these are needed in a salt YAML file (specifically a configuration file), they should be formatted with surrounding strings to force YAML to serialize them as strings:

```
>>> import yaml
>>> yaml.safe_load('2014-01-20 14:23:23')
datetime.datetime(2014, 1, 20, 14, 23, 23)
>>> yaml.safe_load('"2014-01-20 14:23:23"')
'2014-01-20 14:23:23'
```

Additionally, numbers formatted like `XXXX-XX-XX` will also be converted (or YAML will attempt to convert them, and error out if it doesn't think the date is a real one). Thus, for example, if a minion were to have an ID of `4017-16-20` the minion would not start because YAML would complain that the date was out of range. The workaround is the same, surround the offending string with quotes:

```
>>> import yaml
>>> yaml.safe_load('4017-16-20')
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "/usr/local/lib/python2.7/site-packages/yaml/__init__.py", line 93, in safe_
↪load
    return load(stream, SafeLoader)
  File "/usr/local/lib/python2.7/site-packages/yaml/__init__.py", line 71, in load
    return loader.get_single_data()
  File "/usr/local/lib/python2.7/site-packages/yaml/constructor.py", line 39, in get_
↪single_data
    return self.construct_document(node)
  File "/usr/local/lib/python2.7/site-packages/yaml/constructor.py", line 43, in
↪construct_document
    data = self.construct_object(node)
  File "/usr/local/lib/python2.7/site-packages/yaml/constructor.py", line 88, in
↪construct_object
    data = constructor(self, node)
  File "/usr/local/lib/python2.7/site-packages/yaml/constructor.py", line 312, in
↪construct_yaml_timestamp
    return datetime.date(year, month, day)
ValueError: month must be in 1..12
>>> yaml.safe_load('"4017-16-20"')
'4017-16-20'
```

Keys Limited to 1024 Characters

Simple keys are limited to a single line and cannot be longer that 1024 characters. This is a limitation from PyYaml, as seen in a comment in PyYAML's code, and applies to anything parsed by YAML in Salt.

4.10.11 Live Python Debug Output

If the minion or master seems to be unresponsive, a SIGUSR1 can be passed to the processes to display where in the code they are running. If encountering a situation like this, this debug information can be invaluable. First make sure the master of minion are running in the foreground:

```
salt-master -l debug
salt-minion -l debug
```

Then pass the signal to the master or minion when it seems to be unresponsive:

```
killall -SIGUSR1 salt-master
killall -SIGUSR1 salt-minion
```

Also under BSD and macOS in addition to SIGUSR1 signal, debug subroutine set up for SIGINFO which has an advantage of being sent by Ctrl+T shortcut.

When filing an issue or sending questions to the mailing list for a problem with an unresponsive daemon this information can be invaluable.

4.10.12 Salt 0.16.x minions cannot communicate with a 0.17.x master

As of release 0.17.1 you can no longer run different versions of Salt on your Master and Minion servers. This is due to a protocol change for security purposes. The Salt team will continue to attempt to ensure versions are as backwards compatible as possible.

4.10.13 Debugging the Master and Minion

A list of common *master* and *minion* troubleshooting steps provide a starting point for resolving issues you may encounter.

4.11 Frequently Asked Questions

FAQ

- *Frequently Asked Questions*
  - *Is Salt open-core?*
  - *I think I found a bug! What should I do?*
  - *What ports should I open on my firewall?*
  - *I'm seeing weird behavior (including but not limited to packages not installing their users properly)*
  - *My script runs every time I run a* state.apply*. Why?*
  - *When I run* test.ping*, why don't the Minions that aren't responding return anything? Returning* `False` *would be helpful.*
  - *How does Salt determine the Minion's id?*
  - *I'm trying to manage packages/services but I get an error saying that the state is not available. Why?*
  - *Why aren't my custom modules/states/etc. available on my Minions?*
  - *Module X isn't available, even though the shell command it uses is installed. Why?*
  - *Can I run different versions of Salt on my Master and Minion?*
  - *Does Salt support backing up managed files?*
  - *Is it possible to deploy a file to a specific minion, without other minions having access to it?*
  - *What is the best way to restart a Salt Minion daemon using Salt after upgrade?*
    - *Upgrade without automatic restart*

    - *Restart using states*
    - *Restart using remote executions*
    - *Waiting for minions to come back online*
  - *Salting the Salt Master*
  - *Is Targeting using Grain Data Secure?*
  - *Why Did the Value for a Grain Change on Its Own?*

4.11.1 Is Salt open-core?

No. Salt is 100% committed to being open-source, including all of our APIs. It is developed under the Apache 2.0 license, allowing it to be used in both open and proprietary projects.

To expand on this a little:

There is much argument over the actual definition of "open core". From our standpoint, Salt is open source because

1. It is a standalone product that anyone is free to use.
2. It is developed in the open with contributions accepted from the community for the good of the project.
3. There are no features of Salt itself that are restricted to separate proprietary products distributed by SaltStack, Inc.
4. Because of our Apache 2.0 license, Salt can be used as the foundation for a project or even a proprietary tool.
5. Our APIs are open and documented (any lack of documentation is an oversight as opposed to an intentional decision by SaltStack the company) and available for use by anyone.

SaltStack the company does make proprietary products which use Salt and its libraries, like company is free to do, but we do so via the APIs, NOT by forking Salt and creating a different, closed-source version of it for paying customers.

4.11.2 I think I found a bug! What should I do?

The salt-users mailing list as well as the salt IRC channel can both be helpful resources to confirm if others are seeing the issue and to assist with immediate debugging.

To report a bug to the Salt project, please follow the instructions in *reporting a bug*.

4.11.3 What ports should I open on my firewall?

Minions need to be able to connect to the Master on TCP ports 4505 and 4506. Minions do not need any inbound ports open. More detailed information on firewall settings can be found *here*.

4.11.4 I'm seeing weird behavior (including but not limited to packages not installing their users properly)

This is often caused by SELinux. Try disabling SELinux or putting it in permissive mode and see if the weird behavior goes away.

4.11.5 My script runs every time I run a *state.apply*. Why?

You are probably using *`cmd.run`* rather than *`cmd.wait`*. A *`cmd.wait`* state will only run when there has been a change in a state that it is watching.

A *`cmd.run`* state will run the corresponding command *every time* (unless it is prevented from running by the `unless` or `onlyif` arguments).

More details can be found in the documentation for the *`cmd`* states.

4.11.6 When I run *test.ping*, why don't the Minions that aren't responding return anything? Returning `False` would be helpful.

When you run *test.ping* the Master tells Minions to run commands/functions, and listens for the return data, printing it to the screen when it is received. If it doesn't receive anything back, it doesn't have anything to display for that Minion.

There are a couple options for getting information on Minions that are not responding. One is to use the verbose (`-v`) option when you run salt commands, as it will display "Minion did not return" for any Minions which time out.

```
salt -v '*' pkg.install zsh
```

Another option is to use the *`manage.down`* runner:

```
salt-run manage.down
```

Also, if the Master is under heavy load, it is possible that the CLI will exit without displaying return data for all targeted Minions. However, this doesn't mean that the Minions did not return; this only means that the Salt CLI timed out waiting for a response. Minions will still send their return data back to the Master once the job completes. If any expected Minions are missing from the CLI output, the *`jobs.list_jobs`* runner can be used to show the job IDs of the jobs that have been run, and the *`jobs.lookup_jid`* runner can be used to get the return data for that job.

```
salt-run jobs.list_jobs
salt-run jobs.lookup_jid 20130916125524463507
```

If you find that you are often missing Minion return data on the CLI, only to find it with the jobs runners, then this may be a sign that the *`worker_threads`* value may need to be increased in the master config file. Additionally, running your Salt CLI commands with the `-t` option will make Salt wait longer for the return data before the CLI command exits. For instance, the below command will wait up to 60 seconds for the Minions to return:

```
salt -t 60 '*' test.ping
```

4.11.7 How does Salt determine the Minion's id?

If the Minion id is not configured explicitly (using the *`id`* parameter), Salt will determine the id based on the hostname. Exactly how this is determined varies a little between operating systems and is described in detail *here*.

4.11.8 I'm trying to manage packages/services but I get an error saying that the state is not available. Why?

Salt detects the Minion's operating system and assigns the correct package or service management module based on what is detected. However, for certain custom spins and OS derivatives this detection fails. In cases like this, an issue should be opened on our tracker, with the following information:

1. The output of the following command:

   ```
   salt <minion_id> grains.items | grep os
   ```

2. The contents of `/etc/lsb-release`, if present on the Minion.

4.11.9 Why aren't my custom modules/states/etc. available on my Minions?

Custom modules are synced to Minions when `saltutil.sync_modules`, or `saltutil.sync_all` is run.

Similarly, custom states are synced to Minions when `saltutil.sync_states`, or `saltutil.sync_all` is run.

They are both also synced when a *highstate* is triggered.

As of the 2019.2.0 release, as well as 2017.7.7 and 2018.3.2 in their respective release cycles, the `sync` argument to `state.apply`/`state.sls` can be used to sync custom types when running individual SLS files.

Other custom types (renderers, outputters, etc.) have similar behavior, see the documentation for the `saltutil` module for more information.

This reactor example can be used to automatically sync custom types when the minion connects to the master, to help with this chicken-and-egg issue.

4.11.10 Module X isn't available, even though the shell command it uses is installed. Why?

This is most likely a PATH issue. Did you custom-compile the software which the module requires? RHEL/CentOS/etc. in particular override the root user's path in `/etc/init.d/functions`, setting it to `/sbin:/usr/sbin:/bin:/usr/bin`, making software installed into `/usr/local/bin` unavailable to Salt when the Minion is started using the initscript. In version 2014.1.0, Salt will have a better solution for these sort of PATH-related issues, but recompiling the software to install it into a location within the PATH should resolve the issue in the meantime. Alternatively, you can create a symbolic link within the PATH using a `file.symlink` state.

```
/usr/bin/foo:
  file.symlink:
    - target: /usr/local/bin/foo
```

4.11.11 Can I run different versions of Salt on my Master and Minion?

This depends on the versions. In general, it is recommended that Master and Minion versions match.

When upgrading Salt, the master(s) should always be upgraded first. Backwards compatibility for minions running newer versions of salt than their masters is not guaranteed.

Whenever possible, backwards compatibility between new masters and old minions will be preserved. Generally, the only exception to this policy is in case of a security vulnerability.

Recent examples of backwards compatibility breakage include the 0.17.1 release (where all backwards compatibility was broken due to a security fix), and the 2014.1.0 release (which retained compatibility between 2014.1.0 masters and 0.17 minions, but broke compatibility for 2014.1.0 minions and older masters).

4.11.12 Does Salt support backing up managed files?

Yes. Salt provides an easy to use addition to your file.managed states that allow you to back up files via *backup_mode*, backup_mode can be configured on a per state basis, or in the minion config (note that if set in the minion config this would simply be the default method to use, you still need to specify that the file should be backed up!).

4.11.13 Is it possible to deploy a file to a specific minion, without other minions having access to it?

The Salt fileserver does not yet support access control, but it is still possible to do this. As of Salt 2015.5.0, the `file_tree` external pillar is available, and allows the contents of a file to be loaded as Pillar data. This external pillar is capable of assigning Pillar values both to individual minions, and to *nodegroups*. See the `documentation` for details on how to set this up.

Once the external pillar has been set up, the data can be pushed to a minion via a `file.managed` state, using the `contents_pillar` argument:

```
/etc/my_super_secret_file:
  file.managed:
    - user: secret
    - group: secret
    - mode: 600
    - contents_pillar: secret_files:my_super_secret_file
```

In this example, the source file would be located in a directory called `secret_files` underneath the file_tree path for the minion. The syntax for specifying the pillar variable is the same one used for `pillar.get`, with a colon representing a nested dictionary.

Warning: Deploying binary contents using the `file.managed` state is only supported in Salt 2015.8.4 and newer.

4.11.14 What is the best way to restart a Salt Minion daemon using Salt after upgrade?

Updating the `salt-minion` package requires a restart of the `salt-minion` service. But restarting the service while in the middle of a state run interrupts the process of the Minion running states and sending results back to the Master. A common way to workaround that is to schedule restarting the Minion service in the background by issuing a `salt-call` command calling `service.restart` function. This prevents the Minion being disconnected from the Master immediately. Otherwise you would get `Minion did not return. [Not connected]` message as the result of a state run.

Upgrade without automatic restart

Doing the Minion upgrade seems to be a simplest state in your SLS file at first. But the operating systems such as Debian GNU/Linux, Ubuntu and their derivatives start the service after the package installation by default. To prevent this, we need to create policy layer which will prevent the Minion service to restart right after the upgrade:

```
{%- if grains['os_family'] == 'Debian' %}

Disable starting services:
  file.managed:
    - name: /usr/sbin/policy-rc.d
```

(continues on next page)

(continued from previous page)

```
    - user: root
    - group: root
    - mode: 0755
    - contents:
      - '#!/bin/sh'
      - exit 101
    # do not touch if already exists
    - replace: False
    - prereq:
      - pkg: Upgrade Salt Minion

{%- endif %}

Upgrade Salt Minion:
  pkg.installed:
    - name: salt-minion
    - version: 2016.11.3{% if grains['os_family'] == 'Debian' %}+ds-1{% endif %}
    - order: last

Enable Salt Minion:
  service.enabled:
    - name: salt-minion
    - require:
      - pkg: Upgrade Salt Minion

{%- if grains['os_family'] == 'Debian' %}

Enable starting services:
  file.absent:
    - name: /usr/sbin/policy-rc.d
    - onchanges:
      - pkg: Upgrade Salt Minion

{%- endif %}
```

Restart using states

Now we can apply the workaround to restart the Minion in reliable way. The following example works on UNIX-like operating systems:

```
{%- if grains['os'] != 'Windows' %}
Restart Salt Minion:
  cmd.run:
    - name: 'salt-call service.restart salt-minion'
    - bg: True
    - onchanges:
      - pkg: Upgrade Salt Minion
{%- endif %}
```

Note that restarting the `salt-minion` service on Windows operating systems is not always necessary when performing an upgrade. The installer stops the `salt-minion` service, removes it, deletes the contents of the `\salt\bin` directory, installs the new code, re-creates the `salt-minion` service, and starts it (by default). The restart step **would** be necessary during the upgrade process, however, if the minion config was edited after the upgrade or installation. If a minion restart is necessary, the state above can be edited as follows:

```
Restart Salt Minion:
  cmd.run:
{%- if grains['kernel'] == 'Windows' %}
    - name: 'C:\salt\salt-call.bat service.restart salt-minion'
{%- else %}
    - name: 'salt-call service.restart salt-minion'
{%- endif %}
    - bg: True
    - onchanges:
      - pkg: Upgrade Salt Minion
```

However, it requires more advanced tricks to upgrade from legacy version of Salt (before `2016.3.0`) on UNIX-like operating systems, where executing commands in the background is not supported. You also may need to schedule restarting the Minion service using *masterless mode* after all other states have been applied for Salt versions earlier than `2016.11.0`. This allows the Minion to keep the connection to the Master alive for being able to report the final results back to the Master, while the service is restarting in the background. This state should run last or watch for the `pkg` state changes:

```
Restart Salt Minion:
  cmd.run:
{%- if grains['kernel'] == 'Windows' %}
    - name: 'start powershell "Restart-Service -Name salt-minion"'
{%- else %}
    # fork and disown the process
    - name: |-
        exec 0>&- # close stdin
        exec 1>&- # close stdout
        exec 2>&- # close stderr
        nohup salt-call --local service.restart salt-minion &
{%- endif %}
```

Restart using remote executions

Restart the Minion from the command line:

```
salt -G kernel:Windows cmd.run_bg 'C:\salt\salt-call.bat service.restart salt-minion'
salt -C 'not G@kernel:Windows' cmd.run_bg 'salt-call service.restart salt-minion'
```

Waiting for minions to come back online

A common issue in performing automated restarts of a salt minion, for example during an orchestration run, is that it will break the orchestration since the next statement is likely to be attempted before the minion is back online. This can be remedied by inserting a blocking waiting state that only returns when the selected minions are back up (note: this will only work in orchestration states since *manage.up* needs to run on the master):

```
Wait for salt minion:
  loop.until_no_eval:
    - name: saltutil.runner
    - expected:
        - my_minion
    - args:
        - manage.up
    - kwargs:
```

(continues on next page)

(continued from previous page)

```
        tgt: my_minion
    - period: 3
    - init_wait: 3
```

This will, after an initial delay of 3 seconds, execute the *manage.up*-runner targeted specifically for *my_minion*. It will do this every *period* seconds until the *expected* data is returned. The default timeout is 60s but can be configured as well.

4.11.15 Salting the Salt Master

In order to configure a master server via states, the Salt master can also be "salted" in order to enforce state on the Salt master as well as the Salt minions. Salting the Salt master requires a Salt minion to be installed on the same machine as the Salt master. Once the Salt minion is installed, the minion configuration file must be pointed to the local Salt master:

```
master: 127.0.0.1
```

Once the Salt master has been "salted" with a Salt minion, it can be targeted just like any other minion. If the minion on the salted master is running, the minion can be targeted via any usual `salt` command. Additionally, the `salt-call` command can execute operations to enforce state on the salted master without requiring the minion to be running.

More information about salting the Salt master can be found in the salt-formula for salt itself:

https://github.com/saltstack-formulas/salt-formula

Restarting the `salt-master` service using execution module or application of state could be done the same way as for the Salt minion described *above*.

4.11.16 Is Targeting using Grain Data Secure?

Because grains can be set by users that have access to the minion configuration files on the local system, grains are considered less secure than other identifiers in Salt. Use caution when targeting sensitive operations or setting pillar values based on grain data.

The only grain which can be safely used is `grains['id']` which contains the Minion ID.

When possible, you should target sensitive operations and data using the Minion ID. If the Minion ID of a system changes, the Salt Minion's public key must be re-accepted by an administrator on the Salt Master, making it less vulnerable to impersonation attacks.

4.11.17 Why Did the Value for a Grain Change on Its Own?

This is usually the result of an upstream change in an OS distribution that replaces or removes something that Salt was using to detect the grain. Fortunately, when this occurs, you can use Salt to fix it with a command similar to the following:

```
salt -G 'grain:ChangedValue' grains.setvals "{'grain': 'OldValue'}"
```

(Replacing *grain*, *ChangedValue*, and *OldValue* with the grain and values that you want to change / set.)

You should also file an issue describing the change so it can be fixed in Salt.

4.12 Salt Best Practices

Salt's extreme flexibility leads to many questions concerning the structure of configuration files.

This document exists to clarify these points through examples and code.

4.12.1 General rules

1. Modularity and clarity should be emphasized whenever possible.
2. Create clear relations between pillars and states.
3. Use variables when it makes sense but don't overuse them.
4. Store sensitive data in pillar.
5. Don't use grains for matching in your pillar top file for any sensitive pillars.

4.12.2 Structuring States and Formulas

When structuring Salt States and Formulas it is important to begin with the directory structure. A proper directory structure clearly defines the functionality of each state to the user via visual inspection of the state's name.

Reviewing the MySQL Salt Formula it is clear to see the benefits to the end-user when reviewing a sample of the available states:

```
/srv/salt/mysql/files/
/srv/salt/mysql/client.sls
/srv/salt/mysql/map.jinja
/srv/salt/mysql/python.sls
/srv/salt/mysql/server.sls
```

This directory structure would lead to these states being referenced in a top file in the following way:

```
base:
  'web*':
    - mysql.client
    - mysql.python
  'db*':
    - mysql.server
```

This clear definition ensures that the user is properly informed of what each state will do.

Another example comes from the vim-formula:

```
/srv/salt/vim/files/
/srv/salt/vim/absent.sls
/srv/salt/vim/init.sls
/srv/salt/vim/map.jinja
/srv/salt/vim/nerdtree.sls
/srv/salt/vim/pyflakes.sls
/srv/salt/vim/salt.sls
```

Once again viewing how this would look in a top file:

/srv/salt/top.sls:

```
base:
  'web*':
    - vim
    - vim.nerdtree
    - vim.pyflakes
    - vim.salt
  'db*':
    - vim.absent
```

The usage of a clear top-level directory as well as properly named states reduces the overall complexity and leads a user to both understand what will be included at a glance and where it is located.

In addition *Formulas* should be used as often as possible.

Note: Formulas repositories on the saltstack-formulas GitHub organization should not be pointed to directly from systems that automatically fetch new updates such as GitFS or similar tooling. Instead formulas repositories should be forked on GitHub or cloned locally, where unintended, automatic changes will not take place.

4.12.3 Structuring Pillar Files

Pillars are used to store secure and insecure data pertaining to minions. When designing the structure of the `/srv/pillar` directory, the pillars contained within should once again be focused on clear and concise data which users can easily review, modify, and understand.

The `/srv/pillar/` directory is primarily controlled by `top.sls`. It should be noted that the pillar `top.sls` is not used as a location to declare variables and their values. The `top.sls` is used as a way to include other pillar files and organize the way they are matched based on environments or grains.

An example `top.sls` may be as simple as the following:

/srv/pillar/top.sls:

```
base:
  '*':
    - packages
```

Any number of matchers can be added to the base environment. For example, here is an expanded version of the Pillar top file stated above:

/srv/pillar/top.sls:

```
base:
  '*':
    - packages
  'web*':
    - apache
    - vim
```

Or an even more complicated example, using a variety of matchers in numerous environments:

/srv/pillar/top.sls:

```
base:
  '*':
    - apache
```

(continues on next page)

(continued from previous page)

```
dev:
  'os:Debian':
    - match: grain
    - vim
test:
  '* and not G@os: Debian':
    - match: compound
    - emacs
```

It is clear to see through these examples how the top file provides users with power but when used incorrectly it can lead to confusing configurations. This is why it is important to understand that the top file for pillar is not used for variable definitions.

Each SLS file within the `/srv/pillar/` directory should correspond to the states which it matches.

This would mean that the `apache` pillar file should contain data relevant to Apache. Structuring files in this way once again ensures modularity, and creates a consistent understanding throughout our Salt environment. Users can expect that pillar variables found in an Apache state will live inside of an Apache pillar:

`/srv/pillar/apache.sls`:

```
apache:
  lookup:
    name: httpd
    config:
      tmpl: /etc/httpd/httpd.conf
```

While this pillar file is simple, it shows how a pillar file explicitly relates to the state it is associated with.

4.12.4 Variable Flexibility

Salt allows users to define variables in SLS files. When creating a state variables should provide users with as much flexibility as possible. This means that variables should be clearly defined and easy to manipulate, and that sane defaults should exist in the event a variable is not properly defined. Looking at several examples shows how these different items can lead to extensive flexibility.

Although it is possible to set variables locally, this is generally not preferred:

`/srv/salt/apache/conf.sls`:

```
{% set name = 'httpd' %}
{% set tmpl = 'salt://apache/files/httpd.conf' %}

include:
  - apache

apache_conf:
  file.managed:
    - name: {{ name }}
    - source: {{ tmpl }}
    - template: jinja
    - user: root
    - watch_in:
      - service: apache
```

When generating this information it can be easily transitioned to the pillar where data can be overwritten, modified, and applied to multiple states, or locations within a single state:

`/srv/pillar/apache.sls`:

```yaml
apache:
  lookup:
    name: httpd
    config:
      tmpl: salt://apache/files/httpd.conf
```

`/srv/salt/apache/conf.sls`:

```
{% from "apache/map.jinja" import apache with context %}

include:
  - apache

apache_conf:
  file.managed:
    - name: {{ salt['pillar.get']('apache:lookup:name') }}
    - source: {{ salt['pillar.get']('apache:lookup:config:tmpl') }}
    - template: jinja
    - user: root
    - watch_in:
      - service: apache
```

This flexibility provides users with a centralized location to modify variables, which is extremely important as an environment grows.

4.12.5 Modularity Within States

Ensuring that states are modular is one of the key concepts to understand within Salt. When creating a state a user must consider how many times the state could be re-used, and what it relies on to operate. Below are several examples which will iteratively explain how a user can go from a state which is not very modular to one that is:

`/srv/salt/apache/init.sls`:

```yaml
httpd:
  pkg:
    - installed
  service.running:
    - enable: True

/etc/httpd/httpd.conf:
  file.managed:
    - source: salt://apache/files/httpd.conf
    - template: jinja
    - watch_in:
      - service: httpd
```

The example above is probably the worst-case scenario when writing a state. There is a clear lack of focus by naming both the pkg/service, and managed file directly as the state ID. This would lead to changing multiple requires within this state, as well as others that may depend upon the state.

Imagine if a require was used for the `httpd` package in another state, and then suddenly it's a custom package. Now changes need to be made in multiple locations which increases the complexity and leads to a more error prone configuration.

There is also the issue of having the configuration file located in the init, as a user would be unable to simply install the service and use the default conf file.

Our second revision begins to address the referencing by using `- name`, as opposed to direct ID references:

`/srv/salt/apache/init.sls`:

```
apache:
  pkg.installed:
    - name: httpd
  service.running:
    - name: httpd
    - enable: True

apache_conf:
  file.managed:
    - name: /etc/httpd/httpd.conf
    - source: salt://apache/files/httpd.conf
    - template: jinja
    - watch_in:
      - service: apache
```

The above init file is better than our original, yet it has several issues which lead to a lack of modularity. The first of these problems is the usage of static values for items such as the name of the service, the name of the managed file, and the source of the managed file. When these items are hard coded they become difficult to modify and the opportunity to make mistakes arises. It also leads to multiple edits that need to occur when changing these items (imagine if there were dozens of these occurrences throughout the state!). There is also still the concern of the configuration file data living in the same state as the service and package.

In the next example steps will be taken to begin addressing these issues. Starting with the addition of a map.jinja file (as noted in the *Formula documentation*), and modification of static values:

`/srv/salt/apache/map.jinja`:

```
{% set apache = salt['grains.filter_by']({
    'Debian': {
        'server': 'apache2',
        'service': 'apache2',
        'conf': '/etc/apache2/apache.conf',
    },
    'RedHat': {
        'server': 'httpd',
        'service': 'httpd',
        'conf': '/etc/httpd/httpd.conf',
    },
}, merge=salt['pillar.get']('apache:lookup')) %}
```

/srv/pillar/apache.sls:

```
apache:
  lookup:
    config:
      tmpl: salt://apache/files/httpd.conf
```

`/srv/salt/apache/init.sls`:

```
{% from "apache/map.jinja" import apache with context %}

apache:
  pkg.installed:
    - name: {{ apache.server }}
```

(continues on next page)

(continued from previous page)

```
  service.running:
    - name: {{ apache.service }}
    - enable: True

apache_conf:
  file.managed:
    - name: {{ apache.conf }}
    - source: {{ salt['pillar.get']('apache:lookup:config:tmpl') }}
    - template: jinja
    - user: root
    - watch_in:
      - service: apache
```

The changes to this state now allow us to easily identify the location of the variables, as well as ensuring they are flexible and easy to modify. While this takes another step in the right direction, it is not yet complete. Suppose the user did not want to use the provided conf file, or even their own configuration file, but the default apache conf. With the current state setup this is not possible. To attain this level of modularity this state will need to be broken into two states.

`/srv/salt/apache/map.jinja`:

```
{% set apache = salt['grains.filter_by']({
    'Debian': {
        'server': 'apache2',
        'service': 'apache2',
        'conf': '/etc/apache2/apache.conf',
    },
    'RedHat': {
        'server': 'httpd',
        'service': 'httpd',
        'conf': '/etc/httpd/httpd.conf',
    },
}, merge=salt['pillar.get']('apache:lookup')) %}
```

`/srv/pillar/apache.sls`:

```
apache:
  lookup:
    config:
      tmpl: salt://apache/files/httpd.conf
```

`/srv/salt/apache/init.sls`:

```
{% from "apache/map.jinja" import apache with context %}

apache:
  pkg.installed:
    - name: {{ apache.server }}
  service.running:
    - name: {{ apache.service }}
    - enable: True
```

`/srv/salt/apache/conf.sls`:

```
{% from "apache/map.jinja" import apache with context %}

```

(continues on next page)

(continued from previous page)

```
include:
  - apache

apache_conf:
  file.managed:
    - name: {{ apache.conf }}
    - source: {{ salt['pillar.get']('apache:lookup:config:tmpl') }}
    - template: jinja
    - user: root
    - watch_in:
      - service: apache
```

This new structure now allows users to choose whether they only wish to install the default Apache, or if they wish, overwrite the default package, service, configuration file location, or the configuration file itself. In addition to this the data has been broken between multiple files allowing for users to identify where they need to change the associated data.

4.12.6 Storing Secure Data

Secure data refers to any information that you would not wish to share with anyone accessing a server. This could include data such as passwords, keys, or other information.

As all data within a state is accessible by EVERY server that is connected it is important to store secure data within pillar. This will ensure that only those servers which require this secure data have access to it. In this example a use can go from an insecure configuration to one which is only accessible by the appropriate hosts:

`/srv/salt/mysql/testerdb.sls`:

```
testdb:
  mysql_database.present:
    - name: testerdb
```

`/srv/salt/mysql/user.sls`:

```
include:
  - mysql.testerdb

testdb_user:
  mysql_user.present:
    - name: frank
    - password: "test3rdb"
    - host: localhost
    - require:
      - sls: mysql.testerdb
```

Many users would review this state and see that the password is there in plain text, which is quite problematic. It results in several issues which may not be immediately visible.

The first of these issues is clear to most users – the password being visible in this state. This means that any minion will have a copy of this, and therefore the password which is a major security concern as minions may not be locked down as tightly as the master server.

The other issue that can be encountered is access by users on the master. If everyone has access to the states (or their repository), then they are able to review this password. Keeping your password data accessible by only a few users is critical for both security and peace of mind.

There is also the issue of portability. When a state is configured this way it results in multiple changes needing to be made. This was discussed in the sections above but it is a critical idea to drive home. If states are not portable it may result in more work later!

Fixing this issue is relatively simple, the content just needs to be moved to the associated pillar:

`/srv/pillar/mysql.sls`:

```
mysql:
  lookup:
    name: testerdb
    password: test3rdb
    user: frank
    host: localhost
```

`/srv/salt/mysql/testerdb.sls`:

```
testdb:
  mysql_database.present:
    - name: {{ salt['pillar.get']('mysql:lookup:name') }}
```

`/srv/salt/mysql/user.sls`:

```
include:
  - mysql.testerdb

testdb_user:
  mysql_user.present:
    - name: {{ salt['pillar.get']('mysql:lookup:user') }}
    - password: {{ salt['pillar.get']('mysql:lookup:password') }}
    - host: {{ salt['pillar.get']('mysql:lookup:host') }}
    - require:
      - sls: mysql.testerdb
```

Now that the database details have been moved to the associated pillar file, only machines which are targeted via pillar will have access to these details. Access to users who should not be able to review these details can also be prevented while ensuring that they are still able to write states which take advantage of this information.

CHAPTER

FIVE

REMOTE EXECUTION

Running pre-defined or arbitrary commands on remote hosts, also known as remote execution, is the core function of Salt. The following links explore modules and returners, which are two key elements of remote execution.

Salt Execution Modules

Salt execution modules are called by the remote execution system to perform a wide variety of tasks. These modules provide functionality such as installing packages, restarting a service, running a remote command, transferring files, and so on.

Full list of execution modules Contains: a list of core modules that ship with Salt.

Writing execution modules Contains: a guide on how to write Salt modules.

5.1 Running Commands on Salt Minions

Salt can be controlled by a command line client by the root user on the Salt master. The Salt command line client uses the Salt client API to communicate with the Salt master server. The Salt client is straightforward and simple to use.

Using the Salt client commands can be easily sent to the minions.

Each of these commands accepts an explicit *–config* option to point to either the master or minion configuration file. If this option is not provided and the default configuration file does not exist then Salt falls back to use the environment variables `SALT_MASTER_CONFIG` and `SALT_MINION_CONFIG`.

See also:

Configuration

5.1.1 Using the Salt Command

The Salt command needs a few components to send information to the Salt minions. The target minions need to be defined, the function to call and any arguments the function requires.

Defining the Target Minions

The first argument passed to salt, defines the target minions, the target minions are accessed via their hostname. The default target type is a bash glob:

```
salt '*foo.com' sys.doc
```

Salt can also define the target minions with regular expressions:

```
salt -E '.*' cmd.run 'ls -l | grep foo'
```

Or to explicitly list hosts, salt can take a list:

```
salt -L foo.bar.baz,quo.qux cmd.run 'ps aux | grep foo'
```

More Powerful Targets

See *Targeting*.

Calling the Function

The function to call on the specified target is placed after the target specification.

New in version 0.9.8.

Functions may also accept arguments, space-delimited:

```
salt '*' cmd.exec_code python 'import sys; print sys.version'
```

Optional, keyword arguments are also supported:

```
salt '*' pip.install salt timeout=5 upgrade=True
```

They are always in the form of `kwarg=argument`.

Arguments are formatted as YAML:

```
salt '*' cmd.run 'echo "Hello: $FIRST_NAME"' env='{FIRST_NAME: "Joe"}'
```

Note: dictionaries must have curly braces around them (like the `env` keyword argument above). This was changed in 0.15.1: in the above example, the first argument used to be parsed as the dictionary `{'echo "Hello': '$FIRST_NAME"'}`. This was generally not the expected behavior.

If you want to test what parameters are actually passed to a module, use the `test.arg_repr` command:

```
salt '*' test.arg_repr 'echo "Hello: $FIRST_NAME"' env='{FIRST_NAME: "Joe"}'
```

Finding available minion functions

The Salt functions are self documenting, all of the function documentation can be retried from the minions via the `sys.doc()` function:

```
salt '*' sys.doc
```

Compound Command Execution

If a series of commands needs to be sent to a single target specification then the commands can be sent in a single publish. This can make gathering groups of information faster, and lowers the stress on the network for repeated commands.

Compound command execution works by sending a list of functions and arguments instead of sending a single function and argument. The functions are executed on the minion in the order they are defined on the command line, and then the data from all of the commands are returned in a dictionary. This means that the set of commands are called in a predictable way, and the returned data can be easily interpreted.

Executing compound commands if done by passing a comma delimited list of functions, followed by a comma delimited list of arguments:

```
salt '*' cmd.run,test.ping,test.echo 'cat /proc/cpuinfo',,foo
```

The trick to look out for here, is that if a function is being passed no arguments, then there needs to be a placeholder for the absent arguments. This is why in the above example, there are two commas right next to each other. `test.ping` takes no arguments, so we need to add another comma, otherwise Salt would attempt to pass "foo" to `test.ping`.

If you need to pass arguments that include commas, then make sure you add spaces around the commas that separate arguments. For example:

```
salt '*' cmd.run,test.ping,test.echo 'echo "1,2,3"' , , foo
```

You may change the arguments separator using the `--args-separator` option:

```
salt --args-separator=:: '*' some.fun,test.echo params with , comma :: foo
```

5.1.2 CLI Completion

Shell completion scripts for the Salt CLI are available in the `pkg` Salt source directory.

5.2 Writing Execution Modules

Salt execution modules are the functions called by the **`salt`** command.

5.2.1 Modules Are Easy to Write!

Writing Salt execution modules is straightforward.

A Salt execution module is a Python or Cython module placed in a directory called `_modules/` at the root of the Salt fileserver. When using the default fileserver backend (i.e. *`roots`*), unless environments are otherwise defined in the *`file_roots`* config option, the `_modules/` directory would be located in `/srv/salt/_modules` on most systems.

Modules placed in `_modules/` will be synced to the minions when any of the following Salt functions are called:

- *`state.apply`*
- *`saltutil.sync_modules`*
- *`saltutil.sync_all`*

Note that a module's default name is its filename (i.e. `foo.py` becomes module `foo`), but that its name can be overridden by using a *__virtual__ function*.

If a Salt module has errors and cannot be imported, the Salt minion will continue to load without issue and the module with errors will simply be omitted.

If adding a Cython module the file must be named `<modulename>.pyx` so that the loader knows that the module needs to be imported as a Cython module. The compilation of the Cython module is automatic and happens when the minion starts, so only the `*.pyx` file is required.

5.2.2 Zip Archives as Modules

Python 2.3 and higher allows developers to directly import zip archives containing Python code. By setting *enable_zip_modules* to `True` in the minion config, the Salt loader will be able to import `.zip` files in this fashion. This allows Salt module developers to package dependencies with their modules for ease of deployment, isolation, etc.

For a user, Zip Archive modules behave just like other modules. When executing a function from a module provided as the file `my_module.zip`, a user would call a function within that module as `my_module.<function>`.

Creating a Zip Archive Module

A Zip Archive module is structured similarly to a simple Python package. The `.zip` file contains a single directory with the same name as the module. The module code traditionally in `<module_name>.py` goes in `<module_name>/__init__.py`. The dependency packages are subdirectories of `<module_name>/`.

Here is an example directory structure for the `lumberjack` module, which has two library dependencies (`sleep` and `work`) to be included.

```
modules $ ls -R lumberjack
__init__.py     sleep           work

lumberjack/sleep:
__init__.py

lumberjack/work:
__init__.py
```

The contents of `lumberjack/__init__.py` show how to import and use these included libraries.

```python
# Libraries included in lumberjack.zip
from lumberjack import sleep, work


def is_ok(person):
    ''' Checks whether a person is really a lumberjack '''
    return sleep.all_night(person) and work.all_day(person)
```

Then, create the zip:

```
modules $ zip -r lumberjack lumberjack
  adding: lumberjack/ (stored 0%)
  adding: lumberjack/__init__.py (deflated 39%)
  adding: lumberjack/sleep/ (stored 0%)
  adding: lumberjack/sleep/__init__.py (deflated 7%)
  adding: lumberjack/work/ (stored 0%)
  adding: lumberjack/work/__init__.py (deflated 7%)
modules $ unzip -l lumberjack.zip
Archive:  lumberjack.zip
  Length     Date   Time    Name
 --------    ----   ----    ----
        0  08-21-15 20:08   lumberjack/
      348  08-21-15 20:08   lumberjack/__init__.py
        0  08-21-15 19:53   lumberjack/sleep/
       83  08-21-15 19:53   lumberjack/sleep/__init__.py
        0  08-21-15 19:53   lumberjack/work/
       81  08-21-15 19:21   lumberjack/work/__init__.py
```

(continues on next page)

(continued from previous page)

```
 --------                   -------
      512                   6 files
```

Once placed in *file_roots*, Salt users can distribute and use `lumberjack.zip` like any other module.

```
$ sudo salt minion1 saltutil.sync_modules
minion1:
  - modules.lumberjack
$ sudo salt minion1 lumberjack.is_ok 'Michael Palin'
minion1:
  True
```

5.2.3 Cross Calling Execution Modules

All of the Salt execution modules are available to each other and modules can call functions available in other execution modules.

The variable `__salt__` is packed into the modules after they are loaded into the Salt minion.

The `__salt__` variable is a Python dictionary containing all of the Salt functions. Dictionary keys are strings representing the names of the modules and the values are the functions themselves.

Salt modules can be cross-called by accessing the value in the `__salt__` dict:

```
def foo(bar):
    return __salt__['cmd.run'](bar)
```

This code will call the *run* function in the *cmd* module and pass the argument `bar` to it.

5.2.4 Calling Execution Modules on the Salt Master

New in version 2016.11.0.

Execution modules can now also be called via the **`salt-run`** command using the *salt runner*.

5.2.5 Preloaded Execution Module Data

When interacting with execution modules often it is nice to be able to read information dynamically about the minion or to load in configuration parameters for a module.

Salt allows for different types of data to be loaded into the modules by the minion.

Grains Data

The values detected by the Salt Grains on the minion are available in a Python dictionary named `__grains__` and can be accessed from within callable objects in the Python modules.

To see the contents of the grains dictionary for a given system in your deployment run the `grains.items()` function:

```
salt 'hostname' grains.items --output=pprint
```

Any value in a grains dictionary can be accessed as any other Python dictionary. For example, the grain representing the minion ID is stored in the `id` key and from an execution module, the value would be stored in `__grains__['id']`.

Module Configuration

Since parameters for configuring a module may be desired, Salt allows for configuration information from the minion configuration file to be passed to execution modules.

Since the minion configuration file is a YAML document, arbitrary configuration data can be passed in the minion config that is read by the modules. It is therefore **strongly** recommended that the values passed in the configuration file match the module name. A value intended for the `test` execution module should be named `test.<value>`.

The test execution module contains usage of the module configuration and the default configuration file for the minion contains the information and format used to pass data to the modules. *`salt.modules.test`*, `conf/minion`.

`__init__` Function

If you want your module to have different execution modes based on minion configuration, you can use the `__init__(opts)` function to perform initial module setup. The parameter `opts` is the complete minion configuration, as also available in the `__opts__` dict.

```
'''
Cheese module initialization example
'''
def __init__(opts):
    '''
    Allow foreign imports if configured to do so
    '''
    if opts.get('cheese.allow_foreign', False):
        _enable_foreign_products()
```

5.2.6 Strings and Unicode

An execution module author should always assume that strings fed to the module have already decoded from strings into Unicode. In Python 2, these will be of type 'Unicode' and in Python 3 they will be of type `str`. Calling from a state to other Salt sub-systems, should pass Unicode (or bytes if passing binary data). In the rare event that a state needs to write directly to disk, Unicode should be encoded to a string immediately before writing to disk. An author may use `__salt_system_encoding__` to learn what the encoding type of the system is. For example, *'my_string'.encode(__salt_system_encoding__')*.

5.2.7 Outputter Configuration

Since execution module functions can return different data, and the way the data is printed can greatly change the presentation, Salt allows for a specific outputter to be set on a function-by-function basis.

This is done be declaring an `__outputter__` dictionary in the global scope of the module. The `__outputter__` dictionary contains a mapping of function names to Salt *outputters*.

```
__outputter__ = {
    'run': 'txt'
}
```

This will ensure that the `txt` outputter is used to display output from the `run` function.

5.2.8 Virtual Modules

Virtual modules let you override the name of a module in order to use the same name to refer to one of several similar modules. The specific module that is loaded for a virtual name is selected based on the current platform or environment.

For example, packages are managed across platforms using the `pkg` module. `pkg` is a virtual module name that is an alias for the specific package manager module that is loaded on a specific system (for example, *`yumpkg`* on RHEL/CentOS systems , and *`aptpkg`* on Ubuntu).

Virtual module names are set using the `__virtual__` function and the *virtual name*.

5.2.9 `__virtual__` Function

The `__virtual__` function returns either a string, `True`, `False`, or `False` with an *error string*. If a string is returned then the module is loaded using the name of the string as the virtual name. If `True` is returned the module is loaded using the current module name. If `False` is returned the module is not loaded. `False` lets the module perform system checks and prevent loading if dependencies are not met.

Since `__virtual__` is called before the module is loaded, `__salt__` will be unreliable as not all modules will be available at this point in time. The `__pillar__` and `__grains__` *"dunder" dictionaries* are available however.

Note: Modules which return a string from `__virtual__` that is already used by a module that ships with Salt will _override_ the stock module.

Returning Error Information from `__virtual__`

Optionally, Salt plugin modules, such as execution, state, returner, beacon, etc. modules may additionally return a string containing the reason that a module could not be loaded. For example, an execution module called `cheese` and a corresponding state module also called `cheese`, both depending on a utility called `enzymes` should have `__virtual__` functions that handle the case when the dependency is unavailable.

```
'''
Cheese execution (or returner/beacon/etc.) module
'''
try:
    import enzymes
    HAS_ENZYMES = True
except ImportError:
    HAS_ENZYMES = False


def __virtual__():
    '''
    only load cheese if enzymes are available
    '''
    if HAS_ENZYMES:
        return 'cheese'
    else:
        return False, 'The cheese execution module cannot be loaded: enzymes
↪unavailable.'
```

(continues on next page)

(continued from previous page)

```python
def slice():
    pass
```

```python
'''
Cheese state module. Note that this works in state modules because it is
guaranteed that execution modules are loaded first
'''

def __virtual__():
    '''
    only load cheese if enzymes are available
    '''
    # predicate loading of the cheese state on the corresponding execution module
    if 'cheese.slice' in __salt__:
        return 'cheese'
    else:
        return False, 'The cheese state module cannot be loaded: enzymes unavailable.'
```

Examples

The package manager modules are among the best examples of using the `__virtual__` function. A table of all the virtual `pkg` modules can be found *here*.

Overriding Virtual Module Providers

Salt often uses OS grains (`os`, `osrelease`, `os_family`, etc.) to determine which module should be loaded as the virtual module for `pkg`, `service`, etc. Sometimes this OS detection is incomplete, with new distros popping up, existing distros changing init systems, etc. The virtual modules likely to be affected by this are in the list below (click each item for more information):

- *pkg*
- *service*
- *user*
- *shadow*
- *group*

If Salt is using the wrong module for one of these, first of all, please report it on the issue tracker, so that this issue can be resolved for a future release. To make it easier to troubleshoot, please also provide the `grains.items` output, taking care to redact any sensitive information.

Then, while waiting for the SaltStack development team to fix the issue, Salt can be made to use the correct module using the `providers` option in the minion config file:

```yaml
providers:
  service: systemd
  pkg: aptpkg
```

The above example will force the minion to use the `systemd` module to provide service management, and the `aptpkg` module to provide package management.

Logging Restrictions

As a rule, logging should not be done anywhere in a Salt module before it is loaded. This rule apples to all code that would run before the `__virtual__()` function, as well as the code within the `__virtual__()` function itself.

If logging statements are made before the virtual function determines if the module should be loaded, then those logging statements will be called repeatedly. This clutters up log files unnecessarily.

Exceptions may be considered for logging statements made at the `trace` level. However, it is better to provide the necessary information by another means. One method is to *return error information* in the `__virtual__()` function.

5.2.10 `__virtualname__`

`__virtualname__` is a variable that is used by the documentation build system to know the virtual name of a module without calling the `__virtual__` function. Modules that return a string from the `__virtual__` function must also set the `__virtualname__` variable.

To avoid setting the virtual name string twice, you can implement `__virtual__` to return the value set for `__virtualname__` using a pattern similar to the following:

```
# Define the module's virtual name
__virtualname__ = 'pkg'


def __virtual__():
    '''
    Confine this module to Mac OS with Homebrew.
    '''

    if salt.utils.path.which('brew') and __grains__['os'] == 'MacOS':
        return __virtualname__
    return False
```

The `__virtual__()` function can return a `True` or `False` boolean, a tuple, or a string. If it returns a `True` value, this `__virtualname__` module-level attribute can be set as seen in the above example. This is the string that the module should be referred to as.

When `__virtual__()` returns a tuple, the first item should be a boolean and the second should be a string. This is typically done when the module should not load. The first value of the tuple is `False` and the second is the error message to display for why the module did not load.

For example:

```
def __virtual__():
    '''
    Only load if git exists on the system
    '''
    if salt.utils.path.which('git') is None:
        return (False,
                'The git execution module cannot be loaded: git unavailable.')
    else:
        return True
```

5.2.11 Documentation

Salt execution modules are documented. The `sys.doc()` function will return the documentation for all available modules:

```
salt '*' sys.doc
```

The `sys.doc` function simply prints out the docstrings found in the modules; when writing Salt execution modules, please follow the formatting conventions for docstrings as they appear in the other modules.

Adding Documentation to Salt Modules

It is strongly suggested that all Salt modules have documentation added.

To add documentation add a Python docstring to the function.

```
def spam(eggs):
    '''
    A function to make some spam with eggs!

    CLI Example::

        salt '*' test.spam eggs
    '''
    return eggs
```

Now when the sys.doc call is executed the docstring will be cleanly returned to the calling terminal.

Documentation added to execution modules in docstrings will automatically be added to the online web-based documentation.

Add Execution Module Metadata

When writing a Python docstring for an execution module, add information about the module using the following field lists:

```
:maintainer:    Thomas Hatch <thatch@saltstack.com, Seth House <shouse@saltstack.com>
:maturity:      new
:depends:       python-mysqldb
:platform:      all
```

The maintainer field is a comma-delimited list of developers who help maintain this module.

The maturity field indicates the level of quality and testing for this module. Standard labels will be determined.

The depends field is a comma-delimited list of modules that this module depends on.

The platform field is a comma-delimited list of platforms that this module is known to run on.

5.2.12 Log Output

You can call the logger from custom modules to write messages to the minion logs. The following code snippet demonstrates writing log messages:

```
import logging

log = logging.getLogger(__name__)

log.info('Here is Some Information')
log.warning('You Should Not Do That')
log.error('It Is Busted')
```

5.2.13 Aliasing Functions

Sometimes one wishes to use a function name that would shadow a python built-in. A common example would be `set()`. To support this, append an underscore to the function definition, `def set_():`, and use the `__func_alias__` feature to provide an alias to the function.

`__func_alias__` is a dictionary where each key is the name of a function in the module, and each value is a string representing the alias for that function. When calling an aliased function from a different execution module, state module, or from the cli, the alias name should be used.

```
__func_alias__ = {
    'set_': 'set',
    'list_': 'list',
}
```

5.2.14 Private Functions

In Salt, Python callable objects contained within an execution module are made available to the Salt minion for use. The only exception to this rule is a callable object with a name starting with an underscore `_`.

Objects Loaded Into the Salt Minion

```
def foo(bar):
    return bar
```

Objects NOT Loaded into the Salt Minion

```
def _foobar(baz): # Preceded with an _
    return baz

cheese = {} # Not a callable Python object
```

5.2.15 Useful Decorators for Modules

Depends Decorator

When writing execution modules there are many times where some of the module will work on all hosts but some functions have an external dependency, such as a service that needs to be installed or a binary that needs to be present on the system.

Instead of trying to wrap much of the code in large try/except blocks, a decorator can be used.

If the dependencies passed to the decorator don't exist, then the salt minion will remove those functions from the module on that host.

If a `fallback_function` is defined, it will replace the function instead of removing it

```
import logging

from salt.utils.decorators import depends

log = logging.getLogger(__name__)

try:
    import dependency_that_sometimes_exists
except ImportError as e:
    log.trace('Failed to import dependency_that_sometimes_exists: {0}'.format(e))

@depends('dependency_that_sometimes_exists')
def foo():
    '''
    Function with a dependency on the "dependency_that_sometimes_exists" module,
    if the "dependency_that_sometimes_exists" is missing this function will not exist
    '''
    return True

def _fallback():
    '''
    Fallback function for the depends decorator to replace a function with
    '''
    return '"dependency_that_sometimes_exists" needs to be installed for this
↪function to exist'

@depends('dependency_that_sometimes_exists', fallback_function=_fallback)
def foo():
    '''
    Function with a dependency on the "dependency_that_sometimes_exists" module.
    If the "dependency_that_sometimes_exists" is missing this function will be
    replaced with "_fallback"
    '''
    return True
```

In addition to global dependencies the depends decorator also supports raw booleans.

```
from salt.utils.decorators import depends

HAS_DEP = False
try:
    import dependency_that_sometimes_exists
    HAS_DEP = True
except ImportError:
    pass

@depends(HAS_DEP)
def foo():
    return True
```

5.3 Executors

Executors are used by minion to execute module functions. Executors can be used to modify the functions behavior, do any pre-execution steps or execute in a specific way like sudo executor.

Executors could be passed as a list and they will be used one-by-one in the order. If an executor returns `None` the next one will be called. If an executor returns non-`None` the execution sequence is terminated and the returned value is used as a result. It's a way executor could control modules execution working as a filter. Note that executor could actually not execute the function but just do something else and return `None` like `splay` executor does. In this case some other executor have to be used as a final executor that will actually execute the function. See examples below.

Executors list could be passed by minion config file in the following way:

```
module_executors:
  - splay
  - direct_call
splaytime: 30
```

The same could be done by command line:

```
salt -t 40 --module-executors='[splay, direct_call]' --executor-opts='{splaytime: 30}
↪' '*' test.version
```

And the same command called via netapi will look like this:

```
curl -sSk https://localhost:8000 \
    -H 'Accept: application/x-yaml' \
    -H 'X-Auth-Token: 697adbdc8fe971d09ae4c2a3add7248859c87079' \
    -H 'Content-type: application/json' \
    -d '[{
        "client": "local",
        "tgt": "*",
        "fun": "test.version",
        "module_executors": ["splay", "direct_call"],
        "executor_opts": {"splaytime": 10}
        }]'
```

See also:

The full list of executors

5.3.1 Writing Salt Executors

A Salt executor is written in a similar manner to a Salt execution module. Executor is a python module placed into the `executors` folder and containing the `execute` function with the following signature:

```
def execute(opts, data, func, args, kwargs)
```

Where the args are:

`opts`: Dictionary containing the minion configuration options

`data`: Dictionary containing the load data including `executor_opts` passed via cmdline/API.

`func`, **`args`**, **`kwargs`**: Execution module function to be executed and it's arguments. For instance the simplest `direct_call` executor just runs it as `func(*args, **kwargs)`.

Returns: `None` if the execution sequence must be continued with the next executor. Error string or execution result if the job is done and execution must be stopped.

Specific options could be passed to the executor via minion config or via `executor_opts` argument. For instance to access `splaytime` option set by minion config executor should access `opts.get('splaytime')`. To access the option set by commandline or API `data.get('executor_opts', {}).get('splaytime')` should be used. So if an option is safe and must be accessible by user executor should check it in both places, but if an option is unsafe it should be read from the only config ignoring the passed request data.

There is also a function named `all_missing_func` which the name of the `func` is passed, which can be used to verify if the command should still be run, even if it is not loaded in minion_mods.

CHAPTER

SIX

CONFIGURATION MANAGEMENT

Salt contains a robust and flexible configuration management framework, which is built on the remote execution core. This framework executes on the minions, allowing effortless, simultaneous configuration of tens of thousands of hosts, by rendering language specific state files. The following links provide resources to learn more about state and renderers.

States Express the state of a host using small, easy to read, easy to understand configuration files. *No programming required.*

Full list of states Contains: list of install packages, create users, transfer files, start services, and so on.

Pillar System Contains: description of Salt's Pillar system.

Highstate data structure Contains: a dry vocabulary and technical representation of the configuration format that states represent.

Writing states Contains: a guide on how to write Salt state modules, easily extending Salt to directly manage more software.

Note: Salt execution modules are different from state modules and cannot be called as a state in an SLS file. In other words, this will not work:

```
moe:
  user.rename:
    - new_name: larry
    - onlyif: id moe
```

You must use the `module` states to call execution modules directly. Here's an example:

```
rename_moe:
  module.run:
    - name: user.rename
    - m_name: moe
    - new_name: larry
    - onlyif: id moe
```

Renderers Renderers use state configuration files written in a variety of languages, templating engines, or files. Salt's configuration management system is, under the hood, language agnostic.

Full list of renderers Contains: a list of renderers. YAML is one choice, but many systems are available, from alternative templating engines to the PyDSL language for rendering sls formulas.

Renderers Contains: more information about renderers. Salt states are only concerned with the ultimate highstate data structure, not how the data structure was created.

6.1 State System Reference

Salt offers an interface to manage the configuration or "state" of the Salt minions. This interface is a fully capable mechanism used to enforce the state of systems from a central manager.

6.1.1 Mod Aggregate State Runtime Modifications

New in version 2014.7.0.

The mod_aggregate system was added in the 2014.7.0 release of Salt and allows for runtime modification of the executing state data. Simply put, it allows for the data used by Salt's state system to be changed on the fly at runtime, kind of like a configuration management JIT compiler or a runtime import system. All in all, it makes Salt much more dynamic.

How it Works

The best example is the `pkg` state. One of the major requests in Salt has long been adding the ability to install all packages defined at the same time. The mod_aggregate system makes this a reality. While executing Salt's state system, when a `pkg` state is reached the `mod_aggregate` function in the state module is called. For `pkg` this function scans all of the other states that are slated to run, and picks up the references to `name` and `pkgs`, then adds them to `pkgs` in the first state. The result is a single call to yum, apt-get, pacman, etc as part of the first package install.

How to Use it

Note: Since this option changes the basic behavior of the state runtime, after it is enabled states should be executed using *test=True* to ensure that the desired behavior is preserved.

In config files

The first way to enable aggregation is with a configuration option in either the master or minion configuration files. Salt will invoke `mod_aggregate` the first time it encounters a state module that has aggregate support.

If this option is set in the master config it will apply to all state runs on all minions, if set in the minion config it will only apply to said minion.

Enable for all states:

```
state_aggregate: True
```

Enable for only specific state modules:

```
state_aggregate:
  - pkg
```

In states

The second way to enable aggregation is with the state-level `aggregate` keyword. In this configuration, Salt will invoke the `mod_aggregate` function the first time it encounters this keyword. Any additional occurrences of the keyword will be ignored as the aggregation has already taken place.

The following example will trigger `mod_aggregate` when the `lamp_stack` state is processed resulting in a single call to the underlying package manager.

```
lamp_stack:
  pkg.installed:
    - pkgs:
      - php
      - mysql-client
    - aggregate: True

memcached:
  pkg.installed:
    - name: memcached
```

Adding mod_aggregate to a State Module

Adding a mod_aggregate routine to an existing state module only requires adding an additional function to the state module called mod_aggregate.

The mod_aggregate function just needs to accept three parameters and return the low data to use. Since mod_aggregate is working on the state runtime level it does need to manipulate *low data*.

The three parameters are *low*, *chunks*, and *running*. The *low* option is the low data for the state execution which is about to be called. The *chunks* is the list of all of the low data dictionaries which are being executed by the runtime and the *running* dictionary is the return data from all of the state executions which have already be executed.

This example, simplified from the pkg state, shows how to create mod_aggregate functions:

```
def mod_aggregate(low, chunks, running):
    '''
    The mod_aggregate function which looks up all packages in the available
    low chunks and merges them into a single pkgs ref in the present low data
    '''
    pkgs = []
    # What functions should we aggregate?
    agg_enabled = [
            'installed',
            'latest',
            'removed',
            'purged',
            ]
    # The `low` data is just a dict with the state, function (fun) and
    # arguments passed in from the sls
    if low.get('fun') not in agg_enabled:
        return low
    # Now look into what other things are set to execute
    for chunk in chunks:
        # The state runtime uses "tags" to track completed jobs, it may
        # look familiar with the _|-
        tag = __utils__['state.gen_tag'](chunk)
        if tag in running:
```

(continues on next page)

(continued from previous page)

```
                # Already ran the pkg state, skip aggregation
                continue
            if chunk.get('state') == 'pkg':
                if '__agg__' in chunk:
                    continue
                # Check for the same function
                if chunk.get('fun') != low.get('fun'):
                    continue
                # Pull out the pkg names!
                if 'pkgs' in chunk:
                    pkgs.extend(chunk['pkgs'])
                    chunk['__agg__'] = True
                elif 'name' in chunk:
                    pkgs.append(chunk['name'])
                    chunk['__agg__'] = True
    if pkgs:
        if 'pkgs' in low:
            low['pkgs'].extend(pkgs)
        else:
            low['pkgs'] = pkgs
    # The low has been modified and needs to be returned to the state
    # runtime for execution
    return low
```

6.1.2 Altering States

Note: This documentation has been moved *here*.

6.1.3 File State Backups

In 0.10.2 a new feature was added for backing up files that are replaced by the file.managed and file.recurse states. The new feature is called the backup mode. Setting the backup mode is easy, but it can be set in a number of places.

The backup_mode can be set in the minion config file:

```
backup_mode: minion
```

Or it can be set for each file:

```
/etc/ssh/sshd_config:
  file.managed:
    - source: salt://ssh/sshd_config
    - backup: minion
```

Backed-up Files

The files will be saved in the minion cachedir under the directory named `file_backup`. The files will be in the location relative to where they were under the root filesystem and be appended with a timestamp. This should make them easy to browse.

Interacting with Backups

Starting with version 0.17.0, it will be possible to list, restore, and delete previously-created backups.

Listing

The backups for a given file can be listed using *file.list_backups*:

```
# salt foo.bar.com file.list_backups /tmp/foo.txt
foo.bar.com:
    ----------
    0:
        ----------
        Backup Time:
            Sat Jul 27 2013 17:48:41.738027
        Location:
            /var/cache/salt/minion/file_backup/tmp/foo.txt_Sat_Jul_27_17:48:41_738027_
↪2013
        Size:
            13
    1:
        ----------
        Backup Time:
            Sat Jul 27 2013 17:48:28.369804
        Location:
            /var/cache/salt/minion/file_backup/tmp/foo.txt_Sat_Jul_27_17:48:28_369804_
↪2013
        Size:
            35
```

Restoring

Restoring is easy using *file.restore_backup*, just pass the path and the numeric id found with *file.list_backups*:

```
# salt foo.bar.com file.restore_backup /tmp/foo.txt 1
foo.bar.com:
    ----------
    comment:
        Successfully restored /var/cache/salt/minion/file_backup/tmp/foo.txt_Sat_Jul_
↪27_17:48:28_369804_2013 to /tmp/foo.txt
    result:
        True
```

The existing file will be backed up, just in case, as can be seen if *file.list_backups* is run again:

```
# salt foo.bar.com file.list_backups /tmp/foo.txt
foo.bar.com:
    ----------
    0:
        ----------
        Backup Time:
            Sat Jul 27 2013 18:00:19.822550
        Location:
```

(continues on next page)

(continued from previous page)

```
            /var/cache/salt/minion/file_backup/tmp/foo.txt_Sat_Jul_27_18:00:19_822550_
↪2013
        Size:
            53
    1:
        ----------
        Backup Time:
            Sat Jul 27 2013 17:48:41.738027
        Location:
            /var/cache/salt/minion/file_backup/tmp/foo.txt_Sat_Jul_27_17:48:41_738027_
↪2013
        Size:
            13
    2:
        ----------
        Backup Time:
            Sat Jul 27 2013 17:48:28.369804
        Location:
            /var/cache/salt/minion/file_backup/tmp/foo.txt_Sat_Jul_27_17:48:28_369804_
↪2013
        Size:
            35
```

Note: Since no state is being run, restoring a file will not trigger any watches for the file. So, if you are restoring a config file for a service, it will likely still be necessary to run a `service.restart`.

Deleting

Deleting backups can be done using `file.delete_backup`:

```
# salt foo.bar.com file.delete_backup /tmp/foo.txt 0
foo.bar.com:
    ----------
    comment:
        Successfully removed /var/cache/salt/minion/file_backup/tmp/foo.txt_Sat_Jul_
↪27_18:00:19_822550_2013
    result:
        True
```

6.1.4 Understanding State Compiler Ordering

Note: This tutorial is an intermediate level tutorial. Some basic understanding of the state system and writing Salt Formulas is assumed.

Salt's state system is built to deliver all of the power of configuration management systems without sacrificing simplicity. This tutorial is made to help users understand in detail just how the order is defined for state executions in Salt.

This tutorial is written to represent the behavior of Salt as of version 0.17.0.

Compiler Basics

To understand ordering in depth some very basic knowledge about the state compiler is very helpful. No need to worry though, this is very high level!

High Data and Low Data

When defining Salt Formulas in YAML the data that is being represented is referred to by the compiler as High Data. When the data is initially loaded into the compiler it is a single large python dictionary, this dictionary can be viewed raw by running:

```
salt '*' state.show_highstate
```

This "High Data" structure is then compiled down to "Low Data". The Low Data is what is matched up to create individual executions in Salt's configuration management system. The low data is an ordered list of single state calls to execute. Once the low data is compiled the evaluation order can be seen.

The low data can be viewed by running:

```
salt '*' state.show_lowstate
```

Note: The state execution module contains MANY functions for evaluating the state system and is well worth a read! These routines can be very useful when debugging states or to help deepen one's understanding of Salt's state system.

As an example, a state written thusly:

```
apache:
  pkg.installed:
    - name: httpd
  service.running:
    - name: httpd
    - watch:
      - file: apache_conf
      - pkg: apache

apache_conf:
  file.managed:
    - name: /etc/httpd/conf.d/httpd.conf
    - source: salt://apache/httpd.conf
```

Will have High Data which looks like this represented in json:

```
{
    "apache": {
        "pkg": [
            {
                "name": "httpd"
            },
            "installed",
            {
                "order": 10000
            }
        ],
```

(continues on next page)

(continued from previous page)

```
        "service": [
            {
                "name": "httpd"
            },
            {
                "watch": [
                    {
                        "file": "apache_conf"
                    },
                    {
                        "pkg": "apache"
                    }
                ]
            },
            "running",
            {
                "order": 10001
            }
        ],
        "__sls__": "blah",
        "__env__": "base"
    },
    "apache_conf": {
        "file": [
            {
                "name": "/etc/httpd/conf.d/httpd.conf"
            },
            {
                "source": "salt://apache/httpd.conf"
            },
            "managed",
            {
                "order": 10002
            }
        ],
        "__sls__": "blah",
        "__env__": "base"
    }
}
```

The subsequent Low Data will look like this:

```
[
    {
        "name": "httpd",
        "state": "pkg",
        "__id__": "apache",
        "fun": "installed",
        "__env__": "base",
        "__sls__": "blah",
        "order": 10000
    },
    {
        "name": "httpd",
        "watch": [
            {
```

(continues on next page)

(continued from previous page)

```
                "file": "apache_conf"
            },
            {
                "pkg": "apache"
            }
        ],
        "state": "service",
        "__id__": "apache",
        "fun": "running",
        "__env__": "base",
        "__sls__": "blah",
        "order": 10001
    },
    {
        "name": "/etc/httpd/conf.d/httpd.conf",
        "source": "salt://apache/httpd.conf",
        "state": "file",
        "__id__": "apache_conf",
        "fun": "managed",
        "__env__": "base",
        "__sls__": "blah",
        "order": 10002
    }
]
```

This tutorial discusses the Low Data evaluation and the state runtime.

Ordering Layers

Salt defines 2 order interfaces which are evaluated in the state runtime and defines these orders in a number of passes.

Definition Order

Note: The Definition Order system can be disabled by turning the option `state_auto_order` to `False` in the master configuration file.

The top level of ordering is the *Definition Order*. The *Definition Order* is the order in which states are defined in salt formulas. This is very straightforward on basic states which do not contain `include` statements or a `top` file, as the states are just ordered from the top of the file, but the include system starts to bring in some simple rules for how the *Definition Order* is defined.

Looking back at the ”Low Data” and ”High Data” shown above, the order key has been transparently added to the data to enable the *Definition Order*.

The Include Statement

Basically, if there is an include statement in a formula, then the formulas which are included will be run BEFORE the contents of the formula which is including them. Also, the include statement is a list, so they will be loaded in the order in which they are included.

In the following case:

`foo.sls`

```
include:
  - bar
  - baz
```

`bar.sls`

```
include:
  - quo
```

`baz.sls`

```
include:
  - qux
```

In the above case if `state.apply foo` were called then the formulas will be loaded in the following order:

1. quo
2. bar
3. qux
4. baz
5. foo

The *order* Flag

The *Definition Order* happens transparently in the background, but the ordering can be explicitly overridden using the `order` flag in states:

```
apache:
  pkg.installed:
    - name: httpd
    - order: 1
```

This order flag will over ride the definition order, this makes it very simple to create states that are always executed first, last or in specific stages, a great example is defining a number of package repositories that need to be set up before anything else, or final checks that need to be run at the end of a state run by using `order: last` or `order: -1`.

When the order flag is explicitly set the *Definition Order* system will omit setting an order for that state and directly use the order flag defined.

Lexicographical Fall-back

Salt states were written to ALWAYS execute in the same order. Before the introduction of *Definition Order* in version 0.17.0 everything was ordered lexicographically according to the name of the state, then function then id.

This is the way Salt has always ensured that states always run in the same order regardless of where they are deployed, the addition of the *Definition Order* method mealy makes this finite ordering easier to follow.

The lexicographical ordering is still applied but it only has any effect when two order statements collide. This means that if multiple states are assigned the same order number that they will fall back to lexicographical ordering to ensure that every execution still happens in a finite order.

Note: If running with `state_auto_order: False` the `order` key is not set automatically, since the Lexicographical order can be derived from other keys.

Requisite Ordering

Salt states are fully declarative, in that they are written to declare the state in which a system should be. This means that components can require that other components have been set up successfully. Unlike the other ordering systems, the *Requisite* system in Salt is evaluated at runtime.

The requisite system is also built to ensure that the ordering of execution never changes, but is always the same for a given set of states. This is accomplished by using a runtime that processes states in a completely predictable order instead of using an event loop based system like other declarative configuration management systems.

Runtime Requisite Evaluation

The requisite system is evaluated as the components are found, and the requisites are always evaluated in the same order. This explanation will be followed by an example, as the raw explanation may be a little dizzying at first as it creates a linear dependency evaluation sequence.

The "Low Data" is an ordered list or dictionaries, the state runtime evaluates each dictionary in the order in which they are arranged in the list. When evaluating a single dictionary it is checked for requisites, requisites are evaluated in order, `require` then `watch` then `prereq`.

Note: If using requisite in statements like require_in and watch_in these will be compiled down to require and watch statements before runtime evaluation.

Each requisite contains an ordered list of requisites, these requisites are looked up in the list of dictionaries and then executed. Once all requisites have been evaluated and executed then the requiring state can safely be run (or not run if requisites have not been met).

This means that the requisites are always evaluated in the same order, again ensuring one of the core design principals of Salt's State system to ensure that execution is always finite is intact.

Simple Runtime Evaluation Example

Given the above "Low Data" the states will be evaluated in the following order:

1. The pkg.installed is executed ensuring that the apache package is installed, it contains no requisites and is therefore the first defined state to execute.
2. The service.running state is evaluated but NOT executed, a watch requisite is found, therefore they are read in order, the runtime first checks for the file, sees that it has not been executed and calls for the file state to be evaluated.
3. The file state is evaluated AND executed, since it, like the pkg state does not contain any requisites.
4. The evaluation of the service state continues, it next checks the pkg requisite and sees that it is met, with all requisites met the service state is now executed.

Best Practice

The best practice in Salt is to choose a method and stick with it, official states are written using requisites for all associations since requisites create clean, traceable dependency trails and make for the most portable formulas. To accomplish something similar to how classical imperative systems function all requisites can be omitted and the `failhard` option then set to `True` in the master configuration, this will stop all state runs at the first instance of a failure.

In the end, using requisites creates very tight and fine grained states, not using requisites makes full sequence runs and while slightly easier to write, and gives much less control over the executions.

6.1.5 Extending External SLS Data

Sometimes a state defined in one SLS file will need to be modified from a separate SLS file. A good example of this is when an argument needs to be overwritten or when a service needs to watch an additional state.

The Extend Declaration

The standard way to extend is via the extend declaration. The extend declaration is a top level declaration like `include` and encapsulates ID declaration data included from other SLS files. A standard extend looks like this:

```
include:
  - http
  - ssh

extend:
  apache:
    file:
      - name: /etc/httpd/conf/httpd.conf
      - source: salt://http/httpd2.conf
  ssh-server:
    service:
      - watch:
        - file: /etc/ssh/banner

/etc/ssh/banner:
  file.managed:
    - source: salt://ssh/banner
```

A few critical things happened here, first off the SLS files that are going to be extended are included, then the extend dec is defined. Under the extend dec 2 IDs are extended, the apache ID's file state is overwritten with a new name and source. Then the ssh server is extended to watch the banner file in addition to anything it is already watching.

Extend is a Top Level Declaration

This means that `extend` can only be called once in an sls, if it is used twice then only one of the extend blocks will be read. So this is WRONG:

```
include:
  - http
  - ssh

extend:
```

(continues on next page)

(continued from previous page)

```
  apache:
    file:
      - name: /etc/httpd/conf/httpd.conf
      - source: salt://http/httpd2.conf
# Second extend will overwrite the first!! Only make one
extend:
  ssh-server:
    service:
      - watch:
        - file: /etc/ssh/banner
```

The Requisite "in" Statement

Since one of the most common things to do when extending another SLS is to add states for a service to watch, or anything for a watcher to watch, the requisite in statement was added to 0.9.8 to make extending the watch and require lists easier. The ssh-server extend statement above could be more cleanly defined like so:

```
include:
  - ssh

/etc/ssh/banner:
  file.managed:
    - source: salt://ssh/banner
    - watch_in:
      - service: ssh-server
```

Rules to Extend By

There are a few rules to remember when extending states:

1. Always include the SLS being extended with an include declaration
2. Requisites (watch and require) are appended to, everything else is overwritten
3. extend is a top level declaration, like an ID declaration, cannot be declared twice in a single SLS
4. Many IDs can be extended under the extend declaration

6.1.6 Failhard Global Option

Normally, when a state fails Salt continues to execute the remainder of the defined states and will only refuse to execute states that require the failed state.

But the situation may exist, where you would want all state execution to stop if a single state execution fails. The capability to do this is called `failing hard`.

State Level Failhard

A single state can have a failhard set, this means that if this individual state fails that all state execution will immediately stop. This is a great thing to do if there is a state that sets up a critical config file and setting a require for each state that reads the config would be cumbersome. A good example of this would be setting up a package manager early on:

```
/etc/yum.repos.d/company.repo:
  file.managed:
    - source: salt://company/yumrepo.conf
    - user: root
    - group: root
    - mode: 644
    - order: 1
    - failhard: True
```

In this situation, the yum repo is going to be configured before other states, and if it fails to lay down the config file, than no other states will be executed. It is possible to override a Global Failhard (see below) by explicitly setting it to `False` in the state.

Global Failhard

It may be desired to have failhard be applied to every state that is executed, if this is the case, then failhard can be set in the master configuration file. Setting failhard in the master configuration file will result in failing hard when any minion gathering states from the master have a state fail.

This is NOT the default behavior, normally Salt will only fail states that require a failed state.

Using the global failhard is generally not recommended, since it can result in states not being executed or even checked. It can also be confusing to see states failhard if an admin is not actively aware that the failhard has been set.

To use the global failhard set *`failhard`* to `True` in the master configuration file.

6.1.7 Global State Arguments

Note: This documentation has been moved *here*.

6.1.8 Highstate data structure definitions

The Salt State Tree

A state tree is a collection of `SLS` files and directories that live under the directory specified in *`file_roots`*.

Note: Directory names or filenames in the state tree cannot contain a period, with the exception of the period in the .sls file suffix.

Top file

The main state file that instructs minions what environment and modules to use during state execution.

Configurable via *`state_top`*.

See also:

A detailed description of the top file

Include declaration

Defines a list of *Module reference* strings to include in this `SLS`.

Occurs only in the top level of the SLS data structure.

Example:

```
include:
  - edit.vim
  - http.server
```

Module reference

The name of a SLS module defined by a separate SLS file and residing on the Salt Master. A module named `edit.vim` is a reference to the SLS file `salt://edit/vim.sls`.

ID declaration

Defines an individual *highstate* component. Always references a value of a dictionary containing keys referencing *State declaration* and *Requisite declaration*. Can be overridden by a *Name declaration* or a *Names declaration*.

Occurs on the top level or under the *Extend declaration*.

Must be unique across entire state tree. If the same ID declaration is used twice, only the first one matched will be used. All subsequent ID declarations with the same name will be ignored.

Note: Naming gotchas

In Salt versions earlier than 0.9.7, ID declarations containing dots would result in unpredictable output.

Extend declaration

Extends a *Name declaration* from an included `SLS module`. The keys of the extend declaration always refer to an existing *ID declaration* which have been defined in included `SLS modules`.

Occurs only in the top level and defines a dictionary.

States cannot be extended more than once in a single state run.

Extend declarations are useful for adding-to or overriding parts of a *State declaration* that is defined in another `SLS` file. In the following contrived example, the shown `mywebsite.sls` file is `include` -ing and `extend` -ing the `apache.sls` module in order to add a `watch` declaration that will restart Apache whenever the Apache configuration file, `mywebsite` changes.

```
include:
  - apache

extend:
  apache:
    service:
      - watch:
        - file: mywebsite
```

(continues on next page)

(continued from previous page)

```
mywebsite:
  file.managed:
    - name: /var/www/mysite
```

See also:

watch_in and require_in

Sometimes it is more convenient to use the *watch_in* or *require_in* syntax instead of extending another `SLS` file.

State Requisites

State declaration

A list which contains one string defining the *Function declaration* and any number of *Function arg declaration* dictionaries.

Can, optionally, contain a number of additional components like the name override components — *name* and *names*. Can also contain *requisite declarations*.

Occurs under an *ID declaration*.

Requisite declaration

A list containing *requisite references*.

Used to build the action dependency tree. While Salt states are made to execute in a deterministic order, this order is managed by requiring and watching other Salt states.

Occurs as a list component under a *State declaration* or as a key under an *ID declaration*.

Requisite reference

A single key dictionary. The key is the name of the referenced *State declaration* and the value is the ID of the referenced *ID declaration*.

Occurs as a single index in a *Requisite declaration* list.

Function declaration

The name of the function to call within the state. A state declaration can contain only a single function declaration.

For example, the following state declaration calls the `installed` function in the `pkg` state module:

```
httpd:
  pkg.installed: []
```

The function can be declared inline with the state as a shortcut. The actual data structure is compiled to this form:

```
httpd:
  pkg:
    - installed
```

Where the function is a string in the body of the state declaration. Technically when the function is declared in dot notation the compiler converts it to be a string in the state declaration list. Note that the use of the first example more than once in an ID declaration is invalid yaml.

INVALID:

```
httpd:
  pkg.installed
  service.running
```

When passing a function without arguments and another state declaration within a single ID declaration, then the long or "standard" format needs to be used since otherwise it does not represent a valid data structure.

VALID:

```
httpd:
  pkg.installed: []
  service.running: []
```

Occurs as the only index in the *State declaration* list.

Function arg declaration

A single key dictionary referencing a Python type which is to be passed to the named *Function declaration* as a parameter. The type must be the data type expected by the function.

Occurs under a *Function declaration*.

For example in the following state declaration `user`, `group`, and `mode` are passed as arguments to the *`managed`* function in the `file` state module:

```
/etc/http/conf/http.conf:
  file.managed:
    - user: root
    - group: root
    - mode: 644
```

Name declaration

Overrides the `name` argument of a *State declaration*. If `name` is not specified the *ID declaration* satisfies the `name` argument.

The name is always a single key dictionary referencing a string.

Overriding `name` is useful for a variety of scenarios.

For example, avoiding clashing ID declarations. The following two state declarations cannot both have `/etc/motd` as the ID declaration:

```
motd_perms:
  file.managed:
    - name: /etc/motd
    - mode: 644

motd_quote:
  file.append:
```

(continues on next page)

(continued from previous page)

```
    - name: /etc/motd
    - text: "Of all smells, bread; of all tastes, salt."
```

Another common reason to override `name` is if the ID declaration is long and needs to be referenced in multiple places. In the example below it is much easier to specify `mywebsite` than to specify `/etc/apache2/sites-available/mywebsite.com` multiple times:

```
mywebsite:
  file.managed:
    - name: /etc/apache2/sites-available/mywebsite.com
    - source: salt://mywebsite.com

a2ensite mywebsite.com:
  cmd.wait:
    - unless: test -L /etc/apache2/sites-enabled/mywebsite.com
    - watch:
      - file: mywebsite

apache2:
  service.running:
    - watch:
      - file: mywebsite
```

Names declaration

Expands the contents of the containing *State declaration* into multiple state declarations, each with its own name.

For example, given the following state declaration:

```
python-pkgs:
  pkg.installed:
    - names:
      - python-django
      - python-crypto
      - python-yaml
```

Once converted into the lowstate data structure the above state declaration will be expanded into the following three state declarations:

```
python-django:
  pkg.installed

python-crypto:
  pkg.installed

python-yaml:
  pkg.installed
```

Other values can be overridden during the expansion by providing an additional dictionary level.

New in version 2014.7.0.

```
ius:
  pkgrepo.managed:
    - humanname: IUS Community Packages for Enterprise Linux 6 - $basearch
```

(continues on next page)

(continued from previous page)

```
    - gpgcheck: 1
    - baseurl: http://mirror.rackspace.com/ius/stable/CentOS/6/$basearch
    - gpgkey: http://dl.iuscommunity.org/pub/ius/IUS-COMMUNITY-GPG-KEY
    - names:
        - ius
        - ius-devel:
            - baseurl: http://mirror.rackspace.com/ius/development/CentOS/6/$basearch
```

Large example

Here is the layout in yaml using the names of the highdata structure components.

```
<Include Declaration>:
  - <Module Reference>
  - <Module Reference>

<Extend Declaration>:
  <ID Declaration>:
    [<overrides>]


# standard declaration

<ID Declaration>:
  <State Module>:
    - <Function>
    - <Function Arg>
    - <Function Arg>
    - <Function Arg>
    - <Name>: <name>
    - <Requisite Declaration>:
      - <Requisite Reference>
      - <Requisite Reference>


# inline function and names

<ID Declaration>:
  <State Module>.<Function>:
    - <Function Arg>
    - <Function Arg>
    - <Function Arg>
    - <Names>:
      - <name>
      - <name>
      - <name>
    - <Requisite Declaration>:
      - <Requisite Reference>
      - <Requisite Reference>


# multiple states for single id

<ID Declaration>:
  <State Module>:
```

(continues on next page)

(continued from previous page)

```
    - <Function>
    - <Function Arg>
    - <Name>: <name>
    - <Requisite Declaration>:
      - <Requisite Reference>
  <State Module>:
    - <Function>
    - <Function Arg>
    - <Names>:
      - <name>
      - <name>
    - <Requisite Declaration>:
      - <Requisite Reference>
```

6.1.9 Include and Exclude

Salt SLS files can include other SLS files and exclude SLS files that have been otherwise included. This allows for an SLS file to easily extend or manipulate other SLS files.

Include

When other SLS files are included, everything defined in the included SLS file will be added to the state run. When including define a list of SLS formulas to include:

```
include:
  - http
  - libvirt
```

The include statement will include SLS formulas from the same environment that the including SLS formula is in. But the environment can be explicitly defined in the configuration to override the running environment, therefore if an SLS formula needs to be included from an external environment named "dev" the following syntax is used:

```
include:
  - dev: http
```

NOTE: `include` does not simply inject the states where you place it in the SLS file. If you need to guarantee order of execution, consider using requisites.

Do not use dots in SLS file names or their directories

The initial implementation of `top.sls` and *Include declaration* followed the python import model where a slash is represented as a period. This means that a SLS file with a period in the name (besides the suffix period) can not be referenced. For example, webserver_1.0.sls is not referenceable because webserver_1.0 would refer to the directory/file webserver_1/0.sls

The same applies for any subdirectories, this is especially 'tricky' when git repos are created. Another command that typically can't render it's output is ``state.show_sls`` of a file in a path that contains a dot.

Relative Include

In Salt 0.16.0, the capability to include SLS formulas which are relative to the running SLS formula was added. Simply precede the formula name with a `.`:

```
include:
  - .virt
  - .virt.hyper
```

In Salt 2015.8, the ability to include SLS formulas which are relative to the parents of the running SLS formula was added. In order to achieve this, precede the formula name with more than one `.` (dot). Much like Python's relative import abilities, two or more leading dots represent a relative include of the parent or parents of the current package, with each `.` representing one level after the first.

The following SLS configuration, if placed within `example.dev.virtual`, would result in `example.http` and `base` being included respectively:

```
include:
  - ..http
  - ...base
```

Exclude

The exclude statement, added in Salt 0.10.3, allows an SLS to hard exclude another SLS file or a specific id. The component is excluded after the high data has been compiled, so nothing should be able to override an exclude.

Since the exclude can remove an id or an sls the type of component to exclude needs to be defined. An exclude statement that verifies that the running *highstate* does not contain the `http` sls and the `/etc/vimrc` id would look like this:

```
exclude:
  - sls: http
  - id: /etc/vimrc
```

Note: The current state processing flow checks for duplicate IDs before processing excludes. An error occurs if duplicate IDs are present even if one of the IDs is targeted by an `exclude`.

6.1.10 State System Layers

The Salt state system is comprised of multiple layers. While using Salt does not require an understanding of the state layers, a deeper understanding of how Salt compiles and manages states can be very beneficial.

Function Call

The lowest layer of functionality in the state system is the direct state function call. State executions are executions of single state functions at the core. These individual functions are defined in state modules and can be called directly via the `state.single` command.

```
salt '*' state.single pkg.installed name='vim'
```

Low Chunk

The low chunk is the bottom of the Salt state compiler. This is a data representation of a single function call. The low chunk is sent to the state caller and used to execute a single state function.

A single low chunk can be executed manually via the `state.low` command.

```
salt '*' state.low '{name: vim, state: pkg, fun: installed}'
```

The passed data reflects what the state execution system gets after compiling the data down from sls formulas.

Low State

The *Low State* layer is the list of low chunks "evaluated" in order. To see what the low state looks like for a *highstate*, run:

```
salt '*' state.show_lowstate
```

This will display the raw lowstate in the order which each low chunk will be evaluated. The order of evaluation is not necessarily the order of execution, since requisites are evaluated at runtime. Requisite execution and evaluation is finite; this means that the order of execution can be ascertained with 100% certainty based on the order of the low state.

High Data

High data is the data structure represented in YAML via SLS files. The High data structure is created by merging the data components rendered inside sls files (or other render systems). The High data can be easily viewed by executing the `state.show_highstate` or `state.show_sls` functions. Since this data is a somewhat complex data structure, it may be easier to read using the json, yaml, or pprint outputters:

```
salt '*' state.show_highstate --out yaml
salt '*' state.show_sls edit.vim --out pprint
```

SLS

Above "High Data", the logical layers are no longer technically required to be executed, or to be executed in a hierarchy. This means that how the High data is generated is optional and very flexible. The SLS layer allows for many mechanisms to be used to render sls data from files or to use the fileserver backend to generate sls and file data from external systems.

The SLS layer can be called directly to execute individual sls formulas.

Note: SLS Formulas have historically been called "SLS files". This is because a single SLS was only constituted in a single file. Now the term "SLS Formula" better expresses how a compartmentalized SLS can be expressed in a much more dynamic way by combining pillar and other sources, and the SLS can be dynamically generated.

To call a single SLS formula named `edit.vim`, execute *`state.apply`* and pass `edit.vim` as an argument:

```
salt '*' state.apply edit.vim
```

HighState

Calling SLS directly logically assigns what states should be executed from the context of the calling minion. The Highstate layer is used to allow for full contextual assignment of what is executed where to be tied to groups of, or individual, minions entirely from the master. This means that the environment of a minion, and all associated execution data pertinent to said minion, can be assigned from the master without needing to execute or configure anything on the target minion. This also means that the minion can independently retrieve information about its complete configuration from the master.

To execute the *highstate* use `state.apply`:

```
salt '*' state.apply
```

Orchestrate

The orchestrate layer expresses the highest functional layer of Salt's automated logic systems. The Overstate allows for stateful and functional orchestration of routines from the master. The orchestrate defines in data execution stages which minions should execute states, or functions, and in what order using requisite logic.

6.1.11 The Orchestrate Runner

Note: This documentation has been moved *here*.

6.1.12 Ordering States

The way in which configuration management systems are executed is a hotly debated topic in the configuration management world. Two major philosophies exist on the subject, to either execute in an imperative fashion where things are executed in the order in which they are defined, or in a declarative fashion where dependencies need to be mapped between objects.

Imperative ordering is finite and generally considered easier to write, but declarative ordering is much more powerful and flexible but generally considered more difficult to create.

Salt has been created to get the best of both worlds. States are evaluated in a finite order, which guarantees that states are always executed in the same order, and the states runtime is declarative, making Salt fully aware of dependencies via the *requisite* system.

State Auto Ordering

Salt always executes states in a finite manner, meaning that they will always execute in the same order regardless of the system that is executing them. This evaluation order makes it easy to know what order the states will be executed in, but it is important to note that the requisite system will override the ordering defined in the files, and the `order` option, described below, will also override the order in which states are executed.

This ordering system can be disabled in preference of lexicographic (classic) ordering by setting the `state_auto_order` option to `False` in the master configuration file. Otherwise, `state_auto_order` defaults to `True`.

How compiler ordering is managed is described further in *Understanding State Compiler Ordering*.

Requisite Statements

Note: The behavior of requisites changed in version 0.9.7 of Salt. This documentation applies to requisites in version 0.9.7 and later.

Often when setting up states any single action will require or depend on another action. Salt allows for the building of relationships between states with requisite statements. A requisite statement ensures that the named state is evaluated before the state requiring it. There are three types of requisite statements in Salt, **require**, **watch**, and **prereq**.

These requisite statements are applied to a specific state declaration:

```
httpd:
  pkg.installed: []
  file.managed:
    - name: /etc/httpd/conf/httpd.conf
    - source: salt://httpd/httpd.conf
    - require:
      - pkg: httpd
```

In this example, the **require** requisite is used to declare that the file /etc/httpd/conf/httpd.conf should only be set up if the pkg state executes successfully.

The requisite system works by finding the states that are required and executing them before the state that requires them. Then the required states can be evaluated to see if they have executed correctly.

Require statements can refer to any state defined in Salt. The basic examples are *pkg*, *service*, and *file*, but any used state can be referenced.

In addition to state declarations such as pkg, file, etc., **sls** type requisites are also recognized, and essentially allow 'chaining' of states. This provides a mechanism to ensure the proper sequence for complex state formulas, especially when the discrete states are split or groups into separate sls files:

```
include:
  - network

httpd:
  pkg.installed: []
  service.running:
    - require:
      - pkg: httpd
      - sls: network
```

In this example, the httpd service running state will not be applied (i.e., the httpd service will not be started) unless both the httpd package is installed AND the network state is satisfied.

Note: Requisite matching

Requisites match on both the ID Declaration and the `name` parameter. Therefore, if using the `pkgs` or `sources` argument to install a list of packages in a pkg state, it's important to note that it is impossible to match an individual package in the list, since all packages are installed as a single state.

Multiple Requisites

The requisite statement is passed as a list, allowing for the easy addition of more requisites. Both requisite types can also be separately declared:

```
httpd:
  pkg.installed: []
  service.running:
```

(continues on next page)

(continued from previous page)

```
    - enable: True
    - watch:
      - file: /etc/httpd/conf/httpd.conf
    - require:
      - pkg: httpd
      - user: httpd
      - group: httpd
  file.managed:
    - name: /etc/httpd/conf/httpd.conf
    - source: salt://httpd/httpd.conf
    - require:
      - pkg: httpd
  user.present: []
  group.present: []
```

In this example, the httpd service is only going to be started if the package, user, group, and file are executed successfully.

Requisite Documentation

For detailed information on each of the individual requisites, *please look here.*

The Order Option

Before using the *order* option, remember that the majority of state ordering should be done with a *Requisite declaration*, and that a requisite declaration will override an *order* option, so a state with order option should not require or required by other states.

The order option is used by adding an order number to a state declaration with the option *order*:

```
vim:
  pkg.installed:
    - order: 1
```

By adding the order option to *1* this ensures that the vim package will be installed in tandem with any other state declaration set to the order *1*.

Any state declared without an order option will be executed after all states with order options are executed.

But this construct can only handle ordering states from the beginning. Certain circumstances will present a situation where it is desirable to send a state to the end of the line. To do this, set the order to `last`:

```
vim:
  pkg.installed:
    - order: last
```

6.1.13 Running States in Parallel

Introduced in Salt version `2017.7.0` it is now possible to run select states in parallel. This is accomplished very easily by adding the `parallel: True` option to your state declaration:

```
nginx:
  service.running:
    - parallel: True
```

Now `nginx` will be started in a separate process from the normal state run and will therefore not block additional states.

Parallel States and Requisites

Parallel States still honor requisites. If a given state requires another state that has been run in parallel then the state runtime will wait for the required state to finish.

Given this example:

```
sleep 10:
  cmd.run:
    - parallel: True

nginx:
  service.running:
    - parallel: True
    - require:
      - cmd: sleep 10

sleep 5:
  cmd.run:
    - parallel: True
```

The `sleep 10` will be started first, then the state system will block on starting nginx until the `sleep 10` completes. Once nginx has been ensured to be running then the `sleep 5` will start.

This means that the order of evaluation of Salt States and requisites are still honored, and given that in the above case, `parallel: True` does not actually speed things up.

To run the above state much faster make sure that the `sleep 5` is evaluated before the `nginx` state

```
sleep 10:
  cmd.run:
    - parallel: True

sleep 5:
  cmd.run:
    - parallel: True

nginx:
  service.running:
    - parallel: True
    - require:
      - cmd: sleep 10
```

Now both of the sleep calls will be started in parallel and `nginx` will still wait for the state it requires, but while it waits the `sleep 5` state will also complete.

Things to be Careful of

Parallel States do not prevent you from creating parallel conflicts on your system. This means that if you start multiple package installs using Salt then the package manager will block or fail. If you attempt to manage the same file with multiple states in parallel then the result can produce an unexpected file.

Make sure that the states you choose to run in parallel do not conflict, or else, like in any parallel programming environment, the outcome may not be what you expect. Doing things like just making all states run in parallel will almost certainly result in unexpected behavior.

With that said, running states in parallel should be safe the vast majority of the time and the most likely culprit for unexpected behavior is running multiple package installs in parallel.

6.1.14 State Providers

New in version 0.9.8.

Salt predetermines what modules should be mapped to what uses based on the properties of a system. These determinations are generally made for modules that provide things like package and service management.

Sometimes in states, it may be necessary to use an alternative module to provide the needed functionality. For instance, an very old Arch Linux system may not be running systemd, so instead of using the systemd service module, you can revert to the default service module:

```
httpd:
  service.running:
    - enable: True
    - provider: service
```

In this instance, the basic *service* module (which manages **sysvinit**-based services) will replace the `systemd` module which is used by default on Arch Linux.

This change only affects this one state though. If it is necessary to make this override for most or every service, it is better to just override the provider in the minion config file, as described *here*.

Also, keep in mind that this only works for states with an identically-named virtual module (*pkg*, *service*, etc.).

Arbitrary Module Redirects

The provider statement can also be used for more powerful means, instead of overwriting or extending the module used for the named service an arbitrary module can be used to provide certain functionality.

```
emacs:
  pkg.installed:
    - provider:
      - cmd: customcmd
```

In this example, the state is being instructed to use a custom module to invoke commands.

Arbitrary module redirects can be used to dramatically change the behavior of a given state.

6.1.15 Requisites and Other Global State Arguments

Requisites

The Salt requisite system is used to create relationships between states. The core idea being that, when one state is dependent somehow on another, that inter-dependency can be easily defined. These dependencies are expressed by declaring the relationships using state names and ID's or names. The generalized form of a requisite target is `<state name> : <ID or name>`. The specific form is defined as a *Requisite Reference*

Requisites come in two types: Direct requisites (such as `require`), and requisite_ins (such as `require_in`). The relationships are directional: a direct requisite requires something from another state. However, a requisite_in inserts a requisite into the targeted state pointing to the targeting state. The following example demonstrates a direct requisite:

```
vim:
  pkg.installed

/etc/vimrc:
  file.managed:
    - source: salt://edit/vimrc
    - require:
      - pkg: vim
```

In the example above, the file `/etc/vimrc` depends on the vim package.

Requisite_in statements are the opposite. Instead of saying "I depend on something", requisite_ins say "Someone depends on me":

```
vim:
  pkg.installed:
    - require_in:
      - file: /etc/vimrc

/etc/vimrc:
  file.managed:
    - source: salt://edit/vimrc
```

So here, with a requisite_in, the same thing is accomplished as in the first example, but the other way around. The vim package is saying "/etc/vimrc depends on me". This will result in a `require` being inserted into the `/etc/vimrc` state which targets the `vim` state.

In the end, a single dependency map is created and everything is executed in a finite and predictable order.

Requisite matching

Requisites need two pieces of information for matching: The state module name – e.g. `pkg` –, and the identifier – e.g. vim –, which can be either the ID (the first line in the stanza) or the `- name` parameter.

```
- require:
  - pkg: vim
```

Glog matching in requisites

New in version 0.9.8.

Glob matching is supported in requisites. This is mostly useful for file changes. In the example below, a change in `/etc/apache2/httpd.conf` or `/etc/apache2/sites-available/default.conf` will reload/restart the service:

```
apache2:
  service.running:
    - watch:
      - file: /etc/apache2/*
```

Omitting state module in requisites

New in version 2016.3.0.

In version 2016.3.0, the state module name was made optional. If the state module is omitted, all states matching the ID will be required, regardless of which module they are using.

```
- require:
  - vim
```

State target matching

In order to understand how state targets are matched, it is helpful to know *how the state compiler is working*. Consider the following example:

```
Deploy server package:
  file.managed:
    - name: /usr/local/share/myapp.tar.xz
    - source: salt://myapp.tar.xz

Extract server package:
  archive.extracted:
    - name: /usr/local/share/myapp
    - source: /usr/local/share/myapp.tar.xz
    - archive_format: tar
    - onchanges:
      - file: Deploy server package
```

The first formula is converted to a dictionary which looks as follows (represented as YAML, some properties omitted for simplicity) as *High Data*:

```
Deploy server package:
  file:
    - managed
    - name: /usr/local/share/myapp.tar.xz
    - source: salt://myapp.tar.xz
```

The `file.managed` format used in the formula is essentially syntactic sugar: at the end, the target is `file`, which is used in the `Extract server package` state above.

Identifier matching

Requisites match on both the ID Declaration and the `name` parameter. This means that, in the "Deploy server package" example above, a `require` requisite would match with `Deploy server package` *or* `/usr/local/share/myapp.tar.xz`, so either of the following versions for "Extract server package" works:

```
# (Archive arguments omitted for simplicity)

# Match by ID declaration
Extract server package:
  archive.extracted:
    - onchanges:
      - file: Deploy server package

# Match by name parameter
Extract server package:
  archive.extracted:
    - onchanges:
      - file: /usr/local/share/myapp.tar.xz
```

Requisite overview

name of requisite	state is only executed if target execution result is	state is only executed if target has changes	order 1.target 2.state (default)	**comment** or description
require	success		default	state will always execute unless target fails
watch	success		default	like require, but adds additional behaviour (mod_watch)
prereq	success	has changes (run individually as dry-run)	switched	like onchanges, except order
onchanges	success	has changes	default	execute state if target execution result is success and target has changes
onfail	failed		default	Only requisite where state exec. if target fails

In this table, the following short form of terms is used:

- **state** (= dependent state): state containing requisite
- **target** (= state target) : state referenced by requisite

Direct Requisite and Requisite_in types

There are several direct requisite statements that can be used in Salt:

- `require`
- `watch`
- `prereq`

- `use`
- `onchanges`
- `onfail`

Each direct requisite also has a corresponding requisite_in:

- `require_in`
- `watch_in`
- `prereq_in`
- `use_in`
- `onchanges_in`
- `onfail_in`

There are several corresponding requisite_any statements:

- `require_any`
- `watch_any`
- `onchanges_any`
- `onfail_any`

All of the requisites define specific relationships and always work with the dependency logic defined above.

require

The use of `require` demands that the required state executes before the dependent state. The state containing the `require` requisite is defined as the dependent state. The state specified in the `require` statement is defined as the required state. If the required state's execution succeeds, the dependent state will then execute. If the required state's execution fails, the dependent state will not execute. In the first example above, the file `/etc/vimrc` will only execute after the vim package is installed successfully.

Require an Entire SLS File

As of Salt 0.16.0, it is possible to require an entire sls file. Do this first by including the sls file and then setting a state to `require` the included sls file:

```
include:
  - foo

bar:
  pkg.installed:
    - require:
      - sls: foo
```

This will add all of the state declarations found in the given sls file. This means that every state in sls *foo* will be required. This makes it very easy to batch large groups of states easily in any requisite statement.

require_any

New in version 2018.3.0.

The use of `require_any` demands that one of the required states executes before the dependent state. The state containing the `require_any` requisite is defined as the dependent state. The states specified in the `require_any` statement are defined as the required states. If at least one of the required state's execution succeeds, the dependent state will then execute. If all of the executions by the required states fail, the dependent state will not execute.

```
A:
  cmd.run:
    - name: echo A
    - require_any:
      - cmd: B
      - cmd: C
      - cmd: D
B:
  cmd.run:
    - name: echo B

C:
  cmd.run:
    - name: /bin/false

D:
  cmd.run:
    - name: echo D
```

In this example *A* will run because at least one of the requirements specified, *B*, *C*, or *D* will succeed.

watch

`watch` statements are used to add additional behavior when there are changes in other states.

Note: If a state should only execute when another state has changes, and otherwise do nothing, the new `onchanges` requisite should be used instead of `watch`. `watch` is designed to add *additional* behavior when there are changes, but otherwise the state executes normally.

The state containing the `watch` requisite is defined as the watching state. The state specified in the `watch` statement is defined as the watched state. When the watched state executes, it will return a dictionary containing a key named "changes". Here are two examples of state return dictionaries, shown in json for clarity:

```
{
    "local": {
        "file_|-/tmp/foo_|-/tmp/foo_|-directory": {
            "comment": "Directory /tmp/foo updated",
            "__run_num__": 0,
            "changes": {
                "user": "bar"
            },
            "name": "/tmp/foo",
            "result": true
        }
    }
```

(continues on next page)

(continued from previous page)

```
}

{
    "local": {
        "pkgrepo_|-salt-minion_|-salt-minion_|-managed": {
            "comment": "Package repo 'salt-minion' already configured",
            "__run_num__": 0,
            "changes": {},
            "name": "salt-minion",
            "result": true
        }
    }
}
```

If the "result" of the watched state is `True`, the watching state *will execute normally*, and if it is `False`, the watching state will never run. This part of `watch` mirrors the functionality of the `require` requisite.

If the "result" of the watched state is `True` *and* the "changes" key contains a populated dictionary (changes occurred in the watched state), then the `watch` requisite can add additional behavior. This additional behavior is defined by the `mod_watch` function within the watching state module. If the `mod_watch` function exists in the watching state module, it will be called *in addition to* the normal watching state. The return data from the `mod_watch` function is what will be returned to the master in this case; the return data from the main watching function is discarded.

If the "changes" key contains an empty dictionary, the `watch` requisite acts exactly like the `require` requisite (the watching state will execute if "result" is `True`, and fail if "result" is `False` in the watched state).

Note: Not all state modules contain `mod_watch`. If `mod_watch` is absent from the watching state module, the `watch` requisite behaves exactly like a `require` requisite.

A good example of using `watch` is with a *`service.running`* state. When a service watches a state, then the service is reloaded/restarted when the watched state changes, in addition to Salt ensuring that the service is running.

```
ntpd:
  service.running:
    - watch:
      - file: /etc/ntp.conf
  file.managed:
    - name: /etc/ntp.conf
    - source: salt://ntp/files/ntp.conf
```

watch_any

New in version 2018.3.0.

The state containing the `watch_any` requisite is defined as the watching state. The states specified in the `watch_any` statement are defined as the watched states. When the watched states execute, they will return a dictionary containing a key named "changes".

If the "result" of any of the watched states is `True`, the watching state *will execute normally*, and if all of them are `False`, the watching state will never run. This part of `watch` mirrors the functionality of the `require` requisite.

If the "result" of any of the watched states is `True` *and* the "changes" key contains a populated dictionary (changes occurred in the watched state), then the `watch` requisite can add additional behavior. This additional behavior is defined by the `mod_watch` function within the watching state module. If the `mod_watch` function exists in the watching state module, it will be called *in addition to* the normal watching state. The return data from the `mod_watch` function is what will be returned to the master in this case; the return data from the main watching function is discarded.

If the "changes" key contains an empty dictionary, the `watch` requisite acts exactly like the `require` requisite (the watching state will execute if "result" is `True`, and fail if "result" is `False` in the watched state).

```
apache2:
  service.running:
    - watch_any:
      - file: /etc/apache2/sites-available/site1.conf
      - file: apache2-site2
  file.managed:
    - name: /etc/apache2/sites-available/site1.conf
    - source: salt://apache2/files/site1.conf
apache2-site2:
  file.managed:
    - name: /etc/apache2/sites-available/site2.conf
    - source: salt://apache2/files/site2.conf
```

In this example, the service will be reloaded/restarted if either of the file.managed states has a result of True and has changes.

prereq

New in version 0.16.0.

`prereq` allows for actions to be taken based on the expected results of a state that has not yet been executed. The state containing the `prereq` requisite is defined as the pre-requiring state. The state specified in the `prereq` statement is defined as the pre-required state.

When a `prereq` requisite is evaluated, the pre-required state reports if it expects to have any changes. It does this by running the pre-required single state as a test-run by enabling `test=True`. This test-run will return a dictionary containing a key named "changes". (See the `watch` section above for examples of "changes" dictionaries.)

If the "changes" key contains a populated dictionary, it means that the pre-required state expects changes to occur when the state is actually executed, as opposed to the test-run. The pre-requiring state will now actually run. If the pre-requiring state executes successfully, the pre-required state will then execute. If the pre-requiring state fails, the pre-required state will not execute.

If the "changes" key contains an empty dictionary, this means that changes are not expected by the pre-required state. Neither the pre-required state nor the pre-requiring state will run.

The best way to define how `prereq` operates is displayed in the following practical example: When a service should be shut down because underlying code is going to change, the service should be off-line while the update occurs. In this example, `graceful-down` is the pre-requiring state and `site-code` is the pre-required state.

```
graceful-down:
  cmd.run:
    - name: service apache graceful
    - prereq:
      - file: site-code

site-code:
  file.recurse:
    - name: /opt/site_code
    - source: salt://site/code
```

In this case the apache server will only be shutdown if the site-code state expects to deploy fresh code via the file.recurse call. The site-code deployment will only be executed if the graceful-down run completes successfully.

onfail

New in version 2014.7.0.

The `onfail` requisite allows for reactions to happen strictly as a response to the failure of another state. This can be used in a number of ways, such as executing a second attempt to set up a service or begin to execute a separate thread of states because of a failure.

The `onfail` requisite is applied in the same way as `require` as `watch`:

```yaml
primary_mount:
  mount.mounted:
    - name: /mnt/share
    - device: 10.0.0.45:/share
    - fstype: nfs

backup_mount:
  mount.mounted:
    - name: /mnt/share
    - device: 192.168.40.34:/share
    - fstype: nfs
    - onfail:
      - mount: primary_mount
```

Note: Setting failhard (*globally* or in *the failing state*) to `True` will cause `onfail`, `onfail_in` and `onfail_any` requisites to be ignored. If you want to combine a global failhard set to True with `onfail`, `onfail_in` or `onfail_any`, you will have to explicitly set failhard to `False` (overriding the global setting) in the state that could fail.

Note: Beginning in the `2016.11.0` release of Salt, `onfail` uses OR logic for multiple listed `onfail` requisites. Prior to the `2016.11.0` release, `onfail` used AND logic. See Issue #22370 for more information.

onfail_any

New in version 2018.3.0.

The `onfail_any` requisite allows for reactions to happen strictly as a response to the failure of at least one other state. This can be used in a number of ways, such as executing a second attempt to set up a service or begin to execute a separate thread of states because of a failure.

The `onfail_any` requisite is applied in the same way as `require_any` and `watch_any`:

```yaml
primary_mount:
  mount.mounted:
    - name: /mnt/share
    - device: 10.0.0.45:/share
    - fstype: nfs

secondary_mount:
```

(continues on next page)

(continued from previous page)

```
  mount.mounted:
    - name: /mnt/code
    - device: 10.0.0.45:/code
    - fstype: nfs

backup_mount:
  mount.mounted:
    - name: /mnt/share
    - device: 192.168.40.34:/share
    - fstype: nfs
    - onfail_any:
      - mount: primary_mount
      - mount: secondary_mount
```

In this example, the *backup_mount* will be mounted if either of the *primary_mount* or *secondary_mount* states results in a failure.

onchanges

New in version 2014.7.0.

The `onchanges` requisite makes a state only apply if the required states generate changes, and if the watched state's "result" is `True`. This can be a useful way to execute a post hook after changing aspects of a system.

If a state has multiple `onchanges` requisites then the state will trigger if any of the watched states changes.

Note: One easy-to-make mistake is to use `onchanges_in` when `onchanges` is supposed to be used. For example, the below configuration is not correct:

```
myservice:
  pkg.installed:
    - name: myservice
  file.managed:
    - name: /etc/myservice/myservice.conf
    - source: salt://myservice/files/myservice.conf
    - mode: 600
  cmd.run:
    - name: /usr/libexec/myservice/post-changes-hook.sh
    - onchanges_in:
      - file: /etc/myservice/myservice.conf
```

This will set up a requisite relationship in which the `cmd.run` state always executes, and the `file.managed` state only executes if the `cmd.run` state has changes (which it always will, since the `cmd.run` state includes the command results as changes).

It may semantically seem like the `cmd.run` state should only run when there are changes in the file state, but remember that requisite relationships involve one state watching another state, and a *requisite_in* does the opposite: it forces the specified state to watch the state with the `requisite_in`.

The correct usage would be:

```
myservice:
  pkg.installed:
    - name: myservice
  file.managed:
```

(continues on next page)

(continued from previous page)

```
    - name: /etc/myservice/myservice.conf
    - source: salt://myservice/files/myservice.conf
    - mode: 600
  cmd.run:
    - name: /usr/libexec/myservice/post-changes-hook.sh
    - onchanges:
      - file: /etc/myservice/myservice.conf
```

---

onchanges_any

New in version 2018.3.0.

The `onchanges_any` requisite makes a state only apply one of the required states generates changes, and if one of the watched state's "result" is `True`. This can be a useful way to execute a post hook after changing aspects of a system.

```
myservice:
  pkg.installed:
    - name: myservice
    - name: yourservice
  file.managed:
    - name: /etc/myservice/myservice.conf
    - source: salt://myservice/files/myservice.conf
    - mode: 600
  file.managed:
    - name: /etc/yourservice/yourservice.conf
    - source: salt://yourservice/files/yourservice.conf
    - mode: 600
  cmd.run:
    - name: /usr/libexec/myservice/post-changes-hook.sh
    - onchanges_any:
      - file: /etc/myservice/myservice.conf
      - file: /etc/your_service/yourservice.conf
```

In this example, the *cmd.run* would be run only if either of the *file.managed* states generated changes and at least one of the watched state's "result" is `True`.

use

The `use` requisite is used to inherit the arguments passed in another id declaration. This is useful when many files need to have the same defaults.

```
/etc/foo.conf:
  file.managed:
    - source: salt://foo.conf
    - template: jinja
    - mkdirs: True
    - user: apache
    - group: apache
    - mode: 755

/etc/bar.conf:
```

(continues on next page)

(continued from previous page)

```
  file.managed:
    - source: salt://bar.conf
    - use:
      - file: /etc/foo.conf
```

The `use` statement was developed primarily for the networking states but can be used on any states in Salt. This makes sense for the networking state because it can define a long list of options that need to be applied to multiple network interfaces.

The `use` statement does not inherit the requisites arguments of the targeted state. This means also a chain of `use` requisites would not inherit inherited options.

runas

New in version 2017.7.0.

The `runas` global option is used to set the user which will be used to run the command in the `cmd.run` module.

```
django:
  pip.installed:
    - name: django >= 1.6, <= 1.7
    - runas: daniel
    - require:
      - pkg: python-pip
```

In the above state, the pip command run by `cmd.run` will be run by the daniel user.

runas_password

New in version 2017.7.2.

The `runas_password` global option is used to set the password used by the runas global option. This is required by `cmd.run` on Windows when `runas` is specified. It will be set when `runas_password` is defined in the state.

```
run_script:
  cmd.run:
    - name: Powershell -NonInteractive -ExecutionPolicy Bypass -File C:\\Temp\\script.
↪ps1
    - runas: frank
    - runas_password: supersecret
```

In the above state, the Powershell script run by `cmd.run` will be run by the frank user with the password `supersecret`.

The _in versions of requisites

All of the requisites also have corresponding requisite_in versions, which do the reverse of their normal counterparts. The examples below all use `require_in` as the example, but note that all of the `_in` requisites work the same way: They result in a normal requisite in the targeted state, which targets the state which has defines the requisite_in. Thus, a `require_in` causes the target state to `require` the targeting state. Similarly, a `watch_in` causes the target state to `watch` the targeting state. This pattern continues for the rest of the requisites.

If a state declaration needs to be required by another state declaration then `require_in` can accommodate it. Therefore, these two sls files would be the same in the end:

Using `require`

```
httpd:
  pkg.installed: []
  service.running:
    - require:
      - pkg: httpd
```

Using `require_in`

```
httpd:
  pkg.installed:
    - require_in:
      - service: httpd
  service.running: []
```

The `require_in` statement is particularly useful when assigning a require in a separate sls file. For instance it may be common for httpd to require components used to set up PHP or mod_python, but the HTTP state does not need to be aware of the additional components that require it when it is set up:

http.sls

```
httpd:
  pkg.installed: []
  service.running:
    - require:
      - pkg: httpd
```

php.sls

```
include:
  - http

php:
  pkg.installed:
    - require_in:
      - service: httpd
```

mod_python.sls

```
include:
  - http

mod_python:
  pkg.installed:
    - require_in:
      - service: httpd
```

Now the httpd server will only start if both php and mod_python are first verified to be installed. Thus allowing for a requisite to be defined ”after the fact”.

Fire Event Notifications

New in version 2015.8.0.

The *fire_event* option in a state will cause the minion to send an event to the Salt Master upon completion of that individual state.

The following example will cause the minion to send an event to the Salt Master with a tag of *salt/state_result/20150505121517276431/dasalt/nano* and the result of the state will be the data field of the event. Notice that the *name* of the state gets added to the tag.

```
nano_stuff:
  pkg.installed:
    - name: nano
    - fire_event: True
```

In the following example instead of setting *fire_event* to *True*, *fire_event* is set to an arbitrary string, which will cause the event to be sent with this tag: *salt/state_result/20150505121725642845/dasalt/custom/tag/nano/finished*

```
nano_stuff:
  pkg.installed:
    - name: nano
    - fire_event: custom/tag/nano/finished
```

Altering States

The state altering system is used to make sure that states are evaluated exactly as the user expects. It can be used to double check that a state preformed exactly how it was expected to, or to make 100% sure that a state only runs under certain conditions. The use of unless or onlyif options help make states even more stateful. The `check_cmd` option helps ensure that the result of a state is evaluated correctly.

Reload

`reload_modules` is a boolean option that forces salt to reload its modules after a state finishes. `reload_pillar` and `reload_grains` can also be set. See *Reloading Modules*.

```
grains_refresh:
  module.run:
   - name: saltutil.refresh_grains
   - reload_grains: true

grains_read:
  module.run:
   - name: grains.items
```

Unless

New in version 2014.7.0.

The `unless` requisite specifies that a state should only run when any of the specified commands return `False`. The `unless` requisite operates as NAND and is useful in giving more granular control over when a state should execute.

NOTE: Under the hood `unless` calls `cmd.retcode` with `python_shell=True`. This means the commands referenced by `unless` will be parsed by a shell, so beware of side-effects as this shell will be run with the same privileges as the salt-minion. Also be aware that the boolean value is determined by the shell's concept of `True` and `False`, rather than Python's concept of `True` and `False`.

```
vim:
  pkg.installed:
    - unless:
      - rpm -q vim-enhanced
      - ls /usr/bin/vim
```

In the example above, the state will only run if either the vim-enhanced package is not installed (returns `False`) or if /usr/bin/vim does not exist (returns `False`). The state will run if both commands return `False`.

However, the state will not run if both commands return `True`.

Unless checks are resolved for each name to which they are associated.

For example:

```
deploy_app:
  cmd.run:
    - names:
      - first_deploy_cmd
      - second_deploy_cmd
    - unless: some_check
```

In the above case, `some_check` will be run prior to _each_ name – once for `first_deploy_cmd` and a second time for `second_deploy_cmd`.

Changed in version 3000: The `unless` requisite can take a module as a dictionary field in unless. The dictionary must contain an argument `fun` which is the module that is being run, and everything else must be passed in under the args key or will be passed as individual kwargs to the module function.

```
install apache on debian based distros:
  cmd.run:
    - name: make install
    - cwd: /path/to/dir/whatever-2.1.5/
    - unless:
      - fun: file.file_exists
        path: /usr/local/bin/whatever
```

```
set mysql root password:
  debconf.set:
    - name: mysql-server-5.7
    - data:
        'mysql-server/root_password': {'type': 'password', 'value': {{pillar['mysql.
↪pass']}} }
    - unless:
      - fun: pkg.version
        args:
          - mysql-server-5.7
```

Onlyif

New in version 2014.7.0.

The `onlyif` requisite specifies that if each command listed in `onlyif` returns `True`, then the state is run. If any of the specified commands return `False`, the state will not run.

NOTE: Under the hood `onlyif` calls `cmd.retcode` with `python_shell=True`. This means the commands referenced by `onlyif` will be parsed by a shell, so beware of side-effects as this shell will be run with the same privileges as the salt-minion. Also be aware that the boolean value is determined by the shell's concept of `True` and `False`, rather than Python's concept of `True` and `False`.

```
stop-volume:
  module.run:
    - name: glusterfs.stop_volume
    - m_name: work
    - onlyif:
      - gluster volume status work
    - order: 1

remove-volume:
  module.run:
    - name: glusterfs.delete
    - m_name: work
    - onlyif:
      - gluster volume info work
    - watch:
      - cmd: stop-volume
```

The above example ensures that the stop_volume and delete modules only run if the gluster commands return a 0 ret value.

Changed in version 3000: The `onlyif` requisite can take a module as a dictionary field in onlyif. The dictionary must contain an argument `fun` which is the module that is being run, and everything else must be passed in under the args key or will be passed as individual kwargs to the module function.

```
install apache on redhat based distros:
  pkg.latest:
    - name: httpd
    - onlyif:
      - fun: match.grain
        tgt: 'os_family: RedHat'

install apache on debian based distros:
  pkg.latest:
    - name: apache2
    - onlyif:
      - fun: match.grain
        tgt: 'os_family: Debian'
```

```
arbitrary file example:
  file.touch:
    - name: /path/to/file
    - onlyif:
      - fun: file.search
        args:
          - /etc/crontab
          - 'entry1'
```

runas

New in version 2014.7.0.

listen and its counterpart listen_in trigger mod_wait functions for states, when those states succeed and result in changes, similar to how watch its counterpart watch_in. Unlike watch and watch_in, listen, and listen_in will not modify the order of states and can be used to ensure your states are executed in the order they are defined. All listen/listen_in actions will occur at the end of a state run, after all states have completed.

```
restart-apache2:
  service.running:
    - name: apache2
    - listen:
      - file: /etc/apache2/apache2.conf

configure-apache2:
  file.managed:
    - name: /etc/apache2/apache2.conf
    - source: salt://apache2/apache2.conf
```

This example will cause apache2 to be restarted when the apache2.conf file is changed, but the apache2 restart will happen at the end of the state run.

```
restart-apache2:
  service.running:
    - name: apache2

configure-apache2:
  file.managed:
    - name: /etc/apache2/apache2.conf
    - source: salt://apache2/apache2.conf
    - listen_in:
      - service: apache2
```

This example does the same as the above example, but puts the state argument on the file resource, rather than the service resource.

check_cmd

New in version 2014.7.0.

Check Command is used for determining that a state did or did not run as expected.

NOTE: Under the hood `check_cmd` calls `cmd.retcode` with `python_shell=True`. This means the commands referenced by unless will be parsed by a shell, so beware of side-effects as this shell will be run with the same privileges as the salt-minion.

```
comment-repo:
  file.replace:
    - name: /etc/yum.repos.d/fedora.repo
    - pattern: '^enabled=0'
    - repl: enabled=1
    - check_cmd:
      - "! grep 'enabled=0' /etc/yum.repos.d/fedora.repo"
```

This will attempt to do a replace on all `enabled=0` in the .repo file, and replace them with `enabled=1`. The `check_cmd` is just a bash command. It will do a grep for `enabled=0` in the file, and if it finds any, it will return a 0, which will be inverted by the leading `!`, causing `check_cmd` to set the state as failed. If it returns a 1, meaning it didn't find any `enabled=0`, it will be inverted by the leading `!`, returning a 0, and declaring the function succeeded.

NOTE: This requisite `check_cmd` functions differently than the `check_cmd` of the `file.managed` state.

Overriding Checks

There are two commands used for the above checks.

`mod_run_check` is used to check for `onlyif` and `unless`. If the goal is to override the global check for these to variables, include a `mod_run_check` in the salt/states/ file.

`mod_run_check_cmd` is used to check for the check_cmd options. To override this one, include a `mod_run_check_cmd` in the states file for the state.

Retrying States

New in version 2017.7.0.

The retry option in a state allows it to be executed multiple times until a desired result is obtained or the maximum number of attempts have been made.

The retry option can be configured by the `attempts`, `until`, `interval`, and `splay` parameters.

The `attempts` parameter controls the maximum number of times the state will be run. If not specified or if an invalid value is specified, `attempts` will default to `2`.

The `until` parameter defines the result that is required to stop retrying the state. If not specified or if an invalid value is specified, `until` will default to `True`

The `interval` parameter defines the amount of time, in seconds, that the system will wait between attempts. If not specified or if an invalid value is specified, `interval` will default to `30`.

The `splay` parameter allows the `interval` to be additionally spread out. If not specified or if an invalid value is specified, `splay` defaults to `0` (i.e. no splaying will occur).

The following example will run the pkg.installed state until it returns `True` or it has been run `5` times. Each attempt will be `60` seconds apart and the interval will be splayed up to an additional `10` seconds:

```
my_retried_state:
  pkg.installed:
    - name: nano
    - retry:
        attempts: 5
        until: True
        interval: 60
        splay: 10
```

The following example will run the pkg.installed state with all the defaults for `retry`. The state will run up to `2` times, each attempt being `30` seconds apart, or until it returns `True`.

```
install_nano:
  pkg.installed:
    - name: nano
    - retry: True
```

The following example will run the file.exists state every `30` seconds up to `15` times or until the file exists (i.e. the state returns `True`).

```
wait_for_file:
  file.exists:
    - name: /path/to/file
    - retry:
```

(continues on next page)

(continued from previous page)

```
        attempts: 15
        interval: 30
```

Return data from a retried state

When a state is retried, the returned output is as follows:

The `result` return value is the `result` from the final run. For example, imagine a state set to `retry` up to three times or `until True`. If the state returns `False` on the first run and then `True` on the second, the `result` of the state will be `True`.

The `started` return value is the `started` from the first run.

The `duration` return value is the total duration of all attempts plus the retry intervals.

The `comment` return value will include the result and comment from all previous attempts.

For example:

```yaml
wait_for_file:
  file.exists:
    - name: /path/to/file
    - retry:
        attempts: 10
        interval: 2
        splay: 5
```

Would return similar to the following. The state result in this case is `False` (file.exist was run 10 times with a 2 second interval, but the file specified did not exist on any run).

```
      ID: wait_for_file
Function: file.exists
  Result: False
 Comment: Attempt 1: Returned a result of "False", with the following comment:
↪"Specified path /path/to/file does not exist"
          Attempt 2: Returned a result of "False", with the following comment:
↪"Specified path /path/to/file does not exist"
          Attempt 3: Returned a result of "False", with the following comment:
↪"Specified path /path/to/file does not exist"
          Attempt 4: Returned a result of "False", with the following comment:
↪"Specified path /path/to/file does not exist"
          Attempt 5: Returned a result of "False", with the following comment:
↪"Specified path /path/to/file does not exist"
          Attempt 6: Returned a result of "False", with the following comment:
↪"Specified path /path/to/file does not exist"
          Attempt 7: Returned a result of "False", with the following comment:
↪"Specified path /path/to/file does not exist"
          Attempt 8: Returned a result of "False", with the following comment:
↪"Specified path /path/to/file does not exist"
          Attempt 9: Returned a result of "False", with the following comment:
↪"Specified path /path/to/file does not exist"
          Specified path /path/to/file does not exist
 Started: 09:08:12.903000
Duration: 47000.0 ms
 Changes:
```

6.1.16 Startup States

Sometimes it may be desired that the salt minion execute a state run when it is started. This alleviates the need for the master to initiate a state run on a new minion and can make provisioning much easier.

As of Salt 0.10.3 the minion config reads options that allow for states to be executed at startup. The options are *startup_states*, *sls_list*, and *top_file*.

The *startup_states* option can be passed one of a number of arguments to define how to execute states. The available options are:

highstate Execute *`state.apply`*

sls Read in the `sls_list` option and execute the named sls files

top Read in the `top_file` option and execute states based on that top file on the Salt Master

Examples:

Execute *`state.apply`* to run the *highstate* when starting the minion:

```
startup_states: highstate
```

Execute the sls files *edit.vim* and *hyper*:

```
startup_states: sls

sls_list:
  - edit.vim
  - hyper
```

6.1.17 State Testing

Executing a Salt state run can potentially change many aspects of a system and it may be desirable to first see what a state run is going to change before applying the run.

Salt has a test interface to report on exactly what will be changed, this interface can be invoked on any of the major state run functions:

```
salt '*' state.apply test=True
salt '*' state.apply mysls test=True
salt '*' state.single test=True
```

The test run is mandated by adding the `test=True` option to the states. The return information will show states that will be applied in yellow and the result is reported as `None`.

Default Test

If the value `test` is set to `True` in the minion configuration file then states will default to being executed in test mode. If this value is set then states can still be run by calling test=False:

```
salt '*' state.apply test=False
salt '*' state.apply mysls test=False
salt '*' state.single test=False
```

6.1.18 The Top File

Introduction

Most infrastructures are made up of groups of machines, each machine in the group performing a role similar to others. Those groups of machines work in concert with each other to create an application stack.

To effectively manage those groups of machines, an administrator needs to be able to create roles for those groups. For example, a group of machines that serve front-end web traffic might have roles which indicate that those machines should all have the Apache webserver package installed and that the Apache service should always be running.

In Salt, the file which contains a mapping between groups of machines on a network and the configuration roles that should be applied to them is called a `top file`.

Top files are named `top.sls` by default and they are so-named because they always exist in the "top" of a directory hierarchy that contains state files. That directory hierarchy is called a `state tree`.

A Basic Example

Top files have three components:

- **Environment**: A state tree directory containing a set of state files to configure systems.
- **Target**: A grouping of machines which will have a set of states applied to them.
- **State files**: A list of state files to apply to a target. Each state file describes one or more states to be configured and enforced on the targeted machines.

The relationship between these three components is nested as follows:

- Environments contain targets
- Targets contain states

Putting these concepts together, we can describe a scenario in which all minions with an ID that begins with `web` have an `apache` state applied to them:

```
base:          # Apply SLS files from the directory root for the 'base' environment
  'web*':      # All minions with a minion_id that begins with 'web'
    - apache   # Apply the state file named 'apache.sls'
```

Environments

Environments are directory hierarchies which contain a top file and a set of state files.

Environments can be used in many ways, however there is no requirement that they be used at all. In fact, the most common way to deploy Salt is with a single environment, called `base`. It is recommended that users only create multiple environments if they have a use case which specifically calls for multiple versions of state trees.

Getting Started with Top Files

Each environment is defined inside a salt master configuration variable called, *`file_roots`* .

In the most common single-environment setup, only the `base` environment is defined in *`file_roots`* along with only one directory path for the state tree.

```
file_roots:
  base:
    - /srv/salt
```

In the above example, the top file will only have a single environment to pull from.

Next is a simple single-environment top file placed in `/srv/salt/top.sls`, illustrating that for the environment called `base`, all minions will have the state files named `core.sls` and `edit.sls` applied to them.

```
base:
  '*':
    - core
    - edit
```

Assuming the `file_roots` configuration from above, Salt will look in the `/srv/salt` directory for `core.sls` and `edit.sls`.

Multiple Environments

In some cases, teams may wish to create versioned state trees which can be used to test Salt configurations in isolated sets of systems such as a staging environment before deploying states into production.

For this case, multiple environments can be used to accomplish this task.

To create multiple environments, the *`file_roots`* option can be expanded:

```
file_roots:
  dev:
    - /srv/salt/dev
  qa:
    - /srv/salt/qa
  prod:
    - /srv/salt/prod
```

In the above, we declare three environments: `dev`, `qa` and `prod`. Each environment has a single directory assigned to it.

Our top file references the environments:

```
dev:
  'webserver*':
    - webserver
  'db*':
    - db
qa:
  'webserver*':
    - webserver
  'db*':
    - db
prod:
  'webserver*':
    - webserver
  'db*':
    - db
```

As seen above, the top file now declares the three environments and for each, target expressions are defined to map minions to state files. For example, all minions which have an ID beginning with the string `webserver` will have the webserver state from the requested environment assigned to it.

In this manner, a proposed change to a state could first be made in a state file in `/srv/salt/dev` and then be applied to development webservers before moving the state into QA by copying the state file into `/srv/salt/qa`.

Choosing an Environment to Target

The top file is used to assign a minion to an environment unless overridden using the methods described below. The environment in the top file must match valid fileserver environment (a.k.a. `saltenv`) in order for any states to be applied to that minion. When using the default fileserver backend, environments are defined in *`file_roots`*.

The states that will be applied to a minion in a given environment can be viewed using the *`state.show_top`* function.

Minions may be pinned to a particular environment by setting the *`environment`* value in the minion configuration file. In doing so, a minion will only request files from the environment to which it is assigned.

The environment may also be dynamically selected at runtime by passing it to the `salt`, `salt-call` or `salt-ssh` command. This is most commonly done with functions in the `state` module by using the `saltenv` argument. For example, to run a `highstate` on all minions, using only the top file and SLS files in the `prod` environment, run: `salt '*' state.highstate saltenv=prod`.

Note: Not all functions accept `saltenv` as an argument, see the documentation for an individual function documentation to verify.

Shorthand

If you assign only one SLS to a system, as in this example, a shorthand is also available:

```
base:
  '*': global
dev:
  'webserver*': webserver
  'db*':        db
qa:
  'webserver*': webserver
  'db*':        db
prod:
  'webserver*': webserver
  'db*':        db
```

Advanced Minion Targeting

In the examples above, notice that all of the target expressions are globs. The default match type in top files (since version 2014.7.0) is actually the *compound matcher*, not the glob matcher as in the CLI.

A single glob, when passed through the compound matcher, acts the same way as matching by glob, so in most cases the two are indistinguishable. However, there is an edge case in which a minion ID contains whitespace. While it is not recommended to include spaces in a minion ID, Salt will not stop you from doing so. However, since compound expressions are parsed word-by-word, if a minion ID contains spaces it will fail to match. In this edge case, it will be necessary to explicitly use the `glob` matcher:

```
base:
  'minion 1':
```

(continues on next page)

(continued from previous page)

```
    - match: glob
    - foo
```

The available match types which can be set for a target expression in the top file are:

Match Type	Description
glob	Full minion ID or glob expression to match multiple minions (e.g. `minion123` or `minion*`)
pcre	Perl-compatible regular expression (PCRE) matching a minion ID (e.g. `web[0-3].domain.com`)
grain	Match a *grain*, optionally using globbing (e.g. `kernel:Linux` or `kernel:*BSD`)
grain_pcre	Match a *grain* using PCRE (e.g. `kernel:(Free|Open)BSD`)
list	Comma-separated list of minions (e.g. `minion1,minion2,minion3`)
pillar	*Pillar* match, optionally using globbing (e.g. `role:webserver` or `role:web*`)
pillar_pcre	*Pillar* match using PCRE (e.g. `role:web(server|proxy)`
pillar_exact	*Pillar* match with no globbing or PCRE (e.g. `role:webserver`)
ipcidr	Subnet or IP address (e.g. `172.17.0.0/16` or `10.2.9.80`)
data	Match values kept in the minion's datastore (created using the *`data`* execution module)
range	*Range* cluster
compound	Complex expression combining multiple match types (see *here*)
nodegroup	Pre-defined compound expressions in the master config file (see *here*)

Below is a slightly more complex top file example, showing some of the above match types:

```
# All files will be taken from the file path specified in the base
# environment in the ``file_roots`` configuration value.

base:
    # All minions which begin with the strings 'nag1' or any minion with
    # a grain set called 'role' with the value of 'monitoring' will have
    # the 'server.sls' state file applied from the 'nagios/' directory.

    'nag1* or G@role:monitoring':
        - nagios.server

    # All minions get the following three state files applied

    '*':
        - ldap-client
        - networking
        - salt.minion

    # All minions which have an ID that begins with the phrase
    # 'salt-master' will have an SLS file applied that is named
    # 'master.sls' and is in the 'salt' directory, underneath
    # the root specified in the ``base`` environment in the
    # configuration value for ``file_roots``.

    'salt-master*':
        - salt.master

    # Minions that have an ID matching the following regular
    # expression will have the state file called 'web.sls' in the
    # nagios/mon directory applied. Additionally, minions matching
    # the regular expression will also have the 'server.sls' file
```

(continues on next page)

(continued from previous page)

```
    # in the apache/ directory applied.

    # NOTE!
    #
    # Take note of the 'match' directive here, which tells Salt
    # to treat the target string as a regex to be matched!

    '^(memcache|web).(qa|prod).loc$':
        - match: pcre
        - nagios.mon.web
        - apache.server

    # Minions that have a grain set indicating that they are running
    # the Ubuntu operating system will have the state file called
    # 'ubuntu.sls' in the 'repos' directory applied.
    #
    # Again take note of the 'match' directive here which tells
    # Salt to match against a grain instead of a minion ID.

    'os:Ubuntu':
        - match: grain
        - repos.ubuntu

    # Minions that are either RedHat or CentOS should have the 'epel.sls'
    # state applied, from the 'repos/' directory.

    'os:(RedHat|CentOS)':
        - match: grain_pcre
        - repos.epel

    # The three minions with the IDs of 'foo', 'bar' and 'baz' should
    # have 'database.sls' applied.

    'foo,bar,baz':
        - match: list
        - database

    # Any minion for which the pillar key 'somekey' is set and has a value
    # of that key matching 'abc' will have the 'xyz.sls' state applied.

    'somekey:abc':
        - match: pillar
        - xyz
```

How Top Files Are Compiled

When a *highstate* is executed and an environment is specified (either using the `environment` config option or by passing the saltenv when executing the *highstate*), then that environment's top file is the only top file used to assign states to minions, and only states from the specified environment will be run.

The remainder of this section applies to cases in which a *highstate* is executed without an environment specified.

With no environment specified, the minion will look for a top file in each environment, and each top file will be processed to determine the SLS files to run on the minions. By default, the top files from each environment will be merged together. In configurations with many environments, such as with *GitFS* where each branch and tag is treated as a distinct environment, this may cause unexpected results as SLS files from older tags cause defunct SLS files to be included in the highstate. In cases like this, it can be helpful to set *top_file_merging_strategy* to `same` to force each environment to use its own top file.

```
top_file_merging_strategy: same
```

Another option would be to set *state_top_saltenv* to a specific environment, to ensure that any top files in other environments are disregarded:

```
state_top_saltenv: base
```

With *GitFS*, it can also be helpful to simply manage each environment's top file separately, and/or manually specify the environment when executing the highstate to avoid any complicated merging scenarios. *gitfs_saltenv_whitelist* and *gitfs_saltenv_blacklist* can also be used to hide unneeded branches and tags from GitFS to reduce the number of top files in play.

When using multiple environments, it is not necessary to create a top file for each environment. The easiest-to-maintain approach is to use a single top file placed in the `base` environment. This is often infeasible with *GitFS* though, since branching/tagging can easily result in extra top files. However, when only the default (`roots`) fileserver backend is used, a single top file in the `base` environment is the most common way of configuring a *highstate*.

The following minion configuration options affect how top files are compiled when no environment is specified, it is recommended to follow the below four links to learn more about how these options work:

- *state_top_saltenv*
- *top_file_merging_strategy*
- *env_order*
- *default_top*

Top File Compilation Examples

For the scenarios below, assume the following configuration:

/etc/salt/master:

```
file_roots:
  base:
    - /srv/salt/base
  dev:
    - /srv/salt/dev
  qa:
    - /srv/salt/qa
```

/srv/salt/base/top.sls:

```
base:
  '*':
    - base1
dev:
  '*':
    - dev1
qa:
  '*':
    - qa1
```

/srv/salt/dev/top.sls:

```
base:
  'minion1':
    - base2
dev:
  'minion2':
    - dev2
qa:
  '*':
    - qa1
    - qa2
```

Note: For the purposes of these examples, there is no top file in the `qa` environment.

Scenario 1 - `dev` Environment Specified

In this scenario, the *highstate* was either invoked with `saltenv=dev` or the minion has `environment: dev` set in the minion config file. The result will be that only the `dev2` SLS from the dev environment will be part of the *highstate*, and it will be applied to minion2, while minion1 will have no states applied to it.

If the `base` environment were specified, the result would be that only the `base1` SLS from the `base` environment would be part of the *highstate*, and it would be applied to all minions.

If the `qa` environment were specified, the *highstate* would exit with an error.

Scenario 2 - No Environment Specified, `top_file_merging_strategy` is "merge"

In this scenario, assuming that the `base` environment's top file was evaluated first, the `base1`, `dev1`, and `qa1` states would be applied to all minions. If, for instance, the `qa` environment is not defined in **/srv/salt/base/top.sls**, then because there is no top file for the `qa` environment, no states from the `qa` environment would be applied.

Scenario 3 - No Environment Specified, `top_file_merging_strategy` is "same"

Changed in version 2016.11.0: In prior versions, "same" did not quite work as described below (see here). This has now been corrected. It was decided that changing something like top file handling in a point release had the potential to unexpectedly impact users' top files too much, and it would be better to make this correction in a feature release.

In this scenario, `base1` from the `base` environment is applied to all minions. Additionally, `dev2` from the `dev` environment is applied to minion2.

If *`default_top`* is unset (or set to `base`, which happens to be the default), then `qa1` from the `qa` environment will be applied to all minions. If *`default_top`* were set to `dev`, then both `qa1` and `qa2` from the `qa` environment would be applied to all minions.

Scenario 4 - No Environment Specified, `top_file_merging_strategy` is "merge_all"

New in version 2016.11.0.

In this scenario, all configured states in all top files are applied. From the `base` environment, `base1` would be applied to all minions, with `base2` being applied only to `minion1`. From the `dev` environment, `dev1` would be applied to all minions, with `dev2` being applied only to `minion2`. Finally, from the `qa` environment, both the `qa1` and `qa2` states will be applied to all minions. Note that the `qa1` states would not be applied twice, even though `qa1` appears twice.

6.1.19 SLS Template Variable Reference

The template engines available to sls files and file templates come loaded with a number of context variables. These variables contain information and functions to assist in the generation of templates. See each variable below for its availability – not all variables are available in all templating contexts.

Salt

The *salt* variable is available to abstract the salt library functions. This variable is a python dictionary containing all of the functions available to the running salt minion. It is available in all salt templates.

```
{% for file in salt['cmd.run']('ls -1 /opt/to_remove').splitlines() %}
/opt/to_remove/{{ file }}:
  file.absent
{% endfor %}
```

Opts

The *opts* variable abstracts the contents of the minion's configuration file directly to the template. The *opts* variable is a dictionary. It is available in all templates.

```
{{ opts['cachedir'] }}
```

The `config.get` function also searches for values in the *opts* dictionary.

Pillar

The *pillar* dictionary can be referenced directly, and is available in all templates:

```
{{ pillar['key'] }}
```

Using the `pillar.get` function via the *salt* variable is generally recommended since a default can be safely set in the event that the value is not available in pillar and dictionaries can be traversed directly:

```
{{ salt['pillar.get']('key', 'failover_value') }}
{{ salt['pillar.get']('stuff:more:deeper') }}
```

Grains

The *grains* dictionary makes the minion's grains directly available, and is available in all templates:

```
{{ grains['os'] }}
```

The `grains.get` function can be used to traverse deeper grains and set defaults:

```
{{ salt['grains.get']('os') }}
```

saltenv

The *saltenv* variable is available in only in sls files when gathering the sls from an environment.

```
{{ saltenv }}
```

sls

The *sls* variable contains the sls reference value, and is only available in the actual SLS file (not in any files referenced in that SLS). The sls reference value is the value used to include the sls in top files or via the include option.

```
{{ sls }}
```

slspath

The *slspath* variable contains the path to the directory of the current sls file. The value of *slspath* in files referenced in the current sls depends on the reference method. For jinja includes *slspath* is the path to the current directory of the file. For salt includes *slspath* is the path to the directory of the included file.

```
{{ slspath }}
```

6.1.20 State Modules

State Modules are the components that map to actual enforcement and management of Salt states.

States are Easy to Write!

State Modules should be easy to write and straightforward. The information passed to the SLS data structures will map directly to the states modules.

Mapping the information from the SLS data is simple, this example should illustrate:

```
/etc/salt/master: # maps to "name", unless a "name" argument is specified below
  file.managed: # maps to <filename>.<function> - e.g. "managed" in https://github.
↪com/saltstack/salt/tree/master/salt/states/file.py
    - user: root # one of many options passed to the manage function
    - group: root
    - mode: 644
    - source: salt://salt/master
```

Therefore this SLS data can be directly linked to a module, function, and arguments passed to that function.

This does issue the burden, that function names, state names and function arguments should be very human readable inside state modules, since they directly define the user interface.

Keyword Arguments

Salt passes a number of keyword arguments to states when rendering them, including the environment, a unique identifier for the state, and more. Additionally, keep in mind that the requisites for a state are part of the keyword arguments. Therefore, if you need to iterate through the keyword arguments in a state, these must be considered and handled appropriately. One such example is in the `pkgrepo.managed` state, which needs to be able to handle arbitrary keyword arguments and pass them to module execution functions. An example of how these keyword arguments can be handled can be found here.

Best Practices

A well-written state function will follow these steps:

Note: This is an extremely simplified example. Feel free to browse the source code for Salt's state modules to see other examples.

1. Set up the return dictionary and perform any necessary input validation (type checking, looking for use of mutually-exclusive arguments, etc.).

   ```
   ret = {'name': name,
          'result': False,
          'changes': {},
          'comment': ''}

   if foo and bar:
       ret['comment'] = 'Only one of foo and bar is permitted'
       return ret
   ```

2. Check if changes need to be made. This is best done with an information-gathering function in an accompanying *execution module*. The state should be able to use the return from this function to tell whether or not the minion is already in the desired state.

   ```
   result = __salt__['modname.check'](name)
   ```

3. If step 2 found that the minion is already in the desired state, then exit immediately with a `True` result and without making any changes.

   ```
   if result:
       ret['result'] = True
       ret['comment'] = '{0} is already installed'.format(name)
       return ret
   ```

4. If step 2 found that changes *do* need to be made, then check to see if the state was being run in test mode (i.e. with `test=True`). If so, then exit with a `None` result, a relevant comment, and (if possible) a `changes` entry describing what changes would be made.

   ```
   if __opts__['test']:
       ret['result'] = None
       ret['comment'] = '{0} would be installed'.format(name)
       ret['changes'] = result
       return ret
   ```

5. Make the desired changes. This should again be done using a function from an accompanying execution module. If the result of that function is enough to tell you whether or not an error occurred, then you can exit with a `False` result and a relevant comment to explain what happened.

   ```
   result = __salt__['modname.install'](name)
   ```

6. Perform the same check from step 2 again to confirm whether or not the minion is in the desired state. Just as in step 2, this function should be able to tell you by its return data whether or not changes need to be made.

```
ret['changes'] = __salt__['modname.check'](name)
```

As you can see here, we are setting the `changes` key in the return dictionary to the result of the `modname.check` function (just as we did in step 4). The assumption here is that the information-gathering function will return a dictionary explaining what changes need to be made. This may or may not fit your use case.

7. Set the return data and return!

```
if ret['changes']:
    ret['comment'] = '{0} failed to install'.format(name)
else:
    ret['result'] = True
    ret['comment'] = '{0} was installed'.format(name)

return ret
```

Using Custom State Modules

Before the state module can be used, it must be distributed to minions. This can be done by placing them into `salt://_states/`. They can then be distributed manually to minions by running *`saltutil.sync_states`* or *`saltutil.sync_all`*. Alternatively, when running a *highstate* custom types will automatically be synced.

NOTE: Writing state modules with hyphens in the filename will cause issues with !pyobjects routines. Best practice to stick to underscores.

Any custom states which have been synced to a minion, that are named the same as one of Salt's default set of states, will take the place of the default state with the same name. Note that a state module's name defaults to one based on its filename (i.e. `foo.py` becomes state module `foo`), but that its name can be overridden by using a *__virtual__ function*.

Cross Calling Execution Modules from States

As with Execution Modules, State Modules can also make use of the `__salt__` and `__grains__` data. See *cross calling execution modules*.

It is important to note that the real work of state management should not be done in the state module unless it is needed. A good example is the pkg state module. This module does not do any package management work, it just calls the pkg execution module. This makes the pkg state module completely generic, which is why there is only one pkg state module and many backend pkg execution modules.

On the other hand some modules will require that the logic be placed in the state module, a good example of this is the file module. But in the vast majority of cases this is not the best approach, and writing specific execution modules to do the backend work will be the optimal solution.

Cross Calling State Modules

All of the Salt state modules are available to each other and state modules can call functions available in other state modules.

The variable `__states__` is packed into the modules after they are loaded into the Salt minion.

The `__states__` variable is a Python dictionary containing all of the state modules. Dictionary keys are strings representing the names of the modules and the values are the functions themselves.

Salt state modules can be cross-called by accessing the value in the `__states__` dict:

```
ret = __states__['file.managed'](name='/tmp/myfile', source='salt://myfile')
```

This code will call the *managed* function in the `file` state module and pass the arguments `name` and `source` to it.

Return Data

A State Module must return a dict containing the following keys/values:

- **name**: The same value passed to the state as "name".
- **changes**: A dict describing the changes made. Each thing changed should be a key, with its value being another dict with keys called "old" and "new" containing the old/new values. For example, the pkg state's **changes** dict has one key for each package changed, with the "old" and "new" keys in its sub-dict containing the old and new versions of the package. For example, the final changes dictionary for this scenario would look something like this:

  ```
  ret['changes'].update({'my_pkg_name': {'old': '',
                                         'new': 'my_pkg_name-1.0'}})
  ```

- **result**: A tristate value. `True` if the action was successful, `False` if it was not, or `None` if the state was run in test mode, `test=True`, and changes would have been made if the state was not run in test mode.

	live mode	test mode
no changes	`True`	`True`
successful changes	`True`	`None`
failed changes	`False`	`False` or `None`

  Note: Test mode does not predict if the changes will be successful or not, and hence the result for pending changes is usually `None`.

  However, if a state is going to fail and this can be determined in test mode without applying the change, `False` can be returned.

- **comment**: A list of strings or a single string summarizing the result. Note that support for lists of strings is available as of Salt 2018.3.0. Lists of strings will be joined with newlines to form the final comment; this is useful to allow multiple comments from subparts of a state. Prefer to keep line lengths short (use multiple lines as needed), and end with punctuation (e.g. a period) to delimit multiple comments.

Note: States should not return data which cannot be serialized such as frozensets.

Test State

All states should check for and support `test` being passed in the options. This will return data about what changes would occur if the state were actually run. An example of such a check could look like this:

```
# Return comment of changes if test.
if __opts__['test']:
    ret['result'] = None
    ret['comment'] = 'State Foo will execute with param {0}'.format(bar)
    return ret
```

Make sure to test and return before performing any real actions on the minion.

Note: Be sure to refer to the `result` table listed above and displaying any possible changes when writing support for `test`. Looking for changes in a state is essential to `test=true` functionality. If a state is predicted to have no changes when `test=true` (or `test: true` in a config file) is used, then the result of the final state **should not** be `None`.

Watcher Function

If the state being written should support the watch requisite then a watcher function needs to be declared. The watcher function is called whenever the watch requisite is invoked and should be generic to the behavior of the state itself.

The watcher function should accept all of the options that the normal state functions accept (as they will be passed into the watcher function).

A watcher function typically is used to execute state specific reactive behavior, for instance, the watcher for the service module restarts the named service and makes it useful for the watcher to make the service react to changes in the environment.

The watcher function also needs to return the same data that a normal state function returns.

Mod_init Interface

Some states need to execute something only once to ensure that an environment has been set up, or certain conditions global to the state behavior can be predefined. This is the realm of the mod_init interface.

A state module can have a function called **mod_init** which executes when the first state of this type is called. This interface was created primarily to improve the pkg state. When packages are installed the package metadata needs to be refreshed, but refreshing the package metadata every time a package is installed is wasteful. The mod_init function for the pkg state sets a flag down so that the first, and only the first, package installation attempt will refresh the package database (the package database can of course be manually called to refresh via the `refresh` option in the pkg state).

The mod_init function must accept the **Low State Data** for the given executing state as an argument. The low state data is a dict and can be seen by executing the state.show_lowstate function. Then the mod_init function must return a bool. If the return value is True, then the mod_init function will not be executed again, meaning that the needed behavior has been set up. Otherwise, if the mod_init function returns False, then the function will be called the next time.

A good example of the mod_init function is found in the pkg state module:

```python
def mod_init(low):
    '''
    Refresh the package database here so that it only needs to happen once
    '''
    if low['fun'] == 'installed' or low['fun'] == 'latest':
        rtag = __gen_rtag()
        if not os.path.exists(rtag):
            open(rtag, 'w+').write('')
        return True
    else:
        return False
```

The mod_init function in the pkg state accepts the low state data as `low` and then checks to see if the function being called is going to install packages, if the function is not going to install packages then there is no need to refresh the package database. Therefore if the package database is prepared to refresh, then return True and the mod_init will not be called the next time a pkg state is evaluated, otherwise return False and the mod_init will be called next time a pkg state is evaluated.

Log Output

You can call the logger from custom modules to write messages to the minion logs. The following code snippet demonstrates writing log messages:

```
import logging

log = logging.getLogger(__name__)

log.info('Here is Some Information')
log.warning('You Should Not Do That')
log.error('It Is Busted')
```

Strings and Unicode

A state module author should always assume that strings fed to the module have already decoded from strings into Unicode. In Python 2, these will be of type 'Unicode' and in Python 3 they will be of type `str`. Calling from a state to other Salt sub-systems, such as execution modules should pass Unicode (or bytes if passing binary data). In the rare event that a state needs to write directly to disk, Unicode should be encoded to a string immediately before writing to disk. An author may use `__salt_system_encoding__` to learn what the encoding type of the system is. For example, *'my_string'.encode(__salt_system_encoding__').*

Full State Module Example

The following is a simplistic example of a full state module and function. Remember to call out to execution modules to perform all the real work. The state module should only perform "before" and "after" checks.

1. Make a custom state module by putting the code into a file at the following path: **/srv/salt/_states/my_custom_state.py**.
2. Distribute the custom state module to the minions:

```
salt '*' saltutil.sync_states
```

3. Write a new state to use the custom state by making a new state file, for instance **/srv/salt/my_custom_state.sls**.
4. Add the following SLS configuration to the file created in Step 3:

```
human_friendly_state_id:        # An arbitrary state ID declaration.
  my_custom_state:              # The custom state module name.
    - enforce_custom_thing      # The function in the custom state module.
    - name: a_value             # Maps to the ``name`` parameter in the custom
↪function.
    - foo: Foo                  # Specify the required ``foo`` parameter.
    - bar: False                # Override the default value for the ``bar``
↪parameter.
```

Example state module

```python
import salt.exceptions

def enforce_custom_thing(name, foo, bar=True):
    '''
    Enforce the state of a custom thing

    This state module does a custom thing. It calls out to the execution module
    ``my_custom_module`` in order to check the current system and perform any
    needed changes.

    name
        The thing to do something to
    foo
        A required argument
    bar : True
        An argument with a default value
    '''
    ret = {
        'name': name,
        'changes': {},
        'result': False,
        'comment': '',
        }

    # Start with basic error-checking. Do all the passed parameters make sense
    # and agree with each-other?
    if bar == True and foo.startswith('Foo'):
        raise salt.exceptions.SaltInvocationError(
            'Argument "foo" cannot start with "Foo" if argument "bar" is True.')

    # Check the current state of the system. Does anything need to change?
    current_state = __salt__['my_custom_module.current_state'](name)

    if current_state == foo:
        ret['result'] = True
        ret['comment'] = 'System already in the correct state'
        return ret

    # The state of the system does need to be changed. Check if we're running
    # in ``test=true`` mode.
    if __opts__['test'] == True:
        ret['comment'] = 'The state of "{0}" will be changed.'.format(name)
        ret['changes'] = {
            'old': current_state,
            'new': 'Description, diff, whatever of the new state',
        }

        # Return ``None`` when running with ``test=true``.
        ret['result'] = None

        return ret

    # Finally, make the actual change and return the result.
    new_state = __salt__['my_custom_module.change_state'](name, foo)
```

(continues on next page)

(continued from previous page)

```
    ret['comment'] = 'The state of "{0}" was changed!'.format(name)

    ret['changes'] = {
        'old': current_state,
        'new': new_state,
    }

    ret['result'] = True

    return ret
```

6.1.21 State Management

State management, also frequently called Software Configuration Management (SCM), is a program that puts and keeps a system into a predetermined state. It installs software packages, starts or restarts services or puts configuration files in place and watches them for changes.

Having a state management system in place allows one to easily and reliably configure and manage a few servers or a few thousand servers. It allows configurations to be kept under version control.

Salt States is an extension of the Salt Modules that we discussed in the previous *remote execution* tutorial. Instead of calling one-off executions the state of a system can be easily defined and then enforced.

6.1.22 Understanding the Salt State System Components

The Salt state system is comprised of a number of components. As a user, an understanding of the SLS and renderer systems are needed. But as a developer, an understanding of Salt states and how to write the states is needed as well.

Note: States are compiled and executed only on minions that have been targeted. To execute functions directly on masters, see *runners*.

Salt SLS System

The primary system used by the Salt state system is the SLS system. SLS stands for **S**a**L**t **S**tate.

The Salt States are files which contain the information about how to configure Salt minions. The states are laid out in a directory tree and can be written in many different formats.

The contents of the files and the way they are laid out is intended to be as simple as possible while allowing for maximum flexibility. The files are laid out in states and contains information about how the minion needs to be configured.

SLS File Layout

SLS files are laid out in the Salt file server.

A simple layout can look like this:

```
top.sls
ssh.sls
sshd_config
users/init.sls
users/admin.sls
salt/master.sls
web/init.sls
```

The `top.sls` file is a key component. The `top.sls` files is used to determine which SLS files should be applied to which minions.

The rest of the files with the `.sls` extension in the above example are state files.

Files without a `.sls` extensions are seen by the Salt master as files that can be downloaded to a Salt minion.

States are translated into dot notation. For example, the `ssh.sls` file is seen as the ssh state and the `users/admin.sls` file is seen as the users.admin state.

Files named `init.sls` are translated to be the state name of the parent directory, so the `web/init.sls` file translates to the `web` state.

In Salt, everything is a file; there is no "magic translation" of files and file types. This means that a state file can be distributed to minions just like a plain text or binary file.

SLS Files

The Salt state files are simple sets of data. Since SLS files are just data they can be represented in a number of different ways.

The default format is YAML generated from a Jinja template. This allows for the states files to have all the language constructs of Python and the simplicity of YAML.

State files can then be complicated Jinja templates that translate down to YAML, or just plain and simple YAML files.

The State files are simply common data structures such as dictionaries and lists, constructed using a templating language such as YAML.

Here is an example of a Salt State:

```yaml
vim:
  pkg.installed: []

salt:
  pkg.latest:
    - name: salt
  service.running:
    - names:
      - salt-master
      - salt-minion
    - require:
      - pkg: salt
    - watch:
      - file: /etc/salt/minion

/etc/salt/minion:
  file.managed:
    - source: salt://salt/minion
    - user: root
```

(continues on next page)

(continued from previous page)

```
    - group: root
    - mode: 644
    - require:
      - pkg: salt
```

This short stanza will ensure that vim is installed, Salt is installed and up to date, the salt-master and salt-minion daemons are running and the Salt minion configuration file is in place. It will also ensure everything is deployed in the right order and that the Salt services are restarted when the watched file updated.

The Top File

The top file controls the mapping between minions and the states which should be applied to them.

The top file specifies which minions should have which SLS files applied and which environments they should draw those SLS files from.

The top file works by specifying environments on the top-level.

Each environment contains *target expressions* to match minions. Finally, each target expression contains a list of Salt states to apply to matching minions:

```
base:
  '*':
    - salt
    - users
    - users.admin
  'saltmaster.*':
    - match: pcre
    - salt.master
```

This above example uses the base environment which is built into the default Salt setup.

The base environment has target expressions. The first one matches all minions, and the SLS files below it apply to all minions.

The second expression is a regular expression that will match all minions with an ID matching `saltmaster.*` and specifies that for those minions, the salt.master state should be applied.

Important: Since version 2014.7.0, the default matcher (when one is not explicitly defined as in the second expression in the above example) is the *compound* matcher. Since this matcher parses individual words in the expression, minion IDs containing spaces will not match properly using this matcher. Therefore, if your target expression is designed to match a minion ID containing spaces, it will be necessary to specify a different match type (such as `glob`). For example:

```
base:
  'test minion':
    - match: glob
    - foo
    - bar
    - baz
```

A full table of match types available in the top file can be found *here*.

Reloading Modules

Some Salt states require that specific packages be installed in order for the module to load. As an example the *pip* state module requires the pip package for proper name and version parsing.

In most of the common cases, Salt is clever enough to transparently reload the modules. For example, if you install a package, Salt reloads modules because some other module or state might require just that package which was installed.

On some edge-cases salt might need to be told to reload the modules. Consider the following state file which we'll call `pep8.sls`:

```
python-pip:
  cmd.run:
    - name: |
        easy_install --script-dir=/usr/bin -U pip
    - cwd: /

pep8:
  pip.installed:
    - require:
      - cmd: python-pip
```

The above example installs pip using `easy_install` from setuptools and installs pep8 using *pip*, which, as told earlier, requires pip to be installed system-wide. Let's execute this state:

```
salt-call state.apply pep8
```

The execution output would be something like:

```
----------
    State: - pip
    Name:      pep8
    Function:  installed
        Result:    False
        Comment:   State pip.installed found in sls pep8 is unavailable

        Changes:

Summary
------------
Succeeded: 1
Failed:    1
------------
Total:     2
```

If we executed the state again the output would be:

```
----------
    State: - pip
    Name:      pep8
    Function:  installed
        Result:    True
        Comment:   Package was successfully installed
        Changes:   pep8==1.4.6: Installed

Summary
------------
```

(continues on next page)

(continued from previous page)

```
Succeeded: 2
Failed:    0
------------
Total:     2
```

Since we installed pip using *cmd*, Salt has no way to know that a system-wide package was installed.

On the second execution, since the required pip package was installed, the state executed correctly.

Note: Salt does not reload modules on every state run because doing so would greatly slow down state execution.

So how do we solve this *edge-case*? `reload_modules`!

`reload_modules` is a boolean option recognized by salt on **all** available states which forces salt to reload its modules once a given state finishes.

The modified state file would now be:

```
python-pip:
  cmd.run:
    - name: |
        easy_install --script-dir=/usr/bin -U pip
    - cwd: /
    - reload_modules: true

pep8:
  pip.installed:
    - require:
      - cmd: python-pip
```

Let's run it, once:

```
salt-call state.apply pep8
```

The output is:

```
----------
    State: - pip
    Name:      pep8
    Function:  installed
        Result:    True
        Comment:   Package was successfully installed
        Changes:   pep8==1.4.6: Installed

Summary
------------
Succeeded: 2
Failed:    0
------------
Total:     2
```

CHAPTER

SEVEN

RETURN CODES

When the `salt` or `salt-call` CLI commands result in an error, the command will exit with a return code of **1**. Error cases consist of the following:

1. Errors are encountered while running *States*, or any state returns a `False` result
2. Any exception is raised
3. In the case of remote-execution functions, when the return data is a Python dictionary with a key named either `result` or `success`, which has a value of `False`

7.1 Retcode Passthrough

In addition to the cases listed above, if a state or remote-execution function sets a nonzero value in the `retcode` key of the *__context__* dictionary, the command will exit with a return code of **1**. For those developing custom states and execution modules, using `__context__['retcode']` can be a useful way of signaling that an error has occurred:

```
if something_went_wrong:
    __context__['retcode'] = 42
```

This is actually how states signal that they have failed. Different cases result in different codes being set in the *__context__* dictionary:

- **1** is set when any error is encountered in the state compiler (missing SLS file, etc.)
- **2** is set when any state returns a `False` result
- **5** is set when Pillar data fails to be compiled before running the state(s)

When the `--retcode-passthrough` flag is used with `salt-call`, then `salt-call` will exit with whichever retcode was set in the *__context__* dictionary, rather than the default behavior which simply exits with **1** for any error condition.

CHAPTER

EIGHT

UTILITY MODULES - CODE REUSE IN CUSTOM MODULES

New in version 2015.5.0.

Changed in version 2016.11.0: These can now be synced to the Master for use in custom Runners, and in custom execution modules called within Pillar SLS files.

When extending Salt by writing custom (*state modules*), *execution modules*, etc., sometimes there is a need for a function to be available to more than just one kind of custom module. For these cases, Salt supports what are called "utility modules". These modules are like normal execution modules, but instead of being invoked in Salt code using `__salt__`, the `__utils__` prefix is used instead.

For example, assuming the following simple utility module, saved to `salt://_utils/foo.py`

```
# -*- coding: utf-8 -*-
'''
My utils module
---------------

This module contains common functions for use in my other custom types.
'''

def bar():
    return 'baz'
```

Once synced to a minion, this function would be available to other custom Salt types like so:

```
# -*- coding: utf-8 -*-
'''
My awesome execution module
---------------------------
'''

def observe_the_awesomeness():
    '''
    Prints information from my utility module

    CLI Example:

    .. code-block:: bash

        salt '*' mymodule.observe_the_awesomeness
    '''
    return __utils__['foo.bar']()
```

Utility modules, like any other kind of Salt extension, support using a *__virtual__ function* to conditionally load them, or load them under a different namespace. For instance, if the utility module above were named `salt://_utils/mymodule.py` it could be made to be loaded as the `foo` utility module with a `__virtual__` function.

```
# -*- coding: utf-8 -*-
'''
My utils module
---------------

This module contains common functions for use in my other custom types.
'''

def __virtual__():
    '''
    Load as a different name
    '''
    return 'foo'

def bar():
    return 'baz'
```

New in version 2018.3.0: Instantiating objects from classes declared in util modules works with Master side modules, such as Runners, Outputters, etc.

Also you could even write your utility modules in object oriented fashion:

```
# -*- coding: utf-8 -*-
'''
My OOP-style utils module
-------------------------

This module contains common functions for use in my other custom types.
'''

class Foo(object):

    def __init__(self):
        pass

    def bar(self):
        return 'baz'
```

And import them into other custom modules:

```
# -*- coding: utf-8 -*-
'''
My awesome execution module
---------------------------
'''

import mymodule

def observe_the_awesomeness():
    '''
    Prints information from my utility module

    CLI Example:

    .. code-block:: bash
```

(continues on next page)

(continued from previous page)

```
        salt '*' mymodule.observe_the_awesomeness
    '''
    foo = mymodule.Foo()
    return foo.bar()
```

These are, of course, contrived examples, but they should serve to show some of the possibilities opened up by writing utility modules. Keep in mind though that states still have access to all of the execution modules, so it is not necessary to write a utility module to make a function available to both a state and an execution module. One good use case for utility modules is one where it is necessary to invoke the same function from a custom *outputter*/returner, as well as an execution module.

Utility modules placed in `salt://_utils/` will be synced to the minions when a *highstate* is run, as well as when any of the following Salt functions are called:

- `saltutil.sync_utils`
- `saltutil.sync_all`

As of the 2019.2.0 release, as well as 2017.7.7 and 2018.3.2 in their respective release cycles, the `sync` argument to `state.apply`/`state.sls` can be used to sync custom types when running individual SLS files.

To sync to the Master, use either of the following:

- `saltutil.sync_utils`
- `saltutil.sync_all`

CHAPTER

NINE

EVENTS & REACTOR

9.1 Event System

The Salt Event System is used to fire off events enabling third party applications or external processes to react to behavior within Salt. The event system uses a publish-subscribe pattern, otherwise know as pub/sub.

9.1.1 Event Bus

The event system is comprised of a two primary components, which make up the concept of an Event Bus:

- The event sockets, which publish events
- The event library, which can listen to events and send events into the salt system

Events are published onto the event bus and event bus subscribers listen for the published events.

The event bus is used for both inter-process communication as well as network transport in Salt. Inter-process communication is provided through UNIX domain sockets (UDX).

The Salt Master and each Salt Minion has their own event bus.

9.1.2 Event types

Salt Master Events

These events are fired on the Salt Master event bus. This list is **not** comprehensive.

Authentication events

`salt/auth`

Fired when a minion performs an authentication check with the master.

Variables

- **`id`** – The minion ID.
- **`act`** – The current status of the minion key: `accept`, `pend`, `reject`.
- **`pub`** – The minion public key.

Note: Minions fire auth events on fairly regular basis for a number of reasons. Writing reactors to respond to events through the auth cycle can lead to infinite reactor event loops (minion tries to auth, reactor responds by doing something that generates another auth event, minion sends auth event, etc.). Consider reacting to `salt/key` or `salt/minion/<MID>/start` or firing a custom event tag instead.

Start events

`salt/minion/<MID>/start`

Fired every time a minion connects to the Salt master.

Variables **`id`** – The minion ID.

Key events

`salt/key`

Fired when accepting and rejecting minions keys on the Salt master. These happen as a result of actions undertaken by the *salt-key* command.

Variables

- **`id`** – The minion ID.
- **`act`** – The new status of the minion key: `accept`, `delete`,

Warning: If a master is in `auto_accept mode`, `salt/key` events will not be fired when the keys are accepted. In addition, pre-seeding keys (like happens through *Salt-Cloud*) will not cause firing of these events.

Job events

`salt/job/<JID>/new`

Fired as a new job is sent out to minions.

Variables

- **`jid`** – The job ID.
- **`tgt`** – The target of the job: `*`, a minion ID, `G@os_family:RedHat`, etc.
- **`tgt_type`** – The type of targeting used: `glob`, `grain`, `compound`, etc.
- **`fun`** – The function to run on minions: `test.version`, `network.interfaces`, etc.
- **`arg`** – A list of arguments to pass to the function that will be called.
- **`minions`** – A list of minion IDs that Salt expects will return data for this job.
- **`user`** – The name of the user that ran the command as defined in Salt's Publisher ACL or external auth.

`salt/job/<JID>/ret/<MID>`

Fired each time a minion returns data for a job.

Variables

- **`id`** – The minion ID.

- **`jid`** – The job ID.
- **`retcode`** – The return code for the job.
- **`fun`** – The function the minion ran. E.g., `test.version`.
- **`return`** – The data returned from the execution module.

`salt/job/<JID>/prog/<MID>/<RUN NUM>`

Fired each time a each function in a state run completes execution. Must be enabled using the *`state_events`* option.

Variables

- **`data`** – The data returned from the state module function.
- **`id`** – The minion ID.
- **`jid`** – The job ID.

Runner Events

`salt/run/<JID>/new`

Fired as a runner begins execution

Variables

- **`jid`** – The job ID.
- **`fun`** – The name of the runner function, with `runner.` prepended to it (e.g. `runner.jobs.lookup_jid`)
- **`fun_args`** – The arguments passed to the runner function (e.g. `['20160829225914848058']`)
- **`user`** – The user who executed the runner (e.g. `root`)

`salt/run/<JID>/ret`

Fired when a runner function returns

Variables

- **`jid`** – The job ID.
- **`fun`** – The name of the runner function, with `runner.` prepended to it (e.g. `runner.jobs.lookup_jid`)
- **`fun_args`** – The arguments passed to the runner function (e.g. `['20160829225914848058']`)
- **`return`** – The data returned by the runner function

`salt/run/<JID>/args`

New in version 2016.11.0.

Fired by the *`state.orchestrate`* runner

Variables

- **`name`** – The ID declaration for the orchestration job (i.e. the line above `salt.state`, `salt.function`, `salt.runner`, etc.)
- **`type`** – The type of orchestration job being run (e.g. `state`)
- **`tgt`** – The target expression (e.g. `*`). Included for `state` and `function` types only.

- **`args`** – The args passed to the orchestration job. **Note**: for `state` and `function` types, also includes a `tgt_type` value which shows what kind of match (`glob`, `pcre`, etc.) was used. This value was named `expr_form` in the 2016.11 release cycle but has been renamed to `tgt_type` in 2017.7.0 for consistency with other events.

Presence Events

`salt/presence/present`

Events fired on a regular interval about currently connected, newly connected, or recently disconnected minions. Requires the *`presence_events`* setting to be enabled.

Variables **`present`** – A list of minions that are currently connected to the Salt master.

`salt/presence/change`

Fired when the Presence system detects new minions connect or disconnect.

Variables

- **`new`** – A list of minions that have connected since the last presence event.
- **`lost`** – A list of minions that have disconnected since the last presence event.

Cloud Events

Unlike other Master events, `salt-cloud` events are not fired on behalf of a Salt Minion. Instead, `salt-cloud` events are fired on behalf of a VM. This is because the minion-to-be may not yet exist to fire events to or also may have been destroyed.

This behavior is reflected by the `name` variable in the event data for `salt-cloud` events as compared to the `id` variable for Salt Minion-triggered events.

`salt/cloud/<VM NAME>/creating`

Fired when salt-cloud starts the VM creation process.

Variables

- **`name`** – the name of the VM being created.
- **`event`** – description of the event.
- **`provider`** – the cloud provider of the VM being created.
- **`profile`** – the cloud profile for the VM being created.

`salt/cloud/<VM NAME>/deploying`

Fired when the VM is available and salt-cloud begins deploying Salt to the new VM.

Variables

- **`name`** – the name of the VM being created.
- **`event`** – description of the event.
- **`kwargs`** – options available as the deploy script is invoked: `conf_file`, `deploy_command`, `display_ssh_output`, `host`, `keep_tmp`, `key_filename`, `make_minion`, `minion_conf`, `name`, `parallel`, `preseed_minion_keys`, `script`, `script_args`, `script_env`, `sock_dir`, `start_action`, `sudo`, `tmp_dir`, `tty`, `username`

`salt/cloud/<VM NAME>/requesting`

Fired when salt-cloud sends the request to create a new VM.

Variables

- **`event`** – description of the event.
- **`location`** – the location of the VM being requested.
- **`kwargs`** – options available as the VM is being requested: `Action`, `ImageId`, `InstanceType`, `KeyName`, `MaxCount`, `MinCount`, `SecurityGroup.1`

`salt/cloud/<VM NAME>/querying`

Fired when salt-cloud queries data for a new instance.

Variables

- **`event`** – description of the event.
- **`instance_id`** – the ID of the new VM.

`salt/cloud/<VM NAME>/tagging`

Fired when salt-cloud tags a new instance.

Variables

- **`event`** – description of the event.
- **`tags`** – tags being set on the new instance.

`salt/cloud/<VM NAME>/waiting_for_ssh`

Fired while the salt-cloud deploy process is waiting for ssh to become available on the new instance.

Variables

- **`event`** – description of the event.
- **`ip_address`** – IP address of the new instance.

`salt/cloud/<VM NAME>/deploy_script`

Fired once the deploy script is finished.

Variables **`event`** – description of the event.

`salt/cloud/<VM NAME>/created`

Fired once the new instance has been fully created.

Variables

- **`name`** – the name of the VM being created.
- **`event`** – description of the event.
- **`instance_id`** – the ID of the new instance.
- **`provider`** – the cloud provider of the VM being created.
- **`profile`** – the cloud profile for the VM being created.

`salt/cloud/<VM NAME>/destroying`

Fired when salt-cloud requests the destruction of an instance.

Variables

- **`name`** – the name of the VM being created.
- **`event`** – description of the event.
- **`instance_id`** – the ID of the new instance.

`salt/cloud/<VM NAME>/destroyed`

Fired when an instance has been destroyed.

Variables

- **name** – the name of the VM being created.
- **event** – description of the event.
- **instance_id** – the ID of the new instance.

9.1.3 Listening for Events

Salt's event system is used heavily within Salt and it is also written to integrate heavily with existing tooling and scripts. There is a variety of ways to consume it.

From the CLI

The quickest way to watch the event bus is by calling the *state.event runner*:

```
salt-run state.event pretty=True
```

That runner is designed to interact with the event bus from external tools and shell scripts. See the documentation for more examples.

Remotely via the REST API

Salt's event bus can be consumed *salt.netapi.rest_cherrypy.app.Events* as an HTTP stream from external tools or services.

```
curl -SsNk https://salt-api.example.com:8000/events?token=05A3
```

From Python

Python scripts can access the event bus only as the same system user that Salt is running as.

The event system is accessed via the event library and can only be accessed by the same system user that Salt is running as. To listen to events a SaltEvent object needs to be created and then the get_event function needs to be run. The SaltEvent object needs to know the location that the Salt Unix sockets are kept. In the configuration this is the `sock_dir` option. The `sock_dir` option defaults to "/var/run/salt/master" on most systems.

The following code will check for a single event:

```
import salt.config
import salt.utils.event

opts = salt.config.client_config('/etc/salt/master')

event = salt.utils.event.get_event(
        'master',
        sock_dir=opts['sock_dir'],
        transport=opts['transport'],
        opts=opts)

data = event.get_event()
```

Events will also use a "tag". Tags allow for events to be filtered by prefix. By default all events will be returned. If only authentication events are desired, then pass the tag "salt/auth".

The `get_event` method has a default poll time assigned of 5 seconds. To change this time set the "wait" option.

The following example will only listen for auth events and will wait for 10 seconds instead of the default 5.

```
data = event.get_event(wait=10, tag='salt/auth')
```

To retrieve the tag as well as the event data, pass `full=True`:

```
evdata = event.get_event(wait=10, tag='salt/job', full=True)

tag, data = evdata['tag'], evdata['data']
```

Instead of looking for a single event, the `iter_events` method can be used to make a generator which will continually yield salt events.

The iter_events method also accepts a tag but not a wait time:

```
for data in event.iter_events(tag='salt/auth'):
    print(data)
```

And finally event tags can be globbed, such as they can be in the Reactor, using the fnmatch library.

```
import fnmatch

import salt.config
import salt.utils.event

opts = salt.config.client_config('/etc/salt/master')

sevent = salt.utils.event.get_event(
        'master',
        sock_dir=opts['sock_dir'],
        transport=opts['transport'],
        opts=opts)

while True:
    ret = sevent.get_event(full=True)
    if ret is None:
        continue

    if fnmatch.fnmatch(ret['tag'], 'salt/job/*/ret/*'):
        do_something_with_job_return(ret['data'])
```

9.1.4 Firing Events

It is possible to fire events on either the minion's local bus or to fire events intended for the master.

To fire a local event from the minion on the command line call the *`event.fire`* execution function:

```
salt-call event.fire '{"data": "message to be sent in the event"}' 'tag'
```

To fire an event to be sent up to the master from the minion call the *`event.send`* execution function. Remember YAML can be used at the CLI in function arguments:

```
salt-call event.send 'myco/mytag/success' '{success: True, message: "It works!"}'
```

If a process is listening on the minion, it may be useful for a user on the master to fire an event to it. An example of listening local events on a minion on a non-Windows system:

```
# Job on minion
import salt.utils.event

opts = salt.config.minion_config('/etc/salt/minion')
event = salt.utils.event.MinionEvent(opts)

for evdata in event.iter_events(match_type = 'regex',
                                tag = 'custom/.*'):
    # do your processing here...
```

And an example of listening local events on a Windows system:

```
# Job on minion
import salt.utils.event

opts = salt.config.minion_config(salt.minion.DEFAULT_MINION_OPTS)
event = salt.utils.event.MinionEvent(opts)

for evdata in event.iter_events(match_type = 'regex',
                                tag = 'custom/.*'):
    # do your processing here...
```

```
salt minionname event.fire '{"data": "message for the minion"}' 'customtag/african/
↪unladen'
```

9.1.5 Firing Events from Python

From Salt execution modules

Events can be very useful when writing execution modules, in order to inform various processes on the master when a certain task has taken place. This is easily done using the normal cross-calling syntax:

```
# /srv/salt/_modules/my_custom_module.py

def do_something():
    '''
    Do something and fire an event to the master when finished

    CLI Example::

        salt '*' my_custom_module:do_something
    '''
    # do something!
    __salt__['event.send']('myco/my_custom_module/finished', {
        'finished': True,
        'message': "The something is finished!",
    })
```

From Custom Python Scripts

Firing events from custom Python code is quite simple and mirrors how it is done at the CLI:

```
import salt.client

caller = salt.client.Caller()

ret = called.cmd('event.send',
                 'myco/event/success'
                 { 'success': True,
                   'message': "It works!" })

if not ret:
    # the event could not be sent, process the error here
```

9.2 Beacons

Beacons let you use the Salt event system to monitor non-Salt processes. The beacon system allows the minion to hook into a variety of system processes and continually monitor these processes. When monitored activity occurs in a system process, an event is sent on the Salt event bus that can be used to trigger a *reactor*.

Salt beacons can currently monitor and send Salt events for many system activities, including:

- file system changes
- system load
- service status
- shell activity, such as user login
- network and disk usage

See *beacon modules* for a current list.

Note: Salt beacons are an event generation mechanism. Beacons leverage the Salt *reactor* system to make changes when beacon events occur.

9.2.1 Configuring Beacons

Salt beacons do not require any changes to the system components that are being monitored, everything is configured using Salt.

Beacons are typically enabled by placing a `beacons:` top level block in `/etc/salt/minion` or any file in `/etc/salt/minion.d/` such as `/etc/salt/minion.d/beacons.conf` or add it to pillars for that minion:

```
beacons:
  inotify:
    - files:
        /etc/important_file: {}
        /opt: {}
```

The beacon system, like many others in Salt, can also be configured via the minion pillar, grains, or local config file.

Note: The *inotify* beacon only works on OSes that have *inotify* kernel support. Currently this excludes FreeBSD, macOS, and Windows.

All beacon configuration is done using list based configuration.

New in version Neon.

Multiple copies of a particular Salt beacon can be configured by including the `beacon_module` parameter in the beacon configuration.

```
beacons:
  watch_importand_file:
    - files:
        /etc/important_file: {}
    - beacon_module: inotify
  watch_another_file:
    - files:
        /etc/another_file: {}
    - beacon_module: inotify
```

Beacon Monitoring Interval

Beacons monitor on a 1-second interval by default. To set a different interval, provide an `interval` argument to a beacon. The following beacons run on 5- and 10-second intervals:

```
beacons:
  inotify:
    - files:
        /etc/important_file: {}
        /opt: {}
    - interval: 5
    - disable_during_state_run: True
  load:
    - averages:
        1m:
          - 0.0
          - 2.0
        5m:
          - 0.0
          - 1.5
        15m:
          - 0.1
          - 1.0
    - interval: 10
```

Avoiding Event Loops

It is important to carefully consider the possibility of creating a loop between a reactor and a beacon. For example, one might set up a beacon which monitors whether a file is read which in turn fires a reactor to run a state which in turn reads the file and re-fires the beacon.

To avoid these types of scenarios, the `disable_during_state_run` argument may be set. If a state run is in progress, the beacon will not be run on its regular interval until the minion detects that the state run has completed, at which point the normal beacon interval will resume.

```
beacons:
  inotify:
    - files:
        /etc/important_file: {}
    - disable_during_state_run: True
```

Note: For beacon writers: If you need extra stuff to happen, like closing file handles for the `disable_during_state_run` to actually work, you can add a *close()* function to the beacon to run those extra things. See the *inotify* beacon.

9.2.2 Beacon Example

This example demonstrates configuring the *`inotify`* beacon to monitor a file for changes, and then restores the file to its original contents if a change was made.

Note: The inotify beacon requires Pyinotify on the minion, install it using `salt myminion pkg.install python-inotify`.

Create Watched File

Create the file named `/etc/important_file` and add some simple content:

```
important_config: True
```

Add Beacon Configs to Minion

On the Salt minion, add the following configuration to `/etc/salt/minion.d/beacons.conf`:

```
beacons:
  inotify:
    - files:
        /etc/important_file:
          mask:
            - modify
    - disable_during_state_run: True
```

Save the configuration file and restart the minion service. The beacon is now set up to notify salt upon modifications made to the file.

Note: The `disable_during_state_run: True` parameter *prevents* the inotify beacon from generating reactor events due to salt itself modifying the file.

View Events on the Master

On your Salt master, start the event runner using the following command:

```
salt-run state.event pretty=true
```

This runner displays events as they are received by the master on the Salt event bus. To test the beacon you set up in the previous section, make and save a modification to `/etc/important_file`. You'll see an event similar to the following on the event bus:

```
{
 "_stamp": "2015-09-09T15:59:37.972753",
 "data": {
     "change": "IN_IGNORED",
     "id": "larry",
     "path": "/etc/important_file"
 },
 "tag": "salt/beacon/larry/inotify//etc/important_file"
}
```

This indicates that the event is being captured and sent correctly. Now you can create a reactor to take action when this event occurs.

Create a Reactor

This reactor reverts the file named `/etc/important_file` to the contents provided by salt each time it is modified.

Reactor SLS

On your Salt master, create a file named `/srv/reactor/revert.sls`.

Note: If the `/srv/reactor` directory doesn't exist, create it.

```
mkdir -p /srv/reactor
```

Add the following to `/srv/reactor/revert.sls`:

```
revert-file:
  local.state.apply:
    - tgt: {{ data['data']['id'] }}
    - arg:
      - maintain_important_file
```

Note: In addition to *setting* `disable_during_state_run: True` for an inotify beacon whose reaction is to modify the watched file, it is important to ensure the state applied is also *idempotent*.

Note: The expression `{{ data['data']['id'] }}` *is correct* as it matches the event structure *shown above*.

State SLS

Create the state sls file referenced by the reactor sls file. This state file will be located at `/srv/salt/maintain_important_file.sls`.

```
important_file:
  file.managed:
    - name: /etc/important_file
    - contents: |
        important_config: True
```

Master Config

Configure the master to map the inotify beacon event to the `revert` reaction in `/etc/salt/master.d/reactor.conf`:

```
reactor:
  - salt/beacon/*/inotify//etc/important_file:
    - /srv/reactor/revert.sls
```

Note: You can have only one top level `reactor` section, so if one already exists, add this code to the existing section. See *here* to learn more about reactor SLS syntax.

Start the Salt Master in Debug Mode

To help with troubleshooting, start the Salt master in debug mode:

```
service salt-master stop
salt-master -l debug
```

When debug logging is enabled, event and reactor data are displayed so you can discover syntax and other issues.

Trigger the Reactor

On your minion, make and save another change to `/etc/important_file`. On the Salt master, you'll see debug messages that indicate the event was received and the `state.apply` job was sent. When you inspect the file on the minion, you'll see that the file contents have been restored to `important_config: True`.

All beacons are configured using a similar process of enabling the beacon, writing a reactor SLS (and state SLS if needed), and mapping a beacon event to the reactor SLS.

9.2.3 Writing Beacon Plugins

Beacon plugins use the standard Salt loader system, meaning that many of the constructs from other plugin systems holds true, such as the `__virtual__` function.

The important function in the Beacon Plugin is the `beacon` function. When the beacon is configured to run, this function will be executed repeatedly by the minion. The `beacon` function therefore cannot block and should be as lightweight as possible. The `beacon` also must return a list of dicts, each dict in the list will be translated into an event on the master.

Beacons may also choose to implement a `validate` function which takes the beacon configuration as an argument and ensures that it is valid prior to continuing. This function is called automatically by the Salt loader when a beacon is loaded.

Please see the *`inotify`* beacon as an example.

The *beacon* Function

The beacons system will look for a function named *beacon* in the module. If this function is not present then the beacon will not be fired. This function is called on a regular basis and defaults to being called on every iteration of the minion, which can be tens to hundreds of times a second. This means that the *beacon* function cannot block and should not be CPU or IO intensive.

The beacon function will be passed in the configuration for the executed beacon. This makes it easy to establish a flexible configuration for each called beacon. This is also the preferred way to ingest the beacon's configuration as it allows for the configuration to be dynamically updated while the minion is running by configuring the beacon in the minion's pillar.

The Beacon Return

The information returned from the beacon is expected to follow a predefined structure. The returned value needs to be a list of dictionaries (standard python dictionaries are preferred, no ordered dicts are needed).

The dictionaries represent individual events to be fired on the minion and master event buses. Each dict is a single event. The dict can contain any arbitrary keys but the 'tag' key will be extracted and added to the tag of the fired event.

The return data structure would look something like this:

```
[{'changes': ['/foo/bar'], 'tag': 'foo'},
 {'changes': ['/foo/baz'], 'tag': 'bar'}]
```

Calling Execution Modules

Execution modules are still the preferred location for all work and system interaction to happen in Salt. For this reason the __*salt*__ variable is available inside the beacon.

Please be careful when calling functions in __*salt*__, while this is the preferred means of executing complicated routines in Salt not all of the execution modules have been written with beacons in mind. Watch out for execution modules that may be CPU intense or IO bound. Please feel free to add new execution modules and functions to back specific beacons.

Distributing Custom Beacons

Custom beacons can be distributed to minions via the standard methods, see *Modular Systems*.

9.3 Reactor System

Salt's Reactor system gives Salt the ability to trigger actions in response to an event. It is a simple interface to watching Salt's event bus for event tags that match a given pattern and then running one or more commands in response.

This system binds sls files to event tags on the master. These sls files then define reactions. This means that the reactor system has two parts. First, the reactor option needs to be set in the master configuration file. The reactor option allows for event tags to be associated with sls reaction files. Second, these reaction files use highdata (like the state system) to define reactions to be executed.

9.3.1 Event System

A basic understanding of the event system is required to understand reactors. The event system is a local ZeroMQ PUB interface which fires salt events. This event bus is an open system used for sending information notifying Salt and other systems about operations.

The event system fires events with a very specific criteria. Every event has a **tag**. Event tags allow for fast top-level filtering of events. In addition to the tag, each event has a data structure. This data structure is a dictionary, which contains information about the event.

9.3.2 Mapping Events to Reactor SLS Files

Reactor SLS files and event tags are associated in the master config file. By default this is /etc/salt/master, or /etc/salt/master.d/reactor.conf.

New in version 2014.7.0: Added Reactor support for `salt://` file paths.

In the master config section 'reactor:' is a list of event tags to be matched and each event tag has a list of reactor SLS files to be run.

```
reactor:                            # Master config section "reactor"

  - 'salt/minion/*/start':          # Match tag "salt/minion/*/start"
    - /srv/reactor/start.sls        # Things to do when a minion starts
    - /srv/reactor/monitor.sls      # Other things to do

  - 'salt/cloud/*/destroyed':       # Globs can be used to match tags
    - /srv/reactor/destroy/*.sls    # Globs can be used to match file names

  - 'myco/custom/event/tag':        # React to custom event tags
    - salt://reactor/mycustom.sls   # Reactor files can come from the salt fileserver
```

Note: In the above example, `salt://reactor/mycustom.sls` refers to the `base` environment. To pull this file from a different environment, use the *querystring syntax* (e.g. `salt://reactor/mycustom.sls?saltenv=reactor`).

Reactor SLS files are similar to State and Pillar SLS files. They are by default YAML + Jinja templates and are passed familiar context variables. Click *here* for more detailed information on the variables available in Jinja templating.

Here is the SLS for a simple reaction:

```
{% if data['id'] == 'mysql1' %}
highstate_run:
  local.state.apply:
    - tgt: mysql1
{% endif %}
```

This simple reactor file uses Jinja to further refine the reaction to be made. If the `id` in the event data is `mysql1` (in other words, if the name of the minion is `mysql1`) then the following reaction is defined. The same data structure and compiler used for the state system is used for the reactor system. The only difference is that the data is matched up to the salt command API and the runner system. In this example, a command is published to the `mysql1` minion with a function of *`state.apply`*, which performs a *highstate*. Similarly, a runner can be called:

```
{% if data['data']['custom_var'] == 'runit' %}
call_runit_orch:
  runner.state.orchestrate:
    - args:
      - mods: orchestrate.runit
{% endif %}
```

This example will execute the state.orchestrate runner and intiate an execution of the `runit` orchestrator located at `/srv/salt/orchestrate/runit.sls`.

9.3.3 Types of Reactions

Name	Description
local	Runs a *remote-execution function* on targeted minions
runner	Executes a *runner function*
wheel	Executes a *wheel function* on the master
caller	Runs a *remote-execution function* on a masterless minion

---

Note: The `local` and `caller` reaction types will likely be renamed in a future release. These reaction types were named after Salt's internal client interfaces, and are not intuitively named. Both `local` and `caller` will continue to work in Reactor SLS files, however.

9.3.4 Where to Put Reactor SLS Files

Reactor SLS files can come both from files local to the master, and from any of backends enabled via the *`fileserver_backend`* config option. Files placed in the Salt fileserver can be referenced using a `salt://` URL, just like they can in State SLS files.

It is recommended to place reactor and orchestrator SLS files in their own uniquely-named subdirectories such as `orch/`, `orchestrate/`, `react/`, `reactor/`, etc., to keep them organized.

9.3.5 Writing Reactor SLS

The different reaction types were developed separately and have historically had different methods for passing arguments. For the 2017.7.2 release a new, unified configuration schema has been introduced, which applies to all reaction types.

The old config schema will continue to be supported, and there is no plan to deprecate it at this time.

Local Reactions

A `local` reaction runs a *remote-execution function* on the targeted minions.

The old config schema required the positional and keyword arguments to be manually separated by the user under `arg` and `kwarg` parameters. However, this is not very user-friendly, as it forces the user to distinguish which type of argument is which, and make sure that positional arguments are ordered properly. Therefore, the new config schema is recommended if the master is running a supported release.

The below two examples are equivalent:

Supported in 2017.7.2 and later

```
install_zsh:
  local.state.single:
    - tgt: 'kernel:Linux'
    - tgt_type: grain
    - args:
      - fun: pkg.installed
      - name: zsh
      - fromrepo: updates
```

Supported in all releases

```
install_zsh:
  local.state.single:
    - tgt: 'kernel:Linux'
    - tgt_type: grain
    - arg:
      - pkg.installed
      - zsh
    - kwarg:
        fromrepo: updates
```

This reaction would be equivalent to running the following Salt command:

```
salt -G 'kernel:Linux' state.single pkg.installed name=zsh fromrepo=updates
```

Note: Any other parameters in the *`LocalClient().cmd_async()`* method can be passed at the same indentation level as `tgt`.

Note: `tgt_type` is only required when the target expression defined in `tgt` uses a *target type* other than a minion ID glob.

The `tgt_type` argument was named `expr_form` in releases prior to 2017.7.0.

Runner Reactions

Runner reactions execute *runner functions* locally on the master.

The old config schema called for passing arguments to the reaction directly under the name of the runner function. However, this can cause unpredictable interactions with the Reactor system's internal arguments. It is also possible to pass positional and keyword arguments under `arg` and `kwarg` like above in *local reactions*, but as noted above this is not very user-friendly. Therefore, the new config schema is recommended if the master is running a supported release.

The below two examples are equivalent:

Supported in 2017.7.2 and later

```
deploy_app:
  runner.state.orchestrate:
    - args:
      - mods: orchestrate.deploy_app
      - pillar:
          event_tag: {{ tag }}
          event_data: {{ data['data
↪']|json }}
```

Supported in all releases

```
deploy_app:
  runner.state.orchestrate:
    - mods: orchestrate.deploy_app
    - kwarg:
        pillar:
          event_tag: {{ tag }}
          event_data: {{ data['data
↪']|json }}
```

Assuming that the event tag is `foo`, and the data passed to the event is `{'bar': 'baz'}`, then this reaction is equivalent to running the following Salt command:

```
salt-run state.orchestrate mods=orchestrate.deploy_app pillar='{"event_tag": "foo", "event_data": {"bar": "baz"}}'
```

Wheel Reactions

Wheel reactions run *wheel functions* locally on the master.

Like *runner reactions*, the old config schema called for wheel reactions to have arguments passed directly under the name of the *wheel function* (or in `arg` or `kwarg` parameters).

The below two examples are equivalent:

Supported in 2017.7.2 and later

```
remove_key:
  wheel.key.delete:
    - args:
      - match: {{ data['id'] }}
```

Supported in all releases

```
remove_key:
  wheel.key.delete:
    - match: {{ data['id'] }}
```

Caller Reactions

Caller reactions run *remote-execution functions* on a minion daemon's Reactor system. To run a Reactor on the minion, it is necessary to configure the `Reactor Engine` in the minion config file, and then setup your watched events in a `reactor` section in the minion config file as well.

Note: Masterless Minions use this Reactor

This is the only way to run the Reactor if you use masterless minions.

Both the old and new config schemas involve passing arguments under an `args` parameter. However, the old config schema only supports positional arguments. Therefore, the new config schema is recommended if the masterless minion is running a supported release.

The below two examples are equivalent:

Supported in 2017.7.2 and later

```
touch_file:
  caller.file.touch:
    - args:
      - name: /tmp/foo
```

Supported in all releases

```
touch_file:
  caller.file.touch:
    - args:
      - /tmp/foo
```

This reaction is equivalent to running the following Salt command:

```
salt-call file.touch name=/tmp/foo
```

9.3.6 Best Practices for Writing Reactor SLS Files

The Reactor works as follows:

1. The Salt Reactor watches Salt's event bus for new events.
2. Each event's tag is matched against the list of event tags configured under the `reactor` section in the Salt Master config.
3. The SLS files for any matches are rendered into a data structure that represents one or more function calls.
4. That data structure is given to a pool of worker threads for execution.

Matching and rendering Reactor SLS files is done sequentially in a single process. For that reason, reactor SLS files should contain few individual reactions (one, if at all possible). Also, keep in mind that reactions are fired asynchronously (with the exception of *caller*) and do *not* support *requisites*.

Complex Jinja templating that calls out to slow *remote-execution* or *runner* functions slows down the rendering and causes other reactions to pile up behind the current one. The worker pool is designed to handle complex and long-running processes like *orchestration* jobs.

Therefore, when complex tasks are in order, *orchestration* is a natural fit. Orchestration SLS files can be more complex, and use requisites. Performing a complex task using orchestration lets the Reactor system fire off the orchestration job and proceed with processing other reactions.

9.3.7 Jinja Context

Reactor SLS files only have access to a minimal Jinja context. `grains` and `pillar` are *not* available. The `salt` object is available for calling *remote-execution* or *runner* functions, but it should be used sparingly and only for quick tasks for the reasons mentioned above.

In addition to the `salt` object, the following variables are available in the Jinja context:

- `tag` - the tag from the event that triggered execution of the Reactor SLS file
- `data` - the event's data dictionary

The `data` dict will contain an `id` key containing the minion ID, if the event was fired from a minion, and a `data` key containing the data passed to the event.

9.3.8 Advanced State System Capabilities

Reactor SLS files, by design, do not support *requisites*, ordering, `onlyif/unless` conditionals and most other powerful constructs from Salt's State system.

Complex Master-side operations are best performed by Salt's Orchestrate system so using the Reactor to kick off an Orchestrate run is a very common pairing.

For example:

```
# /etc/salt/master.d/reactor.conf
# A custom event containing: {"foo": "Foo!", "bar: "bar*", "baz": "Baz!"}
reactor:
  - my/custom/event:
    - /srv/reactor/some_event.sls
```

```
# /srv/reactor/some_event.sls
invoke_orchestrate_file:
  runner.state.orchestrate:
    - args:
        - mods: orchestrate.do_complex_thing
        - pillar:
            event_tag: {{ tag }}
            event_data: {{ data|json }}
```

```
# /srv/salt/orchestrate/do_complex_thing.sls
{% set tag = salt.pillar.get('event_tag') %}
{% set data = salt.pillar.get('event_data') %}

# Pass data from the event to a custom runner function.
# The function expects a 'foo' argument.
do_first_thing:
  salt.runner:
    - name: custom_runner.custom_function
    - foo: {{ data.foo }}

# Wait for the runner to finish then send an execution to minions.
# Forward some data from the event down to the minion's state run.
do_second_thing:
  salt.state:
    - tgt: {{ data.bar }}
    - sls:
      - do_thing_on_minion
    - kwarg:
        pillar:
          baz: {{ data.baz }}
    - require:
      - salt: do_first_thing
```

9.3.9 Beacons and Reactors

An event initiated by a beacon, when it arrives at the master will be wrapped inside a second event, such that the data object containing the beacon information will be `data['data']`, rather than `data`.

For example, to access the `id` field of the beacon event in a reactor file, you will need to reference `{{ data['data']['id'] }}` rather than `{{ data['id'] }}` as for events initiated directly on the event bus.

Similarly, the data dictionary attached to the event would be located in `{{ data['data']['data'] }}` instead of `{{ data['data'] }}`.

See the *beacon documentation* for examples.

9.3.10 Manually Firing an Event

From the Master

Use the *`event.send`* runner:

```
salt-run event.send foo '{orchestrate: refresh}'
```

From the Minion

To fire an event to the master from a minion, call *`event.send`*:

```
salt-call event.send foo '{orchestrate: refresh}'
```

To fire an event to the minion's local event bus, call *`event.fire`*:

```
salt-call event.fire '{orchestrate: refresh}' foo
```

Referencing Data Passed in Events

Assuming any of the above examples, any reactor SLS files triggered by watching the event tag `foo` will execute with `{{ data['data']['orchestrate'] }}` equal to `'refresh'`.

9.3.11 Getting Information About Events

The best way to see exactly what events have been fired and what data is available in each event is to use the *`state.event runner`*.

See also:

Common Salt Events

Example usage:

```
salt-run state.event pretty=True
```

Example output:

```
salt/job/20150213001905721678/new       {
    "_stamp": "2015-02-13T00:19:05.724583",
    "arg": [],
    "fun": "test.ping",
    "jid": "20150213001905721678",
    "minions": [
        "jerry"
    ],
    "tgt": "*",
    "tgt_type": "glob",
    "user": "root"
}
salt/job/20150213001910749506/ret/jerry {
    "_stamp": "2015-02-13T00:19:11.136730",
    "cmd": "_return",
    "fun": "saltutil.find_job",
    "fun_args": [
        "20150213001905721678"
    ],
    "id": "jerry",
    "jid": "20150213001910749506",
    "retcode": 0,
    "return": {},
    "success": true
}
```

9.3.12 Debugging the Reactor

The best window into the Reactor is to run the master in the foreground with debug logging enabled. The output will include when the master sees the event, what the master does in response to that event, and it will also include the rendered SLS file (or any errors generated while rendering the SLS file).

1. Stop the master.
2. Start the master manually:

```
salt-master -l debug
```

3. Look for log entries in the form:

```
[DEBUG   ] Gathering reactors for tag foo/bar
[DEBUG   ] Compiling reactions for tag foo/bar
[DEBUG   ] Rendered data from file: /path/to/the/reactor_file.sls:
<... Rendered output appears here. ...>
```

The rendered output is the result of the Jinja parsing and is a good way to view the result of referencing Jinja variables. If the result is empty then Jinja produced an empty result and the Reactor will ignore it.

Passing Event Data to Minions or Orchestration as Pillar

An interesting trick to pass data from the Reactor SLS file to *`state.apply`* is to pass it as inline Pillar data since both functions take a keyword argument named `pillar`.

The following example uses Salt's Reactor to listen for the event that is fired when the key for a new minion is accepted on the master using `salt-key`.

`/etc/salt/master.d/reactor.conf`:

```
reactor:
  - 'salt/key':
    - /srv/salt/haproxy/react_new_minion.sls
```

The Reactor then fires a :*`state.apply`* command targeted to the HAProxy servers and passes the ID of the new minion from the event to the state file via inline Pillar.

`/srv/salt/haproxy/react_new_minion.sls`:

```
{% if data['act'] == 'accept' and data['id'].startswith('web') %}
add_new_minion_to_pool:
  local.state.apply:
    - tgt: 'haproxy*'
    - args:
      - mods: haproxy.refresh_pool
      - pillar:
          new_minion: {{ data['id'] }}
{% endif %}
```

The above command is equivalent to the following command at the CLI:

```
salt 'haproxy*' state.apply haproxy.refresh_pool pillar='{new_minion: minionid}'
```

This works with Orchestrate files as well:

```
call_some_orchestrate_file:
  runner.state.orchestrate:
    - args:
      - mods: orchestrate.some_orchestrate_file
      - pillar:
          stuff: things
```

Which is equivalent to the following command at the CLI:

```
salt-run state.orchestrate orchestrate.some_orchestrate_file pillar='{stuff: things}'
```

Finally, that data is available in the state file using the normal Pillar lookup syntax. The following example is grabbing web server names and IP addresses from *Salt Mine*. If this state is invoked from the Reactor then the custom Pillar value from above will be available and the new minion will be added to the pool but with the `disabled` flag so that HAProxy won't yet direct traffic to it.

`/srv/salt/haproxy/refresh_pool.sls`:

```
{% set new_minion = salt['pillar.get']('new_minion') %}

listen web *:80
    balance source
    {% for server,ip in salt['mine.get']('web*', 'network.interfaces', ['eth0']).
↪items() %}
    {% if server == new_minion %}
    server {{ server }} {{ ip }}:80 disabled
    {% else %}
    server {{ server }} {{ ip }}:80 check
    {% endif %}
    {% endfor %}
```

9.3.13 A Complete Example

In this example, we're going to assume that we have a group of servers that will come online at random and need to have keys automatically accepted. We'll also add that we don't want all servers being automatically accepted. For this example, we'll assume that all hosts that have an id that starts with 'ink' will be automatically accepted and have `state.apply` executed. On top of this, we're going to add that a host coming up that was replaced (meaning a new key) will also be accepted.

Our master configuration will be rather simple. All minions that attempte to authenticate will match the **tag** of **salt/auth**. When it comes to the minion key being accepted, we get a more refined **tag** that includes the minion id, which we can use for matching.

`/etc/salt/master.d/reactor.conf`:

```
reactor:
  - 'salt/auth':
    - /srv/reactor/auth-pending.sls
  - 'salt/minion/ink*/start':
    - /srv/reactor/auth-complete.sls
```

In this SLS file, we say that if the key was rejected we will delete the key on the master and then also tell the master to ssh in to the minion and tell it to restart the minion, since a minion process will die if the key is rejected.

We also say that if the key is pending and the id starts with ink we will accept the key. A minion that is waiting on a pending key will retry authentication every ten seconds by default.

`/srv/reactor/auth-pending.sls`:

```
{# Ink server failed to authenticate -- remove accepted key #}
{% if not data['result'] and data['id'].startswith('ink') %}
minion_remove:
  wheel.key.delete:
    - args:
      - match: {{ data['id'] }}
minion_rejoin:
  local.cmd.run:
    - tgt: salt-master.domain.tld
    - args:
      - cmd: ssh -o UserKnownHostsFile=/dev/null -o StrictHostKeyChecking=no "{{ data[
↪'id'] }}" 'sleep 10 && /etc/init.d/salt-minion restart'
{% endif %}

{# Ink server is sending new key -- accept this key #}
{% if 'act' in data and data['act'] == 'pend' and data['id'].startswith('ink') %}
minion_add:
  wheel.key.accept:
    - args:
      - match: {{ data['id'] }}
{% endif %}
```

No if statements are needed here because we already limited this action to just Ink servers in the master configuration.

`/srv/reactor/auth-complete.sls`:

```
{# When an Ink server connects, run state.apply. #}
highstate_run:
  local.state.apply:
    - tgt: {{ data['id'] }}
    - ret: smtp
```

The above will also return the *highstate* result data using the *smtp_return* returner (use virtualname like when using from the command line with *–return*). The returner needs to be configured on the minion for this to work. See `salt.returners.smtp_return` documentation for that.

9.3.14 Syncing Custom Types on Minion Start

Salt will sync all custom types (by running a `saltutil.sync_all`) on every *highstate*. However, there is a chicken-and-egg issue where, on the initial *highstate*, a minion will not yet have these custom types synced when the top file is first compiled. This can be worked around with a simple reactor which watches for `salt/minion/*/start` events, which each minion fires when it first starts up and connects to the master.

On the master, create **/srv/reactor/sync_grains.sls** with the following contents:

```
sync_grains:
  local.saltutil.sync_grains:
    - tgt: {{ data['id'] }}
```

And in the master config file, add the following reactor configuration:

```
reactor:
  - 'salt/minion/*/start':
    - /srv/reactor/sync_grains.sls
```

This will cause the master to instruct each minion to sync its custom grains when it starts, making these grains available when the initial *highstate* is executed.

Other types can be synced by replacing `local.saltutil.sync_grains` with `local.saltutil.sync_modules`, `local.saltutil.sync_all`, or whatever else suits the intended use case.

Also, if it is not desirable that *every* minion syncs on startup, the `*` can be replaced with a different glob to narrow down the set of minions which will match that reactor (e.g. `salt/minion/appsrv*/start`, which would only match minion IDs beginning with `appsrv`).

CHAPTER

TEN

ORCHESTRATION

10.1 Orchestrate Runner

Executing states or highstate on a minion is perfect when you want to ensure that minion configured and running the way you want. Sometimes however you want to configure a set of minions all at once.

For example, if you want to set up a load balancer in front of a cluster of web servers you can ensure the load balancer is set up first, and then the same matching configuration is applied consistently across the whole cluster.

Orchestration is the way to do this.

10.1.1 The Orchestrate Runner

New in version 0.17.0.

Note: Orchestrate Deprecates OverState

The Orchestrate Runner (originally called the state.sls runner) offers all the functionality of the OverState, but with some advantages:

- All *Requisites and Other Global State Arguments* available in states can be used.
- The states/functions will also work on salt-ssh minions.

The Orchestrate Runner replaced the OverState system in Salt 2015.8.0.

The orchestrate runner generalizes the Salt state system to a Salt master context. Whereas the `state.sls`, `state.highstate`, et al. functions are concurrently and independently executed on each Salt minion, the `state.orchestrate` runner is executed on the master, giving it a master-level view and control over requisites, such as state ordering and conditionals. This allows for inter minion requisites, like ordering the application of states on different minions that must not happen simultaneously, or for halting the state run on all minions if a minion fails one of its states.

The `state.sls`, `state.highstate`, et al. functions allow you to statefully manage each minion and the `state.orchestrate` runner allows you to statefully manage your entire infrastructure.

Writing SLS Files

Orchestrate SLS files are stored in the same location as State SLS files. This means that both `file_roots` and `gitfs_remotes` impact what SLS files are available to the reactor and orchestrator.

It is recommended to keep reactor and orchestrator SLS files in their own uniquely named subdirectories such as `_orch/`, `orch/`, `_orchestrate/`, `react/`, `_reactor/`, etc. This will avoid duplicate naming and will help prevent confusion.

Executing the Orchestrate Runner

The Orchestrate Runner command format is the same as for the `state.sls` function, except that since it is a runner, it is executed with `salt-run` rather than `salt`. Assuming you have a state.sls file called `/srv/salt/orch/webserver.sls` the following command, run on the master, will apply the states defined in that file.

```
salt-run state.orchestrate orch.webserver
```

Note: `state.orch` is a synonym for `state.orchestrate`

Changed in version 2014.1.1: The runner function was renamed to `state.orchestrate` to avoid confusion with the *`state.sls`* execution function. In versions 0.17.0 through 2014.1.0, `state.sls` must be used.

Masterless Orchestration

New in version 2016.11.0.

To support salt orchestration on masterless minions, the Orchestrate Runner is available as an execution module. The syntax for masterless orchestration is exactly the same, but it uses the `salt-call` command and the minion configuration must contain the `file_mode: local` option. Alternatively, use `salt-call --local` on the command line.

```
salt-call --local state.orchestrate orch.webserver
```

Note: Masterless orchestration supports only the `salt.state` command in an sls file; it does not (currently) support the `salt.function` command.

Examples

Function

To execute a function, use *`salt.function`*:

```
# /srv/salt/orch/cleanfoo.sls
cmd.run:
  salt.function:
    - tgt: '*'
    - arg:
      - rm -rf /tmp/foo
```

```
salt-run state.orchestrate orch.cleanfoo
```

If you omit the "name" argument, the ID of the state will be the default name, or in the case of `salt.function`, the execution module function to run. You can specify the "name" argument to avoid conflicting IDs:

```
copy_some_file:
  salt.function:
    - name: file.copy
    - tgt: '*'
    - arg:
      - /path/to/file
      - /tmp/copy_of_file
    - kwarg:
        remove_existing: true
```

Fail Functions

When running a remote execution function in orchestration, certain return values for those functions may indicate failure, while the function itself doesn't set a return code. For those circumstances, using a "fail function" allows for a more flexible means of assessing success or failure.

A fail function can be written as part of a *custom execution module*. The function should accept one argument, and return a boolean result. For example:

```
def check_func_result(retval):
    if some_condition:
        return True
    else:
        return False
```

The function can then be referenced in orchestration SLS like so:

```
do_stuff:
  salt.function:
    - name: modname.funcname
    - tgt: '*'
    - fail_function: mymod.check_func_result
```

Important: Fail functions run *on the master*, so they must be synced using `salt-run saltutil.sync_modules`.

State

To execute a state, use *`salt.state`*.

```
# /srv/salt/orch/webserver.sls
install_nginx:
  salt.state:
    - tgt: 'web*'
    - sls:
      - nginx
```

```
salt-run state.orchestrate orch.webserver
```

Highstate

To run a highstate, set `highstate: True` in your state config:

```
# /srv/salt/orch/web_setup.sls
webserver_setup:
  salt.state:
    - tgt: 'web*'
    - highstate: True
```

```
salt-run state.orchestrate orch.web_setup
```

Runner

To execute another runner, use *`salt.runner`*. For example to use the `cloud.profile` runner in your orchestration state additional options to replace values in the configured profile, use this:

```
# /srv/salt/orch/deploy.sls
create_instance:
  salt.runner:
    - name: cloud.profile
    - prof: cloud-centos
    - provider: cloud
    - instances:
      - server1
    - opts:
        minion:
          master: master1
```

To get a more dynamic state, use jinja variables together with `inline pillar data`. Using the same example but passing on pillar data, the state would be like this.

```
# /srv/salt/orch/deploy.sls
{% set servers = salt['pillar.get']('servers', 'test') %}
{% set master = salt['pillar.get']('master', 'salt') %}
create_instance:
  salt.runner:
    - name: cloud.profile
    - prof: cloud-centos
    - provider: cloud
    - instances:
      - {{ servers }}
    - opts:
        minion:
          master: {{ master }}
```

To execute with pillar data.

```
salt-run state.orch orch.deploy pillar='{"servers": "newsystem1",
"master": "mymaster"}'
```

Return Codes in Runner/Wheel Jobs

New in version 2018.3.0.

State (`salt.state`) jobs are able to report failure via the *state return dictionary*. Remote execution (`salt.function`) jobs are able to report failure by setting a `retcode` key in the `__context__` dictionary. However, runner (`salt.runner`) and wheel (`salt.wheel`) jobs would only report a `False` result when the runner/wheel function raised an exception. As of the 2018.3.0 release, it is now possible to set a retcode in runner and wheel functions just as you can do in remote execution functions. Here is some example pseudocode:

```
def myrunner():
    ...
    do stuff
    ...
    if some_error_condition:
        __context__['retcode'] = 1
    return result
```

This allows a custom runner/wheel function to report its failure so that requisites can accurately tell that a job has failed.

More Complex Orchestration

Many states/functions can be configured in a single file, which when combined with the full suite of *Requisites and Other Global State Arguments*, can be used to easily configure complex orchestration tasks. Additionally, the states/functions will be executed in the order in which they are defined, unless prevented from doing so by any *Requisites and Other Global State Arguments*, as is the default in SLS files since 0.17.0.

```
bootstrap_servers:
  salt.function:
    - name: cmd.run
    - tgt: 10.0.0.0/24
    - tgt_type: ipcidr
    - arg:
      - bootstrap

storage_setup:
  salt.state:
    - tgt: 'role:storage'
    - tgt_type: grain
    - sls: ceph
    - require:
      - salt: webserver_setup

webserver_setup:
  salt.state:
    - tgt: 'web*'
    - highstate: True
```

Given the above setup, the orchestration will be carried out as follows:

1. The shell command `bootstrap` will be executed on all minions in the 10.0.0.0/24 subnet.
2. A Highstate will be run on all minions whose ID starts with ”web”, since the `storage_setup` state requires it.
3. Finally, the `ceph` SLS target will be executed on all minions which have a grain called `role` with a value of `storage`.

Note: Remember, salt-run is *always* executed on the master.

10.1.2 Parsing Results Programmatically

Orchestration jobs return output in a specific data structure. That data structure is represented differently depending on the outputter used. With the default outputter for orchestration, you get a nice human-readable output. Assume the following orchestration SLS:

```
good_state:
  salt.state:
    - tgt: myminion
    - sls:
    - succeed_with_changes

bad_state:
  salt.state:
    - tgt: myminion
    - sls:
    - fail_with_changes

mymod.myfunc:
  salt.function:
    - tgt: myminion

mymod.myfunc_false_result:
  salt.function:
    - tgt: myminion
```

Running this using the default outputter would produce output which looks like this:

```
fa5944a73aa8_master:
----------
          ID: good_state
    Function: salt.state
      Result: True
     Comment: States ran successfully. Updating myminion.
     Started: 21:08:02.681604
    Duration: 265.565 ms
     Changes:
              myminion:
              ----------
                        ID: test succeed with changes
                  Function: test.succeed_with_changes
                    Result: True
                   Comment: Success!
                   Started: 21:08:02.835893
                  Duration: 0.375 ms
                   Changes:
                            ----------
                            testing:
                                ----------
                                new:
                                    Something pretended to change
                                old:
                                    Unchanged

              Summary for myminion
              ------------
              Succeeded: 1 (changed=1)
              Failed:    0
```

(continues on next page)

(continued from previous page)

```
              ------------
              Total states run:     1
              Total run time:   0.375 ms
----------
          ID: bad_state
    Function: salt.state
      Result: False
     Comment: Run failed on minions: myminion
     Started: 21:08:02.947702
    Duration: 177.01 ms
     Changes:
              myminion:
              ----------
                        ID: test fail with changes
                  Function: test.fail_with_changes
                    Result: False
                   Comment: Failure!
                   Started: 21:08:03.116634
                  Duration: 0.502 ms
                   Changes:
                            ----------
                            testing:
                                ----------
                                new:
                                    Something pretended to change
                                old:
                                    Unchanged

              Summary for myminion
              ------------
              Succeeded: 0 (changed=1)
              Failed:    1
              ------------
              Total states run:     1
              Total run time:   0.502 ms
----------
          ID: mymod.myfunc
    Function: salt.function
      Result: True
     Comment: Function ran successfully. Function mymod.myfunc ran on myminion.
     Started: 21:08:03.125011
    Duration: 159.488 ms
     Changes:
              myminion:
                  True
----------
          ID: mymod.myfunc_false_result
    Function: salt.function
      Result: False
     Comment: Running function mymod.myfunc_false_result failed on minions: myminion.
↪Function mymod.myfunc_false_result ran on myminion.
     Started: 21:08:03.285148
    Duration: 176.787 ms
     Changes:
              myminion:
                  False
```

(continues on next page)

(continued from previous page)

```

Summary for fa5944a73aa8_master
------------
Succeeded: 2 (changed=4)
Failed:    2
------------
Total states run:     4
Total run time: 778.850 ms
```

However, using the `json` outputter, you can get the output in an easily loadable and parsable format:

```
salt-run state.orchestrate test --out=json
```

```
{
    "outputter": "highstate",
    "data": {
        "fa5944a73aa8_master": {
            "salt_|-good_state_|-good_state_|-state": {
                "comment": "States ran successfully. Updating myminion.",
                "name": "good_state",
                "start_time": "21:35:16.868345",
                "result": true,
                "duration": 267.299,
                "__run_num__": 0,
                "__jid__": "20171130213516897392",
                "__sls__": "test",
                "changes": {
                    "ret": {
                        "myminion": {
                            "test_|-test succeed with changes_|-test succeed with
↪changes_|-succeed_with_changes": {
                                "comment": "Success!",
                                "name": "test succeed with changes",
                                "start_time": "21:35:17.022592",
                                "result": true,
                                "duration": 0.362,
                                "__run_num__": 0,
                                "__sls__": "succeed_with_changes",
                                "changes": {
                                    "testing": {
                                        "new": "Something pretended to change",
                                        "old": "Unchanged"
                                    }
                                },
                                "__id__": "test succeed with changes"
                            }
                        }
                    },
                    "out": "highstate"
                },
                "__id__": "good_state"
            },
            "salt_|-bad_state_|-bad_state_|-state": {
                "comment": "Run failed on minions: test",
                "name": "bad_state",
                "start_time": "21:35:17.136511",
```

(continues on next page)

(continued from previous page)

```
                "result": false,
                "duration": 197.635,
                "__run_num__": 1,
                "__jid__": "20171130213517202203",
                "__sls__": "test",
                "changes": {
                    "ret": {
                        "myminion": {
                            "test_|-test fail with changes_|-test fail with changes_|-
↪fail_with_changes": {
                                "comment": "Failure!",
                                "name": "test fail with changes",
                                "start_time": "21:35:17.326268",
                                "result": false,
                                "duration": 0.509,
                                "__run_num__": 0,
                                "__sls__": "fail_with_changes",
                                "changes": {
                                    "testing": {
                                        "new": "Something pretended to change",
                                        "old": "Unchanged"
                                    }
                                },
                                "__id__": "test fail with changes"
                            }
                        }
                    },
                    "out": "highstate"
                },
                "__id__": "bad_state"
            },
            "salt_|-mymod.myfunc_|-mymod.myfunc_|-function": {
                "comment": "Function ran successfully. Function mymod.myfunc ran on
↪myminion.",
                "name": "mymod.myfunc",
                "start_time": "21:35:17.334373",
                "result": true,
                "duration": 151.716,
                "__run_num__": 2,
                "__jid__": "20171130213517361706",
                "__sls__": "test",
                "changes": {
                    "ret": {
                        "myminion": true
                    },
                    "out": "highstate"
                },
                "__id__": "mymod.myfunc"
            },
            "salt_|-mymod.myfunc_false_result-mymod.myfunc_false_result-function": {
                "comment": "Running function mymod.myfunc_false_result failed on
↪minions: myminion. Function mymod.myfunc_false_result ran on myminion.",
                "name": "mymod.myfunc_false_result",
                "start_time": "21:35:17.486625",
                "result": false,
                "duration": 174.241,
```

(continues on next page)

(continued from previous page)

```
                "__run_num__": 3,
                "__jid__": "20171130213517536270",
                "__sls__": "test",
                "changes": {
                    "ret": {
                        "myminion": false
                    },
                    "out": "highstate"
                },
                "__id__": "mymod.myfunc_false_result"
            }
        }
    },
    "retcode": 1
}
```

The 2018.3.0 release includes a couple fixes to make parsing this data easier and more accurate. The first is the ability to set a *return code* in a custom runner or wheel function, as noted above. The second is a change to how failures are included in the return data. Prior to the 2018.3.0 release, minions that failed a `salt.state` orchestration job would show up in the `comment` field of the return data, in a human-readable string that was not easily parsed. They are now included in the `changes` dictionary alongside the minions that succeeded. In addition, `salt.function` jobs which failed because the *fail function* returned `False` used to handle their failures in the same way `salt.state` jobs did, and this has likewise been corrected.

Running States on the Master without a Minion

The orchestrate runner can be used to execute states on the master without using a minion. For example, assume that `salt://foo.sls` contains the following SLS:

```
/etc/foo.conf:
  file.managed:
    - source: salt://files/foo.conf
    - mode: 0600
```

In this case, running `salt-run state.orchestrate foo` would be the equivalent of running a `state.sls foo`, but it would execute on the master only, and would not require a minion daemon to be running on the master.

This is not technically orchestration, but it can be useful in certain use cases.

Limitations

Only one SLS target can be run at a time using this method, while using *`state.sls`* allows for multiple SLS files to be passed in a comma-separated list.

CHAPTER

ELEVEN

SOLARIS

This section contains details on Solaris specific quirks and workarounds.

Note: Solaris refers to both Solaris 10 comaptible platforms like Solaris 10, illumos, SmartOS, OmniOS, OpenIndiana,... and Oracle Solaris 11 platforms.

11.1 Solaris-specific Behaviour

Salt is capable of managing Solaris systems, however due to various differences between the operating systems, there are some things you need to keep in mind.

This document will contain any quirks that apply across Salt or limitations in some modules.

11.1.1 FQDN/UQDN

On Solaris platforms the FQDN will not always be properly detected. If an IPv6 address is configured pythons `` `socket.getfqdn()` `` fails to return a FQDN and returns the nodename instead. For a full breakdown see the following issue on github: #37027

11.1.2 Grains

Not all grains are available or some have empty or 0 as value. Mostly grains that are depenend on hardware discovery like: - num_gpus - gpus

Also some resolver related grains like: - domain - dns:options - dns:sortlist

CHAPTER

TWELVE

SALT SSH

12.1 Getting Started

Salt SSH is very easy to use, simply set up a basic *roster* file of the systems to connect to and run `salt-ssh` commands in a similar way as standard `salt` commands.

- Salt ssh is considered production ready in version 2014.7.0
- Python is required on the remote system (unless using the `-r` option to send raw ssh commands)
- On many systems, the `salt-ssh` executable will be in its own package, usually named `salt-ssh`
- The Salt SSH system does not supersede the standard Salt communication systems, it simply offers an SSH-based alternative that does not require ZeroMQ and a remote agent. Be aware that since all communication with Salt SSH is executed via SSH it is substantially slower than standard Salt with ZeroMQ.
- At the moment fileserver operations must be wrapped to ensure that the relevant files are delivered with the `salt-ssh` commands. The state module is an exception, which compiles the state run on the master, and in the process finds all the references to `salt://` paths and copies those files down in the same tarball as the state run. However, needed fileserver wrappers are still under development.

12.2 Salt SSH Roster

The roster system in Salt allows for remote minions to be easily defined.

Note: See the *SSH roster docs* for more details.

Simply create the roster file, the default location is */etc/salt/roster*:

```
web1: 192.168.42.1
```

This is a very basic roster file where a Salt ID is being assigned to an IP address. A more elaborate roster can be created:

```
web1:
  host: 192.168.42.1 # The IP addr or DNS hostname
  user: fred         # Remote executions will be executed as user fred
  passwd: foobarbaz  # The password to use for login, if omitted, keys are used
  sudo: True         # Whether to sudo to root, not enabled by default
web2:
  host: 192.168.42.2
```

Note: sudo works only if NOPASSWD is set for user in /etc/sudoers: `fred ALL=(ALL) NOPASSWD: ALL`

12.3 Deploy ssh key for salt-ssh

By default, salt-ssh will generate key pairs for ssh, the default path will be `/etc/salt/pki/master/ssh/salt-ssh.rsa`. The key generation happens when you run `salt-ssh` for the first time.

You can use ssh-copy-id, (the OpenSSH key deployment tool) to deploy keys to your servers.

```
ssh-copy-id -i /etc/salt/pki/master/ssh/salt-ssh.rsa.pub user@server.demo.com
```

One could also create a simple shell script, named salt-ssh-copy-id.sh as follows:

```
#!/bin/bash
if [ -z $1 ]; then
   echo $0 user@host.com
   exit 0
fi
ssh-copy-id -i /etc/salt/pki/master/ssh/salt-ssh.rsa.pub $1
```

Note: Be certain to chmod +x salt-ssh-copy-id.sh.

```
./salt-ssh-copy-id.sh user@server1.host.com
./salt-ssh-copy-id.sh user@server2.host.com
```

Once keys are successfully deployed, salt-ssh can be used to control them.

Alternatively ssh agent forwarding can be used by setting the priv to agent-forwarding.

12.4 Calling Salt SSH

Note: `salt-ssh` on RHEL/CentOS 5

The `salt-ssh` command requires at least python 2.6, which is not installed by default on RHEL/CentOS 5. An easy workaround in this situation is to use the `-r` option to run a raw shell command that installs python26:

```
salt-ssh centos-5-minion -r 'yum -y install epel-release ; yum -y install python26'
```

Note: `salt-ssh` on systems with Python 3.x

Salt, before the 2017.7.0 release, does not support Python 3.x which is the default on for example the popular 16.04 LTS release of Ubuntu. An easy workaround for this scenario is to use the `-r` option similar to the example above:

```
salt-ssh ubuntu-1604-minion -r 'apt update ; apt install -y python-minimal'
```

The `salt-ssh` command can be easily executed in the same way as a salt command:

```
salt-ssh '*' test.version
```

Commands with `salt-ssh` follow the same syntax as the `salt` command.

The standard salt functions are available! The output is the same as `salt` and many of the same flags are available. Please see http://docs.saltstack.com/ref/cli/salt-ssh.html for all of the available options.

12.4.1 Raw Shell Calls

By default `salt-ssh` runs Salt execution modules on the remote system, but `salt-ssh` can also execute raw shell commands:

```
salt-ssh '*' -r 'ifconfig'
```

12.5 States Via Salt SSH

The Salt State system can also be used with `salt-ssh`. The state system abstracts the same interface to the user in `salt-ssh` as it does when using standard `salt`. The intent is that Salt Formulas defined for standard `salt` will work seamlessly with `salt-ssh` and vice-versa.

The standard Salt States walkthroughs function by simply replacing `salt` commands with `salt-ssh`.

12.6 Targeting with Salt SSH

Due to the fact that the targeting approach differs in salt-ssh, only glob and regex targets are supported as of this writing, the remaining target systems still need to be implemented.

Note: By default, Grains are settable through `salt-ssh`. By default, these grains will *not* be persisted across reboots.

See the "thin_dir" setting in *Roster documentation* for more details.

12.7 Configuring Salt SSH

Salt SSH takes its configuration from a master configuration file. Normally, this file is in `/etc/salt/master`. If one wishes to use a customized configuration file, the `-c` option to Salt SSH facilitates passing in a directory to look inside for a configuration file named `master`.

12.7.1 Minion Config

New in version 2015.5.1.

Minion config options can be defined globally using the master configuration option `ssh_minion_opts`. It can also be defined on a per-minion basis with the `minion_opts` entry in the roster.

12.8 Running Salt SSH as non-root user

By default, Salt read all the configuration from /etc/salt/. If you are running Salt SSH with a regular user you have to modify some paths or you will get "Permission denied" messages. You have to modify two parameters: `pki_dir` and `cachedir`. Those should point to a full path writable for the user.

It's recommended not to modify /etc/salt for this purpose. Create a private copy of /etc/salt for the user and run the command with `-c /new/config/path`.

12.9 Define CLI Options with Saltfile

If you are commonly passing in CLI options to `salt-ssh`, you can create a `Saltfile` to automatically use these options. This is common if you're managing several different salt projects on the same server.

So you can `cd` into a directory that has a `Saltfile` with the following YAML contents:

```
salt-ssh:
  config_dir: path/to/config/dir
  ssh_log_file: salt-ssh.log
  ssh_max_procs: 30
  ssh_wipe: True
```

Instead of having to call `salt-ssh --config-dir=path/to/config/dir --max-procs=30 --wipe \* test.version` you can call `salt-ssh \* test.version`.

Boolean-style options should be specified in their YAML representation.

Note: The option keys specified must match the destination attributes for the options specified in the parser `salt.utils.parsers.SaltSSHOptionParser`. For example, in the case of the `--wipe` command line option, its `dest` is configured to be `ssh_wipe` and thus this is what should be configured in the `Saltfile`. Using the names of flags for this option, being `wipe: True` or `w: True`, will not work.

Note: For the *Saltfile* to be automatically detected it needs to be named *Saltfile* with a capital *S* and be readable by the user running salt-ssh.

At last you can create `~/.salt/Saltfile` and `salt-ssh` will automatically load it by default.

12.10 Debugging salt-ssh

One common approach for debugging `salt-ssh` is to simply use the tarball that salt ships to the remote machine and call `salt-call` directly.

To determine the location of `salt-call`, simply run `salt-ssh` with the `-ltrace` flag and look for a line containing the string, SALT_ARGV. This contains the `salt-call` command that `salt-ssh` attempted to execute.

It is recommended that one modify this command a bit by removing the `-l quiet`, `--metadata` and `--output json` to get a better idea of what's going on the target system.

12.10.1 Salt Rosters

Salt rosters are pluggable systems added in Salt 0.17.0 to facilitate the `salt-ssh` system. The roster system was created because `salt-ssh` needs a means to identify which systems need to be targeted for execution.

See also:

roster modules

Note: The Roster System is not needed or used in standard Salt because the master does not need to be initially aware of target systems, since the Salt Minion checks itself into the master.

Since the roster system is pluggable, it can be easily augmented to attach to any existing systems to gather information about what servers are presently available and should be attached to by `salt-ssh`. By default the roster file is located at /etc/salt/roster.

How Rosters Work

The roster system compiles a data structure internally referred to as `targets`. The `targets` is a list of target systems and attributes about how to connect to said systems. The only requirement for a roster module in Salt is to return the `targets` data structure.

Targets Data

The information which can be stored in a roster `target` is the following:

```
<Salt ID>:       # The id to reference the target system with
    host:        # The IP address or DNS name of the remote host
    user:        # The user to log in as
    passwd:      # The password to log in with

    # Optional parameters
    port:        # The target system's ssh port number
    sudo:        # Boolean to run command via sudo
    sudo_user:   # Str: Set this to execute Salt as a sudo user other than root.
                 # This user must be in the same system group as the remote user
                 # that is used to login and is specified above. Alternatively,
                 # the user must be a super-user.
    tty:         # Boolean: Set this option to True if sudo is also set to
                 # True and requiretty is also set on the target system
    priv:        # File path to ssh private key, defaults to salt-ssh.rsa
                 # The priv can also be set to agent-forwarding to not specify
                 # a key, but use ssh agent forwarding
    priv_passwd: # Passphrase for ssh private key
    timeout:     # Number of seconds to wait for response when establishing
                 # an SSH connection
    minion_opts: # Dictionary of minion opts
    thin_dir:    # The target system's storage directory for Salt
                 # components. Defaults to /tmp/salt-<hash>.
    cmd_umask:   # umask to enforce for the salt-call command. Should be in
                 # octal (so for 0o077 in YAML you would do 0077, or 63)
```

Target Defaults

The *roster_defaults* dictionary in the master config is used to set the default login variables for minions in the roster so that the same arguments do not need to be passed with commandline arguments.

```
roster_defaults:
  user: daniel
  sudo: True
  priv: /root/.ssh/id_rsa
  tty: True
```

thin_dir

Salt needs to upload a standalone environment to the target system, and this defaults to /tmp/salt-<hash>. This directory will be cleaned up per normal systems operation.

If you need a persistent Salt environment, for instance to set persistent grains, this value will need to be changed.

CHAPTER

THIRTEEN

THORIUM COMPLEX REACTOR

The original Salt Reactor is based on the idea of listening for a specific event and then reacting to it. This model comes with many logical limitations, for instance it is very difficult (and hacky) to fire a reaction based on aggregate data or based on multiple events.

The Thorium reactor is intended to alleviate this problem in a very elegant way. Instead of using extensive jinja routines or complex python sls files the aggregation of data and the determination of what should run becomes isolated to the sls data logic, makes the definitions much cleaner.

13.1 Starting the Thorium Engine

To enable the thorium engine add the following configuration to the engines section of your Salt Master or Minion configuration file and restart the daemon:

```
engines:
  - thorium: {}
```

13.2 Thorium Modules

Because of its specialized nature, Thorium uses its own set of modules. However, many of these modules are designed to wrap the more commonly-used Salt subsystems. These modules are:

- local: Execution modules
- runner: Runner modules
- wheel: Wheel modules

There are other modules that ship with Thorium as well. Some of these will be highlighted later in this document.

13.3 Writing Thorium Formulas

Like some other Salt subsystems, Thorium uses its own directory structure. The default location for this structure is `/srv/thorium/`, but it can be changed using the `thorium_roots` setting in the `master` configuration file.

This would explicitly set the roots to the default:

```
thorium_roots:
  base:
    - /srv/thorium
```

Example `thorium_roots` configuration:

```
thorium_roots:
  base:
    - /etc/salt/thorium
```

It is also possible to use gitfs with Thorium, using the `thoriumenv` or `thorium_top` settings.

Example using `thorium_top`:

```
thorium_top: salt://thorium/top.sls
gitfs_provider: pygit2

gitfs_remotes:
  - git@github.com:user/repo.git:
    - name: salt-backend
    - root: salt
    - base: master
  - git@github.com:user/repo.git:
    - name: thorium-backend
    - root: thorium
    - base: master
    - mountpoint: salt://thorium
```

Note: When using this method don't forget to prepend the mountpoint to files served by this repo, for example `top.sls`:

```
base:
  '*':
    - thorium.key_clean
```

Example using `thoriumenv`:

```
thoriumenv: thorium
gitfs_provider: pygit2

gitfs_remotes:
  - git@github.com:user/repo.git:
    - name: salt-backend
    - root: salt
    - base: master
  - git@github.com:user/repo.git:
    - name: thorium-backend
    - root: thorium
    - saltenv:
      - thorium:
        - ref: master
```

Note: When using this method all state will run under the defined environment, for example `top.sls`:

```
thorium:
  '*':
    - key_clean
```

13.3.1 The Thorium top.sls File

Thorium uses its own `top.sls` file, which follows the same convention as is found in `/srv/salt/`:

```
<srv>:
  <target>:
    - <formula 1>
    - <formula 2>
    - <etc...>
```

For instance, a `top.sls` using a standard `base` environment and a single Thorium formula called `key_clean`, would look like:

```
base:
  '*':
    - key_clean
```

Take note that the target in a Thorium `top.sls` is not used; it only exists to follow the same convention as other `top.sls` files. Leave this set to `'*'` in your own Thorium `top.sls`.

13.3.2 Thorium Formula Files

Thorium SLS files are processed by the same state compiler that processes Salt state files. This means that features like requisites, templates, and so on are available.

Let's take a look at an example, and then discuss each component of it. This formula uses Thorium to detect when a minion has disappeared and then deletes the key from the master when the minion has been gone for 60 seconds:

```
statreg:
  status.reg

keydel:
  key.timeout:
    - delete: 60
    - require:
      - status: statreg
```

There are two stanzas in this formula, whose IDs are `statreg` and `keydel`. The first stanza, `statreg`, tells Thorium to keep track of minion status beacons in its *register*. We'll talk more about the register in a moment.

The second stanza, `keydel`, is the one that does the real work. It uses the `key` module to apply an expiration (using the `timeout` function) to a minion. Because `delete` is set to `60`, this is a 60 second expiration. If a minion does not check in at least once every 60 seconds, its key will be deleted from the master. This particular function also allows you to use `reject` instead of `delete`, allowing for a minion to be rejected instead of deleted if it does not check in within the specified time period.

There is also a `require` requisite in this stanza. It states that the `key.timeout` function will not be called unless the `status.reg` function in the `statreg` codeblock has been successfully called first.

13.3.3 Thorium Links to Beacons

The above example was added in the 2016.11.0 release of Salt and makes use of the `status` beacon also added in the 2016.11.0 release. For the above Thorium state to function properly you will also need to enable the `status` beacon in the `minion` configuration file:

```
beacons:
  status:
    - interval: 10
```

This will cause the minion to use the status beacon to check in with the master every 10 seconds.

13.4 The Thorium Register

In order to keep track of information, Thorium uses an in-memory register (or rather, collection of registers) on the master. These registers are only populated when told to by a formula, and they normally will be erased when the master is restarted. It is possible to persist the registers to disk, but we'll get to that in a moment.

The example above uses `status.reg` to populate a register for you, which is automatically used by the `key.timeout` function. However, you can set your own register values as well, using the `reg` module.

Because Thorium watches the event bus, the `reg` module is designed to look for user-specified tags, and then extract data from the payload of events that match those tags. For instance, the following stanza will look for an event with a tag of `my/custom/event`:

```
foo:
  reg.list:
    - add: bar
    - match: my/custom/event
```

When such an event is found, the data found in the payload dictionary key of `bar` will be stored in a register called `foo`. This register will store that data in a `list`. You may also use `reg.set` to add data to a `set()` instead.

If you would like to see a copy of the register as it is stored in memory, you can use the `file.save` function:

```
myreg:
  file.save
```

In this case, each time the register is updated, a copy will be saved in JSON format at `/var/cache/salt/master/thorium/saves/myreg`. If you would like to see when particular events are added to a list-type register, you may add a `stamp` option to `reg.list` (but not `reg.set`). With the above two stanzas put together, this would look like:

```
foo:
  reg.list:
    - add: bar
    - match: my/custom/event
    - stamp: True

myreg:
  file.save
```

If you would like to only keep a certain number of the most recent register entries, you may also add a `prune` option to `reg.list` (but not `reg.set`):

```
foo:
  reg.list:
    - add: bar
    - match: my/custom/event
    - stamp: True
    - prune: 50
```

This example will only keep the 50 most recent entries in the `foo` register.

13.4.1 Using Register Data

Putting data in a register is useless if you don't do anything with it. The `check` module is designed to examine register data and determine whether it matches the given parameters. For instance, the `check.contains` function will return `True` if the given `value` is contained in the specified register:

```
foo:
  reg.list:
    - add: bar
    - match: my/custom/event
    - stamp: True
    - prune: 50
  check.contains:
    - value: somedata
```

Used with a `require` requisite, we can call one of the wrapper modules and perform an operation. For example:

```
shell_test:
  local.cmd:
    - tgt: dufresne
    - func: cmd.run
    - arg:
      - echo 'thorium success' > /tmp/thorium.txt
    - require:
      - check: foo
```

This stanza will only run if the `check.contains` function under the `foo` ID returns true (meaning the match was found).

There are a number of other functions in the `check` module which use different means of comparing values:

- `gt`: Check whether the register entry is greater than the given value
- `gte`: Check whether the register entry is greater than or equal to the given value
- `lt`: Check whether the register entry is less than the given value
- `lte`: Check whether the register entry is less than or equal to the given value
- `eq`: Check whether the register entry is equal to the given value
- `ne`: Check whether the register entry is not equal to the given value

There is also a function called `check.event` which does not examine the register. Instead, it looks directly at an event as it is coming in on the event bus, and returns `True` if that event's tag matches. For example:

```
salt/foo/*/bar:
  check.event

run_remote_ex:
  local.cmd:
    - tgt: '*'
    - func: test.version
    - require:
      - check: salt/foo/*/bar
```

This formula will look for an event whose tag is `salt/foo/<anything>/bar` and if it comes in, issue a `test.version` to all minions.

13.4.2 Register Persistence

It is possible to persist the register data to disk when a master is stopped gracefully, and reload it from disk when the master starts up again. This functionality is provided by the returner subsystem, and is enabled whenever any returner containing a `load_reg` and a `save_reg` function is used.

CHAPTER

FOURTEEN

SALT CLOUD

14.1 Configuration

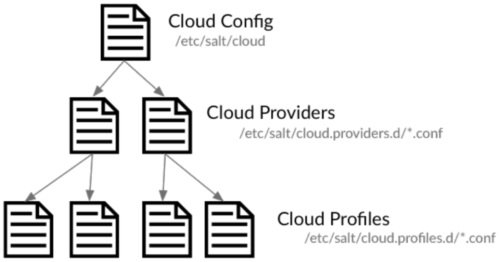
Salt Cloud provides a powerful interface to interact with cloud hosts. This interface is tightly integrated with Salt, and new virtual machines are automatically connected to your Salt master after creation.

Since Salt Cloud is designed to be an automated system, most configuration is done using the following YAML configuration files:

- `/etc/salt/cloud`: The main configuration file, contains global settings that apply to all cloud hosts. See *Salt Cloud Configuration*.
- `/etc/salt/cloud.providers.d/*.conf`: Contains settings that configure a specific cloud host, such as credentials, region settings, and so on. Since configuration varies significantly between each cloud host, a separate file should be created for each cloud host. In Salt Cloud, a provider is synonymous with a cloud host (Amazon EC2, Google Compute Engine, Rackspace, and so on). See *Provider Specifics*.
- `/etc/salt/cloud.profiles.d/*.conf`: Contains settings that define a specific VM type. A profile defines the systems specs and image, and any other settings that are specific to this VM type. Each specific VM type is called a profile, and multiple profiles can be defined in a profile file. Each profile references a parent provider that defines the cloud host in which the VM is created (the provider settings are in the provider configuration explained above). Based on your needs, you might define different profiles for web servers, database servers, and so on. See *VM Profiles*.

14.2 Configuration Inheritance

Configuration settings are inherited in order from the cloud config => providers => profile.


Cloud Config
/etc/salt/cloud
Cloud Providers
/etc/salt/cloud.providers.d/*.conf
Cloud Profiles
/etc/salt/cloud.profiles.d/*.conf

For example, if you wanted to use the same image for all virtual machines for a specific provider, the image name could be placed in the provider file. This value is inherited by all profiles that use that provider, but is overridden if a image name is defined in the profile.

Most configuration settings can be defined in any file, the main difference being how that setting is inherited.

14.3 QuickStart

The *Salt Cloud Quickstart* walks you through defining a provider, a VM profile, and shows you how to create virtual machines using Salt Cloud.

Note that if you installed Salt via Salt Bootstrap, it may not have automatically installed salt-cloud for you. Use your distribution's package manager to install the `salt-cloud` package from the same repo that you used to install Salt. These repos will automatically be setup by Salt Bootstrap.

Alternatively, the `-L` option can be passed to the Salt Bootstrap script when installing Salt. The `-L` option will install `salt-cloud` and the required `libcloud` package.

14.4 Using Salt Cloud

14.4.1 salt-cloud

Provision virtual machines in the cloud with Salt

Synopsis

```
salt-cloud -m /etc/salt/cloud.map

salt-cloud -m /etc/salt/cloud.map NAME

salt-cloud -m /etc/salt/cloud.map NAME1 NAME2

salt-cloud -p PROFILE NAME

salt-cloud -p PROFILE NAME1 NAME2 NAME3 NAME4 NAME5 NAME6
```

Description

Salt Cloud is the system used to provision virtual machines on various public clouds via a cleanly controlled profile and mapping system.

Options

`--version`
Print the version of Salt that is running.

`--versions-report`
Show program's dependencies and version number, and then exit

`-h, --help`
Show the help message and exit

`-c` `CONFIG_DIR`, **`--config-dir`**`=CONFIG_dir`
The location of the Salt configuration directory. This directory contains the configuration files for Salt master and minions. The default location on most systems is `/etc/salt`.

Execution Options

-L LOCATION, **--location**=LOCATION
Specify which region to connect to.

-a ACTION, **--action**=ACTION
Perform an action that may be specific to this cloud provider. This argument requires one or more instance names to be specified.

-f <FUNC-NAME> <PROVIDER>, **--function**=<FUNC-NAME> <PROVIDER>
Perform an function that may be specific to this cloud provider, that does not apply to an instance. This argument requires a provider to be specified (i.e.: nova).

-p PROFILE, **--profile**=PROFILE
Select a single profile to build the named cloud VMs from. The profile must be defined in the specified profiles file.

-m MAP, **--map**=MAP
Specify a map file to use. If used without any other options, this option will ensure that all of the mapped VMs are created. If the named VM already exists then it will be skipped.

-H, **--hard**
When specifying a map file, the default behavior is to ensure that all of the VMs specified in the map file are created. If the –hard option is set, then any VMs that exist on configured cloud providers that are not specified in the map file will be destroyed. Be advised that this can be a destructive operation and should be used with care.

-d, **--destroy**
Pass in the name(s) of VMs to destroy, salt-cloud will search the configured cloud providers for the specified names and destroy the VMs. Be advised that this is a destructive operation and should be used with care. Can be used in conjunction with the -m option to specify a map of VMs to be deleted.

-P, **--parallel**
Normally when building many cloud VMs they are executed serially. The -P option will run each cloud vm build in a separate process allowing for large groups of VMs to be build at once.

Be advised that some cloud provider's systems don't seem to be well suited for this influx of vm creation. When creating large groups of VMs watch the cloud provider carefully.

-u, **--update-bootstrap**
Update salt-bootstrap to the latest stable bootstrap release.

-y, **--assume-yes**
Default yes in answer to all confirmation questions.

-k, **--keep-tmp**
Do not remove files from /tmp/ after deploy.sh finishes.

--show-deploy-args
Include the options used to deploy the minion in the data returned.

--script-args=SCRIPT_ARGS
Script arguments to be fed to the bootstrap script when deploying the VM.

Query Options

-Q, **--query**
Execute a query and return some information about the nodes running on configured cloud providers

-F, --full-query
Execute a query and print out all available information about all cloud VMs. Can be used in conjunction with -m to display only information about the specified map.

-S, --select-query
Execute a query and print out selected information about all cloud VMs. Can be used in conjunction with -m to display only information about the specified map.

--list-providers
Display a list of configured providers.

--list-profiles
New in version 2014.7.0.

Display a list of configured profiles. Pass in a cloud provider to view the provider's associated profiles, such as `digitalocean`, or pass in `all` to list all the configured profiles.

Cloud Providers Listings

--list-locations=`LIST_LOCATIONS`
Display a list of locations available in configured cloud providers. Pass the cloud provider that available locations are desired on, such as ”linode”, or pass ”all” to list locations for all configured cloud providers

--list-images=`LIST_IMAGES`
Display a list of images available in configured cloud providers. Pass the cloud provider that available images are desired on, such as ”linode”, or pass ”all” to list images for all configured cloud providers

--list-sizes=`LIST_SIZES`
Display a list of sizes available in configured cloud providers. Pass the cloud provider that available sizes are desired on, such as ”AWS”, or pass ”all” to list sizes for all configured cloud providers

Cloud Credentials

--set-password=`<USERNAME> <PROVIDER>`
Configure password for a cloud provider and save it to the keyring. PROVIDER can be specified with or without a driver, for example: ”–set-password bob rackspace” or more specific ”–set-password bob rackspace:openstack” DEPRECATED!

Output Options

--out
Pass in an alternative outputter to display the return of data. This outputter can be any of the available outputters:

> `highstate`, `json`, `key`, `overstatestage`, `pprint`, `raw`, `txt`, `yaml`, and *many others*.

Some outputters are formatted only for data returned from specific functions. If an outputter is used that does not support the data passed into it, then Salt will fall back on the `pprint` outputter and display the return data using the Python `pprint` standard library module.

--out-indent `OUTPUT_INDENT`, **--output-indent** `OUTPUT_INDENT`
Print the output indented by the provided value in spaces. Negative values disable indentation. Only applicable in outputters that support indentation.

--out-file=`OUTPUT_FILE`, **--output-file**=`OUTPUT_FILE`
Write the output to the specified file.

--out-file-append, --output-file-append
Append the output to the specified file.

--no-color
Disable all colored output

--force-color
Force colored output

Note: When using colored output the color codes are as follows:

`green` denotes success, `red` denotes failure, `blue` denotes changes and success and `yellow` denotes a expected future change in configuration.

--state-output=STATE_OUTPUT, **--state_output**=STATE_OUTPUT
Override the configured state_output value for minion output. One of 'full', 'terse', 'mixed', 'changes' or 'filter'. Default: 'none'.

--state-verbose=STATE_VERBOSE, **--state_verbose**=STATE_VERBOSE
Override the configured state_verbose value for minion output. Set to True or False. Default: none.

Examples

To create 4 VMs named web1, web2, db1, and db2 from specified profiles:

```
salt-cloud -p fedora_rackspace web1 web2 db1 db2
```

To read in a map file and create all VMs specified therein:

```
salt-cloud -m /path/to/cloud.map
```

To read in a map file and create all VMs specified therein in parallel:

```
salt-cloud -m /path/to/cloud.map -P
```

To delete any VMs specified in the map file:

```
salt-cloud -m /path/to/cloud.map -d
```

To delete any VMs NOT specified in the map file:

```
salt-cloud -m /path/to/cloud.map -H
```

To display the status of all VMs specified in the map file:

```
salt-cloud -m /path/to/cloud.map -Q
```

See also

salt-cloud(7) *salt(7)* *salt-master(1)* *salt-minion(1)*

14.4.2 Salt Cloud basic usage

Salt Cloud needs, at least, one configured *Provider* and *Profile* to be functional.

Creating a VM

To create a VM with salt cloud, use command:

```
salt-cloud -p <profile> name_of_vm
```

Assuming there is a profile configured as following:

```
fedora_rackspace:
    provider: my-rackspace-config
    image: Fedora 17
    size: 256 server
    script: bootstrap-salt
```

Then, the command to create new VM named `fedora_http_01` is:

```
salt-cloud -p fedora_rackspace fedora_http_01
```

Destroying a VM

To destroy a created-by-salt-cloud VM, use command:

```
salt-cloud -d name_of_vm
```

For example, to delete the VM created on above example, use:

```
salt-cloud -d fedora_http_01
```

14.4.3 VM Profiles

Salt cloud designates virtual machines inside the profile configuration file. The profile configuration file defaults to `/etc/salt/cloud.profiles` and is a yaml configuration. The syntax for declaring profiles is simple:

```
fedora_rackspace:
    provider: my-rackspace-config
    image: Fedora 17
    size: 256 server
    script: bootstrap-salt
```

It should be noted that the `script` option defaults to `bootstrap-salt`, and does not normally need to be specified. Further examples in this document will not show the `script` option.

A few key pieces of information need to be declared and can change based on the cloud provider. A number of additional parameters can also be inserted:

```
centos_rackspace:
  provider: my-rackspace-config
  image: CentOS 6.2
  size: 1024 server
  minion:
    master: salt.example.com
    append_domain: webs.example.com
    grains:
      role: webserver
```

The image must be selected from available images. Similarly, sizes must be selected from the list of sizes. To get a list of available images and sizes use the following command:

```
salt-cloud --list-images openstack
salt-cloud --list-sizes openstack
```

Some parameters can be specified in the main Salt cloud configuration file and then are applied to all cloud profiles. For instance if only a single cloud provider is being used then the provider option can be declared in the Salt cloud configuration file.

Multiple Configuration Files

In addition to `/etc/salt/cloud.profiles`, profiles can also be specified in any file matching `cloud.profiles.d/*conf` which is a sub-directory relative to the profiles configuration file(with the above configuration file as an example, `/etc/salt/cloud.profiles.d/*.conf`). This allows for more extensible configuration, and plays nicely with various configuration management tools as well as version control systems.

Larger Example

```
rhel_ec2:
  provider: my-ec2-config
  image: ami-e565ba8c
  size: t1.micro
  minion:
    cheese: edam

ubuntu_ec2:
  provider: my-ec2-config
  image: ami-7e2da54e
  size: t1.micro
  minion:
    cheese: edam

ubuntu_rackspace:
  provider: my-rackspace-config
  image: Ubuntu 12.04 LTS
  size: 256 server
  minion:
    cheese: edam

fedora_rackspace:
  provider: my-rackspace-config
  image: Fedora 17
  size: 256 server
  minion:
    cheese: edam

cent_linode:
  provider: my-linode-config
  image: CentOS 6.2 64bit
  size: Linode 512

cent_gogrid:
  provider: my-gogrid-config
  image: 12834
```

(continues on next page)

(continued from previous page)

```
  size: 512MB

cent_joyent:
  provider: my-joyent-config
  image: centos-7
  size: g4-highram-16G
```

14.4.4 Cloud Map File

A number of options exist when creating virtual machines. They can be managed directly from profiles and the command line execution, or a more complex map file can be created. The map file allows for a number of virtual machines to be created and associated with specific profiles. The map file is designed to be run once to create these more complex scenarios using salt-cloud.

Map files have a simple format, specify a profile and then a list of virtual machines to make from said profile:

```
fedora_small:
  - web1
  - web2
  - web3
fedora_high:
  - redis1
  - redis2
  - redis3
cent_high:
  - riak1
  - riak2
  - riak3
```

This map file can then be called to roll out all of these virtual machines. Map files are called from the salt-cloud command with the -m option:

```
$ salt-cloud -m /path/to/mapfile
```

Remember, that as with direct profile provisioning the -P option can be passed to create the virtual machines in parallel:

```
$ salt-cloud -m /path/to/mapfile -P
```

Note: Due to limitations in the GoGrid API, instances cannot be provisioned in parallel with the GoGrid driver. Map files will work with GoGrid, but the `-P` argument should not be used on maps referencing GoGrid instances.

A map file can also be enforced to represent the total state of a cloud deployment by using the `--hard` option. When using the hard option any vms that exist but are not specified in the map file will be destroyed:

```
$ salt-cloud -m /path/to/mapfile -P -H
```

Be careful with this argument, it is very dangerous! In fact, it is so dangerous that in order to use it, you must explicitly enable it in the main configuration file.

```
enable_hard_maps: True
```

A map file can include grains and minion configuration options:

```
fedora_small:
  - web1:
      minion:
        log_level: debug
      grains:
        cheese: tasty
        omelet: du fromage
  - web2:
      minion:
        log_level: warn
      grains:
        cheese: more tasty
        omelet: with peppers
```

Any top level data element from your profile may be overridden in the map file:

```
fedora_small:
  - web1:
      size: t2.micro
  - web2:
      size: t2.nano
```

As of Salt 2017.7.0, nested elements are merged, and can can be specified individually without having to repeat the complete definition for each top level data element. In this example a separate MAC is assigned to each VMware instance while inheriting device parameters for for disk and network configuration:

```
nyc-vm:
  - db1:
      devices:
        network:
          Network Adapter 1:
            mac: '44:44:44:44:44:41'
  - db2:
      devices:
        network:
          Network Adapter 1:
            mac: '44:44:44:44:44:42'
```

A map file may also be used with the various query options:

```
$ salt-cloud -m /path/to/mapfile -Q
{'ec2': {'web1': {'id': 'i-e6aqfegb',
                     'image': None,
                     'private_ips': [],
                     'public_ips': [],
                     'size': None,
                     'state': 0}},
         'web2': {'Absent'}}
```

...or with the delete option:

```
$ salt-cloud -m /path/to/mapfile -d
The following virtual machines are set to be destroyed:
  web1
  web2

Proceed? [N/y]
```

> **Warning:** Specifying Nodes with Maps on the Command Line Specifying the name of a node or nodes with the maps options on the command line is *not* supported. This is especially important to remember when using `--destroy` with maps; `salt-cloud` will ignore any arguments passed in which are not directly relevant to the map file. *When using ``–destroy`` with a map, every node in the map file will be deleted!* Maps don't provide any useful information for destroying individual nodes, and should not be used to destroy a subset of a map.

Setting up New Salt Masters

Bootstrapping a new master in the map is as simple as:

```
fedora_small:
  - web1:
      make_master: True
  - web2
  - web3
```

Notice that **ALL** bootstrapped minions from the map will answer to the newly created salt-master.

To make any of the bootstrapped minions answer to the bootstrapping salt-master as opposed to the newly created salt-master, as an example:

```
fedora_small:
  - web1:
      make_master: True
      minion:
        master: <the local master ip address>
        local_master: True
  - web2
  - web3
```

The above says the minion running on the newly created salt-master responds to the local master, ie, the master used to bootstrap these VMs.

Another example:

```
fedora_small:
  - web1:
      make_master: True
  - web2
  - web3:
      minion:
        master: <the local master ip address>
        local_master: True
```

The above example makes the `web3` minion answer to the local master, not the newly created master.

Using Direct Map Data

When using modules that access the `CloudClient` directly (notably, the `cloud` execution and runner modules), it is possible to pass in the contents of a map file, rather than a path to the location of the map file.

Normally when using these modules, the path to the map file is passed in using:

```
salt-run cloud.map_run /path/to/cloud.map
```

To pass in the actual map data, use the `map_data` argument:

```
salt-run cloud.map_run map_data='{"centos7": [{"saltmaster": {"minion": \
    {"transport": "tcp"}, "make_master": true, "master": {"transport": \
    "tcp"}}}, {"minion001": {"minion": {"transport": "tcp"}}}]}'
```

14.4.5 Cloud Actions

Once a VM has been created, there are a number of actions that can be performed on it. The "reboot" action can be used across all providers, but all other actions are specific to the cloud provider. In order to perform an action, you may specify it from the command line, including the name(s) of the VM to perform the action on:

```
$ salt-cloud -a reboot vm_name
$ salt-cloud -a reboot vm1 vm2 vm2
```

Or you may specify a map which includes all VMs to perform the action on:

```
$ salt-cloud -a reboot -m /path/to/mapfile
```

The following is an example list of actions currently supported by `salt-cloud`:

```
all providers:
    - reboot
ec2:
    - start
    - stop
joyent:
    - stop
linode:
    - start
    - stop
```

Another useful reference for viewing more `salt-cloud` actions is the *Salt Cloud Feature Matrix*.

14.4.6 Cloud Functions

Cloud functions work much the same way as cloud actions, except that they don't perform an operation on a specific instance, and so do not need a machine name to be specified. However, since they perform an operation on a specific cloud provider, that provider must be specified.

```
$ salt-cloud -f show_image ec2 image=ami-fd20ad94
```

There are three universal salt-cloud functions that are extremely useful for gathering information about instances on a provider basis:

- `list_nodes`: Returns some general information about the instances for the given provider.
- `list_nodes_full`: Returns all information about the instances for the given provider.
- `list_nodes_select`: Returns select information about the instances for the given provider.

```
$ salt-cloud -f list_nodes linode
$ salt-cloud -f list_nodes_full linode
$ salt-cloud -f list_nodes_select linode
```

Another useful reference for viewing `salt-cloud` functions is the *Salt Cloud Feature Matrix*.

14.5 Core Configuration

14.5.1 Install Salt Cloud

Salt Cloud is now part of Salt proper. It was merged in as of *Salt version 2014.1.0*.

On Ubuntu, install Salt Cloud by using following command:

```
sudo add-apt-repository ppa:saltstack/salt
sudo apt-get update
sudo apt-get install salt-cloud
```

If using Salt Cloud on macOS, `curl-ca-bundle` must be installed. Presently, this package is not available via `brew`, but it is available using MacPorts:

```
sudo port install curl-ca-bundle
```

Salt Cloud depends on `apache-libcloud`. Libcloud can be installed via pip with `pip install apache-libcloud`.

Installing Salt Cloud for development

Installing Salt for development enables Salt Cloud development as well, just make sure `apache-libcloud` is installed as per above paragraph.

See these instructions: *Installing Salt for development*.

14.5.2 Core Configuration

A number of core configuration options and some options that are global to the VM profiles can be set in the cloud configuration file. By default this file is located at `/etc/salt/cloud`.

Thread Pool Size

When salt cloud is operating in parallel mode via the `-P` argument, you can control the thread pool size by specifying the `pool_size` parameter with a positive integer value.

By default, the thread pool size will be set to the number of VMs that salt cloud is operating on.

```
pool_size: 10
```

Minion Configuration

The default minion configuration is set up in this file. Minions created by salt-cloud derive their configuration from this file. Almost all parameters found in *Configuring the Salt Minion* can be used here.

```
minion:
  master: saltmaster.example.com
```

In particular, this is the location to specify the location of the salt master and its listening port, if the port is not set to the default.

Similar to most other settings, Minion configuration settings are inherited across configuration files. For example, the master setting might be contained in the main `cloud` configuration file as demonstrated above, but additional settings can be placed in the provider, profile or map configuration files:

```
ec2-web:
  size: t1.micro
  minion:
    environment: test
    startup_states: sls
    sls_list:
      - web
```

When salt cloud creates a new minon, it can automatically add grain information to the minion configuration file identifying the sources originally used to define it.

The generated grain information will appear similar to:

```
grains:
  salt-cloud:
    driver: ec2
    provider: my_ec2:ec2
    profile: ec2-web
```

The generation of the salt-cloud grain can be suppressed by the option `enable_cloud_grains: 'False'` in the cloud configuration file.

Cloud Configuration Syntax

The data specific to interacting with public clouds is set up *here*.

Cloud provider configuration settings can live in several places. The first is in `/etc/salt/cloud`:

```
# /etc/salt/cloud
providers:
  my-aws-migrated-config:
    id: HJGRYCILJLKJYG
    key: 'kdjgfsgm;woormgl/aserigjksjdhasdfgn'
    keyname: test
    securitygroup: quick-start
    private_key: /root/test.pem
    driver: ec2
```

Cloud provider configuration data can also be housed in `/etc/salt/cloud.providers` or any file matching `/etc/salt/cloud.providers.d/*.conf`. All files in any of these locations will be parsed for cloud provider data.

Using the example configuration above:

```
# /etc/salt/cloud.providers
# or could be /etc/salt/cloud.providers.d/*.conf
my-aws-config:
  id: HJGRYCILJLKJYG
  key: 'kdjgfsgm;woormgl/aserigjksjdhasdfgn'
  keyname: test
  securitygroup: quick-start
  private_key: /root/test.pem
  driver: ec2
```

Note: Salt Cloud provider configurations within `/etc/cloud.provider.d/` should not specify the `providers` starting key.

It is also possible to have multiple cloud configuration blocks within the same alias block. For example:

```
production-config:
  - id: HJGRYCILJLKJYG
    key: 'kdjgfsgm;woormgl/aserigjksjdhasdfgn'
    keyname: test
    securitygroup: quick-start
    private_key: /root/test.pem
    driver: ec2

  - user: example_user
    apikey: 123984bjjas87034
    driver: rackspace
```

However, using this configuration method requires a change with profile configuration blocks. The provider alias needs to have the provider key value appended as in the following example:

```
rhel_aws_dev:
  provider: production-config:ec2
  image: ami-e565ba8c
  size: t1.micro

rhel_aws_prod:
  provider: production-config:ec2
  image: ami-e565ba8c
  size: High-CPU Extra Large Instance

database_prod:
  provider: production-config:rackspace
  image: Ubuntu 12.04 LTS
  size: 256 server
```

Notice that because of the multiple entries, one has to be explicit about the provider alias and name, from the above example, `production-config: ec2`.

This data interactions with the `salt-cloud` binary regarding its `--list-location`, `--list-images`, and `--list-sizes` which needs a cloud provider as an argument. The argument used should be the configured cloud provider alias. If the provider alias has multiple entries, `<provider-alias>: <provider-name>` should be used.

To allow for a more extensible configuration, `--providers-config`, which defaults to `/etc/salt/cloud.providers`, was added to the cli parser. It allows for the providers' configuration to be added on a per-file basis.

Pillar Configuration

It is possible to configure cloud providers using pillars. This is only used when inside the cloud module. You can setup a variable called `cloud` that contains your profile, provider, and map to pass that information to the cloud servers instead of having to copy the full configuration to every minion. In your pillar file, you would use something like this:

```
cloud:
  ssh_key_name: saltstack
```

(continues on next page)

(continued from previous page)

```
  ssh_key_file: /root/.ssh/id_rsa
  update_cachedir: True
  diff_cache_events: True

  providers:
    my-openstack:
      driver: openstack
      region_name: ORD
      cloud: mycloud

  profiles:
    ubuntu-openstack:
      provider: my-openstack
      size: ds512M
      image: CentOS 7
      script_args: git develop

  maps:
    my-dev-map:
      ubuntu-openstack:
        - dev-test01
        - dev-test02
        - dev-test03
        - dev-test04
    my-prd-map:
      ubuntu-openstack:
        - prd-web01
        - prd-web02
            minion:
              id: custom-minion-id-app1-stack1-frontend
            grains:
              roles:
                - webserver
              deployment: datacenter4-openstack
        - prod-db01
        - prod-db02
```

Cloud Configurations

Scaleway

To use Salt Cloud with Scaleway, you need to get an `access key` and an `API token`. `API tokens` are unique identifiers associated with your Scaleway account. To retrieve your `access key` and `API token`, log-in to the Scaleway control panel, open the pull-down menu on your account name and click on "My Credentials" link.

If you do not have `API token` you can create one by clicking the "Create New Token" button on the right corner.

```
my-scaleway-config:
  access_key: 15cf404d-4560-41b1-9a0c-21c3d5c4ff1f
  token: a7347ec8-5de1-4024-a5e3-24b77d1ba91d
  driver: scaleway
```

Note: In the cloud profile that uses this provider configuration, the syntax for the `provider` required field would be `provider: my-scaleway-config`.

Rackspace

Rackspace cloud requires two configuration options; a `user` and an `apikey`:

```
my-rackspace-config:
  user: example_user
  apikey: 123984bjjas87034
  driver: rackspace
```

Note: In the cloud profile that uses this provider configuration, the syntax for the `provider` required field would be `provider: my-rackspace-config`.

Amazon AWS

A number of configuration options are required for Amazon AWS including `id`, `key`, `keyname`, `securitygroup`, and `private_key`:

```
my-aws-quick-start:
  id: HJGRYCILJLKJYG
  key: 'kdjgfsgm;woormgl/aserigjksjdhasdfgn'
  keyname: test
  securitygroup: quick-start
  private_key: /root/test.pem
  driver: ec2

my-aws-default:
  id: HJGRYCILJLKJYG
  key: 'kdjgfsgm;woormgl/aserigjksjdhasdfgn'
  keyname: test
  securitygroup: default
  private_key: /root/test.pem
  driver: ec2
```

Note: In the cloud profile that uses this provider configuration, the syntax for the `provider` required field would be either `provider: my-aws-quick-start` or `provider: my-aws-default`.

Linode

Linode requires a single API key, but the default root password also needs to be set:

```
my-linode-config:
  apikey: asldkgfakl;sdfjsjaslfjaklsdjf;askldjfaaklsjdfhasldsadfghdkf
  password: F00barbaz
  ssh_pubkey: ssh-ed25519
↪AAAAC3NzaC1lZDI1NTE5AAAAIKHEOLLbeXgaqRQT9NBAopVz366SdYc0KKX33vAnq+2R user@host
```

(continues on next page)

(continued from previous page)

```
  ssh_key_file: ~/.ssh/id_ed25519
  driver: linode
```

The password needs to be 8 characters and contain lowercase, uppercase, and numbers.

Note: In the cloud profile that uses this provider configuration, the syntax for the `provider` required field would be `provider: my-linode-config`

Joyent Cloud

The Joyent cloud requires three configuration parameters: The username and password that are used to log into the Joyent system, as well as the location of the private SSH key associated with the Joyent account. The SSH key is needed to send the provisioning commands up to the freshly created virtual machine.

```
my-joyent-config:
  user: fred
  password: saltybacon
  private_key: /root/joyent.pem
  driver: joyent
```

Note: In the cloud profile that uses this provider configuration, the syntax for the `provider` required field would be `provider: my-joyent-config`

GoGrid

To use Salt Cloud with GoGrid, log into the GoGrid web interface and create an API key. Do this by clicking on ”My Account” and then going to the API Keys tab.

The `apikey` and the `sharedsecret` configuration parameters need to be set in the configuration file to enable interfacing with GoGrid:

```
my-gogrid-config:
  apikey: asdff7896asdh789
  sharedsecret: saltybacon
  driver: gogrid
```

Note: In the cloud profile that uses this provider configuration, the syntax for the `provider` required field would be `provider: my-gogrid-config`.

OpenStack

Using Salt for OpenStack uses the *shade <https://docs.openstack.org/shade/latest/>* driver managed by the openstack-infra team.

This driver can be configured using the `/etc/openstack/clouds.yml` file with *os-client-config <https://docs.openstack.org/os-client-config/latest/>*

```
myopenstack:
  driver: openstack
  region_name: RegionOne
  cloud: mycloud
```

Or by just configuring the same auth block directly in the cloud provider config.

```
myopenstack:
  driver: openstack
  region_name: RegionOne
  auth:
    username: 'demo'
    password: secret
    project_name: 'demo'
    auth_url: 'http://openstack/identity'
```

Both of these methods support using the *vendor <https://docs.openstack.org/os-client-config/latest/user/vendor-support.html>* options.

For more information, look at *Openstack Cloud Driver Docs*

DigitalOcean

Using Salt for DigitalOcean requires a `client_key` and an `api_key`. These can be found in the DigitalOcean web interface, in the "My Settings" section, under the API Access tab.

```
my-digitalocean-config:
  driver: digitalocean
  personal_access_token: xxx
  location: New York 1
```

Note: In the cloud profile that uses this provider configuration, the syntax for the `provider` required field would be `provider: my-digital-ocean-config`.

Parallels

Using Salt with Parallels requires a `user`, `password` and `URL`. These can be obtained from your cloud provider.

```
my-parallels-config:
  user: myuser
  password: xyzzy
  url: https://api.cloud.xmission.com:4465/paci/v1.0/
  driver: parallels
```

Note: In the cloud profile that uses this provider configuration, the syntax for the `provider` required field would be `provider: my-parallels-config`.

Proxmox

Using Salt with Proxmox requires a `user`, `password`, and `URL`. These can be obtained from your cloud host. Both PAM and PVE users can be used.

```
my-proxmox-config:
  driver: proxmox
  user: saltcloud@pve
  password: xyzzy
  url: your.proxmox.host
```

Note: In the cloud profile that uses this provider configuration, the syntax for the `provider` required field would be `provider: my-proxmox-config`.

LXC

The lxc driver uses saltify to install salt and attach the lxc container as a new lxc minion. As soon as we can, we manage baremetal operation over SSH. You can also destroy those containers via this driver.

```
devhost10-lxc:
  target: devhost10
  driver: lxc
```

And in the map file:

```
devhost10-lxc:
  provider: devhost10-lxc
  from_container: ubuntu
  backing: lvm
  sudo: True
  size: 3g
  ip: 10.0.3.9
  minion:
    master: 10.5.0.1
    master_port: 4506
  lxc_conf:
    - lxc.utsname: superlxc
```

Note: In the cloud profile that uses this provider configuration, the syntax for the `provider` required field would be `provider: devhost10-lxc`.

Saltify

The Saltify driver is a new, experimental driver designed to install Salt on a remote machine, virtual or bare metal, using SSH. This driver is useful for provisioning machines which are already installed, but not Salted. For more information about using this driver and for configuration examples, please see the *Gettting Started with Saltify* documentation.

Vagrant

The Vagrant driver is a new, experimental driver for controlling a VagrantBox virtual machine, and installing Salt on it. The target host machine must be a working salt minion, which is controlled via the salt master using salt-api. For more information, see *Getting Started With Vagrant*.

Extending Profiles and Cloud Providers Configuration

As of 0.8.7, the option to extend both the profiles and cloud providers configuration and avoid duplication was added. The extends feature works on the current profiles configuration, but, regarding the cloud providers configuration, **only** works in the new syntax and respective configuration files, i.e. `/etc/salt/salt/cloud.providers` or `/etc/salt/cloud.providers.d/*.conf`.

Note: Extending cloud profiles and providers is not recursive. For example, a profile that is extended by a second profile is possible, but the second profile cannot be extended by a third profile.

Also, if a profile (or provider) is extending another profile and each contains a list of values, the lists from the extending profile will override the list from the original profile. The lists are not merged together.

Extending Profiles

Some example usage on how to use `extends` with profiles. Consider `/etc/salt/salt/cloud.profiles` containing:

```
development-instances:
  provider: my-ec2-config
  size: t1.micro
  ssh_username: ec2_user
  securitygroup:
    - default
  deploy: False

Amazon-Linux-AMI-2012.09-64bit:
  image: ami-54cf5c3d
  extends: development-instances

Fedora-17:
  image: ami-08d97e61
  extends: development-instances

CentOS-5:
  provider: my-aws-config
  image: ami-09b61d60
  extends: development-instances
```

The above configuration, once parsed would generate the following profiles data:

```
[{'deploy': False,
  'image': 'ami-08d97e61',
  'profile': 'Fedora-17',
  'provider': 'my-ec2-config',
  'securitygroup': ['default'],
  'size': 't1.micro',
```

(continues on next page)

(continued from previous page)

```
  'ssh_username': 'ec2_user'},
 {'deploy': False,
  'image': 'ami-09b61d60',
  'profile': 'CentOS-5',
  'provider': 'my-aws-config',
  'securitygroup': ['default'],
  'size': 't1.micro',
  'ssh_username': 'ec2_user'},
 {'deploy': False,
  'image': 'ami-54cf5c3d',
  'profile': 'Amazon-Linux-AMI-2012.09-64bit',
  'provider': 'my-ec2-config',
  'securitygroup': ['default'],
  'size': 't1.micro',
  'ssh_username': 'ec2_user'},
 {'deploy': False,
  'profile': 'development-instances',
  'provider': 'my-ec2-config',
  'securitygroup': ['default'],
  'size': 't1.micro',
  'ssh_username': 'ec2_user'}]
```

Pretty cool right?

Extending Providers

Some example usage on how to use `extends` within the cloud providers configuration. Consider `/etc/salt/salt/cloud.providers` containing:

```
my-develop-envs:
  - id: HJGRYCILJLKJYG
    key: 'kdjgfsgm;woormgl/aserigjksjdhasdfgn'
    keyname: test
    securitygroup: quick-start
    private_key: /root/test.pem
    location: ap-southeast-1
    availability_zone: ap-southeast-1b
    driver: ec2

  - user: myuser@mycorp.com
    password: mypass
    ssh_key_name: mykey
    ssh_key_file: '/etc/salt/ibm/mykey.pem'
    location: Raleigh
    driver: ibmsce


my-productions-envs:
  - extends: my-develop-envs:ibmsce
    user: my-production-user@mycorp.com
    location: us-east-1
    availability_zone: us-east-1
```

The above configuration, once parsed would generate the following providers data:

```
'providers': {
    'my-develop-envs': [
        {'availability_zone': 'ap-southeast-1b',
         'id': 'HJGRYCILJLKJYG',
         'key': 'kdjgfsgm;woormgl/aserigjksjdhasdfgn',
         'keyname': 'test',
         'location': 'ap-southeast-1',
         'private_key': '/root/test.pem',
         'driver': 'aws',
         'securitygroup': 'quick-start'
        },
        {'location': 'Raleigh',
         'password': 'mypass',
         'driver': 'ibmsce',
         'ssh_key_file': '/etc/salt/ibm/mykey.pem',
         'ssh_key_name': 'mykey',
         'user': 'myuser@mycorp.com'
        }
    ],
    'my-productions-envs': [
        {'availability_zone': 'us-east-1',
         'location': 'us-east-1',
         'password': 'mypass',
         'driver': 'ibmsce',
         'ssh_key_file': '/etc/salt/ibm/mykey.pem',
         'ssh_key_name': 'mykey',
         'user': 'my-production-user@mycorp.com'
        }
    ]
}
```

14.6 Windows Configuration

14.6.1 Spinning up Windows Minions

It is possible to use Salt Cloud to spin up Windows instances, and then install Salt on them. This functionality is available on all cloud providers that are supported by Salt Cloud. However, it may not necessarily be available on all Windows images.

Requirements

Note: Support `winexe` and `impacket` has been deprecated and will be removed in Sodium. These dependencies are replaced by `pypsexec` and `smbprotocol` respectivly. These are pure python alternatives that are compatible with all supported python versions.

Salt Cloud makes use of *impacket* and *winexe* to set up the Windows Salt Minion installer.

impacket is usually available as either the *impacket* or the *python-impacket* package, depending on the distribution. More information on *impacket* can be found at the project home:

- impacket project home

winexe is less commonly available in distribution-specific repositories. However, it is currently being built for various distributions in 3rd party channels:

- RPMs at pbone.net
- openSUSE Build Service
- pypsexec project home
- smbprotocol project home

Optionally WinRM can be used instead of *winexe* if the python module *pywinrm* is available and WinRM is supported on the target Windows version. Information on pywinrm can be found at the project home:

- pywinrm project home

Additionally, a copy of the Salt Minion Windows installer must be present on the system on which Salt Cloud is running. This installer may be downloaded from saltstack.com:

- SaltStack Download Area

Self Signed Certificates with WinRM

Salt-Cloud can use versions of `pywinrm<=0.1.1` or `pywinrm>=0.2.1`.

For versions greater than *0.2.1*, `winrm_verify_ssl` needs to be set to *False* if the certificate is self signed and not verifiable.

Firewall Settings

Because Salt Cloud makes use of *smbclient* and *winexe*, port 445 must be open on the target image. This port is not generally open by default on a standard Windows distribution, and care must be taken to use an image in which this port is open, or the Windows firewall is disabled.

If supported by the cloud provider, a PowerShell script may be used to open up this port automatically, using the cloud provider's *userdata*. The following script would open up port 445, and apply the changes:

```
<powershell>
New-NetFirewallRule -Name "SMB445" -DisplayName "SMB445" -Protocol TCP -LocalPort 445
Set-Item (dir wsman:\localhost\Listener\*\Port -Recurse).pspath 445 -Force
Restart-Service winrm
</powershell>
```

For EC2, this script may be saved as a file, and specified in the provider or profile configuration as *userdata_file*. For instance:

```
my-ec2-config:
  # Pass userdata to the instance to be created
  userdata_file: /etc/salt/windows-firewall.ps1
```

Note: From versions 2016.11.0 and 2016.11.3, this file was passed through the master's `renderer` to template it. However, this caused issues with non-YAML data, so templating is no longer performed by default. To template the userdata_file, add a `userdata_template` option to the cloud profile:

```
my-ec2-config:
  # Pass userdata to the instance to be created
  userdata_file: /etc/salt/windows-firewall.ps1
  userdata_template: jinja
```

If no `userdata_template` is set in the cloud profile, then the master configuration will be checked for a *`userdata_template`* value. If this is not set, then no templating will be performed on the userdata_file.

To disable templating in a cloud profile when a *`userdata_template`* has been set in the master configuration file, simply set `userdata_template` to `False` in the cloud profile:

```
my-ec2-config:
  # Pass userdata to the instance to be created
  userdata_file: /etc/salt/windows-firewall.ps1
  userdata_template: False
```

If you are using WinRM on EC2 the HTTPS port for the WinRM service must also be enabled in your userdata. By default EC2 Windows images only have insecure HTTP enabled. To enable HTTPS and basic authentication required by pywinrm consider the following userdata example:

```
<powershell>
New-NetFirewallRule -Name "SMB445" -DisplayName "SMB445" -Protocol TCP -LocalPort 445
New-NetFirewallRule -Name "WINRM5986" -DisplayName "WINRM5986" -Protocol TCP -
↪LocalPort 5986

winrm quickconfig -q
winrm set winrm/config/winrs '@{MaxMemoryPerShellMB="300"}'
winrm set winrm/config '@{MaxTimeoutms="1800000"}'
winrm set winrm/config/service/auth '@{Basic="true"}'

$SourceStoreScope = 'LocalMachine'
$SourceStorename = 'Remote Desktop'

$SourceStore = New-Object -TypeName System.Security.Cryptography.X509Certificates.
↪X509Store -ArgumentList $SourceStorename, $SourceStoreScope
$SourceStore.Open([System.Security.Cryptography.X509Certificates.OpenFlags]::ReadOnly)

$cert = $SourceStore.Certificates | Where-Object -FilterScript {
    $_.subject -like '*'
}

$DestStoreScope = 'LocalMachine'
$DestStoreName = 'My'

$DestStore = New-Object -TypeName System.Security.Cryptography.X509Certificates.
↪X509Store -ArgumentList $DestStoreName, $DestStoreScope
$DestStore.Open([System.Security.Cryptography.X509Certificates.OpenFlags]::ReadWrite)
$DestStore.Add($cert)

$SourceStore.Close()
$DestStore.Close()

winrm create winrm/config/listener?Address=*+Transport=HTTPS `@`
↪{CertificateThumbprint=`"($cert.Thumbprint)`"`}

Restart-Service winrm
</powershell>
```

No certificate store is available by default on EC2 images and creating one does not seem possible without an MMC (cannot be automated). To use the default EC2 Windows images the above copies the RDP store.

Configuration

Configuration is set as usual, with some extra configuration settings. The location of the Windows installer on the machine that Salt Cloud is running on must be specified. This may be done in any of the regular configuration files (main, providers, profiles, maps). For example:

Setting the installer in `/etc/salt/cloud.providers`:

```
my-softlayer:
  driver: softlayer
  user: MYUSER1138
  apikey: 'e3b68aa711e6deadc62d5b76355674beef7cc3116062ddbacafe5f7e465bfdc9'
  minion:
    master: saltmaster.example.com
  win_installer: /root/Salt-Minion-2014.7.0-AMD64-Setup.exe
  win_username: Administrator
  win_password: letmein
  smb_port: 445
```

The default Windows user is *Administrator*, and the default Windows password is blank.

If WinRM is to be used `use_winrm` needs to be set to *True*. `winrm_port` can be used to specify a custom port (must be HTTPS listener). And `winrm_verify_ssl` can be set to *False* to use a self signed certificate.

Auto-Generated Passwords on EC2

On EC2, when the *win_password* is set to *auto*, Salt Cloud will query EC2 for an auto-generated password. This password is expected to take at least 4 minutes to generate, adding additional time to the deploy process.

When the EC2 API is queried for the auto-generated password, it will be returned in a message encrypted with the specified *keyname*. This requires that the appropriate *private_key* file is also specified. Such a profile configuration might look like:

```
windows-server-2012:
  provider: my-ec2-config
  image: ami-c49c0dac
  size: m1.small
  securitygroup: windows
  keyname: mykey
  private_key: /root/mykey.pem
  userdata_file: /etc/salt/windows-firewall.ps1
  win_installer: /root/Salt-Minion-2014.7.0-AMD64-Setup.exe
  win_username: Administrator
  win_password: auto
```

14.7 Cloud Provider Specifics

14.7.1 Getting Started With Aliyun ECS

The Aliyun ECS (Elastic Computer Service) is one of the most popular public cloud hosts in China. This cloud host can be used to manage aliyun instance using salt-cloud.

http://www.aliyun.com/

Dependencies

This driver requires the Python `requests` library to be installed.

Configuration

Using Salt for Aliyun ECS requires aliyun access key id and key secret. These can be found in the aliyun web interface, in the "User Center" section, under "My Service" tab.

```
# Note: This example is for /etc/salt/cloud.providers or any file in the
# /etc/salt/cloud.providers.d/ directory.

my-aliyun-config:
  # aliyun Access Key ID
  id: wDGEwGregedg3435gDgxd
  # aliyun Access Key Secret
  key: GDd45t43RDBTrkkkg43934t34qT43t4dgegerGEgg
  location: cn-qingdao
  driver: aliyun
```

Note: Changed in version 2015.8.0.

The `provider` parameter in cloud provider definitions was renamed to `driver`. This change was made to avoid confusion with the `provider` parameter that is used in cloud profile definitions. Cloud provider definitions now use `driver` to refer to the Salt cloud module that provides the underlying functionality to connect to a cloud host, while cloud profiles continue to use `provider` to refer to provider configurations that you define.

Profiles

Cloud Profiles

Set up an initial profile at `/etc/salt/cloud.profiles` or in the `/etc/salt/cloud.profiles.d/` directory:

```
aliyun_centos:
    provider: my-aliyun-config
    size: ecs.t1.small
    location: cn-qingdao
    securitygroup: G1989096784427999
    image: centos6u3_64_20G_aliaegis_20130816.vhd
```

Sizes can be obtained using the `--list-sizes` option for the `salt-cloud` command:

```
# salt-cloud --list-sizes my-aliyun-config
my-aliyun-config:
    ----------
    aliyun:
        ----------
        ecs.c1.large:
            ----------
            CpuCoreCount:
                8
            InstanceTypeId:
```

(continues on next page)

(continued from previous page)

```
                ecs.c1.large
            MemorySize:
                16.0

...SNIP...
```

Images can be obtained using the `--list-images` option for the `salt-cloud` command:

```
# salt-cloud --list-images my-aliyun-config
my-aliyun-config:
    ----------
    aliyun:
        ----------
        centos5u8_64_20G_aliaegis_20131231.vhd:
            ----------
            Architecture:
                x86_64
            Description:

            ImageId:
                centos5u8_64_20G_aliaegis_20131231.vhd
            ImageName:
                CentOS 5.8 64⊠
            ImageOwnerAlias:
                system
            ImageVersion:
                1.0
            OSName:
                CentOS  5.8 64⊠
            Platform:
                CENTOS5
            Size:
                20
            Visibility:
                public
...SNIP...
```

Locations can be obtained using the `--list-locations` option for the `salt-cloud` command:

```
my-aliyun-config:
    ----------
    aliyun:
        ----------
        cn-beijing:
            ----------
            LocalName:
                ⊠⊠
            RegionId:
                cn-beijing
        cn-hangzhou:
            ----------
            LocalName:
                ⊠⊠
            RegionId:
                cn-hangzhou
        cn-hongkong:
```

(continues on next page)

(continued from previous page)

```
            ----------
            LocalName:
                ☒☒
            RegionId:
                cn-hongkong
        cn-qingdao:
            ----------
            LocalName:
                ☒☒
            RegionId:
                cn-qingdao
```

Security Group can be obtained using the `-f list_securitygroup` option for the `salt-cloud` command:

```
# salt-cloud --location=cn-qingdao -f list_securitygroup my-aliyun-config
my-aliyun-config:
    ----------
    aliyun:
        ----------
        G1989096784427999:
            ----------
            Description:
                G1989096784427999
            SecurityGroupId:
                G1989096784427999
```

Note: Aliyun ECS REST API documentation is available from Aliyun ECS API.

14.7.2 Getting Started With Azure

New in version 2014.1.0.

Azure is a cloud service by Microsoft providing virtual machines, SQL services, media services, and more. This document describes how to use Salt Cloud to create a virtual machine on Azure, with Salt installed.

More information about Azure is located at http://www.windowsazure.com/.

Dependencies

- Microsoft Azure SDK for Python >= 1.0.2
- The python-requests library, for Python < 2.7.9.
- A Microsoft Azure account
- OpenSSL (to generate the certificates)
- Salt

Configuration

Set up the provider config at `/etc/salt/cloud.providers.d/azure.conf`:

```
# Note: This example is for /etc/salt/cloud.providers.d/azure.conf

my-azure-config:
  driver: azure
  subscription_id: 3287abc8-f98a-c678-3bde-326766fd3617
  certificate_path: /etc/salt/azure.pem

  # Set up the location of the salt master
  #
  minion:
    master: saltmaster.example.com

  # Optional
  management_host: management.core.windows.net
```

The certificate used must be generated by the user. OpenSSL can be used to create the management certificates. Two certificates are needed: a .cer file, which is uploaded to Azure, and a .pem file, which is stored locally.

To create the .pem file, execute the following command:

```
openssl req -x509 -nodes -days 365 -newkey rsa:1024 -keyout /etc/salt/azure.pem -out /
↪etc/salt/azure.pem
```

To create the .cer file, execute the following command:

```
openssl x509 -inform pem -in /etc/salt/azure.pem -outform der -out /etc/salt/azure.cer
```

After creating these files, the .cer file will need to be uploaded to Azure via the "Upload a Management Certificate" action of the "Management Certificates" tab within the "Settings" section of the management portal.

Optionally, a `management_host` may be configured, if necessary for the region.

Note: Changed in version 2015.8.0.

The `provider` parameter in cloud provider definitions was renamed to `driver`. This change was made to avoid confusion with the `provider` parameter that is used in cloud profile definitions. Cloud provider definitions now use `driver` to refer to the Salt cloud module that provides the underlying functionality to connect to a cloud host, while cloud profiles continue to use `provider` to refer to provider configurations that you define.

Cloud Profiles

Set up an initial profile at `/etc/salt/cloud.profiles`:

```
azure-ubuntu:
  provider: my-azure-config
  image: 'b39f27a8b8c64d52b05eac6a62ebad85__Ubuntu-12_04_3-LTS-amd64-server-20131003-
↪en-us-30GB'
  size: Small
  location: 'East US'
  ssh_username: azureuser
  ssh_password: verybadpass
  slot: production
  media_link: 'http://portalvhdabcdefghijklmn.blob.core.windows.net/vhds'
  virtual_network_name: azure-virtual-network
  subnet_name: azure-subnet
```

These options are described in more detail below. Once configured, the profile can be realized with a salt command:

```
salt-cloud -p azure-ubuntu newinstance
```

This will create an salt minion instance named `newinstance` in Azure. If the command was executed on the salt-master, its Salt key will automatically be signed on the master.

Once the instance has been created with salt-minion installed, connectivity to it can be verified with Salt:

```
salt newinstance test.version
```

Profile Options

The following options are currently available for Azure.

provider

The name of the provider as configured in */etc/salt/cloud.providers.d/azure.conf*.

image

The name of the image to use to create a VM. Available images can be viewed using the following command:

```
salt-cloud --list-images my-azure-config
```

size

The name of the size to use to create a VM. Available sizes can be viewed using the following command:

```
salt-cloud --list-sizes my-azure-config
```

location

The name of the location to create a VM in. Available locations can be viewed using the following command:

```
salt-cloud --list-locations my-azure-config
```

affinity_group

The name of the affinity group to create a VM in. Either a `location` or an `affinity_group` may be specified, but not both. See Affinity Groups below.

ssh_username

The user to use to log into the newly-created VM to install Salt.

ssh_password

The password to use to log into the newly-created VM to install Salt.

slot

The environment to which the hosted service is deployed. Valid values are *staging* or *production*. When set to *production*, the resulting URL of the new VM will be *<vm_name>.cloudapp.net*. When set to *staging*, the resulting URL will contain a generated hash instead.

media_link

This is the URL of the container that will store the disk that this VM uses. Currently, this container must already exist. If a VM has previously been created in the associated account, a container should already exist. In the web interface, go into the Storage area and click one of the available storage selections. Click the Containers link, and then copy the URL from the container that will be used. It generally looks like:

```
http://portalvhdabcdefghijklmn.blob.core.windows.net/vhds
```

service_name

The name of the service in which to create the VM. If this is not specified, then a service will be created with the same name as the VM.

virtual_network_name

Optional. The name of the virtual network for the VM to join. If this is not specified, then no virtual network will be joined.

subnet_name

Optional. The name of the subnet in the virtual network for the VM to join. Requires that a `virtual_network_name` is specified.

Show Instance

This action is a thin wrapper around `--full-query`, which displays details on a single instance only. In an environment with several machines, this will save a user from having to sort through all instance data, just to examine a single instance.

```
salt-cloud -a show_instance myinstance
```

Destroying VMs

There are certain options which can be specified in the global cloud configuration file (usually `/etc/salt/cloud`) which affect Salt Cloud's behavior when a VM is destroyed.

cleanup_disks

New in version 2015.8.0.

Default is `False`. When set to `True`, Salt Cloud will wait for the VM to be destroyed, then attempt to destroy the main disk that is associated with the VM.

cleanup_vhds

New in version 2015.8.0.

Default is `False`. Requires `cleanup_disks` to be set to `True`. When also set to `True`, Salt Cloud will ask Azure to delete the VHD associated with the disk that is also destroyed.

cleanup_services

New in version 2015.8.0.

Default is `False`. Requires `cleanup_disks` to be set to `True`. When also set to `True`, Salt Cloud will wait for the disk to be destroyed, then attempt to remove the service that is associated with the VM. Because the disk belongs to the service, the disk must be destroyed before the service can be.

Managing Hosted Services

New in version 2015.8.0.

An account can have one or more hosted services. A hosted service is required in order to create a VM. However, as mentioned above, if a hosted service is not specified when a VM is created, then one will automatically be created with the name of the name. The following functions are also available.

create_service

Create a hosted service. The following options are available.

name

Required. The name of the hosted service to create.

label

Required. A label to apply to the hosted service.

description

Optional. A longer description of the hosted service.

location

Required, if `affinity_group` is not set. The location in which to create the hosted service. Either the `location` or the `affinity_group` must be set, but not both.

affinity_group

Required, if `location` is not set. The affinity group in which to create the hosted service. Either the `location` or the `affinity_group` must be set, but not both.

extended_properties

Optional. Dictionary containing name/value pairs of hosted service properties. You can have a maximum of 50 extended property name/value pairs. The maximum length of the Name element is 64 characters, only alphanumeric characters and underscores are valid in the Name, and the name must start with a letter. The value has a maximum length of 255 characters.

CLI Example

The following example illustrates creating a hosted service.

```
salt-cloud -f create_service my-azure name=my-service label=my-service location='West
↪US'
```

show_service

Return details about a specific hosted service. Can also be called with `get_service`.

```
salt-cloud -f show_storage my-azure name=my-service
```

list_services

List all hosted services associates with the subscription.

```
salt-cloud -f list_services my-azure-config
```

delete_service

Delete a specific hosted service.

```
salt-cloud -f delete_service my-azure name=my-service
```

Managing Storage Accounts

New in version 2015.8.0.

Salt Cloud can manage storage accounts associated with the account. The following functions are available. Deprecated marked as deprecated are marked as such as per the SDK documentation, but are still included for completeness with the SDK.

create_storage

Create a storage account. The following options are supported.

name

Required. The name of the storage account to create.

label

Required. A label to apply to the storage account.

description

Optional. A longer description of the storage account.

location

Required, if `affinity_group` is not set. The location in which to create the storage account. Either the `location` or the `affinity_group` must be set, but not both.

affinity_group

Required, if `location` is not set. The affinity group in which to create the storage account. Either the `location` or the `affinity_group` must be set, but not both.

extended_properties

Optional. Dictionary containing name/value pairs of storage account properties. You can have a maximum of 50 extended property name/value pairs. The maximum length of the Name element is 64 characters, only alphanumeric characters and underscores are valid in the Name, and the name must start with a letter. The value has a maximum length of 255 characters.

geo_replication_enabled

Deprecated. Replaced by the account_type parameter.

account_type

Specifies whether the account supports locally-redundant storage, geo-redundant storage, zone-redundant storage, or read access geo-redundant storage. Possible values are:

- Standard_LRS
- Standard_ZRS
- Standard_GRS
- Standard_RAGRS

CLI Example

The following example illustrates creating a storage account.

```
salt-cloud -f create_storage my-azure name=my-storage label=my-storage location='West
↪US'
```

list_storage

List all storage accounts associates with the subscription.

```
salt-cloud -f list_storage my-azure-config
```

show_storage

Return details about a specific storage account. Can also be called with `get_storage`.

```
salt-cloud -f show_storage my-azure name=my-storage
```

update_storage

Update details concerning a storage account. Any of the options available in `create_storage` can be used, but the name cannot be changed.

```
salt-cloud -f update_storage my-azure name=my-storage label=my-storage
```

delete_storage

Delete a specific storage account.

```
salt-cloud -f delete_storage my-azure name=my-storage
```

show_storage_keys

Returns the primary and secondary access keys for the specified storage account.

```
salt-cloud -f show_storage_keys my-azure name=my-storage
```

regenerate_storage_keys

Regenerate storage account keys. Requires a key_type ("primary" or "secondary") to be specified.

```
salt-cloud -f regenerate_storage_keys my-azure name=my-storage key_type=primary
```

Managing Disks

New in version 2015.8.0.

When a VM is created, a disk will also be created for it. The following functions are available for managing disks. Deprecated marked as deprecated are marked as such as per the SDK documentation, but are still included for completeness with the SDK.

show_disk

Return details about a specific disk. Can also be called with `get_disk`.

```
salt-cloud -f show_disk my-azure name=my-disk
```

list_disks

List all disks associates with the account.

```
salt-cloud -f list_disks my-azure
```

update_disk

Update details for a disk. The following options are available.

name

Required. The name of the disk to update.

has_operating_system

Deprecated.

label

Required. The label for the disk.

media_link

Deprecated. The location of the disk in the account, including the storage container that it is in. This should not need to be changed.

new_name

Deprecated. If renaming the disk, the new name.

os

Deprecated.

CLI Example

The following example illustrates updating a disk.

```
salt-cloud -f update_disk my-azure name=my-disk label=my-disk
```

delete_disk

Delete a specific disk.

```
salt-cloud -f delete_disk my-azure name=my-disk
```

Managing Service Certificates

New in version 2015.8.0.

Stored at the cloud service level, these certificates are used by your deployed services. For more information on service certificates, see the following link:

- Manage Certificates

The following functions are available.

list_service_certificates

List service certificates associated with the account.

```
salt-cloud -f list_service_certificates my-azure
```

show_service_certificate

Show the data for a specific service certificate associated with the account. The `name`, `thumbprint`, and `thumbalgorithm` can be obtained from `list_service_certificates`. Can also be called with `get_service_certificate`.

```
salt-cloud -f show_service_certificate my-azure name=my_service_certificate \
    thumbalgorithm=sha1 thumbprint=0123456789ABCDEF
```

add_service_certificate

Add a service certificate to the account. This requires that a certificate already exists, which is then added to the account. For more information on creating the certificate itself, see:

- Create a Service Certificate for Azure

The following options are available.

name

Required. The name of the hosted service that the certificate will belong to.

data

Required. The base-64 encoded form of the pfx file.

certificate_format

Required. The service certificate format. The only supported value is pfx.

password

The certificate password.

```
salt-cloud -f add_service_certificate my-azure name=my-cert \
    data='...CERT_DATA...' certificate_format=pfx password=verybadpass
```

delete_service_certificate

Delete a service certificate from the account. The `name`, `thumbprint`, and `thumbalgorithm` can be obtained from `list_service_certificates`.

```
salt-cloud -f delete_service_certificate my-azure \
    name=my_service_certificate \
    thumbalgorithm=sha1 thumbprint=0123456789ABCDEF
```

Managing Management Certificates

New in version 2015.8.0.

A Azure management certificate is an X.509 v3 certificate used to authenticate an agent, such as Visual Studio Tools for Windows Azure or a client application that uses the Service Management API, acting on behalf of the subscription owner to manage subscription resources. Azure management certificates are uploaded to Azure and stored at the subscription level. The management certificate store can hold up to 100 certificates per subscription. These certificates are used to authenticate your Windows Azure deployment.

For more information on management certificates, see the following link.

- Manage Certificates

The following functions are available.

list_management_certificates

List management certificates associated with the account.

```
salt-cloud -f list_management_certificates my-azure
```

show_management_certificate

Show the data for a specific management certificate associated with the account. The `name`, `thumbprint`, and `thumbalgorithm` can be obtained from `list_management_certificates`. Can also be called with `get_management_certificate`.

```
salt-cloud -f show_management_certificate my-azure name=my_management_certificate \
    thumbalgorithm=sha1 thumbprint=0123456789ABCDEF
```

add_management_certificate

Management certificates must have a key length of at least 2048 bits and should reside in the Personal certificate store. When the certificate is installed on the client, it should contain the private key of the certificate. To upload to the certificate to the Microsoft Azure Management Portal, you must export it as a .cer format file that does not contain the private key. For more information on creating management certificates, see the following link:

- Create and Upload a Management Certificate for Azure

The following options are available.

public_key

A base64 representation of the management certificate public key.

thumbprint

The thumb print that uniquely identifies the management certificate.

data

The certificate's raw data in base-64 encoded .cer format.

```
salt-cloud -f add_management_certificate my-azure public_key='...PUBKEY...' \
    thumbprint=0123456789ABCDEF data='...CERT_DATA...'
```

delete_management_certificate

Delete a management certificate from the account. The `thumbprint` can be obtained from `list_management_certificates`.

```
salt-cloud -f delete_management_certificate my-azure thumbprint=0123456789ABCDEF
```

Virtual Network Management

New in version 2015.8.0.

The following are functions for managing virtual networks.

list_virtual_networks

List input endpoints associated with the deployment.

```
salt-cloud -f list_virtual_networks my-azure service=myservice deployment=mydeployment
```

Managing Input Endpoints

New in version 2015.8.0.

Input endpoints are used to manage port access for roles. Because endpoints cannot be managed by the Azure Python SDK, Salt Cloud uses the API directly. With versions of Python before 2.7.9, the `requests-python` package needs to be installed in order for this to work. Additionally, the following needs to be set in the master's configuration file:

```
backend: requests
```

The following functions are available.

list_input_endpoints

List input endpoints associated with the deployment

```
salt-cloud -f list_input_endpoints my-azure service=myservice deployment=mydeployment
```

show_input_endpoint

Show an input endpoint associated with the deployment

```
salt-cloud -f show_input_endpoint my-azure service=myservice \
    deployment=mydeployment name=SSH
```

add_input_endpoint

Add an input endpoint to the deployment. Please note that there may be a delay before the changes show up. The following options are available.

service

Required. The name of the hosted service which the VM belongs to.

deployment

Required. The name of the deployment that the VM belongs to. If the VM was created with Salt Cloud, the deployment name probably matches the VM name.

role

Required. The name of the role that the VM belongs to. If the VM was created with Salt Cloud, the role name probably matches the VM name.

name

Required. The name of the input endpoint. This typically matches the port that the endpoint is set to. For instance, port 22 would be called SSH.

port

Required. The public (Internet-facing) port that is used for the endpoint.

local_port

Optional. The private port on the VM itself that will be matched with the port. This is typically the same as the `port`. If this value is not specified, it will be copied from `port`.

protocol

Required. Either `tcp` or `udp`.

enable_direct_server_return

Optional. If an internal load balancer exists in the account, it can be used with a direct server return. The default value is `False`. Please see the following article for an explanation of this option.

- Load Balancing for Azure Infrastructure Services

timeout_for_tcp_idle_connection

Optional. The default value is `4`. Please see the following article for an explanation of this option.

- Configurable Idle Timeout for Azure Load Balancer

CLI Example

The following example illustrates adding an input endpoint.

```
salt-cloud -f add_input_endpoint my-azure service=myservice \
    deployment=mydeployment role=myrole name=HTTP local_port=80 \
    port=80 protocol=tcp enable_direct_server_return=False \
    timeout_for_tcp_idle_connection=4
```

update_input_endpoint

Updates the details for a specific input endpoint. All options from `add_input_endpoint` are supported.

```
salt-cloud -f update_input_endpoint my-azure service=myservice \
    deployment=mydeployment role=myrole name=HTTP local_port=80 \
    port=80 protocol=tcp enable_direct_server_return=False \
    timeout_for_tcp_idle_connection=4
```

delete_input_endpoint

Delete an input endpoint from the deployment. Please note that there may be a delay before the changes show up. The following items are required.

CLI Example

The following example illustrates deleting an input endpoint.

service

The name of the hosted service which the VM belongs to.

deployment

The name of the deployment that the VM belongs to. If the VM was created with Salt Cloud, the deployment name probably matches the VM name.

role

The name of the role that the VM belongs to. If the VM was created with Salt Cloud, the role name probably matches the VM name.

name

The name of the input endpoint. This typically matches the port that the endpoint is set to. For instance, port 22 would be called SSH.

```
salt-cloud -f delete_input_endpoint my-azure service=myservice \
    deployment=mydeployment role=myrole name=HTTP
```

Managing Affinity Groups

New in version 2015.8.0.

Affinity groups allow you to group your Azure services to optimize performance. All services and VMs within an affinity group will be located in the same region. For more information on Affinity groups, see the following link:

- Create an Affinity Group in the Management Portal

The following functions are available.

list_affinity_groups

List input endpoints associated with the account

```
salt-cloud -f list_affinity_groups my-azure
```

show_affinity_group

Show an affinity group associated with the account

```
salt-cloud -f show_affinity_group my-azure service=myservice \
    deployment=mydeployment name=SSH
```

create_affinity_group

Create a new affinity group. The following options are supported.

name

Required. The name of the new affinity group.

location

Required. The region in which the affinity group lives.

label

Required. A label describing the new affinity group.

description

Optional. A longer description of the affinity group.

```
salt-cloud -f create_affinity_group my-azure name=my_affinity_group \
   label=my-affinity-group location='West US'
```

update_affinity_group

Update an affinity group's properties

```
salt-cloud -f update_affinity_group my-azure name=my_group label=my_group
```

delete_affinity_group

Delete a specific affinity group associated with the account

```
salt-cloud -f delete_affinity_group my-azure name=my_affinity_group
```

Managing Blob Storage

New in version 2015.8.0.

Azure storage containers and their contents can be managed with Salt Cloud. This is not as elegant as using one of the other available clients in Windows, but it benefits Linux and Unix users, as there are fewer options available on those platforms.

Blob Storage Configuration

Blob storage must be configured differently than the standard Azure configuration. Both a `storage_account` and a `storage_key` must be specified either through the Azure provider configuration (in addition to the other Azure configuration) or via the command line.

```
storage_account: mystorage
storage_key: ffhj334fDSGFEGDFGFDewr34fwfsFSDFwe==
```

storage_account

This is one of the storage accounts that is available via the `list_storage` function.

storage_key

Both a primary and a secondary `storage_key` can be obtained by running the `show_storage_keys` function. Either key may be used.

Blob Functions

The following functions are made available through Salt Cloud for managing blog storage.

make_blob_url

Creates the URL to access a blob

```
salt-cloud -f make_blob_url my-azure container=mycontainer blob=myblob
```

container

Name of the container.

blob

Name of the blob.

account

Name of the storage account. If not specified, derives the host base from the provider configuration.

protocol

Protocol to use: 'http' or 'https'. If not specified, derives the host base from the provider configuration.

host_base

Live host base URL. If not specified, derives the host base from the provider configuration.

list_storage_containers

List containers associated with the storage account

```
salt-cloud -f list_storage_containers my-azure
```

create_storage_container

Create a storage container

```
salt-cloud -f create_storage_container my-azure name=mycontainer
```

name

Name of container to create.

meta_name_values

Optional. A dict with name_value pairs to associate with the container as metadata. Example:{'Category':'test'}

blob_public_access

Optional. Possible values include: container, blob

fail_on_exist

Specify whether to throw an exception when the container exists.

show_storage_container

Show a container associated with the storage account

```
salt-cloud -f show_storage_container my-azure name=myservice
```

name

Name of container to show.

show_storage_container_metadata

Show a storage container's metadata

```
salt-cloud -f show_storage_container_metadata my-azure name=myservice
```

name

Name of container to show.

lease_id

If specified, show_storage_container_metadata only succeeds if the container's lease is active and matches this ID.

set_storage_container_metadata

Set a storage container's metadata

```
salt-cloud -f set_storage_container my-azure name=mycontainer \
    x_ms_meta_name_values='{"my_name": "my_value"}'
```

name

Name of existing container. meta_name_values ```````````` A dict containing name, value for metadata. Example: {'category':'test'} lease_id ```` If specified, set_storage_container_metadata only succeeds if the container's lease is active and matches this ID.

show_storage_container_acl

Show a storage container's acl

```
salt-cloud -f show_storage_container_acl my-azure name=myservice
```

name

Name of existing container.

lease_id

If specified, show_storage_container_acl only succeeds if the container's lease is active and matches this ID.

set_storage_container_acl

Set a storage container's acl

```
salt-cloud -f set_storage_container my-azure name=mycontainer
```

name

Name of existing container.

signed_identifiers

SignedIdentifers instance

blob_public_access

Optional. Possible values include: container, blob

lease_id

If specified, set_storage_container_acl only succeeds if the container's lease is active and matches this ID.

delete_storage_container

Delete a container associated with the storage account

```
salt-cloud -f delete_storage_container my-azure name=mycontainer
```

name

Name of container to create.

fail_not_exist

Specify whether to throw an exception when the container exists.

lease_id

If specified, delete_storage_container only succeeds if the container's lease is active and matches this ID.

lease_storage_container

Lease a container associated with the storage account

```
salt-cloud -f lease_storage_container my-azure name=mycontainer
```

name

Name of container to create.

lease_action

Required. Possible values: acquire|renew|release|break|change

lease_id

Required if the container has an active lease.

lease_duration

Specifies the duration of the lease, in seconds, or negative one (-1) for a lease that never expires. A non-infinite lease can be between 15 and 60 seconds. A lease duration cannot be changed using renew or change. For backwards compatibility, the default is 60, and the value is only used on an acquire operation.

lease_break_period

Optional. For a break operation, this is the proposed duration of seconds that the lease should continue before it is broken, between 0 and 60 seconds. This break period is only used if it is shorter than the time remaining on the lease. If longer, the time remaining on the lease is used. A new lease will not be available before the break period has expired, but the lease may be held for longer than the break period. If this header does not appear with a break operation, a fixed-duration lease breaks after the remaining lease period elapses, and an infinite lease breaks immediately.

proposed_lease_id

Optional for acquire, required for change. Proposed lease ID, in a GUID string format.

list_blobs

List blobs associated with the container

```
salt-cloud -f list_blobs my-azure container=mycontainer
```

container

The name of the storage container

prefix

Optional. Filters the results to return only blobs whose names begin with the specified prefix.

marker

Optional. A string value that identifies the portion of the list to be returned with the next list operation. The operation returns a marker value within the response body if the list returned was not complete. The marker value may then be used in a subsequent call to request the next set of list items. The marker value is opaque to the client.

maxresults

Optional. Specifies the maximum number of blobs to return, including all BlobPrefix elements. If the request does not specify maxresults or specifies a value greater than 5,000, the server will return up to 5,000 items. Setting maxresults to a value less than or equal to zero results in error response code 400 (Bad Request).

include

Optional. Specifies one or more datasets to include in the response. To specify more than one of these options on the URI, you must separate each option with a comma. Valid values are:

```
snapshots:
    Specifies that snapshots should be included in the
    enumeration. Snapshots are listed from oldest to newest in
    the response.
metadata:
    Specifies that blob metadata be returned in the response.
uncommittedblobs:
    Specifies that blobs for which blocks have been uploaded,
    but which have not been committed using Put Block List
    (REST API), be included in the response.
copy:
    Version 2012-02-12 and newer. Specifies that metadata
    related to any current or previous Copy Blob operation
    should be included in the response.
```

delimiter

Optional. When the request includes this parameter, the operation returns a BlobPrefix element in the response body that acts as a placeholder for all blobs whose names begin with the same substring up to the appearance of the delimiter character. The delimiter may be a single character or a string.

show_blob_service_properties

Show a blob's service properties

```
salt-cloud -f show_blob_service_properties my-azure
```

set_blob_service_properties

Sets the properties of a storage account's Blob service, including Windows Azure Storage Analytics. You can also use this operation to set the default request version for all incoming requests that do not have a version specified.

```
salt-cloud -f set_blob_service_properties my-azure
```

properties

a StorageServiceProperties object.

timeout

Optional. The timeout parameter is expressed in seconds.

show_blob_properties

Returns all user-defined metadata, standard HTTP properties, and system properties for the blob.

```
salt-cloud -f show_blob_properties my-azure container=mycontainer blob=myblob
```

container

Name of existing container.

blob

Name of existing blob.

lease_id

Required if the blob has an active lease.

set_blob_properties

Set a blob's properties

```
salt-cloud -f set_blob_properties my-azure
```

container

Name of existing container.

blob

Name of existing blob.

blob_cache_control

Optional. Modifies the cache control string for the blob.

blob_content_type

Optional. Sets the blob's content type.

blob_content_md5

Optional. Sets the blob's MD5 hash.

blob_content_encoding

Optional. Sets the blob's content encoding.

blob_content_language

Optional. Sets the blob's content language.

lease_id

Required if the blob has an active lease.

blob_content_disposition

Optional. Sets the blob's Content-Disposition header. The Content-Disposition response header field conveys additional information about how to process the response payload, and also can be used to attach additional metadata. For example, if set to attachment, it indicates that the user-agent should not display the response, but instead show a Save As dialog with a filename other than the blob name specified.

put_blob

Upload a blob

```
salt-cloud -f put_blob my-azure container=base name=top.sls blob_path=/srv/salt/top.
↪sls
salt-cloud -f put_blob my-azure container=base name=content.txt blob_content='Some
↪content'
```

container

Name of existing container.

name

Name of existing blob.

blob_path

The path on the local machine of the file to upload as a blob. Either this or blob_content must be specified.

blob_content

The actual content to be uploaded as a blob. Either this or blob_path must me specified.

cache_control

Optional. The Blob service stores this value but does not use or modify it.

content_language

Optional. Specifies the natural languages used by this resource.

content_md5

Optional. An MD5 hash of the blob content. This hash is used to verify the integrity of the blob during transport. When this header is specified, the storage service checks the hash that has arrived with the one that was sent. If the two hashes do not match, the operation will fail with error code 400 (Bad Request).

blob_content_type

Optional. Set the blob's content type.

blob_content_encoding

Optional. Set the blob's content encoding.

blob_content_language

Optional. Set the blob's content language.

blob_content_md5

Optional. Set the blob's MD5 hash.

blob_cache_control

Optional. Sets the blob's cache control.

meta_name_values

A dict containing name, value for metadata.

lease_id

Required if the blob has an active lease.

get_blob

Download a blob

```
salt-cloud -f get_blob my-azure container=base name=top.sls local_path=/srv/salt/top.
↪sls
salt-cloud -f get_blob my-azure container=base name=content.txt return_content=True
```

container

Name of existing container.

name

Name of existing blob.

local_path

The path on the local machine to download the blob to. Either this or return_content must be specified.

return_content

Whether or not to return the content directly from the blob. If specified, must be True or False. Either this or the local_path must be specified.

snapshot

Optional. The snapshot parameter is an opaque DateTime value that, when present, specifies the blob snapshot to retrieve.

lease_id

Required if the blob has an active lease.

progress_callback

callback for progress with signature function(current, total) where current is the number of bytes transferred so far, and total is the size of the blob.

max_connections

Maximum number of parallel connections to use when the blob size exceeds 64MB. Set to 1 to download the blob chunks sequentially. Set to 2 or more to download the blob chunks in parallel. This uses more system resources but will download faster.

max_retries

Number of times to retry download of blob chunk if an error occurs.

retry_wait

Sleep time in secs between retries.

14.7.3 Getting Started With Azure ARM

New in version 2016.11.0.

Azure is a cloud service by Microsoft providing virtual machines, SQL services, media services, and more. Azure ARM (aka, the Azure Resource Manager) is a next generation version of the Azure portal and API. This document describes how to use Salt Cloud to create a virtual machine on Azure ARM, with Salt installed.

More information about Azure is located at http://www.windowsazure.com/.

Dependencies

- azure >= 2.0.0rc6
- azure-common >= 1.1.4
- azure-mgmt >= 0.30.0rc6
- azure-mgmt-compute >= 0.33.0
- azure-mgmt-network >= 0.30.0rc6
- azure-mgmt-resource >= 0.30.0
- azure-mgmt-storage >= 0.30.0rc6
- azure-mgmt-web >= 0.30.0rc6
- azure-storage >= 0.32.0
- msrestazure >= 0.4.21
- A Microsoft Azure account
- Salt

Installation Tips

Because the `azure` library requires the `cryptography` library, which is compiled on-the-fly by `pip`, you may need to install the development tools for your operating system.

Before you install `azure` with `pip`, you should make sure that the required libraries are installed.

Debian

For Debian and Ubuntu, the following command will ensure that the required dependencies are installed:

```
sudo apt-get install build-essential libssl-dev libffi-dev python-dev
```

Red Hat

For Fedora and RHEL-derivatives, the following command will ensure that the required dependencies are installed:

```
sudo yum install gcc libffi-devel python-devel openssl-devel
```

Configuration

Set up the provider config at `/etc/salt/cloud.providers.d/azurearm.conf`:

```
# Note: This example is for /etc/salt/cloud.providers.d/azurearm.conf

my-azurearm-config:
  driver: azurearm
  master: salt.example.com
  subscription_id: 01234567-890a-bcde-f012-34567890abdc

  # https://apps.dev.microsoft.com/#/appList
  username: <username>@<subdomain>.onmicrosoft.com
  password: verybadpass
  location: westus
  resource_group: my_rg

  # Optional
  network_resource_group: my_net_rg
  cleanup_disks: True
  cleanup_vhds: True
  cleanup_data_disks: True
  cleanup_interfaces: True
  custom_data: 'This is custom data'
  expire_publisher_cache: 604800  # 7 days
  expire_offer_cache: 518400  # 6 days
  expire_sku_cache: 432000  # 5 days
  expire_version_cache: 345600  # 4 days
  expire_group_cache: 14400  # 4 hours
  expire_interface_cache: 3600  # 1 hour
  expire_network_cache: 3600  # 1 hour
```

Cloud Profiles

Set up an initial profile at `/etc/salt/cloud.profiles`:

```
azure-ubuntu-pass:
  provider: my-azure-config
  image: Canonical|UbuntuServer|14.04.5-LTS|14.04.201612050
  size: Standard_D1_v2
  location: eastus
  ssh_username: azureuser
  ssh_password: verybadpass

azure-ubuntu-key:
  provider: my-azure-config
  image: Canonical|UbuntuServer|14.04.5-LTS|14.04.201612050
  size: Standard_D1_v2
```

(continues on next page)

(continued from previous page)

```
  location: eastus
  ssh_username: azureuser
  ssh_publickeyfile: /path/to/ssh_public_key.pub

azure-win2012:
  provider: my-azure-config
  image: MicrosoftWindowsServer|WindowsServer|2012-R2-Datacenter|latest
  size: Standard_D1_v2
  location: westus
  win_username: azureuser
  win_password: verybadpass
```

These options are described in more detail below. Once configured, the profile can be realized with a salt command:

```
salt-cloud -p azure-ubuntu newinstance
```

This will create an salt minion instance named `newinstance` in Azure. If the command was executed on the salt-master, its Salt key will automatically be signed on the master.

Once the instance has been created with salt-minion installed, connectivity to it can be verified with Salt:

```
salt newinstance test.version
```

Profile Options

The following options are currently available for Azure ARM.

provider

The name of the provider as configured in */etc/salt/cloud.providers.d/azure.conf*.

image

Required. The name of the image to use to create a VM. Available images can be viewed using the following command:

```
salt-cloud --list-images my-azure-config
```

As you will see in `--list-images`, image names are comprised of the following fields, separated by the pipe (|) character:

```
publisher: For example, Canonical or MicrosoftWindowsServer
offer: For example, UbuntuServer or WindowsServer
sku: Such as 14.04.5-LTS or 2012-R2-Datacenter
version: Such as 14.04.201612050 or latest
```

It is possible to specify the URL or resource ID path of a custom image that you have access to, such as:

```
https://<mystorage>.blob.core.windows.net/system/Microsoft.Compute/Images/<mystorage>/
↪template-osDisk.01234567-890a-bcdef0123-4567890abcde.vhd
```

or:

```
/subscriptions/XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX/resourceGroups/myRG/providers/
↪Microsoft.Compute/images/myImage
```

size

Required. The name of the size to use to create a VM. Available sizes can be viewed using the following command:

```
salt-cloud --list-sizes my-azure-config
```

location

Required. The name of the location to create a VM in. Available locations can be viewed using the following command:

```
salt-cloud --list-locations my-azure-config
```

ssh_username

Required for Linux. The admin user to add on the instance. It is also used to log into the newly-created VM to install Salt.

ssh_keyfile

Required if using SSH key authentication. The path on the Salt master to the SSH private key used during the minion bootstrap process.

ssh_publickeyfile

Use either `ssh_publickeyfile` or `ssh_password`. The path on the Salt master to the SSH public key which will be pushed to the Linux VM.

ssh_password

Use either `ssh_publickeyfile` or `ssh_password`. The password for the admin user on the newly-created Linux virtual machine.

win_username

Required for Windows. The user to use to log into the newly-created Windows VM to install Salt.

win_password

Required for Windows. The password to use to log into the newly-created Windows VM to install Salt.

win_installer

Required for Windows. The path to the Salt installer to be uploaded.

resource_group

Required. The resource group that all VM resources (VM, network interfaces, etc) will be created in.

network_resource_group

Optional. If specified, then the VM will be connected to the virtual network in this resource group, rather than the parent resource group of the instance. The VM interfaces and IPs will remain in the configured `resource_group` with the VM.

network

Required. The virtual network that the VM will be spun up in.

subnet

Optional. The subnet inside the virtual network that the VM will be spun up in. Default is `default`.

allocate_public_ip

Optional. Default is `False`. If set to `True`, a public IP will be created and assigned to the VM.

load_balancer

Optional. The load-balancer for the VM's network interface to join. If specified the backend_pool option need to be set.

backend_pool

Optional. Required if the load_balancer option is set. The load-balancer's Backend Pool the VM's network interface will join.

iface_name

Optional. The name to apply to the VM's network interface. If not supplied, the value will be set to `<VM name>-iface0`.

dns_servers

Optional. A **list** of the DNS servers to configure for the network interface (will be set on the VM by the DHCP of the VNET).

```
my-azurearm-profile:
  provider: azurearm-provider
  network: mynetwork
  dns_servers:
    - 10.1.1.4
    - 10.1.1.5
```

availability_set

Optional. If set, the VM will be added to the specified availability set.

volumes

Optional. A list of dictionaries describing data disks to attach to the instance can be specified using this setting. The data disk dictionaries are passed entirely to the Azure DataDisk object, so ad-hoc options can be handled as long as they are valid properties of the object.

```
volumes:
- disk_size_gb: 50
  caching: ReadWrite
- disk_size_gb: 100
  caching: ReadWrite
  managed_disk:
    storage_account_type: Standard_LRS
```

cleanup_disks

Optional. Default is `False`. If set to `True`, disks will be cleaned up when the VM that they belong to is deleted.

cleanup_vhds

Optional. Default is `False`. If set to `True`, VHDs will be cleaned up when the VM and disk that they belong to are deleted. Requires `cleanup_disks` to be set to `True`.

cleanup_data_disks

Optional. Default is `False`. If set to `True`, data disks (non-root volumes) will be cleaned up whtn the VM that they are attached to is deleted. Requires `cleanup_disks` to be set to `True`.

cleanup_interfaces

Optional. Default is `False`. Normally when a VM is deleted, its associated interfaces and IPs are retained. This is useful if you expect the deleted VM to be recreated with the same name and network settings. If you would like interfaces and IPs to be deleted when their associated VM is deleted, set this to `True`.

userdata

Optional. Any custom cloud data that needs to be specified. How this data is used depends on the operating system and image that is used. For instance, Linux images that use `cloud-init` will import this data for use with that program. Some Windows images will create a file with a copy of this data, and others will ignore it. If a Windows image creates a file, then the location will depend upon the version of Windows. This will be ignored if the `userdata_file` is specified.

userdata_file

Optional. The path to a file to be read and submitted to Azure as user data. How this is used depends on the operating system that is being deployed. If used, any `userdata` setting will be ignored.

userdata_sendkeys

Optional. Set to `True` in order to generate salt minion keys and provide them as variables to the userdata script when running it through the template renderer. The keys can be referenced as `{{opts['priv_key']}}` and `{{opts['pub_key']}}`.

userdata_template

Optional. Enter the renderer, such as `jinja`, to be used for the userdata script template.

wait_for_ip_timeout

Optional. Default is `600`. When waiting for a VM to be created, Salt Cloud will attempt to connect to the VM's IP address until it starts responding. This setting specifies the maximum time to wait for a response.

wait_for_ip_interval

Optional. Default is `10`. How long to wait between attempts to connect to the VM's IP.

wait_for_ip_interval_multiplier

Optional. Default is `1`. Increase the interval by this multiplier after each request; helps with throttling.

expire_publisher_cache

Optional. Default is `604800`. When fetching image data using `--list-images`, a number of web calls need to be made to the Azure ARM API. This is normally very fast when performed using a VM that exists inside Azure itself, but can be very slow when made from an external connection.

By default, the publisher data will be cached, and only updated every `604800` seconds (7 days). If you need the publisher cache to be updated at a different frequency, change this setting. Setting it to `0` will turn off the publisher cache.

expire_offer_cache

Optional. Default is `518400`. See `expire_publisher_cache` for details on why this exists.

By default, the offer data will be cached, and only updated every `518400` seconds (6 days). If you need the offer cache to be updated at a different frequency, change this setting. Setting it to `0` will turn off the publiser cache.

expire_sku_cache

Optional. Default is `432000`. See `expire_publisher_cache` for details on why this exists.

By default, the sku data will be cached, and only updated every `432000` seconds (5 days). If you need the sku cache to be updated at a different frequency, change this setting. Setting it to `0` will turn off the sku cache.

expire_version_cache

Optional. Default is `345600`. See `expire_publisher_cache` for details on why this exists.

By default, the version data will be cached, and only updated every `345600` seconds (4 days). If you need the version cache to be updated at a different frequency, change this setting. Setting it to `0` will turn off the version cache.

expire_group_cache

Optional. Default is `14400`. See `expire_publisher_cache` for details on why this exists.

By default, the resource group data will be cached, and only updated every `14400` seconds (4 hours). If you need the resource group cache to be updated at a different frequency, change this setting. Setting it to `0` will turn off the resource group cache.

expire_interface_cache

Optional. Default is `3600`. See `expire_publisher_cache` for details on why this exists.

By default, the interface data will be cached, and only updated every `3600` seconds (1 hour). If you need the interface cache to be updated at a different frequency, change this setting. Setting it to `0` will turn off the interface cache.

expire_network_cache

Optional. Default is `3600`. See `expire_publisher_cache` for details on why this exists.

By default, the network data will be cached, and only updated every `3600` seconds (1 hour). If you need the network cache to be updated at a different frequency, change this setting. Setting it to `0` will turn off the network cache.

Other Options

Other options relevant to Azure ARM.

storage_account

Required for actions involving an Azure storage account.

storage_key

Required for actions involving an Azure storage account.

Show Instance

This action is a thin wrapper around `--full-query`, which displays details on a single instance only. In an environment with several machines, this will save a user from having to sort through all instance data, just to examine a single instance.

```
salt-cloud -a show_instance myinstance
```

14.7.4 Getting Started with CloudStack

CloudStack is one the most popular cloud projects. It's an open source project to build public and/or private clouds. You can use Salt Cloud to launch CloudStack instances.

Dependencies

- Libcloud >= 0.13.2

Configuration

Using Salt for CloudStack, requires an `API key` and a `secret key` along with the API address endpoint information.

```
# Note: This example is for /etc/salt/cloud.providers or any file in the
# /etc/salt/cloud.providers.d/ directory.

exoscale:
  driver: cloudstack
  host: api.exoscale.com
  path: /compute
  apikey: EXOAPIKEY
  secretkey: EXOSECRETKEYINYOURACCOUNT
```

Note: Changed in version 2015.8.0.

The `provider` parameter in cloud provider definitions was renamed to `driver`. This change was made to avoid confusion with the `provider` parameter that is used in cloud profile definitions. Cloud provider definitions now use `driver` to refer to the Salt cloud module that provides the underlying functionality to connect to a cloud host, while cloud profiles continue to use `provider` to refer to provider configurations that you define.

Profiles

Cloud Profiles

Set up an initial profile at `/etc/salt/cloud.profiles` or in the `/etc/salt/cloud.profiles.d/` directory:

```
exoscale-ubuntu:
  provider: exoscale-config
  image: Linux Ubuntu 18.04
  size: Small
  location: ch-gva-2
  ssh_username: ubuntu
```

Locations can be obtained using the `--list-locations` option for the `salt-cloud` command:

```
# salt-cloud --list-locations exoscale-config
exoscale:
    ----------
    cloudstack:
        ----------
        ch-dk-2:
            ----------
            country:
                Unknown
            driver:
            id:
                91e5e9e4-c9ed-4b76-bee4-427004b3baf9
            name:
                ch-dk-2
        ch-gva-2:
            ----------
            country:
                Unknown
            driver:
            id:
                1128bd56-b4d9-4ac6-a7b9-c715b187ce11
            name:
                ch-gva-2
```

Sizes can be obtained using the `--list-sizes` option for the `salt-cloud` command:

```
# salt-cloud --list-sizes exoscale
exoscale:
    ----------
    cloudstack:
        ----------
        Extra-large:
            ----------
            bandwidth:
                0
            disk:
                0
            driver:
            extra:
                ----------
                cpu:
                    4
            get_uuid:
            id:
                350dc5ea-fe6d-42ba-b6c0-efb8b75617ad
            name:
                Extra-large
            price:
```

(continues on next page)

(continued from previous page)

```
                0
            ram:
                16384
            uuid:
                edb4cd4ae14bbf152d451b30c4b417ab095a5bfe
...SNIP...
```

Images can be obtained using the `--list-images` option for the `salt-cloud` command:

```
# salt-cloud --list-images exoscale
exoscale:
    ----------
    cloudstack:
        ----------
        Linux CentOS 6.6 64-bit:
            ----------
            driver:
            extra:
                ----------
                displaytext:
                    Linux CentOS 6.6 64-bit 10G Disk (2014-12-01-bac8e0)
                format:
                    QCOW2
                hypervisor:
                    KVM
                os:
                    Other PV (64-bit)
                size:
                    10737418240
            get_uuid:
            id:
                aa69ae64-1ea9-40af-8824-c2c3344e8d7c
            name:
                Linux CentOS 6.6 64-bit
            uuid:
                f26b4f54ec8591abdb6b5feb3b58f720aa438fee
...SNIP...
```

CloudStack specific settings

securitygroup

New in version 2017.7.0.

You can specify a list of security groups (by name or id) that should be assigned to the VM:

```
exoscale:
  provider: cloudstack
  securitygroup:
    - default
    - salt-master
```

14.7.5 Getting Started With DigitalOcean

DigitalOcean is a public cloud host that specializes in Linux instances.

Configuration

Using Salt for DigitalOcean requires a `personal_access_token`, an `ssh_key_file`, and at least one SSH key name in `ssh_key_names`. More `ssh_key_names` can be added by separating each key with a comma. The `personal_access_token` can be found in the DigitalOcean web interface in the ”Apps & API” section. The SSH key name can be found under the ”SSH Keys” section.

```
# Note: This example is for /etc/salt/cloud.providers or any file in the
# /etc/salt/cloud.providers.d/ directory.

my-digitalocean-config:
  driver: digitalocean
  personal_access_token: xxx
  ssh_key_file: /path/to/ssh/key/file
  ssh_key_names: my-key-name,my-key-name-2
  location: New York 1
```

Note: Changed in version 2015.8.0.

The `provider` parameter in cloud provider definitions was renamed to `driver`. This change was made to avoid confusion with the `provider` parameter that is used in cloud profile definitions. Cloud provider definitions now use `driver` to refer to the Salt cloud module that provides the underlying functionality to connect to a cloud host, while cloud profiles continue to use `provider` to refer to provider configurations that you define.

Profiles

Cloud Profiles

Set up an initial profile at `/etc/salt/cloud.profiles` or in the `/etc/salt/cloud.profiles.d/` directory:

```
digitalocean-ubuntu:
  provider: my-digitalocean-config
  image: 14.04 x64
  size: 512MB
  location: New York 1
  private_networking: True
  backups_enabled: True
  ipv6: True
  create_dns_record: True
  userdata_file: /etc/salt/cloud.userdata.d/setup
  tags:
    - tag1
    - tag2
    - tag3
```

Locations can be obtained using the `--list-locations` option for the `salt-cloud` command:

```
# salt-cloud --list-locations my-digitalocean-config
my-digitalocean-config:
    ----------
    digitalocean:
        ----------
        Amsterdam 1:
            ----------
            available:
                False
            features:
                [u'backups']
            name:
                Amsterdam 1
            sizes:
                []
            slug:
                ams1
...SNIP...
```

Sizes can be obtained using the `--list-sizes` option for the `salt-cloud` command:

```
# salt-cloud --list-sizes my-digitalocean-config
my-digitalocean-config:
    ----------
    digitalocean:
        ----------
        512MB:
            ----------
            cost_per_hour:
                0.00744
            cost_per_month:
                5.0
            cpu:
                1
            disk:
                20
            id:
                66
            memory:
                512
            name:
                512MB
            slug:
                None
...SNIP...
```

Images can be obtained using the `--list-images` option for the `salt-cloud` command:

```
# salt-cloud --list-images my-digitalocean-config
my-digitalocean-config:
    ----------
    digitalocean:
        ----------
        10.1:
            ----------
            created_at:
                2015-01-20T20:04:34Z
```

(continues on next page)

(continued from previous page)

```
            distribution:
                FreeBSD
            id:
                10144573
            min_disk_size:
                20
            name:
                10.1
            public:
                True
...SNIP...
```

Profile Specifics:

ssh_username

If using a FreeBSD image from DigitalOcean, you'll need to set the `ssh_username` setting to `freebsd` in your profile configuration.

```
digitalocean-freebsd:
  provider: my-digitalocean-config
  image: 10.2
  size: 512MB
  ssh_username: freebsd
```

userdata_file

New in version 2016.11.6.

Use *userdata_file* to specify the userdata file to upload for use with cloud-init if available.

```
my-openstack-config:
  # Pass userdata to the instance to be created
  userdata_file: /etc/salt/cloud-init/packages.yml
```

```
my-do-config:
  # Pass userdata to the instance to be created
  userdata_file: /etc/salt/cloud-init/packages.yml
  userdata_template: jinja
```

If no `userdata_template` is set in the cloud profile, then the master configuration will be checked for a *`userdata_template`* value. If this is not set, then no templating will be performed on the userdata_file.

To disable templating in a cloud profile when a *`userdata_template`* has been set in the master configuration file, simply set `userdata_template` to `False` in the cloud profile:

```
my-do-config:
  # Pass userdata to the instance to be created
  userdata_file: /etc/salt/cloud-init/packages.yml
  userdata_template: False
```

Miscellaneous Information

Note: DigitalOcean's concept of `Applications` is nothing more than a pre-configured instance (same as a normal Droplet). You will find examples such `Docker 0.7 Ubuntu 13.04 x64` and `Wordpress on Ubuntu 12.10` when using the `--list-images` option. These names can be used just like the rest of the standard instances when specifying an image in the cloud profile configuration.

Note: If your domain's DNS is managed with DigitalOcean, and your minion name matches your DigitalOcean managed DNS domain, you can automatically create A and AAA records for newly created droplets. Use `create_dns_record: True` in your config to enable this. Adding `delete_dns_record: True` to also delete records when a droplet is destroyed is optional. Due to limitations in salt-cloud design, the destroy code does not have access to the VM config data. WHETHER YOU ADD `create_dns_record: True` OR NOT, salt-cloud WILL attempt to delete your DNS records if the minion name matches. This will prevent advertising any recycled IP addresses for destroyed minions.

Note: If you need to perform the bootstrap using the local interface for droplets, this can be done by setting `ssh_interface: private` in your config. By default the salt-cloud script would run on the public interface however if firewall is preventing the connection to the Droplet over the public interface you might need to set this option to connect via private interface. Also, to use this feature `private_networking: True` must be set in the config.

Note: Additional documentation is available from DigitalOcean.

14.7.6 Getting Started With Dimension Data Cloud

Dimension Data are a global IT Services company and form part of the NTT Group. Dimension Data provide IT-as-a-Service to customers around the globe on their cloud platform (Compute as a Service). The CaaS service is available either on one of the public cloud instances or as a private instance on premises.

http://cloud.dimensiondata.com/

CaaS has its own non-standard API , SaltStack provides a wrapper on top of this API with common methods with other IaaS solutions and Public cloud providers. Therefore, you can use the Dimension Data module to communicate with both the public and private clouds.

Dependencies

This driver requires the Python `apache-libcloud` and `netaddr` library to be installed.

Configuration

When you instantiate a driver you need to pass the following arguments to the driver constructor:

- `user_id` - Your Dimension Data Cloud username
- `key` - Your Dimension Data Cloud password
- `region` - The region key, one of the possible region keys

Possible regions:

- `dd-na` : Dimension Data North America (USA)
- `dd-eu` : Dimension Data Europe
- `dd-af` : Dimension Data Africa
- `dd-au` : Dimension Data Australia
- `dd-latam` : Dimension Data Latin America
- `dd-ap` : Dimension Data Asia Pacific
- `dd-canada` : Dimension Data Canada region

```
# Note: This example is for /etc/salt/cloud.providers or any file in the
# /etc/salt/cloud.providers.d/ directory.

my-dimensiondata-config:
  user_id: my_username
  key: myPassword!
  region: dd-na
  driver: dimensiondata
```

Note: In version 2015.8.0, the `provider` parameter in cloud provider definitions was renamed to `driver`. This change was made to avoid confusion with the `provider` parameter that is used in cloud profile definitions. Cloud provider definitions now use `driver` to refer to the Salt cloud module that provides the underlying functionality to connect to a cloud host, while cloud profiles continue to use `provider` to refer to provider configurations that you define.

Profiles

Cloud Profiles

Dimension Data images have an inbuilt size configuration, there is no list of sizes (although, if the command –list-sizes is run a default will be returned).

Images can be obtained using the `--list-images` option for the `salt-cloud` command:

```
# salt-cloud --list-images my-dimensiondata-config
my-dimensiondata-config:
----------
dimensiondata:
    ----------
    CSfM SharePoint 2013 Trial:
        ----------
        driver:
        extra:
            ----------
            OS_displayName:
                WIN2012R2S/64
            OS_type:
                None
            cpu:
            created:
```

(continues on next page)

(continued from previous page)

```
                2015-03-19T18:36:06.000Z
            description:
                Windows 2012 R2 Standard 64-bit installed with SharePoint 2013 and
↪Visual Studio 2013 Pro (Trial Version)
            location:
            memoryGb:
                12
            osImageKey:
                T-WIN-2012R2-STD-SP2013-VS2013-64-4-12-100
        get_uuid:
        id:
            0df4677e-d380-4e9b-9469-b529ee0214c5
        name:
            CSfM SharePoint 2013 Trial
        uuid:
            28c077f1be970ee904541407b377e3ff87a9ac69
    CentOS 5 32-bit 2 CPU:
        ----------
        driver:
        extra:
            ----------
            OS_displayName:
                CENTOS5/32
            OS_type:
                None
            cpu:
            created:
                2015-10-21T14:52:29.000Z
            description:
                CentOS Release 5.11 32-bit
            location:
            memoryGb:
                4
            osImageKey:
                T-CENT-5-32-2-4-10
        get_uuid:
        id:
            a8046bd1-04ea-4668-bf32-bf8d5540faed
        name:
            CentOS 5 32-bit 2 CPU
        uuid:
            4d7dd59929fed6f4228db861b609da64997773a7

...SNIP...
```

Locations can be obtained using the `--list-locations` option for the `salt-cloud` command:

```
my-dimensiondata-config:
    ----------
    dimensiondata:
        ----------
        Australia - Melbourne:
            ----------
            country:
                Australia
            driver:
```

(continues on next page)

(continued from previous page)

```
            id:
                AU2
            name:
                Australia - Melbourne
        Australia - Melbourne MCP2:
            ----------
            country:
                Australia
            driver:
            id:
                AU10
            name:
                Australia - Melbourne MCP2
        Australia - Sydney:
            ----------
            country:
                Australia
            driver:
            id:
                AU1
            name:
                Australia - Sydney
        Australia - Sydney MCP2:
            ----------
            country:
                Australia
            driver:
            id:
                AU9
            name:
                Australia - Sydney MCP2
        New Zealand:
            ----------
            country:
                New Zealand
            driver:
            id:
                AU8
            name:
                New Zealand
        New_Zealand:
            ----------
            country:
                New Zealand
            driver:
            id:
                AU11
            name:
                New_Zealand
```

Note: Dimension Data Cloud REST API documentation is available from Dimension Data MCP 2.

14.7.7 Getting Started With AWS EC2

Amazon EC2 is a very widely used public cloud platform and one of the core platforms Salt Cloud has been built to support.

Previously, the suggested driver for AWS EC2 was the `aws` driver. This has been deprecated in favor of the `ec2` driver. Configuration using the old `aws` driver will still function, but that driver is no longer in active development.

Dependencies

This driver requires the Python `requests` library to be installed.

Configuration

The following example illustrates some of the options that can be set. These parameters are discussed in more detail below.

```
# Note: This example is for /etc/salt/cloud.providers or any file in the
# /etc/salt/cloud.providers.d/ directory.

my-ec2-southeast-public-ips:
  # Set up the location of the salt master
  #
  minion:
    master: saltmaster.example.com

  # Set up grains information, which will be common for all nodes
  # using this provider
  grains:
    node_type: broker
    release: 1.0.1

  # Specify whether to use public or private IP for deploy script.
  #
  # Valid options are:
  #     private_ips - The salt-cloud command is run inside the EC2
  #     public_ips - The salt-cloud command is run outside of EC2
  #
  ssh_interface: public_ips

  # Optionally configure the Windows credential validation number of
  # retries and delay between retries.  This defaults to 10 retries
  # with a one second delay betwee retries
  win_deploy_auth_retries: 10
  win_deploy_auth_retry_delay: 1

  # Set the EC2 access credentials (see below)
  #
  id: 'use-instance-role-credentials'
  key: 'use-instance-role-credentials'

  # If 'role_arn' is specified the above credentials are used to
  # to assume to the role. By default, role_arn is set to None.
  role_arn: arn:aws:iam::012345678910:role/SomeRoleName

  # Make sure this key is owned by corresponding user (default 'salt') with
↪permissions 0400.
```

(continues on next page)

(continued from previous page)

```
  #
  private_key: /etc/salt/my_test_key.pem
  keyname: my_test_key
  securitygroup: default

  # Optionally configure default region
  # Use salt-cloud --list-locations <provider> to obtain valid regions
  #
  location: ap-southeast-1
  availability_zone: ap-southeast-1b

  # Configure which user to use to run the deploy script. This setting is
  # dependent upon the AMI that is used to deploy. It is usually safer to
  # configure this individually in a profile, than globally. Typical users
  # are:
  #
  # Amazon Linux -> ec2-user
  # RHEL         -> ec2-user
  # CentOS       -> ec2-user
  # Ubuntu       -> ubuntu
  # Debian       -> admin
  #
  ssh_username: ec2-user

  # Optionally add an IAM profile
  iam_profile: 'arn:aws:iam::123456789012:instance-profile/ExampleInstanceProfile'

  driver: ec2


my-ec2-southeast-private-ips:
  # Set up the location of the salt master
  #
  minion:
    master: saltmaster.example.com

  # Specify whether to use public or private IP for deploy script.
  #
  # Valid options are:
  #     private_ips - The salt-master is also hosted with EC2
  #     public_ips - The salt-master is hosted outside of EC2
  #
  ssh_interface: private_ips

  # Optionally configure the Windows credential validation number of
  # retries and delay between retries.  This defaults to 10 retries
  # with a one second delay betwee retries
  win_deploy_auth_retries: 10
  win_deploy_auth_retry_delay: 1

  # Set the EC2 access credentials (see below)
  #
  id: 'use-instance-role-credentials'
  key: 'use-instance-role-credentials'

  # Make sure this key is owned by root with permissions 0400.
```

(continues on next page)

(continued from previous page)

```
#
private_key: /etc/salt/my_test_key.pem
keyname: my_test_key

# This one should NOT be specified if VPC was not configured in AWS to be
# the default. It might cause an error message which says that network
# interfaces and an instance-level security groups may not be specified
# on the same request.
#
securitygroup: default

# Optionally configure default region
#
location: ap-southeast-1
availability_zone: ap-southeast-1b

# Configure which user to use to run the deploy script. This setting is
# dependent upon the AMI that is used to deploy. It is usually safer to
# configure this individually in a profile, than globally. Typical users
# are:
#
# Amazon Linux -> ec2-user
# RHEL         -> ec2-user
# CentOS       -> ec2-user
# Ubuntu       -> ubuntu
#
ssh_username: ec2-user

# Optionally add an IAM profile
iam_profile: 'my other profile name'

driver: ec2
```

Note: Changed in version 2015.8.0.

The `provider` parameter in cloud provider definitions was renamed to `driver`. This change was made to avoid confusion with the `provider` parameter that is used in cloud profile definitions. Cloud provider definitions now use `driver` to refer to the Salt cloud module that provides the underlying functionality to connect to a cloud host, while cloud profiles continue to use `provider` to refer to provider configurations that you define.

Access Credentials

The `id` and `key` settings may be found in the Security Credentials area of the AWS Account page:

https://portal.aws.amazon.com/gp/aws/securityCredentials

Both are located in the Access Credentials area of the page, under the Access Keys tab. The `id` setting is labeled Access Key ID, and the `key` setting is labeled Secret Access Key.

Note: if either `id` or `key` is set to 'use-instance-role-credentials' it is assumed that Salt is running on an AWS instance, and the instance role credentials will be retrieved and used. Since both the `id` and `key` are required parameters for the AWS ec2 provider, it is recommended to set both to 'use-instance-role-credentials' for this functionality.

A ”static” and ”permanent” Access Key ID and Secret Key can be specified, but this is not recommended. Instance role keys are rotated on a regular basis, and are the recommended method of specifying AWS credentials.

Windows Deploy Timeouts

For Windows instances, it may take longer than normal for the instance to be ready. In these circumstances, the provider configuration can be configured with a `win_deploy_auth_retries` and/or a `win_deploy_auth_retry_delay` setting, which default to 10 retries and a one second delay between retries. These retries and timeouts relate to validating the Administrator password once AWS provides the credentials via the AWS API.

Key Pairs

In order to create an instance with Salt installed and configured, a key pair will need to be created. This can be done in the EC2 Management Console, in the Key Pairs area. These key pairs are unique to a specific region. Keys in the us-east-1 region can be configured at:

https://console.aws.amazon.com/ec2/home?region=us-east-1#s=KeyPairs

Keys in the us-west-1 region can be configured at

https://console.aws.amazon.com/ec2/home?region=us-west-1#s=KeyPairs

…and so on. When creating a key pair, the browser will prompt to download a pem file. This file must be placed in a directory accessible by Salt Cloud, with permissions set to either 0400 or 0600.

Security Groups

An instance on EC2 needs to belong to a security group. Like key pairs, these are unique to a specific region. These are also configured in the EC2 Management Console. Security groups for the us-east-1 region can be configured at:

https://console.aws.amazon.com/ec2/home?region=us-east-1#s=SecurityGroups

…and so on.

A security group defines firewall rules which an instance will adhere to. If the salt-master is configured outside of EC2, the security group must open the SSH port (usually port 22) in order for Salt Cloud to install Salt.

IAM Profile

Amazon EC2 instances support the concept of an instance profile, which is a logical container for the IAM role. At the time that you launch an EC2 instance, you can associate the instance with an instance profile, which in turn corresponds to the IAM role. Any software that runs on the EC2 instance is able to access AWS using the permissions associated with the IAM role.

Scaffolding the profile is a 2-step configuration process:

1. Configure an IAM Role from the IAM Management Console.
2. Attach this role to a new profile. It can be done with the AWS CLI:

```
> aws iam create-instance-profile --instance-profile-name PROFILE_NAME
> aws iam add-role-to-instance-profile --instance-profile-name PROFILE_
↪NAME --role-name ROLE_NAME
```

Once the profile is created, you can use the **PROFILE_NAME** to configure your cloud profiles.

Cloud Profiles

Set up an initial profile at `/etc/salt/cloud.profiles`:

```
base_ec2_private:
  provider: my-ec2-southeast-private-ips
  image: ami-e565ba8c
  size: t2.micro
  ssh_username: ec2-user

base_ec2_public:
  provider: my-ec2-southeast-public-ips
  image: ami-e565ba8c
  size: t2.micro
  ssh_username: ec2-user

base_ec2_db:
  provider: my-ec2-southeast-public-ips
  image: ami-e565ba8c
  size: m1.xlarge
  ssh_username: ec2-user
  volumes:
    - { size: 10, device: /dev/sdf }
    - { size: 10, device: /dev/sdg, type: io1, iops: 1000 }
    - { size: 10, device: /dev/sdh, type: io1, iops: 1000 }
    - { size: 10, device: /dev/sdi, tags: {"Environment": "production"} }
  # optionally add tags to profile:
  tag: {'Environment': 'production', 'Role': 'database'}
  # force grains to sync after install
  sync_after_install: grains

base_ec2_vpc:
  provider: my-ec2-southeast-public-ips
  image: ami-a73264ce
  size: m1.xlarge
  ssh_username: ec2-user
  script:  /etc/salt/cloud.deploy.d/user_data.sh
  network_interfaces:
    - DeviceIndex: 0
      PrivateIpAddresses:
        - Primary: True
      #auto assign public ip (not EIP)
      AssociatePublicIpAddress: True
      SubnetId: subnet-813d4bbf
      SecurityGroupId:
        - sg-750af413
  del_root_vol_on_destroy: True
  del_all_vols_on_destroy: True
  volumes:
    - { size: 10, device: /dev/sdf }
    - { size: 10, device: /dev/sdg, type: io1, iops: 1000 }
    - { size: 10, device: /dev/sdh, type: io1, iops: 1000 }
  tag: {'Environment': 'production', 'Role': 'database'}
  sync_after_install: grains
```

The profile can now be realized with a salt command:

```
# salt-cloud -p base_ec2 ami.example.com
# salt-cloud -p base_ec2_public ami.example.com
# salt-cloud -p base_ec2_private ami.example.com
```

This will create an instance named `ami.example.com` in EC2. The minion that is installed on this instance will have an `id` of `ami.example.com`. If the command was executed on the salt-master, its Salt key will automatically be signed on the master.

Once the instance has been created with salt-minion installed, connectivity to it can be verified with Salt:

```
# salt 'ami.example.com' test.version
```

Required Settings

The following settings are always required for EC2:

```
# Set the EC2 login data
my-ec2-config:
  id: HJGRYCILJLKJYG
  key: 'kdjgfsgm;woormgl/aserigjksjdhasdfgn'
  keyname: test
  securitygroup: quick-start
  private_key: /root/test.pem
  driver: ec2
```

Optional Settings

EC2 allows a userdata file to be passed to the instance to be created. This functionality was added to Salt in the 2015.5.0 release.

```
my-ec2-config:
  # Pass userdata to the instance to be created
  userdata_file: /etc/salt/my-userdata-file
```

Note: From versions 2016.11.0 and 2016.11.3, this file was passed through the master's *renderer* to template it. However, this caused issues with non-YAML data, so templating is no longer performed by default. To template the userdata_file, add a `userdata_template` option to the cloud profile:

```
my-ec2-config:
  # Pass userdata to the instance to be created
  userdata_file: /etc/salt/my-userdata-file
  userdata_template: jinja
```

If no `userdata_template` is set in the cloud profile, then the master configuration will be checked for a *userdata_template* value. If this is not set, then no templating will be performed on the userdata_file.

To disable templating in a cloud profile when a *userdata_template* has been set in the master configuration file, simply set `userdata_template` to `False` in the cloud profile:

```
my-ec2-config:
  # Pass userdata to the instance to be created
  userdata_file: /etc/salt/my-userdata-file
  userdata_template: False
```

EC2 allows a location to be set for servers to be deployed in. Availability zones exist inside regions, and may be added to increase specificity.

```yaml
my-ec2-config:
  # Optionally configure default region
  location: ap-southeast-1
  availability_zone: ap-southeast-1b
```

EC2 instances can have a public or private IP, or both. When an instance is deployed, Salt Cloud needs to log into it via SSH to run the deploy script. By default, the public IP will be used for this. If the salt-cloud command is run from another EC2 instance, the private IP should be used.

```yaml
my-ec2-config:
  # Specify whether to use public or private IP for deploy script
  # private_ips or public_ips
  ssh_interface: public_ips
```

Many EC2 instances do not allow remote access to the root user by default. Instead, another user must be used to run the deploy script using sudo. Some common usernames include ec2-user (for Amazon Linux), ubuntu (for Ubuntu instances), admin (official Debian) and bitnami (for images provided by Bitnami).

```yaml
my-ec2-config:
  # Configure which user to use to run the deploy script
  ssh_username: ec2-user
```

Multiple usernames can be provided, in which case Salt Cloud will attempt to guess the correct username. This is mostly useful in the main configuration file:

```yaml
my-ec2-config:
  ssh_username:
    - ec2-user
    - ubuntu
    - admin
    - bitnami
```

Multiple security groups can also be specified in the same fashion:

```yaml
my-ec2-config:
  securitygroup:
    - default
    - extra
```

EC2 instances can be added to an AWS Placement Group by specifying the `placementgroup` option:

```yaml
my-ec2-config:
  placementgroup: my-aws-placement-group
```

Your instances may optionally make use of EC2 Spot Instances. The following example will request that spot instances be used and your maximum bid will be $0.10. Keep in mind that different spot prices may be needed based on the current value of the various EC2 instance sizes. You can check current and past spot instance pricing via the EC2 API or AWS Console.

```yaml
my-ec2-config:
  spot_config:
    spot_price: 0.10
```

You can optionally specify tags to apply to the EC2 spot instance request. A spot instance request itself is an object in AWS. The following example will set two tags on the spot instance request.

```
my-ec2-config:
  spot_config:
    spot_price: 0.10
    tag:
      tag0: value
      tag1: value
```

By default, the spot instance type is set to 'one-time', meaning it will be launched and, if it's ever terminated for whatever reason, it will not be recreated. If you would like your spot instances to be relaunched after a termination (by you or AWS), set the `type` to 'persistent'.

NOTE: Spot instances are a great way to save a bit of money, but you do run the risk of losing your spot instances if the current price for the instance size goes above your maximum bid.

The following parameters may be set in the cloud configuration file to control various aspects of the spot instance launching:

- `wait_for_spot_timeout`: seconds to wait before giving up on spot instance launch (default=600)
- `wait_for_spot_interval`: seconds to wait in between polling requests to determine if a spot instance is available (default=30)
- `wait_for_spot_interval_multiplier`: a multiplier to add to the interval in between requests, which is useful if AWS is throttling your requests (default=1)
- `wait_for_spot_max_failures`: maximum number of failures before giving up on launching your spot instance (default=10)

If you find that you're being throttled by AWS while polling for spot instances, you can set the following in your core cloud configuration file that will double the polling interval after each request to AWS.

```
wait_for_spot_interval: 1
wait_for_spot_interval_multiplier: 2
```

See the AWS Spot Instances documentation for more information.

Block device mappings enable you to specify additional EBS volumes or instance store volumes when the instance is launched. This setting is also available on each cloud profile. Note that the number of instance stores varies by instance type. If more mappings are provided than are supported by the instance type, mappings will be created in the order provided and additional mappings will be ignored. Consult the AWS documentation for a listing of the available instance stores, and device names.

```
my-ec2-config:
  block_device_mappings:
    - DeviceName: /dev/sdb
      VirtualName: ephemeral0
    - DeviceName: /dev/sdc
      VirtualName: ephemeral1
```

You can also use block device mappings to change the size of the root device at the provisioning time. For example, assuming the root device is '/dev/sda', you can set its size to 100G by using the following configuration.

```
my-ec2-config:
  block_device_mappings:
    - DeviceName: /dev/sda
      Ebs.VolumeSize: 100
```

(continues on next page)

(continued from previous page)

```
      Ebs.VolumeType: gp2
      Ebs.SnapshotId: dummy0
    - DeviceName: /dev/sdb
      # required for devices > 2TB
      Ebs.VolumeType: gp2
      Ebs.VolumeSize: 3001
```

Tagging of block devices can be set on a per device basis. For example, you may have multiple devices defined in your block_device_mappings structure. You have the option to set tags on any of one device or all of them as shown in the following configuration.

```
my-ec2-config:
  block_device_mappings:
    - DeviceName: /dev/sda
      Ebs.VolumeSize: 100
      Ebs.VolumeType: gp2
      tag:
        tag0: myserver
        tag1: value
    - DeviceName: /dev/sdb
      Ebs.VolumeType: gp2
      Ebs.VolumeSize: 3001
      tag:
        tagX: value
        tagY: value
```

You can configure any AWS valid tag name as shown in the above example, including 'Name'. If you do not configure the tag 'Name', it will be automatically created with a value set to the virtual machine name. If you configure the tag 'Name', the value you configure will be used rather than defaulting to the virtual machine name as shown in the following configuration.

```
my-ec2-config:
  block_device_mappings:
    - DeviceName: /dev/sda
      Ebs.VolumeSize: 100
      Ebs.VolumeType: gp2
      tag:
        Name: myserver
        tag0: value
        tag1: value
    - DeviceName: /dev/sdb
      Ebs.VolumeType: gp2
      Ebs.VolumeSize: 3001
      tag:
        Name: customvalue
        tagX: value
        tagY: value
```

Existing EBS volumes may also be attached (not created) to your instances or you can create new EBS volumes based on EBS snapshots. To simply attach an existing volume use the `volume_id` parameter.

```
device: /dev/xvdj
volume_id: vol-12345abcd
```

Or, to create a volume from an EBS snapshot, use the `snapshot` parameter.

```
device: /dev/xvdj
snapshot: snap-abcd12345
```

Note that `volume_id` will take precedence over the `snapshot` parameter.

Tags can be set once an instance has been launched.

```
my-ec2-config:
    tag:
        tag0: value
        tag1: value
```

Setting up a Master inside EC2

Salt Cloud can configure Salt Masters as well as Minions. Use the `make_master` setting to use this functionality.

```
my-ec2-config:
  # Optionally install a Salt Master in addition to the Salt Minion
  make_master: True
```

When creating a Salt Master inside EC2 with `make_master: True`, or when the Salt Master is already located and configured inside EC2, by default, minions connect to the master's public IP address during Salt Cloud's provisioning process. Depending on how your security groups are defined, the minions may or may not be able to communicate with the master. In order to use the master's private IP in EC2 instead of the public IP, set the `salt_interface` to `private_ips`.

```
my-ec2-config:
  # Optionally set the IP configuration to private_ips
  salt_interface: private_ips
```

Modify EC2 Tags

One of the features of EC2 is the ability to tag resources. In fact, under the hood, the names given to EC2 instances by salt-cloud are actually just stored as a tag called Name. Salt Cloud has the ability to manage these tags:

```
salt-cloud -a get_tags mymachine
salt-cloud -a set_tags mymachine tag1=somestuff tag2='Other stuff'
salt-cloud -a del_tags mymachine tag1,tag2,tag3
```

It is possible to manage tags on any resource in EC2 with a Resource ID, not just instances:

```
salt-cloud -f get_tags my_ec2 resource_id=af5467ba
salt-cloud -f set_tags my_ec2 resource_id=af5467ba tag1=somestuff
salt-cloud -f del_tags my_ec2 resource_id=af5467ba tags=tag1,tag2,tag3
```

Rename EC2 Instances

As mentioned above, EC2 instances are named via a tag. However, renaming an instance by renaming its tag will cause the salt keys to mismatch. A rename function exists which renames both the instance, and the salt keys.

```
salt-cloud -a rename mymachine newname=yourmachine
```

Rename on Destroy

When instances on EC2 are destroyed, there will be a lag between the time that the action is sent, and the time that Amazon cleans up the instance. During this time, the instance still retains a Name tag, which will cause a collision if the creation of an instance with the same name is attempted before the cleanup occurs. In order to avoid such collisions, Salt Cloud can be configured to rename instances when they are destroyed. The new name will look something like:

```
myinstance-DEL20f5b8ad4eb64ed88f2c428df80a1a0c
```

In order to enable this, add rename_on_destroy line to the main configuration file:

```
my-ec2-config:
  rename_on_destroy: True
```

Listing Images

Normally, images can be queried on a cloud provider by passing the `--list-images` argument to Salt Cloud. This still holds true for EC2:

```
salt-cloud --list-images my-ec2-config
```

However, the full list of images on EC2 is extremely large, and querying all of the available images may cause Salt Cloud to behave as if frozen. Therefore, the default behavior of this option may be modified, by adding an `owner` argument to the provider configuration:

```
owner: aws-marketplace
```

The possible values for this setting are `amazon`, `aws-marketplace`, `self`, `<AWS account ID>` or `all`. The default setting is `amazon`. Take note that `all` and `aws-marketplace` may cause Salt Cloud to appear as if it is freezing, as it tries to handle the large amount of data.

It is also possible to perform this query using different settings without modifying the configuration files. To do this, call the `avail_images` function directly:

```
salt-cloud -f avail_images my-ec2-config owner=aws-marketplace
```

EC2 Images

The following are lists of available AMI images, generally sorted by OS. These lists are on 3rd-party websites, are not managed by Salt Stack in any way. They are provided here as a reference for those who are interested, and contain no warranty (express or implied) from anyone affiliated with Salt Stack. Most of them have never been used, much less tested, by the Salt Stack team.

- Arch Linux
- FreeBSD
- Fedora
- CentOS
- Ubuntu
- Debian
- OmniOS

- All Images on Amazon

NOTE: If `image` of a profile does not start with `ami-`, latest image with that name will be used. For example, to create a CentOS 7 profile, instead of using the AMI like `image: ami-1caef165`, we can use its name like `image: 'CentOS Linux 7 x86_64 HVM EBS ENA 1803_01'`. We can also use a pattern like below to get the latest CentOS 7:

```
profile-id:
  provider: provider-name
  subnetid: subnet-XXXXXXXX
  image: 'CentOS Linux 7 x86_64 HVM EBS *'
  size: m1.medium
  ssh_username: centos
  securitygroupid:
    - sg-XXXXXXXX
  securitygroupname:
    - AnotherSecurityGroup
    - AndThirdSecurityGroup
```

show_image

This is a function that describes an AMI on EC2. This will give insight as to the defaults that will be applied to an instance using a particular AMI.

```
$ salt-cloud -f show_image ec2 image=ami-fd20ad94
```

show_instance

This action is a thin wrapper around `--full-query`, which displays details on a single instance only. In an environment with several machines, this will save a user from having to sort through all instance data, just to examine a single instance.

```
$ salt-cloud -a show_instance myinstance
```

ebs_optimized

This argument enables switching of the EbsOptimized setting which default to 'false'. Indicates whether the instance is optimized for EBS I/O. This optimization provides dedicated throughput to Amazon EBS and an optimized configuration stack to provide optimal Amazon EBS I/O performance. This optimization isn't available with all instance types. Additional usage charges apply when using an EBS-optimized instance.

This setting can be added to the profile or map file for an instance.

If set to True, this setting will enable an instance to be EbsOptimized

```
ebs_optimized: True
```

This can also be set as a cloud provider setting in the EC2 cloud configuration:

```
my-ec2-config:
  ebs_optimized: True
```

del_root_vol_on_destroy

This argument overrides the default DeleteOnTermination setting in the AMI for the EBS root volumes for an instance. Many AMIs contain 'false' as a default, resulting in orphaned volumes in the EC2 account, which may unknowingly be charged to the account. This setting can be added to the profile or map file for an instance.

If set, this setting will apply to the root EBS volume

```
del_root_vol_on_destroy: True
```

This can also be set as a cloud provider setting in the EC2 cloud configuration:

```
my-ec2-config:
  del_root_vol_on_destroy: True
```

del_all_vols_on_destroy

This argument overrides the default DeleteOnTermination setting in the AMI for the not-root EBS volumes for an instance. Many AMIs contain 'false' as a default, resulting in orphaned volumes in the EC2 account, which may unknowingly be charged to the account. This setting can be added to the profile or map file for an instance.

If set, this setting will apply to any (non-root) volumes that were created by salt-cloud using the 'volumes' setting.

The volumes will not be deleted under the following conditions * If a volume is detached before terminating the instance * If a volume is created without this setting and attached to the instance

```
del_all_vols_on_destroy: True
```

This can also be set as a cloud provider setting in the EC2 cloud configuration:

```
my-ec2-config:
  del_all_vols_on_destroy: True
```

The setting for this may be changed on all volumes of an existing instance using one of the following commands:

```
salt-cloud -a delvol_on_destroy myinstance
salt-cloud -a keepvol_on_destroy myinstance
salt-cloud -a show_delvol_on_destroy myinstance
```

The setting for this may be changed on a volume on an existing instance using one of the following commands:

```
salt-cloud -a delvol_on_destroy myinstance device=/dev/sda1
salt-cloud -a delvol_on_destroy myinstance volume_id=vol-1a2b3c4d
salt-cloud -a keepvol_on_destroy myinstance device=/dev/sda1
salt-cloud -a keepvol_on_destroy myinstance volume_id=vol-1a2b3c4d
salt-cloud -a show_delvol_on_destroy myinstance device=/dev/sda1
salt-cloud -a show_delvol_on_destroy myinstance volume_id=vol-1a2b3c4d
```

EC2 Termination Protection

EC2 allows the user to enable and disable termination protection on a specific instance. An instance with this protection enabled cannot be destroyed. The EC2 driver adds a show_term_protect action to the regular EC2 functionality.

```
salt-cloud -a show_term_protect mymachine
salt-cloud -a enable_term_protect mymachine
salt-cloud -a disable_term_protect mymachine
```

Alternate Endpoint

Normally, EC2 endpoints are build using the region and the service_url. The resulting endpoint would follow this pattern:

```
ec2.<region>.<service_url>
```

This results in an endpoint that looks like:

```
ec2.us-east-1.amazonaws.com
```

There are other projects that support an EC2 compatibility layer, which this scheme does not account for. This can be overridden by specifying the endpoint directly in the main cloud configuration file:

```
my-ec2-config:
  endpoint: myendpoint.example.com:1138/services/Cloud
```

Volume Management

The EC2 driver has several functions and actions for management of EBS volumes.

Creating Volumes

A volume may be created, independent of an instance. A zone must be specified. A size or a snapshot may be specified (in GiB). If neither is given, a default size of 10 GiB will be used. If a snapshot is given, the size of the snapshot will be used.

The following parameters may also be set (when providing a snapshot OR size):

- `type`: choose between standard (magnetic disk), gp2 (SSD), or io1 (provisioned IOPS). (default=standard)
- `iops`: the number of IOPS (only applicable to io1 volumes) (default varies on volume size)
- `encrypted`: enable encryption on the volume (default=false)

```
salt-cloud -f create_volume ec2 zone=us-east-1b
salt-cloud -f create_volume ec2 zone=us-east-1b size=10
salt-cloud -f create_volume ec2 zone=us-east-1b snapshot=snap12345678
salt-cloud -f create_volume ec2 size=10 type=standard
salt-cloud -f create_volume ec2 size=10 type=gp2
salt-cloud -f create_volume ec2 size=10 type=io1 iops=1000
```

Attaching Volumes

Unattached volumes may be attached to an instance. The following values are required; name or instance_id, volume_id, and device.

```
salt-cloud -a attach_volume myinstance volume_id=vol-12345 device=/dev/sdb1
```

Show a Volume

The details about an existing volume may be retrieved.

```
salt-cloud -a show_volume myinstance volume_id=vol-12345
salt-cloud -f show_volume ec2 volume_id=vol-12345
```

Detaching Volumes

An existing volume may be detached from an instance.

```
salt-cloud -a detach_volume myinstance volume_id=vol-12345
```

Deleting Volumes

A volume that is not attached to an instance may be deleted.

```
salt-cloud -f delete_volume ec2 volume_id=vol-12345
```

Managing Key Pairs

The EC2 driver has the ability to manage key pairs.

Creating a Key Pair

A key pair is required in order to create an instance. When creating a key pair with this function, the return data will contain a copy of the private key. This private key is not stored by Amazon, will not be obtainable past this point, and should be stored immediately.

```
salt-cloud -f create_keypair ec2 keyname=mykeypair
```

Importing a Key Pair

```
salt-cloud -f import_keypair ec2 keyname=mykeypair file=/path/to/id_rsa.pub
```

Show a Key Pair

This function will show the details related to a key pair, not including the private key itself (which is not stored by Amazon).

```
salt-cloud -f show_keypair ec2 keyname=mykeypair
```

Delete a Key Pair

This function removes the key pair from Amazon.

```
salt-cloud -f delete_keypair ec2 keyname=mykeypair
```

Launching instances into a VPC

Simple launching into a VPC

In the amazon web interface, identify the id or the name of the subnet into which your image should be created. Then, edit your cloud.profiles file like so:-

```yaml
profile-id:
  provider: provider-name
  subnetid: subnet-XXXXXXXX
  image: ami-XXXXXXXX
  size: m1.medium
  ssh_username: ubuntu
  securitygroupid:
    - sg-XXXXXXXX
  securitygroupname:
    - AnotherSecurityGroup
    - AndThirdSecurityGroup
```

Note that 'subnetid' takes precedence over 'subnetname', but 'securitygroupid' and 'securitygroupname' are merged together to generate a single list for SecurityGroups of instances.

Specifying interface properties

New in version 2014.7.0.

Launching into a VPC allows you to specify more complex configurations for the network interfaces of your virtual machines, for example:-

```yaml
profile-id:
  provider: provider-name
  image: ami-XXXXXXXX
  size: m1.medium
  ssh_username: ubuntu

  # Do not include either 'subnetid', 'subnetname', 'securitygroupid' or
  # 'securitygroupname' here if you are going to manually specify
  # interface configuration
  #
  network_interfaces:
    - DeviceIndex: 0
      SubnetId: subnet-XXXXXXXX
      SecurityGroupId:
        - sg-XXXXXXXX

      # Uncomment this line if you would like to set an explicit private
      # IP address for the ec2 instance
      #
      # PrivateIpAddress: 192.168.1.66

      # Uncomment this to associate an existing Elastic IP Address with
      # this network interface:
      #
      # associate_eip: eipalloc-XXXXXXXX

      # You can allocate more than one IP address to an interface. Use the
```

(continues on next page)

(continued from previous page)

```
      # 'ip addr list' command to see them.
      #
      # SecondaryPrivateIpAddressCount: 2

      # Uncomment this to allocate a new Elastic IP Address to this
      # interface (will be associated with the primary private ip address
      # of the interface
      #
      # allocate_new_eip: True

      # Uncomment this instead to allocate a new Elastic IP Address to
      # both the primary private ip address and each of the secondary ones
      #
      allocate_new_eips: True

      # Uncomment this if you're creating NAT instances. Allows an instance
      # to accept IP packets with destinations other than itself.
      # SourceDestCheck: False

    - DeviceIndex: 1
      subnetname: XXXXXXXX-Subnet
      securitygroupname:
        - XXXXXXXX-SecurityGroup
        - YYYYYYYY-SecurityGroup
```

Note that it is an error to assign a 'subnetid', 'subnetname', 'securitygroupid' or 'securitygroupname' to a profile where the interfaces are manually configured like this. These are both really properties of each network interface, not of the machine itself.

14.7.8 Getting Started With GoGrid

GoGrid is a public cloud host that supports Linux and Windows.

Configuration

To use Salt Cloud with GoGrid log into the GoGrid web interface and create an API key. Do this by clicking on "My Account" and then going to the API Keys tab.

The `apikey` and the `sharedsecret` configuration parameters need to be set in the configuration file to enable interfacing with GoGrid:

```
# Note: This example is for /etc/salt/cloud.providers or any file in the
# /etc/salt/cloud.providers.d/ directory.

my-gogrid-config:
  driver: gogrid
  apikey: asdff7896asdh789
  sharedsecret: saltybacon
```

Note: A Note about using Map files with GoGrid:

Due to limitations in the GoGrid API, instances cannot be provisioned in parallel with the GoGrid driver. Map files will work with GoGrid, but the `-P` argument should not be used on maps referencing GoGrid instances.

Note: Changed in version 2015.8.0.

The `provider` parameter in cloud provider definitions was renamed to `driver`. This change was made to avoid confusion with the `provider` parameter that is used in cloud profile definitions. Cloud provider definitions now use `driver` to refer to the Salt cloud module that provides the underlying functionality to connect to a cloud host, while cloud profiles continue to use `provider` to refer to provider configurations that you define.

Profiles

Cloud Profiles

Set up an initial profile at `/etc/salt/cloud.profiles` or in the `/etc/salt/cloud.profiles.d/` directory:

```
gogrid_512:
  provider: my-gogrid-config
  size: 512MB
  image: CentOS 6.2 (64-bit) w/ None
```

Sizes can be obtained using the `--list-sizes` option for the `salt-cloud` command:

```
# salt-cloud --list-sizes my-gogrid-config
my-gogrid-config:
    ----------
    gogrid:
        ----------
        512MB:
            ----------
            bandwidth:
                None
            disk:
                30
            driver:
            get_uuid:
            id:
                512MB
            name:
                512MB
            price:
                0.095
            ram:
                512
            uuid:
                bde1e4d7c3a643536e42a35142c7caac34b060e9
...SNIP...
```

Images can be obtained using the `--list-images` option for the `salt-cloud` command:

```
# salt-cloud --list-images my-gogrid-config
my-gogrid-config:
    ----------
    gogrid:
        ----------
        CentOS 6.4 (64-bit) w/ None:
```

(continues on next page)

(continued from previous page)

```
            ----------
            driver:
            extra:
                ----------
            get_uuid:
            id:
                18094
            name:
                CentOS 6.4 (64-bit) w/ None
            uuid:
                bfd4055389919e01aa6261828a96cf54c8dcc2c4
...SNIP...
```

Assigning IPs

New in version 2015.8.0.

The GoGrid API allows IP addresses to be manually assigned. Salt Cloud supports this functionality by allowing an IP address to be specified using the `assign_public_ip` argument. This likely makes the most sense inside a map file, but it may also be used inside a profile.

```
gogrid_512:
  provider: my-gogrid-config
  size: 512MB
  image: CentOS 6.2 (64-bit) w/ None
  assign_public_ip: 11.38.257.42
```

14.7.9 Getting Started With Google Compute Engine

Google Compute Engine (GCE) is Google-infrastructure as a service that lets you run your large-scale computing workloads on virtual machines. This document covers how to use Salt Cloud to provision and manage your virtual machines hosted within Google's infrastructure.

You can find out more about GCE and other Google Cloud Platform services at https://cloud.google.com.

Dependencies

- LibCloud >= 1.0.0

Changed in version 2017.7.0.

- A Google Cloud Platform account with Compute Engine enabled
- A registered Service Account for authorization
- Oh, and obviously you'll need salt

Google Compute Engine Setup

1. Sign up for Google Cloud Platform

   Go to https://cloud.google.com and use your Google account to sign up for Google Cloud Platform and complete the guided instructions.

2. Create a Project

   Next, go to the console at https://cloud.google.com/console and create a new Project. Make sure to select your new Project if you are not automatically directed to the Project.

   Projects are a way of grouping together related users, services, and billing. You may opt to create multiple Projects and the remaining instructions will need to be completed for each Project if you wish to use GCE and Salt Cloud to manage your virtual machines.

3. Enable the Google Compute Engine service

   In your Project, either just click *Compute Engine* to the left, or go to the *APIs & auth* section and *APIs* link and enable the Google Compute Engine service.

4. Create a Service Account

   To set up authorization, navigate to *APIs & auth* section and then the *Credentials* link and click the *CREATE NEW CLIENT ID* button. Select *Service Account* and click the *Create Client ID* button. This will automatically download a `.json` file, which may or may not be used in later steps, depending on your version of `libcloud`.

   Look for a new *Service Account* section in the page and record the generated email address for the matching key/fingerprint. The email address will be used in the `service_account_email_address` of the `/etc/salt/cloud.providers` or the `/etc/salt/cloud.providers.d/*.conf` file.

5. Key Format

   Note: If you are using `libcloud >= 0.17.0` it is recommended that you use the `JSON format` file you downloaded above and skip to the *Provider Configuration* section below, using the JSON file **in place of 'NEW.pem'** in the documentation.

   If you are using an older version of libcloud or are unsure of the version you have, please follow the instructions below to generate and format a new P12 key.

   In the new *Service Account* section, click *Generate new P12 key*, which will automatically download a `.p12` private key file. The `.p12` private key needs to be converted to a format compatible with libcloud. This new Google-generated private key was encrypted using *notasecret* as a passphrase. Use the following command and record the location of the converted private key and record the location for use in the `service_account_private_key` of the `/etc/salt/cloud` file:

   ```
   openssl pkcs12 -in ORIG.p12 -passin pass:notasecret \
   -nodes -nocerts | openssl rsa -out NEW.pem
   ```

Provider Configuration

Set up the provider cloud config at `/etc/salt/cloud.providers` or `/etc/salt/cloud.providers.d/*.conf`:

```
gce-config:
  # Set up the Project name and Service Account authorization
  project: "your-project-id"
  service_account_email_address: "123-a5gt@developer.gserviceaccount.com"
  service_account_private_key: "/path/to/your/NEW.pem"

  # Set up the location of the salt master
  minion:
    master: saltmaster.example.com
```

(continues on next page)

(continued from previous page)

```
  # Set up grains information, which will be common for all nodes
  # using this provider
  grains:
    node_type: broker
    release: 1.0.1

  driver: gce
```

Note: Empty strings as values for `service_account_private_key` and `service_account_email_address` can be used on GCE instances. This will result in the service account assigned to the GCE instance being used.

Note: The value provided for `project` must not contain underscores or spaces and is labeled as ”Project ID” on the Google Developers Console.

Note: Changed in version 2015.8.0.

The `provider` parameter in cloud provider definitions was renamed to `driver`. This change was made to avoid confusion with the `provider` parameter that is used in cloud profile definitions. Cloud provider definitions now use `driver` to refer to the Salt cloud module that provides the underlying functionality to connect to a cloud host, while cloud profiles continue to use `provider` to refer to provider configurations that you define.

Profile Configuration

Set up an initial profile at `/etc/salt/cloud.profiles` or `/etc/salt/cloud.profiles.d/*.conf`:

```
my-gce-profile:
  image: centos-6
  size: n1-standard-1
  location: europe-west1-b
  network: default
  subnetwork: default
  tags: '["one", "two", "three"]'
  metadata: '{"one": "1", "2": "two"}'
  use_persistent_disk: True
  delete_boot_pd: False
  deploy: True
  make_master: False
  provider: gce-config
```

The profile can be realized now with a salt command:

```
salt-cloud -p my-gce-profile gce-instance
```

This will create an salt minion instance named `gce-instance` in GCE. If the command was executed on the salt-master, its Salt key will automatically be signed on the master.

Once the instance has been created with a salt-minion installed, connectivity to it can be verified with Salt:

```
salt gce-instance test.version
```

GCE Specific Settings

Consult the sample profile below for more information about GCE specific settings. Some of them are mandatory and are properly labeled below but typically also include a hard-coded default.

Initial Profile

Set up an initial profile at `/etc/salt/cloud.profiles` or `/etc/salt/cloud.profiles.d/gce.conf`:

```
my-gce-profile:
  image: centos-6
  size: n1-standard-1
  location: europe-west1-b
  network: default
  subnetwork: default
  tags: '["one", "two", "three"]'
  metadata: '{"one": "1", "2": "two"}'
  use_persistent_disk: True
  delete_boot_pd: False
  ssh_interface: public_ips
  external_ip: "ephemeral"
```

image

Image is used to define what Operating System image should be used to for the instance. Examples are Debian 7 (wheezy) and CentOS 6. Required.

size

A 'size', in GCE terms, refers to the instance's 'machine type'. See the on-line documentation for a complete list of GCE machine types. Required.

location

A 'location', in GCE terms, refers to the instance's 'zone'. GCE has the notion of both Regions (e.g. us-central1, europe-west1, etc) and Zones (e.g. us-central1-a, us-central1-b, etc). Required.

network

Use this setting to define the network resource for the instance. All GCE projects contain a network named 'default' but it's possible to use this setting to create instances belonging to a different network resource.

subnetwork

Use this setting to define the subnetwork an instance will be created in. This requires that the network your instance is created under has a mode of 'custom' or 'auto'. Additionally, the subnetwork your instance is created under is associated with the location you provide.

New in version 2017.7.0.

tags

GCE supports instance/network tags and this setting allows you to set custom tags. It should be a list of strings and must be parse-able by the python ast.literal_eval() function to convert it to a python list.

metadata

GCE supports instance metadata and this setting allows you to set custom metadata. It should be a hash of key/value strings and parse-able by the python ast.literal_eval() function to convert it to a python dictionary.

use_persistent_disk

Use this setting to ensure that when new instances are created, they will use a persistent disk to preserve data between instance terminations and re-creations.

delete_boot_pd

In the event that you wish the boot persistent disk to be permanently deleted when you destroy an instance, set delete_boot_pd to True.

ssh_interface

New in version 2015.5.0.

Specify whether to use public or private IP for deploy script.

Valid options are:

- private_ips: The salt-master is also hosted with GCE
- public_ips: The salt-master is hosted outside of GCE

external_ip

Per instance setting: Used a named fixed IP address to this host.

Valid options are:

- ephemeral: The host will use a GCE ephemeral IP
- None: No external IP will be configured on this host.

Optionally, pass the name of a GCE address to use a fixed IP address. If the address does not already exist, it will be created.

ex_disk_type

GCE supports two different disk types, `pd-standard` and `pd-ssd`. The default disk type setting is `pd-standard`. To specify using an SSD disk, set `pd-ssd` as the value.

New in version 2014.7.0.

ip_forwarding

GCE instances can be enabled to use IP Forwarding. When set to `True`, this options allows the instance to send/receive non-matching src/dst packets. Default is `False`.

New in version 2015.8.1.

Profile with scopes

Scopes can be specified by setting the optional `ex_service_accounts` key in your cloud profile. The following example enables the bigquery scope.

```
my-gce-profile:
 image: centos-6
  ssh_username: salt
  size: f1-micro
  location: us-central1-a
  network: default
  subnetwork: default
  tags: '["one", "two", "three"]'
  metadata: '{"one": "1", "2": "two",
              "sshKeys": ""}'
  use_persistent_disk: True
  delete_boot_pd: False
  deploy: False
  make_master: False
  provider: gce-config
  ex_service_accounts:
    - scopes:
      - bigquery
```

Email can also be specified as an (optional) parameter.

```
my-gce-profile:
...snip
  ex_service_accounts:
    - scopes:
      - bigquery
      email: default
```

There can be multiple entries for scopes since `ex-service_accounts` accepts a list of dictionaries. For more information refer to the libcloud documentation on specifying service account scopes.

SSH Remote Access

GCE instances do not allow remote access to the root user by default. Instead, another user must be used to run the deploy script using sudo. Append something like this to `/etc/salt/cloud.profiles` or `/etc/salt/cloud.profiles.d/*.conf`:

```
my-gce-profile:
    ...

    # SSH to GCE instances as gceuser
    ssh_username: gceuser

    # Use the local private SSH key file located here
    ssh_keyfile: /etc/cloud/google_compute_engine
```

If you have not already used this SSH key to login to instances in this GCE project you will also need to add the public key to your projects metadata at https://cloud.google.com/console. You could also add it via the metadata setting too:

```
my-gce-profile:
    ...

    metadata: '{"one": "1", "2": "two",
                "sshKeys": "gceuser:ssh-rsa <Your SSH Public Key> gceuser@host"}'
```

Single instance details

This action is a thin wrapper around `--full-query`, which displays details on a single instance only. In an environment with several machines, this will save a user from having to sort through all instance data, just to examine a single instance.

```
salt-cloud -a show_instance myinstance
```

Destroy, persistent disks, and metadata

As noted in the provider configuration, it's possible to force the boot persistent disk to be deleted when you destroy the instance. The way that this has been implemented is to use the instance metadata to record the cloud profile used when creating the instance. When `destroy` is called, if the instance contains a `salt-cloud-profile` key, it's value is used to reference the matching profile to determine if `delete_boot_pd` is set to `True`.

Be aware that any GCE instances created with salt cloud will contain this custom `salt-cloud-profile` metadata entry.

List various resources

It's also possible to list several GCE resources similar to what can be done with other providers. The following commands can be used to list GCE zones (locations), machine types (sizes), and images.

```
salt-cloud --list-locations gce
salt-cloud --list-sizes gce
salt-cloud --list-images gce
```

Persistent Disk

The Compute Engine provider provides functions via salt-cloud to manage your Persistent Disks. You can create and destroy disks as well as attach and detach them from running instances.

Create

When creating a disk, you can create an empty disk and specify its size (in GB), or specify either an 'image' or 'snapshot'.

```
salt-cloud -f create_disk gce disk_name=pd location=us-central1-b size=200
```

Delete

Deleting a disk only requires the name of the disk to delete

```
salt-cloud -f delete_disk gce disk_name=old-backup
```

Attach

Attaching a disk to an existing instance is really an 'action' and requires both an instance name and disk name. It's possible to use this ation to create bootable persistent disks if necessary. Compute Engine also supports attaching a persistent disk in READ_ONLY mode to multiple instances at the same time (but then cannot be attached in READ_WRITE to any instance).

```
salt-cloud -a attach_disk myinstance disk_name=pd mode=READ_WRITE boot=yes
```

Detach

Detaching a disk is also an action against an instance and only requires the name of the disk. Note that this does *not* safely sync and umount the disk from the instance. To ensure no data loss, you must first make sure the disk is unmounted from the instance.

```
salt-cloud -a detach_disk myinstance disk_name=pd
```

Show disk

It's also possible to look up the details for an existing disk with either a function or an action.

```
salt-cloud -a show_disk myinstance disk_name=pd
salt-cloud -f show_disk gce disk_name=pd
```

Create snapshot

You can take a snapshot of an existing disk's content. The snapshot can then in turn be used to create other persistent disks. Note that to prevent data corruption, it is strongly suggested that you unmount the disk prior to taking a snapshot. You must name the snapshot and provide the name of the disk.

```
salt-cloud -f create_snapshot gce name=backup-20140226 disk_name=pd
```

Delete snapshot

You can delete a snapshot when it's no longer needed by specifying the name of the snapshot.

```
salt-cloud -f delete_snapshot gce name=backup-20140226
```

Show snapshot

Use this function to look up information about the snapshot.

```
salt-cloud -f show_snapshot gce name=backup-20140226
```

Networking

Compute Engine supports multiple private networks per project. Instances within a private network can easily communicate with each other by an internal DNS service that resolves instance names. Instances within a private network can also communicate with either directly without needing special routing or firewall rules even if they span different regions/zones.

Networks also support custom firewall rules. By default, traffic between instances on the same private network is open to all ports and protocols. Inbound SSH traffic (port 22) is also allowed but all other inbound traffic is blocked.

Create network

New networks require a name and CIDR range if they don't have a 'mode'. Optionally, 'mode' can be provided. Supported modes are 'auto', 'custom', 'legacy'. Optionally, 'description' can be provided to add an extra note to your network. New instances can be created and added to this network by setting the network name during create. It is not possible to add/remove existing instances to a network.

```
salt-cloud -f create_network gce name=mynet cidr=10.10.10.0/24
salt-cloud -f create_network gce name=mynet mode=auto description=some optional info.
```

Changed in version 2017.7.0.

Destroy network

Destroy a network by specifying the name. If a resource is currently using the target network an exception will be raised.

```
salt-cloud -f delete_network gce name=mynet
```

Show network

Specify the network name to view information about the network.

```
salt-cloud -f show_network gce name=mynet
```

Create subnetwork

New subnetworks require a name, region, and CIDR range. Optionally, 'description' can be provided to add an extra note to your subnetwork. New instances can be created and added to this subnetwork by setting the subnetwork name during create. It is not possible to add/remove existing instances to a subnetwork.

```
salt-cloud -f create_subnetwork gce name=mynet network=mynet region=us-central1
↪cidr=10.0.10.0/24
salt-cloud -f create_subnetwork gce name=mynet network=mynet region=us-central1
↪cidr=10.10.10.0/24 description=some info about my subnet.
```

New in version 2017.7.0.

Destroy subnetwork

Destroy a subnetwork by specifying the name and region. If a resource is currently using the target subnetwork an exception will be raised.

```
salt-cloud -f delete_subnetwork gce name=mynet region=us-central1
```

New in version 2017.7.0.

Show subnetwork

Specify the subnetwork name to view information about the subnetwork.

```
salt-cloud -f show_subnetwork gce name=mynet
```

New in version 2017.7.0.

Create address

Create a new named static IP address in a region.

```
salt-cloud -f create_address gce name=my-fixed-ip region=us-central1
```

Delete address

Delete an existing named fixed IP address.

```
salt-cloud -f delete_address gce name=my-fixed-ip region=us-central1
```

Show address

View details on a named address.

```
salt-cloud -f show_address gce name=my-fixed-ip region=us-central1
```

Create firewall

You'll need to create custom firewall rules if you want to allow other traffic than what is described above. For instance, if you run a web service on your instances, you'll need to explicitly allow HTTP and/or SSL traffic. The firewall rule must have a name and it will use the 'default' network unless otherwise specified with a 'network' attribute. Firewalls also support instance tags for source/destination

```
salt-cloud -f create_fwrule gce name=web allow=tcp:80,tcp:443,icmp
```

Delete firewall

Deleting a firewall rule will prevent any previously allowed traffic for the named firewall rule.

```
salt-cloud -f delete_fwrule gce name=web
```

Show firewall

Use this function to review an existing firewall rule's information.

```
salt-cloud -f show_fwrule gce name=web
```

Load Balancer

Compute Engine possess a load-balancer feature for splitting traffic across multiple instances. Please reference the documentation for a more complete description.

The load-balancer functionality is slightly different than that described in Google's documentation. The concept of *TargetPool* and *ForwardingRule* are consolidated in salt-cloud/libcloud. HTTP Health Checks are optional.

HTTP Health Check

HTTP Health Checks can be used as a means to toggle load-balancing across instance members, or to detect if an HTTP site is functioning. A common use-case is to set up a health check URL and if you want to toggle traffic on/off to an instance, you can temporarily have it return a non-200 response. A non-200 response to the load-balancer's health check will keep the LB from sending any new traffic to the "down" instance. Once the instance's health check URL beings returning 200-responses, the LB will again start to send traffic to it. Review Compute Engine's documentation for allowable parameters. You can use the following salt-cloud functions to manage your HTTP health checks.

```
salt-cloud -f create_hc gce name=myhc path=/ port=80
salt-cloud -f delete_hc gce name=myhc
salt-cloud -f show_hc gce name=myhc
```

Load-balancer

When creating a new load-balancer, it requires a name, region, port range, and list of members. There are other optional parameters for protocol, and list of health checks. Deleting or showing details about the LB only requires the name.

```
salt-cloud -f create_lb gce name=lb region=... ports=80 members=w1,w2,w3
salt-cloud -f delete_lb gce name=lb
salt-cloud -f show_lb gce name=lb
```

You can also create a load balancer using a named fixed IP addressby specifying the name of the address. If the address does not exist yet it will be created.

```
salt-cloud -f create_lb gce name=my-lb region=us-central1 ports=234 members=s1,s2,s3␣
↪address=my-lb-ip
```

Attach and Detach LB

It is possible to attach or detach an instance from an existing load-balancer. Both the instance and load-balancer must exist before using these functions.

```
salt-cloud -f attach_lb gce name=lb member=w4
salt-cloud -f detach_lb gce name=lb member=oops
```

14.7.10 Getting Started With HP Cloud

HP Cloud is a major public cloud platform and uses the libcloud *openstack* driver. The current version of OpenStack that HP Cloud uses is Havana. When an instance is booted, it must have a floating IP added to it in order to connect to it and further below you will see an example that adds context to this statement.

Set up a cloud provider configuration file

To use the *openstack* driver for HP Cloud, set up the cloud provider configuration file as in the example shown below:

`/etc/salt/cloud.providers.d/hpcloud.conf`:

```
hpcloud-config:
  # Set the location of the salt-master
  #
  minion:
    master: saltmaster.example.com

  # Configure HP Cloud using the OpenStack plugin
  #
  identity_url: https://region-b.geo-1.identity.hpcloudsvc.com:35357/v2.0/tokens
  compute_name: Compute
  protocol: ipv4

  # Set the compute region:
  #
  compute_region: region-b.geo-1

  # Configure HP Cloud authentication credentials
  #
  user: myname
  tenant: myname-project1
  password: xxxxxxxxx

  # keys to allow connection to the instance launched
```

(continues on next page)

(continued from previous page)

```
  #
  ssh_key_name: yourkey
  ssh_key_file: /path/to/key/yourkey.priv

  driver: openstack
```

The subsequent example that follows is using the openstack driver.

Note: Changed in version 2015.8.0.

The `provider` parameter in cloud provider definitions was renamed to `driver`. This change was made to avoid confusion with the `provider` parameter that is used in cloud profile definitions. Cloud provider definitions now use `driver` to refer to the Salt cloud module that provides the underlying functionality to connect to a cloud host, while cloud profiles continue to use `provider` to refer to provider configurations that you define.

Compute Region

Originally, HP Cloud, in its OpenStack Essex version (1.0), had 3 availability zones in one region, US West (region-a.geo-1), which each behaved each as a region.

This has since changed, and the current OpenStack Havana version of HP Cloud (1.1) now has simplified this and now has two regions to choose from:

```
region-a.geo-1 -> US West
region-b.geo-1 -> US East
```

Authentication

The `user` is the same user as is used to log into the HP Cloud management UI. The `tenant` can be found in the upper left under "Project/Region/Scope". It is often named the same as `user` albeit with a `-project1` appended. The `password` is of course what you created your account with. The management UI also has other information such as being able to select US East or US West.

Set up a cloud profile config file

The profile shown below is a know working profile for an Ubuntu instance. The profile configuration file is stored in the following location:

`/etc/salt/cloud.profiles.d/hp_ae1_ubuntu.conf`:

```
hp_ae1_ubuntu:
    provider: hp_ae1
    image: 9302692b-b787-4b52-a3a6-daebb79cb498
    ignore_cidr: 10.0.0.1/24
    networks:
      - floating: Ext-Net
    size: standard.small
    ssh_key_file: /root/keys/test.key
    ssh_key_name: test
    ssh_username: ubuntu
```

Some important things about the example above:

- The `image` parameter can use either the image name or image ID which you can obtain by running in the example below (this case US East):

```
# salt-cloud --list-images hp_ae1
```

- The parameter `ignore_cidr` specifies a range of addresses to ignore when trying to connect to the instance. In this case, it's the range of IP addresses used for an private IP of the instance.
- The parameter `networks` is very important to include. In previous versions of Salt Cloud, this is what made it possible for salt-cloud to be able to attach a floating IP to the instance in order to connect to the instance and set up the minion. The current version of salt-cloud doesn't require it, though having it is of no harm either. Newer versions of salt-cloud will use this, and without it, will attempt to find a list of floating IP addresses to use regardless.
- The `ssh_key_file` and `ssh_key_name` are the keys that will make it possible to connect to the instance to set up the minion
- The `ssh_username` parameter, in this case, being that the image used will be ubuntu, will make it possible to not only log in but install the minion

Launch an instance

To instantiate a machine based on this profile (example):

```
# salt-cloud -p hp_ae1_ubuntu ubuntu_instance_1
```

After several minutes, this will create an instance named ubuntu_instance_1 running in HP Cloud in the US East region and will set up the minion and then return information about the instance once completed.

Manage the instance

Once the instance has been created with salt-minion installed, connectivity to it can be verified with Salt:

```
# salt ubuntu_instance_1 ping
```

SSH to the instance

Additionally, the instance can be accessed via SSH using the floating IP assigned to it

```
# ssh ubuntu@<floating ip>
```

Using a private IP

Alternatively, in the cloud profile, using the private IP to log into the instance to set up the minion is another option, particularly if salt-cloud is running within the cloud on an instance that is on the same network with all the other instances (minions)

The example below is a modified version of the previous example. Note the use of `ssh_interface`:

```
hp_ae1_ubuntu:
    provider: hp_ae1
    image: 9302692b-b787-4b52-a3a6-daebb79cb498
    size: standard.small
```

(continues on next page)

(continued from previous page)

```
    ssh_key_file: /root/keys/test.key
    ssh_key_name: test
    ssh_username: ubuntu
    ssh_interface: private_ips
```

With this setup, salt-cloud will use the private IP address to ssh into the instance and set up the salt-minion

14.7.11 Getting Started With Joyent

Joyent is a public cloud host that supports SmartOS, Linux, FreeBSD, and Windows.

Dependencies

This driver requires the Python `requests` library to be installed.

Configuration

The Joyent cloud requires three configuration parameters. The user name and password that are used to log into the Joyent system, and the location of the private ssh key associated with the Joyent account. The ssh key is needed to send the provisioning commands up to the freshly created virtual machine.

```
# Note: This example is for /etc/salt/cloud.providers or any file in the
# /etc/salt/cloud.providers.d/ directory.

my-joyent-config:
  driver: joyent
  user: fred
  password: saltybacon
  private_key: /root/mykey.pem
  keyname: mykey
```

Note: Changed in version 2015.8.0.

The `provider` parameter in cloud provider definitions was renamed to `driver`. This change was made to avoid confusion with the `provider` parameter that is used in cloud profile definitions. Cloud provider definitions now use `driver` to refer to the Salt cloud module that provides the underlying functionality to connect to a cloud host, while cloud profiles continue to use `provider` to refer to provider configurations that you define.

Profiles

Cloud Profiles

Set up an initial profile at `/etc/salt/cloud.profiles` or in the `/etc/salt/cloud.profiles.d/` directory:

```
joyent_512:
  provider: my-joyent-config
  size: g4-highcpu-512M
  image: ubuntu-16.04
```

Sizes can be obtained using the `--list-sizes` option for the `salt-cloud` command:

```
# salt-cloud --list-sizes my-joyent-config
my-joyent-config:
    ----------
    joyent:
        ----------
        g4-highcpu-512M:
            ----------
            default:
                False
            description:
                Compute Optimized 512M RAM - 1 vCPU - 10 GB Disk
            disk:
                10240
            group:
                Compute Optimized
            id:
                14aea8fc-d0f8-11e5-bfe4-a7458dbc6c99
            lwps:
                4000
            memory:
                512
            name:
                g4-highcpu-512M
            swap:
                2048
            vcpus:
                0
            version:
                1.0.3
...SNIP...
```

Images can be obtained using the `--list-images` option for the `salt-cloud` command:

```
# salt-cloud --list-images my-joyent-config
my-joyent-config:
    ----------
    joyent:
        ----------
        base:
            ----------
            description:
                A 32-bit SmartOS image with just essential packages
                installed. Ideal for users who are comfortabl e with
                setting up their own environment and tools.
            files:
                |_
                  ----------
                  compression:
                      gzip
                  sha1:
                      b00a77408ddd9aeac85085b68b1cd22a07353956
                  size:
                      106918297
            homepage:
                http://wiki.joyent.com/jpc2/Base+Instance
            id:
```

(continues on next page)

(continued from previous page)

```
                00aec452-6e81-11e4-8474-ebfec9a1a911
            name:
                base
            os:
                smartos
            owner:
                9dce1460-0c4c-4417-ab8b-25ca478c5a78
            public:
                True
            published_at:
                2014-11-17T17:41:46Z
            requirements:
                ----------
            state:
                active
            type:
                smartmachine
            version:
                14.3.0

...SNIP...
```

SmartDataCenter

This driver can also be used with the Joyent SmartDataCenter project. More details can be found at:

Using SDC requires that an api_host_suffix is set. The default value for this is *.api.joyentcloud.com*. All characters, including the leading ., should be included:

```
api_host_suffix: .api.myhostname.com
```

Miscellaneous Configuration

The following configuration items can be set in either `provider` or `profile` confuration files.

use_ssl

When set to `True` (the default), attach `https://` to any URL that does not already have `http://` or `https://` included at the beginning. The best practice is to leave the protocol out of the URL, and use this setting to manage it.

verify_ssl

When set to `True` (the default), the underlying web library will verify the SSL certificate. This should only be set to `False` for debugging.`

14.7.12 Getting Started With Libvirt

Libvirt is a toolkit to interact with the virtualization capabilities of recent versions of Linux (and other OSes). This driver Salt cloud provider is currently geared towards libvirt with qemu-kvm.

http://www.libvirt.org/

Host Dependencies

- libvirt >= 1.2.18 (older might work)

Salt-Cloud Dependencies

- libvirt-python

Provider Configuration

For every KVM host a provider needs to be set up. The provider currently maps to one libvirt daemon (e.g. one KVM host).

Set up the provider cloud configuration file at `/etc/salt/cloud.providers` or `/etc/salt/cloud.providers.d/*.conf`.

```
# Set up a provider with qemu+ssh protocol
kvm-via-ssh:
  driver: libvirt
  url: qemu+ssh://user@kvm.company.com/system?socket=/var/run/libvirt/libvirt-sock

# Or connect to a local libvirt instance
local-kvm:
  driver: libvirt
  url: qemu:///system
  # work around flag for XML validation errors while cloning
  validate_xml: no
```

Cloud Profiles

Virtual machines get cloned from so called Cloud Profiles. Profiles can be set up at `/etc/salt/cloud.profiles` or `/etc/salt/cloud.profiles.d/*.conf`:

- Configure a profile to be used:

```
centos7:
  # points back at provider configuration
  provider: local-kvm
  base_domain: base-centos7-64
  ip_source: ip-learning
  ssh_username: root
  password: my-very-secret-password
  # /tmp is mounted noexec.. do workaround
  deploy_command: sh /tmp/.saltcloud/deploy.sh
  script_args: -F
  # grains to add to the minion
  grains:
    clones-are-awesome: true
  # override minion settings
  minion:
    master: 192.168.16.1
    master_port: 5506
```

The profile can be realized now with a salt command:

```
# salt-cloud -p centos7 my-centos7-clone
```

This will create an instance named `my-centos7-clone` on the cloud host. Also the minion id will be set to `my-centos7-clone`.

If the command was executed on the salt-master, its Salt key will automatically be signed on the master.

Once the instance has been created with salt-minion installed, connectivity to it can be verified with Salt:

```
# salt my-centos7-clone test.version
```

Required Settings

The following settings are always required for libvirt:

```
centos7:
  provider: local-kvm
  # the domain to clone
  base_domain: base-centos7-64
  # how to obtain the IP address of the cloned instance
  # ip-learning or qemu-agent
  ip_source: ip-learning
```

The `ip_source` setting controls how the IP address of the cloned instance is determined. When using `ip-learning` the IP is requested from libvirt. This needs a recent libvirt version and may only work for NAT networks. Another option is to use `qemu-agent` this requires that the qemu-agent is installed and configured to run at startup in the base domain.

Optional Settings

```
# Username and password
ssh_username: root
password: my-secret-password

# Cloning strategy: full or quick
clone_strategy: quick
```

The `clone_strategy` controls how the clone is done. In case of `full` the disks are copied creating a standalone clone. If `quick` is used the disks of the base domain are used as backing disks for the clone. This results in nearly instantaneous clones at the expense of slower write performance. The quick strategy has a number of requirements:

- The disks must be of type qcow2
- The base domain must be turned off
- The base domain must not change after creating the clone

14.7.13 Getting Started With Linode

Linode is a public cloud host with a focus on Linux instances.

Starting with the 2015.8.0 release of Salt, the Linode driver uses Linode's native REST API. There are no external dependencies required to use the Linode driver, other than a Linode account.

Provider Configuration

Linode requires a single API key, but the default root password for new instances also needs to be set. The password needs to be eight characters and contain lowercase, uppercase, and numbers.

Set up the provider cloud configuration file at `/etc/salt/cloud.providers` or `/etc/salt/cloud.providers.d/*.conf`.

```
my-linode-config:
  apikey: 'asldkgfakl;sdfjsjaslfjaklsdjf;askldjfaaklsjdfhasldsadfghdkf'
  password: 'F00barbaz'
  driver: linode
```

Note: Changed in version 2015.8.0.

The `provider` parameter in cloud provider definitions was renamed to `driver`. This change was made to avoid confusion with the `provider` parameter that is used in cloud profile definitions. Cloud provider definitions now use `driver` to refer to the Salt cloud module that provides the underlying functionality to connect to a cloud host, while cloud profiles continue to use `provider` to refer to provider configurations that you define.

Profile Configuration

Linode profiles require a `provider`, `size`, `image`, and `location`. Set up an initial profile at `/etc/salt/cloud.profiles` or `/etc/salt/cloud.profiles.d/*.conf`:

```
linode_1024:
  provider: my-linode-config
  size: Linode 2GB
  image: CentOS 7
  location: London, England, UK
```

The profile can be realized now with a salt command:

```
salt-cloud -p linode_1024 linode-instance
```

This will create an salt minion instance named `linode-instance` in Linode. If the command was executed on the salt-master, its Salt key will automatically be signed on the master.

Once the instance has been created with a salt-minion installed, connectivity to it can be verified with Salt:

```
salt linode-instance test.version
```

Listing Sizes

Sizes can be obtained using the `--list-sizes` option for the `salt-cloud` command:

```
# salt-cloud --list-sizes my-linode-config
my-linode-config:
    ----------
    linode:
        ----------
        Linode 2GB:
            ----------
```

(continues on next page)

(continued from previous page)

```
            AVAIL:
                ----------
                10:
                    500
                11:
                    500
                2:
                    500
                3:
                    500
                4:
                    500
                6:
                    500
                7:
                    500
                8:
                    500
                9:
                    500
            CORES:
                1
            DISK:
                50
            HOURLY:
                0.015
            LABEL:
                Linode 2GB
            PLANID:
                2
            PRICE:
                10.0
            RAM:
                2048
            XFER:
                2000
...SNIP...
```

Listing Images

Images can be obtained using the `--list-images` option for the `salt-cloud` command:

```
# salt-cloud --list-images my-linode-config
my-linode-config:
    ----------
    linode:
        ----------
        Arch Linux 2015.02:
            ----------
            CREATE_DT:
                2015-02-20 14:17:16.0
            DISTRIBUTIONID:
                138
            IS64BIT:
                1
```

(continues on next page)

(continued from previous page)

```
            LABEL:
                Arch Linux 2015.02
            MINIMAGESIZE:
                800
            REQUIRESPVOPSKERNEL:
                1
...SNIP...
```

Listing Locations

Locations can be obtained using the `--list-locations` option for the `salt-cloud` command:

```
# salt-cloud --list-locations my-linode-config
my-linode-config:
    ----------
    linode:
        ----------
        Atlanta, GA, USA:
            ----------
            ABBR:
                atlanta
            DATACENTERID:
                4
            LOCATION:
                Atlanta, GA, USA
...SNIP...
```

Linode Specific Settings

There are several options outlined below that can be added to either the Linode provider of profile configuration files. Some options are mandatory and are properly labeled below but typically also include a hard-coded default.

image

Image is used to define what Operating System image should be used for the instance. Examples are `Ubuntu 14.04 LTS` and `CentOS 7`. This option should be specified in the profile config. Required.

location

Location is used to define which Linode data center the instance will reside in. Required.

size

Size is used to define the instance's "plan type" which includes memory, storage, and price. Required.

assign_private_ip

New in version 2016.3.0.

Assigns a private IP address to a Linode when set to True. Default is False.

ssh_interface

New in version 2016.3.0.

Specify whether to use a public or private IP for the deploy script. Valid options are:

- public_ips: The salt-master is hosted outside of Linode. Default.
- private_ips: The salt-master is also hosted within Linode.

If specifying `private_ips`, the Linodes must be hosted within the same data center and have the Network Helper enabled on your entire account. The instance that is running the Salt-Cloud provisioning command must also have a private IP assigned to it.

Newer accounts created on Linode have the Network Helper setting enabled by default, account-wide. Legacy accounts do not have this setting enabled by default. To enable the Network Helper on your Linode account, please see Linode's Network Helper documentation.

If you're running into problems, be sure to restart the instance that is running Salt Cloud after adding its own private IP address or enabling the Network Helper.

clonefrom

Setting the clonefrom option to a specified instance enables the new instance to be cloned from the named instance instead of being created from scratch. If using the clonefrom option, it is likely a good idea to also specify `script_args: -C` if a minion is already installed on the to-be-cloned instance. See the *Cloning* section below for more information.

Cloning

To clone a Linode, add a profile with a `clonefrom` key, and a `script_args: -C`. `clonefrom` should be the name of the Linode that is the source for the clone. `script_args: -C` passes a -C to the salt-bootstrap script, which only configures the minion and doesn't try to install a new copy of salt-minion. This way the minion gets new keys and the keys get pre-seeded on the master, and the `/etc/salt/minion` file has the right minion 'id:' declaration.

Cloning requires a post 2015-02-01 salt-bootstrap.

It is safest to clone a stopped machine. To stop a machine run

```
salt-cloud -a stop machine_to_clone
```

To create a new machine based on another machine, add an entry to your linode cloud profile that looks like this:

```
li-clone:
  provider: my-linode-config
  clonefrom: machine_to_clone
  script_args: -C -F
```

Then run salt-cloud as normal, specifying `-p li-clone`. The profile name can be anything; It doesn't have to be `li-clone`.

`clonefrom:` is the name of an existing machine in Linode from which to clone. `Script_args: -C -F` is necessary to avoid re-deploying Salt via salt-bootstrap. `-C` will just re-deploy keys so the new minion will not have a duplicate key or minion_id on the Master, and `-F` will force a rewrite of the Minion config file on the new Minion.

If `-F` isn't provided, the new Minion will have the `machine_to_clone`'s Minion ID, instead of its own Minion ID, which can cause problems.

Note: Pull Request #733 to the salt-bootstrap repo makes the `-F` argument non-necessary. Once that change is released into a stable version of the Bootstrap Script, the `-C` argument will be sufficient for the `script_args` setting.

If the `machine_to_clone` does not have Salt installed on it, refrain from using the `script_args: -C -F` altogether, because the new machine will need to have Salt installed.

14.7.14 Getting Started With LXC

The LXC module is designed to install Salt in an LXC container on a controlled and possibly remote minion.

In other words, Salt will connect to a minion, then from that minion:

- Provision and configure a container for networking access
- Use those modules to deploy salt and re-attach to master.
  - *lxc runner*
  - *lxc module*
  - *seed*

Limitations

- You can only act on one minion and one provider at a time.
- Listing images must be targeted to a particular LXC provider (nothing will be outputted with `all`)

Operation

Salt's LXC support does use *lxc.init* via the *lxc.cloud_init_interface* and seeds the minion via *seed.mkconfig*.

You can provide to those lxc VMs a profile and a network profile like if you were directly using the minion module.

Order of operation:

- Create the LXC container on the desired minion (clone or template)
- Change LXC config options (if any need to be changed)
- Start container
- Change base passwords if any
- Change base DNS configuration if necessary
- Wait for LXC container to be up and ready for ssh
- Test SSH connection and bailout in error
- Upload deploy script and seeds, then re-attach the minion.

Provider configuration

Here is a simple provider configuration:

```
# Note: This example goes in /etc/salt/cloud.providers or any file in the
# /etc/salt/cloud.providers.d/ directory.
devhost10-lxc:
  target: devhost10
  driver: lxc
```

Note: Changed in version 2015.8.0.

The `provider` parameter in cloud provider definitions was renamed to `driver`. This change was made to avoid confusion with the `provider` parameter that is used in cloud profile definitions. Cloud provider definitions now use `driver` to refer to the Salt cloud module that provides the underlying functionality to connect to a cloud host, while cloud profiles continue to use `provider` to refer to provider configurations that you define.

Profile configuration

Please read *LXC Management with Salt* before anything else. And specially *Profiles*.

Here are the options to configure your containers:

target Host minion id to install the lxc Container into

lxc_profile Name of the profile or inline options for the LXC vm creation/cloning, please see *Container Profiles*.

network_profile Name of the profile or inline options for the LXC vm network settings, please see *Network Profiles*.

nic_opts Totally optional. Per interface new-style configuration options mappings which will override any profile default option:

```
eth0: {'mac': '00:16:3e:01:29:40',
              'gateway': None, (default)
              'link': 'br0', (default)
              'gateway': None, (default)
              'netmask': '', (default)
              'ip': '22.1.4.25'}}
```

password password for root and sysadmin users

dnsservers List of DNS servers to use. This is optional.

minion minion configuration (see *Minion Configuration in Salt Cloud*)

bootstrap_delay specify the time to wait (in seconds) between container creation and salt bootstrap execution. It is useful to ensure that all essential services have started before the bootstrap script is executed. By default there's no wait time between container creation and bootstrap unless you are on systemd where we wait that the system is no more in starting state.

bootstrap_shell shell for bootstraping script (default: /bin/sh)

script defaults to salt-boostrap

script_args arguments which are given to the bootstrap script. the {0} placeholder will be replaced by the path which contains the minion config and key files, eg:

```
script_args="-c {0}"
```

Using profiles:

```
# Note: This example would go in /etc/salt/cloud.profiles or any file in the
# /etc/salt/cloud.profiles.d/ directory.
devhost10-lxc:
  provider: devhost10-lxc
  lxc_profile: foo
  network_profile: bar
  minion:
    master: 10.5.0.1
    master_port: 4506
```

Using inline profiles (eg to override the network bridge):

```
devhost11-lxc:
  provider: devhost10-lxc
  lxc_profile:
    clone_from: foo
  network_profile:
    etho:
      link: lxcbr0
  minion:
    master: 10.5.0.1
    master_port: 4506
```

Using a lxc template instead of a clone:

```
devhost11-lxc:
  provider: devhost10-lxc
  lxc_profile:
    template: ubuntu
    # options:
    #   release: trusty
  network_profile:
    etho:
      link: lxcbr0
  minion:
    master: 10.5.0.1
    master_port: 4506
```

Static ip:

```
# Note: This example would go in /etc/salt/cloud.profiles or any file in the
# /etc/salt/cloud.profiles.d/ directory.
devhost10-lxc:
  provider: devhost10-lxc
  nic_opts:
    eth0:
      ipv4: 10.0.3.9
  minion:
    master: 10.5.0.1
    master_port: 4506
```

DHCP:

```
# Note: This example would go in /etc/salt/cloud.profiles or any file in the
# /etc/salt/cloud.profiles.d/ directory.
devhost10-lxc:
  provider: devhost10-lxc
  minion:
    master: 10.5.0.1
    master_port: 4506
```

Driver Support

- Container creation
- Image listing (LXC templates)
- Running container information (IP addresses, etc.)

14.7.15 Getting Started With 1and1

1&1 is one of the world's leading Web hosting providers. 1&1 currently offers a wide range of Web hosting products, including email solutions and high-end servers in 10 different countries including Germany, Spain, Great Britain and the United States. From domains to 1&1 MyWebsite to eBusiness solutions like Cloud Hosting and Web servers for complex tasks, 1&1 is well placed to deliver a high quality service to its customers. All 1&1 products are hosted in 1&1's high-performance, green data centers in the USA and Europe.

Dependencies

- 1and1 >= 1.2.0

Configuration

- Using the new format, set up the cloud configuration at `/etc/salt/cloud.providers` or `/etc/salt/cloud.providers.d/oneandone.conf`:

```
my-oneandone-config:
  driver: oneandone

  # Set the location of the salt-master
  #
  minion:
    master: saltmaster.example.com

  # Configure oneandone authentication credentials
  #
  api_token: <api_token>
  ssh_private_key: /path/to/id_rsa
  ssh_public_key: /path/to/id_rsa.pub
```

Authentication

The `api_key` is used for API authorization. This token can be obtained from the CloudPanel in the Management section below Users.

Profiles

Here is an example of a profile:

```
oneandone_fixed_size:
  provider: my-oneandone-config
  description: Small instance size server
  fixed_instance_size: S
  appliance_id: 8E3BAA98E3DFD37857810E0288DD8FBA

oneandone_custom_size:
  provider: my-oneandone-config
  description: Custom size server
  vcore: 2
  cores_per_processor: 2
  ram: 8
  appliance_id: 8E3BAA98E3DFD37857810E0288DD8FBA
  hdds:
  -
    is_main: true
    size: 20
  -
    is_main: false
    size: 20
```

The following list explains some of the important properties.

fixed_instance_size_id When creating a server, either `fixed_instance_size_id` or custom hardware params containing `vcore`, `cores_per_processor`, `ram`, and `hdds` must be provided. Can be one of the IDs listed among the output of the following command:

```
salt-cloud --list-sizes oneandone
```

vcore Total amount of processors.

cores_per_processor Number of cores per processor.

ram RAM memory size in GB.

hdds Hard disks.

appliance_id ID of the image that will be installed on server. Can be one of the IDs listed in the output of the following command:

```
salt-cloud --list-images oneandone
```

datacenter_id ID of the datacenter where the server will be created. Can be one of the IDs listed in the output of the following command:

```
salt-cloud --list-locations oneandone
```

description Description of the server.

password Password of the server. Password must contain more than 8 characters using uppercase letters, numbers and other special symbols.

power_on Power on server after creation. Default is set to true.

firewall_policy_id Firewall policy ID. If it is not provided, the server will assign the best firewall policy, creating a new one if necessary. If the parameter is sent with a 0 value, the server will be created with all ports blocked.

ip_id IP address ID.

load_balancer_id Load balancer ID.

monitoring_policy_id Monitoring policy ID.

deploy Set to False if Salt should not be installed on the node.

wait_for_timeout The timeout to wait in seconds for provisioning resources such as servers. The default wait_for_timeout is 15 minutes.

public_key_ids List of public key IDs (ssh key).

Functions

- Create an SSH key

```
sudo salt-cloud -f create_ssh_key my-oneandone-config name='SaltTest' description=
↪'SaltTestDescription'
```

- Create a block storage

```
sudo salt-cloud -f create_block_storage my-oneandone-config name='SaltTest2'⊠
↪description='SaltTestDescription' size=50 datacenter_id=
↪'5091F6D8CBFEF9C26ACE957C652D5D49'
```

For more information concerning cloud profiles, see *here*.

14.7.16 Getting Started with OpenNebula

OpenNebula is an open-source solution for the comprehensive management of virtualized data centers to enable the mixed use of private, public, and hybrid IaaS clouds.

Dependencies

The driver requires Python's `lxml` library to be installed. It also requires an OpenNebula installation running version `4.12` or greater.

Configuration

The following example illustrates some of the options that can be set. These parameters are discussed in more detail below.

```
# Note: This example is for /etc/salt/cloud.providers or any file in the
# /etc/salt/cloud.providers.d/ directory.

my-opennebula-provider:
  # Set up the location of the salt master
  #
  minion:
    master: saltmaster.example.com

  # Define xml_rpc setting which Salt-Cloud uses to connect to the OpenNebula API.⊠
↪Required.
  #
```

(continues on next page)

(continued from previous page)

```
  xml_rpc: http://localhost:2633/RPC2

  # Define the OpenNebula access credentials. This can be the main "oneadmin" user
↪that OpenNebula uses as the
  # OpenNebula main admin, or it can be a user defined in the OpenNebula instance.
↪Required.
  #
  user: oneadmin
  password: JHGhgsayu32jsa

  # Define the private key location that is used by OpenNebula to access new VMs.
↪This setting is required if
  # provisioning new VMs or accessing VMs previously created with the associated
↪public key.
  #
  private_key: /path/to/private/key

  driver: opennebula
```

Access Credentials

The Salt Cloud driver for OpenNebula was written using OpenNebula's native XML RPC API. Every interaction with OpenNebula's API requires a `username` and `password` to make the connection from the machine running Salt Cloud to API running on the OpenNebula instance. Based on the access credentials passed in, OpenNebula filters the commands that the user can perform or the information for which the user can query. For example, the images that a user can view with a `--list-images` command are the images that the connected user and the connected user's groups can access.

Key Pairs

Salt Cloud needs to be able to access a virtual machine in order to install the Salt Minion by using a public/private key pair. The virtual machine will need to be seeded with the public key, which is laid down by the OpenNebula template. Salt Cloud then uses the corresponding private key, provided by the `private_key` setting in the cloud provider file, to SSH into the new virtual machine.

To seed the virtual machine with the public key, the public key must be added to the OpenNebula template. If using the OpenNebula web interface, navigate to the template, then click `Update`. Click the `Context` tab. Under the `Network & SSH` section, click `Add SSH Contextualization` and paste the public key in the `Public Key` box. Don't forget to save your changes by clicking the green `Update` button.

Note: The key pair must not have a pass-phrase.

Cloud Profiles

Set up an initial profile at either `/etc/salt/cloud.profiles` or the `/etc/salt/cloud.profiles.d/` directory.

```
my-opennebula-profile:
  provider: my-opennebula-provider
  image: Ubuntu-14.04
```

The profile can now be realized with a salt command:

```
salt-cloud -p my-opennebula-profile my-new-vm
```

This will create a new instance named `my-new-vm` in OpenNebula. The minion that is installed on this instance will have a minion id of `my-new-vm`. If the command was executed on the salt-master, its Salt key will automatically be signed on the master.

Once the instance has been created with salt-minion installed, connectivity to it can be verified with Salt:

```
salt my-new-vm test.version
```

OpenNebula uses an image –> template –> virtual machine paradigm where the template draws on the image, or disk, and virtual machines are created from templates. Because of this, there is no need to define a `size` in the cloud profile. The size of the virtual machine is defined in the template.

Change Disk Size

You can now change the size of a VM on creation by cloning an image and expanding the size. You can accomplish this by the following cloud profile settings below.

```
my-opennebula-profile:
  provider: my-opennebula-provider
  image: Ubuntu-14.04
  disk:
    disk0:
      disk_type: clone
      size: 8096
      image: centos7-base-image-v2
    disk1:
      disk_type: volatile
      type: swap
      size: 4096
    disk2:
      disk_type: volatile
      size: 4096
      type: fs
      format: ext3
```

There are currently two different disk_types a user can use: volatile and clone. Clone which is required when specifying devices will clone an image in open nebula and will expand it to the size specified in the profile settings. By default this will clone the image attached to the template specified in the profile but a user can add the *image* argument under the disk definition.

For example the profile below will not use Ubuntu-14.04 for the cloned disk image. It will use the centos7-base-image image:

```
my-opennebula-profile:
  provider: my-opennebula-provider
  image: Ubuntu-14.04
  disk:
    disk0:
      disk_type: clone
      size: 8096
      image: centos7-base-image
```

If you want to use the image attached to the template set in the profile you can simply remove the image argument as show below. The profile below will clone the image Ubuntu-14.04 and expand the disk to 8GB.:

```
my-opennebula-profile:
  provider: my-opennebula-provider
  image: Ubuntu-14.04
  disk:
    disk0:
      disk_type: clone
      size: 8096
```

A user can also currently specify swap or fs disks. Below is an example of this profile setting:

```
my-opennebula-profile:
  provider: my-opennebula-provider
  image: Ubuntu-14.04
  disk:
    disk0:
      disk_type: clone
      size: 8096
    disk1:
      disk_type: volatile
      type: swap
      size: 4096
    disk2:
      disk_type: volatile
      size: 4096
      type: fs
      format: ext3
```

The example above will attach both a swap disk and a ext3 filesystem with a size of 4GB. To note if you define other disks you have to define the image disk to clone because the template will write over the entire 'DISK=[]' template definition on creation.

Required Settings

The following settings are always required for OpenNebula:

```
my-opennebula-config:
  xml_rpc: http://localhost:26633/RPC2
  user: oneadmin
  password: JHGhgsayu32jsa
  driver: opennebula
```

Required Settings for VM Deployment

The settings defined in the *Required Settings* section are required for all interactions with OpenNebula. However, when deploying a virtual machine via Salt Cloud, an additional setting, `private_key`, is also required:

```
my-opennebula-config:
  private_key: /path/to/private/key
```

Listing Images

Images can be queried on OpenNebula by passing the `--list-images` argument to Salt Cloud:

```
salt-cloud --list-images opennebula
```

Listing Locations

In OpenNebula, locations are defined as `hosts`. Locations, or ”hosts”, can be querried on OpenNebula by passing the `--list-locations` argument to Salt Cloud:

```
salt-cloud --list-locations opennebula
```

Listing Sizes

Sizes are defined by templates in OpenNebula. As such, the `--list-sizes` call returns an empty dictionary since there are no sizes to return.

Additional OpenNebula API Functionality

The Salt Cloud driver for OpenNebula was written using OpenNebula’s native XML RPC API. As such, many `--function` and `--action` calls were added to the OpenNebula driver to enhance support for an OpenNebula infrastructure with additional control from Salt Cloud. See the *`OpenNebula function definitions`* for more information.

Access via DNS entry instead of IP

Some OpenNebula installations do not assign IP addresses to new VMs, instead they establish the new VM’s hostname based on OpenNebula’s name of the VM, and then allocate an IP out of DHCP with dynamic DNS attaching the hostname. This driver supports this behavior by adding the entry *fqdn_base* to the driver configuration or the OpenNebula profile with a value matching the base fully-qualified domain. For example:

```
# Note: This example is for /etc/salt/cloud.providers or any file in the
# /etc/salt/cloud.providers.d/ directory.

my-opennebula-provider:
  [...]
  fqdn_base: corp.example.com
  [...]
```

14.7.17 Getting Started with Openstack

Openstack Cloud Driver

depends shade>=1.19.0

OpenStack is an open source project that is in use by a number a cloud providers, each of which have their own ways of using it.

This OpenStack driver uses a the shade python module which is managed by the OpenStack Infra team. This module is written to handle all the different versions of different OpenStack tools for salt, so most commands are just passed over to the module to handle everything.

Provider

There are two ways to configure providers for this driver. The first one is to just let shade handle everything, and configure using os-client-config and setting up */etc/openstack/clouds.yml*.

```yaml
clouds:
  democloud:
    region_name: RegionOne
    auth:
      username: 'demo'
      password: secret
      project_name: 'demo'
      auth_url: 'http://openstack/identity'
```

And then this can be referenced in the salt provider based on the *democloud* name.

```yaml
myopenstack:
  driver: openstack
  cloud: democloud
  region_name: RegionOne
```

This allows for just using one configuration for salt-cloud and for any other openstack tools which are all using */etc/openstack/clouds.yml*

The other method allows for specifying everything in the provider config, instead of using the extra configuration file. This will allow for passing salt-cloud configs only through pillars for minions without having to write a clouds.yml file on each minion.abs

```yaml
myopenstack:
  driver: openstack
  region_name: RegionOne
  auth:
    username: 'demo'
    password: secret
    project_name: 'demo'
    user_domain_name: default,
    project_domain_name: default,
    auth_url: 'http://openstack/identity'
```

Or if you need to use a profile to setup some extra stuff, it can be passed as a *profile* to use any of the vendor config options.

```yaml
myrackspace:
  driver: openstack
  profile: rackspace
  auth:
    username: rackusername
    api_key: myapikey
  region_name: ORD
  auth_type: rackspace_apikey
```

And this will pull in the profile for rackspace and setup all the correct options for the auth_url and different api versions for services.

Profile

Most of the options for building servers are just passed on to the create_server function from shade.

The salt specific ones are:

- ssh_key_file: The path to the ssh key that should be used to login to the machine to bootstrap it
- ssh_key_file: The name of the keypair in openstack
- userdata_template: The renderer to use if the userdata is a file that is templated. Default: False
- ssh_interface: The interface to use to login for bootstrapping: public_ips, private_ips, floating_ips, fixed_ips

```
centos:
  provider: myopenstack
  image: CentOS 7
  size: ds1G
  ssh_key_name: mykey
  ssh_key_file: /root/.ssh/id_rsa
```

This is the minimum setup required.

If metadata is set to make sure that the host has finished setting up the *wait_for_metadata* can be set.

```
centos:
  provider: myopenstack
  image: CentOS 7
  size: ds1G
  ssh_key_name: mykey
  ssh_key_file: /root/.ssh/id_rsa
  meta:
    build_config: rack_user_only
  wait_for_metadata:
    rax_service_level_automation: Complete
    rackconnect_automation_status: DEPLOYED
```

Anything else from the create_server docs can be passed through here.

- **image**: **Image dict, name or ID to boot with. image is required** unless boot_volume is given.
- **flavor**: Flavor dict, name or ID to boot onto.
- **auto_ip**: **Whether to take actions to find a routable IP for** the server. (defaults to True)
- **ips**: List of IPs to attach to the server (defaults to None)
- **ip_pool**: **Name of the network or floating IP pool to get an** address from. (defaults to None)
- **root_volume**: **Name or ID of a volume to boot from** (defaults to None - deprecated, use boot_volume)
- **boot_volume**: **Name or ID of a volume to boot from** (defaults to None)
- **terminate_volume**: **If booting from a volume, whether it should** be deleted when the server is destroyed. (defaults to False)
- **volumes**: (optional) A list of volumes to attach to the server
- **meta**: **(optional) A dict of arbitrary key/value metadata to** store for this server. Both keys and values must be <=255 characters.
- **files**: **(optional, deprecated) A dict of files to overwrite** on the server upon boot. Keys are file names (i.e. `/etc/passwd`) and values are the file contents (either as a string or as a file-like object). A maximum of five entries is allowed, and each file must be 10k or less.
- **reservation_id**: a UUID for the set of servers being requested.
- **min_count**: **(optional extension) The minimum number of** servers to launch.

- **max_count: (optional extension) The maximum number of** servers to launch.
- **security_groups**: A list of security group names
- **userdata: user data to pass to be exposed by the metadata** server this can be a file type object as well or a string.
- **key_name: (optional extension) name of previously created** keypair to inject into the instance.
- **availability_zone: Name of the availability zone for instance** placement.
- **block_device_mapping: (optional) A dict of block** device mappings for this server.
- **block_device_mapping_v2: (optional) A dict of block** device mappings for this server.
- **nics: (optional extension) an ordered list of nics to be** added to this server, with information about connected networks, fixed IPs, port etc.
- **scheduler_hints: (optional extension) arbitrary key-value pairs** specified by the client to help boot an instance
- **config_drive: (optional extension) value for config drive** either boolean, or volume-id
- **disk_config: (optional extension) control how the disk is** partitioned when the server is created. possible values are 'AUTO' or 'MANUAL'.
- **admin_pass: (optional extension) add a user supplied admin** password.
- **timeout: (optional) Seconds to wait, defaults to 60**. See the `wait` parameter.
- **reuse_ips: (optional) Whether to attempt to reuse pre-existing** floating ips should a floating IP be needed (defaults to True)
- **network: (optional) Network dict or name or ID to attach the** server to. Mutually exclusive with the nics parameter. Can also be be a list of network names or IDs or network dicts.
- **boot_from_volume: Whether to boot from volume. 'boot_volume'** implies True, but boot_from_volume=True with no boot_volume is valid and will create a volume from the image and use that.
- **volume_size: When booting an image from volume, how big should** the created volume be? Defaults to 50.
- **nat_destination: Which network should a created floating IP** be attached to, if it's not possible to infer from the cloud's configuration. (Optional, defaults to None)
- **group: ServerGroup dict, name or id to boot the server in**. If a group is provided in both scheduler_hints and in the group param, the group param will win. (Optional, defaults to None)

Note: If there is anything added, that is not in this list, it can be added to an *extras* dictionary for the profile, and that will be to the create_server function.

14.7.18 Getting Started With Parallels

Parallels Cloud Server is a product by Parallels that delivers a cloud hosting solution. The PARALLELS module for Salt Cloud enables you to manage instances hosted using PCS. Further information can be found at:

http://www.parallels.com/products/pcs/

- Using the old format, set up the cloud configuration at `/etc/salt/cloud`:

```
# Set up the location of the salt master
#
minion:
    master: saltmaster.example.com

# Set the PARALLELS access credentials (see below)
#
PARALLELS.user: myuser
PARALLELS.password: badpass

# Set the access URL for your PARALLELS host
#
PARALLELS.url: https://api.cloud.xmission.com:4465/paci/v1.0/
```

- Using the new format, set up the cloud configuration at `/etc/salt/cloud.providers` or `/etc/salt/cloud.providers.d/parallels.conf`:

```
my-parallels-config:
  # Set up the location of the salt master
  #
  minion:
    master: saltmaster.example.com

  # Set the PARALLELS access credentials (see below)
  #
  user: myuser
  password: badpass

  # Set the access URL for your PARALLELS provider
  #
  url: https://api.cloud.xmission.com:4465/paci/v1.0/
  driver: parallels
```

Note: Changed in version 2015.8.0.

The `provider` parameter in cloud provider definitions was renamed to `driver`. This change was made to avoid confusion with the `provider` parameter that is used in cloud profile definitions. Cloud provider definitions now use `driver` to refer to the Salt cloud module that provides the underlying functionality to connect to a cloud host, while cloud profiles continue to use `provider` to refer to provider configurations that you define.

Access Credentials

The `user`, `password`, and `url` will be provided to you by your cloud host. These are all required in order for the PARALLELS driver to work.

Cloud Profiles

Set up an initial profile at `/etc/salt/cloud.profiles` or `/etc/salt/cloud.profiles.d/parallels.conf`:

```
parallels-ubuntu:
    provider: my-parallels-config
    image: ubuntu-12.04-x86_64
```

The profile can be realized now with a salt command:

```
# salt-cloud -p parallels-ubuntu myubuntu
```

This will create an instance named `myubuntu` on the cloud host. The minion that is installed on this instance will have an `id` of `myubuntu`. If the command was executed on the salt-master, its Salt key will automatically be signed on the master.

Once the instance has been created with salt-minion installed, connectivity to it can be verified with Salt:

```
# salt myubuntu test.version
```

Required Settings

The following settings are always required for PARALLELS:

- Using the old cloud configuration format:

```
PARALLELS.user: myuser
PARALLELS.password: badpass
PARALLELS.url: https://api.cloud.xmission.com:4465/paci/v1.0/
```

- Using the new cloud configuration format:

```
my-parallels-config:
  user: myuser
  password: badpass
  url: https://api.cloud.xmission.com:4465/paci/v1.0/
  driver: parallels
```

Optional Settings

Unlike other cloud providers in Salt Cloud, Parallels does not utilize a `size` setting. This is because Parallels allows the end-user to specify a more detailed configuration for their instances than is allowed by many other cloud hosts. The following options are available to be used in a profile, with their default settings listed.

```
# Description of the instance. Defaults to the instance name.
desc: <instance_name>

# How many CPU cores, and how fast they are (in MHz)
cpu_number: 1
cpu_power: 1000

# How many megabytes of RAM
ram: 256

# Bandwidth available, in kbps
bandwidth: 100

# How many public IPs will be assigned to this instance
ip_num: 1

# Size of the instance disk (in GiB)
disk_size: 10
```

(continues on next page)

(continued from previous page)

```
# Username and password
ssh_username: root
password: <value from PARALLELS.password>

# The name of the image, from ``salt-cloud --list-images parallels``
image: ubuntu-12.04-x86_64
```

14.7.19 Getting Started With ProfitBricks

ProfitBricks provides an enterprise-grade Infrastructure as a Service (IaaS) solution that can be managed through a browser-based "Data Center Designer" (DCD) tool or via an easy to use API. A unique feature of the ProfitBricks platform is that it allows you to define your own settings for cores, memory, and disk size without being tied to a particular server size.

Dependencies

- profitbricks >= 4.1.1

Configuration

- Using the new format, set up the cloud configuration at `/etc/salt/cloud.providers` or `/etc/salt/cloud.providers.d/profitbricks.conf`:

```
my-profitbricks-config:
  driver: profitbricks

  # Set the location of the salt-master
  #
  minion:
    master: saltmaster.example.com

  # Configure ProfitBricks authentication credentials
  #
  username: user@domain.com
  password: 123456
  # datacenter is the UUID of a pre-existing virtual data center.
  datacenter: 9e6709a0-6bf9-4bd6-8692-60349c70ce0e
  # delete_volumes is forcing a deletion of all volumes attached to a server on a
↪deletion of a server
  delete_volumes: true
  # Connect to public LAN ID 1.
  public_lan: 1
  ssh_public_key: /path/to/id_rsa.pub
  ssh_private_key: /path/to/id_rsa
```

Note: Changed in version 2015.8.0.

The `provider` parameter in cloud provider definitions was renamed to `driver`. This change was made to avoid confusion with the `provider` parameter that is used in cloud profile definitions. Cloud provider definitions now use `driver` to refer to the Salt cloud module that provides the underlying functionality to connect to a cloud host, while cloud profiles continue to use `provider` to refer to provider configurations that you define.

Virtual Data Center

ProfitBricks uses the concept of Virtual Data Centers. These are logically separated from one another and allow you to have a self-contained environment for all servers, volumes, networking, snapshots, and so forth.

A list of existing virtual data centers can be retrieved with the following command:

```
salt-cloud -f list_datacenters my-profitbricks-config
```

A new data center can be created with the following command:

```
salt-cloud -f create_datacenter my-profitbricks-config name=example location=us/las␣
↪description="my description"
```

Authentication

The `username` and `password` are the same as those used to log into the ProfitBricks "Data Center Designer".

Profiles

Here is an example of a profile:

```
profitbricks_staging
  provider: my-profitbricks-config
  size: Micro Instance
  image_alias: 'ubuntu:latest'
  # image or image_alias must be provided
  # image: 2f98b678-6e7e-11e5-b680-52540066fee9
  cores: 2
  ram: 4096
  public_lan: 1
  private_lan: 2
  ssh_public_key: /path/to/id_rsa.pub
  ssh_private_key: /path/to/id_rsa
  ssh_interface: private_lan

profitbricks_production:
  provider: my-profitbricks-config
  image: Ubuntu-15.10-server-2016-05-01
  image_password: MyPassword1
  disk_type: SSD
  disk_size: 40
  cores: 8
  cpu_family: INTEL_XEON
  ram: 32768
  public_lan: 1
  public_ips:
    - 172.217.18.174
  private_lan: 2
  private_ips:
    - 192.168.100.10
  public_firewall_rules:
    Allow SSH:
      protocol: TCP
      source_ip: 1.2.3.4
```

(continues on next page)

(continued from previous page)

```
      port_range_start: 22
      port_range_end: 22
    Allow Ping:
      protocol: ICMP
      icmp_type: 8
  ssh_public_key: /path/to/id_rsa.pub
  ssh_private_key: /path/to/id_rsa
  ssh_interface: private_lan
  volumes:
    db_data:
      disk_size: 500
    db_log:
      disk_size: 50
      disk_type: SSD
```

Locations can be obtained using the `--list-locations` option for the `salt-cloud` command:

```
# salt-cloud --list-locations my-profitbricks-config
```

Images can be obtained using the `--list-sizes` option for the `salt-cloud` command:

```
# salt-cloud --list-images my-profitbricks-config
```

Sizes can be obtained using the `--list-sizes` option for the `salt-cloud` command:

```
# salt-cloud --list-sizes my-profitbricks-config
```

Changed in version 2019.2.0: One or more public IP address can be reserved with the following command:

```
# salt-cloud -f reserve_ipblock  my-profitbricks-config location='us/ewr' size=1
```

Profile Specifics:

The following list explains some of the important properties.

- `size` - Can be one of the options listed in the output of the following command:

  ```
  salt-cloud --list-sizes my-profitbricks-config
  ```

- `image` - Can be one of the options listed in the output of the following command:

  ```
  salt-cloud --list-images my-profitbricks-config
  ```

- `image_alias` - Can be one of the options listed in the output of the following command:

  ```
  salt-cloud -f list_images my-profitbricks-config
  ```

- `disk_size` - This option allows you to override the size of the disk as defined by the size. The disk size is set in gigabytes (GB).
- `disk_type` - This option allow the disk type to be set to HDD or SSD. The default is HDD.

  New in version 2019.2.0.

- `image_password` - A password is set on the image for the ”root” or ”Administrator” account. This field may only be set during volume creation. Only valid with ProfitBricks supplied HDD (not ISO) images. The password must contain at least 8 and no more than 50 characters. Only these characters are allowed: [a-z][A-Z][0-9]

- `cores` - This option allows you to override the number of CPU cores as defined by the size.
- `ram` - This option allows you to override the amount of RAM defined by the size. The value must be a multiple of 256, e.g. 256, 512, 768, 1024, and so forth.
- `public_lan` - This option will connect the server to the specified public LAN. If no LAN exists, then a new public LAN will be created. The value accepts a LAN ID (integer).

  New in version 2019.2.0.
- `public_ips` - Public IPs assigned to the NIC in the public LAN.
- `public_firewall_rules` - This option allows for a list of firewall rules assigned to the public network interface.

```
Firewall Rule Name:
  protocol: <protocol> (TCP, UDP, ICMP)
  source_mac: <source-mac>
  source_ip: <source-ip>
  target_ip: <target-ip>
  port_range_start: <port-range-start>
  port_range_end: <port-range-end>
  icmp_type: <icmp-type>
  icmp_code: <icmp-code>
```

- `private_lan` - This option will connect the server to the specified private LAN. If no LAN exists, then a new private LAN will be created. The value accepts a LAN ID (integer).

  New in version 2019.2.0.
- `private_ips` - Private IPs assigned in the private LAN. NAT setting is ignored when this setting is active.
- `private_firewall_rules` - This option allows for a list of firewall rules assigned to the private network interface.

```
Firewall Rule Name:
  protocol: <protocol> (TCP, UDP, ICMP)
  source_mac: <source-mac>
  source_ip: <source-ip>
  target_ip: <target-ip>
  port_range_start: <port-range-start>
  port_range_end: <port-range-end>
  icmp_type: <icmp-type>
  icmp_code: <icmp-code>
```

- `ssh_private_key` - Full path to the SSH private key file
- `ssh_public_key` - Full path to the SSH public key file
- `ssh_interface` - This option will use the private LAN IP for node connections (such as as bootstrapping the node) instead of the public LAN IP. The value accepts 'private_lan'.
- `cpu_family` - This option allow the CPU family to be set to AMD_OPTERON or INTEL_XEON. The default is AMD_OPTERON.
- `volumes` - This option allows a list of additional volumes by name that will be created and attached to the server. Each volume requires 'disk_size' and, optionally, 'disk_type'. The default is HDD.
- `deploy` - Set to `False` if Salt should not be installed on the node.
- `wait_for_timeout` - The timeout to wait in seconds for provisioning resources such as servers. The default wait_for_timeout is 15 minutes.

For more information concerning cloud profiles, see *here*.

14.7.20 Getting Started With Proxmox

Proxmox Virtual Environment is a complete server virtualization management solution, based on OpenVZ(in Proxmox up to 3.4)/LXC(from Proxmox 4.0 and up) and full virtualization with KVM. Further information can be found at:

http://www.proxmox.org/

Dependencies

- IPy >= 0.81
- requests >= 2.2.1

Please note: This module allows you to create OpenVZ/LXC containers and KVM VMs, but installing Salt on it will only be done on containers rather than a KVM virtual machine.

- Set up the cloud configuration at `/etc/salt/cloud.providers` or `/etc/salt/cloud.providers.d/proxmox.conf`:

```
my-proxmox-config:
  # Set up the location of the salt master
  #
  minion:
    master: saltmaster.example.com

  # Set the PROXMOX access credentials (see below)
  #
  user: myuser@pve
  password: badpass

  # Set the access URL for your PROXMOX host
  #
  url: your.proxmox.host
  driver: proxmox
```

Note: Changed in version 2015.8.0.

The `provider` parameter in cloud provider definitions was renamed to `driver`. This change was made to avoid confusion with the `provider` parameter that is used in cloud profile definitions. Cloud provider definitions now use `driver` to refer to the Salt cloud module that provides the underlying functionality to connect to a cloud host, while cloud profiles continue to use `provider` to refer to provider configurations that you define.

Access Credentials

The `user`, `password`, and `url` will be provided to you by your cloud host. These are all required in order for the PROXMOX driver to work.

Cloud Profiles

Set up an initial profile at `/etc/salt/cloud.profiles` or `/etc/salt/cloud.profiles.d/proxmox.conf`:

- Configure a profile to be used:

```
proxmox-ubuntu:
    provider: my-proxmox-config
    image: local:vztmpl/ubuntu-12.04-standard_12.04-1_amd64.tar.gz
    technology: lxc

    # host needs to be set to the configured name of the proxmox host
    # and not the ip address or FQDN of the server
    host: myvmhost
    ip_address: 192.168.100.155
    password: topsecret
```

The profile can be realized now with a salt command:

```
# salt-cloud -p proxmox-ubuntu myubuntu
```

This will create an instance named `myubuntu` on the cloud host. The minion that is installed on this instance will have a `hostname` of `myubuntu`. If the command was executed on the salt-master, its Salt key will automatically be signed on the master.

Once the instance has been created with salt-minion installed, connectivity to it can be verified with Salt:

```
# salt myubuntu test.version
```

Required Settings

The following settings are always required for PROXMOX:

- Using the new cloud configuration format:

```
my-proxmox-config:
  driver: proxmox
  user: saltcloud@pve
  password: xyzzy
  url: your.proxmox.host
```

Optional Settings

Unlike other cloud providers in Salt Cloud, Proxmox does not utilize a `size` setting. This is because Proxmox allows the end-user to specify a more detailed configuration for their instances, than is allowed by many other cloud providers. The following options are available to be used in a profile, with their default settings listed.

```
# Description of the instance.
desc: <instance_name>

# How many CPU cores, and how fast they are (in MHz)
cpus: 1
cpuunits: 1000

```

(continues on next page)

(continued from previous page)

```
# How many megabytes of RAM
memory: 256

# How much swap space in MB
swap: 256

# Whether to auto boot the vm after the host reboots
onboot: 1

# Size of the instance disk (in GiB)
disk: 10

# Host to create this vm on
host: myvmhost

# Nameservers. Defaults to host
nameserver: 8.8.8.8 8.8.4.4

# Username and password
ssh_username: root
password: <value from PROXMOX.password>

# The name of the image, from ``salt-cloud --list-images proxmox``
image: local:vztmpl/ubuntu-12.04-standard_12.04-1_amd64.tar.gz

# Whether or not to verify the SSL cert on the Proxmox host
verify_ssl: False

# Network interfaces, netX
net0: name=eth0,bridge=vmbr0,ip=dhcp

# Public key to add to /root/.ssh/authorized_keys.
pubkey: 'ssh-rsa AAAAB3NzaC1yc2EAAAADAQABA...'
```

QEMU

Some functionnalities works differently if you use 'qemu' as technology. In order to create a new VM with qemu, you need to specificy some more information. You can also clone a qemu template which already is on your Proxmox server.

QEMU profile file (for a new VM):

```
proxmox-win7:
  # Image of the new VM
  image: image.iso # You can get all your available images using 'salt-cloud --list-
↪images provider_name' (Ex: 'salt-cloud --list-images my-proxmox-config')

  # Technology used to create the VM ('qemu', 'openvz'(on Proxmox <4.x) or 'lxc'(on
↪Proxmox 4.x+))
  technology: qemu

  # Proxmox node name
  host: node_name

  # Proxmox password
```

(continues on next page)

(continued from previous page)

```yaml
  password: your_password

  # Workaround https://github.com/saltstack/salt/issues/27821
  size: ''

  # RAM size (MB)
  memory: 2048

  # OS Type enum (other / wxp / w2k / w2k3 / w2k8 / wvista / win7 / win8 / l24 / l26 /
↪ solaris)
  ostype: win7

  # Hard disk location
  sata0: <location>:<size>, format=<qcow2/vmdk/raw>, size=<size>GB #Example:
↪local:120,format=qcow2,size=120GB

  #CD/DVD Drive
  ide2: <content_location>,media=cdrom #Example: local:iso/name.iso,media=cdrom

  # Network Device
  net0:<model>,bridge=<bridge> #Example: e1000,bridge=vmbr0

  # Enable QEMU Guest Agent (0 / 1)
  agent: 1

  # VM name
  name: Test
```

More information about these parameters can be found on Proxmox API (http://pve.proxmox.com/pve2-api-doc/) under the 'POST' method of nodes/{node}/qemu

QEMU profile file (for a clone):

```yaml
proxmox-win7:
  # Enable Clone
  clone: True

  # New VM description
  clone_description: 'description'

  # New VM name
  clone_name: 'name'

  # New VM format (qcow2 / raw / vmdk)
  clone_format: qcow2

  # Full clone (1) or Link clone (0)
  clone_full: 0

  # VMID of Template to clone
  clone_from: ID

  # Technology used to create the VM ('qemu' or 'lxc')
  technology: qemu

  # Proxmox node name
  host: node_name
```

(continues on next page)

(continued from previous page)

```
  # Proxmox password
  password: your_password

  # Workaround https://github.com/saltstack/salt/issues/27821
  size: ''
```

More information can be found on Proxmox API under the 'POST' method of /nodes/{node}/qemu/{vmid}/clone

---

Note: The Proxmox API offers a lot more options and parameters, which are not yet supported by this salt-cloud 'overlay'. Feel free to add your contribution by forking the github repository and modifying the following file: `salt/cloud/clouds/proxmox.py`

An easy way to support more parameters for VM creation would be to add the names of the optional parameters in the 'create_nodes(vm_)' function, under the 'qemu' technology. But it requires you to dig into the code ...

14.7.21 Getting Started With Scaleway

Scaleway is the first IaaS host worldwide to offer an ARM based cloud. It's the ideal platform for horizontal scaling with BareMetal SSD servers. The solution provides on demand resources: it comes with on-demand SSD storage, movable IPs , images, security group and an Object Storage solution. https://scaleway.com

Configuration

Using Salt for Scaleway, requires an `access key` and an `API token`. `API tokens` are unique identifiers associated with your Scaleway account. To retrieve your `access key` and `API token`, log-in to the Scaleway control panel, open the pull-down menu on your account name and click on "My Credentials" link.

If you do not have API token you can create one by clicking the "Create New Token" button on the right corner.

```
# Note: This example is for /etc/salt/cloud.providers or any file in the
# /etc/salt/cloud.providers.d/ directory.

my-scaleway-config:
  access_key: 15cf404d-4560-41b1-9a0c-21c3d5c4ff1f
  token: a7347ec8-5de1-4024-a5e3-24b77d1ba91d
  driver: scaleway
```

Note: Changed in version 2015.8.0.

The `provider` parameter in cloud provider definitions was renamed to `driver`. This change was made to avoid confusion with the `provider` parameter that is used in cloud profile definitions. Cloud provider definitions now use `driver` to refer to the Salt cloud module that provides the underlying functionality to connect to a cloud host, while cloud profiles continue to use `provider` to refer to provider configurations that you define.

Profiles

Cloud Profiles

Set up an initial profile at /etc/salt/cloud.profiles or in the /etc/salt/cloud.profiles.d/ directory:

```
scaleway-ubuntu:
  provider: my-scaleway-config
  image: Ubuntu Trusty (14.04 LTS)
```

Images can be obtained using the `--list-images` option for the `salt-cloud` command:

```
#salt-cloud --list-images my-scaleway-config
my-scaleway-config:
  ----------
  scaleway:
      ----------
      069fd876-eb04-44ab-a9cd-47e2fa3e5309:
          ----------
          arch:
              arm
          creation_date:
              2015-03-12T09:35:45.764477+00:00
          default_bootscript:
              {u'kernel': {u'dtb': u'', u'title': u'Pimouss 3.2.34-30-std', u'id': u
↪'cfda4308-cd6f-4e51-9744-905fc0da370f', u'path': u'kernel/pimouss-uImage-3.2.34-30-
↪std'}, u'title': u'3.2.34-std #30 (stable)', u'id': u'c5af0215-2516-4316-befc-
↪5da1cfad609c', u'initrd': {u'path': u'initrd/c1-uInitrd', u'id': u'1be14b1b-e24c-
↪48e5-b0b6-7ba452e42b92', u'title': u'C1 initrd'}, u'bootcmdargs': {u'id': u
↪'d22c4dde-e5a4-47ad-abb9-d23b54d542ff', u'value': u'ip=dhcp boot=local root=/dev/
↪nbd0 USE_XNBD=1 nbd.max_parts=8'}, u'organization': u'11111111-1111-4111-8111-
↪111111111111', u'public': True}
          extra_volumes:
              []
          id:
              069fd876-eb04-44ab-a9cd-47e2fa3e5309
          modification_date:
              2015-04-24T12:02:16.820256+00:00
          name:
              Ubuntu Vivid (15.04)
          organization:
              a283af0b-d13e-42e1-a43f-855ffbf281ab
          public:
              True
          root_volume:
              {u'name': u'distrib-ubuntu-vivid-2015-03-12_10:32-snapshot', u'id': u
↪'a6d02e63-8dee-4bce-b627-b21730f35a05', u'volume_type': u'l_ssd', u'size':☒
↪50000000000L}
...
```

Execute a query and return all information about the nodes running on configured cloud providers using the `-Q` option for the `salt-cloud` command:

```
# salt-cloud -F
[INFO    ] salt-cloud starting
[INFO    ] Starting new HTTPS connection (1): api.scaleway.com
my-scaleway-config:
  ----------
  scaleway:
      ----------
      salt-manager:
          ----------
          creation_date:
```

(continues on next page)

(continued from previous page)

```
              2015-06-03T08:17:38.818068+00:00
          hostname:
              salt-manager
...
```

Note: Additional documentation about Scaleway can be found at https://www.scaleway.com/docs.

14.7.22 Getting Started With Saltify

The Saltify driver is a driver for installing Salt on existing machines (virtual or bare metal).

Dependencies

The Saltify driver has no external dependencies.

Configuration

Because the Saltify driver does not use an actual cloud provider host, it can have a simple provider configuration. The only thing that is required to be set is the driver name, and any other potentially useful information, like the location of the salt-master:

```
# Note: This example is for /etc/salt/cloud.providers file or any file in
# the /etc/salt/cloud.providers.d/ directory.

my-saltify-config:
  minion:
    master: 111.222.333.444
  driver: saltify
```

However, if you wish to use the more advanced capabilities of salt-cloud, such as rebooting, listing, and disconnecting machines, then the salt master must fill the role usually performed by a vendor's cloud management system. The salt master must be running on the salt-cloud machine, and created nodes must be connected to the master.

Additional information about which configuration options apply to which actions can be studied in the *Saltify Module documentation* and the *Miscellaneous Salt Cloud Options* document.

Profiles

Saltify requires a separate profile to be configured for each machine that needs Salt installed[1]. The initial profile can be set up at `/etc/salt/cloud.profiles` or in the `/etc/salt/cloud.profiles.d/` directory. Each profile requires both an `ssh_host` and an `ssh_username` key parameter as well as either an `key_filename` or a `password`.

Profile configuration example:

---

[1] Unless you are using a map file to provide the unique parameters.

```
# /etc/salt/cloud.profiles.d/saltify.conf

salt-this-machine:
  ssh_host: 12.34.56.78
  ssh_username: root
  key_filename: '/etc/salt/mysshkey.pem'
  provider: my-saltify-config
```

The machine can now be "Salted" with the following command:

```
salt-cloud -p salt-this-machine my-machine
```

This will install salt on the machine specified by the cloud profile, `salt-this-machine`, and will give the machine the minion id of `my-machine`. If the command was executed on the salt-master, its Salt key will automatically be accepted by the master.

Once a salt-minion has been successfully installed on the instance, connectivity to it can be verified with Salt:

```
salt my-machine test.version
```

Destroy Options

New in version 2018.3.0.

For obvious reasons, the `destroy` action does not actually vaporize hardware. If the salt master is connected, it can tear down parts of the client machines. It will remove the client's key from the salt master, and can execute the following options:

```
- remove_config_on_destroy: true
  # default: true
  # Deactivate salt-minion on reboot and
  # delete the minion config and key files from its "/etc/salt" directory,
  #   NOTE: If deactivation was unsuccessful (older Ubuntu machines) then when
  #   salt-minion restarts it will automatically create a new, unwanted, set
  #   of key files. Use the "force_minion_config" option to replace them.

- shutdown_on_destroy: false
  # default: false
  # last of all, send a "shutdown" command to the client.
```

Wake On LAN

New in version 2018.3.0.

In addition to connecting a hardware machine to a Salt master, you have the option of sending a wake-on-LAN magic packet to start that machine running.

The "magic packet" must be sent by an existing salt minion which is on the same network segment as the target machine. (Or your router must be set up especially to route WoL packets.) Your target machine must be set up to listen for WoL and to respond appropriately.

You must provide the Salt node id of the machine which will send the WoL packet (parameter `wol_sender_node`), and the hardware MAC address of the machine you intend to wake, (parameter `wake_on_lan_mac`). If both parameters are defined, the WoL will be sent. The cloud master will then sleep a while (parameter `wol_boot_wait`) to give the target machine time to boot up before we start probing its SSH port to begin deploying Salt to it. The default sleep time is 30 seconds.

```
# /etc/salt/cloud.profiles.d/saltify.conf

salt-this-machine:
  ssh_host: 12.34.56.78
  ssh_username: root
  key_filename: '/etc/salt/mysshkey.pem'
  provider: my-saltify-config
  wake_on_lan_mac: '00:e0:4c:70:2a:b2'  # found with ifconfig
  wol_sender_node: bevymaster  # its on this network segment
  wol_boot_wait: 45  # seconds to sleep
```

Using Map Files

The settings explained in the section above may also be set in a map file. An example of how to use the Saltify driver with a map file follows:

```
# /etc/salt/saltify-map

make_salty:
  - my-instance-0:
      ssh_host: 12.34.56.78
      ssh_username: root
      password: very-bad-password
  - my-instance-1:
      ssh_host: 44.33.22.11
      ssh_username: root
      password: another-bad-pass
```

In this example, the names `my-instance-0` and `my-instance-1` will be the identifiers of the deployed minions.

Note: The `ssh_host` directive is also used for Windows hosts, even though they do not typically run the SSH service. It indicates IP address or host name for the target system.

Note: When using a cloud map with the Saltify driver, the name of the profile to use, in this case `make_salty`, must be defined in a profile config. For example:

```
# /etc/salt/cloud.profiles.d/saltify.conf

make_salty:
  provider: my-saltify-config
```

The machines listed in the map file can now be ”Salted” by applying the following salt map command:

```
salt-cloud -m /etc/salt/saltify-map
```

This command will install salt on the machines specified in the map and will give each machine their minion id of `my-instance-0` and `my-instance-1`, respectively. If the command was executed on the salt-master, its Salt key will automatically be signed on the master.

Connectivity to the new ”Salted” instances can now be verified with Salt:

```
salt 'my-instance-*' test.version
```

Bulk Deployments

When deploying large numbers of Salt Minions using Saltify, it may be preferable to organize the configuration in a way that duplicates data as little as possible. For example, if a group of target systems have the same credentials, they can be specified in the profile, rather than in a map file.

```
# /etc/salt/cloud.profiles.d/saltify.conf

make_salty:
  provider: my-saltify-config
  ssh_username: root
  password: very-bad-password
```

```
# /etc/salt/saltify-map

make_salty:
  - my-instance-0:
      ssh_host: 12.34.56.78
  - my-instance-1:
      ssh_host: 44.33.22.11
```

If `ssh_host` is not provided, its default value will be the Minion identifier (`my-instance-0` and `my-instance-1`, in the example above). For deployments with working DNS resolution, this can save a lot of redundant data in the map. Here is an example map file using DNS names instead of IP addresses:

```
# /etc/salt/saltify-map

make_salty:
  - my-instance-0
  - my-instance-1
```

Credential Verification

Because the Saltify driver does not actually create VM's, unlike other salt-cloud drivers, it has special behaviour when the `deploy` option is set to `False`. When the cloud configuration specifies `deploy: False`, the Saltify driver will attempt to authenticate to the target node(s) and return `True` for each one that succeeds. This can be useful to verify ports, protocols, services and credentials are correctly configured before a live deployment.

Return values:

- `True`: Credential verification succeeded
- `False`: Credential verification succeeded
- `None`: Credential verification was not attempted.

14.7.23 Getting Started With SoftLayer

SoftLayer is a public cloud host, and baremetal hardware hosting service.

Dependencies

The SoftLayer driver for Salt Cloud requires the softlayer package, which is available at PyPI:

https://pypi.python.org/pypi/SoftLayer

This package can be installed using `pip` or `easy_install`:

```
# pip install softlayer
# easy_install softlayer
```

Configuration

Set up the cloud config at `/etc/salt/cloud.providers`:

```
# Note: These examples are for /etc/salt/cloud.providers

  my-softlayer:
    # Set up the location of the salt master
    minion:
      master: saltmaster.example.com

    # Set the SoftLayer access credentials (see below)
    user: MYUSER1138
    apikey: 'e3b68aa711e6deadc62d5b76355674beef7cc3116062ddbacafe5f7e465bfdc9'

    driver: softlayer


  my-softlayer-hw:
    # Set up the location of the salt master
    minion:
      master: saltmaster.example.com

    # Set the SoftLayer access credentials (see below)
    user: MYUSER1138
    apikey: 'e3b68aa711e6deadc62d5b76355674beef7cc3116062ddbacafe5f7e465bfdc9'

    driver: softlayer_hw
```

Note: Changed in version 2015.8.0.

The `provider` parameter in cloud provider definitions was renamed to `driver`. This change was made to avoid confusion with the `provider` parameter that is used in cloud profile definitions. Cloud provider definitions now use `driver` to refer to the Salt cloud module that provides the underlying functionality to connect to a cloud host, while cloud profiles continue to use `provider` to refer to provider configurations that you define.

Access Credentials

The `user` setting is the same user as is used to log into the SoftLayer Administration area. The `apikey` setting is found inside the Admin area after logging in:

- Hover over the `Account` menu item.
- Click the `Users` link.
- Find the `API Key` column and click `View`.

Profiles

Cloud Profiles

Set up an initial profile at `/etc/salt/cloud.profiles`:

```
base_softlayer_ubuntu:
  provider: my-softlayer
  image: UBUNTU_LATEST
  cpu_number: 1
  ram: 1024
  disk_size: 100
  local_disk: True
  hourly_billing: True
  domain: example.com
  location: sjc01
  # Optional
  max_net_speed: 1000
  private_vlan: 396
  private_network: True
  private_ssh: True
  # Use a dedicated host instead of cloud
  dedicated_host_id: 1234
  # May be used _instead_of_ image
  global_identifier: 320d8be5-46c0-dead-cafe-13e3c51
```

Most of the above items are required; optional items are specified below.

image

Images to build an instance can be found using the `--list-images` option:

```
# salt-cloud --list-images my-softlayer
```

The setting used will be labeled as `template`.

cpu_number

This is the number of CPU cores that will be used for this instance. This number may be dependent upon the image that is used. For instance:

```
Red Hat Enterprise Linux 6 - Minimal Install (64 bit) (1 - 4 Core):
    ----------
    name:
        Red Hat Enterprise Linux 6 - Minimal Install (64 bit) (1 - 4 Core)
    template:
        REDHAT_6_64
Red Hat Enterprise Linux 6 - Minimal Install (64 bit) (5 - 100 Core):
    ----------
    name:
        Red Hat Enterprise Linux 6 - Minimal Install (64 bit) (5 - 100 Core)
    template:
        REDHAT_6_64
```

Note that the template (meaning, the *image* option) for both of these is the same, but the names suggests how many CPU cores are supported.

ram

This is the amount of memory, in megabytes, that will be allocated to this instance.

disk_size

The amount of disk space that will be allocated to this image, in gigabytes.

```yaml
base_softlayer_ubuntu:
  disk_size: 100
```

Using Multiple Disks

New in version 2015.8.1.

SoftLayer allows up to 5 disks to be specified for a virtual machine upon creation. Multiple disks can be specified either as a list or a comma-delimited string. The first `disk_size` specified in the string or list will be the first disk size assigned to the VM.

List Example: .. code-block:: yaml

> **base_softlayer_ubuntu:** disk_size: ['100', '20', '20']

String Example: .. code-block:: yaml

> **base_softlayer_ubuntu:** disk_size: '100, 20, 20'

local_disk

When true the disks for the computing instance will be provisioned on the host which it runs, otherwise SAN disks will be provisioned.

hourly_billing

When true the computing instance will be billed on hourly usage, otherwise it will be billed on a monthly basis.

domain

The domain name that will be used in the FQDN (Fully Qualified Domain Name) for this instance. The *domain* setting will be used in conjunction with the instance name to form the FQDN.

use_fqdn

If set to True, the Minion will be identified by the FQDN (Fully Qualified Domain Name) which is a result of combining the `domain` configuration value and the Minion name specified either via the CLI or a map file rather than only using the short host name, or Minion ID. Default is False.

New in version 2016.3.0.

For example, if the value of `domain` is `example.com` and a new VM was created via the CLI with `salt-cloud -p base_softlayer_ubuntu my-vm`, the resulting Minion ID would be `my-vm.example.com`.

Note: When enabling the `use_fqdn` setting, the Minion ID will be the FQDN and will interact with salt commands with the FQDN instead of the short hostname. However, due to the way the SoftLayer API is constructed, some Salt Cloud functions such as listing nodes or destroying VMs will only list the short hostname of the VM instead of the FQDN.

Example output displaying the SoftLayer hostname quirk mentioned in the note above (note the Minion ID is `my-vm.example.com`, but the VM to be destroyed is listed with its short hostname, `my-vm`):

```
# salt-key -L
Accepted Keys:
my-vm.example.com
Denied Keys:
Unaccepted Keys:
Rejected Keys:
#
#
# salt my-vm.example.com test.version
my-vm.example.com:
    2018.3.4
#
#
# salt-cloud -d my-vm.example.com
[INFO    ] salt-cloud starting
[INFO    ] POST https://api.softlayer.com/xmlrpc/v3.1/SoftLayer_Account
The following virtual machines are set to be destroyed:
  softlayer-config:
    softlayer:
      my-vm

Proceed? [N/y] y
... proceeding
[INFO    ] Destroying in non-parallel mode.
[INFO    ] POST https://api.softlayer.com/xmlrpc/v3.1/SoftLayer_Account
[INFO    ] POST https://api.softlayer.com/xmlrpc/v3.1/SoftLayer_Virtual_Guest
softlayer-config:
    ----------
    softlayer:
        ----------
        my-vm:
            True
```

location

Images to build an instance can be found using the *–list-locations* option:

```
# salt-cloud --list-location my-softlayer
```

max_net_speed

Specifies the connection speed for the instance's network components. This setting is optional. By default, this is set to 10.

post_uri

Specifies the uri location of the script to be downloaded and run after the instance is provisioned.

New in version 2015.8.1.

Example: .. code-block:: yaml

> **base_softlayer_ubuntu**: post_uri: 'https://SOMESERVERIP:8000/myscript.sh'

public_vlan

If it is necessary for an instance to be created within a specific frontend VLAN, the ID for that VLAN can be specified in either the provider or profile configuration.

This ID can be queried using the *list_vlans* function, as described below. This setting is optional.

If this setting is set to *None*, salt-cloud will connect to the private ip of the server.

Note: If this setting is not provided and the server is not built with a public vlan, *private_ssh* or *private_wds* will need to be set to make sure that salt-cloud attempts to connect to the private ip.

private_vlan

If it is necessary for an instance to be created within a specific backend VLAN, the ID for that VLAN can be specified in either the provider or profile configuration.

This ID can be queried using the *list_vlans* function, as described below. This setting is optional.

private_network

If a server is to only be used internally, meaning it does not have a public VLAN associated with it, this value would be set to True. This setting is optional. The default is False.

private_ssh or private_wds

Whether to run the deploy script on the server using the public IP address or the private IP address. If set to True, Salt Cloud will attempt to SSH or WinRM into the new server using the private IP address. The default is False. This settiong is optional.

global_identifier

When creating an instance using a custom template, this option is set to the corresponding value obtained using the *list_custom_images* function. This option will not be used if an *image* is set, and if an *image* is not set, it is required.

The profile can be realized now with a salt command:

```
# salt-cloud -p base_softlayer_ubuntu myserver
```

Using the above configuration, this will create *myserver.example.com*.

Once the instance has been created with salt-minion installed, connectivity to it can be verified with Salt:

```
# salt 'myserver.example.com' test.version
```

Dedicated Host

Soflayer allows the creation of new VMs in a dedicated host. This means that you can order and pay a fixed amount for a bare metal dedicated host and use it to provision as many VMs as you can fit in there. If you want your VMs to be launched in a dedicated host, instead of Sofltayer's cloud, set the `dedicated_host_id` parameter in your profile.

dedicated_host_id

The id of the dedicated host where the VMs should be created. If not set, VMs will be created in Softlayer's cloud instead.

Bare metal Profiles

Set up an initial profile at `/etc/salt/cloud.profiles`:

```
base_softlayer_hw_centos:
  provider: my-softlayer-hw
  # CentOS 6.0 - Minimal Install (64 bit)
  image: 13963
  # 2 x 2.0 GHz Core Bare Metal Instance - 2 GB Ram
  size: 1921
  # 500GB SATA II
  hdd: 1267
  # San Jose 01
  location: 168642
  domain: example.com
  # Optional
  vlan: 396
  port_speed: 273
  banwidth: 248
```

Most of the above items are required; optional items are specified below.

image

Images to build an instance can be found using the *–list-images* option:

```
# salt-cloud --list-images my-softlayer-hw
```

A list of *id's and names will be provided. The 'name* will describe the operating system and architecture. The *id* will be the setting to be used in the profile.

size

Sizes to build an instance can be found using the *–list-sizes* option:

```
# salt-cloud --list-sizes my-softlayer-hw
```

A list of *id‘s and names will be provided. The ‘name* will describe the speed and quantity of CPU cores, and the amount of memory that the hardware will contain. The *id* will be the setting to be used in the profile.

hdd

There is currently only one size of hard disk drive (HDD) that is available for hardware instances on SoftLayer:

```
1267: 500GB SATA II
```

The *hdd* setting in the profile should be 1267. Other sizes may be added in the future.

location

Locations to build an instance can be found using the *–list-images* option:

```
# salt-cloud --list-locations my-softlayer-hw
```

A list of IDs and names will be provided. The *location* will describe the location in human terms. The *id* will be the setting to be used in the profile.

domain

The domain name that will be used in the FQDN (Fully Qualified Domain Name) for this instance. The *domain* setting will be used in conjunction with the instance name to form the FQDN.

vlan

If it is necessary for an instance to be created within a specific VLAN, the ID for that VLAN can be specified in either the provider or profile configuration.

This ID can be queried using the *list_vlans* function, as described below.

port_speed

Specifies the speed for the instance's network port. This setting refers to an ID within the SoftLayer API, which sets the port speed. This setting is optional. The default is 273, or, 100 Mbps Public & Private Networks. The following settings are available:

- 273: 100 Mbps Public & Private Networks
- 274: 1 Gbps Public & Private Networks
- 21509: 10 Mbps Dual Public & Private Networks (up to 20 Mbps)
- 21513: 100 Mbps Dual Public & Private Networks (up to 200 Mbps)
- 2314: 1 Gbps Dual Public & Private Networks (up to 2 Gbps)
- 272: 10 Mbps Public & Private Networks

bandwidth

Specifies the network bandwidth available for the instance. This setting refers to an ID within the SoftLayer API, which sets the bandwidth. This setting is optional. The default is 248, or, 5000 GB Bandwidth. The following settings are available:

- 248: 5000 GB Bandwidth
- 129: 6000 GB Bandwidth
- 130: 8000 GB Bandwidth
- 131: 10000 GB Bandwidth
- 36: Unlimited Bandwidth (10 Mbps Uplink)
- 125: Unlimited Bandwidth (100 Mbps Uplink)

Actions

The following actions are currently supported by the SoftLayer Salt Cloud driver.

show_instance

This action is a thin wrapper around *–full-query*, which displays details on a single instance only. In an environment with several machines, this will save a user from having to sort through all instance data, just to examine a single instance.

```
$ salt-cloud -a show_instance myinstance
```

Functions

The following functions are currently supported by the SoftLayer Salt Cloud driver.

list_vlans

This function lists all VLANs associated with the account, and all known data from the SoftLayer API concerning those VLANs.

```
$ salt-cloud -f list_vlans my-softlayer
$ salt-cloud -f list_vlans my-softlayer-hw
```

The *id* returned in this list is necessary for the *vlan* option when creating an instance.

list_custom_images

This function lists any custom templates associated with the account, that can be used to create a new instance.

```
$ salt-cloud -f list_custom_images my-softlayer
```

The *globalIdentifier* returned in this list is necessary for the *global_identifier* option when creating an image using a custom template.

Optional Products for SoftLayer HW

The softlayer_hw driver supports the ability to add optional products, which are supported by SoftLayer's API. These products each have an ID associated with them, that can be passed into Salt Cloud with the *optional_products* option:

```
softlayer_hw_test:
  provider: my-softlayer-hw
  # CentOS 6.0 - Minimal Install (64 bit)
  image: 13963
  # 2 x 2.0 GHz Core Bare Metal Instance - 2 GB Ram
  size: 1921
  # 500GB SATA II
  hdd: 1267
  # San Jose 01
  location: 168642
  domain: example.com
  optional_products:
    # MySQL for Linux
    - id: 28
    # Business Continuance Insurance
    - id: 104
```

These values can be manually obtained by looking at the source of an order page on the SoftLayer web interface. For convenience, many of these values are listed here:

Public Secondary IP Addresses

- 22: 4 Public IP Addresses
- 23: 8 Public IP Addresses

Primary IPv6 Addresses

- 17129: 1 IPv6 Address

Public Static IPv6 Addresses

- 1481: /64 Block Static Public IPv6 Addresses

OS-Specific Addon

- 17139: XenServer Advanced for XenServer 6.x
- 17141: XenServer Enterprise for XenServer 6.x
- 2334: XenServer Advanced for XenServer 5.6
- 2335: XenServer Enterprise for XenServer 5.6
- 13915: Microsoft WebMatrix
- 21276: VMware vCenter 5.1 Standard

Control Panel Software

- 121: cPanel/WHM with Fantastico and RVskin
- 20778: Parallels Plesk Panel 11 (Linux) 100 Domain w/ Power Pack
- 20786: Parallels Plesk Panel 11 (Windows) 100 Domain w/ Power Pack
- 20787: Parallels Plesk Panel 11 (Linux) Unlimited Domain w/ Power Pack
- 20792: Parallels Plesk Panel 11 (Windows) Unlimited Domain w/ Power Pack
- 2340: Parallels Plesk Panel 10 (Linux) 100 Domain w/ Power Pack
- 2339: Parallels Plesk Panel 10 (Linux) Unlimited Domain w/ Power Pack
- 13704: Parallels Plesk Panel 10 (Windows) Unlimited Domain w/ Power Pack

Database Software

- 29: MySQL 5.0 for Windows
- 28: MySQL for Linux
- 21501: Riak 1.x
- 20893: MongoDB
- 30: Microsoft SQL Server 2005 Express
- 92: Microsoft SQL Server 2005 Workgroup
- 90: Microsoft SQL Server 2005 Standard
- 94: Microsoft SQL Server 2005 Enterprise
- 1330: Microsoft SQL Server 2008 Express
- 1340: Microsoft SQL Server 2008 Web
- 1337: Microsoft SQL Server 2008 Workgroup
- 1334: Microsoft SQL Server 2008 Standard
- 1331: Microsoft SQL Server 2008 Enterprise
- 2179: Microsoft SQL Server 2008 Express R2
- 2173: Microsoft SQL Server 2008 Web R2
- 2183: Microsoft SQL Server 2008 Workgroup R2
- 2180: Microsoft SQL Server 2008 Standard R2
- 2176: Microsoft SQL Server 2008 Enterprise R2

Anti-Virus & Spyware Protection

- 594: McAfee VirusScan Anti-Virus - Windows
- 414: McAfee Total Protection - Windows

Insurance

- 104: Business Continuance Insurance

Monitoring

- 55: Host Ping
- 56: Host Ping and TCP Service Monitoring

Notification

- 57: Email and Ticket

Advanced Monitoring

- 2302: Monitoring Package - Basic
- 2303: Monitoring Package - Advanced
- 2304: Monitoring Package - Premium Application

Response

- 58: Automated Notification
- 59: Automated Reboot from Monitoring
- 60: 24x7x365 NOC Monitoring, Notification, and Response

Intrusion Detection & Protection

- 413: McAfee Host Intrusion Protection w/Reporting

Hardware & Software Firewalls

- 411: APF Software Firewall for Linux
- 894: Microsoft Windows Firewall
- 410: 10Mbps Hardware Firewall
- 409: 100Mbps Hardware Firewall
- 408: 1000Mbps Hardware Firewall

14.7.24 Getting Started With Tencent Cloud

Tencent Cloud is a secure, reliable and high-performance cloud compute service provided by Tencent. It is the 2nd largest Cloud Provider in China.

Dependencies

The Tencent Cloud driver for Salt Cloud requires the `tencentcloud-sdk-python` package, which is available at PyPI:

https://pypi.org/project/tencentcloud-sdk-python/

This package can be installed using `pip` or `easy_install`:

```
# pip install tencentcloud-sdk-python
# easy_install tencentcloud-sdk-python
```

Provider Configuration

To use this module, set up the cloud configuration at `/etc/salt/cloud.providers` or `/etc/salt/cloud.providers.d/*.conf`:

```
my-tencentcloud-config:
  driver: tencentcloud
  # Tencent Cloud Secret Id
  id: AKIDA64pOio9BMemkApzevX0HS169S4b750A
  # Tencent Cloud Secret Key
  key: 8r2xmPn0C5FDvRAlmcJimiTZKVRsk260
  # Tencent Cloud Region
  location: ap-guangzhou
```

Configuration Parameters

driver

Required. `tencentcloud` to use this module.

id

Required. Your Tencent Cloud secret id.

key

Required. Your Tencent Cloud secret key.

location

Optional. If this value is not specified, the default is `ap-guangzhou`. Available locations can be found using the `--list-locations` option:

```
# salt-cloud --list-location my-tencentcloud-config
```

Profile Configuration

Tencent Cloud profiles require a `provider`, `availability_zone`, `image` and `size`. Set up an initial profile at `/etc/salt/cloud.profiles` or `/etc/salt/cloud.profiles.d/*.conf`:

```
tencentcloud-guangzhou-s1sm1:
    provider: my-tencentcloud-config
    availability_zone: ap-guangzhou-3
    image: img-31tjrtph
    size: S1.SMALL1
    allocate_public_ip: True
    internet_max_bandwidth_out: 1
    password: '153e41ec96140152'
    securitygroups:
        - sg-5e90804b
```

Configuration Parameters

provider

Required. Name of entry in `salt/cloud.providers.d/???` file.

availability_zone

Required. The availability zone that the instance is located in. Available zones can be found using the `list_availability_zones` function:

```
# salt-cloud -f list_availability_zones my-tencentcloud-config
```

image

Required. The image id to use for the instance. Available images can be found using the `--list-images` option:

```
# salt-cloud --list-images my-tencentcloud-config
```

size

Required. Instance type for instance can be found using the `--list-sizes` option.

```
# salt-cloud --list-sizes my-tencentcloud-config
```

securitygroups

Optional. A list of security group ids to associate with. Available security group ids can be found using the `list_securitygroups` function:

```
# salt-cloud -f list_securitygroups my-tencentcloud-config
```

Multiple security groups are supported:

```
tencentcloud-guangzhou-s1sm1:
    securitygroups:
        - sg-5e90804b
        - sg-8kpynf2t
```

hostname

Optional. The hostname of the instance.

instance_charge_type

Optional. The charge type of the instance. Valid values are `PREPAID`, `POSTPAID_BY_HOUR` and `SPOTPAID`. The default is `POSTPAID_BY_HOUR`.

instance_charge_type_prepaid_renew_flag

Optional. When enabled, the instance will be renew automatically when it reaches the end of the prepaid tenancy. Valid values are `NOTIFY_AND_AUTO_RENEW`, `NOTIFY_AND_MANUAL_RENEW` and `DISABLE_NOTIFY_AND_MANUAL_RENEW`.

Note: This value is only used when `instance_charge_type` is set to `PREPAID`.

instance_charge_type_prepaid_period

Optional. The tenancy time in months of the prepaid instance, Valid values are `1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 24, 36`.

Note: This value is only used when `instance_charge_type` is set to `PREPAID`.

allocate_public_ip

Optional. Associate a public ip address with an instance in a VPC or Classic. Boolean value, default is `false`.

internet_max_bandwidth_out

Optional. Maximum outgoing bandwidth to the public network, measured in Mbps (Mega bits per second). Value range: `[0, 100]`. If this value is not specified, the default is `0` Mbps.

internet_charge_type

Optional. Internet charge type of the instance. Valid values are `BANDWIDTH_PREPAID`, `TRAFFIC_POSTPAID_BY_HOUR`, `BANDWIDTH_POSTPAID_BY_HOUR` and `BANDWIDTH_PACKAGE`. The default is `TRAFFIC_POSTPAID_BY_HOUR`.

key_name

Optional. The key pair to use for the instance, for example `skey-16jig7tx`.

password

Optional. Login password for the instance.

private_ip

Optional. The private ip to be assigned to this instance, must be in the provided subnet and available.

project_id

Optional. The project this instance belongs to, defaults to `0`.

vpc_id

Optional. The id of a VPC network. If you want to create instances in a VPC network, this parameter must be set.

subnet_id

Optional. The id of a VPC subnet. If you want to create instances in VPC network, this parameter must be set.

system_disk_size

Optional. Size of the system disk. Value range: `[50, 1000]`, and unit is `GB`. Default is `50` GB.

system_disk_type

Optional. Type of the system disk. Valid values are `CLOUD_BASIC`, `CLOUD_SSD` and `CLOUD_PREMIUM`, default value is `CLOUD_BASIC`.

Actions

The following actions are supported by the Tencent Cloud Salt Cloud driver.

show_instance

This action is a thin wrapper around `--full-query`, which displays details on a single instance only. In an environment with several machines, this will save a user from having to sort through all instance data, just to examine a single instance.

```
$ salt-cloud -a show_instance myinstance
```

show_disk

Return disk details about a specific instance.

```
$ salt-cloud -a show_disk myinstance
```

destroy

Destroy a Tencent Cloud instance.

```
$ salt-cloud -a destroy myinstance
```

start

Start a Tencent Cloud instance.

```
$ salt-cloud -a start myinstance
```

stop

Stop a Tencent Cloud instance.

```
$ salt-cloud -a stop myinstance
```

reboot

Reboot a Tencent Cloud instance.

```
$ salt-cloud -a reboot myinstance
```

Functions

The following functions are currently supported by the Tencent Cloud Salt Cloud driver.

list_securitygroups

Lists all Tencent Cloud security groups in current region.

```
$ salt-cloud -f list_securitygroups my-tencentcloud-config
```

list_availability_zones

Lists all Tencent Cloud availability zones in current region.

```
$ salt-cloud -f list_availability_zones my-tencentcloud-config
```

list_custom_images

Lists any custom images associated with the account. These images can be used to create a new instance.

```
$ salt-cloud -f list_custom_images my-tencentcloud-config
```

show_image

Return details about a specific image. This image can be used to create a new instance.

```
$ salt-cloud -f show_image tencentcloud image=img-31tjrtph
```

14.7.25 Getting Started With Vagrant

The Vagrant driver is a new, experimental driver for spinning up a VagrantBox virtual machine, and installing Salt on it.

Dependencies

The Vagrant driver itself has no external dependencies.

The machine which will host the VagrantBox must be an already existing minion of the cloud server's Salt master. It must have Vagrant installed, and a Vagrant-compatible virtual machine engine, such as VirtualBox. (Note: The Vagrant driver does not depend on the salt-cloud VirtualBox driver in any way.)

[Caution: The version of Vagrant packaged for `apt install` in Ubuntu 16.04 will not connect a bridged network adapter correctly. Use a version downloaded directly from the web site.]

Include the Vagrant guest editions plugin: `vagrant plugin install vagrant-vbguest`.

Configuration

Configuration of the client virtual machine (using VirtualBox, VMware, etc) will be done by Vagrant as specified in the Vagrantfile on the host machine.

Salt-cloud will push the commands to install and provision a salt minion on the virtual machine, so you need not (perhaps **should** not) provision salt in your Vagrantfile, in most cases.

If, however, your cloud master cannot open an SSH connection to the child VM, you may **need** to let Vagrant provision the VM with Salt, and use some other method (such as passing a pillar dictionary to the VM) to pass the master's IP address to the VM. The VM can then attempt to reach the salt master in the usual way for non-cloud minions. Specify the profile configuration argument as `deploy: False` to prevent the cloud master from trying.

```
# Note: This example is for /etc/salt/cloud.providers file or any file in
# the /etc/salt/cloud.providers.d/ directory.

my-vagrant-config:
  minion:
    master: 111.222.333.444
  provider: vagrant
```

Because the Vagrant driver needs a place to store the mapping between the node name you use for Salt commands and the Vagrantfile which controls the VM, you must configure your salt minion as a Salt smb server. (See *host provisioning example* below.)

Profiles

Vagrant requires a profile to be configured for each machine that needs Salt installed. The initial profile can be set up at `/etc/salt/cloud.profiles` or in the `/etc/salt/cloud.profiles.d/` directory.

Each profile requires a `vagrantfile` parameter. If the Vagrantfile has definitions for multiple machines then you need a `machine` parameter,

Salt-cloud uses SSH to provision the minion. There must be a routable path from the cloud master to the VM. Usually, you will want to use a bridged network adapter for SSH. The address may not be known until DHCP assigns it. If `ssh_host` is not defined, and `target_network` is defined, the driver will attempt to read the address from the output of an `ifconfig` command. Lacking either setting, the driver will try to use the value Vagrant returns as its `ssh_host`, which will work only if the cloud master is running somewhere on the same host.

The `target_network` setting should be used to identify the IP network your bridged adapter is expected to appear on. Use CIDR notation, like `target_network: '2001:DB8::/32'` or `target_network: '192.0.2.0/24'`.

Profile configuration example:

```
# /etc/salt/cloud.profiles.d/vagrant.conf

vagrant-machine:
  host: my-vhost  # the Salt id of the virtual machine's host computer.
  provider: my-vagrant-config
  cwd: /srv/machines  # the path to your Vagrantfile.
  vagrant_runas: my-username  # the username who defined the Vagrantbox on the host
  # vagrant_up_timeout: 300 # (seconds) timeout for cmd.run of the "vagrant up"␣
↪command
  # vagrant_provider: '' # option for "vagrant up" like: "--provider vmware_fusion"
  # ssh_host: None  # "None" means try to find the routable IP address from "ifconfig"
  # ssh_username: '' # also required when ssh_host is used.
  # target_network: None  # Expected CIDR address range of your bridged network
  # force_minion_config: false  # Set "true" to re-purpose an existing VM
```

The machine can now be created and configured with the following command:

```
salt-cloud -p vagrant-machine my-id
```

This will create the machine specified by the cloud profile `vagrant-machine`, and will give the machine the minion id of `my-id`. If the cloud master is also the salt-master, its Salt key will automatically be accepted on the master.

Once a salt-minion has been successfully installed on the instance, connectivity to it can be verified with Salt:

```
salt my-id test.version
```

Provisioning a Vagrant cloud host (example)

In order to query or control minions it created, each host minion needs to track the Salt node names associated with any guest virtual machines on it. It does that using a Salt sdb database.

The Salt sdb is not configured by default. The following example shows a simple installation.

This example assumes:

- you are on a large network using the 10.x.x.x IP address space
- your Salt master's Salt id is "bevymaster"

- it will also be your salt-cloud controller
- it is at hardware address 10.124.30.7
- it is running a recent Debian family Linux (raspbian)
- your workstation is a Salt minion of bevymaster
- your workstation's minion id is "my_laptop"
- VirtualBox has been installed on "my_laptop" (apt install is okay)
- Vagrant was installed from vagrantup.com. (not the 16.04 Ubuntu apt)
- "my_laptop" has done "vagrant plugin install vagrant-vbguest"
- the VM you want to start is on "my_laptop" at "/home/my_username/Vagrantfile"

```
# file /etc/salt/minion.d/vagrant_sdb.conf on host computer "my_laptop"
#  -- this sdb database is required by the Vagrant module --
vagrant_sdb_data:  # The sdb database must have this name.
  driver: sqlite3  # Let's use SQLite to store the data ...
  database: /var/cache/salt/vagrant.sqlite  # ... in this file ...
  table: sdb  # ... using this table name.
  create_table: True  # if not present
```

Remember to re-start your minion after changing its configuration files…

```
sudo systemctl restart salt-minion
```

```
# -*- mode: ruby -*-
# file /home/my_username/Vagrantfile on host computer "my_laptop"
BEVY = "bevy1"
DOMAIN = BEVY + ".test"  # .test is an ICANN reserved non-public TLD

# must supply a list of names to avoid Vagrant asking for interactive input
def get_good_ifc()   # try to find a working Ubuntu network adapter name
  addr_infos = Socket.getifaddrs
  addr_infos.each do |info|
    a = info.addr
    if a and a.ip? and not a.ip_address.start_with?("127.")
     return info.name
     end
  end
  return "eth0"  # fall back to an old reliable name
end

Vagrant.configure(2) do |config|
  config.ssh.forward_agent = true  # so you can use git ssh://...

  # add a bridged network interface. (try to detect name, then guess MacOS names, too)
  interface_guesses = [get_good_ifc(), 'en0: Ethernet', 'en1: Wi-Fi (AirPort)']
  config.vm.network "public_network", bridge: interface_guesses
  if ARGV[0] == "up"
    puts "Trying bridge network using interfaces: #{interface_guesses}"
  end
  config.vm.provision "shell", inline: "ip address", run: "always"  # make user feel
↪good

  # . . . . . . . . . . . . Define machine QUAIL1 . . . . . . . . . . . . . .
  config.vm.define "quail1", primary: true do |quail_config|
```

(continues on next page)

(continued from previous page)

```
    quail_config.vm.box = "boxesio/xenial64-standard"  # a public VMware & Virtualbox
↪box
    quail_config.vm.hostname = "quail1." + DOMAIN  # supply a name in our bevy
    quail_config.vm.provider "virtualbox" do |v|
        v.memory = 1024       # limit memory for the virtual box
        v.cpus = 1
        v.linked_clone = true # make a soft copy of the base Vagrant box
        v.customize ["modifyvm", :id, "--natnet1", "192.168.128.0/24"]  # do not use
↪10.x network for NAT
    end
  end
end
```

```
# file /etc/salt/cloud.profiles.d/my_vagrant_profiles.conf on bevymaster
q1:
  host: my_laptop  # the Salt id of your virtual machine host
  machine: quail1   # a machine name in the Vagrantfile (if not primary)
  vagrant_runas: my_username  # owner of Vagrant box files on "my_laptop"
  cwd: '/home/my_username' # the path (on "my_laptop") of the Vagrantfile
  provider: my_vagrant_provider  # name of entry in provider.conf file
  target_network: '10.0.0.0/8'  # VM external address will be somewhere here
```

```
# file /etc/salt/cloud.providers.d/vagrant_provider.conf on bevymaster
my_vagrant_provider:
  driver: vagrant
  minion:
    master: 10.124.30.7  # the hard address of the master
```

Create and use your new Salt minion

- Typing on the Salt master computer `bevymaster`, tell it to create a new minion named `v1` using profile `q1`…

```
sudo salt-cloud -p q1 v1
sudo salt v1 network.ip_addrs
  [ you get a list of IP addresses, including the bridged one ]
```

- logged in to your laptop (or some other computer known to GitHub)…

  [NOTE:] if you are using MacOS, you need to type `ssh-add -K` after each boot, unless you use one of the methods in this gist.

```
ssh -A vagrant@< the bridged network address >
  # [ or, if you are at /home/my_username/ on my_laptop ]
vagrant ssh quail1
```

- then typing on your new node ”v1” (a.k.a. quail1.bevy1.test)…

```
password: vagrant
  # [ stuff types out ... ]

ls -al /vagrant
  # [ should be shared /home/my_username from my_laptop ]

# you can access other network facilities using the ssh authorization
```

(continues on next page)

(continued from previous page)

```
# as recorded in your ~.ssh/ directory on my_laptop ...

sudo apt update
sudo apt install git
git clone ssh://git@github.com/yourID/your_project
# etc...
```

14.7.26 Getting Started with VEXXHOST

VEXXHOST is a cloud computing host which provides Canadian cloud computing services which are based in Monteral and use the libcloud OpenStack driver. VEXXHOST currently runs the Havana release of OpenStack. When provisioning new instances, they automatically get a public IP and private IP address. Therefore, you do not need to assign a floating IP to access your instance after it's booted.

Cloud Provider Configuration

To use the *openstack* driver for the VEXXHOST public cloud, you will need to set up the cloud provider configuration file as in the example below:

`/etc/salt/cloud.providers.d/vexxhost.conf`: In order to use the VEXXHOST public cloud, you will need to setup a cloud provider configuration file as in the example below which uses the OpenStack driver.

```
my-vexxhost-config:
  # Set the location of the salt-master
  #
  minion:
    master: saltmaster.example.com

  # Configure VEXXHOST using the OpenStack plugin
  #
  identity_url: http://auth.api.thenebulacloud.com:5000/v2.0/tokens
  compute_name: nova

  # Set the compute region:
  #
  compute_region: na-yul-nhs1

  # Configure VEXXHOST authentication credentials
  #
  user: your-tenant-id
  password: your-api-key
  tenant: your-tenant-name

  # keys to allow connection to the instance launched
  #
  ssh_key_name: yourkey
  ssh_key_file: /path/to/key/yourkey.priv

  driver: openstack
```

Note: Changed in version 2015.8.0.

The `provider` parameter in cloud provider definitions was renamed to `driver`. This change was made to avoid confusion with the `provider` parameter that is used in cloud profile definitions. Cloud provider definitions now use `driver` to refer to the Salt cloud module that provides the underlying functionality to connect to a cloud host, while cloud profiles continue to use `provider` to refer to provider configurations that you define.

Authentication

All of the authentication fields that you need can be found by logging into your VEXXHOST customer center. Once you've logged in, you will need to click on "CloudConsole" and then click on "API Credentials".

Cloud Profile Configuration

In order to get the correct image UUID and the instance type to use in the cloud profile, you can run the following command respectively:

```
# salt-cloud --list-images=vexxhost-config
# salt-cloud --list-sizes=vexxhost-config
```

Once you have that, you can go ahead and create a new cloud profile. This profile will build an Ubuntu 12.04 LTS *nb.2G* instance.

`/etc/salt/cloud.profiles.d/vh_ubuntu1204_2G.conf`:

```
vh_ubuntu1204_2G:
  provider: my-vexxhost-config
  image: 4051139f-750d-4d72-8ef0-074f2ccc7e5a
  size: nb.2G
```

Provision an instance

To create an instance based on the sample profile that we created above, you can run the following *salt-cloud* command.

```
# salt-cloud -p vh_ubuntu1204_2G vh_instance1
```

Typically, instances are provisioned in under 30 seconds on the VEXXHOST public cloud. After the instance provisions, it will be set up a minion and then return all the instance information once it's complete.

Once the instance has been setup, you can test connectivity to it by running the following command:

```
# salt vh_instance1 test.version
```

You can now continue to provision new instances and they will all automatically be set up as minions of the master you've defined in the configuration file.

14.7.27 Getting Started With Virtualbox

The Virtualbox cloud module allows you to manage a **local** Virtualbox hypervisor. Remote hypervisors may come later on.

Dependencies

The virtualbox module for Salt Cloud requires the Virtualbox SDK which is contained in a virtualbox installation from

https://www.virtualbox.org/wiki/Downloads

Configuration

The Virtualbox cloud module just needs to use the virtualbox driver for now. Virtualbox will be run as the running user.

`/etc/salt/cloud.providers` or `/etc/salt/cloud.providers.d/virtualbox.conf`:

```
virtualbox-config:
   driver: virtualbox
```

Profiles

Set up an initial profile at `/etc/salt/cloud.profiles` or `/etc/salt/cloud.profiles.d/virtualbox.conf`:

```
virtualbox-test:
    provider: virtualbox-config
    clonefrom: VM_to_clone_from
    # Optional
    power_on: True
    deploy: True
    ssh_username: a_username
    password: a_password
    sudo: a_username
    sudo_password: a_password
    # Example minion config
    minion:
        master: localhost
    make_master: True
```

`clonefrom` **Mandatory** Enter the name of the VM/template to clone from.

So far only machines can only be cloned and automatically provisioned by Salt Cloud.

Provisioning

In order to provision when creating a new machine `power_on` and `deploy` have to be `True`.

Furthermore to connect to the VM `ssh_username` and `password` will have to be set.

`sudo` and `sudo_password` are the credentials for getting root access in order to deploy salt

Actions

`start` Attempt to boot a VM by name. VMs should have unique names in order to boot the correct one.

`stop` Attempt to stop a VM. This is akin to a force shutdown or 5 second press.

Functions

`show_image` Show all available information about a VM given by the *image* parameter

```
$ salt-cloud -f show_image virtualbox image=my_vm_name
```

14.7.28 Getting Started With VMware

New in version 2015.5.4.

Author: Nitin Madhok <nmadhok@clemson.edu>

The VMware cloud module allows you to manage VMware ESX, ESXi, and vCenter.

Dependencies

The vmware module for Salt Cloud requires the `pyVmomi` package, which is available at PyPI:

https://pypi.python.org/pypi/pyvmomi

This package can be installed using *pip* or *easy_install*:

```
pip install pyvmomi
easy_install pyvmomi
```

Note: Version 6.0 of pyVmomi has some problems with SSL error handling on certain versions of Python. If using version 6.0 of pyVmomi, the machine that you are running the proxy minion process from must have either Python 2.7.9 or newer This is due to an upstream dependency in pyVmomi 6.0 that is not supported in Python version 2.6 to 2.7.8. If the version of Python running the salt-cloud command is not in the supported range, you will need to install an earlier version of pyVmomi. See Issue #29537 for more information.

Note: pyVmomi doesn't expose the ability to specify the locale when connecting to VMware. This causes parsing issues when connecting to an instance of VMware running under a non-English locale. Until this feature is added upstream Issue #38402 contains a workaround.

Configuration

The VMware cloud module needs the vCenter or ESX/ESXi URL, username and password to be set up in the cloud configuration at `/etc/salt/cloud.providers` or `/etc/salt/cloud.providers.d/vmware.conf`:

```
my-vmware-config:
  driver: vmware
  user: 'DOMAIN\user'
  password: 'verybadpass'
  url: '10.20.30.40'

vcenter01:
  driver: vmware
  user: 'DOMAIN\user'
  password: 'verybadpass'
  url: 'vcenter01.domain.com'
```

(continues on next page)

(continued from previous page)

```
  protocol: 'https'
  port: 443

vcenter02:
  driver: vmware
  user: 'DOMAIN\user'
  password: 'verybadpass'
  url: 'vcenter02.domain.com'
  protocol: 'http'
  port: 80

esx01:
  driver: vmware
  user: 'admin'
  password: 'verybadpass'
  url: 'esx01.domain.com'
```

Note: Optionally, `protocol` and `port` can be specified if the vCenter server is not using the defaults. Default is `protocol: https` and `port: 443`.

Note: Changed in version 2015.8.0.

The `provider` parameter in cloud provider configuration was renamed to `driver`. This change was made to avoid confusion with the `provider` parameter that is used in cloud profile configuration. Cloud provider configuration now uses `driver` to refer to the salt-cloud driver that provides the underlying functionality to connect to a cloud provider, while cloud profile configuration continues to use `provider` to refer to the cloud provider configuration that you define.

Profiles

Set up an initial profile at `/etc/salt/cloud.profiles` or `/etc/salt/cloud.profiles.d/vmware.conf`:

```
vmware-centos6.5:
  provider: vcenter01
  clonefrom: test-vm

  ## Optional arguments
  num_cpus: 4
  memory: 8GB
  devices:
    cd:
      CD/DVD drive 1:
        device_type: datastore_iso_file
        iso_path: "[nap004-1] vmimages/tools-isoimages/linux.iso"
      CD/DVD drive 2:
        device_type: client_device
        mode: atapi
        controller: IDE 2
      CD/DVD drive 3:
        device_type: client_device
```

(continues on next page)

(continued from previous page)

```
        mode: passthrough
        controller: IDE 3
    disk:
      Hard disk 1:
        size: 30
      Hard disk 2:
        size: 20
        controller: SCSI controller 2
      Hard disk 3:
        size: 5
        controller: SCSI controller 3
        datastore: smalldiskdatastore
    network:
      Network adapter 1:
        name: 10.20.30-400-Test
        switch_type: standard
        ip: 10.20.30.123
        gateway: [10.20.30.110]
        subnet_mask: 255.255.255.128
        domain: example.com
      Network adapter 2:
        name: 10.30.40-500-Dev-DHCP
        adapter_type: e1000
        switch_type: distributed
        mac: '00:16:3e:e8:19:0f'
      Network adapter 3:
        name: 10.40.50-600-Prod
        adapter_type: vmxnet3
        switch_type: distributed
        ip: 10.40.50.123
        gateway: [10.40.50.110]
        subnet_mask: 255.255.255.128
        domain: example.com
    scsi:
      SCSI controller 1:
        type: lsilogic
      SCSI controller 2:
        type: lsilogic_sas
        bus_sharing: virtual
      SCSI controller 3:
        type: paravirtual
        bus_sharing: physical
    ide:
      IDE 2
      IDE 3

  domain: example.com
  dns_servers:
    - 123.127.255.240
    - 123.127.255.241
    - 123.127.255.242

  resourcepool: Resources
  cluster: Prod

  datastore: HUGE-DATASTORE-Cluster
```

(continues on next page)

(continued from previous page)

```
  folder: Development
  datacenter: DC1
  host: c4212n-002.domain.com
  template: False
  power_on: True
  extra_config:
    mem.hotadd: 'yes'
    guestinfo.foo: bar
    guestinfo.domain: foobar.com
    guestinfo.customVariable: customValue
  annotation: Created by Salt-Cloud

  deploy: True
  customization: True
  private_key: /root/.ssh/mykey.pem
  ssh_username: cloud-user
  password: veryVeryBadPassword
  minion:
    master: 123.127.193.105

  file_map:
    /path/to/local/custom/script: /path/to/remote/script
    /path/to/local/file: /path/to/remote/file
    /srv/salt/yum/epel.repo: /etc/yum.repos.d/epel.repo

  hardware_version: 10
  image: centos64Guest

  #For Windows VM
  win_username: Administrator
  win_password: administrator
  win_organization_name: ABC-Corp
  plain_text: True
  win_installer: /root/Salt-Minion-2015.8.4-AMD64-Setup.exe
  win_user_fullname: Windows User
```

`provider` Enter the name that was specified when the cloud provider config was created.

`clonefrom` Enter the name of the VM/template to clone from. If not specified, the VM will be created without cloning.

`num_cpus` Enter the number of vCPUS that you want the VM/template to have. If not specified, the current VM/template's vCPU count is used.

`cores_per_socket` Enter the number of cores per vCPU that you want the VM/template to have. If not specified, this will default to 1.

> **Note:** Cores per socket should be less than or equal to the total number of vCPUs assigned to the VM/template.

New in version 2016.11.0.

`memory` Enter the memory size (in MB or GB) that you want the VM/template to have. If not specified, the current VM/template's memory size is used. Example `memory: 8GB` or `memory: 8192MB`.

`devices` Enter the device specifications here. Currently, the following devices can be created or reconfigured:

cd Enter the CD/DVD drive specification here. If the CD/DVD drive doesn't exist, it will be created with the specified configuration. If the CD/DVD drive already exists, it will be reconfigured with the specifications. The following options can be specified per CD/DVD drive:

device_type Specify how the CD/DVD drive should be used. Currently supported types are `client_device` and `datastore_iso_file`. Default is `device_type: client_device`

iso_path Enter the path to the iso file present on the datastore only if `device_type: datastore_iso_file`. The syntax to specify this is `iso_path: "[datastoreName] vmimages/tools-isoimages/linux.iso"`. This field is ignored if `device_type: client_device`

mode Enter the mode of connection only if `device_type: client_device`. Currently supported modes are `passthrough` and `atapi`. This field is ignored if `device_type: datastore_iso_file`. Default is `mode: passthrough`

controller Specify the IDE controller label to which this drive should be attached. This should be specified only when creating both the specified IDE controller as well as the CD/DVD drive at the same time.

disk Enter the disk specification here. If the hard disk doesn't exist, it will be created with the provided size. If the hard disk already exists, it will be expanded if the provided size is greater than the current size of the disk.

size Enter the size of disk in GB

thin_provision Specifies whether the disk should be thin provisioned or not. Default is `thin_provision: False`. .. versionadded:: 2016.3.0

eagerly_scrub Specifies whether the disk should be rewrite with zeros during thick provisioning or not. Default is `eagerly_scrub: False`. .. versionadded:: 2018.3.0

controller Specify the SCSI controller label to which this disk should be attached. This should be specified only when creating both the specified SCSI controller as well as the hard disk at the same time.

datastore The name of a valid datastore should you wish the new disk to be in a datastore other than the default for the VM.

network Enter the network adapter specification here. If the network adapter doesn't exist, a new network adapter will be created with the specified network name, type and other configuration. If the network adapter already exists, it will be reconfigured with the specifications. The following additional options can be specified per network adapter (See example above):

name Enter the network name you want the network adapter to be mapped to.

adapter_type Enter the network adapter type you want to create. Currently supported types are `vmxnet`, `vmxnet2`, `vmxnet3`, `e1000` and `e1000e`. If no type is specified, by default `vmxnet3` will be used.

switch_type Enter the type of switch to use. This decides whether to use a standard switch network or a distributed virtual portgroup. Currently supported types are `standard` for standard portgroups and `distributed` for distributed virtual portgroups.

ip Enter the static IP you want the network adapter to be mapped to. If the network specified is DHCP enabled, you do not have to specify this.

gateway Enter the gateway for the network as a list. If the network specified is DHCP enabled, you do not have to specify this.

subnet_mask Enter the subnet mask for the network. If the network specified is DHCP enabled, you do not have to specify this.

domain Enter the domain to be used with the network adapter. If the network specified is DHCP enabled, you do not have to specify this.

mac Enter the MAC for this network adapter. If not specified an address will be selected automatically.

scsi Enter the SCSI controller specification here. If the SCSI controller doesn't exist, a new SCSI controller will be created of the specified type. If the SCSI controller already exists, it will be reconfigured with the specifications. The following additional options can be specified per SCSI controller:

type Enter the SCSI controller type you want to create. Currently supported types are `lsilogic`, `lsilogic_sas` and `paravirtual`. Type must be specified when creating a new SCSI controller.

bus_sharing Specify this if sharing of virtual disks between virtual machines is desired. The following can be specified:

virtual Virtual disks can be shared between virtual machines on the same server.

physical Virtual disks can be shared between virtual machines on any server.

no Virtual disks cannot be shared between virtual machines.

ide Enter the IDE controller specification here. If the IDE controller doesn't exist, a new IDE controller will be created. If the IDE controller already exists, no further changes to it will me made.

`domain` Enter the global domain name to be used for DNS. If not specified and if the VM name is a FQDN, `domain` is set to the domain from the VM name. Default is `local`.

`dns_servers` Enter the list of DNS servers to use in order of priority.

`resourcepool` Enter the name of the resourcepool to which the new virtual machine should be attached. This determines what compute resources will be available to the clone.

Note:

- For a clone operation from a virtual machine, it will use the same resourcepool as the original virtual machine unless specified.
- For a clone operation from a template to a virtual machine, specifying either this or cluster is required. If both are specified, the resourcepool value will be used.
- For a clone operation to a template, this argument is ignored.

`cluster` Enter the name of the cluster whose resource pool the new virtual machine should be attached to.

Note:

- For a clone operation from a virtual machine, it will use the same cluster's resourcepool as the original virtual machine unless specified.
- For a clone operation from a template to a virtual machine, specifying either this or resourcepool is required. If both are specified, the resourcepool value will be used.
- For a clone operation to a template, this argument is ignored.

`datastore` Enter the name of the datastore or the datastore cluster where the virtual machine should be located on physical storage. If not specified, the current datastore is used.

Note:

- If you specify a datastore cluster name, DRS Storage recommendation is automatically applied.

- If you specify a datastore name, DRS Storage recommendation is disabled.

`folder` Enter the name of the folder that will contain the new virtual machine.

Note:

- For a clone operation from a VM/template, the new VM/template will be added to the same folder that the original VM/template belongs to unless specified.
- If both folder and datacenter are specified, the folder value will be used.

`datacenter` Enter the name of the datacenter that will contain the new virtual machine.

Note:

- For a clone operation from a VM/template, the new VM/template will be added to the same folder that the original VM/template belongs to unless specified.
- If both folder and datacenter are specified, the folder value will be used.

`host` Enter the name of the target host where the virtual machine should be registered.

If not specified:

Note:

- If resource pool is not specified, current host is used.
- If resource pool is specified, and the target pool represents a stand-alone host, the host is used.
- If resource pool is specified, and the target pool represents a DRS-enabled cluster, a host selected by DRS is used.
- If resource pool is specified and the target pool represents a cluster without DRS enabled, an InvalidArgument exception be thrown.

`template` Specifies whether the new virtual machine should be marked as a template or not. Default is `template: False`.

`power_on` Specifies whether the new virtual machine should be powered on or not. If `template: True` is set, this field is ignored. Default is `power_on: True`.

`extra_config` Specifies the additional configuration information for the virtual machine. This describes a set of modifications to the additional options. If the key is already present, it will be reset with the new value provided. Otherwise, a new option is added. Keys with empty values will be removed.

`annotation` User-provided description of the virtual machine. This will store a message in the vSphere interface, under the annotations section in the Summary view of the virtual machine.

`deploy` Specifies if salt should be installed on the newly created VM. Default is `True` so salt will be installed using the bootstrap script. If `template: True` or `power_on: False` is set, this field is ignored and salt will not be installed.

`wait_for_ip_timeout` When `deploy: True`, this timeout determines the maximum time to wait for VMware tools to be installed on the virtual machine. If this timeout is reached, an attempt to determine the client's IP will be made by resolving the VM's name. By lowering this value a salt bootstrap can be fully automated for systems that are not built with VMware tools. Default is `wait_for_ip_timeout: 1200`.

`customization` Specify whether the new virtual machine should be customized or not. If `customization: False` is set, the new virtual machine will not be customized. Default is `customization: True`.

`private_key` Specify the path to the private key to use to be able to ssh to the VM.

`ssh_username` Specify the username to use in order to ssh to the VM. Default is `root`

`password` Specify a password to use in order to ssh to the VM. If `private_key` is specified, you do not need to specify this.

`minion` Specify custom minion configuration you want the salt minion to have. A good example would be to specify the `master` as the IP/DNS name of the master.

`file_map` Specify file/files you want to copy to the VM before the bootstrap script is run and salt is installed. A good example of using this would be if you need to put custom repo files on the server in case your server will be in a private network and cannot reach external networks.

`hardware_version` Specify the virtual hardware version for the vm/template that is supported by the host.

`image` Specify the guest id of the VM. For a full list of supported values see the VMware vSphere documentation:

http://pubs.vmware.com/vsphere-60/topic/com.vmware.wssdk.apiref.doc/vim.vm.GuestOsDescriptor.GuestOsIdentifier.html

> **Note:** For a clone operation, this argument is ignored.

`win_username` Specify windows vm administrator account.

> **Note:** Windows template should have "administrator" account.

`win_password` Specify windows vm administrator account password.

> **Note:** During network configuration (if network specified), it is used to specify new administrator password for the machine.

`win_organization_name`

Specify windows vm user's organization. Default organization name is Organization VMware vSphere documentation:

https://www.vmware.com/support/developer/vc-sdk/visdk25pubs/ReferenceGuide/vim.vm.customization.UserData.html

`win_user_fullname`

Specify windows vm user's fullname. Default fullname is "Windows User" VMware vSphere documentation:

https://www.vmware.com/support/developer/vc-sdk/visdk25pubs/ReferenceGuide/vim.vm.customization.UserData.html

`plain_text` Flag to specify whether or not the password is in plain text, rather than encrypted. VMware vSphere documentation:

https://www.vmware.com/support/developer/vc-sdk/visdk25pubs/ReferenceGuide/vim.vm.customization.Password.html

`win_installer` Specify windows minion client installer path

`win_run_once` Specify a list of commands to run on first login to a windows minion

https://www.vmware.com/support/developer/vc-sdk/visdk25pubs/ReferenceGuide/vim.vm.customization.GuiRunOnce.html

Cloning a VM

Cloning VMs/templates is the easiest and the preferred way to work with VMs using the VMware driver.

Note: Cloning operations are unsupported on standalone ESXi hosts, a vCenter server will be required.

Example of a minimal profile:

```
my-minimal-clone:
  provider: vcenter01
  clonefrom: 'test-vm'
```

When cloning a VM, all the profile configuration parameters are optional and the configuration gets inherited from the clone.

Example to add/resize a disk:

```
my-disk-example:
  provider: vcenter01
  clonefrom: 'test-vm'

  devices:
    disk:
      Hard disk 1:
        size: 30
```

Depending on the configuration of the VM that is getting cloned, the disk in the resulting clone will differ.

Note:

- If the VM has no disk named 'Hard disk 1' an empty disk with the specified size will be added to the clone.
- If the VM has a disk named 'Hard disk 1' and the size specified is larger than the original disk, an empty disk with the specified size will be added to the clone.
- If the VM has a disk named 'Hard disk 1' and the size specified is smaller than the original disk, an empty disk with the original size will be added to the clone.

Example to reconfigure the memory and number of vCPUs:

```
my-disk-example:
  provider: vcenter01
  clonefrom: 'test-vm'

  memory: 16GB
  num_cpus: 8
```

Cloning a Template

Cloning a template works similar to cloning a VM except for the fact that a resource pool or cluster must be specified additionally in the profile.

Example of a minimal profile:

```
my-template-clone:
 provider: vcenter01
 clonefrom: 'test-template'
 cluster: 'Prod'
```

Cloning from a Snapshot

New in version 2016.3.5.

Cloning from a snapshot requires that one of the supported options be set in the cloud profile.

Supported options are `createNewChildDiskBacking`, `moveChildMostDiskBacking`, `moveAllDiskBackingsAndAllowSharing` and `moveAllDiskBackingsAndDisallowSharing`.

Example of a minimal profile:

```
my-template-clone:
  provider: vcenter01
  clonefrom: 'salt_vm'
  snapshot:
    disk_move_type: createNewChildDiskBacking
    # these types are also supported
    # disk_move_type: moveChildMostDiskBacking
    # disk_move_type: moveAllDiskBackingsAndAllowSharing
    # disk_move_type: moveAllDiskBackingsAndDisallowSharing
```

Creating a VM

New in version 2016.3.0.

Creating a VM from scratch means that more configuration has to be specified in the profile because there is no place to inherit configuration from.

Note: Unlike most cloud drivers that use prepared images, creating VMs using VMware cloud driver needs an installation method that requires no human interaction. For Example: preseeded ISO, kickstart URL or network PXE boot.

Example of a minimal profile:

```
my-minimal-profile:
  provider: esx01
  datastore: esx01-datastore
  resourcepool: Resources
  folder: vm
```

Note: The example above contains the minimum required configuration needed to create a VM from scratch. The resulting VM will only have 1 VCPU, 32MB of RAM and will not have any storage or networking.

Example of a complete profile:

```yaml
my-complete-example:
  provider: esx01
  datastore: esx01-datastore
  resourcepool: Resources
  folder: vm

  num_cpus: 2
  memory: 8GB

  image: debian7_64Guest

  devices:
    scsi:
      SCSI controller 0:
        type: lsilogic_sas
    ide:
      IDE 0: {}
      IDE 1: {}
    disk:
      Hard disk 0:
        controller: 'SCSI controller 0'
        size: 20
        mode: 'independent_nonpersistent'
    cd:
      CD/DVD drive 0:
        controller: 'IDE 0'
        device_type: datastore_iso_file
        iso_path: '[esx01-datastore] debian-8-with-preseed.iso'
    network:
      Network adapter 0:
        name: 'VM Network'
        swith_type: standard
```

Note: Depending on VMware ESX/ESXi version, an exact match for `image` might not be available. In such cases, the closest match to another `image` should be used. In the example above, a Debian 8 VM is created using the image `debian7_64Guest` which is for a Debian 7 guest.

Specifying disk backing mode

New in version 2016.3.5.

Disk backing mode can now be specified when cloning a VM. This option can be set in the cloud profile as shown in example below:

```yaml
my-vm:
  provider: esx01
  datastore: esx01-datastore
  resourcepool: Resources
```

(continues on next page)

(continued from previous page)

```
  folder: vm


  devices:
    disk:
      Hard disk 1:
        mode: 'independent_nonpersistent'
        size: 42
      Hard disk 2:
        mode: 'independent_nonpersistent'
```

14.7.29 Getting Started With Xen

The Xen cloud driver works with Citrix XenServer.

It can be used with a single XenServer or a XenServer resource pool.

Setup Dependencies

This driver requires a copy of the freely available `XenAPI.py` Python module.

Information about the Xen API Python module in the XenServer SDK can be found at https://xenserver.org/partners/developing-products-for-xenserver.html

Place a copy of this module on your system. For example, it can be placed in the *site packages* location on your system.

The location of *site packages* can be determined by running:

```
python -m site --user-site
```

Provider Configuration

Xen requires login credentials to a XenServer.

Set up the provider cloud configuration file at `/etc/salt/cloud.providers` or `/etc/salt/cloud.providers.d/*.conf`.

```
# /etc/salt/cloud.providers.d/myxen.conf
myxen:
  driver: xen
  url: https://10.0.0.120
  user: root
  password: p@ssw0rd
```

url: The `url` option supports both `http` and `https` uri prefixes.

user: A valid user id to login to the XenServer host.

password: The associated password for the user.

Note: Changed in version 2015.8.0.

The `provider` parameter in cloud provider definitions was renamed to `driver`. This change was made to avoid confusion with the `provider` parameter that is used in cloud profile definitions. Cloud provider definitions now use `driver` to refer to the Salt cloud module that provides the underlying functionality to connect to a cloud host, while cloud profiles continue to use `provider` to refer to provider configurations that you define.

Profile Configuration

Xen profiles require a `provider` and `image`.

provider: This will be the name of your defined provider.

image: The name of the VM template used to clone or copy.

clone: The default behavior is to clone a template or VM. This is very fast, but requires the source template or VM to be in the same storage repository of the new target system. If the source and target are in different storage repositories then you must copy the source and not clone it by setting `clone: False`.

deploy: The provisioning process will attempt to install the Salt minion service on the new target system by default. This will require login credentials for Salt cloud to login via ssh to it. The `user` and `password` options are required. If `deploy` is set to `False` then these options are not needed.

resource_pool: The name of the resource pool used for this profile.

storage_repo: The name of the storage repository for the target system.

ipv4_cidr: If template is Windows, and running guest tools then a static ip address can be set.

ipv4_gw: If template is Windows, and running guest tools then a gateway can be set.

Set up an initial profile at `/etc/salt/cloud.profiles` or in the `/etc/salt/cloud.profiles.d/` directory:

```yaml
# file: /etc/salt/cloud.profiles.d/xenprofiles.conf
sles:
  provider: myxen
  deploy: False
  image: sles12sp2-template

suse:
  user: root
  password: p@ssw0rd
  provider: myxen
  image: opensuseleap42_2-template
  storage_repo: 'Local storage'
  clone: False
  minion:
    master: 10.0.0.20

w2k12:
  provider: myxen
  image: w2k12svr-template
  clone: True
  userdata_file: /srv/salt/win/files/windows-firewall.ps1
  win_installer: /srv/salt/win/files/Salt-Minion-2016.11.3-AMD64-Setup.exe
  win_username: Administrator
  win_password: p@ssw0rd
  use_winrm: False
  ipv4_cidr: 10.0.0.215/24
  ipv4_gw: 10.0.0.1
  minion:
    master: 10.0.0.21
```

The first example will create a clone of the sles12sp2-template in the same storage repository without deploying the Salt minion.

The second example will make a copy of the image and deploy a new suse VM with the Salt minion installed.

The third example will create a clone of the Windows 2012 template and deploy the Salt minion.

The profile can be used with a salt command:

```
salt-cloud -p suse  xenvm02
```

This will create an salt minion instance named `xenvm02` in Xen. If the command was executed on the salt-master, its Salt key will automatically be signed on the master.

Once the instance has been created with a salt-minion installed, connectivity to it can be verified with Salt:

```
salt xenvm02 test.version
```

Listing Sizes

Sizes can be obtained using the `--list-sizes` option for the `salt-cloud` command:

```
# salt-cloud --list-sizes myxen
```

Note: Since size information is build in a template this command is not implemented.

Listing Images

Images can be obtained using the `--list-images` option for the `salt-cloud` command:

```
# salt-cloud --list-images myxen
```

This command will return a list of templates with details.

Listing Locations

Locations can be obtained using the `--list-locations` option for the `salt-cloud` command:

```
# salt-cloud --list-locations myxen
```

Returns a list of resource pools.

14.8 Miscellaneous Options

14.8.1 Miscellaneous Salt Cloud Options

This page describes various miscellaneous options available in Salt Cloud

Deploy Script Arguments

Custom deploy scripts are unlikely to need custom arguments to be passed to them, but salt-bootstrap has been extended quite a bit, and this may be necessary. script_args can be specified in either the profile or the map file, to pass arguments to the deploy script:

```
ec2-amazon:
  provider: my-ec2-config
  image: ami-1624987f
  size: t1.micro
  ssh_username: ec2-user
  script: bootstrap-salt
  script_args: -c /tmp/
```

This has also been tested to work with pipes, if needed:

```
script_args: '| head'
```

Selecting the File Transport

By default, Salt Cloud uses SFTP to transfer files to Linux hosts. However, if SFTP is not available, or specific SCP functionality is needed, Salt Cloud can be configured to use SCP instead.

```
file_transport: sftp
file_transport: scp
```

Sync After Install

Salt allows users to create custom modules, grains, and states which can be synchronised to minions to extend Salt with further functionality.

This option will inform Salt Cloud to synchronise your custom modules, grains, states or all these to the minion just after it has been created. For this to happen, the following line needs to be added to the main cloud configuration file:

```
sync_after_install: all
```

The available options for this setting are:

```
modules
grains
states
all
```

Setting Up New Salt Masters

It has become increasingly common for users to set up multi-hierarchal infrastructures using Salt Cloud. This sometimes involves setting up an instance to be a master in addition to a minion. With that in mind, you can now lay down master configuration on a machine by specifying master options in the profile or map file.

```
make_master: True
```

This will cause Salt Cloud to generate master keys for the instance, and tell salt-bootstrap to install the salt-master package, in addition to the salt-minion package.

The default master configuration is usually appropriate for most users, and will not be changed unless specific master configuration has been added to the profile or map:

```
master:
  user: root
  interface: 0.0.0.0
```

Setting Up a Salt Syndic with Salt Cloud

In addition to *setting up new Salt Masters*, *syndics* can also be provisioned using Salt Cloud. In order to set up a Salt Syndic via Salt Cloud, a Salt Master needs to be installed on the new machine and a master configuration file needs to be set up using the `make_master` setting. This setting can be defined either in a profile config file or in a map file:

```
make_master: True
```

To install the Salt Syndic, the only other specification that needs to be configured is the `syndic_master` key to specify the location of the master that the syndic will be reporting to. This modification needs to be placed in the `master` setting, which can be configured either in the profile, provider, or `/etc/salt/cloud` config file:

```
master:
  syndic_master: 123.456.789  # may be either an IP address or a hostname
```

Many other Salt Syndic configuration settings and specifications can be passed through to the new syndic machine via the `master` configuration setting. See the *Salt Syndic* documentation for more information.

SSH Port

By default ssh port is set to port 22. If you want to use a custom port in provider, profile, or map blocks use ssh_port option.

New in version 2015.5.0.

```
ssh_port: 2222
```

Delete SSH Keys

When Salt Cloud deploys an instance, the SSH pub key for the instance is added to the known_hosts file for the user that ran the salt-cloud command. When an instance is deployed, a cloud host generally recycles the IP address for the instance. When Salt Cloud attempts to deploy an instance using a recycled IP address that has previously been accessed from the same machine, the old key in the known_hosts file will cause a conflict.

In order to mitigate this issue, Salt Cloud can be configured to remove old keys from the known_hosts file when destroying the node. In order to do this, the following line needs to be added to the main cloud configuration file:

```
delete_sshkeys: True
```

Keeping /tmp/ Files

When Salt Cloud deploys an instance, it uploads temporary files to /tmp/ for salt-bootstrap to put in place. After the script has run, they are deleted. To keep these files around (mostly for debugging purposes), the –keep-tmp option can be added:

```
salt-cloud -p myprofile mymachine --keep-tmp
```

For those wondering why /tmp/ was used instead of /root/, this had to be done for images which require the use of sudo, and therefore do not allow remote root logins, even for file transfers (which makes /root/ unavailable).

Hide Output From Minion Install

By default Salt Cloud will stream the output from the minion deploy script directly to STDOUT. Although this can been very useful, in certain cases you may wish to switch this off. The following config option is there to enable or disable this output:

```
display_ssh_output: False
```

Connection Timeout

There are several stages when deploying Salt where Salt Cloud needs to wait for something to happen. The VM getting it's IP address, the VM's SSH port is available, etc.

If you find that the Salt Cloud defaults are not enough and your deployment fails because Salt Cloud did not wait log enough, there are some settings you can tweak.

Note

All settings should be provided in lowercase All values should be provided in seconds

---

You can tweak these settings globally, per cloud provider, or event per profile definition.

wait_for_ip_timeout

The amount of time Salt Cloud should wait for a VM to start and get an IP back from the cloud host. Default: varies by cloud provider (between 5 and 25 minutes)

wait_for_ip_interval

The amount of time Salt Cloud should sleep while querying for the VM's IP. Default: varies by cloud provider (between .5 and 10 seconds)

ssh_connect_timeout

The amount of time Salt Cloud should wait for a successful SSH connection to the VM. Default: varies by cloud provider (between 5 and 15 minutes)

wait_for_passwd_timeout

The amount of time until an ssh connection can be established via password or ssh key. Default: varies by cloud provider (mostly 15 seconds)

wait_for_passwd_maxtries

The number of attempts to connect to the VM until we abandon. Default: 15 attempts

wait_for_fun_timeout

Some cloud drivers check for an available IP or a successful SSH connection using a function, namely, SoftLayer, and SoftLayer-HW. So, the amount of time Salt Cloud should retry such functions before failing. Default: 15 minutes.

wait_for_spot_timeout

The amount of time Salt Cloud should wait before an EC2 Spot instance is available. This setting is only available for the EC2 cloud driver. Default: 10 minutes

Salt Cloud Cache

Salt Cloud can maintain a cache of node data, for supported providers. The following options manage this functionality.

update_cachedir

On supported cloud providers, whether or not to maintain a cache of nodes returned from a –full-query. The data will be stored in `msgpack` format under `<SALT_CACHEDIR>/cloud/active/<DRIVER>/<PROVIDER>/<NODE_NAME>.p`. This setting can be True or False.

diff_cache_events

When the cloud cachedir is being managed, if differences are encountered between the data that is returned live from the cloud host and the data in the cache, fire events which describe the changes. This setting can be True or False.

Some of these events will contain data which describe a node. Because some of the fields returned may contain sensitive data, the `cache_event_strip_fields` configuration option exists to strip those fields from the event return.

```yaml
cache_event_strip_fields:
  - password
  - priv_key
```

The following are events that can be fired based on this data.

salt/cloud/minionid/cache_node_new

A new node was found on the cloud host which was not listed in the cloud cachedir. A dict describing the new node will be contained in the event.

salt/cloud/minionid/cache_node_missing

A node that was previously listed in the cloud cachedir is no longer available on the cloud host.

salt/cloud/minionid/cache_node_diff

One or more pieces of data in the cloud cachedir has changed on the cloud host. A dict containing both the old and the new data will be contained in the event.

SSH Known Hosts

Normally when bootstrapping a VM, salt-cloud will ignore the SSH host key. This is because it does not know what the host key is before starting (because it doesn't exist yet). If strict host key checking is turned on without the key in the `known_hosts` file, then the host will never be available, and cannot be bootstrapped.

If a provider is able to determine the host key before trying to bootstrap it, that provider's driver can add it to the `known_hosts` file, and then turn on strict host key checking. This can be set up in the main cloud configuration file (normally `/etc/salt/cloud`) or in the provider-specific configuration file:

```
known_hosts_file: /path/to/.ssh/known_hosts
```

If this is not set, it will default to `/dev/null`, and strict host key checking will be turned off.

It is highly recommended that this option is *not* set, unless the user has verified that the provider supports this functionality, and that the image being used is capable of providing the necessary information. At this time, only the EC2 driver supports this functionality.

SSH Agent

New in version 2015.5.0.

If the ssh key is not stored on the server salt-cloud is being run on, set ssh_agent, and salt-cloud will use the forwarded ssh-agent to authenticate.

```
ssh_agent: True
```

File Map Upload

New in version 2014.7.0.

The `file_map` option allows an arbitrary group of files to be uploaded to the target system before running the deploy script. This functionality requires a provider uses salt.utils.cloud.bootstrap(), which is currently limited to the ec2, gce, openstack and nova drivers.

The `file_map` can be configured globally in `/etc/salt/cloud`, or in any cloud provider or profile file. For example, to upload an extra package or a custom deploy script, a cloud profile using `file_map` might look like:

```
ubuntu14:
  provider: ec2-config
  image: ami-98aa1cf0
  size: t1.micro
  ssh_username: root
  securitygroup: default
  file_map:
    /local/path/to/custom/script: /remote/path/to/use/custom/script
    /local/path/to/package: /remote/path/to/store/package
```

Running Pre-Flight Commands

New in version 2018.3.0.

To execute specified preflight shell commands on a VM before the deploy script is run, use the `preflight_cmds` option. These must be defined as a list in a cloud configuration file. For example:

```
my-cloud-profile:
  provider: linode-config
  image: Ubuntu 16.04 LTS
  size: Linode 2048
  preflight_cmds:
    - whoami
    - echo 'hello world!'
```

These commands will run in sequence **before** the bootstrap script is executed.

Force Minion Config

New in version 2018.3.0.

The `force_minion_config` option requests the bootstrap process to overwrite an existing minion configuration file and public/private key files. Default: False

This might be important for drivers (such as `saltify`) which are expected to take over a connection from a former salt master.

```
my_saltify_provider:
  driver: saltify
  force_minion_config: true
```

14.9 Troubleshooting Steps

14.9.1 Troubleshooting Salt Cloud

This page describes various steps for troubleshooting problems that may arise while using Salt Cloud.

Virtual Machines Are Created, But Do Not Respond

Are TCP ports 4505 and 4506 open on the master? This is easy to overlook on new masters. Information on how to open firewall ports on various platforms can be found *here*.

Generic Troubleshooting Steps

This section describes a set of instructions that are useful to a large number of situations, and are likely to solve most issues that arise.

Debug Mode

Frequently, running Salt Cloud in debug mode will reveal information about a deployment which would otherwise not be obvious:

```
salt-cloud -p myprofile myinstance -l debug
```

Keep in mind that a number of messages will appear that look at first like errors, but are in fact intended to give developers factual information to assist in debugging. A number of messages that appear will be for cloud providers that you do not have configured; in these cases, the message usually is intended to confirm that they are not configured.

Salt Bootstrap

By default, Salt Cloud uses the Salt Bootstrap script to provision instances:

This script is packaged with Salt Cloud, but may be updated without updating the Salt package:

```
salt-cloud -u
```

The Bootstrap Log

If the default deploy script was used, there should be a file in the `/tmp/` directory called `bootstrap-salt.log`. This file contains the full output from the deployment, including any errors that may have occurred.

Keeping Temp Files

Salt Cloud uploads minion-specific files to instances once they are available via SSH, and then executes a deploy script to put them into the correct place and install Salt. The `--keep-tmp` option will instruct Salt Cloud not to remove those files when finished with them, so that the user may inspect them for problems:

```
salt-cloud -p myprofile myinstance --keep-tmp
```

By default, Salt Cloud will create a directory on the target instance called `/tmp/.saltcloud/`. This directory should be owned by the user that is to execute the deploy script, and should have permissions of `0700`.

Most cloud hosts are configured to use `root` as the default initial user for deployment, and as such, this directory and all files in it should be owned by the `root` user.

The `/tmp/.saltcloud/` directory should the following files:

- A `deploy.sh` script. This script should have permissions of `0755`.
- A `.pem` and `.pub` key named after the minion. The `.pem` file should have permissions of `0600`. Ensure that the `.pem` and `.pub` files have been properly copied to the `/etc/salt/pki/minion/` directory.
- A file called `minion`. This file should have been copied to the `/etc/salt/` directory.
- Optionally, a file called `grains`. This file, if present, should have been copied to the `/etc/salt/` directory.

Unprivileged Primary Users

Some cloud hosts, most notably EC2, are configured with a different primary user. Some common examples are `ec2-user`, `ubuntu`, `fedora`, and `bitnami`. In these cases, the `/tmp/.saltcloud/` directory and all files in it should be owned by this user.

Some cloud hosts, such as EC2, are configured to not require these users to provide a password when using the `sudo` command. Because it is more secure to require `sudo` users to provide a password, other hosts are configured that way.

If this instance is required to provide a password, it needs to be configured in Salt Cloud. A password for sudo to use may be added to either the provider configuration or the profile configuration:

```
sudo_password: mypassword
```

`/tmp/` is Mounted as `noexec`

It is more secure to mount the `/tmp/` directory with a `noexec` option. This is uncommon on most cloud hosts, but very common in private environments. To see if the `/tmp/` directory is mounted this way, run the following command:

```
mount | grep tmp
```

The if the output of this command includes a line that looks like this, then the `/tmp/` directory is mounted as `noexec`:

```
tmpfs on /tmp type tmpfs (rw,noexec)
```

If this is the case, then the `deploy_command` will need to be changed in order to run the deploy script through the `sh` command, rather than trying to execute it directly. This may be specified in either the provider or the profile config:

```
deploy_command: sh /tmp/.saltcloud/deploy.sh
```

Please note that by default, Salt Cloud will place its files in a directory called `/tmp/.saltcloud/`. This may be also be changed in the provider or profile configuration:

```
tmp_dir: /tmp/.saltcloud/
```

If this directory is changed, then the `deploy_command` need to be changed in order to reflect the `tmp_dir` configuration.

Executing the Deploy Script Manually

If all of the files needed for deployment were successfully uploaded to the correct locations, and contain the correct permissions and ownerships, the deploy script may be executed manually in order to check for other issues:

```
cd /tmp/.saltcloud/
./deploy.sh
```

14.10 Extending Salt Cloud

14.10.1 Writing Cloud Driver Modules

Salt Cloud runs on a module system similar to the main Salt project. The modules inside saltcloud exist in the `salt/cloud/clouds` directory of the salt source.

There are two basic types of cloud modules. If a cloud host is supported by libcloud, then using it is the fastest route to getting a module written. The Apache Libcloud project is located at:

http://libcloud.apache.org/

Not every cloud host is supported by libcloud. Additionally, not every feature in a supported cloud host is necessarily supported by libcloud. In either of these cases, a module can be created which does not rely on libcloud.

All Driver Modules

The following functions are required by all driver modules, whether or not they are based on libcloud.

The __virtual__() Function

This function determines whether or not to make this cloud module available upon execution. Most often, it uses `get_configured_provider()` to determine if the necessary configuration has been set up. It may also check for necessary imports, to decide whether to load the module. In most cases, it will return a `True` or `False` value. If the name of the driver used does not match the filename, then that name should be returned instead of `True`. An example of this may be seen in the Azure module:

https://github.com/saltstack/salt/tree/master/salt/cloud/clouds/msazure.py

The get_configured_provider() Function

This function uses `config.is_provider_configured()` to determine whether all required information for this driver has been configured. The last value in the list of required settings should be followed by a comma.

Libcloud Based Modules

Writing a cloud module based on libcloud has two major advantages. First of all, much of the work has already been done by the libcloud project. Second, most of the functions necessary to Salt have already been added to the Salt Cloud project.

The create() Function

The most important function that does need to be manually written is the `create()` function. This is what is used to request a virtual machine to be created by the cloud host, wait for it to become available, and then (optionally) log in and install Salt on it.

A good example to follow for writing a cloud driver module based on libcloud is the module provided for Linode:

https://github.com/saltstack/salt/tree/master/salt/cloud/clouds/linode.py

The basic flow of a `create()` function is as follows:

- Send a request to the cloud host to create a virtual machine.

- Wait for the virtual machine to become available.
- Generate kwargs to be used to deploy Salt.
- Log into the virtual machine and deploy Salt.
- Return a data structure that describes the newly-created virtual machine.

At various points throughout this function, events may be fired on the Salt event bus. Four of these events, which are described below, are required. Other events may be added by the user, where appropriate.

When the `create()` function is called, it is passed a data structure called `vm_`. This dict contains a composite of information describing the virtual machine to be created. A dict called `__opts__` is also provided by Salt, which contains the options used to run Salt Cloud, as well as a set of configuration and environment variables.

The first thing the `create()` function must do is fire an event stating that it has started the create process. This event is tagged `salt/cloud/<vm name>/creating`. The payload contains the names of the VM, profile, and provider.

A set of kwargs is then usually created, to describe the parameters required by the cloud host to request the virtual machine.

An event is then fired to state that a virtual machine is about to be requested. It is tagged as `salt/cloud/<vm name>/requesting`. The payload contains most or all of the parameters that will be sent to the cloud host. Any private information (such as passwords) should not be sent in the event.

After a request is made, a set of deploy kwargs will be generated. These will be used to install Salt on the target machine. Windows options are supported at this point, and should be generated, even if the cloud host does not currently support Windows. This will save time in the future if the host does eventually decide to support Windows.

An event is then fired to state that the deploy process is about to begin. This event is tagged `salt/cloud/<vm name>/deploying`. The payload for the event will contain a set of deploy kwargs, useful for debugging purposed. Any private data, including passwords and keys (including public keys) should be stripped from the deploy kwargs before the event is fired.

If any Windows options have been passed in, the `salt.utils.cloud.deploy_windows()` function will be called. Otherwise, it will be assumed that the target is a Linux or Unix machine, and the `salt.utils.cloud.deploy_script()` will be called.

Both of these functions will wait for the target machine to become available, then the necessary port to log in, then a successful login that can be used to install Salt. Minion configuration and keys will then be uploaded to a temporary directory on the target by the appropriate function. On a Windows target, the Windows Minion Installer will be run in silent mode. On a Linux/Unix target, a deploy script (`bootstrap-salt.sh`, by default) will be run, which will auto-detect the operating system, and install Salt using its native package manager. These do not need to be handled by the developer in the cloud module.

The `salt.utils.cloud.validate_windows_cred()` function has been extended to take the number of retries and retry_delay parameters in case a specific cloud host has a delay between providing the Windows credentials and the credentials being available for use. In their `create()` function, or as a sub-function called during the creation process, developers should use the `win_deploy_auth_retries` and `win_deploy_auth_retry_delay` parameters from the provider configuration to allow the end-user the ability to customize the number of tries and delay between tries for their particular host.

After the appropriate deploy function completes, a final event is fired which describes the virtual machine that has just been created. This event is tagged `salt/cloud/<vm name>/created`. The payload contains the names of the VM, profile, and provider.

Finally, a dict (queried from the provider) which describes the new virtual machine is returned to the user. Because this data is not fired on the event bus it can, and should, return any passwords that were returned by the cloud host. In some cases (for example, Rackspace), this is the only time that the password can be queried by the user; post-creation queries may not contain password information (depending upon the host).

The libcloudfuncs Functions

A number of other functions are required for all cloud hosts. However, with libcloud-based modules, these are all provided for free by the libcloudfuncs library. The following two lines set up the imports:

```
from salt.cloud.libcloudfuncs import *   # pylint: disable=W0614,W0401
import salt.utils.functools
```

And then a series of declarations will make the necessary functions available within the cloud module.

```
get_size = salt.utils.functools.namespaced_function(get_size, globals())
get_image = salt.utils.functools.namespaced_function(get_image, globals())
avail_locations = salt.utils.functools.namespaced_function(avail_locations, globals())
avail_images = salt.utils.functools.namespaced_function(avail_images, globals())
avail_sizes = salt.utils.functools.namespaced_function(avail_sizes, globals())
script = salt.utils.functools.namespaced_function(script, globals())
destroy = salt.utils.functools.namespaced_function(destroy, globals())
list_nodes = salt.utils.functools.namespaced_function(list_nodes, globals())
list_nodes_full = salt.utils.functools.namespaced_function(list_nodes_full, globals())
list_nodes_select = salt.utils.functools.namespaced_function(list_nodes_select,
↪globals())
show_instance = salt.utils.functools.namespaced_function(show_instance, globals())
```

If necessary, these functions may be replaced by removing the appropriate declaration line, and then adding the function as normal.

These functions are required for all cloud modules, and are described in detail in the next section.

Non-Libcloud Based Modules

In some cases, using libcloud is not an option. This may be because libcloud has not yet included the necessary driver itself, or it may be that the driver that is included with libcloud does not contain all of the necessary features required by the developer. When this is the case, some or all of the functions in `libcloudfuncs` may be replaced. If they are all replaced, the libcloud imports should be absent from the Salt Cloud module.

A good example of a non-libcloud driver is the DigitalOcean driver:

https://github.com/saltstack/salt/tree/master/salt/cloud/clouds/digitalocean.py

The `create()` Function

The `create()` function must be created as described in the libcloud-based module documentation.

The get_size() Function

This function is only necessary for libcloud-based modules, and does not need to exist otherwise.

The get_image() Function

This function is only necessary for libcloud-based modules, and does not need to exist otherwise.

The avail_locations() Function

This function returns a list of locations available, if the cloud host uses multiple data centers. It is not necessary if the cloud host uses only one data center. It is normally called using the `--list-locations` option.

```
salt-cloud --list-locations my-cloud-provider
```

The avail_images() Function

This function returns a list of images available for this cloud provider. There are not currently any known cloud providers that do not provide this functionality, though they may refer to images by a different name (for example, ”templates”). It is normally called using the `--list-images` option.

```
salt-cloud --list-images my-cloud-provider
```

The avail_sizes() Function

This function returns a list of sizes available for this cloud provider. Generally, this refers to a combination of RAM, CPU, and/or disk space. This functionality may not be present on some cloud providers. For example, the Parallels module breaks down RAM, CPU, and disk space into separate options, whereas in other providers, these options are baked into the image. It is normally called using the `--list-sizes` option.

```
salt-cloud --list-sizes my-cloud-provider
```

The script() Function

This function builds the deploy script to be used on the remote machine. It is likely to be moved into the `salt.utils.cloud` library in the near future, as it is very generic and can usually be copied wholesale from another module. An excellent example is in the Azure driver.

The destroy() Function

This function irreversibly destroys a virtual machine on the cloud provider. Before doing so, it should fire an event on the Salt event bus. The tag for this event is `salt/cloud/<vm name>/destroying`. Once the virtual machine has been destroyed, another event is fired. The tag for that event is `salt/cloud/<vm name>/destroyed`.

This function is normally called with the `-d` options:

```
salt-cloud -d myinstance
```

The list_nodes() Function

This function returns a list of nodes available on this cloud provider, using the following fields:

- id (str)
- image (str)
- size (str)

- state (str)
- private_ips (list)
- public_ips (list)

No other fields should be returned in this function, and all of these fields should be returned, even if empty. The private_ips and public_ips fields should always be of a list type, even if empty, and the other fields should always be of a str type. This function is normally called with the `-Q` option:

```
salt-cloud -Q
```

The list_nodes_full() Function

All information available about all nodes should be returned in this function. The fields in the list_nodes() function should also be returned, even if they would not normally be provided by the cloud provider. This is because some functions both within Salt and 3rd party will break if an expected field is not present. This function is normally called with the `-F` option:

```
salt-cloud -F
```

The list_nodes_select() Function

This function returns only the fields specified in the `query.selection` option in `/etc/salt/cloud`. Because this function is so generic, all of the heavy lifting has been moved into the `salt.utils.cloud` library.

A function to call `list_nodes_select()` still needs to be present. In general, the following code can be used as-is:

```
def list_nodes_select(call=None):
    '''
    Return a list of the VMs that are on the provider, with select fields
    '''
    return salt.utils.cloud.list_nodes_select(
        list_nodes_full('function'), __opts__['query.selection'], call,
    )
```

However, depending on the cloud provider, additional variables may be required. For instance, some modules use a `conn` object, or may need to pass other options into `list_nodes_full()`. In this case, be sure to update the function appropriately:

```
def list_nodes_select(conn=None, call=None):
    '''
    Return a list of the VMs that are on the provider, with select fields
    '''
    if not conn:
        conn = get_conn()   # pylint: disable=E0602

    return salt.utils.cloud.list_nodes_select(
        list_nodes_full(conn, 'function'),
        __opts__['query.selection'],
        call,
    )
```

This function is normally called with the `-S` option:

```
salt-cloud -S
```

The show_instance() Function

This function is used to display all of the information about a single node that is available from the cloud provider. The simplest way to provide this is usually to call `list_nodes_full()`, and return just the data for the requested node. It is normally called as an action:

```
salt-cloud -a show_instance myinstance
```

Actions and Functions

Extra functionality may be added to a cloud provider in the form of an `--action` or a `--function`. Actions are performed against a cloud instance/virtual machine, and functions are performed against a cloud provider.

Actions

Actions are calls that are performed against a specific instance or virtual machine. The `show_instance` action should be available in all cloud modules. Actions are normally called with the `-a` option:

```
salt-cloud -a show_instance myinstance
```

Actions must accept a `name` as a first argument, may optionally support any number of kwargs as appropriate, and must accept an argument of `call`, with a default of `None`.

Before performing any other work, an action should normally verify that it has been called correctly. It may then perform the desired feature, and return useful information to the user. A basic action looks like:

```
def show_instance(name, call=None):
'''
Show the details from EC2 concerning an AMI
'''
if call != 'action':
    raise SaltCloudSystemExit(
        'The show_instance action must be called with -a or --action.'
    )

return _get_node(name)
```

Please note that generic kwargs, if used, are passed through to actions as `kwargs` and not `**kwargs`. An example of this is seen in the Functions section.

Functions

Functions are called that are performed against a specific cloud provider. An optional function that is often useful is `show_image`, which describes an image in detail. Functions are normally called with the `-f` option:

```
salt-cloud -f show_image my-cloud-provider image='Ubuntu 13.10 64-bit'
```

A function may accept any number of kwargs as appropriate, and must accept an argument of `call` with a default of `None`.

Before performing any other work, a function should normally verify that it has been called correctly. It may then perform the desired feature, and return useful information to the user. A basic function looks like:

```
def show_image(kwargs, call=None):
    '''
    Show the details from EC2 concerning an AMI
    '''
    if call != 'function':
        raise SaltCloudSystemExit(
            'The show_image action must be called with -f or --function.'
        )

    params = {'ImageId.1': kwargs['image'],
              'Action': 'DescribeImages'}
    result = query(params)
    log.info(result)

    return result
```

Take note that generic kwargs are passed through to functions as `kwargs` and not `**kwargs`.

14.10.2 Cloud deployment scripts

Salt Cloud works primarily by executing a script on the virtual machines as soon as they become available. The script that is executed is referenced in the cloud profile as the `script`. In older versions, this was the `os` argument. This was changed in 0.8.2.

A number of legacy scripts exist in the deploy directory in the saltcloud source tree. The preferred method is currently to use the salt-bootstrap script. A stable version is included with each release tarball starting with 0.8.4. The most updated version can be found at:

https://github.com/saltstack/salt-bootstrap

Note that, somewhat counter-intuitively, this script is referenced as `bootstrap-salt` in the configuration.

You can specify a deploy script in the cloud configuration file (`/etc/salt/cloud` by default):

```
script: bootstrap-salt
```

Or in a provider:

```
my-provider:
  # snip...
  script: bootstrap-salt
```

Or in a profile:

```
my-profile:
  provider: my-provider
  # snip...
  script: bootstrap-salt
```

If you do not specify a script argument in your cloud configuration file, provider configuration or profile configuration, the ”bootstrap-salt” script will be used by default.

Other Generic Deploy Scripts

If you want to be assured of always using the latest Salt Bootstrap script, there are a few generic templates available in the deploy directory of your saltcloud source tree:

```
curl-bootstrap
curl-bootstrap-git
python-bootstrap
wget-bootstrap
wget-bootstrap-git
```

These are example scripts which were designed to be customized, adapted, and refit to meet your needs. One important use of them is to pass options to the salt-bootstrap script, such as updating to specific git tags.

Custom Deploy Scripts

If the Salt Bootstrap script does not meet your needs, you may write your own. The script should be written in shell and is a Jinja template. Deploy scripts need to execute a number of functions to do a complete salt setup. These functions include:

1. Install the salt minion. If this can be done via system packages this method is HIGHLY preferred.
2. Add the salt minion keys before the minion is started for the first time. The minion keys are available as strings that can be copied into place in the Jinja template under the dict named "vm".
3. Start the salt-minion daemon and enable it at startup time.
4. Set up the minion configuration file from the "minion" data available in the Jinja template.

A good, well commented example of this process is the Fedora deployment script:

https://github.com/saltstack/salt/blob/master/salt/cloud/deploy/Fedora.sh

A number of legacy deploy scripts are included with the release tarball. None of them are as functional or complete as Salt Bootstrap, and are still included for academic purposes.

Custom deploy scripts are picked up from `/etc/salt/cloud.deploy.d` by default, but you can change the location of deploy scripts with the cloud configuration `deploy_scripts_search_path`. Additionally, if your deploy script has the extension `.sh`, you can leave out the extension in your configuration.

For example, if your custom deploy script is located in `/etc/salt/cloud.deploy.d/my_deploy.sh`, you could specify it in a cloud profile like this:

```
my-profile:
  provider: my-provider
  # snip...
  script: my_deploy
```

You're also free to use the full path to the script if you like. Using full paths, your script doesn't have to live inside `/etc/salt/cloud.deploy.d` or whatever you've configured with `deploy_scripts_search_path`.

Post-Deploy Commands

Once a minion has been deployed, it has the option to run a salt command. Normally, this would be the `state.apply`, which would finish provisioning the VM. Another common option (for testing) is to use `test.version`. This is configured in the main cloud config file:

```
start_action: state.apply
```

This is currently considered to be experimental functionality, and may not work well with all cloud hosts. If you experience problems with Salt Cloud hanging after Salt is deployed, consider using Startup States instead:

http://docs.saltstack.com/ref/states/startup.html

Skipping the Deploy Script

For whatever reason, you may want to skip the deploy script altogether. This results in a VM being spun up much faster, with absolutely no configuration. This can be set from the command line:

```
salt-cloud --no-deploy -p micro_aws my_instance
```

Or it can be set from the main cloud config file:

```
deploy: False
```

Or it can be set from the provider's configuration:

```
RACKSPACE.user: example_user
RACKSPACE.apikey: 123984bjjas87034
RACKSPACE.deploy: False
```

Or even on the VM's profile settings:

```
ubuntu_aws:
  provider: my-ec2-config
  image: ami-7e2da54e
  size: t1.micro
  deploy: False
```

The default for deploy is True.

In the profile, you may also set the script option to `None`:

```
script: None
```

This is the slowest option, since it still uploads the None deploy script and executes it.

Updating Salt Bootstrap

Salt Bootstrap can be updated automatically with `salt-cloud`:

```
salt-cloud -u
salt-cloud --update-bootstrap
```

Bear in mind that this updates to the latest **stable** version from:

https://bootstrap.saltstack.com/stable/bootstrap-salt.sh

To update Salt Bootstrap script to the **develop** version, run the following command on the Salt minion host with `salt-cloud` installed:

```
salt-call config.gather_bootstrap_script 'https://bootstrap.saltstack.com/develop/
↪bootstrap-salt.sh'
```

Or just download the file manually:

```
curl -L 'https://bootstrap.saltstack.com/develop' > /etc/salt/cloud.deploy.d/
↪bootstrap-salt.sh
```

Keeping /tmp/ Files

When Salt Cloud deploys an instance, it uploads temporary files to /tmp/ for salt-bootstrap to put in place. After the script has run, they are deleted. To keep these files around (mostly for debugging purposes), the –keep-tmp option can be added:

```
salt-cloud -p myprofile mymachine --keep-tmp
```

For those wondering why /tmp/ was used instead of /root/, this had to be done for images which require the use of sudo, and therefore do not allow remote root logins, even for file transfers (which makes /root/ unavailable).

Deploy Script Arguments

Custom deploy scripts are unlikely to need custom arguments to be passed to them, but salt-bootstrap has been extended quite a bit, and this may be necessary. script_args can be specified in either the profile or the map file, to pass arguments to the deploy script:

```
aws-amazon:
  provider: my-ec2-config
  image: ami-1624987f
  size: t1.micro
  ssh_username: ec2-user
  script: bootstrap-salt
  script_args: -c /tmp/
```

This has also been tested to work with pipes, if needed:

```
script_args: '| head'
```

14.11 Using Salt Cloud from Salt

14.11.1 Using the Salt Modules for Cloud

In addition to the `salt-cloud` command, Salt Cloud can be called from Salt, in a variety of different ways. Most users will be interested in either the execution module or the state module, but it is also possible to call Salt Cloud as a runner.

Because the actual work will be performed on a remote minion, the normal Salt Cloud configuration must exist on any target minion that needs to execute a Salt Cloud command. Because Salt Cloud now supports breaking out configuration into individual files, the configuration is easily managed using Salt's own `file.managed` state function. For example, the following directories allow this configuration to be managed easily:

```
/etc/salt/cloud.providers.d/
/etc/salt/cloud.profiles.d/
```

Minion Keys

Keep in mind that when creating minions, Salt Cloud will create public and private minion keys, upload them to the minion, and place the public key on the machine that created the minion. It will *not* attempt to place any public minion keys on the master, unless the minion which was used to create the instance is also the Salt Master. This is because granting arbitrary minions access to modify keys on the master is a serious security risk, and must be avoided.

Execution Module

The `cloud` module is available to use from the command line. At the moment, almost every standard Salt Cloud feature is available to use. The following commands are available:

list_images

This command is designed to show images that are available to be used to create an instance using Salt Cloud. In general they are used in the creation of profiles, but may also be used to create an instance directly (see below). Listing images requires a provider to be configured, and specified:

```
salt myminion cloud.list_images my-cloud-provider
```

list_sizes

This command is designed to show sizes that are available to be used to create an instance using Salt Cloud. In general they are used in the creation of profiles, but may also be used to create an instance directly (see below). This command is not available for all cloud providers; see the provider-specific documentation for details. Listing sizes requires a provider to be configured, and specified:

```
salt myminion cloud.list_sizes my-cloud-provider
```

list_locations

This command is designed to show locations that are available to be used to create an instance using Salt Cloud. In general they are used in the creation of profiles, but may also be used to create an instance directly (see below). This command is not available for all cloud providers; see the provider-specific documentation for details. Listing locations requires a provider to be configured, and specified:

```
salt myminion cloud.list_locations my-cloud-provider
```

query

This command is used to query all configured cloud providers, and display all instances associated with those accounts. By default, it will run a standard query, returning the following fields:

`id` The name or ID of the instance, as used by the cloud provider.

`image` The disk image that was used to create this instance.

`private_ips` Any public IP addresses currently assigned to this instance.

`public_ips` Any private IP addresses currently assigned to this instance.

size The size of the instance; can refer to RAM, CPU(s), disk space, etc., depending on the cloud provider.

state The running state of the instance; for example, `running`, `stopped`, `pending`, etc. This state is dependent upon the provider.

This command may also be used to perform a full query or a select query, as described below. The following usages are available:

```
salt myminion cloud.query
salt myminion cloud.query list_nodes
salt myminion cloud.query list_nodes_full
```

full_query

This command behaves like the `query` command, but lists all information concerning each instance as provided by the cloud provider, in addition to the fields returned by the `query` command.

```
salt myminion cloud.full_query
```

select_query

This command behaves like the `query` command, but only returned select fields as defined in the `/etc/salt/cloud` configuration file. A sample configuration for this section of the file might look like:

```
query.selection:
  - id
  - key_name
```

This configuration would only return the `id` and `key_name` fields, for those cloud providers that support those two fields. This would be called using the following command:

```
salt myminion cloud.select_query
```

profile

This command is used to create an instance using a profile that is configured on the target minion. Please note that the profile must be configured before this command can be used with it.

```
salt myminion cloud.profile ec2-centos64-x64 my-new-instance
```

Please note that the execution module does *not* run in parallel mode. Using multiple minions to create instances can effectively perform parallel instance creation.

create

This command is similar to the `profile` command, in that it is used to create a new instance. However, it does not require a profile to be pre-configured. Instead, all of the options that are normally configured in a profile are passed directly to Salt Cloud to create the instance:

```
salt myminion cloud.create my-ec2-config my-new-instance \
    image=ami-1624987f size='t1.micro' ssh_username=ec2-user \
    securitygroup=default delvol_on_destroy=True
```

Please note that the execution module does *not* run in parallel mode. Using multiple minions to create instances can effectively perform parallel instance creation.

destroy

This command is used to destroy an instance or instances. This command will search all configured providers and remove any instance(s) which matches the name(s) passed in here. The results of this command are *non-reversable* and should be used with caution.

```
salt myminion cloud.destroy myinstance
salt myminion cloud.destroy myinstance1,myinstance2
```

action

This command implements both the `action` and the `function` commands used in the standard `salt-cloud` command. If one of the standard `action` commands is used, an instance name must be provided. If one of the standard `function` commands is used, a provider configuration must be named.

```
salt myminion cloud.action start instance=myinstance
salt myminion cloud.action show_image provider=my-ec2-config \
    image=ami-1624987f
```

The actions available are largely dependent upon the module for the specific cloud provider. The following actions are available for all cloud providers:

`list_nodes` This is a direct call to the `query` function as described above, but is only performed against a single cloud provider. A provider configuration must be included.

`list_nodes_select` This is a direct call to the `full_query` function as described above, but is only performed against a single cloud provider. A provider configuration must be included.

`list_nodes_select` This is a direct call to the `select_query` function as described above, but is only performed against a single cloud provider. A provider configuration must be included.

`show_instance` This is a thin wrapper around `list_nodes`, which returns the full information about a single instance. An instance name must be provided.

State Module

A subset of the execution module is available through the `cloud` state module. Not all functions are currently included, because there is currently insufficient code for them to perform statefully. For example, a command to create an instance may be issued with a series of options, but those options cannot currently be statefully managed. Additional states to manage these options will be released at a later time.

cloud.present

This state will ensure that an instance is present inside a particular cloud provider. Any option that is normally specified in the `cloud.create` execution module and function may be declared here, but only the actual presence of the instance will be managed statefully.

```
my-instance-name:
  cloud.present:
    - cloud_provider: my-ec2-config
    - image: ami-1624987f
    - size: 't1.micro'
    - ssh_username: ec2-user
    - securitygroup: default
    - delvol_on_destroy: True
```

cloud.profile

This state will ensure that an instance is present inside a particular cloud provider. This function calls the `cloud.profile` execution module and function, but as with `cloud.present`, only the actual presence of the instance will be managed statefully.

```
my-instance-name:
  cloud.profile:
    - profile: ec2-centos64-x64
```

cloud.absent

This state will ensure that an instance (identified by name) does not exist in any of the cloud providers configured on the target minion. Please note that this state is *non-reversable* and may be considered especially destructive when issued as a cloud state.

```
my-instance-name:
  cloud.absent
```

Runner Module

The `cloud` runner module is executed on the master, and performs actions using the configuration and Salt modules on the master itself. This means that any public minion keys will also be properly accepted by the master.

Using the functions in the runner module is no different than using those in the execution module, outside of the behavior described in the above paragraph. The following functions are available inside the runner:

- list_images
- list_sizes
- list_locations
- query
- full_query
- select_query
- profile
- destroy
- action

Outside of the standard usage of `salt-run` itself, commands are executed as usual:

```
salt-run cloud.profile ec2-centos64-x86_64 my-instance-name
```

CloudClient

The execution, state, and runner modules ultimately all use the CloudClient library that ships with Salt. To use the CloudClient library locally (either on the master or a minion), create a client object and issue a command against it:

```
import salt.cloud
import pprint
client = salt.cloud.CloudClient('/etc/salt/cloud')
nodes = client.query()
pprint.pprint(nodes)
```

Reactor

Examples of using the reactor with Salt Cloud are available in the ec2-autoscale-reactor and salt-cloud-reactor formulas.

14.12 Feature Comparison

14.12.1 Feature Matrix

A number of features are available in most cloud hosts, but not all are available everywhere. This may be because the feature isn't supported by the cloud host itself, or it may only be that the feature has not yet been added to Salt Cloud. In a handful of cases, it is because the feature does not make sense for a particular cloud provider (Saltify, for instance).

This matrix shows which features are available in which cloud hosts, as far as Salt Cloud is concerned. This is not a comprehensive list of all features available in all cloud hosts, and should not be used to make business decisions concerning choosing a cloud host. In most cases, adding support for a feature to Salt Cloud requires only a little effort.

Legacy Drivers

Both AWS and Rackspace are listed as "Legacy". This is because those drivers have been replaced by other drivers, which are generally the preferred method for working with those hosts.

The EC2 driver should be used instead of the AWS driver, when possible. The OpenStack driver should be used instead of the Rackspace driver, unless the user is dealing with instances in "the old cloud" in Rackspace.

Note for Developers

When adding new features to a particular cloud host, please make sure to add the feature to this table. Additionally, if you notice a feature that is not properly listed here, pull requests to fix them is appreciated.

Standard Features

These are features that are available for almost every cloud host.

	AWS (Legacy)	CloudStack	Digital Ocean	EC2	GoGrid	JoyEnt	Linode	OpenStack	Parallels	Rackspace (Legacy)	Saltify	Vagrant	Softlayer	Softlayer Hardware	Aliyun	Tencent Cloud
Query	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	[1]	[1]	Yes	Yes	Yes	Yes
Full Query	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	[1]	[1]	Yes	Yes	Yes	Yes
Selective Query	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	[1]	[1]	Yes	Yes	Yes	Yes
List Sizes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	[2]	[2]	Yes	Yes	Yes	Yes
List Images	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
List Locations	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	[2]	[2]	Yes	Yes	Yes	Yes
create	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	[1]	Yes	Yes	Yes	Yes
destroy	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	[1]	[1]	Yes	Yes	Yes	Yes

[1] Yes, if salt-api is enabled.

[2] Always returns *{}*.

Actions

These are features that are performed on a specific instance, and require an instance name to be passed in. For example:

```
# salt-cloud -a attach_volume ami.example.com
```

Actions	AWS (Legacy)	CloudStack	Digital Ocean	EC2	GoGrid	JoyEnt	Linode	OpenStack	Parallels	Rackspace (Legacy)	**Saltify&** Vagrant	Softlayer	Softlayer Hardware	Aliyun	Tencent Cloud
attach_volume				Yes											
create_attach_volumes	Yes			Yes											
del_tags	Yes			Yes											
delvol_on_destroy				Yes											
detach_volume				Yes											
disable_term_protect	Yes			Yes											
enable_term_protect	Yes			Yes											
get_tags	Yes			Yes											
keepvol_on_destroy				Yes											
list_keypairs			Yes												
rename	Yes			Yes											
set_tags	Yes			Yes											
show_delvol_on_destroy				Yes											
show_instance			Yes	Yes			Yes		Yes			Yes	Yes	Yes	Yes
show_term_protect				Yes											
start	Yes			Yes		Yes	Yes		Yes					Yes	Yes
stop	Yes			Yes		Yes	Yes		Yes					Yes	Yes
take_action						Yes									

Functions

These are features that are performed against a specific cloud provider, and require the name of the provider to be passed in. For example:

```
# salt-cloud -f list_images my_digitalocean
```

Functions	AWS (Legacy)	CloudStack	Digital Ocean	EC2	GoGrid	JoyEnt	Linode	OpenStack	Parallels	Rackspace (Legacy)	**Saltify&** Vagrant	Softlayer	Softlayer Hardware	Aliyun	Tencent Cloud
block_device_mappings	Yes														
create_keypair				Yes											
create_volume				Yes											
delete_key						Yes									
delete_keypair				Yes											
delete_volume				Yes											
get_image			Yes			Yes			Yes					Yes	
get_ip		Yes													
get_key		Yes													
get_keyid			Yes												
get_keypair		Yes													
get_networkid		Yes													
get_node						Yes									
get_password		Yes													
get_size			Yes			Yes								Yes	
get_spot_config				Yes											

Continued on next page

Table 1 – continued from previous page

Functions	AWS (Legacy)	CloudStack	Digital Ocean	EC2	GoGrid	JoyEnt	Linode	OpenStack	Parallels	Rackspace (Legacy)	Saltify & Vagrant	Softlayer	Softlayer Hardware	Aliyun	Tencent Cloud
get_subnetid				Yes											
iam_profile	Yes			Yes										Yes	
import_key						Yes									
key_list						Yes									
keyname	Yes			Yes											
list_availability_zones				Yes										Yes	Yes
list_custom_images												Yes			Yes
list_keys						Yes									
list_nodes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
list_nodes_full	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
list_nodes_select	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
list_vlans												Yes	Yes		
rackconnect								Yes							
reboot				Yes		Yes					[1]			Yes	Yes
reformat_node						Yes									
securitygroup	Yes			Yes											
securitygroupid				Yes										Yes	
show_image				Yes					Yes					Yes	Yes
show_key						Yes									
show_keypair			Yes	Yes											
show_volume				Yes											

[1] Yes, if salt-api is enabled.

14.13 Tutorials

14.13.1 Salt Cloud Quickstart

Salt Cloud is built-in to Salt, and the easiest way to run Salt Cloud is directly from your Salt Master.

Note that if you installed Salt via Salt Bootstrap, it may not have automatically installed salt-cloud for you. Use your distribution's package manager to install the `salt-cloud` package from the same repo that you used to install Salt. These repos will automatically be setup by Salt Bootstrap.

Alternatively, the `-L` option can be passed to the Salt Bootstrap script when installing Salt. The `-L` option will install `salt-cloud` and the required `libcloud` package.

This quickstart walks you through the basic steps of setting up a cloud host and defining some virtual machines to create.

Note: Salt Cloud has its own process and does not rely on the Salt Master, so it can be installed on a standalone minion instead of your Salt Master.

Define a Provider

The first step is to add the credentials for your cloud host. Credentials and other settings provided by the cloud host are stored in provider configuration files. Provider configurations contain the details needed to connect to a cloud host such as EC2, GCE, Rackspace, etc., and any global options that you want set on your cloud minions (such as the location of your Salt Master).

On your Salt Master, browse to `/etc/salt/cloud.providers.d/` and create a file called `<provider>.conf`, replacing `<provider>` with `ec2`, `softlayer`, and so on. The name helps you identify the contents, and is not important as long as the file ends in `.conf`.

Next, browse to the *Provider specifics* and add any required settings for your cloud host to this file. Here is an example for Amazon EC2:

```
my-ec2:
  driver: ec2
  # Set the EC2 access credentials (see below)
  #
  id: 'HJGRYCILJLKJYG'
  key: 'kdjgfsgm;woormgl/aserigjksjdhasdfgn'
  # Make sure this key is owned by root with permissions 0400.
  #
  private_key: /etc/salt/my_test_key.pem
  keyname: my_test_key
  securitygroup: default
  # Optional: Set up the location of the Salt Master
  #
  minion:
    master: saltmaster.example.com
```

The required configuration varies between cloud hosts so make sure you read the provider specifics.

List Cloud Provider Options

You can now query the cloud provider you configured for available locations, images, and sizes. This information is used when you set up VM profiles.

```
salt-cloud --list-locations <provider_name>  # my-ec2 in the previous example
salt-cloud --list-images <provider_name>
salt-cloud --list-sizes <provider_name>
```

Replace `<provider_name>` with the name of the provider configuration you defined.

Create VM Profiles

On your Salt Master, browse to `/etc/salt/cloud.profiles.d/` and create a file called `<profile>.conf`, replacing `<profile>` with `ec2`, `softlayer`, and so on. The file must end in `.conf`.

You can now add any custom profiles you'd like to define to this file. Here are a few examples:

```
micro_ec2:
  provider: my-ec2
  image: ami-d514f291
  size: t1.micro

medium_ec2:
```

(continues on next page)

(continued from previous page)

```
  provider: my-ec2
  image: ami-d514f291
  size: m3.medium

large_ec2:
  provider: my-ec2
  image: ami-d514f291
  size: m3.large
```

Notice that the `provider` in our profile matches the provider name that we defined? That is how Salt Cloud knows how to connect to a cloud host to create a VM with these attributes.

Create VMs

VMs are created by calling `salt-cloud` with the following options:

```
salt-cloud -p <profile> <name1> <name2> ...
```

For example:

```
salt-cloud -p micro_ec2 minion1 minion2
```

Destroy VMs

Add a `-d` and the minion name you provided to destroy:

```
salt-cloud -d minion1 minion2
```

Query VMs

You can view details about the VMs you've created using `--query`:

```
salt-cloud --query
```

Cloud Map

Now that you know how to create and destoy individual VMs, next you should learn how to use a cloud map to create a number of VMs at once.

Cloud maps let you define a map of your infrastructure and quickly provision any number of VMs. On subsequent runs, any VMs that do not exist are created, and VMs that are already configured are left unmodified.

See *Cloud Map File*.

14.13.2 Using Salt Cloud with the Event Reactor

One of the most powerful features of the Salt framework is the Event Reactor. As the Reactor was in development, Salt Cloud was regularly updated to take advantage of the Reactor upon completion. As such, various aspects of both the creation and destruction of instances with Salt Cloud fire events to the Salt Master, which can be used by the Event Reactor.

Event Structure

As of this writing, all events in Salt Cloud have a tag, which includes the ID of the instance being managed, and a payload which describes the task that is currently being handled. A Salt Cloud tag looks like:

```
salt/cloud/<minion_id>/<task>
```

For instance, the first event fired when creating an instance named `web1` would look like:

```
salt/cloud/web1/creating
```

Assuming this instance is using the `ec2-centos` profile, which is in turn using the `ec2-config` provider, the payload for this tag would look like:

```
{'name': 'web1',
 'profile': 'ec2-centos',
 'provider': 'ec2-config:ec2'}
```

Available Events

When an instance is created in Salt Cloud, whether by map, profile, or directly through an API, a minimum of five events are normally fired. More may be available, depending upon the cloud provider being used. Some of the common events are described below.

salt/cloud/<minion_id>/creating

This event states simply that the process to create an instance has begun. At this point in time, no actual work has begun. The payload for this event includes:

name profile provider

salt/cloud/<minion_id>/requesting

Salt Cloud is about to make a request to the cloud provider to create an instance. At this point, all of the variables required to make the request have been gathered, and the payload of the event will reflect those variables which do not normally pose a security risk. What is returned here is dependent upon the cloud provider. Some common variables are:

name image size location

salt/cloud/<minion_id>/querying

The instance has been successfully requested, but the necessary information to log into the instance (such as IP address) is not yet available. This event marks the beginning of the process to wait for this information.

The payload for this event normally only includes the `instance_id`.

salt/cloud/<minion_id>/waiting_for_ssh

The information required to log into the instance has been retrieved, but the instance is not necessarily ready to be accessed. Following this event, Salt Cloud will wait for the IP address to respond to a ping, then wait for the specified port (usually 22) to respond to a connection, and on Linux systems, for SSH to become available. Salt Cloud will attempt to issue the `date` command on the remote system, as a means to check for availability. If no `ssh_username` has been specified, a list of usernames (starting with `root`) will be attempted. If one or more usernames was configured for `ssh_username`, they will be added to the beginning of the list, in order.

The payload for this event normally only includes the `ip_address`.

salt/cloud/<minion_id>/deploying

The necessary port has been detected as available, and now Salt Cloud can log into the instance, upload any files used for deployment, and run the deploy script. Once the script has completed, Salt Cloud will log back into the instance and remove any remaining files.

A number of variables are used to deploy instances, and the majority of these will be available in the payload. Any keys, passwords or other sensitive data will be scraped from the payload. Most of the variables returned will be related to the profile or provider config, and any default values that could have been changed in the profile or provider, but weren't.

salt/cloud/<minion_id>/created

The deploy sequence has completed, and the instance is now available, Salted, and ready for use. This event is the final task for Salt Cloud, before returning instance information to the user and exiting.

The payload for this event contains little more than the initial `creating` event. This event is required in all cloud providers.

Filtering Events

When creating a VM, it is possible with certain tags to filter how much information is sent to the event bus. The tags that can be filtered on any provider are:

- `salt/cloud/<minion_id>/creating`
- `salt/cloud/<minion_id>/requesting`
- `salt/cloud/<minion_id>/created`

Other providers may allow other tags to be filtered; when that is the case, the documentation for that provider will contain more details.

To filter information, create a section in your `/etc/salt/cloud` file called `filter_events`. Create a section for each tag that you want to filter, using the last segment of the tag. For instance, use `creating` to represent `salt/cloud/<minion_id>/creating`:

```yaml
filter_events:
  creating:
    keys:
      - name
      - profile
      - provider
```

Any keys listed here will be added to the default keys that are already set to be displayed for that provider. If you wish to start with a clean slate and only show the keys specified, add another option called `use_defaults` and set it to `False`.

```
filter_events:
  creating:
    keys:
      - name
      - profile
      - provider
    use_defaults: False
```

Configuring the Event Reactor

The Event Reactor is built into the Salt Master process, and as such is configured via the master configuration file. Normally this will be a YAML file located at `/etc/salt/master`. Additionally, master configuration items can be stored, in YAML format, inside the `/etc/salt/master.d/` directory.

These configuration items may be stored in either location; however, they may only be stored in one location. For organizational and security purposes, it may be best to create a single configuration file, which contains only Event Reactor configuration, at `/etc/salt/master.d/reactor`.

The Event Reactor uses a top-level configuration item called `reactor`. This block contains a list of tags to be watched for, each of which also includes a list of `sls` files. For instance:

```
reactor:
  - 'salt/minion/*/start':
    - '/srv/reactor/custom-reactor.sls'
  - 'salt/cloud/*/created':
    - '/srv/reactor/cloud-alert.sls'
  - 'salt/cloud/*/destroyed':
    - '/srv/reactor/cloud-destroy-alert.sls'
```

The above configuration configures reactors for three different tags: one which is fired when a minion process has started and is available to receive commands, one which is fired when a cloud instance has been created, and one which is fired when a cloud instance is destroyed.

Note that each tag contains a wildcard (`*`) in it. For each of these tags, this will normally refer to a `minion_id`. This is not required of event tags, but is very common.

Reactor SLS Files

Reactor `sls` files should be placed in the `/srv/reactor/` directory for consistency between environments, but this is not currently enforced by Salt.

Reactor `sls` files follow a similar format to other `sls` files in Salt. By default they are written in YAML and can be templated using Jinja, but since they are processed through Salt's rendering system, any available renderer (JSON, Mako, Cheetah, etc.) can be used.

As with other `sls` files, each stanza will start with a declaration ID, followed by the function to run, and then any arguments for that function. For example:

```
# /srv/reactor/cloud-alert.sls
new_instance_alert:
  cmd.pagerduty.create_event:
    - tgt: alertserver
    - kwarg:
        description: "New instance: {{ data['name'] }}"
        details: "New cloud instance created on {{ data['provider'] }}"
```

(continues on next page)

(continued from previous page)

```
        service_key: 1626dead5ecafe46231e968eb1be29c4
        profile: my-pagerduty-account
```

When the Event Reactor receives an event notifying it that a new instance has been created, this `sls` will create a new incident in PagerDuty, using the configured PagerDuty account.

The declaration ID in this example is `new_instance_alert`. The function called is `cmd.pagerduty.create_event`. The `cmd` portion of this function specifies that an execution module and function will be called, in this case, the `pagerduty.create_event` function.

Because an execution module is specified, a target (`tgt`) must be specified on which to call the function. In this case, a minion called `alertserver` has been used. Any arguments passed through to the function are declared in the `kwarg` block.

Example: Reactor-Based Highstate

When Salt Cloud creates an instance, by default it will install the Salt Minion onto the instance, along with any specified minion configuration, and automatically accept that minion's keys on the master. One of the configuration options that can be specified is `startup_states`, which is commonly set to `highstate`. This will tell the minion to immediately apply a *highstate*, as soon as it is able to do so.

This can present a problem with some system images on some cloud hosts. For instance, Salt Cloud can be configured to log in as either the `root` user, or a user with `sudo` access. While some hosts commonly use images that lock out remote `root` access and require a user with `sudo` privileges to log in (notably EC2, with their `ec2-user` login), most cloud hosts fall back to `root` as the default login on all images, including for operating systems (such as Ubuntu) which normally disallow remote `root` login.

For users of these operating systems, it is understandable that a *highstate* would include configuration to block remote `root` logins again. However, Salt Cloud may not have finished cleaning up its deployment files by the time the minion process has started, and kicked off a *highstate* run. Users have reported errors from Salt Cloud getting locked out while trying to clean up after itself.

The goal of a startup state may be achieved using the Event Reactor. Because a minion fires an event when it is able to receive commands, this event can effectively be used inside the reactor system instead. The following will point the reactor system to the right `sls` file:

```
reactor:
  - 'salt/cloud/*/created':
    - '/srv/reactor/startup_highstate.sls'
```

And the following `sls` file will start a *highstate* run on the target minion:

```
# /srv/reactor/startup_highstate.sls
reactor_highstate:
  cmd.state.apply:
    - tgt: {{ data['name'] }}
```

Because this event will not be fired until Salt Cloud has cleaned up after itself, the *highstate* run will not step on salt-cloud's toes. And because every file on the minion is configurable, including `/etc/salt/minion`, the `startup_states` can still be configured for future minion restarts, if desired.

CHAPTER

FIFTEEN

SALT PROXY MINION

Proxy minions are a developing Salt feature that enables controlling devices that, for whatever reason, cannot run a standard salt-minion. Examples include network gear that has an API but runs a proprietary OS, devices with limited CPU or memory, or devices that could run a minion, but for security reasons, will not.

There are some *proxy modules* available, but if your device interface is not currently supported you will most likely have to write the interface yourself, because there are an infinite number of controllable devices. Fortunately, this is only as difficult as the actual interface to the proxied device. Devices that have an existing Python module (PyUSB for example) would be relatively simple to interface. Code to control a device that has an HTML REST-based interface should be easy. Code to control your typical housecat would be excellent source material for a PhD thesis.

Salt proxy-minions provide the 'plumbing' that allows device enumeration and discovery, control, status, remote execution, and state management.

See the *Proxy Minion Walkthrough* for an end-to-end demonstration of a working REST-based proxy minion.

See the *Proxy Minion SSH Walkthrough* for an end-to-end demonstration of a working SSH proxy minion.

See *Proxyminion States* to configure and run `salt-proxy` on a remote minion. Specify all your master side proxy (pillar) configuration and use this state to remotely configure proxies on one or more minions.

See *Proxyminion Beacon* to help with easy configuration and management of `salt-proxy` processes.

15.1 New in 2017.7.0

The `proxy_merge_grains_in_module` configuration variable introduced in 2016.3, has been changed, defaulting to `True`.

The connection with the remote device is kept alive by default, when the module implements the `alive` function and `proxy_keep_alive` is set to `True`. The polling interval is set using the `proxy_keep_alive_interval` option which defaults to 1 minute.

The developers are also able to use the `proxy_always_alive`, when designing a proxy module flexible enough to open the connection with the remote device only when required.

15.2 New in 2016.11.0

Proxy minions now support configuration files with names ending in '*.conf' and placed in /etc/salt/proxy.d.

Proxy minions can now be configured in /etc/salt/proxy or /etc/salt/proxy.d instead of just pillar. Configuration format is the same as it would be in pillar.

15.3 New in 2016.3

The deprecated config option `enumerate_proxy_minions` has been removed.

As mentioned in earlier documentation, the `add_proxymodule_to_opts` configuration variable defaults to `False` in this release. This means if you have proxymodules or other code looking in `__opts__['proxymodule']` you will need to set this variable in your `/etc/salt/proxy` file, or modify your code to use the *__proxy__* injected variable.

The `__proxyenabled__` directive now only applies to grains and proxy modules themselves. Standard execution modules and state modules are not prevented from loading for proxy minions.

Enhancements in grains processing have made the `__proxyenabled__` directive somewhat redundant in dynamic grains code. It is still required, but best practices for the `__virtual__` function in grains files have changed. It is now recommended that the `__virtual__` functions check to make sure they are being loaded for the correct proxytype, example below:

```
def __virtual__():
    '''
    Only work on proxy
    '''
    try:
        if salt.utils.platform.is_proxy() and \
           __opts__['proxy']['proxytype'] == 'ssh_sample':
            return __virtualname__
    except KeyError:
        pass

    return False
```

The try/except block above exists because grains are processed very early in the proxy minion startup process, sometimes earlier than the proxy key in the `__opts__` dictionary is populated.

Grains are loaded so early in startup that no dunder dictionaries are present, so `__proxy__`, `__salt__`, etc. are not available. Custom grains located in `/srv/salt/_grains` and in the salt install grains directory can now take a single argument, `proxy`, that is identical to `__proxy__`. This enables patterns like

```
def get_ip(proxy):
    '''
    Ask the remote device what IP it has
    '''
    return {'ip':proxy['proxymodulename.get_ip']()}
```

Then the grain `ip` will contain the result of calling the `get_ip()` function in the proxymodule called `proxymodulename`.

Proxy modules now benefit from including a function called `initialized()`. This function should return `True` if the proxy's `init()` function has been successfully called. This is needed to make grains processing easier.

Finally, if there is a function called `grains` in the proxymodule, it will be executed on proxy-minion startup and its contents will be merged with the rest of the proxy's grains. Since older proxy-minions might have used other methods to call such a function and add its results to grains, this is config-gated by a new proxy configuration option called `proxy_merge_grains_in_module`. This defaults to `True` in the **2017.7.0** release.

15.4 New in 2015.8.2

BREAKING CHANGE: Adding the *proxymodule* variable to __opts__ is deprecated. The *proxymodule* variable has been moved a new globally-injected variable called *__proxy__*. A related configuration option called *add_proxymodule_to_opts* has been added and defaults to *True*. In the next major release, 2016.3.0, this variable will default to False.

In the meantime, proxies that functioned under 2015.8.0 and .1 should continue to work under 2015.8.2. You should rework your proxy code to use *__proxy__* as soon as possible.

The *rest_sample* example proxy minion has been updated to use *__proxy__*.

This change was made because proxymodules are a LazyLoader object, but LazyLoaders cannot be serialized. *__opts__* gets serialized, and so things like *saltutil.sync_all* and *state.highstate* would throw exceptions.

Support has been added to Salt's loader allowing custom proxymodules to be placed in `salt://_proxy`. Proxy minions that need these modules will need to be restarted to pick up any changes. A corresponding utility function, `saltutil.sync_proxymodules`, has been added to sync these modules to minions.

In addition, a salt.utils helper function called *is_proxy()* was added to make it easier to tell when the running minion is a proxy minion. **NOTE: This function was renamed to salt.utils.platform.is_proxy() for the 2018.3.0 release**

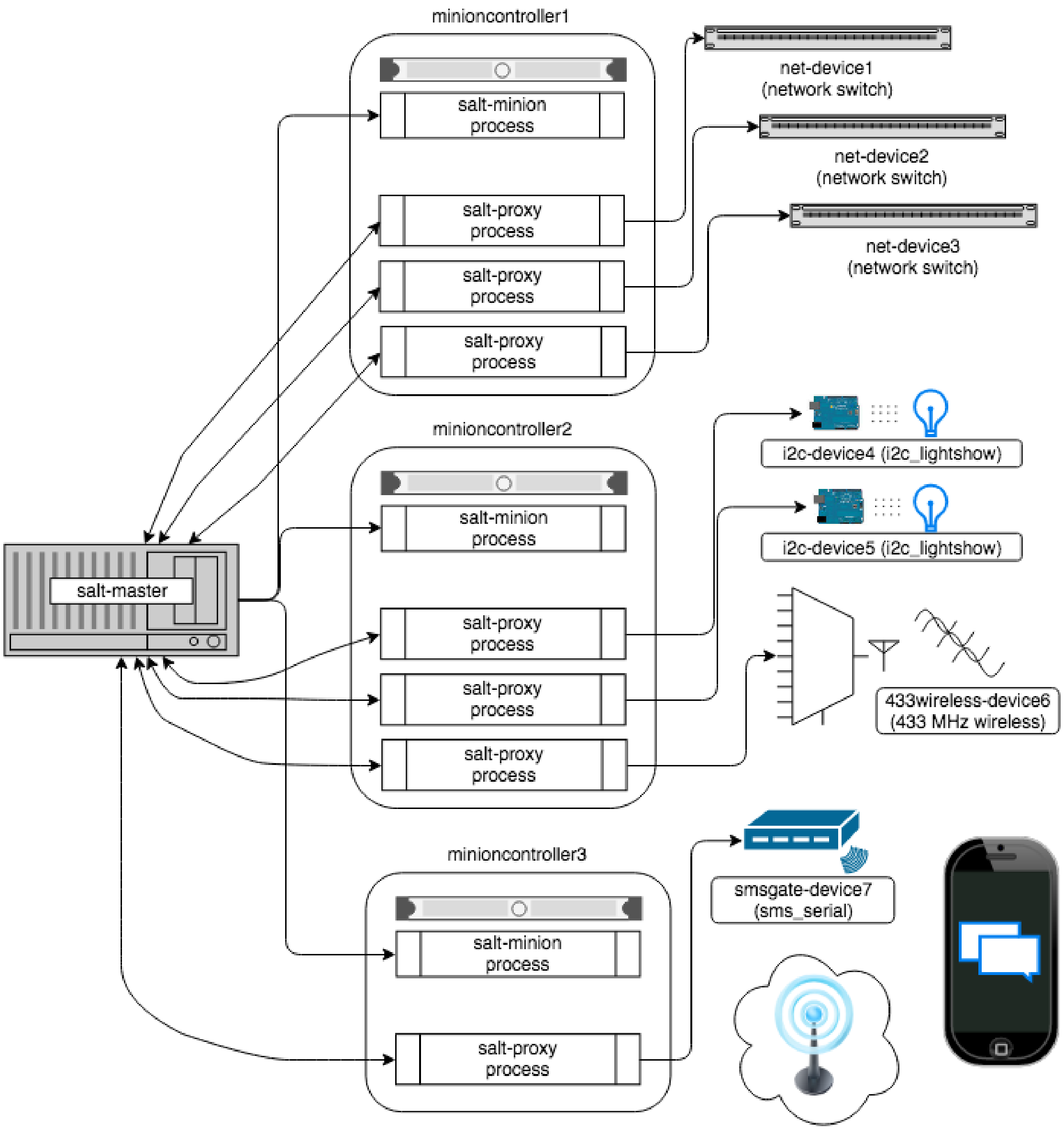
15.5 New in 2015.8

Starting with the 2015.8 release of Salt, proxy processes are no longer forked off from a controlling minion. Instead, they have their own script `salt-proxy` which takes mostly the same arguments that the standard Salt minion does with the addition of `--proxyid`. This is the id that the salt-proxy will use to identify itself to the master. Proxy configurations are still best kept in Pillar and their format has not changed.

This change allows for better process control and logging. Proxy processes can now be listed with standard process management utilities (`ps` from the command line). Also, a full Salt minion is no longer required (though it is still strongly recommended) on machines hosting proxies.

15.6 Getting Started

The following diagram may be helpful in understanding the structure of a Salt installation that includes proxy-minions:


minioncontroller1
salt-minion process
salt-proxy process
salt-proxy process
salt-proxy process
net-device1 (network switch)
net-device2 (network switch)
net-device3 (network switch)
minioncontroller2
salt-minion process
salt-proxy process
salt-proxy process
salt-proxy process
i2c-device4 (i2c_lightshow)
i2c-device5 (i2c_lightshow)
433wireless-device6 (433 MHz wireless)
salt-master
minioncontroller3
salt-minion process
salt-proxy process
smsgate-device7 (sms_serial)

The key thing to remember is the left-most section of the diagram. Salt's nature is to have a minion connect to a master, then the master may control the minion. However, for proxy minions, the target device cannot run a minion.

After the proxy minion is started and initiates its connection to the device, it connects back to the salt-master and for all intents and purposes looks like just another minion to the Salt master.

To create support for a proxied device one needs to create four things:

1. The *proxy_connection_module* (located in salt/proxy).
2. The *grains support code* (located in salt/grains).
3. *Salt modules* specific to the controlled device.
4. *Salt states* specific to the controlled device.

15.6.1 Configuration parameters

Proxy minions require no configuration parameters in /etc/salt/master.

Salt's Pillar system is ideally suited for configuring proxy-minions (though they can be configured in /etc/salt/proxy as well). Proxies can either be designated via a pillar file in pillar_roots, or through an external pillar. External pillars afford the opportunity for interfacing with a configuration management system, database, or other knowledgeable system that that may already contain all the details of proxy targets. To use static files in pillar_roots, pattern your files after the following examples, which are based on the diagram above:

/srv/pillar/top.sls

```
base:
  net-device1:
    - net-device1
  net-device2:
    - net-device2
  net-device3:
    - net-device3
  i2c-device4:
    - i2c-device4
  i2c-device5:
    - i2c-device5
  433wireless-device6:
    - 433wireless-device6
  smsgate-device7:
    - device7
```

/srv/pillar/net-device1.sls

```
proxy:
  proxytype: networkswitch
  host: 172.23.23.5
  username: root
  passwd: letmein
```

/srv/pillar/net-device2.sls

```
proxy:
  proxytype: networkswitch
  host: 172.23.23.6
  username: root
  passwd: letmein
```

/srv/pillar/net-device3.sls

```
proxy:
  proxytype: networkswitch
  host: 172.23.23.7
  username: root
  passwd: letmein
```

/srv/pillar/i2c-device4.sls

```
proxy:
  proxytype: i2c_lightshow
  i2c_address: 1
```

/srv/pillar/i2c-device5.sls

```
proxy:
  proxytype: i2c_lightshow
  i2c_address: 2
```

`/srv/pillar/433wireless-device6.sls`

```
proxy:
  proxytype: 433mhz_wireless
```

`/srv/pillar/smsgate-device7.sls`

```
proxy:
  proxytype: sms_serial
  deventry: /dev/tty04
```

Note the contents of each minioncontroller key may differ widely based on the type of device that the proxy-minion is managing.

In the above example

- net-devices 1, 2, and 3 are network switches that have a management interface available at a particular IP address.
- i2c-devices 4 and 5 are very low-level devices controlled over an i2c bus. In this case the devices are physically connected to machine 'minioncontroller2', and are addressable on the i2c bus at their respective i2c addresses.
- 433wireless-device6 is a 433 MHz wireless transmitter, also physically connected to minioncontroller2
- smsgate-device7 is an SMS gateway connected to machine minioncontroller3 via a serial port.

Because of the way pillar works, each of the salt-proxy processes that fork off the proxy minions will only see the keys specific to the proxies it will be handling.

Proxies can be configured in /etc/salt/proxy or with files in /etc/salt/proxy.d as of Salt's 2016.11.0 release.

Also, in general, proxy-minions are lightweight, so the machines that run them could conceivably control a large number of devices. To run more than one proxy from a single machine, simply start an additional proxy process with `--proxyid` set to the id to which you want the proxy to bind. It is possible for the proxy services to be spread across many machines if necessary, or intentionally run on machines that need to control devices because of some physical interface (e.g. i2c and serial above). Another reason to divide proxy services might be security. In more secure environments only certain machines may have a network path to certain devices.

15.6.2 Proxymodules

A proxy module encapsulates all the code necessary to interface with a device. Proxymodules are located inside the salt.proxy module, or can be placed in the `_proxy` directory in your file_roots (default is `/srv/salt/_proxy`. At a minimum a proxymodule object must implement the following functions:

`__virtual__()`: This function performs the same duty that it does for other types of Salt modules. Logic goes here to determine if the module can be loaded, checking for the presence of Python modules on which the proxy depends. Returning `False` will prevent the module from loading.

`init(opts)`: Perform any initialization that the device needs. This is a good place to bring up a persistent connection to a device, or authenticate to create a persistent authorization token.

`initialized()`: Returns True if `init()` was successfully called.

`shutdown()`: Code to cleanly shut down or close a connection to a controlled device goes here. This function must exist, but can contain only the keyword `pass` if there is no shutdown logic required.

`ping()`: While not required, it is highly recommended that this function also be defined in the proxymodule. The code for `ping` should contact the controlled device and make sure it is really available.

`alive(opts)`: Another optional function, it is used together with the `proxy_keep_alive` option (default: `True`). This function should return a boolean value corresponding to the state of the connection. If the connection is down, will try to restart (`shutdown` followed by `init`). The polling frequency is controlled using the `proxy_keep_alive_interval` option, in minutes.

`grains()`: Rather than including grains in /srv/salt/_grains or in the standard install directories for grains, grains can be computed and returned by this function. This function will be called automatically if `proxy_merge_grains_in_module` is set to `True` in /etc/salt/proxy. This variable defaults to `True` in the release code-named *2017.7.0*.

Pre 2015.8 the proxymodule also must have an `id()` function. 2015.8 and following don't use this function because the proxy's id is required on the command line.

Here is an example proxymodule used to interface to a *very* simple REST server. Code for the server is in the salt-contrib GitHub repository.

This proxymodule enables "service" enumeration, starting, stopping, restarting, and status; "package" installation, and a ping.

```
# -*- coding: utf-8 -*-
'''
This is a simple proxy-minion designed to connect to and communicate with
the bottle-based web service contained in https://github.com/saltstack/salt-contrib/
↪tree/master/proxyminion_rest_example
'''
from __future__ import absolute_import

# Import python libs
import logging
import salt.utils.http

HAS_REST_EXAMPLE = True

# This must be present or the Salt loader won't load this module
__proxyenabled__ = ['rest_sample']


# Variables are scoped to this module so we can have persistent data
# across calls to fns in here.
GRAINS_CACHE = {}
DETAILS = {}

# Want logging!
log = logging.getLogger(__file__)


# This does nothing, it's here just as an example and to provide a log
# entry when the module is loaded.
def __virtual__():
    '''
    Only return if all the modules are available
    '''
    log.debug('rest_sample proxy __virtual__() called...')
    return True
```

(continues on next page)

(continued from previous page)

```
def _complicated_function_that_determines_if_alive():
    return True

# Every proxy module needs an 'init', though you can
# just put DETAILS['initialized'] = True here if nothing
# else needs to be done.

def init(opts):
    log.debug('rest_sample proxy init() called...')
    DETAILS['initialized'] = True

    # Save the REST URL
    DETAILS['url'] = opts['proxy']['url']

    # Make sure the REST URL ends with a '/'
    if not DETAILS['url'].endswith('/'):
        DETAILS['url'] += '/'

def alive(opts):
    '''
    This function returns a flag with the connection state.
    It is very useful when the proxy minion establishes the communication
    via a channel that requires a more elaborated keep-alive mechanism, e.g.
    NETCONF over SSH.
    '''
    log.debug('rest_sample proxy alive() called...')
    return _complicated_function_that_determines_if_alive()


def initialized():
    '''
    Since grains are loaded in many different places and some of those
    places occur before the proxy can be initialized, return whether
    our init() function has been called
    '''
    return DETAILS.get('initialized', False)


def grains():
    '''
    Get the grains from the proxied device
    '''
    if not DETAILS.get('grains_cache', {}):
        r = salt.utils.http.query(DETAILS['url']+'info', decode_type='json',
↪decode=True)
        DETAILS['grains_cache'] = r['dict']
    return DETAILS['grains_cache']


def grains_refresh():
    '''
    Refresh the grains from the proxied device
    '''
    DETAILS['grains_cache'] = None
    return grains()
```

(continues on next page)

(continued from previous page)

```
def fns():
    return {'details': 'This key is here because a function in '
                      'grains/rest_sample.py called fns() here in the proxymodule.'}


def service_start(name):
    '''
    Start a "service" on the REST server
    '''
    r = salt.utils.http.query(DETAILS['url']+'service/start/'+name, decode_type='json
↪', decode=True)
    return r['dict']


def service_stop(name):
    '''
    Stop a "service" on the REST server
    '''
    r = salt.utils.http.query(DETAILS['url']+'service/stop/'+name, decode_type='json',
↪ decode=True)
    return r['dict']


def service_restart(name):
    '''
    Restart a "service" on the REST server
    '''
    r = salt.utils.http.query(DETAILS['url']+'service/restart/'+name, decode_type=
↪'json', decode=True)
    return r['dict']


def service_list():
    '''
    List "services" on the REST server
    '''
    r = salt.utils.http.query(DETAILS['url']+'service/list', decode_type='json',⊠
↪decode=True)
    return r['dict']


def service_status(name):
    '''
    Check if a service is running on the REST server
    '''
    r = salt.utils.http.query(DETAILS['url']+'service/status/'+name, decode_type='json
↪', decode=True)
    return r['dict']


def package_list():
    '''
    List "packages" installed on the REST server
    '''
    r = salt.utils.http.query(DETAILS['url']+'package/list', decode_type='json',⊠
↪decode=True)
```

(continues on next page)

(continued from previous page)

```
    return r['dict']


def package_install(name, **kwargs):
    '''
    Install a "package" on the REST server
    '''
    cmd = DETAILS['url']+'package/install/'+name
    if kwargs.get('version', False):
        cmd += '/'+kwargs['version']
    else:
        cmd += '/1.0'
    r = salt.utils.http.query(cmd, decode_type='json', decode=True)
    return r['dict']


def fix_outage():
    r = salt.utils.http.query(DETAILS['url']+'fix_outage')
    return r


def uptodate(name):

    '''
    Call the REST endpoint to see if the packages on the "server" are up to date.
    '''
    r = salt.utils.http.query(DETAILS['url']+'package/remove/'+name, decode_type='json
↪', decode=True)
    return r['dict']


def package_remove(name):

    '''
    Remove a "package" on the REST server
    '''
    r = salt.utils.http.query(DETAILS['url']+'package/remove/'+name, decode_type='json
↪', decode=True)
    return r['dict']


def package_status(name):
    '''
    Check the installation status of a package on the REST server
    '''
    r = salt.utils.http.query(DETAILS['url']+'package/status/'+name, decode_type='json
↪', decode=True)
    return r['dict']


def ping():
    '''
    Is the REST server up?
    '''
    r = salt.utils.http.query(DETAILS['url']+'ping', decode_type='json', decode=True)
    try:
```

(continues on next page)

(continued from previous page)

```
        return r['dict'].get('ret', False)
    except Exception:
        return False


def shutdown(opts):
    '''
    For this proxy shutdown is a no-op
    '''
    log.debug('rest_sample proxy shutdown() called...')
```

Grains are data about minions. Most proxied devices will have a paltry amount of data as compared to a typical Linux server. By default, a proxy minion will have several grains taken from the host. Salt core code requires values for `kernel`, `os`, and `os_family`–all of these are forced to be `proxy` for proxy-minions.

To add others to your proxy minion for a particular device, create a file in salt/grains named [proxytype].py and place inside it the different functions that need to be run to collect the data you are interested in. Here's an example. Note the function below called `proxy_functions`. It demonstrates how a grains function can take a single argument, which will be set to the value of `__proxy__`. Dunder variables are not yet injected into Salt processes at the time grains are loaded, so this enables us to get a handle to the proxymodule so we can cross-call the functions therein used to commmunicate with the controlled device.

Note that as of 2016.3, grains values can also be calculated in a function called `grains()` in the proxymodule itself. This might be useful if a proxymodule author wants to keep all the code for the proxy interface in the same place instead of splitting it between the proxy and grains directories.

This function will only be called automatically if the configuration variable `proxy_merge_grains_in_module` is set to True in the proxy configuration file (default `/etc/salt/proxy`). This variable defaults to `True` in the release code-named *2017.7.0*.

15.7 The __proxyenabled__ directive

In previous versions of Salt the `__proxyenabled__` directive controlled loading of all Salt modules for proxies (e.g. grains, execution modules, state modules). From 2016.3 on, the only modules that respect `__proxyenabled__` are grains and proxy modules. These modules need to be told which proxy they work with.

`__proxyenabled__` is a list, and can contain a single '*' to indicate a grains module works with all proxies.

Example from `salt/grains/rest_sample.py`:

```
# -*- coding: utf-8 -*-
'''
Generate baseline proxy minion grains
'''
from __future__ import absolute_import
import salt.utils.platform

__proxyenabled__ = ['rest_sample']

__virtualname__ = 'rest_sample'

def __virtual__():
    try:
        if salt.utils.platform.is_proxy() and __opts__['proxy']['proxytype'] == 'rest_
↪sample':
```

(continues on next page)

(continued from previous page)

```
            return __virtualname__
    except KeyError:
        pass

    return False
```

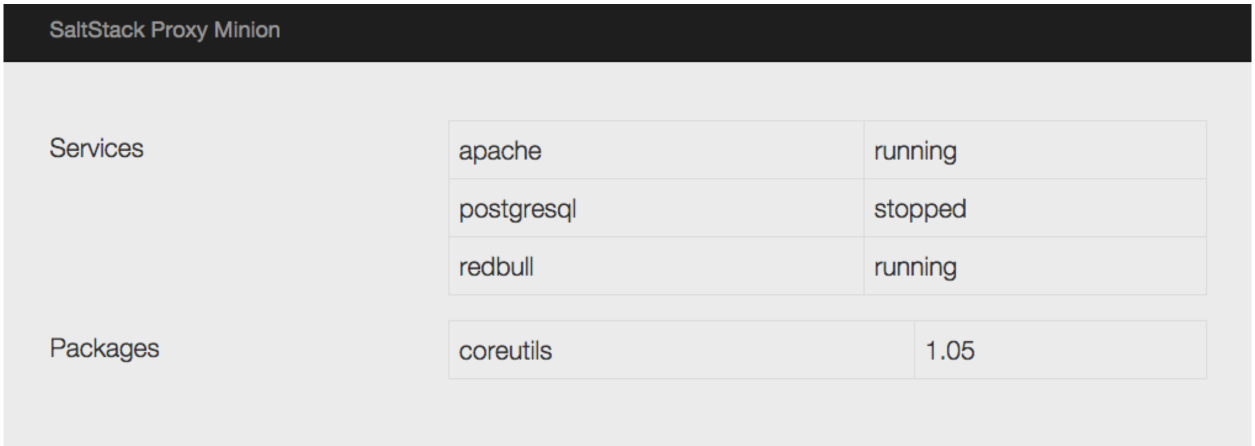
15.7.1 Salt Proxy Minion End-to-End Example

The following is walkthrough that documents how to run a sample REST service and configure one or more proxy minions to talk to and control it.

1. Ideally, create a Python virtualenv in which to run the REST service. This is not strictly required, but without a virtualenv you will need to install `bottle` via pip globally on your system
2. Clone https://github.com/saltstack/salt-contrib and copy the contents of the directory `proxyminion_rest_example` somewhere on a machine that is reachable from the machine on which you want to run the salt-proxy. This machine needs Python 2.7 or later.
3. Install bottle version 0.12.8 via pip or easy_install

```
pip install bottle==0.12.8
```

4. Run `python rest.py --help` for usage
5. Start the REST API on an appropriate port and IP.
6. Load the REST service's status page in your browser by going to the IP/port combination (e.g. http://127.0.0.1:8000)
7. You should see a page entitled "Salt Proxy Minion" with two sections, one for "services" and one for "packages" and you should see a log entry in the terminal where you started the REST process indicating that the index page was retrieved.



SaltStack Proxy Minion

Services	apache	running
	postgresql	stopped
	redbull	running
Packages	coreutils	1.05

Now, configure your salt-proxy.

1. Edit `/etc/salt/proxy` and add an entry for your master's location

```
master: localhost
```

2. On your salt-master, ensure that pillar is configured properly. Select an ID for your proxy (in this example we will name the proxy with the letter 'p' followed by the port the proxy is answering on). In your pillar topfile, place an entry for your proxy:

```
base:
  'p8000':
    - p8000
```

This says that Salt's pillar should load some values for the proxy `p8000` from the file `/srv/pillar/p8000.sls` (if you have not changed your default pillar_roots)

3. In the pillar root for your base environment, create the `p8000.sls` file with the following contents:

```
proxy:
  proxytype: rest_sample
  url: http://<IP your REST listens on>:port
```

In other words, if your REST service is listening on port 8000 on 127.0.0.1 the 'url' key above should say `url: http://127.0.0.1:8000`

4. Make sure your salt-master is running.
5. Start the salt-proxy in debug mode

```
salt-proxy --proxyid=p8000 -l debug
```

6. Accept your proxy's key on your salt-master

```
salt-key -y -a p8000
The following keys are going to be accepted:
Unaccepted Keys:
p8000
Key for minion p8000 accepted.
```

7. Now you should be able to ping your proxy. When you ping, you should see a log entry in the terminal where the REST service is running.

```
salt p8000 test.version
```

8. The REST service implements a degenerately simple pkg and service provider as well as a small set of grains. To "install" a package, use a standard `pkg.install`. If you pass '==' and a verrsion number after the package name then the service will parse that and accept that as the package's version.
9. Try running `salt p8000 grains.items` to see what grains are available. You can target proxies via grains if you like.
10. You can also start and stop the available services (apache, redbull, and postgresql with `service.start`, etc.
11. States can be written to target the proxy. Feel free to experiment with them.

15.8 SSH Proxymodules

See above for a general introduction to writing proxy modules. All of the guidelines that apply to REST are the same for SSH. This sections specifically talks about the SSH proxy module and explains the working of the example proxy module `ssh_sample`.

Here is a simple example proxymodule used to interface to a device over SSH. Code for the SSH shell is in the salt-contrib GitHub repository.

This proxymodule enables "package" installation.

```
# -*- coding: utf-8 -*-
'''
This is a simple proxy-minion designed to connect to and communicate with
a server that exposes functionality via SSH.
This can be used as an option when the device does not provide
an api over HTTP and doesn't have the python stack to run a minion.
'''
from __future__ import absolute_import

# Import python libs
import salt.utils.json
import logging

# Import Salt's libs
from salt.utils.vt_helper import SSHConnection
from salt.utils.vt import TerminalException

# This must be present or the Salt loader won't load this module
__proxyenabled__ = ['ssh_sample']

DETAILS = {}

# Want logging!
log = logging.getLogger(__file__)


# This does nothing, it's here just as an example and to provide a log
# entry when the module is loaded.
def __virtual__():
    '''
    Only return if all the modules are available
    '''
    log.info('ssh_sample proxy __virtual__() called...')

    return True


def init(opts):
    '''
    Required.
    Can be used to initialize the server connection.
    '''
    try:
        DETAILS['server'] = SSHConnection(host=__opts__['proxy']['host'],
                                          username=__opts__['proxy']['username'],
                                          password=__opts__['proxy']['password'])
        # connected to the SSH server
        out, err = DETAILS['server'].sendline('help')

    except TerminalException as e:
        log.error(e)
        return False


def shutdown(opts):
    '''
    Disconnect
```

(continues on next page)

(continued from previous page)

```
    '''
    DETAILS['server'].close_connection()


def parse(out):
    '''
    Extract json from out.

    Parameter
        out: Type string. The data returned by the
        ssh command.
    '''
    jsonret = []
    in_json = False
    for ln_ in out.split('\n'):
        if '{' in ln_:
            in_json = True
        if in_json:
            jsonret.append(ln_)
        if '}' in ln_:
            in_json = False
    return salt.utils.json.loads('\n'.join(jsonret))


def package_list():
    '''
    List "packages" by executing a command via ssh
    This function is called in response to the salt command

    ..code-block::bash
        salt target_minion pkg.list_pkgs

    '''
    # Send the command to execute
    out, err = DETAILS['server'].sendline('pkg_list')

    # "scrape" the output and return the right fields as a dict
    return parse(out)


def package_install(name, **kwargs):
    '''
    Install a "package" on the REST server
    '''
    cmd = 'pkg_install ' + name
    if 'version' in kwargs:
        cmd += '/'+kwargs['version']
    else:
        cmd += '/1.0'

    # Send the command to execute
    out, err = DETAILS['server'].sendline(cmd)

    # "scrape" the output and return the right fields as a dict
    return parse(out)
```

(continues on next page)

(continued from previous page)

```
def package_remove(name):
    '''
    Remove a "package" on the REST server
    '''
    cmd = 'pkg_remove ' + name

    # Send the command to execute
    out, err = DETAILS['server'].sendline(cmd)

    # "scrape" the output and return the right fields as a dict
    return parse(out)
```

15.8.1 Connection Setup

The `init()` method is responsible for connection setup. It uses the `host`, `username` and `password` config variables defined in the pillar data. The `prompt` kwarg can be passed to `SSHConnection` if your SSH server's prompt differs from the example's prompt `(Cmd)`. Instantiating the `SSHConnection` class establishes an SSH connection to the ssh server (using Salt VT).

15.8.2 Command execution

The `package_*` methods use the SSH connection (established in `init()`) to send commands out to the SSH server. The `sendline()` method of `SSHConnection` class can be used to send commands out to the server. In the above example we send commands like `pkg_list` or `pkg_install`. You can send any SSH command via this utility.

15.8.3 Output parsing

Output returned by `sendline()` is a tuple of strings representing the stdout and the stderr respectively. In the toy example shown we simply scrape the output and convert it to a python dictionary, as shown in the `parse` method. You can tailor this method to match your parsing logic.

15.8.4 Connection teardown

The `shutdown` method is responsible for calling the `close_connection()` method of `SSHConnection` class. This ends the SSH connection to the server.

For more information please refer to class SSHConnection.

Salt Proxy Minion SSH End-to-End Example

The following is walkthrough that documents how to run a sample SSH service and configure one or more proxy minions to talk to and control it.

1. This walkthrough uses a custom SSH shell to provide an end to end example. Any other shells can be used too.
2. Setup the proxy command shell as shown https://github.com/saltstack/salt-contrib/tree/master/proxyminion_ssh_example

Now, configure your salt-proxy.

1. Edit `/etc/salt/proxy` and add an entry for your master's location

```
master: localhost
multiprocessing: False
```

2. On your salt-master, ensure that pillar is configured properly. Select an ID for your proxy (in this example we will name the proxy with the letter 'p' followed by the port the proxy is answering on). In your pillar topfile, place an entry for your proxy:

```
base:
  'p8000':
    - p8000
```

This says that Salt's pillar should load some values for the proxy `p8000` from the file `/srv/pillar/p8000.sls` (if you have not changed your default pillar_roots)

3. In the pillar root for your base environment, create the `p8000.sls` file with the following contents:

```
proxy:
  proxytype: ssh_sample
  host: saltyVM
  username: salt
  password: badpass
```

4. Make sure your salt-master is running.
5. Start the salt-proxy in debug mode

```
salt-proxy --proxyid=p8000 -l debug
```

6. Accept your proxy's key on your salt-master

```
salt-key -y -a p8000
The following keys are going to be accepted:
Unaccepted Keys:
p8000
Key for minion p8000 accepted.
```

7. Now you should be able to run commands on your proxy.

```
salt p8000 pkg.list_pkgs
```

8. The SSH shell implements a degenerately simple pkg. To "install" a package, use a standard `pkg.install`. If you pass '==' and a verrsion number after the package name then the service will parse that and accept that as the package's version.

New in version 2015.8.3.

Proxy Minion Beacon

The salt proxy beacon is meant to facilitate configuring multiple proxies on one or many minions. This should simplify configuring and managing multiple `salt-proxy` processes.

1. On your salt-master, ensure that pillar is configured properly. Select an ID for your proxy (in this example we will name the proxy 'p8000'). In your pillar topfile, place an entry for your proxy:

```
base:
  'p8000':
    - p8000
```

This says that Salt's pillar should load some values for the proxy `p8000` from the file `/srv/pillar/p8000.sls` (if you have not changed your default pillar_roots)

2. In the pillar root for your base environment, create the `p8000.sls` file with the following contents:

```
proxy:
  # set proxytype for your proxymodule
  proxytype: ssh_sample
  host: saltyVM
  username: salt
  password: badpass
```

This should complete the proxy setup for `p8000`

3. Configure the `salt_proxy` beacon

```
beacons:
  salt_proxy:
    - proxies:
        p8000: {}
        p8001: {}
```

Once this beacon is configured it will automatically start the `salt-proxy` process. If the `salt-proxy` process is terminated the beacon will re-start it.

4. Accept your proxy's key on your salt-master

```
salt-key -y -a p8000
The following keys are going to be accepted:
Unaccepted Keys:
p8000
Key for minion p8000 accepted.
```

5. Now you should be able to run commands on your proxy.

```
salt p8000 pkg.list_pkgs
```

New in version 2015.8.2.

Proxy Minion States

Salt proxy state can be used to deploy, configure and run a `salt-proxy` instance on your minion. Configure proxy settings on the master side and the state configures and runs `salt-proxy` on the remote end.

1. On your salt-master, ensure that pillar is configured properly. Select an ID for your proxy (in this example we will name the proxy 'p8000'). In your pillar topfile, place an entry for your proxy:

```
base:
  'p8000':
    - p8000
```

This says that Salt's pillar should load some values for the proxy `p8000` from the file `/srv/pillar/p8000.sls` (if you have not changed your default pillar_roots)

2. In the pillar root for your base environment, create the `p8000.sls` file with the following contents:

```
proxy:
  # set proxytype for your proxymodule
  proxytype: ssh_sample
  host: saltyVM
  username: salt
  password: badpass
```

3. Create the following state in your state tree (let's name it salt_proxy.sls)

```
salt-proxy-configure:
  salt_proxy.configure_proxy:
    - proxyname: p8000
    - start: True # start the process if it isn't running
```

4. Make sure your salt-master and salt-minion are running.
5. Run the state salt_proxy on the minion where you want to run `salt-proxy`

Example using `state.sls` to configure and run `salt-proxy`

```
# salt device_minion state.sls salt_proxy
```

This starts salt-proxy on `device_minion`

6. Accept your proxy's key on your salt-master

```
salt-key -y -a p8000
The following keys are going to be accepted:
Unaccepted Keys:
p8000
Key for minion p8000 accepted.
```

7. Now you should be able to run commands on your proxy.

```
salt p8000 pkg.list_pkgs
```

CHAPTER

SIXTEEN

NETWORK AUTOMATION

Network automation is a continuous process of automating the configuration, management and operations of a computer network. Although the abstraction could be compared with the operations on the server side, there are many particular challenges, the most important being that a network device is traditionally closed hardware able to run proprietary software only. In other words, the user is not able to install the salt-minion package directly on a traditional network device. For these reasons, most network devices can be controlled only remotely via *proxy minions* or using the *Salt SSH*. However, there are also vendors producing whitebox equipment (e.g. Arista, Cumulus) or others that have moved the operating system in the container (e.g. Cisco NX-OS, Cisco IOS-XR), allowing the salt-minion to be installed directly on the platform.

16.1 New in Carbon (2016.11)

The methodologies for network automation have been introduced in *2016.11.0*. Network automation support is based on proxy minions.

- `NAPALM proxy`
- `Junos proxy`
- `Cisco NXOS`
- `Cisco NOS`

16.2 NAPALM

NAPALM (Network Automation and Programmability Abstraction Layer with Multivendor support) is an open-sourced Python library that implements a set of functions to interact with different router vendor devices using a unified API. Being vendor-agnostic simplifies operations, as the configuration and interaction with the network device does not rely on a particular vendor.

Beginning with 2017.7.0, the NAPALM modules have been transformed so they can run in both proxy and regular minions. That means, if the operating system allows, the salt-minion package can be installed directly on the network gear. The interface between the network operating system and Salt in that case would be the corresponding NAPALM sub-package.

For example, if the user installs the salt-minion on a Arista switch, the only requirement is napalm-eos.

The following modules are available in 2017.7.0:

- *NAPALM grains*
- *NET execution module* - Networking basic features
- *NTP execution module*
- *BGP execution module*
- *Routes execution module*
- *SNMP execution module*
- *Users execution module*
- *Probes execution module*
- *NTP peers management state*
- *SNMP configuration management state*
- *Users management state*
- *Netconfig state module* - Manage the configuration of network devices using arbitrary templates and the Salt-specific advanced templating methodologies.
- `Network ACL execution module` - Generate and load ACL (firewall) configuration on network devices.

- `Network ACL state` - Manage the firewall configuration. It only requires writing the pillar structure correctly!
- `NAPALM YANG execution module` - Parse, generate and load native device configuration in a standard way, using the OpenConfig/IETF models. This module contains also helpers for the states.
- `NAPALM YANG state module` - Manage the network device configuration according to the YANG models (OpenConfig or IETF).
- `NET finder` - Runner to find details easily and fast. It's smart enough to know what you are looking for. It will search in the details of the network interfaces, IP addresses, MAC address tables, ARP tables and LLDP neighbors.
- `BGP finder` - Runner to search BGP neighbors details.
- `NAPALM syslog` - Engine to import events from the napalm-logs library into the Salt event bus. The events are based on the syslog messages from the network devices and structured following the OpenConfig/IETF YANG models.
- `NAPALM Helpers` - Generic helpers for NAPALM-related operations. For example, the `Compliance report` function can be used inside the state modules to compare the expected and the existing configuration.

16.2.1 Getting started

Install NAPALM - follow the notes and check the platform-specific dependencies.

Salt's Pillar system is ideally suited for configuring proxy-minions (though they can be configured in /etc/salt/proxy as well). Proxies can either be designated via a pillar file in `pillar_roots`, or through an external pillar. External pillars afford the opportunity for interfacing with a configuration management system, database, or other knowledgeable system that may already contain all the details of proxy targets. To use static files in `pillar_roots`, pattern your files after the following examples:

`/etc/salt/pillar/top.sls`

```
base:
  router1:
    - router1
  router2:
    - router2
  switch1:
    - switch1
  switch2:
    - switch2
  cpe1:
    - cpe1
```

`/etc/salt/pillar/router1.sls`

```
proxy:
  proxytype: napalm
  driver: junos
  host: r1.bbone.as1234.net
  username: my_username
  password: my_password
```

`/etc/salt/pillar/router2.sls`

```
proxy:
  proxytype: napalm
  driver: iosxr
  host: r2.bbone.as1234.net
  username: my_username
  password: my_password
  optional_args:
    port: 22022
```

/etc/salt/pillar/switch1.sls

```
proxy:
  proxytype: napalm
  driver: eos
  host: sw1.bbone.as1234.net
  username: my_username
  password: my_password
  optional_args:
    enable_password: my_secret
```

/etc/salt/pillar/switch2.sls

```
proxy:
  proxytype: napalm
  driver: nxos
  host: sw2.bbone.as1234.net
  username: my_username
  password: my_password
```

/etc/salt/pillar/cpe1.sls

```
proxy:
  proxytype: napalm
  driver: ios
  host: cpe1.edge.as1234.net
  username: ''
  password: ''
  optional_args:
    use_keys: True
    auto_rollback_on_error: True
```

16.2.2 CLI examples

Display the complete running configuration on `router1`:

```
$ sudo salt 'router1' net.config source='running'
```

Retrieve the NTP servers configured on all devices:

```
$ sudo salt '*' ntp.servers
router1:
  ----------
  comment:
  out:
      - 1.2.3.4
```

(continues on next page)

(continued from previous page)

```
  result:
      True
cpe1:
  ----------
  comment:
  out:
      - 1.2.3.4
  result:
      True
switch2:
  ----------
  comment:
  out:
      - 1.2.3.4
  result:
      True
router2:
  ----------
  comment:
  out:
      - 1.2.3.4
  result:
      True
switch1:
  ----------
  comment:
  out:
      - 1.2.3.4
  result:
      True
```

Display the ARP tables on all Cisco devices running IOS-XR 5.3.3:

```
$ sudo salt -G 'os:iosxr and version:5.3.3' net.arp
```

Return operational details for interfaces from Arista switches:

```
$ sudo salt -C 'sw* and os:eos' net.interfaces
```

Execute traceroute from the edge of the network:

```
$ sudo salt 'router*' net.traceroute 8.8.8.8 vrf='CUSTOMER1-VRF'
```

Verbatim display from the CLI of Juniper routers:

```
$ sudo salt -C 'router* and G@os:junos' net.cli 'show version and haiku'
```

Retrieve the results of the RPM probes configured on Juniper MX960 routers:

```
$ sudo salt -C 'router* and G@os:junos and G@model:MX960' probes.results
```

Return the list of configured users on the CPEs:

```
$ sudo salt 'cpe*' users.config
```

Using the *BGP finder*, return the list of BGP neighbors that are down:

```
$ sudo salt-run bgp.neighbors up=False
```

Using the `NET finder`, determine the devices containing the pattern "PX-1234-LHR" in their interface description:

```
$ sudo salt-run net.find PX-1234-LHR
```

16.2.3 Cross-platform configuration management example: NTP

Assuming that the user adds the following two lines under `file_roots`:

```
file_roots:
  base:
    - /etc/salt/pillar/
    - /etc/salt/templates/
    - /etc/salt/states/
```

Define the list of NTP peers and servers wanted:

`/etc/salt/pillar/ntp.sls`

```
ntp.servers:
  - 1.2.3.4
  - 5.6.7.8
ntp.peers:
   - 10.11.12.13
   - 14.15.16.17
```

Include the new file: for example, if we want to have the same NTP servers on all network devices, we can add the following line inside the `top.sls` file:

```
'*':
  - ntp
```

`/etc/salt/pillar/top.sls`

```
base:
  '*':
    - ntp
  router1:
    - router1
  router2:
    - router2
  switch1:
    - switch1
  switch2:
    - switch2
  cpe1:
    - cpe1
```

Or include only where needed:

`/etc/salt/pillar/top.sls`

```
base:
  router1:
    - router1
```

(continues on next page)

(continued from previous page)

```
    - ntp
  router2:
    - router2
    - ntp
  switch1:
    - switch1
  switch2:
    - switch2
  cpe1:
    - cpe1
```

Define the cross-vendor template:

/etc/salt/templates/ntp.jinja

```
{%- if grains.vendor|lower == 'cisco' %}
  no ntp
  {%- for server in servers %}
  ntp server {{ server }}
  {%- endfor %}
  {%- for peer in peers %}
  ntp peer {{ peer }}
  {%- endfor %}
{%- elif grains.os|lower == 'junos' %}
  system {
    replace:
    ntp {
      {%- for server in servers %}
      server {{ server }};
      {%- endfor %}
      {%- for peer in peers %}
      peer {{ peer }};
      {%- endfor %}
    }
  }
{%- endif %}
```

Define the SLS state file, making use of the `Netconfig state module`:

/etc/salt/states/router/ntp.sls

```
ntp_config_example:
  netconfig.managed:
    - template_name: salt://ntp.jinja
    - peers: {{ pillar.get('ntp.peers', []) | json }}
    - servers: {{ pillar.get('ntp.servers', []) | json }}
```

Run the state and assure NTP configuration consistency across your multi-vendor network:

```
$ sudo salt 'router*' state.sls router.ntp
```

Besides CLI, the state can be scheduled or executed when triggered by a certain event.

16.3 JUNOS

Juniper has developed a Junos specific proxy infrastructure which allows remote execution and configuration management of Junos devices without having to install SaltStack on the device. The infrastructure includes:

- *Junos proxy*
- *Junos execution module*
- *Junos state module*
- *Junos syslog engine*

The execution and state modules are implemented using junos-eznc (PyEZ). Junos PyEZ is a microframework for Python that enables you to remotely manage and automate devices running the Junos operating system.

16.3.1 Getting started

Install PyEZ on the system which will run the Junos proxy minion. It is required to run Junos specific modules.

```
pip install junos-eznc
```

Next, set the master of the proxy minions.

/etc/salt/proxy

```
master: <master_ip>
```

Add the details of the Junos device. Device details are usually stored in salt pillars. If the you do not wish to store credentials in the pillar, one can setup passwordless ssh.

/srv/pillar/vmx_details.sls

```
proxy:
  proxytype: junos
  host: <hostip>
  username: user
  passwd: secret123
```

Map the pillar file to the proxy minion. This is done in the top file.

/srv/pillar/top.sls

```
base:
  vmx:
    - vmx_details
```

Note: Before starting the Junos proxy make sure that netconf is enabled on the Junos device. This can be done by adding the following configuration on the Junos device.

```
set system services netconf ssh
```

Start the salt master.

```
salt-master -l debug
```

Then start the salt proxy.

```
salt-proxy --proxyid=vmx -l debug
```

Once the master and junos proxy minion have started, we can run execution and state modules on the proxy minion. Below are few examples.

16.3.2 CLI examples

For detailed documentation of all the junos execution modules refer: *`Junos execution module`*

Display device facts.

```
$ sudo salt 'vmx' junos.facts
```

Refresh the Junos facts. This function will also refresh the facts which are stored in salt grains. (Junos proxy stores Junos facts in the salt grains)

```
$ sudo salt 'vmx' junos.facts_refresh
```

Call an RPC.

```
$ sudo salt 'vmx' junos.rpc 'get-interface-information' '/var/log/interface-info.txt' terse=True
```

Install config on the device.

```
$ sudo salt 'vmx' junos.install_config 'salt://my_config.set'
```

Shutdown the junos device.

```
$ sudo salt 'vmx' junos.shutdown shutdown=True in_min=10
```

16.3.3 State file examples

For detailed documentation of all the junos state modules refer: *`Junos state module`*

Executing an RPC on Junos device and storing the output in a file.

`/srv/salt/rpc.sls`

```
get-interface-information:
    junos:
      - rpc
      - dest: /home/user/rpc.log
      - interface_name: lo0
```

Lock the junos device, load the configuration, commit it and unlock the device.

`/srv/salt/load.sls`

```
lock the config:
  junos.lock

salt://configs/my_config.set:
  junos:
    - install_config
```

(continues on next page)

(continued from previous page)

```
    - timeout: 100
    - diffs_file: 'var/log/diff'

commit the changes:
  junos:
    - commit

unlock the config:
  junos.unlock
```

According to the device personality install appropriate image on the device.

/srv/salt/image_install.sls

```
{% if grains['junos_facts']['personality'] == MX %}
salt://images/mx_junos_image.tgz:
  junos:
    - install_os
    - timeout: 100
    - reboot: True
{% elif grains['junos_facts']['personality'] == EX %}
salt://images/ex_junos_image.tgz:
  junos:
    - install_os
    - timeout: 150
{% elif grains['junos_facts']['personality'] == SRX %}
salt://images/srx_junos_image.tgz:
  junos:
    - install_os
    - timeout: 150
{% endif %}
```

16.3.4 Junos Syslog Engine

Junos Syslog Engine is a Salt engine which receives data from various Junos devices, extracts event information and forwards it on the master/minion event bus. To start the engine on the salt master, add the following configuration in the master config file. The engine can also run on the salt minion.

/etc/salt/master

```
engines:
  - junos_syslog:
      port: xxx
```

For junos_syslog engine to receive events, syslog must be set on the Junos device. This can be done via following configuration:

```
set system syslog host <ip-of-the-salt-device> port xxx any any
```

CHAPTER

SEVENTEEN

SALT VIRT

The Salt Virt cloud controller capability was initially added to Salt in version 0.14.0 as an alpha technology.

The initial Salt Virt system supports core cloud operations:

- Virtual machine deployment
- Inspection of deployed VMs
- Virtual machine migration
- Network profiling
- Automatic VM integration with all aspects of Salt
- Image Pre-seeding

Many features are currently under development to enhance the capabilities of the Salt Virt systems.

Note: It is noteworthy that Salt was originally developed with the intent of using the Salt communication system as the backbone to a cloud controller. This means that the Salt Virt system is not an afterthought, simply a system that took the back seat to other development. The original attempt to develop the cloud control aspects of Salt was a project called butter. This project never took off, but was functional and proves the early viability of Salt to be a cloud controller.

Warning: Salt Virt does not work with KVM that is running in a VM. KVM must be running on the base hardware.

17.1 Salt Virt Tutorial

A tutorial about how to get Salt Virt up and running has been added to the tutorial section:

Cloud Controller Tutorial

17.2 The Salt Virt Runner

The point of interaction with the cloud controller is the **virt** runner. The **virt** runner comes with routines to execute specific virtual machine routines.

Reference documentation for the virt runner is available with the runner module documentation:

Virt Runner Reference

17.3 Based on Live State Data

The Salt Virt system is based on using Salt to query live data about hypervisors and then using the data gathered to make decisions about cloud operations. This means that no external resources are required to run Salt Virt, and that the information gathered about the cloud is live and accurate.

17.4 Deploy from Network or Disk

17.4.1 Virtual Machine Disk Profiles

Salt Virt allows for the disks created for deployed virtual machines to be finely configured. The configuration is a simple data structure which is read from the `config.option` function, meaning that the configuration can be stored in the minion config file, the master config file, or the minion's pillar.

This configuration option is called `virt.disk`. The default `virt.disk` data structure looks like this:

```
virt.disk:
  default:
    - system:
      size: 8192
      format: qcow2
      model: virtio
```

Note: The format and model does not need to be defined, Salt will default to the optimal format used by the underlying hypervisor, in the case of kvm this is **qcow2** and **virtio**.

This configuration sets up a disk profile called default. The default profile creates a single system disk on the virtual machine.

Define More Profiles

Many environments will require more complex disk profiles and may require more than one profile, this can be easily accomplished:

```
virt.disk:
  default:
    - system:
        size: 8192
  database:
    - system:
        size: 8192
    - data:
        size: 30720
  web:
    - system:
        size: 1024
    - logs:
        size: 5120
```

This configuration allows for one of three profiles to be selected, allowing virtual machines to be created with different storage needs of the deployed vm.

17.4.2 Virtual Machine Network Profiles

Salt Virt allows for the network devices created for deployed virtual machines to be finely configured. The configuration is a simple data structure which is read from the `config.option` function, meaning that the configuration can be stored in the minion config file, the master config file, or the minion's pillar.

This configuration option is called `virt:nic`. By default the `virt:nic` option is empty but defaults to a data structure which looks like this:

```
virt:
  nic:
    default:
      eth0:
        bridge: br0
        model: virtio
```

Note: The model does not need to be defined, Salt will default to the optimal model used by the underlying hypervisor, in the case of kvm this model is **virtio**

This configuration sets up a network profile called default. The default profile creates a single Ethernet device on the virtual machine that is bridged to the hypervisor's **br0** interface. This default setup does not require setting up the `virt:nic` configuration, and is the reason why a default install only requires setting up the **br0** bridge device on the hypervisor.

Define More Profiles

Many environments will require more complex network profiles and may require more than one profile, this can be easily accomplished:

```
virt:
  nic:
    dual:
      eth0:
        bridge: service_br
      eth1:
        bridge: storage_br
    single:
      eth0:
        bridge: service_br
    triple:
      eth0:
        bridge: service_br
      eth1:
        bridge: storage_br
      eth2:
        bridge: dmz_br
    all:
      eth0:
        bridge: service_br
      eth1:
```

(continues on next page)

(continued from previous page)

```
        bridge: storage_br
      eth2:
        bridge: dmz_br
      eth3:
        bridge: database_br
    dmz:
      eth0:
        bridge: service_br
      eth1:
        bridge: dmz_br
    database:
      eth0:
        bridge: service_br
      eth1:
        bridge: database_br
```

This configuration allows for one of six profiles to be selected, allowing virtual machines to be created which attach to different network depending on the needs of the deployed vm.

CHAPTER

EIGHTEEN

COMMAND LINE REFERENCE

18.1 salt-call

18.1.1 `salt-call`

Synopsis

```
salt-call [options]
```

Description

The salt-call command is used to run module functions locally on a minion instead of executing them from the master. Salt-call is used to run a *Standalone Minion*, and was originally created for *troubleshooting*.

The Salt Master is contacted to retrieve state files and other resources during execution unless the `--local` option is specified.

Note: `salt-call` commands execute from the current user's shell context, while `salt` commands execute from the system's default context.

Options

`--version`
Print the version of Salt that is running.

`--versions-report`
Show program's dependencies and version number, and then exit

`-h`, `--help`
Show the help message and exit

`-c` `CONFIG_DIR`, **`--config-dir`**`=CONFIG_dir`
The location of the Salt configuration directory. This directory contains the configuration files for Salt master and minions. The default location on most systems is `/etc/salt`.

`--hard-crash`
Raise any original exception rather than exiting gracefully Default: False

`-g`, `--grains`
Return the information generated by the Salt grains

-m MODULE_DIRS, **--module-dirs**=MODULE_DIRS
Specify an additional directory to pull modules from. Multiple directories can be provided by passing -m /–module-dirs multiple times.

-d, --doc, --documentation
Return the documentation for the specified module or for all modules if none are specified

--master=MASTER
Specify the master to use. The minion must be authenticated with the master. If this option is omitted, the master options from the minion config will be used. If multi masters are set up the first listed master that responds will be used.

--return RETURNER
Set salt-call to pass the return data to one or many returner interfaces. To use many returner interfaces specify a comma delimited list of returners.

--local
Run salt-call locally, as if there was no master running.

--file-root=FILE_ROOT
Set this directory as the base file root.

--pillar-root=PILLAR_ROOT
Set this directory as the base pillar root.

--retcode-passthrough
Exit with the salt call retcode and not the salt binary retcode

--metadata
Print out the execution metadata as well as the return. This will print out the outputter data, the return code, etc.

--id=ID
Specify the minion id to use. If this option is omitted, the id option from the minion config will be used.

--skip-grains
Do not load grains.

--refresh-grains-cache
Force a refresh of the grains cache

Logging Options

Logging options which override any settings defined on the configuration files.

-l LOG_LEVEL, **--log-level**=LOG_LEVEL
Console logging log level. One of `all`, `garbage`, `trace`, `debug`, `info`, `warning`, `error`, `quiet`. Default: `warning`.

--log-file=LOG_FILE
Log file path. Default: /var/log/salt/minion.

--log-file-level=LOG_LEVEL_LOGFILE
Logfile logging log level. One of `all`, `garbage`, `trace`, `debug`, `info`, `warning`, `error`, `quiet`. Default: `warning`.

Output Options

`--out`

Pass in an alternative outputter to display the return of data. This outputter can be any of the available outputters:

> `highstate`, `json`, `key`, `overstatestage`, `pprint`, `raw`, `txt`, `yaml`, and *many others*.

Some outputters are formatted only for data returned from specific functions. If an outputter is used that does not support the data passed into it, then Salt will fall back on the `pprint` outputter and display the return data using the Python `pprint` standard library module.

`--out-indent` `OUTPUT_INDENT`**`,`** **`--output-indent`** `OUTPUT_INDENT`

Print the output indented by the provided value in spaces. Negative values disable indentation. Only applicable in outputters that support indentation.

`--out-file``=OUTPUT_FILE`**`,`** **`--output-file`**`=OUTPUT_FILE`

Write the output to the specified file.

`--out-file-append, --output-file-append`

Append the output to the specified file.

`--no-color`

Disable all colored output

`--force-color`

Force colored output

Note: When using colored output the color codes are as follows:

`green` denotes success, `red` denotes failure, `blue` denotes changes and success and `yellow` denotes a expected future change in configuration.

`--state-output``=STATE_OUTPUT`**`,`** **`--state_output`**`=STATE_OUTPUT`

Override the configured state_output value for minion output. One of ’full’, ’terse’, ’mixed’, ’changes’ or ’filter’. Default: ’none’.

`--state-verbose``=STATE_VERBOSE`**`,`** **`--state_verbose`**`=STATE_VERBOSE`

Override the configured state_verbose value for minion output. Set to True or False. Default: none.

See also

`salt(1)` *`salt-master(1)`* *`salt-minion(1)`*

18.2 salt

18.2.1 `salt`

Synopsis

> salt ’*’ [options] sys.doc
>
> salt -E ’.*’ [options] sys.doc cmd
>
> salt -G ’os:Arch.*’ [options] test.version

salt -C 'G@os:Arch.* and webserv* or G@kernel:FreeBSD' [options] test.version

Description

Salt allows for commands to be executed across a swath of remote systems in parallel. This means that remote systems can be both controlled and queried with ease.

Options

`--version`
Print the version of Salt that is running.

`--versions-report`
Show program's dependencies and version number, and then exit

`-h`, **`--help`**
Show the help message and exit

`-c` `CONFIG_DIR`, **`--config-dir`**`=CONFIG_dir`
The location of the Salt configuration directory. This directory contains the configuration files for Salt master and minions. The default location on most systems is `/etc/salt`.

`-t` `TIMEOUT`, **`--timeout`**`=TIMEOUT`
The timeout in seconds to wait for replies from the Salt minions. The timeout number specifies how long the command line client will wait to query the minions and check on running jobs. Default: 5

`-s`, **`--static`**
By default as of version 0.9.8 the salt command returns data to the console as it is received from minions, but previous releases would return data only after all data was received. Use the static option to only return the data with a hard timeout and after all minions have returned. Without the static option, you will get a separate JSON string per minion which makes JSON output invalid as a whole.

`--async`
Instead of waiting for the job to run on minions only print the job id of the started execution and complete.

`--subset``=SUBSET`
Execute the routine on a random subset of the targeted minions. The minions will be verified that they have the named function before executing. The SUBSET argument is the count of the minions to target.

`-v` `VERBOSE`, **`--verbose`**
Turn on verbosity for the salt call, this will cause the salt command to print out extra data like the job id.

`--hide-timeout`
Instead of showing the return data for all minions. This option prints only the online minions which could be reached.

`-b` `BATCH`, **`--batch-size`**`=BATCH`
Instead of executing on all targeted minions at once, execute on a progressive set of minions. This option takes an argument in the form of an explicit number of minions to execute at once, or a percentage of minions to execute on.

`-a` `EAUTH`, **`--auth`**`=EAUTH`
Pass in an external authentication medium to validate against. The credentials will be prompted for. The options are *auto*, *keystone*, *ldap*, and *pam*. Can be used with the -T option.

`-T`, **`--make-token`**
Used in conjunction with the -a option. This creates a token that allows for the authenticated user to send commands without needing to re-authenticate.

`--return``=RETURNER`
Choose an alternative returner to call on the minion, if an alternative returner is used then the return will not come back to the command line but will be sent to the specified return system. The options are *carbon*, *cassandra*, *couchbase*, *couchdb*, *elasticsearch*, *etcd*, *hipchat*, *local*, *local_cache*, *memcache*, *mongo*, *mysql*, *odbc*, *postgres*, *redis*, *sentry*, *slack*, *sms*, *smtp*, *sqlite3*, *syslog*, and *xmpp*.

`-d, --doc, --documentation`
Return the documentation for the module functions available on the minions

`--args-separator``=ARGS_SEPARATOR`
Set the special argument used as a delimiter between command arguments of compound commands. This is useful when one wants to pass commas as arguments to some of the commands in a compound command.

Logging Options

Logging options which override any settings defined on the configuration files.

`-l` `LOG_LEVEL`**`, --log-level`**`=LOG_LEVEL`
Console logging log level. One of `all`, `garbage`, `trace`, `debug`, `info`, `warning`, `error`, `quiet`. Default: `warning`.

`--log-file``=LOG_FILE`
Log file path. Default: /var/log/salt/master.

`--log-file-level``=LOG_LEVEL_LOGFILE`
Logfile logging log level. One of `all`, `garbage`, `trace`, `debug`, `info`, `warning`, `error`, `quiet`. Default: `warning`.

Target Selection

The default matching that Salt utilizes is shell-style globbing around the minion id. See https://docs.python.org/2/library/fnmatch.html#module-fnmatch.

`-E, --pcre`
The target expression will be interpreted as a PCRE regular expression rather than a shell glob.

`-L, --list`
The target expression will be interpreted as a comma-delimited list; example: server1.foo.bar,server2.foo.bar,example7.quo.qux

`-G, --grain`
The target expression matches values returned by the Salt grains system on the minions. The target expression is in the format of '<grain value>:<glob expression>'; example: 'os:Arch*'

This was changed in version 0.9.8 to accept glob expressions instead of regular expression. To use regular expression matching with grains, use the –grain-pcre option.

`--grain-pcre`
The target expression matches values returned by the Salt grains system on the minions. The target expression is in the format of '<grain value>:< regular expression>'; example: 'os:Arch.*'

`-N, --nodegroup`
Use a predefined compound target defined in the Salt master configuration file.

`-R, --range`
Instead of using shell globs to evaluate the target, use a range expression to identify targets. Range expressions look like %cluster.

Using the Range option requires that a range server is set up and the location of the range server is referenced in the master configuration file.

`-C, --compound`
Utilize many target definitions to make the call very granular. This option takes a group of targets separated by `and` or `or`. The default matcher is a glob as usual. If something other than a glob is used, preface it with the letter denoting the type; example: 'webserv* and G@os:Debian or E@db*' Make sure that the compound target is encapsulated in quotes.

`-I, --pillar`
Instead of using shell globs to evaluate the target, use a pillar value to identify targets. The syntax for the target is the pillar key followed by a glob expression: "role:production*"

`-S, --ipcidr`
Match based on Subnet (CIDR notation) or IPv4 address.

Output Options

`--out`
Pass in an alternative outputter to display the return of data. This outputter can be any of the available outputters:

> `highstate`, `json`, `key`, `overstatestage`, `pprint`, `raw`, `txt`, `yaml`, and *many others*.

Some outputters are formatted only for data returned from specific functions. If an outputter is used that does not support the data passed into it, then Salt will fall back on the `pprint` outputter and display the return data using the Python `pprint` standard library module.

`--out-indent` `OUTPUT_INDENT`, **`--output-indent`** `OUTPUT_INDENT`
Print the output indented by the provided value in spaces. Negative values disable indentation. Only applicable in outputters that support indentation.

`--out-file``=OUTPUT_FILE`, **`--output-file`**`=OUTPUT_FILE`
Write the output to the specified file.

`--out-file-append, --output-file-append`
Append the output to the specified file.

`--no-color`
Disable all colored output

`--force-color`
Force colored output

Note: When using colored output the color codes are as follows:

`green` denotes success, `red` denotes failure, `blue` denotes changes and success and `yellow` denotes a expected future change in configuration.

`--state-output``=STATE_OUTPUT`, **`--state_output`**`=STATE_OUTPUT`
Override the configured state_output value for minion output. One of 'full', 'terse', 'mixed', 'changes' or 'filter'. Default: 'none'.

`--state-verbose``=STATE_VERBOSE`, **`--state_verbose`**`=STATE_VERBOSE`
Override the configured state_verbose value for minion output. Set to True or False. Default: none.

Note: If using `--out=json`, you will probably want `--static` as well. Without the static option, you will get a separate JSON string per minion which makes JSON output invalid as a whole. This is due to using an iterative outputter. So if you want to feed it to a JSON parser, use `--static` as well.

See also

salt(7) *salt-master(1)* *salt-minion(1)*

18.3 salt-cloud

18.4 salt-cp

18.4.1 `salt-cp`

Copy a file or files to one or more minions

Synopsis

```
salt-cp '*' [ options ] SOURCE [SOURCE2 SOURCE3 ...] DEST

salt-cp -E '.*' [ options ] SOURCE [SOURCE2 SOURCE3 ...] DEST

salt-cp -G 'os:Arch.*' [ options ] SOURCE [SOURCE2 SOURCE3 ...] DEST
```

Description

salt-cp copies files from the master to all of the Salt minions matched by the specified target expression.

Note: salt-cp uses Salt's publishing mechanism. This means the privacy of the contents of the file on the wire is completely dependent upon the transport in use. In addition, if the master or minion is running with debug logging, the contents of the file will be logged to disk.

In addition, this tool is less efficient than the Salt fileserver when copying larger files. It is recommended to instead use *cp.get_file* to copy larger files to minions. However, this requires the file to be located within one of the fileserver directories.

Changed in version 2016.3.7,2016.11.6,2017.7.0: Compression support added, disable with `-n`. Also, if the destination path ends in a path separator (i.e. `/`, or `\` on Windows, the desitination will be assumed to be a directory. Finally, recursion is now supported, allowing for entire directories to be copied.

Changed in version 2016.11.7,2017.7.2: Reverted back to the old copy mode to preserve backward compatibility. The new functionality added in 2016.6.6 and 2017.7.0 is now available using the `-C` or `--chunked` CLI arguments. Note that compression, recursive copying, and support for copying large files is only available in chunked mode.

Options

--version
Print the version of Salt that is running.

--versions-report
Show program's dependencies and version number, and then exit

-h, --help
Show the help message and exit

-c `CONFIG_DIR`, **--config-dir**=`CONFIG_dir`
The location of the Salt configuration directory. This directory contains the configuration files for Salt master and minions. The default location on most systems is `/etc/salt`.

-t `TIMEOUT`, **--timeout**=`TIMEOUT`
The timeout in seconds to wait for replies from the Salt minions. The timeout number specifies how long the command line client will wait to query the minions and check on running jobs. Default: 5

Logging Options

Logging options which override any settings defined on the configuration files.

-l `LOG_LEVEL`, **--log-level**=`LOG_LEVEL`
Console logging log level. One of `all`, `garbage`, `trace`, `debug`, `info`, `warning`, `error`, `quiet`. Default: `warning`.

--log-file=`LOG_FILE`
Log file path. Default: /var/log/salt/master.

--log-file-level=`LOG_LEVEL_LOGFILE`
Logfile logging log level. One of `all`, `garbage`, `trace`, `debug`, `info`, `warning`, `error`, `quiet`. Default: `warning`.

Target Selection

The default matching that Salt utilizes is shell-style globbing around the minion id. See https://docs.python.org/2/library/fnmatch.html#module-fnmatch.

-E, --pcre
The target expression will be interpreted as a PCRE regular expression rather than a shell glob.

-L, --list
The target expression will be interpreted as a comma-delimited list; example: server1.foo.bar,server2.foo.bar,example7.quo.qux

-G, --grain
The target expression matches values returned by the Salt grains system on the minions. The target expression is in the format of '<grain value>:<glob expression>'; example: 'os:Arch*'

This was changed in version 0.9.8 to accept glob expressions instead of regular expression. To use regular expression matching with grains, use the –grain-pcre option.

--grain-pcre
The target expression matches values returned by the Salt grains system on the minions. The target expression is in the format of '<grain value>:< regular expression>'; example: 'os:Arch.*'

-N, --nodegroup
Use a predefined compound target defined in the Salt master configuration file.

-R, --range
Instead of using shell globs to evaluate the target, use a range expression to identify targets. Range expressions look like %cluster.

Using the Range option requires that a range server is set up and the location of the range server is referenced in the master configuration file.

-C, --chunked
Use new chunked mode to copy files. This mode supports large files, recursive directories copying and compression.

New in version 2016.11.7,2017.7.2.

-n, --no-compression
Disable gzip compression in chunked mode.

New in version 2016.3.7,2016.11.6,2017.7.0.

See also

`salt(1)` *`salt-master(1)`* *`salt-minion(1)`*

18.5 salt-extend

18.5.1 `salt-extend`

A utilty to generate extensions to the Salt source-code. This is used for :

- Adding new execution modules, state modules
- Adding unit tests to existing modules
- Adding integration tests to existing modules

Synopsis

```
salt-extend --help
```

Description

`salt-extend` is a templating tool for extending SaltStack. If you're looking to add a module to SaltStack, then the `salt-extend` utility can guide you through the process.

You can use Salt Extend to quickly create templated modules for adding new behaviours to some of the module subsystems within Salt.

Salt Extend takes a template directory and merges it into a SaltStack source code directory.

See also: *Salt Extend*.

Options

`--extension, -e`
The extension type you want to develop, e.g. module, module_unit, state

`--salt-directory, -o`
The path to the salt installation, defaults to .

`--name, -n`
The module name for the new module

`--description, -d`
A description of the new extension

`--no-merge`
Don't merge the new module into the Salt source directory specified by *–salt-directory*, save to a temporary directory and print the directory path

`--debug`
Print debug messages to stdout

See also

`salt-api(1)` *`salt-call(1)`* *`salt-cloud(1)`* *`salt-cp(1)`* *`salt-key(1)`* *`salt-main(1)`* *`salt-master(1)`* *`salt-minion(1)`* *`salt-run(1)`* *`salt-ssh(1)`* *`salt-syndic(1)`*

18.6 salt-key

18.6.1 `salt-key`

Synopsis

```
salt-key [ options ]
```

Description

Salt-key executes simple management of Salt server public keys used for authentication.

On initial connection, a Salt minion sends its public key to the Salt master. This key must be accepted using the `salt-key` command on the Salt master.

Salt minion keys can be in one of the following states:

- **unaccepted**: key is waiting to be accepted.
- **accepted**: key was accepted and the minion can communicate with the Salt master.
- **rejected**: key was rejected using the `salt-key` command. In this state the minion does not receive any communication from the Salt master.
- **denied**: key was rejected automatically by the Salt master. This occurs when a minion has a duplicate ID, or when a minion was rebuilt or had new keys generated and the previous key was not deleted from the Salt master. In this state the minion does not receive any communication from the Salt master.

To change the state of a minion key, use `-d` to delete the key and then accept or reject the key.

Options

`--version`
Print the version of Salt that is running.

`--versions-report`
Show program's dependencies and version number, and then exit

`-h`, **`--help`**
Show the help message and exit

`-c` `CONFIG_DIR`, **`--config-dir`**`=CONFIG_dir`
The location of the Salt configuration directory. This directory contains the configuration files for Salt master and minions. The default location on most systems is `/etc/salt`.

`-u` `USER`, **`--user`**`=USER`
Specify user to run salt-key

`--hard-crash`
Raise any original exception rather than exiting gracefully. Default is False.

`-q`, **`--quiet`**
Suppress output

`-y`, **`--yes`**
Answer 'Yes' to all questions presented, defaults to False

`--rotate-aes-key``=ROTATE_AES_KEY`
Setting this to False prevents the master from refreshing the key session when keys are deleted or rejected, this lowers the security of the key deletion/rejection operation. Default is True.

Logging Options

Logging options which override any settings defined on the configuration files.

`--log-file``=LOG_FILE`
Log file path. Default: /var/log/salt/minion.

`--log-file-level``=LOG_LEVEL_LOGFILE`
Logfile logging log level. One of `all`, `garbage`, `trace`, `debug`, `info`, `warning`, `error`, `quiet`. Default: `warning`.

Output Options

`--out`
Pass in an alternative outputter to display the return of data. This outputter can be any of the available outputters:

> `highstate`, `json`, `key`, `overstatestage`, `pprint`, `raw`, `txt`, `yaml`, and *many others*.

Some outputters are formatted only for data returned from specific functions. If an outputter is used that does not support the data passed into it, then Salt will fall back on the `pprint` outputter and display the return data using the Python `pprint` standard library module.

`--out-indent` `OUTPUT_INDENT`, **`--output-indent`** `OUTPUT_INDENT`
Print the output indented by the provided value in spaces. Negative values disable indentation. Only applicable in outputters that support indentation.

`--out-file``=OUTPUT_FILE`, **`--output-file`**`=OUTPUT_FILE`
Write the output to the specified file.

`--out-file-append`, **`--output-file-append`**
Append the output to the specified file.

`--no-color`
Disable all colored output

`--force-color`
Force colored output

> **Note:** When using colored output the color codes are as follows:
>
> `green` denotes success, `red` denotes failure, `blue` denotes changes and success and `yellow` denotes a expected future change in configuration.

`--state-output``=STATE_OUTPUT`, **`--state_output`**`=STATE_OUTPUT`
Override the configured state_output value for minion output. One of 'full', 'terse', 'mixed', 'changes' or 'filter'. Default: 'none'.

`--state-verbose``=STATE_VERBOSE`, **`--state_verbose`**`=STATE_VERBOSE`
Override the configured state_verbose value for minion output. Set to True or False. Default: none.

Actions

`-l` `ARG`, **`--list`**`=ARG`
List the public keys. The args `pre`, `un`, and `unaccepted` will list unaccepted/unsigned keys. `acc` or `accepted` will list accepted/signed keys. `rej` or `rejected` will list rejected keys. Finally, `all` will list all keys.

`-L`, **`--list-all`**
List all public keys. (Deprecated: use `--list all`)

`-a` `ACCEPT`, **`--accept`**`=ACCEPT`
Accept the specified public key (use –include-all to match rejected keys in addition to pending keys). Globs are supported.

`-A`, **`--accept-all`**
Accepts all pending keys.

`-r` `REJECT`, **`--reject`**`=REJECT`
Reject the specified public key (use –include-all to match accepted keys in addition to pending keys). Globs are supported.

`-R`, **`--reject-all`**
Rejects all pending keys.

`--include-all`
Include non-pending keys when accepting/rejecting.

`-p` `PRINT`, **`--print`**`=PRINT`
Print the specified public key.

`-P`, **`--print-all`**
Print all public keys

`-d` `DELETE`, **`--delete`**`=DELETE`
Delete the specified key. Globs are supported.

`-D, --delete-all`
Delete all keys.

`-f FINGER, --finger=FINGER`
Print the specified key’s fingerprint.

`-F, --finger-all`
Print all keys’ fingerprints.

Key Generation Options

`--gen-keys=GEN_KEYS`
Set a name to generate a keypair for use with salt

`--gen-keys-dir=GEN_KEYS_DIR`
Set the directory to save the generated keypair. Only works with ’gen_keys_dir’ option; default is the current directory.

`--keysize=KEYSIZE`
Set the keysize for the generated key, only works with the ’–gen-keys’ option, the key size must be 2048 or higher, otherwise it will be rounded up to 2048. The default is 2048.

`--gen-signature`
Create a signature file of the master’s public-key named master_pubkey_signature. The signature can be sent to a minion in the master’s auth-reply and enables the minion to verify the master’s public-key cryptographically. This requires a new signing-key-pair which can be auto-created with the –auto-create parameter.

`--priv=PRIV`
The private-key file to create a signature with

`--signature-path=SIGNATURE_PATH`
The path where the signature file should be written

`--pub=PUB`
The public-key file to create a signature for

`--auto-create`
Auto-create a signing key-pair if it does not yet exist

See also

`salt(7)` *`salt-master(1)`* *`salt-minion(1)`*

18.7 salt-master

18.7.1 `salt-master`

The Salt master daemon, used to control the Salt minions

Synopsis

```
salt-master [ options ]
```

Description

The master daemon controls the Salt minions

Options

`--version`
Print the version of Salt that is running.

`--versions-report`
Show program's dependencies and version number, and then exit

`-h, --help`
Show the help message and exit

`-c` `CONFIG_DIR`, **`--config-dir`**`=CONFIG_dir`
The location of the Salt configuration directory. This directory contains the configuration files for Salt master and minions. The default location on most systems is `/etc/salt`.

`-u` `USER`, **`--user`**`=USER`
Specify user to run salt-master

`-d, --daemon`
Run salt-master as a daemon

`--pid-file` `PIDFILE`
Specify the location of the pidfile. Default: /var/run/salt-master.pid

Logging Options

Logging options which override any settings defined on the configuration files.

`-l` `LOG_LEVEL`, **`--log-level`**`=LOG_LEVEL`
Console logging log level. One of `all`, `garbage`, `trace`, `debug`, `info`, `warning`, `error`, `quiet`. Default: `warning`.

`--log-file``=LOG_FILE`
Log file path. Default: /var/log/salt/master.

`--log-file-level``=LOG_LEVEL_LOGFILE`
Logfile logging log level. One of `all`, `garbage`, `trace`, `debug`, `info`, `warning`, `error`, `quiet`. Default: `warning`.

See also

`salt(1)` *`salt(7)`* *`salt-minion(1)`*

18.8 salt-minion

18.8.1 `salt-minion`

The Salt minion daemon, receives commands from a remote Salt master.

Synopsis

```
salt-minion [ options ]
```

Description

The Salt minion receives commands from the central Salt master and replies with the results of said commands.

Options

`--version`
Print the version of Salt that is running.

`--versions-report`
Show program's dependencies and version number, and then exit

`-h`, **`--help`**
Show the help message and exit

`-c` `CONFIG_DIR`, **`--config-dir`**`=CONFIG_dir`
The location of the Salt configuration directory. This directory contains the configuration files for Salt master and minions. The default location on most systems is `/etc/salt`.

`-u` `USER`, **`--user`**`=USER`
Specify user to run salt-minion

`-d`, **`--daemon`**
Run salt-minion as a daemon

`--pid-file` `PIDFILE`
Specify the location of the pidfile. Default: /var/run/salt-minion.pid

Logging Options

Logging options which override any settings defined on the configuration files.

`-l` `LOG_LEVEL`, **`--log-level`**`=LOG_LEVEL`
Console logging log level. One of `all`, `garbage`, `trace`, `debug`, `info`, `warning`, `error`, `quiet`. Default: `warning`.

`--log-file``=LOG_FILE`
Log file path. Default: /var/log/salt/minion.

`--log-file-level``=LOG_LEVEL_LOGFILE`
Logfile logging log level. One of `all`, `garbage`, `trace`, `debug`, `info`, `warning`, `error`, `quiet`. Default: `warning`.

See also

`salt(1)` *`salt(7)`* *`salt-master(1)`*

18.9 salt-proxy

18.9.1 `salt-proxy`

Receives commands from a Salt master and proxies these commands to devices that are unable to run a full minion.

Synopsis

```
salt-proxy [ options ]
```

Description

The Salt proxy minion receives commands from a Salt master, transmits appropriate commands to devices that are unable to run a minion, and replies with the results of said commands.

Options

`--proxyid`
The minion id that this proxy will assume. This is required.

`--version`
Print the version of Salt that is running.

`--versions-report`
Show program's dependencies and version number, and then exit

`-h, --help`
Show the help message and exit

`-c` `CONFIG_DIR`, **`--config-dir`**`=CONFIG_dir`
The location of the Salt configuration directory. This directory contains the configuration files for Salt master and minions. The default location on most systems is `/etc/salt`.

`-u` `USER`, **`--user`**`=USER`
Specify user to run salt-proxy

`-d, --daemon`
Run salt-proxy as a daemon

`--pid-file` `PIDFILE`
Specify the location of the pidfile. Default: `/var/run/salt-proxy-<id>.pid`

Logging Options

Logging options which override any settings defined on the configuration files.

`-l` `LOG_LEVEL`, **`--log-level`**`=LOG_LEVEL`
Console logging log level. One of `all`, `garbage`, `trace`, `debug`, `info`, `warning`, `error`, `quiet`. Default: `warning`.

`--log-file``=LOG_FILE`
Log file path. Default: /var/log/salt/minion.

--log-file-level=LOG_LEVEL_LOGFILE
Logfile logging log level. One of `all`, `garbage`, `trace`, `debug`, `info`, `warning`, `error`, `quiet`. Default: `warning`.

See also

salt(1) salt(7) salt-master(1) salt-minion(1)

18.10 salt-run

18.10.1 `salt-run`

Execute a Salt runner

Synopsis

```
salt-run RUNNER
```

Description

salt-run is the frontend command for executing `Salt Runners`. Salt runners are simple modules used to execute convenience functions on the master

Options

--version
Print the version of Salt that is running.

--versions-report
Show program's dependencies and version number, and then exit

-h, --help
Show the help message and exit

-c CONFIG_DIR, **--config-dir**=CONFIG_dir
The location of the Salt configuration directory. This directory contains the configuration files for Salt master and minions. The default location on most systems is `/etc/salt`.

-t TIMEOUT, **--timeout**=TIMEOUT
The timeout in seconds to wait for replies from the Salt minions. The timeout number specifies how long the command line client will wait to query the minions and check on running jobs. Default: 1

--hard-crash
Raise any original exception rather than exiting gracefully. Default is False.

-d, --doc, --documentation
Display documentation for runners, pass a module or a runner to see documentation on only that module/runner.

Logging Options

Logging options which override any settings defined on the configuration files.

-l `LOG_LEVEL`, **--log-level**=`LOG_LEVEL`
Console logging log level. One of `all`, `garbage`, `trace`, `debug`, `info`, `warning`, `error`, `quiet`. Default: `warning`.

--log-file=`LOG_FILE`
Log file path. Default: /var/log/salt/master.

--log-file-level=`LOG_LEVEL_LOGFILE`
Logfile logging log level. One of `all`, `garbage`, `trace`, `debug`, `info`, `warning`, `error`, `quiet`. Default: `warning`.

See also

salt(1) salt-master(1) salt-minion(1)

18.11 salt-ssh

18.11.1 salt-ssh

Synopsis

```
salt-ssh '*' [ options ] sys.doc

salt-ssh -E '.*' [ options ] sys.doc cmd
```

Description

Salt SSH allows for salt routines to be executed using only SSH for transport

Options

--version
Print the version of Salt that is running.

--versions-report
Show program's dependencies and version number, and then exit

-h, --help
Show the help message and exit

-c `CONFIG_DIR`, **--config-dir**=`CONFIG_dir`
The location of the Salt configuration directory. This directory contains the configuration files for Salt master and minions. The default location on most systems is `/etc/salt`.

--hard-crash
Raise any original exception rather than exiting gracefully. Default: False.

-r, --raw, --raw-shell
Execute a raw shell command.

`--roster`
Define which roster system to use, this defines if a database backend, scanner, or custom roster system is used. Default is the flat file roster.

`--roster-file`
Define an alternative location for the default roster file location. The default roster file is called `roster` and is found in the same directory as the master config file.

New in version 2014.1.0.

`--refresh, --refresh-cache`
Force a refresh of the master side data cache of the target's data. This is needed if a target's grains have been changed and the auto refresh timeframe has not been reached.

`--max-procs`
Set the number of concurrent minions to communicate with. This value defines how many processes are opened up at a time to manage connections, the more running process the faster communication should be, default is 25.

`--extra-filerefs``=EXTRA_FILEREFS`
Pass in extra files to include in the state tarball.

`--min-extra-modules``=MIN_EXTRA_MODS`
One or comma-separated list of extra Python modulesto be included into Minimal Salt.

`--thin-extra-modules``=THIN_EXTRA_MODS`
One or comma-separated list of extra Python modulesto be included into Thin Salt.

`-v, --verbose`
Turn on command verbosity, display jid.

`-s, --static`
Return the data from minions as a group after they all return.

`-w, --wipe`
Remove the deployment of the salt files when done executing.

`-W, --rand-thin-dir`
Select a random temp dir to deploy on the remote system. The dir will be cleaned after the execution.

`-t, --regen-thin, --thin`
Trigger a thin tarball regeneration. This is needed if custom grains/modules/states have been added or updated.

`--python2-bin``=PYTHON2_BIN`
Path to a python2 binary which has salt installed.

`--python3-bin``=PYTHON3_BIN`
Path to a python3 binary which has salt installed.

`--jid``=JID`
Pass a JID to be used instead of generating one.

Authentication Options

`--priv``=SSH_PRIV`
Specify the SSH private key file to be used for authentication.

`--priv-passwd``=SSH_PRIV_PASSWD`
Specify the SSH private key file's passphrase if need be.

-i, --ignore-host-keys
By default ssh host keys are honored and connections will ask for approval. Use this option to disable StrictHostKeyChecking.

--no-host-keys
Fully ignores ssh host keys which by default are honored and connections would ask for approval. Useful if the host key of a remote server has changed and would still error with –ignore-host-keys.

--user=SSH_USER
Set the default user to attempt to use when authenticating.

--passwd
Set the default password to attempt to use when authenticating.

--askpass
Interactively ask for the SSH password with no echo - avoids password in process args and stored in history.

--key-deploy
Set this flag to attempt to deploy the authorized ssh key with all minions. This combined with –passwd can make initial deployment of keys very fast and easy.

--identities-only
Use the only authentication identity files configured in the ssh_config files. See IdentitiesOnly flag in man ssh_config.

--sudo
Run command via sudo.

Scan Roster Options

--scan-ports=SSH_SCAN_PORTS
Comma-separated list of ports to scan in the scan roster.

--scan-timeout=SSH_SCAN_TIMEOUT
Scanning socket timeout for the scan roster.

Logging Options

Logging options which override any settings defined on the configuration files.

-l LOG_LEVEL**, --log-level**=LOG_LEVEL
Console logging log level. One of `all`, `garbage`, `trace`, `debug`, `info`, `warning`, `error`, `quiet`. Default: `warning`.

--log-file=LOG_FILE
Log file path. Default: /var/log/salt/ssh.

--log-file-level=LOG_LEVEL_LOGFILE
Logfile logging log level. One of `all`, `garbage`, `trace`, `debug`, `info`, `warning`, `error`, `quiet`. Default: `warning`.

Target Selection

The default matching that Salt utilizes is shell-style globbing around the minion id. See https://docs.python.org/2/library/fnmatch.html#module-fnmatch.

-E, --pcre
The target expression will be interpreted as a PCRE regular expression rather than a shell glob.

Output Options

--out
Pass in an alternative outputter to display the return of data. This outputter can be any of the available outputters:

> `highstate`, `json`, `key`, `overstatestage`, `pprint`, `raw`, `txt`, `yaml`, and *many others*.

Some outputters are formatted only for data returned from specific functions. If an outputter is used that does not support the data passed into it, then Salt will fall back on the `pprint` outputter and display the return data using the Python `pprint` standard library module.

--out-indent `OUTPUT_INDENT`**, --output-indent** `OUTPUT_INDENT`
Print the output indented by the provided value in spaces. Negative values disable indentation. Only applicable in outputters that support indentation.

--out-file`=OUTPUT_FILE`**, --output-file**`=OUTPUT_FILE`
Write the output to the specified file.

--out-file-append, --output-file-append
Append the output to the specified file.

--no-color
Disable all colored output

--force-color
Force colored output

> **Note:** When using colored output the color codes are as follows:
>
> `green` denotes success, `red` denotes failure, `blue` denotes changes and success and `yellow` denotes a expected future change in configuration.

--state-output`=STATE_OUTPUT`**, --state_output**`=STATE_OUTPUT`
Override the configured state_output value for minion output. One of 'full', 'terse', 'mixed', 'changes' or 'filter'. Default: 'none'.

--state-verbose`=STATE_VERBOSE`**, --state_verbose**`=STATE_VERBOSE`
Override the configured state_verbose value for minion output. Set to True or False. Default: none.

Note: If using `--out=json`, you will probably want `--static` as well. Without the static option, you will get a separate JSON string per minion which makes JSON output invalid as a whole. This is due to using an iterative outputter. So if you want to feed it to a JSON parser, use `--static` as well.

See also

salt(7) salt-master(1) salt-minion(1)

18.12 salt-syndic

18.12.1 `salt-syndic`

The Salt syndic daemon, a special minion that passes through commands from a higher master

Synopsis

```
salt-syndic [ options ]
```

Description

The Salt syndic daemon, a special minion that passes through commands from a higher master.

Options

`--version`
Print the version of Salt that is running.

`--versions-report`
Show program's dependencies and version number, and then exit

`-h`, **`--help`**
Show the help message and exit

`-c` `CONFIG_DIR`, **`--config-dir`**`=CONFIG_dir`
The location of the Salt configuration directory. This directory contains the configuration files for Salt master and minions. The default location on most systems is `/etc/salt`.

`-u` `USER`, **`--user`**`=USER`
Specify user to run salt-syndic

`-d`, **`--daemon`**
Run salt-syndic as a daemon

`--pid-file` `PIDFILE`
Specify the location of the pidfile. Default: /var/run/salt-syndic.pid

Logging Options

Logging options which override any settings defined on the configuration files.

`-l` `LOG_LEVEL`, **`--log-level`**`=LOG_LEVEL`
Console logging log level. One of `all`, `garbage`, `trace`, `debug`, `info`, `warning`, `error`, `quiet`. Default: `warning`.

`--log-file``=LOG_FILE`
Log file path. Default: /var/log/salt/master.

`--log-file-level``=LOG_LEVEL_LOGFILE`
Logfile logging log level. One of `all`, `garbage`, `trace`, `debug`, `info`, `warning`, `error`, `quiet`. Default: `warning`.

See also

salt(1) *salt-master(1)* *salt-minion(1)*

18.13 salt-unity

18.13.1 salt-unity

A unified invocation wrapper around other Salt CLI scripts.

Synopsis

```
salt-unity salt '*' test.version
```

Description

This script takes an argument which is one of the other Salt CLI scripts and invokes that script.

Options

See also

salt-api(1) *salt-call(1)* *salt-cloud(1)* *salt-cp(1)* *salt-key(1)* *salt-main(1)* *salt-master(1)* *salt-minion(1)* *salt-run(1)* *salt-ssh(1)* *salt-syndic(1)*

18.14 salt-api

18.14.1 salt-api

Start interfaces used to remotely connect to the salt master

Synopsis

```
salt-api
```

Description

The Salt API system manages network api connectors for the Salt Master

Options

--version
Print the version of Salt that is running.

--versions-report
Show program's dependencies and version number, and then exit

-h, --help
Show the help message and exit

-c CONFIG_DIR, **--config-dir**=CONFIG_dir
The location of the Salt configuration directory. This directory contains the configuration files for Salt master and minions. The default location on most systems is `/etc/salt`.

-d, --daemon
Run the salt-api as a daemon

--pid-file=PIDFILE
Specify the location of the pidfile. Default: /var/run/salt-api.pid

Logging Options

Logging options which override any settings defined on the configuration files.

-l LOG_LEVEL, **--log-level**=LOG_LEVEL
Console logging log level. One of `all`, `garbage`, `trace`, `debug`, `info`, `warning`, `error`, `quiet`. Default: `warning`.

--log-file=LOG_FILE
Log file path. Default: /var/log/salt/api.

--log-file-level=LOG_LEVEL_LOGFILE
Logfile logging log level. One of `all`, `garbage`, `trace`, `debug`, `info`, `warning`, `error`, `quiet`. Default: `warning`.

See also

salt-api(7) salt(7) salt-master(1)

18.15 spm

18.15.1 spm

Salt Package Manager

Synopsis

```
spm <command> [<argument>]
```

Description

spm is the frontend command for managing Salt packages. Packages normally only include formulas, meaning a group of SLS files that install into the `file_roots` on the Salt Master, but Salt modules can also be installed.

Options

`-y, --assume-yes`
Assume `yes` instead of prompting the other whether or not to proceed with a particular command. Default is False.

`-f, --force`
When presented with a course of action that spm would normally refuse to perform, that action will be performed anyway. This is often destructive, and should be used with caution.

Logging Options

Logging options which override any settings defined on the configuration files.

`-l` `LOG_LEVEL`**`, --log-level`**`=LOG_LEVEL`
Console logging log level. One of `all`, `garbage`, `trace`, `debug`, `info`, `warning`, `error`, `quiet`. Default: `warning`.

`--log-file``=LOG_FILE`
Log file path. Default: /var/log/salt/spm.

`--log-file-level``=LOG_LEVEL_LOGFILE`
Logfile logging log level. One of `all`, `garbage`, `trace`, `debug`, `info`, `warning`, `error`, `quiet`. Default: `warning`.

Commands

`update_repo`
Connect to remote repositories locally configured on the system and download their metadata.

`install`
Install a package from a configured SPM repository. Requires a package name.

`remove`
Remove an installed package from the system. Requires a package name.

`info`
List information about an installed package. Requires a package name.

`files`
List files belonging to an installed package. Requires a package name.

`local`
Perform one of the above options (except for remove) on a package file, instead of on a package in a repository, or an installed package. Requires a valid path to a local file on the system.

`build`
Build a package from a directory containing a FORMULA file. Requires a valid path to a local directory on the system.

create_repo
Scan a directory for valid SPM package files and build an SPM-METADATA file in that directory which describes them.

See also

salt(1) salt-master(1) salt-minion(1)

CHAPTER

NINETEEN

PILLARS

Salt includes a number of built-in external pillars, listed at *pillar modules*.

The below links contain documentation for the configuration options

- *master-side configuration*
- *minion-side configuration*

Note that some of same the configuration options from the master are present in the minion configuration file, these are used in *masterless* mode.

The source for the built-in Salt pillars can be found here: https://github.com/saltstack/salt/blob/master/salt/pillar

CHAPTER

TWENTY

MASTER TOPS

Salt includes a number of built-in subsystems to generate top file data, they are listed at *master tops modules*.

The source for the built-in Salt master tops can be found here: https://github.com/saltstack/salt/blob/master/salt/tops

CHAPTER

TWENTYONE

SALT MODULE REFERENCE

This section contains a list of the Python modules that are used to extend the various subsystems within Salt.

21.1 auth modules

auto	An "Always Approved" eauth interface to test against, not intended for production use
django	Provide authentication using Django Web Framework
file	Provide authentication using local files
keystone	Provide authentication using OpenStack Keystone
ldap	Provide authentication using simple LDAP binds
mysql	Provide authentication using MySQL.
pam	Authenticate against PAM
pki	Authenticate via a PKI certificate.
rest	Provide authentication using a REST call
sharedsecret	Provide authentication using configured shared secret
yubico	Provide authentication using YubiKey.

21.1.1 salt.auth.auto

An "Always Approved" eauth interface to test against, not intended for production use

`salt.auth.auto.`**`auth`**(*username*, *password*)

Authenticate!

21.1.2 salt.auth.django

Provide authentication using Django Web Framework

depends

- Django Web Framework

Django authentication depends on the presence of the django framework in the `PYTHONPATH`, the Django project's `settings.py` file being in the `PYTHONPATH` and accessible via the `DJANGO_SETTINGS_MODULE` environment variable.

Django auth can be defined like any other eauth module:

```
external_auth:
  django:
    fred:
      - .*
      - '@runner'
```

This will authenticate Fred via Django and allow him to run any execution module and all runners.

The authorization details can optionally be located inside the Django database. The relevant entry in the `models.py` file would look like this:

```
class SaltExternalAuthModel(models.Model):
    user_fk = models.ForeignKey(User, on_delete=models.CASCADE)
    minion_or_fn_matcher = models.CharField(max_length=255)
    minion_fn = models.CharField(max_length=255)
```

The *`external_auth`* clause in the master config would then look like this:

```
external_auth:
  django:
    ^model: <fully-qualified reference to model class>
```

When a user attempts to authenticate via Django, Salt will import the package indicated via the keyword `^model`. That model must have the fields indicated above, though the model DOES NOT have to be named 'SaltExternalAuthModel'.

`salt.auth.django.`**`acl`**`(`*username*`)`

Parameters **`username`** – Username to filter for

Returns Dictionary that can be slotted into the `__opts__` structure for eauth that designates the user associated ACL

Database records such as:

username	minion_or_fn_matcher	minion_fn
fred		test.ping
fred	server1	network.interfaces
fred	server1	raid.list
fred	server2	.*
guru	.*	
smartadmin	server1	.*

Should result in an eauth config such as:

```
fred:
  - test.ping
  - server1:
      - network.interfaces
      - raid.list
  - server2:
      - .*
guru:
  - .*
smartadmin:
  - server1:
    - .*
```

`salt.auth.django.`**`auth`**`(`*username*, *password*`)`
Simple Django auth

`salt.auth.django.`**`is_connection_usable`**`()`

21.1.3 salt.auth.file

Provide authentication using local files

New in version 2018.3.0.

The *file* auth module allows simple authentication via local files. Different filetypes are supported, including:

1. Text files, with passwords in plaintext or hashed
2. Apache-style htpasswd files
3. Apache-style htdigest files

Note: The `python-passlib` library is required when using a `^filetype` of `htpasswd` or `htdigest`.

The simplest example is a plaintext file with usernames and passwords:

```
external_auth:
  file:
    ^filename: /etc/insecure-user-list.txt
    gene:
      - .*
    dean:
      - test.*
```

In this example the `/etc/insecure-user-list.txt` file would be formatted as so:

```
dean:goneFishing
gene:OceanMan
```

`^filename` is the only required parameter. Any parameter that begins with a `^` is passed directly to the underlying file authentication function via `kwargs`, with the leading `^` being stripped.

The text file option is configurable to work with legacy formats:

```
external_auth:
  file:
    ^filename: /etc/legacy_users.txt
    ^filetype: text
    ^hashtype: md5
    ^username_field: 2
    ^password_field: 3
    ^field_separator: '|'
    trey:
      - .*
```

This would authenticate users against a file of the following format:

```
46|trey|16a0034f90b06bf3c5982ed8ac41aab4
555|mike|b6e02a4d2cb2a6ef0669e79be6fd02e4
2001|page|14fce21db306a43d3b680da1a527847a
8888|jon|c4e94ba906578ccf494d71f45795c6cb
```

Note: The *`hashutil.digest`* execution function is used for comparing hashed passwords, so any algorithm supported by that function will work.

There is also support for Apache-style `htpasswd` and `htdigest` files:

```
external_auth:
  file:
    ^filename: /var/www/html/.htusers
    ^filetype: htpasswd
    cory:
      - .*
```

When using `htdigest` the `^realm` must be set:

```
external_auth:
  file:
    ^filename: /var/www/html/.htdigest
    ^filetype: htdigest
    ^realm: MySecureRealm
    cory:
      - .*
```

`salt.auth.file.`**`auth`**`(`*username*, *password*`)`

File based authentication

^**filename** The path to the file to use for authentication.

^**filetype** The type of file: `text`, `htpasswd`, `htdigest`.

Default: `text`

^**realm** The realm required by htdigest authentication.

Note: The following parameters are only used with the `text` filetype.

---

^**hashtype** The digest format of the password. Can be `plaintext` or any digest available via *`hashutil.digest`*.

Default: `plaintext`

^**field_separator** The character to use as a delimiter between fields in a text file.

Default: `:`

^**username_field** The numbered field in the text file that contains the username, with numbering beginning at 1 (one).

Default: `1`

^**password_field** The numbered field in the text file that contains the password, with numbering beginning at 1 (one).

Default: `2`

21.1.4 salt.auth.keystone

Provide authentication using OpenStack Keystone

depends

- keystoneclient Python module

`salt.auth.keystone.`**`auth`**`(`*username, password*`)`
Try and authenticate

`salt.auth.keystone.`**`get_auth_url`**`()`
Try and get the URL from the config, else return localhost

21.1.5 salt.auth.ldap

Provide authentication using simple LDAP binds

depends

- ldap Python module

`salt.auth.ldap.`**`auth`**`(`*username, password*`)`
Simple LDAP auth

`salt.auth.ldap.`**`groups`**`(`*username, **kwargs*`)`
Authenticate against an LDAP group

Behavior is highly dependent on if Active Directory is in use.

AD handles group membership very differently than OpenLDAP. See the *External Authentication* documentation for a thorough discussion of available parameters for customizing the search.

OpenLDAP allows you to search for all groups in the directory and returns members of those groups. Then we check against the username entered.

`salt.auth.ldap.`**`process_acl`**`(`*auth_list, opts=None*`)`
Query LDAP, retrieve list of minion_ids from an OU or other search. For each minion_id returned from the LDAP search, copy the perms matchers into the auth dictionary :param auth_list: :param opts: __opts__ for when __opts__ is not injected :return: Modified auth list.

21.1.6 salt.auth.mysql

Provide authentication using MySQL.

When using MySQL as an authentication backend, you will need to create or use an existing table that has a username and a password column.

To get started, create a simple table that holds just a username and a password. The password field will hold a SHA256 checksum.

```
CREATE TABLE `users` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `username` varchar(25) DEFAULT NULL,
  `password` varchar(70) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=latin1;
```

To create a user within MySQL, execute the following statement.

```
INSERT INTO users VALUES (NULL, 'diana', SHA2('secret', 256))
```

```
mysql_auth:
  hostname: localhost
  database: SaltStack
  username: root
  password: letmein
  auth_sql: 'SELECT username FROM users WHERE username = "{0}" AND password = SHA2("
↪{1}", 256)'
```

The *auth_sql* contains the SQL that will validate a user to ensure they are correctly authenticated. This is where you can specify other SQL queries to authenticate users.

Enable MySQL authentication.

```
external_auth:
  mysql:
    damian:
      - test.*
```

depends

- MySQL-python Python module

`salt.auth.mysql.`**`auth`**`(`*username*, *password*`)`

Authenticate using a MySQL user table

21.1.7 salt.auth.pam

Authenticate against PAM

Provides an authenticate function that will allow the caller to authenticate a user against the Pluggable Authentication Modules (PAM) on the system.

Implemented using ctypes, so no compilation is necessary.

There is one extra configuration option for pam. The *pam_service* that is authenticated against. This defaults to *login*

```
auth.pam.service: login
```

Note: Solaris-like (SmartOS, OmniOS, …) systems may need `auth.pam.service` set to `other`.

Note: PAM authentication will not work for the `root` user.

The Python interface to PAM does not support authenticating as `root`.

Note: Using PAM groups with SSSD groups on python2.

To use sssd with the PAM eauth module and groups the *pysss* module is needed. On RedHat/CentOS this is *python-sss*.

This should not be needed with python >= 3.3, because the *os* modules has the *getgrouplist* function.

class `salt.auth.pam.`**`PamConv`**

Wrapper class for pam_conv structure

appdata_ptr
Structure/Union member

conv
Structure/Union member

class `salt.auth.pam.`**PamHandle**
Wrapper class for pam_handle_t

handle
Structure/Union member

class `salt.auth.pam.`**PamMessage**
Wrapper class for pam_message structure

msg
Structure/Union member

msg_style
Structure/Union member

class `salt.auth.pam.`**PamResponse**
Wrapper class for pam_response structure

resp
Structure/Union member

resp_retcode
Structure/Union member

`salt.auth.pam.`**auth**(*username*, *password*, ***kwargs*)
Authenticate via pam

`salt.auth.pam.`**authenticate**(*username*, *password*)
Returns True if the given username and password authenticate for the given service. Returns False otherwise

`username`: the username to authenticate

`password`: the password in plain text

`salt.auth.pam.`**groups**(*username*, **args*, ***kwargs*)
Retrieve groups for a given user for this auth provider

Uses system groups

21.1.8 salt.auth.pki

Authenticate via a PKI certificate.

Note: This module is Experimental and should be used with caution

Provides an authenticate function that will allow the caller to authenticate a user via their public cert against a pre-defined Certificate Authority.

TODO: Add a 'ca_dir' option to configure a directory of CA files, a la Apache.

depends

- pyOpenSSL module

`salt.auth.pki.`**`auth`**`(`*username, password, **kwargs*`)`

Returns True if the given user cert (password is the cert contents) was issued by the CA and if cert's Common Name is equal to username.

Returns False otherwise.

`username`: **we need it to run the auth function from CLI/API**; it should be in master config auth/acl

`password`: **contents of user certificate (pem-encoded user public key)**; why "password"? For CLI, it's the only available name

Configure the CA cert in the master config file:

```
external_auth:
  pki:
    ca_file: /etc/pki/tls/ca_certs/trusted-ca.crt
    your_user:
      - .*
```

21.1.9 salt.auth.rest

Provide authentication using a REST call

REST auth can be defined like any other eauth module:

```
external_auth:
  rest:
    ^url: https://url/for/rest/call
    fred:
      - .*
      - '@runner'
```

If there are entries underneath the ^url entry then they are merged with any responses from the REST call. In the above example, assuming the REST call does not return any additional ACLs, this will authenticate Fred via a REST call and allow him to run any execution module and all runners.

The REST call should return a JSON object that maps to a regular eauth YAML structure as above.

`salt.auth.rest.`**`auth`**`(`*username, password*`)`

REST authentication

`salt.auth.rest.`**`rest_auth_setup`**`()`

21.1.10 salt.auth.sharedsecret

Provide authentication using configured shared secret

```
external_auth:
  sharedsecret:
    fred:
      - .*
      - '@jobs'
```

The shared secret should be added to the master configuration, for example in /etc/salt/master.d/sharedsecret.conf (make sure that file is only readable by the user running the master):

```
sharedsecret: OIUHF_CHANGE_THIS_12h88
```

This auth module should be used with caution. It was initially designed to work with a frontal that takes care of authentication (for example kerberos) and places the shared secret in the HTTP headers to the salt-api call. This salt-api call should really be done on localhost to avoid someone eavesdropping on the shared secret.

See the documentation for cherrypy to setup the headers in your frontal.

New in version Beryllium.

`salt.auth.sharedsecret.`**`auth`**`(`*username, password*`)`
Shared secret authentication

21.1.11 salt.auth.yubico

Provide authentication using YubiKey.

New in version 2015.5.0.

depends yubico-client Python module

To get your YubiKey API key you will need to visit the website below.

https://upgrade.yubico.com/getapikey/

The resulting page will show the generated Client ID (aka AuthID or API ID) and the generated API key (Secret Key). Make a note of both and use these two values in your /etc/salt/master configuration.

/etc/salt/master

```
yubico_users:
  damian:
    id: 12345
    key: ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

```
external_auth:
  yubico:
    damian:
      - test.*
```

Please wait five to ten minutes after generating the key before testing so that the API key will be updated on all the YubiCloud servers.

`salt.auth.yubico.`**`auth`**`(`*username, password*`)`
Authenticate against yubico server

`salt.auth.yubico.`**`groups`**`(`*username, *args, **kwargs*`)`

21.2 beacon modules

adb	Beacon to emit adb device state changes for Android devices
avahi_announce	Beacon to announce via avahi (zeroconf)
bonjour_announce	Beacon to announce via Bonjour (zeroconf)
btmp	Beacon to fire events at failed login of users
diskusage	Beacon to monitor disk usage.
glxinfo	Beacon to emit when a display is available to a linux machine

Continued on next page

Table 2 – continued from previous page

haproxy	Watch current connections of haproxy server backends.
inotify	Watch files and translate the changes into salt events
journald	A simple beacon to watch journald for specific entries
load	Beacon to emit system load averages
log	
memusage	Beacon to monitor memory usage.
napalm_beacon	Watch NAPALM functions and fire events on specific triggers
network_info	Beacon to monitor statistics from ethernet adapters
network_settings	Beacon to monitor network adapter setting changes on Linux
pkg	Watch for pkgs that have upgrades, then fire an event.
proxy_example	Example beacon to use with salt-proxy
ps	Send events covering process status
salt_proxy	Beacon to manage and report the status of one or more salt proxy processes
sensehat	Monitor temperature, humidity and pressure using the SenseHat of a Raspberry Pi
service	Send events covering service status
sh	Watch the shell commands being executed actively.
status	The status beacon is intended to send a basic health check event up to the master, this allows for event driven routines based on presence to be set up.
telegram_bot_msg	Beacon to emit Telegram messages
twilio_txt_msg	Beacon to emit Twilio text messages
wtmp	Beacon to fire events at login of users as registered in the wtmp file

21.2.1 salt.beacons.adb module

Beacon to emit adb device state changes for Android devices

New in version 2016.3.0.

`salt.beacons.adb.`**`beacon`**`(`*config*`)`

Emit the status of all devices returned by adb

Specify the device states that should emit an event, there will be an event for each device with the event type and device specified.

```
beacons:
  adb:
    - states:
        - offline
        - unauthorized
        - missing
    - no_devices_event: True
    - battery_low: 25
```

`salt.beacons.adb.`**`validate`**`(`*config*`)`

Validate the beacon configuration

21.2.2 salt.beacons.avahi_announce module

Beacon to announce via avahi (zeroconf)

New in version 2016.11.0.

Dependencies

- python-avahi
- dbus-python

`salt.beacons.avahi_announce.`**`beacon`**`(`*config*`)`

Broadcast values via zeroconf

If the announced values are static, it is advised to set run_once: True (do not poll) on the beacon configuration.

The following are required configuration settings:

- `servicetype` - The service type to announce
- `port` - The port of the service to announce
- `txt` - The TXT record of the service being announced as a dict. Grains can be used to define TXT values using one of following two formats:
  - `grains.<grain_name>`
  - `grains.<grain_name>[i]` where i is an integer representing the index of the grain to use. If the grain is not a list, the index is ignored.

The following are optional configuration settings:

- `servicename` - Set the name of the service. Will use the hostname from the minion's `host` grain if this value is not set.
- `reset_on_change` - If `True` and there is a change in TXT records detected, it will stop announcing the service and then restart announcing the service. This interruption in service announcement may be desirable if the client relies on changes in the browse records to update its cache of TXT records. Defaults to `False`.
- `reset_wait` - The number of seconds to wait after announcement stops announcing and before it restarts announcing in the case where there is a change in TXT records detected and `reset_on_change` is `True`. Defaults to `0`.
- `copy_grains` - If `True`, Salt will copy the grains passed into the beacon when it backs them up to check for changes on the next iteration. Normally, instead of copy, it would use straight value assignment. This will allow detection of changes to grains where the grains are modified in-place instead of completely replaced. In-place grains changes are not currently done in the main Salt code but may be done due to a custom plug-in. Defaults to `False`.

Example Config

```yaml
beacons:
  avahi_announce:
    - run_once: True
    - servicetype: _demo._tcp
    - port: 1234
    - txt:
        ProdName: grains.productname
        SerialNo: grains.serialnumber
        Comments: 'this is a test'
```

`salt.beacons.avahi_announce.`**`validate`**`(`*config*`)`

Validate the beacon configuration

21.2.3 salt.beacons.bonjour_announce module

Beacon to announce via Bonjour (zeroconf)

`salt.beacons.bonjour_announce.`**`beacon`**`(`*config*`)`

Broadcast values via zeroconf

If the announced values are static, it is advised to set run_once: True (do not poll) on the beacon configuration.

The following are required configuration settings:

- `servicetype` - The service type to announce
- `port` - The port of the service to announce
- `txt` - The TXT record of the service being announced as a dict. Grains can be used to define TXT values using one of following two formats:
  - `grains.<grain_name>`
  - `grains.<grain_name>[i]` where i is an integer representing the index of the grain to use. If the grain is not a list, the index is ignored.

The following are optional configuration settings:

- `servicename` - Set the name of the service. Will use the hostname from the minion's `host` grain if this value is not set.
- `reset_on_change` - If `True` and there is a change in TXT records detected, it will stop announcing the service and then restart announcing the service. This interruption in service announcement may be desirable if the client relies on changes in the browse records to update its cache of TXT records. Defaults to `False`.
- `reset_wait` - The number of seconds to wait after announcement stops announcing and before it restarts announcing in the case where there is a change in TXT records detected and `reset_on_change` is `True`. Defaults to `0`.
- `copy_grains` - If `True`, Salt will copy the grains passed into the beacon when it backs them up to check for changes on the next iteration. Normally, instead of copy, it would use straight value assignment. This will allow detection of changes to grains where the grains are modified in-place instead of completely replaced. In-place grains changes are not currently done in the main Salt code but may be done due to a custom plug-in. Defaults to `False`.

Example Config

```
beacons:
  bonjour_announce:
    - run_once: True
    - servicetype: _demo._tcp
    - port: 1234
    - txt:
        ProdName: grains.productname
        SerialNo: grains.serialnumber
        Comments: 'this is a test'
```

`salt.beacons.bonjour_announce.`**`validate`**`(`*config*`)`

Validate the beacon configuration

21.2.4 salt.beacons.btmp

Beacon to fire events at failed login of users

New in version 2015.5.0.

Example Configuration

```
# Fire events on all failed logins
beacons:
  btmp: []

# Matching on user name, using a default time range
beacons:
  btmp:
    - users:
        gareth:
    - defaults:
        time_range:
            start: '8am'
            end: '4pm'

# Matching on user name, overriding the default time range
beacons:
  btmp:
    - users:
        gareth:
            time_range:
                start: '8am'
                end: '4pm'
    - defaults:
        time_range:
            start: '8am'
            end: '4pm'

# Matching on group name, overriding the default time range
beacons:
  btmp:
    - groups:
        users:
            time_range:
                start: '8am'
                end: '4pm'
    - defaults:
        time_range:
            start: '8am'
            end: '4pm'
```

Use Case: Posting Failed Login Events to Slack

This can be done using the following reactor SLS:

```
report-wtmp:
  runner.salt.cmd:
    - args:
```

(continues on next page)

(continued from previous page)

```
      - fun: slack.post_message
      - channel: mychannel      # Slack channel
      - from_name: someuser     # Slack user
      - message: "Failed login from `{{ data.get('user', '') or 'unknown user' }}` on
↪`{{ data['id'] }}`"
```

Match the event like so in the master config file:

```
reactor:

  - 'salt/beacon/*/btmp/':
    - salt://reactor/btmp.sls
```

Note: This approach uses the *slack execution module* directly on the master, and therefore requires that the master has a slack API key in its configuration:

```
slack:
  api_key: xoxb-XXXXXXXXXXXX-XXXXXXXXXXXX-XXXXXXXXXXXXXXXXXXXXXXXX
```

See the *slack execution module* documentation for more information. While you can use an individual user's API key to post to Slack, a bot user is likely better suited for this. The *slack engine* documentation has information on how to set up a bot user.

`salt.beacons.btmp.`**`beacon`**(*config*)
Read the last btmp file and return information on the failed logins

`salt.beacons.btmp.`**`validate`**(*config*)
Validate the beacon configuration

21.2.5 salt.beacons.diskusage

Beacon to monitor disk usage.

New in version 2015.5.0.

depends python-psutil

`salt.beacons.diskusage.`**`beacon`**(*config*)
Monitor the disk usage of the minion

Specify thresholds for each disk and only emit a beacon if any of them are exceeded.

```
beacons:
  diskusage:
    - /: 63%
    - /mnt/nfs: 50%
```

Windows drives must be quoted to avoid yaml syntax errors

```
beacons:
  diskusage:
    -  interval: 120
    - 'c:\\': 90%
    - 'd:\\': 50%
```

Regular expressions can be used as mount points.

```
beacons:
  diskusage:
    - '^\/(?!home).*$': 90%
    - '^[a-zA-Z]:\\$': 50%
```

The first one will match all mounted disks beginning with "/", except /home The second one will match disks from A:to Z:on a Windows system

Note that if a regular expression are evaluated after static mount points, which means that if a regular expression matches another defined mount point, it will override the previously defined threshold.

`salt.beacons.diskusage.`**`validate`**`(`*config*`)`
Validate the beacon configuration

21.2.6 salt.beacons.glxinfo module

Beacon to emit when a display is available to a linux machine

New in version 2016.3.0.

`salt.beacons.glxinfo.`**`beacon`**`(`*config*`)`
Emit the status of a connected display to the minion

Mainly this is used to detect when the display fails to connect for whatever reason.

```
beacons:
  glxinfo:
    - user: frank
    - screen_event: True
```

`salt.beacons.glxinfo.`**`validate`**`(`*config*`)`
Validate the beacon configuration

21.2.7 salt.beacons.haproxy module

Watch current connections of haproxy server backends. Fire an event when over a specified threshold.

New in version 2016.11.0.

`salt.beacons.haproxy.`**`beacon`**`(`*config*`)`
Check if current number of sessions of a server for a specific haproxy backend is over a defined threshold.

```
beacons:
  haproxy:
    - backends:
        www-backend:
            threshold: 45
            servers:
              - web1
              - web2
    - interval: 120
```

`salt.beacons.haproxy.`**`validate`**`(`*config*`)`
Validate the beacon configuration

21.2.8 salt.beacons.inotify

Watch files and translate the changes into salt events

depends

- pyinotify Python module >= 0.9.5

Caution Using generic mask options like open, access, ignored, and closed_nowrite with reactors can easily cause the reactor to loop on itself. To mitigate this behavior, consider setting the *disable_during_state_run* flag to *True* in the beacon configuration.

note The *inotify* beacon only works on OSes that have *inotify* kernel support.

`salt.beacons.inotify.`**`beacon`**`(`*`config`*`)`

Watch the configured files

Example Config

```
beacons:
  inotify:
    - files:
        /path/to/file/or/dir:
          mask:
            - open
            - create
            - close_write
          recurse: True
          auto_add: True
          exclude:
            - /path/to/file/or/dir/exclude1
            - /path/to/file/or/dir/exclude2
            - /path/to/file/or/dir/regex[a-m]*$:
                regex: True
    - coalesce: True
```

The mask list can contain the following events (the default mask is create, delete, and modify):

- access - File accessed
- attrib - File metadata changed
- close_nowrite - Unwritable file closed
- close_write - Writable file closed
- create - File created in watched directory
- delete - File deleted from watched directory
- delete_self - Watched file or directory deleted
- modify - File modified
- moved_from - File moved out of watched directory
- moved_to - File moved into watched directory
- move_self - Watched file moved
- open - File opened

The mask can also contain the following options:

- dont_follow - Don't dereference symbolic links

- excl_unlink - Omit events for children after they have been unlinked
- oneshot - Remove watch after one event
- onlydir - Operate only if name is directory

recurse: Recursively watch files in the directory

auto_add: Automatically start watching files that are created in the watched directory

exclude: Exclude directories or files from triggering events in the watched directory. Can use regex if regex is set to True

coalesce: If this coalescing option is enabled, events are filtered based on their unicity, only unique events are enqueued, doublons are discarded. An event is unique when the combination of its fields (wd, mask, cookie, name) is unique among events of a same batch. After a batch of events is processed any events are accepted again. This option is top-level (at the same level as the path) and therefore affects all paths that are being watched. This is due to this option being at the Notifier level in pyinotify.

`salt.beacons.inotify.`**`close`**`(`*config*`)`

`salt.beacons.inotify.`**`validate`**`(`*config*`)`
Validate the beacon configuration

21.2.9 salt.beacons.journald

A simple beacon to watch journald for specific entries

`salt.beacons.journald.`**`beacon`**`(`*config*`)`
The journald beacon allows for the systemd journal to be parsed and linked objects to be turned into events.

This beacons config will return all sshd jornal entries

```
beacons:
  journald:
    - services:
        sshd:
          SYSLOG_IDENTIFIER: sshd
          PRIORITY: 6
```

`salt.beacons.journald.`**`validate`**`(`*config*`)`
Validate the beacon configuration

21.2.10 salt.beacons.load

Beacon to emit system load averages

`salt.beacons.load.`**`beacon`**`(`*config*`)`
Emit the load averages of this host.

Specify thresholds for each load average and only emit a beacon if any of them are exceeded.

onchangeonly: when *onchangeonly* is True the beacon will fire events only when the load average pass one threshold. Otherwise, it will fire an event at each beacon interval. The default is False.

***emitatstartup*: when *emitatstartup* is False the beacon will not fire** event when the minion is reload. Applicable only when *onchangeonly* is True. The default is True.

```
beacons:
  load:
    - averages:
        1m:
          - 0.0
          - 2.0
        5m:
          - 0.0
          - 1.5
        15m:
          - 0.1
          - 1.0
    - emitatstartup: True
    - onchangeonly: False
```

`salt.beacons.load.`**`validate`**`(`*config*`)`
Validate the beacon configuration

21.2.11 salt.beacons.log module

21.2.12 salt.beacons.memusage module

Beacon to monitor memory usage.

New in version 2016.3.0.

depends python-psutil

`salt.beacons.memusage.`**`beacon`**`(`*config*`)`
Monitor the memory usage of the minion

Specify thresholds for percent used and only emit a beacon if it is exceeded.

```
beacons:
  memusage:
    - percent: 63%
```

`salt.beacons.memusage.`**`validate`**`(`*config*`)`
Validate the beacon configuration

21.2.13 salt.beacons.napalm_beacon

Watch NAPALM functions and fire events on specific triggers

New in version 2018.3.0.

Note: The `NAPALM` beacon only works only when running under a regular Minion or a Proxy Minion, managed via NAPALM. Check the documentation for the *`NAPALM proxy module`*.

The configuration accepts a list of Salt functions to be invoked, and the corresponding output hierarchy that should be matched against. To invoke a function with certain arguments, they can be specified using the `_args` key, or `_kwargs` for more specific key-value arguments.

The match structure follows the output hierarchy of the NAPALM functions, under the `out` key.

For example, the following is normal structure returned by the *ntp.stats* execution function:

```
{
    "comment": "",
    "result": true,
    "out": [
        {
            "referenceid": ".GPSs.",
            "remote": "172.17.17.1",
            "synchronized": true,
            "reachability": 377,
            "offset": 0.461,
            "when": "860",
            "delay": 143.606,
            "hostpoll": 1024,
            "stratum": 1,
            "jitter": 0.027,
            "type": "-"
        },
        {
            "referenceid": ".INIT.",
            "remote": "172.17.17.2",
            "synchronized": false,
            "reachability": 0,
            "offset": 0.0,
            "when": "-",
            "delay": 0.0,
            "hostpoll": 1024,
            "stratum": 16,
            "jitter": 4000.0,
            "type": "-"
        }
    ]
}
```

In order to fire events when the synchronization is lost with one of the NTP peers, e.g., `172.17.17.2`, we can match it explicitly as:

```
ntp.stats:
  remote: 172.17.17.2
  synchronized: false
```

There is one single nesting level, as the output of `ntp.stats` is just a list of dictionaries, and this beacon will compare each dictionary from the list with the structure examplified above.

Note: When we want to match on any element at a certain level, we can configure `*` to match anything.

Considering a more complex structure consisting on multiple nested levels, e.g., the output of the *bgp.neighbors* execution function, to check when any neighbor from the `global` routing table is down, the match structure would have the format:

```
bgp.neighbors:
  global:
    '*':
      up: false
```

The match structure above will match any BGP neighbor, with any network (`*` matches any AS number), under the `global` VRF. In other words, this beacon will push an event on the Salt bus when there's a BGP neighbor down.

The right operand can also accept mathematical operations (i.e., <, <=, !=, >, >= etc.) when comparing numerical values.

Configuration Example:

```
beacons:
  napalm:
    - net.interfaces:
        # fire events when any interfaces is down
        '*':
          is_up: false
    - net.interfaces:
        # fire events only when the xe-0/0/0 interface is down
        'xe-0/0/0':
          is_up: false
    - ntp.stats:
        # fire when there's any NTP peer unsynchornized
        synchronized: false
    - ntp.stats:
        # fire only when the synchronization
        # with with the 172.17.17.2 NTP server is lost
        _args:
          - 172.17.17.2
        synchronized: false
    - ntp.stats:
        # fire only when there's a NTP peer with
        # synchronization stratum > 5
        stratum: '> 5'
```

Event structure example:

```
{
    "_stamp": "2017-09-05T09:51:09.377202",
    "args": [],
    "data": {
        "comment": "",
        "out": [
            {
                "delay": 0.0,
                "hostpoll": 1024,
                "jitter": 4000.0,
                "offset": 0.0,
                "reachability": 0,
                "referenceid": ".INIT.",
                "remote": "172.17.17.1",
                "stratum": 16,
                "synchronized": false,
                "type": "-",
                "when": "-"
            }
        ],
        "result": true
    },
    "fun": "ntp.stats",
    "id": "edge01.bjm01",
    "kwargs": {},
    "match": {
```

(continues on next page)

(continued from previous page)

```
        "stratum": "> 5"
    }
}
```

The event examplified above has been fired when the device identified by the Minion id `edge01.bjm01` has been synchronized with a NTP server at a stratum level greater than 5.

`salt.beacons.napalm_beacon.`**`beacon`**`(`*config*`)`
Watch napalm function and fire events.

`salt.beacons.napalm_beacon.`**`validate`**`(`*config*`)`
Validate the beacon configuration.

21.2.14 salt.beacons.network_info

Beacon to monitor statistics from ethernet adapters

New in version 2015.5.0.

`salt.beacons.network_info.`**`beacon`**`(`*config*`)`
Emit the network statistics of this host.

Specify thresholds for each network stat and only emit a beacon if any of them are exceeded.

Emit beacon when any values are equal to configured values.

```
beacons:
  network_info:
    - interfaces:
        eth0:
          type: equal
          bytes_sent: 100000
          bytes_recv: 100000
          packets_sent: 100000
          packets_recv: 100000
          errin: 100
          errout: 100
          dropin: 100
          dropout: 100
```

Emit beacon when any values are greater than configured values.

```
beacons:
  network_info:
    - interfaces:
        eth0:
          type: greater
          bytes_sent: 100000
          bytes_recv: 100000
          packets_sent: 100000
          packets_recv: 100000
          errin: 100
          errout: 100
          dropin: 100
          dropout: 100
```

`salt.beacons.network_info.`**`validate`**`(`*config*`)`
Validate the beacon configuration

21.2.15 salt.beacons.network_settings

Beacon to monitor network adapter setting changes on Linux

New in version 2016.3.0.

class `salt.beacons.network_settings.`**`Hashabledict`**

Helper class that implements a hash function for a dictionary

`salt.beacons.network_settings.`**`beacon`**`(`*config*`)`

Watch for changes on network settings

By default, the beacon will emit when there is a value change on one of the settings on watch. The config also support the onvalue parameter for each setting, which instruct the beacon to only emit if the setting changed to the value defined.

Example Config

```
beacons:
  network_settings:
    - interfaces:
        eth0:
          ipaddr:
          promiscuity:
            onvalue: 1
        eth1:
          linkmode:
```

The config above will check for value changes on eth0 ipaddr and eth1 linkmode. It will also emit if the promiscuity value changes to 1.

Beacon items can use the * wildcard to make a definition apply to several interfaces. For example an eth* would apply to all ethernet interfaces.

Setting the argument coalesce = True will combine all the beacon results on a single event. The example below shows how to trigger coalesced results:

```
beacons:
  network_settings:
    - coalesce: True
    - interfaces:
        eth0:
          ipaddr:
          promiscuity:
```

`salt.beacons.network_settings.`**`validate`**`(`*config*`)`

Validate the beacon configuration

21.2.16 salt.beacons.pkg

Watch for pkgs that have upgrades, then fire an event.

New in version 2016.3.0.

`salt.beacons.pkg.`**`beacon`**`(`*config*`)`

Check if installed packages are the latest versions and fire an event for those that have upgrades.

```
beacons:
  pkg:
    - pkgs:
        - zsh
        - apache2
    - refresh: True
```

`salt.beacons.pkg.`**`validate`**(*config*)
Validate the beacon configuration

21.2.17 salt.beacons.proxy_example module

Example beacon to use with salt-proxy

```
beacons:
  proxy_example:
    endpoint: beacon
```

`salt.beacons.proxy_example.`**`beacon`**(*config*)
Called several times each second https://docs.saltstack.com/en/latest/topics/beacons/#the-beacon-function

```
beacons:
  proxy_example:
    - endpoint: beacon
```

`salt.beacons.proxy_example.`**`validate`**(*config*)
Validate the beacon configuration

21.2.18 salt.beacons.ps module

Send events covering process status

`salt.beacons.ps.`**`beacon`**(*config*)
Scan for processes and fire events

Example Config

```
beacons:
  ps:
    - processes:
        salt-master: running
        mysql: stopped
```

The config above sets up beacons to check that processes are running or stopped.

`salt.beacons.ps.`**`validate`**(*config*)
Validate the beacon configuration

21.2.19 salt.beacons.salt_proxy module

Beacon to manage and report the status of one or more salt proxy processes

New in version 2015.8.3.

`salt.beacons.salt_proxy.`**`beacon`**`(`*config*`)`

Handle configured proxies

```
beacons:
  salt_proxy:
    - proxies:
        p8000: {}
        p8001: {}
```

`salt.beacons.salt_proxy.`**`validate`**`(`*config*`)`

Validate the beacon configuration

21.2.20 salt.beacons.sensehat module

Monitor temperature, humidity and pressure using the SenseHat of a Raspberry Pi

New in version 2017.7.0.

maintainer Benedikt Werner <1benediktwerner@gmail.com>

maturity new

depends sense_hat Python module

`salt.beacons.sensehat.`**`beacon`**`(`*config*`)`

Monitor the temperature, humidity and pressure using the SenseHat sensors.

You can either specify a threshold for each value and only emit a beacon if it is exceeded or define a range and emit a beacon when the value is out of range.

Units: * humidity: percent * temperature: degrees Celsius * temperature_from_pressure: degrees Celsius * pressure: Millibars

```
beacons:
  sensehat:
    - sensors:
        humidity: 70%
        temperature: [20, 40]
        temperature_from_pressure: 40
        pressure: 1500
```

`salt.beacons.sensehat.`**`validate`**`(`*config*`)`

Validate the beacon configuration

21.2.21 salt.beacons.service

Send events covering service status

`salt.beacons.service.`**`beacon`**`(`*config*`)`

Scan for the configured services and fire events

Example Config

```
beacons:
  service:
    - services:
        salt-master: {}
        mysql: {}
```

The config above sets up beacons to check for the salt-master and mysql services.

The config also supports two other parameters for each service:

onchangeonly: when *onchangeonly* is True the beacon will fire events only when the service status changes. Otherwise, it will fire an event at each beacon interval. The default is False.

delay: when *delay* is greater than 0 the beacon will fire events only after the service status changes, and the delay (in seconds) has passed. Applicable only when *onchangeonly* is True. The default is 0.

emitatstartup: when *emitatstartup* is False the beacon will not fire event when the minion is reload. Applicable only when *onchangeonly* is True. The default is True.

uncleanshutdown: If *uncleanshutdown* is present it should point to the location of a pid file for the service. Most services will not clean up this pid file if they are shutdown uncleanly (e.g. via *kill -9*) or if they are terminated through a crash such as a segmentation fault. If the file is present, then the beacon will add *uncleanshutdown: True* to the event. If not present, the field will be False. The field is only added when the service is NOT running. Omitting the configuration variable altogether will turn this feature off.

Please note that some init systems can remove the pid file if the service registers as crashed. One such example is nginx on CentOS 7, where the service unit removes the pid file when the service shuts down (IE: the pid file is observed as removed when kill -9 is sent to the nginx master process). The 'uncleanshutdown' option might not be of much use there, unless the unit file is modified.

Here is an example that will fire an event 30 seconds after the state of nginx changes and report an uncleanshutdown. This example is for Arch, which places nginx's pid file in */run*.

```
beacons:
  service:
    - services:
        nginx:
          onchangeonly: True
          delay: 30
          uncleanshutdown: /run/nginx.pid
```

`salt.beacons.service.`**`validate`**`(`*config*`)`
Validate the beacon configuration

21.2.22 salt.beacons.sh

Watch the shell commands being executed actively. This beacon requires strace.

`salt.beacons.sh.`**`beacon`**`(`*config*`)`
Scan the shell execve routines. This beacon will convert all login shells

```
beacons:
  sh: []
```

`salt.beacons.sh.`**`validate`**`(`*config*`)`
Validate the beacon configuration

21.2.23 salt.beacons.status module

The status beacon is intended to send a basic health check event up to the master, this allows for event driven routines based on presence to be set up.

The intention of this beacon is to add the config options to add monitoring stats to the health beacon making it a one stop shop for gathering systems health and status data

New in version 2016.11.0.

To configure this beacon to use the defaults, set up an empty dict for it in the minion config:

```
beacons:
  status: []
```

By default, all of the information from the following execution module functions will be returned:

- loadavg
- cpustats
- meminfo
- vmstats
- time

You can also configure your own set of functions to be returned:

```
beacons:
  status:
    - time:
      - all
    - loadavg:
      - all
```

You may also configure only certain fields from each function to be returned. For instance, the `loadavg` function returns the following fields:

- 1-min
- 5-min
- 15-min

If you wanted to return only the `1-min` and `5-min` fields for `loadavg` then you would configure:

```
beacons:
  status:
    - loadavg:
      - 1-min
      - 5-min
```

Other functions only return a single value instead of a dictionary. With these, you may specify `all` or `0`. The following are both valid:

```
beacons:
  status:
    - time:
      - all

beacons:
  status:
    - time:
      - 0
```

If a `status` function returns a list, you may return the index marker or markers for specific list items:

```
beacons:
  status:
    - w:
      - 0
      - 1
      - 2
```

Warning: Not all status functions are supported for every operating system. Be certain to check the minion log for errors after configuring this beacon.

`salt.beacons.status.`**`beacon`**`(`*config*`)`
Return status for requested information

`salt.beacons.status.`**`validate`**`(`*config*`)`
Validate the the config is a dict

21.2.24 salt.beacons.telegram_bot_msg

Beacon to emit Telegram messages

Requires the python-telegram-bot library

`salt.beacons.telegram_bot_msg.`**`beacon`**`(`*config*`)`
Emit a dict with a key ”msgs” whose value is a list of messages sent to the configured bot by one of the allowed usernames.

```
beacons:
  telegram_bot_msg:
    - token: "<bot access token>"
    - accept_from:
      - "<valid username>"
    - interval: 10
```

`salt.beacons.telegram_bot_msg.`**`validate`**`(`*config*`)`
Validate the beacon configuration

21.2.25 salt.beacons.twilio_txt_msg

Beacon to emit Twilio text messages

`salt.beacons.twilio_txt_msg.`**`beacon`**`(`*config*`)`
Emit a dict name ”texts” whose value is a list of texts.

```
beacons:
  twilio_txt_msg:
    - account_sid: "<account sid>"
    - auth_token: "<auth token>"
    - twilio_number: "+15555555555"
    - interval: 10
```

`salt.beacons.twilio_txt_msg.`**`validate`**`(`*config*`)`
Validate the beacon configuration

21.2.26 salt.beacons.wtmp

Beacon to fire events at login of users as registered in the wtmp file

New in version 2015.5.0.

Example Configuration

```
# Fire events on all logins
beacons:
  wtmp: []

# Matching on user name, using a default time range
beacons:
  wtmp:
    - users:
        gareth:
    - defaults:
        time_range:
            start: '8am'
            end: '4pm'

# Matching on user name, overriding the default time range
beacons:
  wtmp:
    - users:
        gareth:
            time_range:
                start: '7am'
                end: '3pm'
    - defaults:
        time_range:
            start: '8am'
            end: '4pm'

# Matching on group name, overriding the default time range
beacons:
  wtmp:
    - groups:
        users:
            time_range:
                start: '7am'
                end: '3pm'
    - defaults:
        time_range:
            start: '8am'
            end: '4pm'
```

How to Tell What An Event Means

In the events that this beacon fires, a type of `7` denotes a login, while a type of `8` denotes a logout. These values correspond to the `ut_type` value from a wtmp/utmp event (see the `wtmp` manpage for more information). In the extremely unlikely case that your platform uses different values, they can be overridden using a `ut_type` key in the beacon configuration:

```
beacons:
  wtmp:
    - ut_type:
        login: 9
        logout: 10
```

This beacon's events include an `action` key which will be either `login` or `logout` depending on the event type.

Changed in version 2019.2.0: `action` key added to beacon event, and `ut_type` config parameter added.

Use Case: Posting Login/Logout Events to Slack

This can be done using the following reactor SLS:

```
report-wtmp:
  runner.salt.cmd:
    - args:
      - fun: slack.post_message
      - channel: mychannel      # Slack channel
      - from_name: someuser     # Slack user
      - message: "{{ data.get('action', 'Unknown event') | capitalize }} from `{{↵
↪data.get('user', '') or 'unknown user' }}` on `{{ data['id'] }}`"
```

Match the event like so in the master config file:

```
reactor:

  - 'salt/beacon/*/wtmp/':
    - salt://reactor/wtmp.sls
```

Note: This approach uses the *`slack execution module`* directly on the master, and therefore requires that the master has a slack API key in its configuration:

```
slack:
  api_key: xoxb-XXXXXXXXXXXX-XXXXXXXXXXXX-XXXXXXXXXXXXXXXXXXXXXXXX
```

See the *`slack execution module`* documentation for more information. While you can use an individual user's API key to post to Slack, a bot user is likely better suited for this. The *`slack engine`* documentation has information on how to set up a bot user.

`salt.beacons.wtmp.`**`beacon`**(*config*)
Read the last wtmp file and return information on the logins

`salt.beacons.wtmp.`**`validate`**(*config*)
Validate the beacon configuration

21.3 cache modules

`consul`	Minion data cache plugin for Consul key/value data store.
`etcd_cache`	Minion data cache plugin for Etcd key/value data store.

Continued on next page

Table 3 – continued from previous page

localfs	Cache data in filesystem.
mysql_cache	Minion data cache plugin for MySQL database.
redis_cache	Redis

21.3.1 salt.cache.consul

Minion data cache plugin for Consul key/value data store.

New in version 2016.11.2.

depends python-consul >= 0.2.0

It is up to the system administrator to set up and configure the Consul infrastructure. All is needed for this plugin is a working Consul agent with a read-write access to the key-value store.

The related documentation can be found in the Consul documentation.

To enable this cache plugin, the master will need the python client for Consul installed. This can be easily installed with pip:

```
pip install python-consul
```

Optionally, depending on the Consul agent configuration, the following values could be set in the master config. These are the defaults:

```
consul.host: 127.0.0.1
consul.port: 8500
consul.token: None
consul.scheme: http
consul.consistency: default
consul.dc: dc1
consul.verify: True
```

Related docs could be found in the python-consul documentation.

To use the consul as a minion data cache backend, set the master `cache` config value to `consul`:

```
cache: consul
```

`salt.cache.consul.`**`contains`**`(`*bank*, *key*`)`
Checks if the specified bank contains the specified key.

`salt.cache.consul.`**`fetch`**`(`*bank*, *key*`)`
Fetch a key value.

`salt.cache.consul.`**`flush`**`(`*bank*, *key=None*`)`
Remove the key from the cache bank with all the key content.

`salt.cache.consul.`**`list_`**`(`*bank*`)`
Return an iterable object containing all entries stored in the specified bank.

`salt.cache.consul.`**`store`**`(`*bank*, *key*, *data*`)`
Store a key value.

21.3.2 salt.cache.etcd_cache

Minion data cache plugin for Etcd key/value data store.

New in version develop.

It is up to the system administrator to set up and configure the Etcd infrastructure. All is needed for this plugin is a working Etcd agent with a read-write access to the key-value store.

The related documentation can be found in the Etcd documentation.

To enable this cache plugin, the master will need the python client for Etcd installed. This can be easily installed with pip:

```
pip install python-etcd
```

Optionally, depending on the Etcd agent configuration, the following values could be set in the master config. These are the defaults:

```
etcd.host: 127.0.0.1
etcd.port: 2379
etcd.protocol: http
etcd.allow_reconnect: True
etcd.allow_redirect: False
etcd.srv_domain: None
etcd.read_timeout: 60
etcd.username: None
etcd.password: None
etcd.cert: None
etcd.ca_cert: None
```

Related docs could be found in the python-etcd documentation.

To use the etcd as a minion data cache backend, set the master `cache` config value to `etcd`:

```
cache: etcd
```

`salt.cache.etcd_cache.`**`contains`**`(`*bank, key*`)`
Checks if the specified bank contains the specified key.

`salt.cache.etcd_cache.`**`fetch`**`(`*bank, key*`)`
Fetch a key value.

`salt.cache.etcd_cache.`**`flush`**`(`*bank, key=None*`)`
Remove the key from the cache bank with all the key content.

`salt.cache.etcd_cache.`**`ls`**`(`*bank*`)`
Return an iterable object containing all entries stored in the specified bank.

`salt.cache.etcd_cache.`**`store`**`(`*bank, key, data*`)`
Store a key value.

21.3.3 salt.cache.localfs

Cache data in filesystem.

New in version 2016.11.0.

The `localfs` Minion cache module is the default cache module and does not require any configuration.

Expiration values can be set in the relevant config file (`/etc/salt/master` for the master, `/etc/salt/cloud` for Salt Cloud, etc).

`salt.cache.localfs.`**`contains`**`(`*bank, key, cachedir*`)`
Checks if the specified bank contains the specified key.

`salt.cache.localfs.`**`fetch`**`(`*bank*, *key*, *cachedir*`)`
Fetch information from a file.

`salt.cache.localfs.`**`flush`**`(`*bank*, *key=None*, *cachedir=None*`)`
Remove the key from the cache bank with all the key content.

`salt.cache.localfs.`**`get_storage_id`**`(`*kwargs*`)`

`salt.cache.localfs.`**`init_kwargs`**`(`*kwargs*`)`

`salt.cache.localfs.`**`list_`**`(`*bank*, *cachedir*`)`
Return an iterable object containing all entries stored in the specified bank.

`salt.cache.localfs.`**`store`**`(`*bank*, *key*, *data*, *cachedir*`)`
Store information in a file.

`salt.cache.localfs.`**`updated`**`(`*bank*, *key*, *cachedir*`)`
Return the epoch of the mtime for this cache file

21.3.4 salt.cache.mysql_cache

Minion data cache plugin for MySQL database.

New in version develop.

It is up to the system administrator to set up and configure the MySQL infrastructure. All is needed for this plugin is a working MySQL server.

The module requires the *salt_cache* database to exists but creates its own table if needed. The keys are indexed using the *bank* and *etcd_key* columns.

To enable this cache plugin, the master will need the python client for MySQL installed. This can be easily installed with pip:

```
pip install python-mysql
```

Optionally, depending on the MySQL agent configuration, the following values could be set in the master config. These are the defaults:

```
mysql.host: 127.0.0.1
mysql.port: 2379
mysql.user: None
mysql.password: None
mysql.database: salt_cache
mysql.table_name: cache
```

Related docs could be found in the python-mysql documentation.

To use the mysql as a minion data cache backend, set the master `cache` config value to `mysql`:

```
cache: mysql
```

`salt.cache.mysql_cache.`**`contains`**`(`*bank*, *key*`)`
Checks if the specified bank contains the specified key.

`salt.cache.mysql_cache.`**`fetch`**`(`*bank*, *key*`)`
Fetch a key value.

`salt.cache.mysql_cache.`**`flush`**`(`*bank*, *key=None*`)`
Remove the key from the cache bank with all the key content.

`salt.cache.mysql_cache.`**`ls`**`(`*bank*`)`
Return an iterable object containing all entries stored in the specified bank.

`salt.cache.mysql_cache.`**`run_query`**`(`*conn*, *query*, *retries=3*`)`
Get a cursor and run a query. Reconnect up to *retries* times if needed. Returns: cursor, affected rows counter Raises: SaltCacheError, AttributeError, OperationalError

`salt.cache.mysql_cache.`**`store`**`(`*bank*, *key*, *data*`)`
Store a key value.

21.3.5 salt.cache.redis_cache

Redis

Redis plugin for the Salt caching subsystem.

New in version 2017.7.0.

As Redis provides a simple mechanism for very fast key-value store, in order to privde the necessary features for the Salt caching subsystem, the following conventions are used:

- A Redis key consists of the bank name and the cache key separated by `/`, e.g.: `$KEY_minions/alpha/stuff` where `minions/alpha` is the bank name and `stuff` is the key name.
- As the caching subsystem is organised as a tree, we need to store the caching path and identify the bank and its offspring. At the same time, Redis is linear and we need to avoid doing `keys <pattern>` which is very inefficient as it goes through all the keys on the remote Redis server. Instead, each bank hierarchy has a Redis SET associated which stores the list of sub-banks. By default, these keys begin with `$BANK_`.
- In addition, each key name is stored in a separate SET of all the keys within a bank. By default, these SETs begin with `$BANKEYS_`.

For example, to store the key `my-key` under the bank `root-bank/sub-bank/leaf-bank`, the following hierarchy will be built:

```
127.0.0.1:6379> SMEMBERS $BANK_root-bank
1) "sub-bank"
127.0.0.1:6379> SMEMBERS $BANK_root-bank/sub-bank
1) "leaf-bank"
127.0.0.1:6379> SMEMBERS $BANKEYS_root-bank/sub-bank/leaf-bank
1) "my-key"
127.0.0.1:6379> GET $KEY_root-bank/sub-bank/leaf-bank/my-key
"my-value"
```

There are three types of keys stored:

- `$BANK_*` is a Redis SET containing the list of banks under the current bank
- `$BANKEYS_*` is a Redis SET containing the list of keys under the current bank
- `$KEY_*` keeps the value of the key

These prefixes and the separator can be adjusted using the configuration options:

bank_prefix: **`$BANK`** The prefix used for the name of the Redis key storing the list of sub-banks.

bank_keys_prefix: **`$BANKEYS`** The prefix used for the name of the Redis keyt storing the list of keys under a certain bank.

key_prefix: **`$KEY`** The prefix of the Redis keys having the value of the keys to be cached under a certain bank.

separator: **`_`** The separator between the prefix and the key body.

The connection details can be specified using:

host: **`localhost`** The hostname of the Redis server.

port: **`6379`** The Redis server port.

cluster_mode: **`False`** Whether cluster_mode is enabled or not

cluster.startup_nodes: A list of host, port dictionaries pointing to cluster members. At least one is required but multiple nodes are better

```
cache.redis.cluster.startup_nodes
  - host: redis-member-1
    port: 6379
  - host: redis-member-2
    port: 6379
```

cluster.skip_full_coverage_check: **`False`** Some cluster providers restrict certain redis commands such as CONFIG for enhanced security. Set this option to true to skip checks that required advanced privileges.

> **Note**: Most cloud hosted redis clusters will require this to be set to `True`

db: **`'0'`** The database index.

> **Note**: The database index must be specified as string not as integer value!

password: Redis connection password.

unix_socket_path:

New in version 2018.3.1.

Path to a UNIX socket for access. Overrides *host* / *port*.

Configuration Example:

```yaml
cache.redis.host: localhost
cache.redis.port: 6379
cache.redis.db: '0'
cache.redis.password: my pass
cache.redis.bank_prefix: #BANK
cache.redis.bank_keys_prefix: #BANKEYS
cache.redis.key_prefix: #KEY
cache.redis.separator: '@'
```

Cluster Configuration Example:

```yaml
cache.redis.cluster_mode: true
cache.redis.cluster.skip_full_coverage_check: true
cache.redis.cluster.startup_nodes:
  - host: redis-member-1
    port: 6379
  - host: redis-member-2
    port: 6379
cache.redis.db: '0'
cache.redis.password: my pass
cache.redis.bank_prefix: #BANK
cache.redis.bank_keys_prefix: #BANKEYS
```

(continues on next page)

(continued from previous page)

```
cache.redis.key_prefix: #KEY
cache.redis.separator: '@'
```

`salt.cache.redis_cache.`**`contains`**`(`*bank, key*`)`
Checks if the specified bank contains the specified key.

`salt.cache.redis_cache.`**`fetch`**`(`*bank, key*`)`
Fetch data from the Redis cache.

`salt.cache.redis_cache.`**`flush`**`(`*bank, key=None*`)`
Remove the key from the cache bank with all the key content. If no key is specified, remove the entire bank with all keys and sub-banks inside. This function is using the Redis pipelining for best performance. However, when removing a whole bank, in order to re-create the tree, there are a couple of requests made. In total:

- one for node in the hierarchy sub-tree, starting from the bank node
- one pipelined request to get the keys under all banks in the sub-tree
- one pipeline request to remove the corresponding keys

This is not quite optimal, as if we need to flush a bank having a very long list of sub-banks, the number of requests to build the sub-tree may grow quite big.

An improvement for this would be loading a custom Lua script in the Redis instance of the user (using the `register_script` feature) and call it whenever we flush. This script would only need to build this sub-tree causing problems. It can be added later and the behaviour should not change as the user needs to explicitly allow Salt inject scripts in their Redis instance.

`salt.cache.redis_cache.`**`init_kwargs`**`(`*kwargs*`)`

`salt.cache.redis_cache.`**`list_`**`(`*bank*`)`
Lists entries stored in the specified bank.

`salt.cache.redis_cache.`**`store`**`(`*bank, key, data*`)`
Store the data in a Redis key.

21.4 cloud modules

aliyun	AliYun ECS Cloud Module
azurearm	Azure ARM Cloud Module
cloudstack	CloudStack Cloud Module
digitalocean	DigitalOcean Cloud Module
dimensiondata	Dimension Data Cloud Module
ec2	The EC2 Cloud Module
gce	Copyright 2013 Google Inc.
gogrid	GoGrid Cloud Module
joyent	Joyent Cloud Module
linode	Linode Cloud Module using Linode's REST API
lxc	Install Salt on an LXC Container
msazure	Azure Cloud Module
oneandone	1&1 Cloud Server Module
opennebula	OpenNebula Cloud Module
openstack	Openstack Cloud Driver
parallels	Parallels Cloud Module

Continued on next page

Table 4 – continued from previous page

profitbricks	ProfitBricks Cloud Module
proxmox	Proxmox Cloud Module
pyrax	Pyrax Cloud Module
qingcloud	QingCloud Cloud Module
saltify	
scaleway	Scaleway Cloud Module
softlayer	SoftLayer Cloud Module
softlayer_hw	SoftLayer HW Cloud Module
tencentcloud	Tencent Cloud Cloud Module
vagrant	Vagrant Cloud Driver
virtualbox	A salt cloud provider that lets you use virtualbox on your machine and act as a cloud.
vmware	VMware Cloud Module
vultrpy	Vultr Cloud Module using python-vultr bindings
xen	XenServer Cloud Driver

21.4.1 salt.cloud.clouds.aliyun

AliYun ECS Cloud Module

New in version 2014.7.0.

The Aliyun cloud module is used to control access to the aliyun ECS. http://www.aliyun.com/

Use of this module requires the `id` and `key` parameter to be set. Set up the cloud configuration at `/etc/salt/cloud.providers` or `/etc/salt/cloud.providers.d/aliyun.conf`:

```
my-aliyun-config:
  # aliyun Access Key ID
  id: wFGEwgregeqw3435gDger
  # aliyun Access Key Secret
  key: GDE43t43REGTrkilg43934t34qT43t4dgegerGEgg
  location: cn-qingdao
  driver: aliyun
```

depends requests

`salt.cloud.clouds.aliyun.`**`avail_images`**(*kwargs=None, call=None*)
Return a list of the images that are on the provider

`salt.cloud.clouds.aliyun.`**`avail_locations`**(*call=None*)
Return a dict of all available VM locations on the cloud provider with relevant data

`salt.cloud.clouds.aliyun.`**`avail_sizes`**(*call=None*)
Return a list of the image sizes that are on the provider

`salt.cloud.clouds.aliyun.`**`create`**(*vm_*)
Create a single VM from a data dict

`salt.cloud.clouds.aliyun.`**`create_node`**(*kwargs*)
Convenience function to make the rest api call for node creation.

`salt.cloud.clouds.aliyun.`**`destroy`**(*name, call=None*)
Destroy a node.

CLI Example:

```
salt-cloud -a destroy myinstance
salt-cloud -d myinstance
```

`salt.cloud.clouds.aliyun.`**`get_configured_provider`**`()`
Return the first configured instance.

`salt.cloud.clouds.aliyun.`**`get_dependencies`**`()`
Warn if dependencies aren't met.

`salt.cloud.clouds.aliyun.`**`get_image`**`(`*vm_*`)`
Return the image object to use

`salt.cloud.clouds.aliyun.`**`get_location`**`(`*vm_=None*`)`

Return the aliyun region to use, in this order:

- CLI parameter
- VM parameter
- Cloud profile setting

`salt.cloud.clouds.aliyun.`**`get_securitygroup`**`(`*vm_*`)`
Return the security group

`salt.cloud.clouds.aliyun.`**`get_size`**`(`*vm_*`)`
Return the VM's size. Used by create_node().

`salt.cloud.clouds.aliyun.`**`list_availability_zones`**`(`*call=None*`)`
List all availability zones in the current region

`salt.cloud.clouds.aliyun.`**`list_monitor_data`**`(`*kwargs=None*, *call=None*`)`
Get monitor data of the instance. If instance name is missing, will show all the instance monitor data on the region.

CLI Examples:

```
salt-cloud -f list_monitor_data aliyun
salt-cloud -f list_monitor_data aliyun name=AY14051311071990225bd
```

`salt.cloud.clouds.aliyun.`**`list_nodes`**`(`*call=None*`)`
Return a list of the VMs that are on the provider

`salt.cloud.clouds.aliyun.`**`list_nodes_full`**`(`*call=None*`)`
Return a list of the VMs that are on the provider

`salt.cloud.clouds.aliyun.`**`list_nodes_min`**`(`*call=None*`)`
Return a list of the VMs that are on the provider. Only a list of VM names, and their state, is returned. This is the minimum amount of information needed to check for existing VMs.

`salt.cloud.clouds.aliyun.`**`list_nodes_select`**`(`*call=None*`)`
Return a list of the VMs that are on the provider, with select fields

`salt.cloud.clouds.aliyun.`**`list_securitygroup`**`(`*call=None*`)`
Return a list of security group

`salt.cloud.clouds.aliyun.`**`query`**`(`*params=None*`)`
Make a web call to aliyun ECS REST API

`salt.cloud.clouds.aliyun.`**`reboot`**`(`*name*, *call=None*`)`
Reboot a node

CLI Examples:

```
salt-cloud -a reboot myinstance
```

`salt.cloud.clouds.aliyun.`**`script`**`(`*vm_*`)`
Return the script deployment object

`salt.cloud.clouds.aliyun.`**`show_disk`**`(`*name, call=None*`)`
Show the disk details of the instance

CLI Examples:

```
salt-cloud -a show_disk aliyun myinstance
```

`salt.cloud.clouds.aliyun.`**`show_image`**`(`*kwargs, call=None*`)`
Show the details from aliyun image

`salt.cloud.clouds.aliyun.`**`show_instance`**`(`*name, call=None*`)`
Show the details from aliyun instance

`salt.cloud.clouds.aliyun.`**`start`**`(`*name, call=None*`)`
Start a node

CLI Examples:

```
salt-cloud -a start myinstance
```

`salt.cloud.clouds.aliyun.`**`stop`**`(`*name, force=False, call=None*`)`
Stop a node

CLI Examples:

```
salt-cloud -a stop myinstance
salt-cloud -a stop myinstance force=True
```

21.4.2 salt.cloud.clouds.azurearm

Azure ARM Cloud Module

New in version 2016.11.0.

Changed in version 2019.2.0.

The Azure ARM cloud module is used to control access to Microsoft Azure Resource Manager

depends

- azure >= 2.0.0rc6
- azure-common >= 1.1.4
- azure-mgmt >= 0.30.0rc6
- azure-mgmt-compute >= 0.33.0
- azure-mgmt-network >= 0.30.0rc6
- azure-mgmt-resource >= 0.30.0
- azure-mgmt-storage >= 0.30.0rc6
- azure-mgmt-web >= 0.30.0rc6
- azure-storage >= 0.32.0

- msrestazure >= 0.4.21

configuration Required provider parameters:

if using username and password:

- `subscription_id`
- `username`
- `password`

if using a service principal:

- `subscription_id`
- `tenant`
- `client_id`
- `secret`

Optional provider parameters:

cloud_environment: Used to point the cloud driver to different API endpoints, such as Azure GovCloud. Possible valu

- `AZURE_PUBLIC_CLOUD` (default)
- `AZURE_CHINA_CLOUD`
- `AZURE_US_GOV_CLOUD`
- `AZURE_GERMAN_CLOUD`
- HTTP base URL for a custom endpoint, such as Azure Stack. The `/metadata/endpoints` path will be added to the URL.

userdata and userdata_file: Azure Resource Manager uses a separate VirtualMachineExtension object to pass userdata scripts to the virtual machine. Arbitrary shell commands can be passed via the `userdata` parameter, or via a file local to the Salt Cloud system using the `userdata_file` parameter. Note that the local file is not treated as a script by the extension, so "one-liners" probably work best. If greater functionality is desired, a web-hosted script file can be specified via `userdata_file: https://raw.githubusercontent.com/account/repo/master/azure-script.py`, which will be executed on the system after VM creation. For Windows systems, script files ending in `.ps1` will be executed with `powershell.exe`. The `userdata` parameter takes precedence over the `userdata_file` parameter when creating the custom script extension.

win_installer: This parameter, which holds the local path to the Salt Minion installer package, is used to determine if the virtual machine type will be "Windows". Only set this parameter on profiles which install Windows operating systems.

Example `/etc/salt/cloud.providers` or `/etc/salt/cloud.providers.d/azure.conf` configuration:

```
my-azure-config with username and password:
  driver: azurearm
  subscription_id: 3287abc8-f98a-c678-3bde-326766fd3617
  username: larry
  password: 123pass

Or my-azure-config with service principal:
  driver: azurearm
```

(continues on next page)

(continued from previous page)

```
subscription_id: 3287abc8-f98a-c678-3bde-326766fd3617
tenant: ABCDEFAB-1234-ABCD-1234-ABCDEFABCDEF
client_id: ABCDEFAB-1234-ABCD-1234-ABCDEFABCDEF
secret: XXXXXXXXXXXXXXXXXXXXXXXX
cloud_environment: AZURE_US_GOV_CLOUD

The Service Principal can be created with the new Azure CLI (https://github.com/
↪Azure/azure-cli) with:
az ad sp create-for-rbac -n "http://<yourappname>" --role <role> --scopes <scope>
For example, this creates a service principal with 'owner' role for the whole
↪subscription:
az ad sp create-for-rbac -n "http://mysaltapp" --role owner --scopes /subscriptions/
↪3287abc8-f98a-c678-3bde-326766fd3617

*Note: review the details of Service Principals. Owner role is more than you
↪normally need, and you can restrict
scope to a resource group or individual resources.
```

`salt.cloud.clouds.azurearm.`**`avail_images`**(*call=None*)
Return a dict of all available images on the provider

`salt.cloud.clouds.azurearm.`**`avail_locations`**(*call=None*)
Return a dict of all available regions.

`salt.cloud.clouds.azurearm.`**`avail_sizes`**(*call=None*)
Return a list of sizes available from the provider

`salt.cloud.clouds.azurearm.`**`create`**(*vm_*)
Create a single VM from a data dict.

`salt.cloud.clouds.azurearm.`**`create_network_interface`**(*call=None, kwargs=None*)
Create a network interface.

`salt.cloud.clouds.azurearm.`**`create_or_update_vmextension`**(*call=None, kwargs=None*)
New in version 2019.2.0.

Create or update a VM extension object ”inside” of a VM object.

required kwargs:

```
extension_name: myvmextension
virtual_machine_name: myvm
settings: {"commandToExecute": "hostname"}
```

optional kwargs:

```
resource_group: < inferred from cloud configs >
location: < inferred from cloud configs >
publisher: < default: Microsoft.Azure.Extensions >
virtual_machine_extension_type: < default: CustomScript >
type_handler_version: < default: 2.0 >
auto_upgrade_minor_version: < default: True >
protected_settings: < default: None >
```

`salt.cloud.clouds.azurearm.`**`delete_blob`**(*call=None, kwargs=None*)
Delete a blob from a container.

`salt.cloud.clouds.azurearm.`**`delete_interface`**(*call=None, kwargs=None*)
Delete a network interface.

`salt.cloud.clouds.azurearm.`**`delete_managed_disk`**`(`*call=None, kwargs=None*`)`
Delete a managed disk from a resource group.

`salt.cloud.clouds.azurearm.`**`destroy`**`(`*name, call=None, kwargs=None*`)`
Destroy a VM.

CLI Examples:

```
salt-cloud -d myminion
salt-cloud -a destroy myminion service_name=myservice
```

`salt.cloud.clouds.azurearm.`**`get_api_versions`**`(`*call=None, kwargs=None*`)`
Get a resource type api versions

`salt.cloud.clouds.azurearm.`**`get_configured_provider`**`()`
Return the first configured provider instance.

`salt.cloud.clouds.azurearm.`**`get_conn`**`(`*client_type*`)`
Return a connection object for a client type.

`salt.cloud.clouds.azurearm.`**`get_dependencies`**`()`
Warn if dependencies aren't met.

`salt.cloud.clouds.azurearm.`**`get_location`**`(`*call=None, kwargs=None*`)`
Return the location that is configured for this provider

`salt.cloud.clouds.azurearm.`**`get_resource_by_id`**`(`*resource_id, api_version, extract_value=None*`)`
Get an AzureARM resource by id

`salt.cloud.clouds.azurearm.`**`list_blobs`**`(`*call=None, kwargs=None*`)`
List blobs.

`salt.cloud.clouds.azurearm.`**`list_nodes`**`(`*call=None*`)`
List VMs on this Azure account

`salt.cloud.clouds.azurearm.`**`list_nodes_full`**`(`*call=None*`)`
List all VMs on the subscription with full information

`salt.cloud.clouds.azurearm.`**`list_resource_groups`**`(`*call=None*`)`
List resource groups associated with the subscription

`salt.cloud.clouds.azurearm.`**`list_storage_accounts`**`(`*call=None*`)`
List storage accounts within the subscription.

`salt.cloud.clouds.azurearm.`**`list_subnets`**`(`*call=None, kwargs=None*`)`
List subnets in a virtual network.

`salt.cloud.clouds.azurearm.`**`list_virtual_networks`**`(`*call=None, kwargs=None*`)`
List virtual networks.

`salt.cloud.clouds.azurearm.`**`request_instance`**`(`*vm_*`)`
Request a VM from Azure.

`salt.cloud.clouds.azurearm.`**`show_instance`**`(`*name, call=None*`)`
Show the details from AzureARM concerning an instance

`salt.cloud.clouds.azurearm.`**`start`**`(`*name, call=None*`)`
New in version 2019.2.0.

Start a VM

CLI Examples:

```
salt-cloud -a start myminion
```

`salt.cloud.clouds.azurearm.`**`stop`**`(`*name*, *call=None*`)`

New in version 2019.2.0.

Stop (deallocate) a VM

CLI Examples:

```
salt-cloud -a stop myminion
```

21.4.3 salt.cloud.clouds.cloudstack

CloudStack Cloud Module

The CloudStack cloud module is used to control access to a CloudStack based Public Cloud.

depends libcloud >= 0.15

Use of this module requires the `apikey`, `secretkey`, `host` and `path` parameters.

```
my-cloudstack-cloud-config:
  apikey: <your api key >
  secretkey: <your secret key >
  host: localhost
  path: /client/api
  driver: cloudstack
```

`salt.cloud.clouds.cloudstack.`**`avail_images`**`(`*conn=None*, *call=None*`)`

Return a dict of all available VM images on the cloud provider with relevant data

`salt.cloud.clouds.cloudstack.`**`avail_locations`**`(`*conn=None*, *call=None*`)`

Return a dict of all available VM locations on the cloud provider with relevant data

`salt.cloud.clouds.cloudstack.`**`avail_sizes`**`(`*conn=None*, *call=None*`)`

Return a dict of all available VM images on the cloud provider with relevant data

`salt.cloud.clouds.cloudstack.`**`block_device_mappings`**`(`*vm_*`)`

Return the block device mapping:

```
[{'DeviceName': '/dev/sdb', 'VirtualName': 'ephemeral0'},
  {'DeviceName': '/dev/sdc', 'VirtualName': 'ephemeral1'}]
```

`salt.cloud.clouds.cloudstack.`**`cloudstack_displayname`**`(`*vm_*`)`

Return display name of VM:

:: ”minion1”

`salt.cloud.clouds.cloudstack.`**`create`**`(`*vm_*`)`

Create a single VM from a data dict

`salt.cloud.clouds.cloudstack.`**`destroy`**`(`*name*, *conn=None*, *call=None*`)`

Delete a single VM, and all of its volumes

`salt.cloud.clouds.cloudstack.`**`get_configured_provider`**`()`

Return the first configured instance.

`salt.cloud.clouds.cloudstack.`**`get_conn`**`()`

Return a conn object for the passed VM data

`salt.cloud.clouds.cloudstack.`**`get_dependencies`**`()`
Warn if dependencies aren't met.

`salt.cloud.clouds.cloudstack.`**`get_image`**`(`*conn*, *vm_*`)`
Return the image object to use

`salt.cloud.clouds.cloudstack.`**`get_ip`**`(`*data*`)`
Return the IP address of the VM If the VM has public IP as defined by libcloud module then use it Otherwise try to extract the private IP and use that one.

`salt.cloud.clouds.cloudstack.`**`get_key`**`()`
Returns the ssh private key for VM access

`salt.cloud.clouds.cloudstack.`**`get_keypair`**`(`*vm_*`)`
Return the keypair to use

`salt.cloud.clouds.cloudstack.`**`get_location`**`(`*conn*, *vm_*`)`
Return the node location to use

`salt.cloud.clouds.cloudstack.`**`get_networkid`**`(`*vm_*`)`
Return the networkid to use, only valid for Advanced Zone

`salt.cloud.clouds.cloudstack.`**`get_node`**`(`*conn*, *name*`)`
Return a libcloud node for the named VM

`salt.cloud.clouds.cloudstack.`**`get_password`**`(`*vm_*`)`
Return the password to use

`salt.cloud.clouds.cloudstack.`**`get_project`**`(`*conn*, *vm_*`)`
Return the project to use.

`salt.cloud.clouds.cloudstack.`**`get_security_groups`**`(`*conn*, *vm_*`)`
Return a list of security groups to use, defaulting to ['default']

`salt.cloud.clouds.cloudstack.`**`get_size`**`(`*conn*, *vm_*`)`
Return the VM's size object

`salt.cloud.clouds.cloudstack.`**`list_nodes`**`(`*conn=None*, *call=None*`)`
Return a list of the VMs that are on the provider

`salt.cloud.clouds.cloudstack.`**`list_nodes_full`**`(`*conn=None*, *call=None*`)`
Return a list of the VMs that are on the provider, with all fields

`salt.cloud.clouds.cloudstack.`**`list_nodes_select`**`(`*conn=None*, *call=None*`)`
Return a list of the VMs that are on the provider, with select fields

`salt.cloud.clouds.cloudstack.`**`script`**`(`*vm_*`)`
Return the script deployment object

`salt.cloud.clouds.cloudstack.`**`show_instance`**`(`*name*, *call=None*`)`
Show the details from the provider concerning an instance

21.4.4 salt.cloud.clouds.digitalocean

DigitalOcean Cloud Module

The DigitalOcean cloud module is used to control access to the DigitalOcean VPS system.

Use of this module requires a requires a `personal_access_token`, an `ssh_key_file`, and at least one SSH key name in `ssh_key_names`. More `ssh_key_names` can be added by separating each key with a comma. The `personal_access_token` can be found in the DigitalOcean web interface in the "Apps & API" section. The SSH key name can be found under the "SSH Keys" section.

```
# Note: This example is for /etc/salt/cloud.providers or any file in the
# /etc/salt/cloud.providers.d/ directory.

my-digital-ocean-config:
  personal_access_token: xxx
  ssh_key_file: /path/to/ssh/key/file
  ssh_key_names: my-key-name,my-key-name-2
  driver: digitalocean
```

depends requests

`salt.cloud.clouds.digitalocean.`**`assign_floating_ip`**(*kwargs=None, call=None*)

Assign a floating IP

New in version 2016.3.0.

CLI Examples:

```
salt-cloud -f assign_floating_ip my-digitalocean-config droplet_id=1234567␣
↪floating_ip='45.55.96.47'
```

`salt.cloud.clouds.digitalocean.`**`avail_images`**(*call=None*)

Return a list of the images that are on the provider

`salt.cloud.clouds.digitalocean.`**`avail_locations`**(*call=None*)

Return a dict of all available VM locations on the cloud provider with relevant data

`salt.cloud.clouds.digitalocean.`**`avail_sizes`**(*call=None*)

Return a list of the image sizes that are on the provider

`salt.cloud.clouds.digitalocean.`**`create`**(*vm_*)

Create a single VM from a data dict

`salt.cloud.clouds.digitalocean.`**`create_floating_ip`**(*kwargs=None, call=None*)

Create a new floating IP

New in version 2016.3.0.

CLI Examples:

```
salt-cloud -f create_floating_ip my-digitalocean-config region='NYC2'

salt-cloud -f create_floating_ip my-digitalocean-config droplet_id='1234567'
```

`salt.cloud.clouds.digitalocean.`**`create_key`**(*kwargs=None, call=None*)

Upload a public key

`salt.cloud.clouds.digitalocean.`**`create_node`**(*args*)

Create a node

`salt.cloud.clouds.digitalocean.`**`delete_floating_ip`**(*kwargs=None, call=None*)

Delete a floating IP

New in version 2016.3.0.

CLI Examples:

```
salt-cloud -f delete_floating_ip my-digitalocean-config floating_ip='45.55.96.47'
```

`salt.cloud.clouds.digitalocean.`**`destroy`**`(`*name*, *call=None*`)`

Destroy a node. Will check termination protection and warn if enabled.

CLI Example:

```
salt-cloud --destroy mymachine
```

`salt.cloud.clouds.digitalocean.`**`destroy_dns_records`**`(`*fqdn*`)`

Deletes DNS records for the given hostname if the domain is managed with DO.

`salt.cloud.clouds.digitalocean.`**`get_configured_provider`**`()`

Return the first configured instance.

`salt.cloud.clouds.digitalocean.`**`get_dependencies`**`()`

Warn if dependencies aren't met.

`salt.cloud.clouds.digitalocean.`**`get_image`**`(`*vm_*`)`

Return the image object to use

`salt.cloud.clouds.digitalocean.`**`get_keyid`**`(`*keyname*`)`

Return the ID of the keyname

`salt.cloud.clouds.digitalocean.`**`get_location`**`(`*vm_*`)`

Return the VM's location

`salt.cloud.clouds.digitalocean.`**`get_size`**`(`*vm_*`)`

Return the VM's size. Used by create_node().

`salt.cloud.clouds.digitalocean.`**`import_keypair`**`(`*kwargs=None*, *call=None*`)`

Upload public key to cloud provider. Similar to EC2 import_keypair.

New in version 2016.11.0.

kwargs file(mandatory): public key file-name keyname(mandatory): public key name in the provider

`salt.cloud.clouds.digitalocean.`**`list_floating_ips`**`(`*call=None*`)`

Return a list of the floating ips that are on the provider

New in version 2016.3.0.

CLI Examples:

```
salt-cloud -f list_floating_ips my-digitalocean-config
```

`salt.cloud.clouds.digitalocean.`**`list_keypairs`**`(`*call=None*`)`

Return a dict of all available VM locations on the cloud provider with relevant data

`salt.cloud.clouds.digitalocean.`**`list_nodes`**`(`*call=None*`)`

Return a list of the VMs that are on the provider

`salt.cloud.clouds.digitalocean.`**`list_nodes_full`**`(`*call=None*, *for_output=True*`)`

Return a list of the VMs that are on the provider

`salt.cloud.clouds.digitalocean.`**`list_nodes_select`**`(`*call=None*`)`

Return a list of the VMs that are on the provider, with select fields

`salt.cloud.clouds.digitalocean.`**`post_dns_record`**`(`***kwargs*`)`

Creates a DNS record for the given name if the domain is managed with DO.

`salt.cloud.clouds.digitalocean.`**`query`**`(`*method='droplets'*, *droplet_id=None*, *command=None*, *args=None*, *http_method='get'*`)`

Make a web call to DigitalOcean

`salt.cloud.clouds.digitalocean.`**`reboot`**`(`*name, call=None*`)`

Reboot a droplet in DigitalOcean.

New in version 2015.8.8.

name The name of the droplet to restart.

CLI Example:

```
salt-cloud -a reboot droplet_name
```

`salt.cloud.clouds.digitalocean.`**`remove_key`**`(`*kwargs=None, call=None*`)`

Delete public key

`salt.cloud.clouds.digitalocean.`**`script`**`(`*vm_*`)`

Return the script deployment object

`salt.cloud.clouds.digitalocean.`**`show_floating_ip`**`(`*kwargs=None, call=None*`)`

Show the details of a floating IP

New in version 2016.3.0.

CLI Examples:

```
salt-cloud -f show_floating_ip my-digitalocean-config floating_ip='45.55.96.47'
```

`salt.cloud.clouds.digitalocean.`**`show_instance`**`(`*name, call=None*`)`

Show the details from DigitalOcean concerning a droplet

`salt.cloud.clouds.digitalocean.`**`show_keypair`**`(`*kwargs=None, call=None*`)`

Show the details of an SSH keypair

`salt.cloud.clouds.digitalocean.`**`show_pricing`**`(`*kwargs=None, call=None*`)`

Show pricing for a particular profile. This is only an estimate, based on unofficial pricing sources.

New in version 2015.8.0.

CLI Examples:

```
salt-cloud -f show_pricing my-digitalocean-config profile=my-profile
```

`salt.cloud.clouds.digitalocean.`**`start`**`(`*name, call=None*`)`

Start a droplet in DigitalOcean.

New in version 2015.8.8.

name The name of the droplet to start.

CLI Example:

```
salt-cloud -a start droplet_name
```

`salt.cloud.clouds.digitalocean.`**`stop`**`(`*name, call=None*`)`

Stop a droplet in DigitalOcean.

New in version 2015.8.8.

name The name of the droplet to stop.

CLI Example:

```
salt-cloud -a stop droplet_name
```

`salt.cloud.clouds.digitalocean.`**`unassign_floating_ip`**`(`*kwargs=None, call=None*`)`

Unassign a floating IP

New in version 2016.3.0.

CLI Examples:

```
salt-cloud -f unassign_floating_ip my-digitalocean-config floating_ip='45.55.96.47
↪'
```

21.4.5 salt.cloud.clouds.dimensiondata

Dimension Data Cloud Module

This is a cloud module for the Dimension Data Cloud, using the existing Libcloud driver for Dimension Data.

```
# Note: This example is for /etc/salt/cloud.providers
# or any file in the
# /etc/salt/cloud.providers.d/ directory.

my-dimensiondata-config:
  user_id: my_username
  key: myPassword!
  region: dd-na
  driver: dimensiondata
```

maintainer Anthony Shaw <anthonyshaw@apache.org>

depends libcloud >= 1.2.1

`salt.cloud.clouds.dimensiondata.`**`avail_images`**`(`*conn=None, call=None*`)`

Return a dict of all available VM images on the cloud provider with relevant data

`salt.cloud.clouds.dimensiondata.`**`avail_locations`**`(`*conn=None, call=None*`)`

Return a dict of all available VM locations on the cloud provider with relevant data

`salt.cloud.clouds.dimensiondata.`**`avail_sizes`**`(`*conn=None, call=None*`)`

Return a dict of all available VM images on the cloud provider with relevant data

`salt.cloud.clouds.dimensiondata.`**`create`**`(`*vm_*`)`

Create a single VM from a data dict

`salt.cloud.clouds.dimensiondata.`**`create_lb`**`(`*kwargs=None, call=None*`)`

Create a load-balancer configuration. CLI Example:

```
salt-cloud -f create_lb dimensiondata \
    name=dev-lb port=80 protocol=http \
    members=w1,w2,w3 algorithm=ROUND_ROBIN
```

`salt.cloud.clouds.dimensiondata.`**`destroy`**`(`*name, conn=None, call=None*`)`

Delete a single VM

`salt.cloud.clouds.dimensiondata.`**`get_configured_provider`**`()`

Return the first configured instance.

`salt.cloud.clouds.dimensiondata.`**`get_conn`**`()`

Return a conn object for the passed VM data

`salt.cloud.clouds.dimensiondata.`**`get_dependencies`**`()`

Warn if dependencies aren't met.

`salt.cloud.clouds.dimensiondata.`**`get_image`**`(`*conn*, *vm_*`)`
Return the image object to use

`salt.cloud.clouds.dimensiondata.`**`get_lb_conn`**`(`*dd_driver=None*`)`
Return a load-balancer conn object

`salt.cloud.clouds.dimensiondata.`**`get_node`**`(`*conn*, *name*`)`
Return a libcloud node for the named VM

`salt.cloud.clouds.dimensiondata.`**`get_size`**`(`*conn*, *vm_*`)`
Return the VM's size object

`salt.cloud.clouds.dimensiondata.`**`list_nodes`**`(`*conn=None*, *call=None*`)`
Return a list of the VMs that are on the provider

`salt.cloud.clouds.dimensiondata.`**`list_nodes_full`**`(`*conn=None*, *call=None*`)`
Return a list of the VMs that are on the provider, with all fields

`salt.cloud.clouds.dimensiondata.`**`list_nodes_select`**`(`*conn=None*, *call=None*`)`
Return a list of the VMs that are on the provider, with select fields

`salt.cloud.clouds.dimensiondata.`**`preferred_ip`**`(`*vm_*, *ips*`)`
Return the preferred Internet protocol. Either 'ipv4' (default) or 'ipv6'.

`salt.cloud.clouds.dimensiondata.`**`reboot`**`(`*name*, *conn=None*`)`
Reboot a single VM

`salt.cloud.clouds.dimensiondata.`**`script`**`(`*vm_*`)`
Return the script deployment object

`salt.cloud.clouds.dimensiondata.`**`show_instance`**`(`*name*, *call=None*`)`
Show the details from the provider concerning an instance

`salt.cloud.clouds.dimensiondata.`**`ssh_interface`**`(`*vm_*`)`
Return the ssh_interface type to connect to. Either 'public_ips' (default) or 'private_ips'.

`salt.cloud.clouds.dimensiondata.`**`start`**`(`*name*, *call=None*`)`
Stop a VM in DimensionData.

Parameters **name** (*str*) – The name of the VM to stop.

CLI Example:

```
salt-cloud -a stop vm_name
```

`salt.cloud.clouds.dimensiondata.`**`stop`**`(`*name*, *call=None*`)`
Stop a VM in DimensionData.

name: The name of the VM to stop.

CLI Example:

```
salt-cloud -a stop vm_name
```

21.4.6 salt.cloud.clouds.ec2

The EC2 Cloud Module

The EC2 cloud module is used to interact with the Amazon Elastic Compute Cloud.

To use the EC2 cloud module, set up the cloud configuration at `/etc/salt/cloud.providers` or `/etc/salt/cloud.providers.d/ec2.conf`:

```
my-ec2-config:
  # EC2 API credentials: Access Key ID and Secret Access Key.
  # Alternatively, to use IAM Instance Role credentials available via
  # EC2 metadata set both id and key to 'use-instance-role-credentials'
  id: GKTADJGHEIQSXMKKRBJ08H
  key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs

  # If 'role_arn' is specified the above credentials are used to
  # to assume to the role. By default, role_arn is set to None.
  role_arn: arn:aws:iam::012345678910:role/SomeRoleName

  # The ssh keyname to use
  keyname: default
  # The amazon security group
  securitygroup: ssh_open
  # The location of the private key which corresponds to the keyname
  private_key: /root/default.pem

  # Be default, service_url is set to amazonaws.com. If you are using this
  # driver for something other than Amazon EC2, change it here:
  service_url: amazonaws.com

  # The endpoint that is ultimately used is usually formed using the region
  # and the service_url. If you would like to override that entirely, you
  # can explicitly define the endpoint:
  endpoint: myendpoint.example.com:1138/services/Cloud

  # SSH Gateways can be used with this provider. Gateways can be used
  # when a salt-master is not on the same private network as the instance
  # that is being deployed.

  # Defaults to None
  # Required
  ssh_gateway: gateway.example.com

  # Defaults to port 22
  # Optional
  ssh_gateway_port: 22

  # Defaults to root
  # Optional
  ssh_gateway_username: root

  # Default to nc -q0 %h %p
  # Optional
  ssh_gateway_command: "-W %h:%p"

  # One authentication method is required. If both
  # are specified, Private key wins.

  # Private key defaults to None
  ssh_gateway_private_key: /path/to/key.pem

  # Password defaults to None
  ssh_gateway_password: ExamplePasswordHere

  driver: ec2
```

(continues on next page)

(continued from previous page)

```
# Pass userdata to the instance to be created
userdata_file: /etc/salt/my-userdata-file
```

depends requests

`salt.cloud.clouds.ec2.`**`attach_volume`**(*name=None*, *kwargs=None*, *instance_id=None*, *call=None*)

Attach a volume to an instance

`salt.cloud.clouds.ec2.`**`avail_images`**(*kwargs=None*, *call=None*)

Return a dict of all available VM images on the cloud provider.

`salt.cloud.clouds.ec2.`**`avail_locations`**(*call=None*)

List all available locations

`salt.cloud.clouds.ec2.`**`avail_sizes`**(*call=None*)

Return a dict of all available VM sizes on the cloud provider with relevant data. Latest version can be found at:

http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/instance-types.html

`salt.cloud.clouds.ec2.`**`block_device_mappings`**(*vm_*)

Return the block device mapping:

```
[{'DeviceName': '/dev/sdb', 'VirtualName': 'ephemeral0'},
  {'DeviceName': '/dev/sdc', 'VirtualName': 'ephemeral1'}]
```

`salt.cloud.clouds.ec2.`**`copy_snapshot`**(*kwargs=None*, *call=None*)

Copy a snapshot

`salt.cloud.clouds.ec2.`**`create`**(*vm_=None*, *call=None*)

Create a single VM from a data dict

`salt.cloud.clouds.ec2.`**`create_attach_volumes`**(*name*, *kwargs*, *call=None*, *wait_to_finish=True*)

Create and attach volumes to created node

`salt.cloud.clouds.ec2.`**`create_keypair`**(*kwargs=None*, *call=None*)

Create an SSH keypair

`salt.cloud.clouds.ec2.`**`create_snapshot`**(*kwargs=None*, *call=None*, *wait_to_finish=False*)

Create a snapshot.

volume_id The ID of the Volume from which to create a snapshot.

description The optional description of the snapshot.

CLI Exampe:

```
salt-cloud -f create_snapshot my-ec2-config volume_id=vol-351d8826
salt-cloud -f create_snapshot my-ec2-config volume_id=vol-351d8826 \
    description="My Snapshot Description"
```

`salt.cloud.clouds.ec2.`**`create_volume`**(*kwargs=None*, *call=None*, *wait_to_finish=False*)

Create a volume.

zone The availability zone used to create the volume. Required. String.

size The size of the volume, in GiBs. Defaults to `10`. Integer.

snapshot The snapshot-id from which to create the volume. Integer.

type The volume type. This can be gp2 for General Purpose SSD, io1 for Provisioned IOPS SSD, st1 for Throughput Optimized HDD, sc1 for Cold HDD, or standard for Magnetic volumes. String.

iops The number of I/O operations per second (IOPS) to provision for the volume, with a maximum ratio of 50 IOPS/GiB. Only valid for Provisioned IOPS SSD volumes. Integer.

This option will only be set if `type` is also specified as `io1`.

encrypted Specifies whether the volume will be encrypted. Boolean.

If `snapshot` is also given in the list of kwargs, then this value is ignored since volumes that are created from encrypted snapshots are also automatically encrypted.

tags The tags to apply to the volume during creation. Dictionary.

call The `create_volume` function must be called with `-f` or `--function`. String.

wait_to_finish Whether or not to wait for the volume to be available. Boolean. Defaults to `False`.

CLI Examples:

```
salt-cloud -f create_volume my-ec2-config zone=us-east-1b
salt-cloud -f create_volume my-ec2-config zone=us-east-1b tags='{"tag1": "val1",
↪"tag2", "val2"}'
```

`salt.cloud.clouds.ec2.`**`del_tags`**(*name=None*, *kwargs=None*, *call=None*, *instance_id=None*, *resource_id=None*)

Delete tags for a resource. Normally a VM name or instance_id is passed in, but a resource_id may be passed instead. If both are passed in, the instance_id will be used.

CLI Examples:

```
salt-cloud -a del_tags mymachine tags=mytag,
salt-cloud -a del_tags mymachine tags=tag1,tag2,tag3
salt-cloud -a del_tags resource_id=vol-3267ab32 tags=tag1,tag2,tag3
```

`salt.cloud.clouds.ec2.`**`delete_keypair`**(*kwargs=None*, *call=None*)

Delete an SSH keypair

`salt.cloud.clouds.ec2.`**`delete_snapshot`**(*kwargs=None*, *call=None*)

Delete a snapshot

`salt.cloud.clouds.ec2.`**`delete_volume`**(*name=None*, *kwargs=None*, *instance_id=None*, *call=None*)

Delete a volume

`salt.cloud.clouds.ec2.`**`delvol_on_destroy`**(*name*, *kwargs=None*, *call=None*)

Delete all/specified EBS volumes upon instance termination

CLI Example:

```
salt-cloud -a delvol_on_destroy mymachine
```

`salt.cloud.clouds.ec2.`**`describe_snapshots`**(*kwargs=None*, *call=None*)

Describe a snapshot (or snapshots)

snapshot_id One or more snapshot IDs. Multiple IDs must be separated by ",".

owner Return the snapshots owned by the specified owner. Valid values include: self, amazon, <AWS Account ID>. Multiple values must be separated by ",".

restorable_by One or more AWS accounts IDs that can create volumes from the snapshot. Multiple aws account IDs must be separated by ",".

TODO: Add all of the filters.

`salt.cloud.clouds.ec2.`**`describe_volumes`**`(`*kwargs=None*, *call=None*`)`
Describe a volume (or volumes)

volume_id One or more volume IDs. Multiple IDs must be separated by ",".

TODO: Add all of the filters.

`salt.cloud.clouds.ec2.`**`destroy`**`(`*name*, *call=None*`)`
Destroy a node. Will check termination protection and warn if enabled.

CLI Example:

```
salt-cloud --destroy mymachine
```

`salt.cloud.clouds.ec2.`**`detach_volume`**`(`*name=None*, *kwargs=None*, *instance_id=None*, *call=None*`)`
Detach a volume from an instance

`salt.cloud.clouds.ec2.`**`disable_detailed_monitoring`**`(`*name*, *call=None*`)`
Enable/disable detailed monitoring on a node

CLI Example:

`salt.cloud.clouds.ec2.`**`disable_term_protect`**`(`*name*, *call=None*`)`
Disable termination protection on a node

CLI Example:

```
salt-cloud -a disable_term_protect mymachine
```

`salt.cloud.clouds.ec2.`**`enable_detailed_monitoring`**`(`*name*, *call=None*`)`
Enable/disable detailed monitoring on a node

CLI Example:

`salt.cloud.clouds.ec2.`**`enable_term_protect`**`(`*name*, *call=None*`)`
Enable termination protection on a node

CLI Example:

```
salt-cloud -a enable_term_protect mymachine
```

`salt.cloud.clouds.ec2.`**`get_availability_zone`**`(`*vm_*`)`
Return the availability zone to use

`salt.cloud.clouds.ec2.`**`get_configured_provider`**`()`
Return the first configured instance.

`salt.cloud.clouds.ec2.`**`get_console_output`**`(`*name=None*, *location=None*, *instance_id=None*, *call=None*, *kwargs=None*`)`
Show the console output from the instance.

By default, returns decoded data, not the Base64-encoded data that is actually returned from the EC2 API.

`salt.cloud.clouds.ec2.`**`get_dependencies`**`()`
Warn if dependencies aren't met.

`salt.cloud.clouds.ec2.`**`get_imageid`**`(`*vm_*`)`
Returns the ImageId to use

`salt.cloud.clouds.ec2.`**`get_location`**`(`*vm_=None*`)`

Return the EC2 region to use, in this order:

- CLI parameter
- VM parameter
- Cloud profile setting

`salt.cloud.clouds.ec2.`**`get_password_data`**`(`*name=None, kwargs=None, instance_id=None, call=None*`)`

Return password data for a Windows instance.

By default only the encrypted password data will be returned. However, if a key_file is passed in, then a decrypted password will also be returned.

Note that the key_file references the private key that was used to generate the keypair associated with this instance. This private key will _not_ be transmitted to Amazon; it is only used internally inside of Salt Cloud to decrypt data _after_ it has been received from Amazon.

CLI Examples:

```
salt-cloud -a get_password_data mymachine
salt-cloud -a get_password_data mymachine key_file=/root/ec2key.pem
```

Note: PKCS1_v1_5 was added in PyCrypto 2.5

`salt.cloud.clouds.ec2.`**`get_placementgroup`**`(`*vm_*`)`

Returns the PlacementGroup to use

`salt.cloud.clouds.ec2.`**`get_provider`**`(`*vm_=None*`)`

Extract the provider name from vm

`salt.cloud.clouds.ec2.`**`get_spot_config`**`(`*vm_*`)`

Returns the spot instance configuration for the provided vm

`salt.cloud.clouds.ec2.`**`get_ssh_gateway_config`**`(`*vm_*`)`

Return the ssh_gateway configuration.

`salt.cloud.clouds.ec2.`**`get_subnetid`**`(`*vm_*`)`

Returns the SubnetId to use

`salt.cloud.clouds.ec2.`**`get_tags`**`(`*name=None, instance_id=None, call=None, location=None, kwargs=None, resource_id=None*`)`

Retrieve tags for a resource. Normally a VM name or instance_id is passed in, but a resource_id may be passed instead. If both are passed in, the instance_id will be used.

CLI Examples:

```
salt-cloud -a get_tags mymachine
salt-cloud -a get_tags resource_id=vol-3267ab32
```

`salt.cloud.clouds.ec2.`**`get_tenancy`**`(`*vm_*`)`

Returns the Tenancy to use.

Can be ”dedicated” or ”default”. Cannot be present for spot instances.

`salt.cloud.clouds.ec2.`**`iam_profile`**`(`*vm_*`)`

Return the IAM profile.

The IAM instance profile to associate with the instances. This is either the Amazon Resource Name (ARN) of the instance profile or the name of the role.

Type: String

Default: None

Required: No

Example: arn:aws:iam::111111111111:instance-profile/s3access

Example: s3access

`salt.cloud.clouds.ec2.`**`import_keypair`**`(`*kwargs=None*, *call=None*`)`

Import an SSH public key.

New in version 2015.8.3.

`salt.cloud.clouds.ec2.`**`keepvol_on_destroy`**`(`*name*, *kwargs=None*, *call=None*`)`

Do not delete all/specified EBS volumes upon instance termination

CLI Example:

```
salt-cloud -a keepvol_on_destroy mymachine
```

`salt.cloud.clouds.ec2.`**`keyname`**`(`*vm_*`)`

Return the keyname

`salt.cloud.clouds.ec2.`**`list_nodes`**`(`*call=None*`)`

Return a list of the VMs that are on the provider

`salt.cloud.clouds.ec2.`**`list_nodes_full`**`(`*location=None*, *call=None*`)`

Return a list of the VMs that are on the provider

`salt.cloud.clouds.ec2.`**`list_nodes_min`**`(`*location=None*, *call=None*`)`

Return a list of the VMs that are on the provider. Only a list of VM names, and their state, is returned. This is the minimum amount of information needed to check for existing VMs.

`salt.cloud.clouds.ec2.`**`list_nodes_select`**`(`*call=None*`)`

Return a list of the VMs that are on the provider, with select fields

`salt.cloud.clouds.ec2.`**`optimize_providers`**`(`*providers*`)`

Return an optimized list of providers.

We want to reduce the duplication of querying the same region.

If a provider is using the same credentials for the same region the same data will be returned for each provider, thus causing un-wanted duplicate data and API calls to EC2.

`salt.cloud.clouds.ec2.`**`query`**`(`*params=None*, *setname=None*, *requesturl=None*, *location=None*, *return_url=False*, *return_root=False*`)`

`salt.cloud.clouds.ec2.`**`query_instance`**`(`*vm_=None*, *call=None*`)`

Query an instance upon creation from the EC2 API

`salt.cloud.clouds.ec2.`**`queue_instances`**`(`*instances*`)`

Queue a set of instances to be provisioned later. Expects a list.

Currently this only queries node data, and then places it in the cloud cache (if configured). If the salt-cloud-reactor is being used, these instances will be automatically provisioned using that.

For more information about the salt-cloud-reactor, see:

https://github.com/saltstack-formulas/salt-cloud-reactor

`salt.cloud.clouds.ec2.`**`reboot`**`(`*name*, *call=None*`)`

Reboot a node.

CLI Example:

```
salt-cloud -a reboot mymachine
```

salt.cloud.clouds.ec2.**register_image**(*kwargs=None*, *call=None*)

Create an ami from a snapshot

CLI Example:

```
salt-cloud -f register_image my-ec2-config ami_name=my_ami description="my
↪description"
        root_device_name=/dev/xvda snapshot_id=snap-xxxxxxxx
```

salt.cloud.clouds.ec2.**rename**(*name*, *kwargs*, *call=None*)

Properly rename a node. Pass in the new name as "new name".

CLI Example:

```
salt-cloud -a rename mymachine newname=yourmachine
```

salt.cloud.clouds.ec2.**request_instance**(*vm_=None*, *call=None*)

Put together all of the information necessary to request an instance on EC2, and then fire off the request the instance.

Returns data about the instance

salt.cloud.clouds.ec2.**script**(*vm_*)

Return the script deployment object

salt.cloud.clouds.ec2.**securitygroup**(*vm_*)

Return the security group

salt.cloud.clouds.ec2.**securitygroupid**(*vm_*)

Returns the SecurityGroupId

salt.cloud.clouds.ec2.**set_tags**(*name=None*, *tags=None*, *call=None*, *location=None*, *instance_id=None*, *resource_id=None*, *kwargs=None*)

Set tags for a resource. Normally a VM name or instance_id is passed in, but a resource_id may be passed instead. If both are passed in, the instance_id will be used.

CLI Examples:

```
salt-cloud -a set_tags mymachine tag1=somestuff tag2='Other stuff'
salt-cloud -a set_tags resource_id=vol-3267ab32 tag=somestuff
```

salt.cloud.clouds.ec2.**show_delvol_on_destroy**(*name*, *kwargs=None*, *call=None*)

Do not delete all/specified EBS volumes upon instance termination

CLI Example:

```
salt-cloud -a show_delvol_on_destroy mymachine
```

salt.cloud.clouds.ec2.**show_detailed_monitoring**(*name=None*, *instance_id=None*, *call=None*, *quiet=False*)

Show the details from EC2 regarding cloudwatch detailed monitoring.

salt.cloud.clouds.ec2.**show_image**(*kwargs*, *call=None*)

Show the details from EC2 concerning an AMI

salt.cloud.clouds.ec2.**show_instance**(*name=None*, *instance_id=None*, *call=None*, *kwargs=None*)

Show the details from EC2 concerning an AMI.

Can be called as an action (which requires a name):

```
salt-cloud -a show_instance myinstance
```

...or as a function (which requires either a name or instance_id):

```
salt-cloud -f show_instance my-ec2 name=myinstance
salt-cloud -f show_instance my-ec2 instance_id=i-d34db33f
```

salt.cloud.clouds.ec2.**show_keypair**(*kwargs=None, call=None*)
Show the details of an SSH keypair

salt.cloud.clouds.ec2.**show_pricing**(*kwargs=None, call=None*)
Show pricing for a particular profile. This is only an estimate, based on unofficial pricing sources.

CLI Examples:

```
salt-cloud -f show_pricing my-ec2-config profile=my-profile
```

If pricing sources have not been cached, they will be downloaded. Once they have been cached, they will not be updated automatically. To manually update all prices, use the following command:

```
salt-cloud -f update_pricing <provider>
```

New in version 2015.8.0.

salt.cloud.clouds.ec2.**show_term_protect**(*name=None, instance_id=None, call=None, quiet=False*)
Show the details from EC2 concerning an instance's termination protection state

salt.cloud.clouds.ec2.**show_volume**(*kwargs=None, call=None*)
Wrapper around describe_volumes. Here just to keep functionality. Might be depreciated later.

salt.cloud.clouds.ec2.**sign**(*key, msg*)

salt.cloud.clouds.ec2.**ssh_interface**(*vm_*)
Return the ssh_interface type to connect to. Either 'public_ips' (default) or 'private_ips'.

salt.cloud.clouds.ec2.**ssm_create_association**(*name=None, kwargs=None, instance_id=None, call=None*)
Associates the specified SSM document with the specified instance

http://docs.aws.amazon.com/ssm/latest/APIReference/API_CreateAssociation.html

CLI Examples:

```
salt-cloud -a ssm_create_association ec2-instance-name ssm_document=ssm-document-
↪name
```

salt.cloud.clouds.ec2.**ssm_describe_association**(*name=None, kwargs=None, instance_id=None, call=None*)
Describes the associations for the specified SSM document or instance.

http://docs.aws.amazon.com/ssm/latest/APIReference/API_DescribeAssociation.html

CLI Examples:

```
salt-cloud -a ssm_describe_association ec2-instance-name ssm_document=ssm-
↪document-name
```

salt.cloud.clouds.ec2.**start**(*name, call=None*)
Start a node

`salt.cloud.clouds.ec2.`**`stop`**`(`*name*, *call=None*`)`
Stop a node

`salt.cloud.clouds.ec2.`**`update_pricing`**`(`*kwargs=None*, *call=None*`)`
Download most recent pricing information from AWS and convert to a local JSON file.

CLI Examples:

```
salt-cloud -f update_pricing my-ec2-config
salt-cloud -f update_pricing my-ec2-config type=linux
```

New in version 2015.8.0.

`salt.cloud.clouds.ec2.`**`volume_create`**`(`***kwargs*`)`
Wrapper around create_volume. Here just to ensure the compatibility with the cloud module.

`salt.cloud.clouds.ec2.`**`volume_list`**`(`***kwargs*`)`
Wrapper around describe_volumes. Here just to ensure the compatibility with the cloud module.

`salt.cloud.clouds.ec2.`**`wait_for_instance`**`(`*vm_=None*, *data=None*, *ip_address=None*, *display_ssh_output=True*, *call=None*`)`
Wait for an instance upon creation from the EC2 API, to become available

21.4.7 salt.cloud.clouds.gce

Copyright 2013 Google Inc. All Rights Reserved.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

> http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Google Compute Engine Module

The Google Compute Engine module. This module interfaces with Google Compute Engine (GCE). To authenticate to GCE, you will need to create a Service Account. To set up Service Account Authentication, follow the *Google Compute Engine Setup* instructions.

Example Provider Configuration

```
my-gce-config:
  # The Google Cloud Platform Project ID
  project: "my-project-id"
  # The Service Account client ID
  service_account_email_address: 1234567890@developer.gserviceaccount.com
  # The location of the private key (PEM format)
  service_account_private_key: /home/erjohnso/PRIVKEY.pem
  driver: gce
  # Specify whether to use public or private IP for deploy script.
  # Valid options are:
  #     private_ips - The salt-master is also hosted with GCE
```

(continues on next page)

(continued from previous page)

```
#     public_ips - The salt-master is hosted outside of GCE
ssh_interface: public_ips
```

maintainer Eric Johnson <erjohnso@google.com>

maintainer Russell Tolle <russ.tolle@gmail.com>

depends libcloud >= 1.0.0

`salt.cloud.clouds.gce.`**`attach_disk`**`(`*name=None, kwargs=None, call=None*`)`

Attach an existing disk to an existing instance.

CLI Example:

```
salt-cloud -a attach_disk myinstance disk_name=mydisk mode=READ_WRITE
```

`salt.cloud.clouds.gce.`**`attach_lb`**`(`*kwargs=None, call=None*`)`

Add an existing node/member to an existing load-balancer configuration.

CLI Example:

```
salt-cloud -f attach_lb gce name=lb member=myinstance
```

`salt.cloud.clouds.gce.`**`avail_images`**`(`*conn=None*`)`

Return a dict of all available VM images on the cloud provider with relevant data.

Note that for GCE, there are custom images within the project, but the generic images are in other projects. This returns a dict of images in the project plus images in well-known public projects that provide supported images, as listed on this page: https://cloud.google.com/compute/docs/operating-systems/

If image names overlap, the image in the current project is used.

`salt.cloud.clouds.gce.`**`avail_locations`**`(`*conn=None, call=None*`)`

Return a dict of all available VM locations on the cloud provider with relevant data

`salt.cloud.clouds.gce.`**`avail_sizes`**`(`*conn=None*`)`

Return a dict of available instances sizes (a.k.a machine types) and convert them to something more serializable.

`salt.cloud.clouds.gce.`**`create`**`(`*vm_=None, call=None*`)`

Create a single GCE instance from a data dict.

`salt.cloud.clouds.gce.`**`create_address`**`(`*kwargs=None, call=None*`)`

Create a static address in a region.

CLI Example:

```
salt-cloud -f create_address gce name=my-ip region=us-central1 address=IP
```

`salt.cloud.clouds.gce.`**`create_attach_volumes`**`(`*name, kwargs, call=None*`)`

New in version 2017.7.0.

Create and attach multiple volumes to a node. The 'volumes' and 'node' arguments are required, where 'node' is a libcloud node, and 'volumes' is a list of maps, where each map contains:

size The size of the new disk in GB. Required.

type The disk type, either pd-standard or pd-ssd. Optional, defaults to pd-standard.

image An image to use for this new disk. Optional.

snapshot A snapshot to use for this new disk. Optional.

auto_delete An option(bool) to keep or remove the disk upon instance deletion. Optional, defaults to False.

Volumes are attached in the order in which they are given, thus on a new node the first volume will be /dev/sdb, the second /dev/sdc, and so on.

`salt.cloud.clouds.gce.`**`create_disk`**`(`*kwargs=None, call=None*`)`

Create a new persistent disk. Must specify *disk_name* and *location*, and optionally can specify 'disk_type' as pd-standard or pd-ssd, which defaults to pd-standard. Can also specify an *image* or *snapshot* but if neither of those are specified, a *size* (in GB) is required.

CLI Example:

```
salt-cloud -f create_disk gce disk_name=pd size=300 location=us-central1-b
```

`salt.cloud.clouds.gce.`**`create_fwrule`**`(`*kwargs=None, call=None*`)`

Create a GCE firewall rule. The 'default' network is used if not specified.

CLI Example:

```
salt-cloud -f create_fwrule gce name=allow-http allow=tcp:80
```

`salt.cloud.clouds.gce.`**`create_hc`**`(`*kwargs=None, call=None*`)`

Create an HTTP health check configuration.

CLI Example:

```
salt-cloud -f create_hc gce name=hc path=/healthy port=80
```

`salt.cloud.clouds.gce.`**`create_lb`**`(`*kwargs=None, call=None*`)`

Create a load-balancer configuration.

CLI Example:

```
salt-cloud -f create_lb gce name=lb region=us-central1 ports=80
```

`salt.cloud.clouds.gce.`**`create_network`**`(`*kwargs=None, call=None*`)`

… versionchanged:: 2017.7.0 Create a GCE network. Must specify name and cidr.

CLI Example:

```
salt-cloud -f create_network gce name=mynet cidr=10.10.10.0/24 mode=legacy␣
↪description=optional
salt-cloud -f create_network gce name=mynet description=optional
```

`salt.cloud.clouds.gce.`**`create_snapshot`**`(`*kwargs=None, call=None*`)`

Create a new disk snapshot. Must specify *name* and *disk_name*.

CLI Example:

```
salt-cloud -f create_snapshot gce name=snap1 disk_name=pd
```

`salt.cloud.clouds.gce.`**`create_subnetwork`**`(`*kwargs=None, call=None*`)`

… versionadded:: 2017.7.0 Create a GCE Subnetwork. Must specify name, cidr, network, and region.

CLI Example:

```
salt-cloud -f create_subnetwork gce name=mysubnet network=mynet1 region=us-west1␣
↪cidr=10.0.0.0/24 description=optional
```

`salt.cloud.clouds.gce.`**`delete_address`**`(`*kwargs=None, call=None*`)`

Permanently delete a static address.

CLI Example:

```
salt-cloud -f delete_address gce name=my-ip
```

`salt.cloud.clouds.gce.`**`delete_disk`**(*kwargs=None, call=None*)
Permanently delete a persistent disk.

CLI Example:

```
salt-cloud -f delete_disk gce disk_name=pd
```

`salt.cloud.clouds.gce.`**`delete_fwrule`**(*kwargs=None, call=None*)
Permanently delete a firewall rule.

CLI Example:

```
salt-cloud -f delete_fwrule gce name=allow-http
```

`salt.cloud.clouds.gce.`**`delete_hc`**(*kwargs=None, call=None*)
Permanently delete a health check.

CLI Example:

```
salt-cloud -f delete_hc gce name=hc
```

`salt.cloud.clouds.gce.`**`delete_lb`**(*kwargs=None, call=None*)
Permanently delete a load-balancer.

CLI Example:

```
salt-cloud -f delete_lb gce name=lb
```

`salt.cloud.clouds.gce.`**`delete_network`**(*kwargs=None, call=None*)
Permanently delete a network.

CLI Example:

```
salt-cloud -f delete_network gce name=mynet
```

`salt.cloud.clouds.gce.`**`delete_snapshot`**(*kwargs=None, call=None*)
Permanently delete a disk snapshot.

CLI Example:

```
salt-cloud -f delete_snapshot gce name=disk-snap-1
```

`salt.cloud.clouds.gce.`**`delete_subnetwork`**(*kwargs=None, call=None*)
… versionadded:: 2017.7.0 Delete a GCE Subnetwork. Must specify name and region.

CLI Example:

```
salt-cloud -f delete_subnetwork gce name=mysubnet network=mynet1 region=us-west1
```

`salt.cloud.clouds.gce.`**`destroy`**(*vm_name, call=None*)
Call 'destroy' on the instance. Can be called with "-a destroy" or -d

CLI Example:

```
salt-cloud -a destroy myinstance1 myinstance2 ...
salt-cloud -d myinstance1 myinstance2 ...
```

`salt.cloud.clouds.gce.`**`detach_disk`**`(`*name=None, kwargs=None, call=None*`)`

Detach a disk from an instance.

CLI Example:

```
salt-cloud -a detach_disk myinstance disk_name=mydisk
```

`salt.cloud.clouds.gce.`**`detach_lb`**`(`*kwargs=None, call=None*`)`

Remove an existing node/member from an existing load-balancer configuration.

CLI Example:

```
salt-cloud -f detach_lb gce name=lb member=myinstance
```

`salt.cloud.clouds.gce.`**`get_configured_provider`**`()`

Return the first configured instance.

`salt.cloud.clouds.gce.`**`get_conn`**`()`

Return a conn object for the passed VM data

`salt.cloud.clouds.gce.`**`get_dependencies`**`()`

Warn if dependencies aren't met.

`salt.cloud.clouds.gce.`**`get_lb_conn`**`(`*gce_driver=None*`)`

Return a load-balancer conn object

`salt.cloud.clouds.gce.`**`list_nodes`**`(`*conn=None, call=None*`)`

Return a list of the VMs that are on the provider

`salt.cloud.clouds.gce.`**`list_nodes_full`**`(`*conn=None, call=None*`)`

Return a list of the VMs that are on the provider, with all fields

`salt.cloud.clouds.gce.`**`list_nodes_select`**`(`*conn=None, call=None*`)`

Return a list of the VMs that are on the provider, with select fields

`salt.cloud.clouds.gce.`**`reboot`**`(`*vm_name, call=None*`)`

Call GCE 'reset' on the instance.

CLI Example:

```
salt-cloud -a reboot myinstance
```

`salt.cloud.clouds.gce.`**`request_instance`**`(`*vm_*`)`

Request a single GCE instance from a data dict.

`salt.cloud.clouds.gce.`**`script`**`(`*vm_*`)`

Return the script deployment object

`salt.cloud.clouds.gce.`**`show_address`**`(`*kwargs=None, call=None*`)`

Show the details of an existing static address.

CLI Example:

```
salt-cloud -f show_address gce name=mysnapshot region=us-central1
```

`salt.cloud.clouds.gce.`**`show_disk`**`(`*name=None, kwargs=None, call=None*`)`

Show the details of an existing disk.

CLI Example:

```
salt-cloud -a show_disk myinstance disk_name=mydisk
salt-cloud -f show_disk gce disk_name=mydisk
```

`salt.cloud.clouds.gce.`**`show_fwrule`**`(`*kwargs=None, call=None*`)`

Show the details of an existing firewall rule.

CLI Example:

```
salt-cloud -f show_fwrule gce name=allow-http
```

`salt.cloud.clouds.gce.`**`show_hc`**`(`*kwargs=None, call=None*`)`

Show the details of an existing health check.

CLI Example:

```
salt-cloud -f show_hc gce name=hc
```

`salt.cloud.clouds.gce.`**`show_instance`**`(`*vm_name, call=None*`)`

Show the details of the existing instance.

`salt.cloud.clouds.gce.`**`show_lb`**`(`*kwargs=None, call=None*`)`

Show the details of an existing load-balancer.

CLI Example:

```
salt-cloud -f show_lb gce name=lb
```

`salt.cloud.clouds.gce.`**`show_network`**`(`*kwargs=None, call=None*`)`

Show the details of an existing network.

CLI Example:

```
salt-cloud -f show_network gce name=mynet
```

`salt.cloud.clouds.gce.`**`show_pricing`**`(`*kwargs=None, call=None*`)`

Show pricing for a particular profile. This is only an estimate, based on unofficial pricing sources.

New in version 2015.8.0.

CLI Examples:

```
salt-cloud -f show_pricing my-gce-config profile=my-profile
```

`salt.cloud.clouds.gce.`**`show_snapshot`**`(`*kwargs=None, call=None*`)`

Show the details of an existing snapshot.

CLI Example:

```
salt-cloud -f show_snapshot gce name=mysnapshot
```

`salt.cloud.clouds.gce.`**`show_subnetwork`**`(`*kwargs=None, call=None*`)`

… versionadded:: 2017.7.0 Show details of an existing GCE Subnetwork. Must specify name and region.

CLI Example:

```
salt-cloud -f show_subnetwork gce name=mysubnet region=us-west1
```

`salt.cloud.clouds.gce.`**`start`**`(`*vm_name, call=None*`)`

Call GCE 'start on the instance.

New in version 2017.7.0.

CLI Example:

```
salt-cloud -a start myinstance
```

`salt.cloud.clouds.gce.`**`stop`**`(`*vm_name, call=None*`)`

Call GCE 'stop' on the instance.

New in version 2017.7.0.

CLI Example:

```
salt-cloud -a stop myinstance
```

`salt.cloud.clouds.gce.`**`update_pricing`**`(`*kwargs=None, call=None*`)`

Download most recent pricing information from GCE and save locally

CLI Examples:

```
salt-cloud -f update_pricing my-gce-config
```

New in version 2015.8.0.

21.4.8 salt.cloud.clouds.gogrid

GoGrid Cloud Module

The GoGrid cloud module. This module interfaces with the gogrid public cloud service. To use Salt Cloud with GoGrid log into the GoGrid web interface and create an api key. Do this by clicking on "My Account" and then going to the API Keys tab.

Set up the cloud configuration at `/etc/salt/cloud.providers` or `/etc/salt/cloud.providers.d/gogrid.conf`:

```
my-gogrid-config:
  # The generated api key to use
  apikey: asdff7896asdh789
  # The apikey's shared secret
  sharedsecret: saltybacon
  driver: gogrid
```

Note: A Note about using Map files with GoGrid:

Due to limitations in the GoGrid API, instances cannot be provisioned in parallel with the GoGrid driver. Map files will work with GoGrid, but the `-P` argument should not be used on maps referencing GoGrid instances.

Note: A Note about using Map files with GoGrid:

Due to limitations in the GoGrid API, instances cannot be provisioned in parallel with the GoGrid driver. Map files will work with GoGrid, but the `-P` argument should not be used on maps referencing GoGrid instances.

`salt.cloud.clouds.gogrid.`**`avail_images`**`()`

Available images

`salt.cloud.clouds.gogrid.`**`avail_locations`**`()`

Available locations

`salt.cloud.clouds.gogrid.`**`avail_sizes`**`()`
Available sizes

`salt.cloud.clouds.gogrid.`**`create`**`(`*vm_*`)`
Create a single VM from a data dict

`salt.cloud.clouds.gogrid.`**`destroy`**`(`*name*, *call=None*`)`
Destroy a machine by name

CLI Example:

```
salt-cloud -d vm_name
```

`salt.cloud.clouds.gogrid.`**`get_configured_provider`**`()`
Return the first configured instance.

`salt.cloud.clouds.gogrid.`**`list_common_lookups`**`(`*kwargs=None*, *call=None*`)`
List common lookups for a particular type of item

New in version 2015.8.0.

`salt.cloud.clouds.gogrid.`**`list_nodes`**`(`*full=False*, *call=None*`)`
List of nodes, keeping only a brief listing

CLI Example:

```
salt-cloud -Q
```

`salt.cloud.clouds.gogrid.`**`list_nodes_full`**`(`*call=None*`)`
List nodes, with all available information

CLI Example:

```
salt-cloud -F
```

`salt.cloud.clouds.gogrid.`**`list_nodes_select`**`(`*call=None*`)`
Return a list of the VMs that are on the provider, with select fields

CLI Example:

```
salt-cloud -S
```

`salt.cloud.clouds.gogrid.`**`list_passwords`**`(`*kwargs=None*, *call=None*`)`
List all password on the account

New in version 2015.8.0.

`salt.cloud.clouds.gogrid.`**`list_public_ips`**`(`*kwargs=None*, *call=None*`)`
List all available public IPs.

CLI Example: .. code-block:: bash

> salt-cloud -f list_public_ips <provider>

To list unavailable (assigned) IPs, use:

CLI Example: .. code-block:: bash

> salt-cloud -f list_public_ips <provider> state=assigned

New in version 2015.8.0.

`salt.cloud.clouds.gogrid.`**`reboot`**(*name*, *call=None*)

Reboot a machine by name

CLI Example:

```
salt-cloud -a reboot vm_name
```

New in version 2015.8.0.

`salt.cloud.clouds.gogrid.`**`show_instance`**(*name*, *call=None*)

Start a machine by name

CLI Example:

```
salt-cloud -a show_instance vm_name
```

New in version 2015.8.0.

`salt.cloud.clouds.gogrid.`**`start`**(*name*, *call=None*)

Start a machine by name

CLI Example:

```
salt-cloud -a start vm_name
```

New in version 2015.8.0.

`salt.cloud.clouds.gogrid.`**`stop`**(*name*, *call=None*)

Stop a machine by name

CLI Example:

```
salt-cloud -a stop vm_name
```

New in version 2015.8.0.

21.4.9 salt.cloud.clouds.joyent

Joyent Cloud Module

The Joyent Cloud module is used to interact with the Joyent cloud system.

Set up the cloud configuration at `/etc/salt/cloud.providers` or `/etc/salt/cloud.providers.d/joyent.conf`:

```
my-joyent-config:
  driver: joyent
  # The Joyent login user
  user: fred
  # The Joyent user's password
  password: saltybacon
  # The location of the ssh private key that can log into the new VM
  private_key: /root/mykey.pem
  # The name of the private key
  keyname: mykey
```

When creating your profiles for the joyent cloud, add the location attribute to the profile, this will automatically get picked up when performing tasks associated with that vm. An example profile might look like:

```
joyent_512:
  provider: my-joyent-config
  size: g4-highcpu-512M
  image: centos-6
  location: us-east-1
```

This driver can also be used with the Joyent SmartDataCenter project. More details can be found at:

Using SDC requires that an api_host_suffix is set. The default value for this is *.api.joyentcloud.com*. All characters, including the leading ., should be included:

```
api_host_suffix: .api.myhostname.com
```

depends PyCrypto

`salt.cloud.clouds.joyent.`**`avail_images`**`(`*call=None*`)`

Get list of available images

CLI Example:

```
salt-cloud --list-images
```

Can use a custom URL for images. Default is:

```
image_url: images.joyent.com/images
```

`salt.cloud.clouds.joyent.`**`avail_locations`**`(`*call=None*`)`

List all available locations

`salt.cloud.clouds.joyent.`**`avail_sizes`**`(`*call=None*`)`

get list of available packages

CLI Example:

```
salt-cloud --list-sizes
```

`salt.cloud.clouds.joyent.`**`create`**`(`*vm_*`)`

Create a single VM from a data dict

CLI Example:

```
salt-cloud -p profile_name vm_name
```

`salt.cloud.clouds.joyent.`**`create_node`**`(`***kwargs*`)`

convenience function to make the rest api call for node creation.

`salt.cloud.clouds.joyent.`**`delete_key`**`(`*kwargs=None, call=None*`)`

List the keys available

CLI Example:

```
salt-cloud -f delete_key joyent keyname=mykey
```

`salt.cloud.clouds.joyent.`**`destroy`**`(`*name, call=None*`)`

destroy a machine by name

Parameters

- **`name`** – name given to the machine
- **`call`** – call value in this case is 'action'

Returns array of booleans , true if successfully stopped and true if successfully removed

CLI Example:

```
salt-cloud -d vm_name
```

`salt.cloud.clouds.joyent.`**`get_configured_provider`**`()`
Return the first configured instance.

`salt.cloud.clouds.joyent.`**`get_image`**`(`*vm_*`)`
Return the image object to use

`salt.cloud.clouds.joyent.`**`get_location`**`(`*vm_=None*`)`

Return the joyent data center to use, in this order:

- CLI parameter
- VM parameter
- Cloud profile setting

`salt.cloud.clouds.joyent.`**`get_location_path`**`(`*location='us-east-1', api_host_suffix='.api.joyentcloud.com'*`)`
create url from location variable :param location: joyent data center location :return: url

`salt.cloud.clouds.joyent.`**`get_node`**`(`*name*`)`
gets the node from the full node list by name :param name: name of the vm :return: node object

`salt.cloud.clouds.joyent.`**`get_size`**`(`*vm_*`)`
Return the VM's size object

`salt.cloud.clouds.joyent.`**`has_method`**`(`*obj, method_name*`)`
Find if the provided object has a specific method

`salt.cloud.clouds.joyent.`**`import_key`**`(`*kwargs=None, call=None*`)`
List the keys available

CLI Example:

```
salt-cloud -f import_key joyent keyname=mykey keyfile=/tmp/mykey.pub
```

`salt.cloud.clouds.joyent.`**`joyent_node_state`**`(`*id_*`)`
Convert joyent returned state to state common to other data center return values for consistency

Parameters **`id`** – joyent state value

Returns state value

`salt.cloud.clouds.joyent.`**`key_list`**`(`*items=None*`)`
convert list to dictionary using the key as the identifier :param items: array to iterate over :return: dictionary

`salt.cloud.clouds.joyent.`**`list_keys`**`(`*kwargs=None, call=None*`)`
List the keys available

`salt.cloud.clouds.joyent.`**`list_nodes`**`(`*full=False, call=None*`)`
list of nodes, keeping only a brief listing

CLI Example:

```
salt-cloud -Q
```

`salt.cloud.clouds.joyent.`**`list_nodes_full`**`(`*call=None*`)`
list of nodes, maintaining all content provided from joyent listings

CLI Example:

```
salt-cloud -F
```

`salt.cloud.clouds.joyent.`**`list_nodes_select`**`(`*call=None*`)`
Return a list of the VMs that are on the provider, with select fields

`salt.cloud.clouds.joyent.`**`query`**`(`*action=None, command=None, args=None, method='GET', location=None, data=None*`)`
Make a web call to Joyent

`salt.cloud.clouds.joyent.`**`query_instance`**`(`*vm_=None, call=None*`)`
Query an instance upon creation from the Joyent API

`salt.cloud.clouds.joyent.`**`reboot`**`(`*name, call=None*`)`
reboot a machine by name :param name: name given to the machine :param call: call value in this case is 'action' :return: true if successful

CLI Example:

```
salt-cloud -a reboot vm_name
```

`salt.cloud.clouds.joyent.`**`reformat_node`**`(`*item=None, full=False*`)`
Reformat the returned data from joyent, determine public/private IPs and strip out fields if necessary to provide either full or brief content.

Parameters

- **`item`** – node dictionary
- **`full`** – full or brief output

Returns dict

`salt.cloud.clouds.joyent.`**`show_instance`**`(`*name, call=None*`)`
get details about a machine :param name: name given to the machine :param call: call value in this case is 'action' :return: machine information

CLI Example:

```
salt-cloud -a show_instance vm_name
```

`salt.cloud.clouds.joyent.`**`show_key`**`(`*kwargs=None, call=None*`)`
List the keys available

`salt.cloud.clouds.joyent.`**`ssh_interface`**`(`*vm_*`)`
Return the ssh_interface type to connect to. Either 'public_ips' (default) or 'private_ips'.

`salt.cloud.clouds.joyent.`**`start`**`(`*name, call=None*`)`
start a machine by name :param name: name given to the machine :param call: call value in this case is 'action' :return: true if successful

CLI Example:

```
salt-cloud -a start vm_name
```

`salt.cloud.clouds.joyent.`**`stop`**`(`*name, call=None*`)`
stop a machine by name :param name: name given to the machine :param call: call value in this case is 'action' :return: true if successful

CLI Example:

```
salt-cloud -a stop vm_name
```

`salt.cloud.clouds.joyent.`**`take_action`**`(`*name=None, call=None, command=None, data=None, method='GET', location='us-east-1'*`)`

take action call used by start,stop, reboot :param name: name given to the machine :param call: call value in this case is 'action' :command: api path :data: any data to be passed to the api, must be in json format :method: GET,POST,or DELETE :location: data center to execute the command on :return: true if successful

21.4.10 salt.cloud.clouds.linode

Linode Cloud Module using Linode's REST API

The Linode cloud module is used to control access to the Linode VPS system.

Use of this module only requires the `apikey` parameter. However, the default root password for new instances also needs to be set. The password needs to be 8 characters and contain lowercase, uppercase, and numbers.

Set up the cloud configuration at `/etc/salt/cloud.providers` or `/etc/salt/cloud.providers.d/linode.conf`:

```
my-linode-provider:
  apikey: f4ZsmwtB1c7f85Jdu43RgXVDFlNjuJaeIYV8QMftTqKScEB2vSosFSr...
  password: F00barbaz
  driver: linode

linode-profile:
  provider: my-linode-provider
  size: Linode 1024
  image: CentOS 7
  location: London, England, UK
```

`salt.cloud.clouds.linode.`**`avail_images`**`(`*call=None*`)`

Return available Linode images.

CLI Example:

```
salt-cloud --list-images my-linode-config
salt-cloud -f avail_images my-linode-config
```

`salt.cloud.clouds.linode.`**`avail_locations`**`(`*call=None*`)`

Return available Linode datacenter locations.

CLI Example:

```
salt-cloud --list-locations my-linode-config
salt-cloud -f avail_locations my-linode-config
```

`salt.cloud.clouds.linode.`**`avail_sizes`**`(`*call=None*`)`

Return available Linode sizes.

CLI Example:

```
salt-cloud --list-sizes my-linode-config
salt-cloud -f avail_sizes my-linode-config
```

`salt.cloud.clouds.linode.`**`boot`**`(`*name=None, kwargs=None, call=None*`)`

Boot a Linode.

name The name of the Linode to boot. Can be used instead of `linode_id`.

linode_id The ID of the Linode to boot. If provided, will be used as an alternative to `name` and reduces the number of API calls to Linode by one. Will be preferred over `name`.

config_id The ID of the Config to boot. Required.

check_running Defaults to True. If set to False, overrides the call to check if the VM is running before calling the linode.boot API call. Change `check_running` to True is useful during the boot call in the create function, since the new VM will not be running yet.

Can be called as an action (which requires a name):

```
salt-cloud -a boot my-instance config_id=10
```

...or as a function (which requires either a name or linode_id):

```
salt-cloud -f boot my-linode-config name=my-instance config_id=10
salt-cloud -f boot my-linode-config linode_id=1225876 config_id=10
```

`salt.cloud.clouds.linode.`**`clone`**(*kwargs=None*, *call=None*)

Clone a Linode.

linode_id The ID of the Linode to clone. Required.

datacenter_id The ID of the Datacenter where the Linode will be placed. Required.

plan_id The ID of the plan (size) of the Linode. Required.

CLI Example:

```
salt-cloud -f clone my-linode-config linode_id=1234567 datacenter_id=2 plan_id=5
```

`salt.cloud.clouds.linode.`**`create`**(*vm_*)

Create a single Linode VM.

`salt.cloud.clouds.linode.`**`create_config`**(*kwargs=None*, *call=None*)

Creates a Linode Configuration Profile.

name The name of the VM to create the config for.

linode_id The ID of the Linode to create the configuration for.

root_disk_id The Root Disk ID to be used for this config.

swap_disk_id The Swap Disk ID to be used for this config.

data_disk_id The Data Disk ID to be used for this config.

New in version 2016.3.0.

kernel_id The ID of the kernel to use for this configuration profile.

`salt.cloud.clouds.linode.`**`create_data_disk`**(*vm_=None*, *linode_id=None*, *data_size=None*)

Create a data disk for the linode (type is hardcoded to ext4 at the moment)

New in version 2016.3.0.

vm_ The VM profile to create the data disk for.

linode_id The ID of the Linode to create the data disk for.

data_size The size of the disk, in MB.

`salt.cloud.clouds.linode.`**`create_disk_from_distro`**(*vm_*, *linode_id*, *swap_size=None*)

Creates the disk for the Linode from the distribution.

vm_ The VM profile to create the disk for.

linode_id The ID of the Linode to create the distribution disk for. Required.

swap_size The size of the disk, in MB.

`salt.cloud.clouds.linode.`**`create_private_ip`**`(`*linode_id*`)`
Creates a private IP for the specified Linode.

linode_id The ID of the Linode to create the IP address for.

`salt.cloud.clouds.linode.`**`create_swap_disk`**`(`*vm_, linode_id, swap_size=None*`)`
Creates the disk for the specified Linode.

vm_ The VM profile to create the swap disk for.

linode_id The ID of the Linode to create the swap disk for.

swap_size The size of the disk, in MB.

`salt.cloud.clouds.linode.`**`destroy`**`(`*name, call=None*`)`
Destroys a Linode by name.

name The name of VM to be be destroyed.

CLI Example:

```
salt-cloud -d vm_name
```

`salt.cloud.clouds.linode.`**`get_config_id`**`(`*kwargs=None, call=None*`)`
Returns a config_id for a given linode.

New in version 2015.8.0.

name The name of the Linode for which to get the config_id. Can be used instead of `linode_id`.h

linode_id The ID of the Linode for which to get the config_id. Can be used instead of `name`.

CLI Example:

```
salt-cloud -f get_config_id my-linode-config name=my-linode
salt-cloud -f get_config_id my-linode-config linode_id=1234567
```

`salt.cloud.clouds.linode.`**`get_configured_provider`**`()`
Return the first configured instance.

`salt.cloud.clouds.linode.`**`get_data_disk`**`(`*vm_*`)`
Return True if a data disk is requested

New in version 2016.3.0.

`salt.cloud.clouds.linode.`**`get_data_disk_size`**`(`*vm_, swap, linode_id*`)`
Return the size of of the data disk in MB

New in version 2016.3.0.

`salt.cloud.clouds.linode.`**`get_datacenter_id`**`(`*location*`)`
Returns the Linode Datacenter ID.

location The location, or name, of the datacenter to get the ID from.

`salt.cloud.clouds.linode.`**`get_disk_size`**`(`*vm_, swap, linode_id*`)`
Returns the size of of the root disk in MB.

vm_ The VM to get the disk size for.

`salt.cloud.clouds.linode.`**`get_distribution_id`**`(`*vm_*`)`
Returns the distribution ID for a VM

vm_ The VM to get the distribution ID for

`salt.cloud.clouds.linode.`**`get_ips`**`(`*linode_id=None*`)`
Returns public and private IP addresses.

linode_id Limits the IP addresses returned to the specified Linode ID.

`salt.cloud.clouds.linode.`**`get_linode`**`(`*kwargs=None, call=None*`)`
Returns data for a single named Linode.

name The name of the Linode for which to get data. Can be used instead `linode_id`. Note this will induce an additional API call compared to using `linode_id`.

linode_id The ID of the Linode for which to get data. Can be used instead of `name`.

CLI Example:

```
salt-cloud -f get_linode my-linode-config name=my-instance
salt-cloud -f get_linode my-linode-config linode_id=1234567
```

`salt.cloud.clouds.linode.`**`get_linode_id_from_name`**`(`*name*`)`
Returns the Linode ID for a VM from the provided name.

name The name of the Linode from which to get the Linode ID. Required.

`salt.cloud.clouds.linode.`**`get_password`**`(`*vm_*`)`
Return the password to use for a VM.

vm_ The configuration to obtain the password from.

`salt.cloud.clouds.linode.`**`get_plan_id`**`(`*kwargs=None, call=None*`)`
Returns the Linode Plan ID.

label The label, or name, of the plan to get the ID from.

CLI Example:

```
salt-cloud -f get_plan_id linode label="Linode 1024"
```

`salt.cloud.clouds.linode.`**`get_private_ip`**`(`*vm_*`)`
Return True if a private ip address is requested

`salt.cloud.clouds.linode.`**`get_pub_key`**`(`*vm_*`)`
Return the SSH pubkey.

vm_ The configuration to obtain the public key from.

`salt.cloud.clouds.linode.`**`get_swap_size`**`(`*vm_*`)`
Returns the amoutn of swap space to be used in MB.

vm_ The VM profile to obtain the swap size from.

`salt.cloud.clouds.linode.`**`get_vm_size`**`(`*vm_*`)`
Returns the VM's size.

vm_ The VM to get the size for.

`salt.cloud.clouds.linode.`**`list_nodes`**`(`*call=None*`)`
Returns a list of linodes, keeping only a brief listing.

CLI Example:

```
salt-cloud -Q
salt-cloud --query
salt-cloud -f list_nodes my-linode-config
```

Note: The `image` label only displays information about the VM's distribution vendor, such as "Debian" or "RHEL" and does not display the actual image name. This is due to a limitation of the Linode API.

`salt.cloud.clouds.linode.`**`list_nodes_full`**`(`*call=None*`)`

List linodes, with all available information.

CLI Example:

```
salt-cloud -F
salt-cloud --full-query
salt-cloud -f list_nodes_full my-linode-config
```

Note: The `image` label only displays information about the VM's distribution vendor, such as "Debian" or "RHEL" and does not display the actual image name. This is due to a limitation of the Linode API.

`salt.cloud.clouds.linode.`**`list_nodes_min`**`(`*call=None*`)`

Return a list of the VMs that are on the provider. Only a list of VM names and their state is returned. This is the minimum amount of information needed to check for existing VMs.

New in version 2015.8.0.

CLI Example:

```
salt-cloud -f list_nodes_min my-linode-config
salt-cloud --function list_nodes_min my-linode-config
```

`salt.cloud.clouds.linode.`**`list_nodes_select`**`(`*call=None*`)`

Return a list of the VMs that are on the provider, with select fields.

`salt.cloud.clouds.linode.`**`reboot`**`(`*name*, *call=None*`)`

Reboot a linode.

New in version 2015.8.0.

name The name of the VM to reboot.

CLI Example:

```
salt-cloud -a reboot vm_name
```

`salt.cloud.clouds.linode.`**`show_instance`**`(`*name*, *call=None*`)`

Displays details about a particular Linode VM. Either a name or a linode_id must be provided.

New in version 2015.8.0.

name The name of the VM for which to display details.

CLI Example:

```
salt-cloud -a show_instance vm_name
```

Note: The `image` label only displays information about the VM's distribution vendor, such as "Debian" or "RHEL" and does not display the actual image name. This is due to a limitation of the Linode API.

`salt.cloud.clouds.linode.`**`show_pricing`**`(`*kwargs=None*, *call=None*`)`

Show pricing for a particular profile. This is only an estimate, based on unofficial pricing sources.

New in version 2015.8.0.

CLI Example:

```
salt-cloud -f show_pricing my-linode-config profile=my-linode-profile
```

`salt.cloud.clouds.linode.`**`start`**`(`*name*, *call=None*`)`

Start a VM in Linode.

name The name of the VM to start.

CLI Example:

```
salt-cloud -a stop vm_name
```

`salt.cloud.clouds.linode.`**`stop`**`(`*name*, *call=None*`)`

Stop a VM in Linode.

name The name of the VM to stop.

CLI Example:

```
salt-cloud -a stop vm_name
```

`salt.cloud.clouds.linode.`**`update_linode`**`(`*linode_id*, *update_args=None*`)`

Updates a Linode's properties.

linode_id The ID of the Linode to shutdown. Required.

update_args The args to update the Linode with. Must be in dictionary form.

21.4.11 salt.cloud.clouds.lxc

Install Salt on an LXC Container

New in version 2014.7.0.

Please read *core config documentation*.

`salt.cloud.clouds.lxc.`**`avail_images`**`()`

`salt.cloud.clouds.lxc.`**`create`**`(`*vm_*, *call=None*`)`

Create an lxc Container. This function is idempotent and will try to either provision or finish the provision of an lxc container.

NOTE: Most of the initialization code has been moved and merged with the lxc runner and lxc.init functions

`salt.cloud.clouds.lxc.`**`destroy`**`(`*vm_*, *call=None*`)`

Destroy a lxc container

`salt.cloud.clouds.lxc.`**`get_configured_provider`**`(`*vm_=None*`)`

Return the contextual provider of None if no configured one can be found.

`salt.cloud.clouds.lxc.`**`get_provider`**`(`*name*`)` `salt.cloud.clouds.lxc.`**`list_nodes`**(*conn=None, call=None*)

`salt.cloud.clouds.lxc.`**`list_nodes_full`**(*conn=None, call=None*)

`salt.cloud.clouds.lxc.`**`list_nodes_select`**(*call=None*)
Return a list of the VMs that are on the provider, with select fields

`salt.cloud.clouds.lxc.`**`show_instance`**(*name, call=None*)
Show the details from the provider concerning an instance

21.4.12 salt.cloud.clouds.msazure

Azure Cloud Module

The Azure cloud module is used to control access to Microsoft Azure

depends

- Microsoft Azure SDK for Python >= 1.0.2
- python-requests, for Python < 2.7.9

configuration Required provider parameters:

- `apikey`
- `certificate_path`
- `subscription_id`
- `backend`

A Management Certificate (.pem and .crt files) must be created and the .pem file placed on the same machine that salt-cloud is run from. Information on creating the pem file to use, and uploading the associated cer file can be found at:

http://www.windowsazure.com/en-us/develop/python/how-to-guides/service-management/

For users with Python < 2.7.9, `backend` must currently be set to `requests`.

Example `/etc/salt/cloud.providers` or `/etc/salt/cloud.providers.d/azure.conf` configuration:

```
my-azure-config:
  driver: azure
  subscription_id: 3287abc8-f98a-c678-3bde-326766fd3617
  certificate_path: /etc/salt/azure.pem
  management_host: management.core.windows.net
```

`salt.cloud.clouds.msazure.`**`add_input_endpoint`**(*kwargs=None, conn=None, call=None*)
New in version 2015.8.0.

Add an input endpoint to the deployment. Please note that there may be a delay before the changes show up.

CLI Example:

```
salt-cloud -f add_input_endpoint my-azure service=myservice \
    deployment=mydeployment role=myrole name=HTTP local_port=80 \
    port=80 protocol=tcp enable_direct_server_return=False \
    timeout_for_tcp_idle_connection=4
```

`salt.cloud.clouds.msazure.`**`add_management_certificate`**(*kwargs=None, conn=None, call=None*)

New in version 2015.8.0.

Add a new management certificate

CLI Example:

```
salt-cloud -f add_management_certificate my-azure public_key='...PUBKEY...' \
    thumbprint=0123456789ABCDEF data='...CERT_DATA...'
```

`salt.cloud.clouds.msazure.`**`add_service_certificate`**(*kwargs=None, conn=None, call=None*)

New in version 2015.8.0.

Add a new service certificate

CLI Example:

```
salt-cloud -f add_service_certificate my-azure name=my_service_certificate \
    data='...CERT_DATA...' certificate_format=sha1 password=verybadpass
```

`salt.cloud.clouds.msazure.`**`avail_images`**(*conn=None, call=None*)

List available images for Azure

`salt.cloud.clouds.msazure.`**`avail_locations`**(*conn=None, call=None*)

List available locations for Azure

`salt.cloud.clouds.msazure.`**`avail_sizes`**(*call=None*)

Return a list of sizes from Azure

`salt.cloud.clouds.msazure.`**`cleanup_unattached_disks`**(*kwargs=None, conn=None, call=None*)

New in version 2015.8.0.

Cleans up all disks associated with the account, which are not attached. * CAUTION * This is a destructive function with no undo button, and no ”Are you sure?” confirmation!

CLI Examples:

```
salt-cloud -f cleanup_unattached_disks my-azure name=my_disk
salt-cloud -f cleanup_unattached_disks my-azure name=my_disk delete_vhd=True
```

`salt.cloud.clouds.msazure.`**`create`**(*vm_*)

Create a single VM from a data dict

`salt.cloud.clouds.msazure.`**`create_affinity_group`**(*kwargs=None, conn=None, call=None*)

New in version 2015.8.0.

Create a new affinity group

CLI Example:

```
salt-cloud -f create_affinity_group my-azure name=my_affinity_group
```

`salt.cloud.clouds.msazure.`**`create_attach_volumes`**(*name, kwargs, call=None, wait_to_finish=True*)

Create and attach volumes to created node

`salt.cloud.clouds.msazure.`**`create_service`**(*kwargs=None, conn=None, call=None*)

New in version 2015.8.0.

Create a new hosted service

CLI Example:

```
salt-cloud -f create_service my-azure name=my_service label=my_service location=
↪'West US'
```

salt.cloud.clouds.msazure.**create_storage**(*kwargs=None*, *conn=None*, *call=None*)

New in version 2015.8.0.

Create a new storage account

CLI Example:

```
salt-cloud -f create_storage my-azure name=my_storage label=my_storage location=
↪'West US'
```

salt.cloud.clouds.msazure.**create_storage_container**(*kwargs=None*, *storage_conn=None*, *call=None*)

New in version 2015.8.0.

Create a storage container

CLI Example:

```
salt-cloud -f create_storage_container my-azure name=mycontainer
```

name: Name of container to create.

meta_name_values: Optional. A dict with name_value pairs to associate with the container as metadata. Example:{'Category':'test'}

blob_public_access: Optional. Possible values include: container, blob

fail_on_exist: Specify whether to throw an exception when the container exists.

salt.cloud.clouds.msazure.**delete_affinity_group**(*kwargs=None*, *conn=None*, *call=None*)

New in version 2015.8.0.

Delete a specific affinity group associated with the account

CLI Examples:

```
salt-cloud -f delete_affinity_group my-azure name=my_affinity_group
```

salt.cloud.clouds.msazure.**delete_disk**(*kwargs=None*, *conn=None*, *call=None*)

New in version 2015.8.0.

Delete a specific disk associated with the account

CLI Examples:

```
salt-cloud -f delete_disk my-azure name=my_disk
salt-cloud -f delete_disk my-azure name=my_disk delete_vhd=True
```

salt.cloud.clouds.msazure.**delete_input_endpoint**(*kwargs=None*, *conn=None*, *call=None*)

New in version 2015.8.0.

Delete an input endpoint from the deployment. Please note that there may be a delay before the changes show up.

CLI Example:

```
salt-cloud -f delete_input_endpoint my-azure service=myservice \
    deployment=mydeployment role=myrole name=HTTP
```

`salt.cloud.clouds.msazure.`**`delete_management_certificate`**(*kwargs=None*, *conn=None*, *call=None*)

New in version 2015.8.0.

Delete a specific certificate associated with the management

CLI Examples:

```
salt-cloud -f delete_management_certificate my-azure name=my_management_
↪certificate \
    thumbalgorithm=sha1 thumbprint=0123456789ABCDEF
```

`salt.cloud.clouds.msazure.`**`delete_service`**(*kwargs=None*, *conn=None*, *call=None*)

New in version 2015.8.0.

Delete a specific service associated with the account

CLI Examples:

```
salt-cloud -f delete_service my-azure name=my_service
```

`salt.cloud.clouds.msazure.`**`delete_service_certificate`**(*kwargs=None*, *conn=None*, *call=None*)

New in version 2015.8.0.

Delete a specific certificate associated with the service

CLI Examples:

```
salt-cloud -f delete_service_certificate my-azure name=my_service_certificate \
    thumbalgorithm=sha1 thumbprint=0123456789ABCDEF
```

`salt.cloud.clouds.msazure.`**`delete_storage`**(*kwargs=None*, *conn=None*, *call=None*)

New in version 2015.8.0.

Delete a specific storage account

CLI Examples:

```
salt-cloud -f delete_storage my-azure name=my_storage
```

`salt.cloud.clouds.msazure.`**`delete_storage_container`**(*kwargs=None*, *storage_conn=None*, *call=None*)

New in version 2015.8.0.

Delete a container associated with the storage account

CLI Example:

```
salt-cloud -f delete_storage_container my-azure name=mycontainer
```

name: Name of container to create.

fail_not_exist: Specify whether to throw an exception when the container exists.

lease_id: If specified, delete_storage_container only succeeds if the container's lease is active and matches this ID.

`salt.cloud.clouds.msazure.`**`destroy`**`(`*name, conn=None, call=None, kwargs=None*`)`

Destroy a VM

CLI Examples:

```
salt-cloud -d myminion
salt-cloud -a destroy myminion service_name=myservice
```

`salt.cloud.clouds.msazure.`**`get_affinity_group`**`(`*kwargs=None, conn=None, call=None*`)`

New in version 2015.8.0.

Show an affinity group associated with the account

CLI Example:

```
salt-cloud -f show_affinity_group my-azure service=myservice \
    deployment=mydeployment name=SSH
```

`salt.cloud.clouds.msazure.`**`get_blob`**`(`*kwargs=None, storage_conn=None, call=None*`)`

New in version 2015.8.0.

Download a blob

CLI Example:

```
salt-cloud -f get_blob my-azure container=base name=top.sls local_path=/srv/salt/
↪top.sls
salt-cloud -f get_blob my-azure container=base name=content.txt return_
↪content=True
```

container: Name of existing container.

name: Name of existing blob.

local_path: The path on the local machine to download the blob to. Either this or return_content must be specified.

return_content: Whether or not to return the content directly from the blob. If specified, must be True or False. Either this or the local_path must be specified.

snapshot: Optional. The snapshot parameter is an opaque DateTime value that, when present, specifies the blob snapshot to retrieve.

lease_id: Required if the blob has an active lease.

progress_callback: callback for progress with signature function(current, total) where current is the number of bytes transferred so far, and total is the size of the blob.

max_connections: Maximum number of parallel connections to use when the blob size exceeds 64MB. Set to 1 to download the blob chunks sequentially. Set to 2 or more to download the blob chunks in parallel. This uses more system resources but will download faster.

max_retries: Number of times to retry download of blob chunk if an error occurs.

retry_wait: Sleep time in secs between retries.

`salt.cloud.clouds.msazure.`**`get_blob_properties`**`(`*kwargs=None, storage_conn=None, call=None*`)`

New in version 2015.8.0.

Returns all user-defined metadata, standard HTTP properties, and system properties for the blob.

CLI Example:

```
salt-cloud -f show_blob_properties my-azure container=mycontainer blob=myblob
```

container: Name of existing container.

blob: Name of existing blob.

lease_id: Required if the blob has an active lease.

`salt.cloud.clouds.msazure.`**`get_blob_service_properties`**`(`*kwargs=None, storage_conn=None, call=None*`)`

New in version 2015.8.0.

Show a blob's service properties

CLI Example:

```
salt-cloud -f show_blob_service_properties my-azure
```

`salt.cloud.clouds.msazure.`**`get_configured_provider`**`()`

Return the first configured instance.

`salt.cloud.clouds.msazure.`**`get_conn`**`()`

Return a conn object for the passed VM data

`salt.cloud.clouds.msazure.`**`get_dependencies`**`()`

Warn if dependencies aren't met.

`salt.cloud.clouds.msazure.`**`get_deployment`**`(`*kwargs=None, conn=None, call=None*`)`

New in version 2015.8.0.

Return information about a deployment

CLI Example:

```
salt-cloud -f show_deployment my-azure name=my_deployment
```

`salt.cloud.clouds.msazure.`**`get_disk`**`(`*kwargs=None, conn=None, call=None*`)`

New in version 2015.8.0.

Return information about a disk

CLI Example:

```
salt-cloud -f show_disk my-azure name=my_disk
```

`salt.cloud.clouds.msazure.`**`get_input_endpoint`**`(`*kwargs=None, conn=None, call=None*`)`

New in version 2015.8.0.

Show an input endpoint associated with the deployment

CLI Example:

```
salt-cloud -f show_input_endpoint my-azure service=myservice \
    deployment=mydeployment name=SSH
```

`salt.cloud.clouds.msazure.`**`get_management_certificate`**`(`*kwargs=None, conn=None, call=None*`)`

New in version 2015.8.0.

Return information about a management_certificate

CLI Example:

```
salt-cloud -f get_management_certificate my-azure name=my_management_certificate \
    thumbalgorithm=sha1 thumbprint=0123456789ABCDEF
```

`salt.cloud.clouds.msazure.`**`get_operation_status`**(*kwargs=None*, *conn=None*, *call=None*)

New in version 2015.8.0.

Get Operation Status, based on a request ID

CLI Example:

```
salt-cloud -f get_operation_status my-azure id=0123456789abcdef0123456789abcdef
```

`salt.cloud.clouds.msazure.`**`get_service_certificate`**(*kwargs=None*, *conn=None*, *call=None*)

New in version 2015.8.0.

Return information about a service certificate

CLI Example:

```
salt-cloud -f show_service_certificate my-azure name=my_service_certificate \
    thumbalgorithm=sha1 thumbprint=0123456789ABCDEF
```

`salt.cloud.clouds.msazure.`**`get_storage`**(*kwargs=None*, *conn=None*, *call=None*)

New in version 2015.8.0.

List storage service properties

CLI Example:

```
salt-cloud -f show_storage my-azure name=my_storage
```

`salt.cloud.clouds.msazure.`**`get_storage_conn`**(*storage_account=None*, *storage_key=None*, *conn_kwargs=None*)

New in version 2015.8.0.

Return a storage_conn object for the storage account

`salt.cloud.clouds.msazure.`**`get_storage_container`**(*kwargs=None*, *storage_conn=None*, *call=None*)

New in version 2015.8.0.

Show a container associated with the storage account

CLI Example:

```
salt-cloud -f show_storage_container my-azure name=myservice
```

name: Name of container to show.

`salt.cloud.clouds.msazure.`**`get_storage_container_acl`**(*kwargs=None*, *storage_conn=None*, *call=None*)

New in version 2015.8.0.

Show a storage container's acl

CLI Example:

```
salt-cloud -f show_storage_container_acl my-azure name=myservice
```

name: Name of existing container.

lease_id: If specified, show_storage_container_acl only succeeds if the container's lease is active and matches this ID.

`salt.cloud.clouds.msazure.`**`get_storage_container_metadata`**`(`*kwargs=None*, *storage_conn=None*, *call=None*`)`

New in version 2015.8.0.

Show a storage container's metadata

CLI Example:

```
salt-cloud -f show_storage_container_metadata my-azure name=myservice
```

name: Name of container to show.

lease_id: If specified, show_storage_container_metadata only succeeds if the container's lease is active and matches this ID.

`salt.cloud.clouds.msazure.`**`get_storage_keys`**`(`*kwargs=None*, *conn=None*, *call=None*`)`

New in version 2015.8.0.

Show storage account keys

CLI Example:

```
salt-cloud -f show_storage_keys my-azure name=my_storage
```

`salt.cloud.clouds.msazure.`**`lease_storage_container`**`(`*kwargs=None*, *storage_conn=None*, *call=None*`)`

New in version 2015.8.0.

Lease a container associated with the storage account

CLI Example:

```
salt-cloud -f lease_storage_container my-azure name=mycontainer
```

name: Name of container to create.

lease_action: Required. Possible values: acquire|renew|release|break|change

lease_id: Required if the container has an active lease.

lease_duration: Specifies the duration of the lease, in seconds, or negative one (-1) for a lease that never expires. A non-infinite lease can be between 15 and 60 seconds. A lease duration cannot be changed using renew or change. For backwards compatibility, the default is 60, and the value is only used on an acquire operation.

lease_break_period: Optional. For a break operation, this is the proposed duration of seconds that the lease should continue before it is broken, between 0 and 60 seconds. This break period is only used if it is shorter than the time remaining on the lease. If longer, the time remaining on the lease is used. A new lease will not be available before the break period has expired, but the lease may be held for longer than the break period. If this header does not appear with a break operation, a fixed-duration lease breaks after the remaining lease period elapses, and an infinite lease breaks immediately.

proposed_lease_id: Optional for acquire, required for change. Proposed lease ID, in a GUID string format.

`salt.cloud.clouds.msazure.`**`list_affinity_groups`**(*kwargs=None*, *conn=None*, *call=None*)

New in version 2015.8.0.

List input endpoints associated with the deployment

CLI Example:

```
salt-cloud -f list_affinity_groups my-azure
```

`salt.cloud.clouds.msazure.`**`list_blobs`**(*kwargs=None*, *storage_conn=None*, *call=None*)

New in version 2015.8.0.

List blobs associated with the container

CLI Example:

```
salt-cloud -f list_blobs my-azure container=mycontainer
```

container: The name of the storage container

prefix: Optional. Filters the results to return only blobs whose names begin with the specified prefix.

marker: Optional. A string value that identifies the portion of the list to be returned with the next list operation. The operation returns a marker value within the response body if the list returned was not complete. The marker value may then be used in a subsequent call to request the next set of list items. The marker value is opaque to the client.

maxresults: Optional. Specifies the maximum number of blobs to return, including all BlobPrefix elements. If the request does not specify maxresults or specifies a value greater than 5,000, the server will return up to 5,000 items. Setting maxresults to a value less than or equal to zero results in error response code 400 (Bad Request).

include: Optional. Specifies one or more datasets to include in the response. To specify more than one of these options on the URI, you must separate each option with a comma. Valid values are:

snapshots: Specifies that snapshots should be included in the enumeration. Snapshots are listed from oldest to newest in the response.

metadata: Specifies that blob metadata be returned in the response.

uncommittedblobs: Specifies that blobs for which blocks have been uploaded, but which have not been committed using Put Block List (REST API), be included in the response.

copy: Version 2012-02-12 and newer. Specifies that metadata related to any current or previous Copy Blob operation should be included in the response.

delimiter: Optional. When the request includes this parameter, the operation returns a BlobPrefix element in the response body that acts as a placeholder for all blobs whose names begin with the same substring up to the appearance of the delimiter character. The delimiter may be a single character or a string.

`salt.cloud.clouds.msazure.`**`list_disks`**(*kwargs=None*, *conn=None*, *call=None*)

New in version 2015.8.0.

List disks associated with the account

CLI Example:

```
salt-cloud -f list_disks my-azure
```

`salt.cloud.clouds.msazure.`**`list_hosted_services`**(*conn=None*, *call=None*)

List VMs on this Azure account, with full information `salt.cloud.clouds.msazure.`**`list_input_endpoints`**(*kwargs=None*, *conn=None*, *call=None*)

New in version 2015.8.0.

List input endpoints associated with the deployment

CLI Example:

```
salt-cloud -f list_input_endpoints my-azure service=myservice␣
↪deployment=mydeployment
```

`salt.cloud.clouds.msazure.`**`list_management_certificates`**(*kwargs=None*, *conn=None*, *call=None*)

New in version 2015.8.0.

List management certificates associated with the subscription

CLI Example:

```
salt-cloud -f list_management_certificates my-azure name=my_management
```

`salt.cloud.clouds.msazure.`**`list_nodes`**(*conn=None*, *call=None*)

List VMs on this Azure account

`salt.cloud.clouds.msazure.`**`list_nodes_full`**(*conn=None*, *call=None*)

List VMs on this Azure account, with full information

`salt.cloud.clouds.msazure.`**`list_nodes_select`**(*conn=None*, *call=None*)

Return a list of the VMs that are on the provider, with select fields

`salt.cloud.clouds.msazure.`**`list_service_certificates`**(*kwargs=None*, *conn=None*, *call=None*)

New in version 2015.8.0.

List certificates associated with the service

CLI Example:

```
salt-cloud -f list_service_certificates my-azure name=my_service
```

`salt.cloud.clouds.msazure.`**`list_services`**(*kwargs=None*, *conn=None*, *call=None*)

New in version 2015.8.0.

List hosted services associated with the account

CLI Example:

```
salt-cloud -f list_services my-azure
```

`salt.cloud.clouds.msazure.`**`list_storage`**(*kwargs=None*, *conn=None*, *call=None*)

New in version 2015.8.0.

List storage accounts associated with the account

CLI Example:

```
salt-cloud -f list_storage my-azure
```

`salt.cloud.clouds.msazure.`**`list_storage_containers`**(*kwargs=None*, *storage_conn=None*, *call=None*)

New in version 2015.8.0.

List containers associated with the storage account

CLI Example:

```
salt-cloud -f list_storage_containers my-azure
```

`salt.cloud.clouds.msazure.`**`list_storage_services`**(*conn=None*, *call=None*)
List VMs on this Azure account, with full information

`salt.cloud.clouds.msazure.`**`list_virtual_networks`**(*kwargs=None*, *conn=None*, *call=None*)
New in version 2015.8.0.

List input endpoints associated with the deployment

CLI Example:

```
salt-cloud -f list_virtual_networks my-azure service=myservice␣
↪deployment=mydeployment
```

`salt.cloud.clouds.msazure.`**`make_blob_url`**(*kwargs=None*, *storage_conn=None*, *call=None*)
New in version 2015.8.0.

Creates the URL to access a blob

CLI Example:

```
salt-cloud -f make_blob_url my-azure container=mycontainer blob=myblob
```

container: Name of the container.

blob: Name of the blob.

account: Name of the storage account. If not specified, derives the host base from the provider configuration.

protocol: Protocol to use: 'http' or 'https'. If not specified, derives the host base from the provider configuration.

host_base: Live host base URL. If not specified, derives the host base from the provider configuration.

`salt.cloud.clouds.msazure.`**`put_blob`**(*kwargs=None*, *storage_conn=None*, *call=None*)
New in version 2015.8.0.

Upload a blob

CLI Examples:

```
salt-cloud -f put_blob my-azure container=base name=top.sls blob_path=/srv/salt/
↪top.sls
salt-cloud -f put_blob my-azure container=base name=content.txt blob_content=
↪'Some content'
```

container: Name of existing container.

name: Name of existing blob.

blob_path: The path on the local machine of the file to upload as a blob. Either this or blob_content must be specified.

blob_content: The actual content to be uploaded as a blob. Either this or blob_path must me specified.

cache_control: Optional. The Blob service stores this value but does not use or modify it.

content_language: Optional. Specifies the natural languages used by this resource.

content_md5: Optional. An MD5 hash of the blob content. This hash is used to verify the integrity of the blob during transport. When this header is specified, the storage service checks the hash that has arrived with the one that was sent. If the two hashes do not match, the operation will fail with error code 400 (Bad Request).

blob_content_type: Optional. Set the blob's content type.

blob_content_encoding: Optional. Set the blob's content encoding.

blob_content_language: Optional. Set the blob's content language.

blob_content_md5: Optional. Set the blob's MD5 hash.

blob_cache_control: Optional. Sets the blob's cache control.

meta_name_values: A dict containing name, value for metadata.

lease_id: Required if the blob has an active lease.

`salt.cloud.clouds.msazure.`**`query`**`(`*path, method='GET', data=None, params=None, header_dict=None, decode=True*`)`

Perform a query directly against the Azure REST API

`salt.cloud.clouds.msazure.`**`regenerate_storage_keys`**`(`*kwargs=None, conn=None, call=None*`)`

New in version 2015.8.0.

Regenerate storage account keys. Requires a key_type ("primary" or "secondary") to be specified.

CLI Example:

```
salt-cloud -f regenerate_storage_keys my-azure name=my_storage key_type=primary
```

`salt.cloud.clouds.msazure.`**`script`**`(`*vm_*`)`

Return the script deployment object

`salt.cloud.clouds.msazure.`**`set_blob_properties`**`(`*kwargs=None, storage_conn=None, call=None*`)`

New in version 2015.8.0.

Set a blob's properties

CLI Example:

```
salt-cloud -f set_blob_properties my-azure
```

container: Name of existing container.

blob: Name of existing blob.

blob_cache_control: Optional. Modifies the cache control string for the blob.

blob_content_type: Optional. Sets the blob's content type.

blob_content_md5: Optional. Sets the blob's MD5 hash.

blob_content_encoding: Optional. Sets the blob's content encoding.

blob_content_language: Optional. Sets the blob's content language.

lease_id: Required if the blob has an active lease.

blob_content_disposition: Optional. Sets the blob's Content-Disposition header. The Content-Disposition response header field conveys additional information about how to process the response payload, and also can be used to attach additional metadata. For example, if set to attachment, it indicates that the user-agent should not display the response, but instead show a Save As dialog with a filename other than the blob name specified.

`salt.cloud.clouds.msazure.`**`set_blob_service_properties`**`(`*kwargs=None, storage_conn=None, call=None*`)`

New in version 2015.8.0.

Sets the properties of a storage account's Blob service, including Windows Azure Storage Analytics. You can also use this operation to set the default request version for all incoming requests that do not have a version specified.

CLI Example:

```
salt-cloud -f set_blob_service_properties my-azure
```

properties: a StorageServiceProperties object.

timeout: Optional. The timeout parameter is expressed in seconds.

`salt.cloud.clouds.msazure.`**`set_storage_container_acl`**`(`*kwargs=None, storage_conn=None, call=None*`)`

New in version 2015.8.0.

Set a storage container's acl

CLI Example:

```
salt-cloud -f set_storage_container my-azure name=mycontainer
```

name: Name of existing container.

signed_identifiers: SignedIdentifers instance

blob_public_access: Optional. Possible values include: container, blob

lease_id: If specified, set_storage_container_acl only succeeds if the container's lease is active and matches this ID.

`salt.cloud.clouds.msazure.`**`set_storage_container_metadata`**`(`*kwargs=None, storage_conn=None, call=None*`)`

New in version 2015.8.0.

Set a storage container's metadata

CLI Example:

```
salt-cloud -f set_storage_container my-azure name=mycontainer \
    x_ms_meta_name_values='{"my_name": "my_value"}'
```

name: Name of existing container.

meta_name_values: A dict containing name, value for metadata. Example: {'category':'test'}

lease_id: If specified, set_storage_container_metadata only succeeds if the container's lease is active and matches this ID.

`salt.cloud.clouds.msazure.`**`show_affinity_group`**`(`*kwargs=None, conn=None, call=None*`)`

New in version 2015.8.0.

Show an affinity group associated with the account

CLI Example:

```
salt-cloud -f show_affinity_group my-azure service=myservice \
    deployment=mydeployment name=SSH
```

`salt.cloud.clouds.msazure.`**`show_blob_properties`**(*kwargs=None, storage_conn=None, call=None*)

New in version 2015.8.0.

Returns all user-defined metadata, standard HTTP properties, and system properties for the blob.

CLI Example:

```
salt-cloud -f show_blob_properties my-azure container=mycontainer blob=myblob
```

container: Name of existing container.

blob: Name of existing blob.

lease_id: Required if the blob has an active lease.

`salt.cloud.clouds.msazure.`**`show_blob_service_properties`**(*kwargs=None, storage_conn=None, call=None*)

New in version 2015.8.0.

Show a blob's service properties

CLI Example:

```
salt-cloud -f show_blob_service_properties my-azure
```

`salt.cloud.clouds.msazure.`**`show_deployment`**(*kwargs=None, conn=None, call=None*)

New in version 2015.8.0.

Return information about a deployment

CLI Example:

```
salt-cloud -f show_deployment my-azure name=my_deployment
```

`salt.cloud.clouds.msazure.`**`show_disk`**(*kwargs=None, conn=None, call=None*)

New in version 2015.8.0.

Return information about a disk

CLI Example:

```
salt-cloud -f show_disk my-azure name=my_disk
```

`salt.cloud.clouds.msazure.`**`show_input_endpoint`**(*kwargs=None, conn=None, call=None*)

New in version 2015.8.0.

Show an input endpoint associated with the deployment

CLI Example:

```
salt-cloud -f show_input_endpoint my-azure service=myservice \
    deployment=mydeployment name=SSH
```

`salt.cloud.clouds.msazure.`**`show_instance`**(*name*, *call=None*)

Show the details from the provider concerning an instance

`salt.cloud.clouds.msazure.`**`show_management_certificate`**(*kwargs=None*, *conn=None*, *call=None*)

New in version 2015.8.0.

Return information about a management_certificate

CLI Example:

```
salt-cloud -f get_management_certificate my-azure name=my_management_certificate \
    thumbalgorithm=sha1 thumbprint=0123456789ABCDEF
```

`salt.cloud.clouds.msazure.`**`show_service`**(*kwargs=None*, *conn=None*, *call=None*)

New in version 2015.8.0.

List hosted service properties

CLI Example:

```
salt-cloud -f show_service my-azure name=my_service
```

`salt.cloud.clouds.msazure.`**`show_service_certificate`**(*kwargs=None*, *conn=None*, *call=None*)

New in version 2015.8.0.

Return information about a service certificate

CLI Example:

```
salt-cloud -f show_service_certificate my-azure name=my_service_certificate \
    thumbalgorithm=sha1 thumbprint=0123456789ABCDEF
```

`salt.cloud.clouds.msazure.`**`show_storage`**(*kwargs=None*, *conn=None*, *call=None*)

New in version 2015.8.0.

List storage service properties

CLI Example:

```
salt-cloud -f show_storage my-azure name=my_storage
```

`salt.cloud.clouds.msazure.`**`show_storage_container`**(*kwargs=None*, *storage_conn=None*, *call=None*)

New in version 2015.8.0.

Show a container associated with the storage account

CLI Example:

```
salt-cloud -f show_storage_container my-azure name=myservice
```

name: Name of container to show.

`salt.cloud.clouds.msazure.`**`show_storage_container_acl`**(*kwargs=None*, *storage_conn=None*, *call=None*)

New in version 2015.8.0.

Show a storage container's acl

CLI Example:

```
salt-cloud -f show_storage_container_acl my-azure name=myservice
```

name: Name of existing container.

lease_id: If specified, show_storage_container_acl only succeeds if the container's lease is active and matches this ID.

`salt.cloud.clouds.msazure.`**`show_storage_container_metadata`**(*kwargs=None*, *storage_conn=None*, *call=None*)

New in version 2015.8.0.

Show a storage container's metadata

CLI Example:

```
salt-cloud -f show_storage_container_metadata my-azure name=myservice
```

name: Name of container to show.

lease_id: If specified, show_storage_container_metadata only succeeds if the container's lease is active and matches this ID.

`salt.cloud.clouds.msazure.`**`show_storage_keys`**(*kwargs=None*, *conn=None*, *call=None*)

New in version 2015.8.0.

Show storage account keys

CLI Example:

```
salt-cloud -f show_storage_keys my-azure name=my_storage
```

`salt.cloud.clouds.msazure.`**`update_affinity_group`**(*kwargs=None*, *conn=None*, *call=None*)

New in version 2015.8.0.

Update an affinity group's properties

CLI Example:

```
salt-cloud -f update_affinity_group my-azure name=my_group label=my_group
```

`salt.cloud.clouds.msazure.`**`update_disk`**(*kwargs=None*, *conn=None*, *call=None*)

New in version 2015.8.0.

Update a disk's properties

CLI Example:

```
salt-cloud -f update_disk my-azure name=my_disk label=my_disk
salt-cloud -f update_disk my-azure name=my_disk new_name=another_disk
```

`salt.cloud.clouds.msazure.`**`update_input_endpoint`**(*kwargs=None*, *conn=None*, *call=None*, *activity='update'*)

New in version 2015.8.0.

Update an input endpoint associated with the deployment. Please note that there may be a delay before the changes show up.

CLI Example:

```
salt-cloud -f update_input_endpoint my-azure service=myservice \
    deployment=mydeployment role=myrole name=HTTP local_port=80 \
    port=80 protocol=tcp enable_direct_server_return=False \
    timeout_for_tcp_idle_connection=4
```

`salt.cloud.clouds.msazure.`**`update_storage`**`(`*kwargs=None, conn=None, call=None*`)`

New in version 2015.8.0.

Update a storage account's properties

CLI Example:

```
salt-cloud -f update_storage my-azure name=my_storage label=my_storage
```

21.4.13 salt.cloud.clouds.oneandone

1&1 Cloud Server Module

The 1&1 SaltStack cloud module allows a 1&1 server to be automatically deployed and bootstrapped with Salt. It also has functions to create block storages and ssh keys.

depends 1and1 >= 1.2.0

The module requires the 1&1 api_token to be provided. The server should also be assigned a public LAN, a private LAN, or both along with SSH key pairs.

Set up the cloud configuration at `/etc/salt/cloud.providers` or `/etc/salt/cloud.providers.d/oneandone.conf`:

```
my-oneandone-config:
  driver: oneandone
  # The 1&1 api token
  api_token: <your-token>
  # SSH private key filename
  ssh_private_key: /path/to/private_key
  # SSH public key filename
  ssh_public_key: /path/to/public_key
```

```
my-oneandone-profile:
  provider: my-oneandone-config
  # Either provide fixed_instance_size_id or vcore, cores_per_processor, ram, and
↪hdds.
  # Size of the ID desired for the server
  fixed_instance_size: S
  # Total amount of processors
  vcore: 2
  # Number of cores per processor
  cores_per_processor: 2
  # RAM memory size in GB
  ram: 4
  # Hard disks
  hdds:
  -
    is_main: true
    size: 20
  -
```

(continues on next page)

(continued from previous page)

```
    is_main: false
    size: 20
  # ID of the appliance image that will be installed on server
  appliance_id: <ID>
  # ID of the datacenter where the server will be created
  datacenter_id: <ID>
  # Description of the server
  description: My server description
  # Password of the server. Password must contain more than 8 characters
  # using uppercase letters, numbers and other special symbols.
  password: P4$$w0rD
  # Power on server after creation - default True
  power_on: true
  # Firewall policy ID. If it is not provided, the server will assign
  # the best firewall policy, creating a new one if necessary.
  # If the parameter is sent with a 0 value, the server will be created with all
↪ports blocked.
  firewall_policy_id: <ID>
  # IP address ID
  ip_id: <ID>
  # Load balancer ID
  load_balancer_id: <ID>
  # Monitoring policy ID
  monitoring_policy_id: <ID>
```

Set `deploy` to False if Salt should not be installed on the node.

```
my-oneandone-profile:
  deploy: False
```

Create an SSH key

```
sudo salt-cloud -f create_ssh_key my-oneandone-config name='SaltTest' description=
↪'SaltTestDescription'
```

Create a block storage

```
sudo salt-cloud -f create_block_storage my-oneandone-config name='SaltTest2'
description='SaltTestDescription' size=50 datacenter_id=
↪'5091F6D8CBFEF9C26ACE957C652D5D49'
```

`salt.cloud.clouds.oneandone.`**`avail_images`**(*conn=None, call=None*)
Return a list of the server appliances that are on the provider

`salt.cloud.clouds.oneandone.`**`avail_locations`**(*conn=None, call=None*)
List available locations/datacenters for 1&1

`salt.cloud.clouds.oneandone.`**`avail_sizes`**(*call=None*)
Return a dict of all available VM sizes on the cloud provider with relevant data.

`salt.cloud.clouds.oneandone.`**`create`**(*vm_*)
Create a single VM from a data dict

`salt.cloud.clouds.oneandone.`**`create_block_storage`**(*kwargs=None, call=None*)
Create a block storage

`salt.cloud.clouds.oneandone.`**`create_ssh_key`**(*kwargs=None, call=None*)
Create an ssh key

`salt.cloud.clouds.oneandone.`**`destroy`**`(`*name*, *call=None*`)`

destroy a server by name

Parameters

- **`name`** – name given to the server
- **`call`** – call value in this case is 'action'

Returns array of booleans , true if successfully stopped and true if successfully removed

CLI Example:

```
salt-cloud -d vm_name
```

`salt.cloud.clouds.oneandone.`**`get_configured_provider`**`()`

Return the first configured instance.

`salt.cloud.clouds.oneandone.`**`get_conn`**`()`

Return a conn object for the passed VM data

`salt.cloud.clouds.oneandone.`**`get_dependencies`**`()`

Warn if dependencies are not met.

`salt.cloud.clouds.oneandone.`**`get_image`**`(`*vm_*`)`

Return the image object to use

`salt.cloud.clouds.oneandone.`**`get_key_filename`**`(`*vm_*`)`

Check SSH private key file and return absolute path if exists.

`salt.cloud.clouds.oneandone.`**`get_node`**`(`*conn*, *name*`)`

Return a node for the named VM

`salt.cloud.clouds.oneandone.`**`get_size`**`(`*vm_*`)`

Return the VM's size object

`salt.cloud.clouds.oneandone.`**`get_wait_timeout`**`(`*vm_*`)`

Return the wait_for_timeout for resource provisioning.

`salt.cloud.clouds.oneandone.`**`list_nodes`**`(`*conn=None*, *call=None*`)`

Return a list of VMs that are on the provider

`salt.cloud.clouds.oneandone.`**`list_nodes_full`**`(`*conn=None*, *call=None*`)`

Return a list of the VMs that are on the provider, with all fields

`salt.cloud.clouds.oneandone.`**`list_nodes_select`**`(`*conn=None*, *call=None*`)`

Return a list of the VMs that are on the provider, with select fields

`salt.cloud.clouds.oneandone.`**`load_public_key`**`(`*vm_*`)`

Load the public key file if exists.

`salt.cloud.clouds.oneandone.`**`reboot`**`(`*name*, *call=None*`)`

reboot a server by name :param name: name given to the machine :param call: call value in this case is 'action' :return: true if successful

CLI Example:

```
salt-cloud -a reboot vm_name
```

`salt.cloud.clouds.oneandone.`**`script`**`(`*vm_*`)`

Return the script deployment object

`salt.cloud.clouds.oneandone.`**`show_instance`**`(`*name*, *call=None*`)`

Show the details from the provider concerning an instance `salt.cloud.clouds.oneandone.`**`start`**`(`*name*, *call=None*`)`

start a server by name :param name: name given to the machine :param call: call value in this case is 'action' :return: true if successful

CLI Example:

```
salt-cloud -a start vm_name
```

`salt.cloud.clouds.oneandone.`**`stop`**`(`*name*, *call=None*`)`

stop a server by name :param name: name given to the machine :param call: call value in this case is 'action' :return: true if successful

CLI Example:

```
salt-cloud -a stop vm_name
```

21.4.14 salt.cloud.clouds.opennebula

OpenNebula Cloud Module

The OpenNebula cloud module is used to control access to an OpenNebula cloud.

New in version 2014.7.0.

depends lxml

depends OpenNebula installation running version `4.14` or later.

Use of this module requires the `xml_rpc`, `user`, and `password` parameters to be set.

Set up the cloud configuration at `/etc/salt/cloud.providers` or `/etc/salt/cloud.providers.d/opennebula.conf`:

```
my-opennebula-config:
  xml_rpc: http://localhost:2633/RPC2
  user: oneadmin
  password: JHGhgsayu32jsa
  driver: opennebula
```

This driver supports accessing new VM instances via DNS entry instead of IP address. To enable this feature, in the provider or profile file add *fqdn_base* with a value matching the base of your fully-qualified domain name. Example:

```
my-opennebula-config:
  [...]
  fqdn_base: <my.basedomain.com>
  [...]
```

The driver will prepend the hostname to the fqdn_base and do a DNS lookup to find the IP of the new VM.

```
salt-cloud -f image_allocate opennebula datastore_name=default \
    data='NAME="My New Image" DESCRIPTION="Description of the image." \
    PATH=/home/one_user/images/image_name.img'
salt-cloud -f secgroup_allocate opennebula \
    data="Name = test RULE = [PROTOCOL = TCP, RULE_TYPE = inbound, \
    RANGE = 1000:2000]"
```

`salt.cloud.clouds.opennebula.`**`avail_images`**`(`*call=None*`)`

Return available OpenNebula images.

CLI Example:

```
salt-cloud --list-images opennebula
salt-cloud --function avail_images opennebula
salt-cloud -f avail_images opennebula
```

`salt.cloud.clouds.opennebula.`**`avail_locations`**(*call=None*)

Return available OpenNebula locations.

CLI Example:

```
salt-cloud --list-locations opennebula
salt-cloud --function avail_locations opennebula
salt-cloud -f avail_locations opennebula
```

`salt.cloud.clouds.opennebula.`**`avail_sizes`**(*call=None*)

Because sizes are built into templates with OpenNebula, there will be no sizes to return here.

`salt.cloud.clouds.opennebula.`**`create`**(*vm_*)

Create a single VM from a data dict.

vm_ The dictionary use to create a VM.

Optional vm_ dict options for overwriting template:

region_id Optional - OpenNebula Zone ID

memory Optional - In MB

cpu Optional - Percent of host CPU to allocate

vcpu

Optional - Amount of vCPUs to allocate

CLI Example:

```
 salt-cloud -p my-opennebula-profile vm_name

salt-cloud -p my-opennebula-profile vm_name memory=16384 cpu=2.5 vcpu=16
```

`salt.cloud.clouds.opennebula.`**`destroy`**(*name, call=None*)

Destroy a node. Will check termination protection and warn if enabled.

name The name of the vm to be destroyed.

CLI Example:

```
salt-cloud --destroy vm_name
salt-cloud -d vm_name
salt-cloud --action destroy vm_name
salt-cloud -a destroy vm_name
```

`salt.cloud.clouds.opennebula.`**`get_cluster_id`**(*kwargs=None, call=None*)

Returns a cluster's ID from the given cluster name.

New in version 2016.3.0.

CLI Example:

```
salt-cloud -f get_cluster_id opennebula name=my-cluster-name
```

`salt.cloud.clouds.opennebula.`**`get_configured_provider`**`()`

Return the first configured instance.

`salt.cloud.clouds.opennebula.`**`get_datastore_id`**`(`*kwargs=None*, *call=None*`)`

Returns a data store's ID from the given data store name.

New in version 2016.3.0.

CLI Example:

```
salt-cloud -f get_datastore_id opennebula name=my-datastore-name
```

`salt.cloud.clouds.opennebula.`**`get_dependencies`**`()`

Warn if dependencies aren't met.

`salt.cloud.clouds.opennebula.`**`get_host_id`**`(`*kwargs=None*, *call=None*`)`

Returns a host's ID from the given host name.

New in version 2016.3.0.

CLI Example:

```
salt-cloud -f get_host_id opennebula name=my-host-name
```

`salt.cloud.clouds.opennebula.`**`get_image`**`(`*vm_*`)`

Return the image object to use.

vm_ The VM dictionary for which to obtain an image.

`salt.cloud.clouds.opennebula.`**`get_image_id`**`(`*kwargs=None*, *call=None*`)`

Returns an image's ID from the given image name.

New in version 2016.3.0.

CLI Example:

```
salt-cloud -f get_image_id opennebula name=my-image-name
```

`salt.cloud.clouds.opennebula.`**`get_location`**`(`*vm_*`)`

Return the VM's location.

vm_ The VM dictionary for which to obtain a location.

`salt.cloud.clouds.opennebula.`**`get_one_version`**`(`*kwargs=None*, *call=None*`)`

Returns the OpenNebula version.

New in version 2016.3.5.

CLI Example:

```
salt-cloud -f get_one_version one_provider_name
```

`salt.cloud.clouds.opennebula.`**`get_secgroup_id`**`(`*kwargs=None*, *call=None*`)`

Returns a security group's ID from the given security group name.

New in version 2016.3.0.

CLI Example:

```
salt-cloud -f get_secgroup_id opennebula name=my-secgroup-name
```

`salt.cloud.clouds.opennebula.`**`get_template`**`(`*vm_*`)`

Return the template id for a VM.

New in version 2016.11.0.

vm_ The VM dictionary for which to obtain a template.

`salt.cloud.clouds.opennebula.`**`get_template_id`**`(`*kwargs=None, call=None*`)`

Returns a template's ID from the given template name.

New in version 2016.3.0.

CLI Example:

```
salt-cloud -f get_template_id opennebula name=my-template-name
```

`salt.cloud.clouds.opennebula.`**`get_template_image`**`(`*kwargs=None, call=None*`)`

Returns a template's image from the given template name.

New in version 2018.3.0.

```
salt-cloud -f get_template_image opennebula name=my-template-name
```

`salt.cloud.clouds.opennebula.`**`get_vm_id`**`(`*kwargs=None, call=None*`)`

Returns a virtual machine's ID from the given virtual machine's name.

New in version 2016.3.0.

CLI Example:

```
salt-cloud -f get_vm_id opennebula name=my-vm
```

`salt.cloud.clouds.opennebula.`**`get_vn_id`**`(`*kwargs=None, call=None*`)`

Returns a virtual network's ID from the given virtual network's name.

New in version 2016.3.0.

CLI Example:

```
salt-cloud -f get_vn_id opennebula name=my-vn-name
```

`salt.cloud.clouds.opennebula.`**`image_allocate`**`(`*call=None, kwargs=None*`)`

Allocates a new image in OpenNebula.

New in version 2016.3.0.

path The path to a file containing the template of the image to allocate. Syntax within the file can be the usual attribute=value or XML. Can be used instead of `data`.

data The data containing the template of the image to allocate. Syntax can be the usual attribute=value or XML. Can be used instead of `path`.

datastore_id The ID of the data-store to be used for the new image. Can be used instead of `datastore_name`.

datastore_name The name of the data-store to be used for the new image. Can be used instead of `datastore_id`.

CLI Example:

```
salt-cloud -f image_allocate opennebula path=/path/to/image_file.txt datastore_
↪id=1
salt-cloud -f image_allocate opennebula datastore_name=default \
    data='NAME="Ubuntu 14.04" PATH="/home/one_user/images/ubuntu_desktop.img" \
    DESCRIPTION="Ubuntu 14.04 for development."'
```

`salt.cloud.clouds.opennebula.`**`image_clone`**`(`*call=None*, *kwargs=None*`)`

Clones an existing image.

New in version 2016.3.0.

name The name of the new image.

image_id The ID of the image to be cloned. Can be used instead of `image_name`.

image_name The name of the image to be cloned. Can be used instead of `image_id`.

CLI Example:

```
salt-cloud -f image_clone opennebula name=my-new-image image_id=10
salt-cloud -f image_clone opennebula name=my-new-image image_name=my-image-to-
↪clone
```

`salt.cloud.clouds.opennebula.`**`image_delete`**`(`*call=None*, *kwargs=None*`)`

Deletes the given image from OpenNebula. Either a name or an image_id must be supplied.

New in version 2016.3.0.

name The name of the image to delete. Can be used instead of `image_id`.

image_id The ID of the image to delete. Can be used instead of `name`.

CLI Example:

```
salt-cloud -f image_delete opennebula name=my-image
salt-cloud --function image_delete opennebula image_id=100
```

`salt.cloud.clouds.opennebula.`**`image_info`**`(`*call=None*, *kwargs=None*`)`

Retrieves information for a given image. Either a name or an image_id must be supplied.

New in version 2016.3.0.

name The name of the image for which to gather information. Can be used instead of `image_id`.

image_id The ID of the image for which to gather information. Can be used instead of `name`.

CLI Example:

```
salt-cloud -f image_info opennebula name=my-image
salt-cloud --function image_info opennebula image_id=5
```

`salt.cloud.clouds.opennebula.`**`image_persistent`**`(`*call=None*, *kwargs=None*`)`

Sets the Image as persistent or not persistent.

New in version 2016.3.0.

name The name of the image to set. Can be used instead of `image_id`.

image_id The ID of the image to set. Can be used instead of `name`.

persist A boolean value to set the image as persistent or not. Set to true for persistent, false for non-persistent.

CLI Example:

```
salt-cloud -f image_persistent opennebula name=my-image persist=True
salt-cloud --function image_persistent opennebula image_id=5 persist=False
```

`salt.cloud.clouds.opennebula.`**`image_snapshot_delete`**`(`*call=None*, *kwargs=None*`)`

Deletes a snapshot from the image.

New in version 2016.3.0.

image_id The ID of the image from which to delete the snapshot. Can be used instead of `image_name`.

image_name The name of the image from which to delete the snapshot. Can be used instead of `image_id`.

snapshot_id The ID of the snapshot to delete.

CLI Example:

```
salt-cloud -f image_snapshot_delete vm_id=106 snapshot_id=45
salt-cloud -f image_snapshot_delete vm_name=my-vm snapshot_id=111
```

`salt.cloud.clouds.opennebula.`**`image_snapshot_flatten`**(*call=None*, *kwargs=None*)

Flattens the snapshot of an image and discards others.

New in version 2016.3.0.

image_id The ID of the image. Can be used instead of `image_name`.

image_name The name of the image. Can be used instead of `image_id`.

snapshot_id The ID of the snapshot to flatten.

CLI Example:

```
salt-cloud -f image_snapshot_flatten vm_id=106 snapshot_id=45
salt-cloud -f image_snapshot_flatten vm_name=my-vm snapshot_id=45
```

`salt.cloud.clouds.opennebula.`**`image_snapshot_revert`**(*call=None*, *kwargs=None*)

Reverts an image state to a previous snapshot.

New in version 2016.3.0.

image_id The ID of the image to revert. Can be used instead of `image_name`.

image_name The name of the image to revert. Can be used instead of `image_id`.

snapshot_id The ID of the snapshot to which the image will be reverted.

CLI Example:

```
salt-cloud -f image_snapshot_revert vm_id=106 snapshot_id=45
salt-cloud -f image_snapshot_revert vm_name=my-vm snapshot_id=120
```

`salt.cloud.clouds.opennebula.`**`image_update`**(*call=None*, *kwargs=None*)

Replaces the image template contents.

New in version 2016.3.0.

image_id The ID of the image to update. Can be used instead of `image_name`.

image_name The name of the image to update. Can be used instead of `image_id`.

path The path to a file containing the template of the image. Syntax within the file can be the usual attribute=value or XML. Can be used instead of `data`.

data Contains the template of the image. Syntax can be the usual attribute=value or XML. Can be used instead of `path`.

update_type There are two ways to update an image: `replace` the whole template or `merge` the new template with the existing one.

CLI Example:

```
salt-cloud -f image_update opennebula image_id=0 file=/path/to/image_update_file.
↪txt update_type=replace
salt-cloud -f image_update opennebula image_name="Ubuntu 14.04" update_type=merge
↪\
    data='NAME="Ubuntu Dev" PATH="/home/one_user/images/ubuntu_desktop.img" \
    DESCRIPTION = "Ubuntu 14.04 for development."'
```

salt.cloud.clouds.opennebula.**list_clusters**(*call=None*)

Returns a list of clusters in OpenNebula.

New in version 2016.3.0.

CLI Example:

```
salt-cloud -f list_clusters opennebula
```

salt.cloud.clouds.opennebula.**list_datastores**(*call=None*)

Returns a list of data stores on OpenNebula.

New in version 2016.3.0.

CLI Example:

```
salt-cloud -f list_datastores opennebula
```

salt.cloud.clouds.opennebula.**list_hosts**(*call=None*)

Returns a list of hosts on OpenNebula.

New in version 2016.3.0.

CLI Example:

```
salt-cloud -f list_hosts opennebula
```

salt.cloud.clouds.opennebula.**list_nodes**(*call=None*)

Return a list of VMs on OpenNebula.

CLI Example:

```
salt-cloud -Q
salt-cloud --query
salt-cloud --function list_nodes opennebula
salt-cloud -f list_nodes opennebula
```

salt.cloud.clouds.opennebula.**list_nodes_full**(*call=None*)

Return a list of the VMs on OpenNebula.

CLI Example:

```
salt-cloud -F
salt-cloud --full-query
salt-cloud --function list_nodes_full opennebula
salt-cloud -f list_nodes_full opennebula
```

salt.cloud.clouds.opennebula.**list_nodes_select**(*call=None*)

Return a list of the VMs that are on the provider, with select fields.

salt.cloud.clouds.opennebula.**list_security_groups**(*call=None*)

Lists all security groups available to the user and the user's groups.

New in version 2016.3.0.

CLI Example:

```
salt-cloud -f list_security_groups opennebula
```

`salt.cloud.clouds.opennebula.`**`list_templates`**(*call=None*)

Lists all templates available to the user and the user's groups.

New in version 2016.3.0.

CLI Example:

```
salt-cloud -f list_templates opennebula
```

`salt.cloud.clouds.opennebula.`**`list_vns`**(*call=None*)

Lists all virtual networks available to the user and the user's groups.

New in version 2016.3.0.

CLI Example:

```
salt-cloud -f list_vns opennebula
```

`salt.cloud.clouds.opennebula.`**`reboot`**(*name*, *call=None*)

Reboot a VM.

New in version 2016.3.0.

name The name of the VM to reboot.

CLI Example:

```
salt-cloud -a reboot my-vm
```

`salt.cloud.clouds.opennebula.`**`secgroup_allocate`**(*call=None*, *kwargs=None*)

Allocates a new security group in OpenNebula.

New in version 2016.3.0.

path The path to a file containing the template of the security group. Syntax within the file can be the usual attribute=value or XML. Can be used instead of `data`.

data The template data of the security group. Syntax can be the usual attribute=value or XML. Can be used instead of `path`.

CLI Example:

```
salt-cloud -f secgroup_allocate opennebula path=/path/to/secgroup_file.txt
salt-cloud -f secgroup_allocate opennebula \
    data="NAME = test RULE = [PROTOCOL = TCP, RULE_TYPE = inbound, \
    RANGE = 1000:2000]"
```

`salt.cloud.clouds.opennebula.`**`secgroup_clone`**(*call=None*, *kwargs=None*)

Clones an existing security group.

New in version 2016.3.0.

name The name of the new template.

secgroup_id The ID of the security group to be cloned. Can be used instead of `secgroup_name`.

secgroup_name The name of the security group to be cloned. Can be used instead of `secgroup_id`.

CLI Example:

```
salt-cloud -f secgroup_clone opennebula name=my-cloned-secgroup secgroup_id=0
salt-cloud -f secgroup_clone opennebula name=my-cloned-secgroup secgroup_name=my-
↪secgroup
```

`salt.cloud.clouds.opennebula.`**`secgroup_delete`**`(`*call=None, kwargs=None*`)`

Deletes the given security group from OpenNebula. Either a name or a secgroup_id must be supplied.

New in version 2016.3.0.

name The name of the security group to delete. Can be used instead of `secgroup_id`.

secgroup_id The ID of the security group to delete. Can be used instead of `name`.

CLI Example:

```
salt-cloud -f secgroup_delete opennebula name=my-secgroup
salt-cloud --function secgroup_delete opennebula secgroup_id=100
```

`salt.cloud.clouds.opennebula.`**`secgroup_info`**`(`*call=None, kwargs=None*`)`

Retrieves information for the given security group. Either a name or a secgroup_id must be supplied.

New in version 2016.3.0.

name The name of the security group for which to gather information. Can be used instead of `secgroup_id`.

secgroup_id The ID of the security group for which to gather information. Can be used instead of `name`.

CLI Example:

```
salt-cloud -f secgroup_info opennebula name=my-secgroup
salt-cloud --function secgroup_info opennebula secgroup_id=5
```

`salt.cloud.clouds.opennebula.`**`secgroup_update`**`(`*call=None, kwargs=None*`)`

Replaces the security group template contents.

New in version 2016.3.0.

secgroup_id The ID of the security group to update. Can be used instead of `secgroup_name`.

secgroup_name The name of the security group to update. Can be used instead of `secgroup_id`.

path The path to a file containing the template of the security group. Syntax within the file can be the usual attribute=value or XML. Can be used instead of `data`.

data The template data of the security group. Syntax can be the usual attribute=value or XML. Can be used instead of `path`.

update_type There are two ways to update a security group: `replace` the whole template or `merge` the new template with the existing one.

CLI Example:

```
salt-cloud --function secgroup_update opennebula secgroup_id=100 \
    path=/path/to/secgroup_update_file.txt \
    update_type=replace
salt-cloud -f secgroup_update opennebula secgroup_name=my-secgroup update_
↪type=merge \
    data="Name = test RULE = [PROTOCOL = TCP, RULE_TYPE = inbound, RANGE =⊠
↪1000:2000]"
```

`salt.cloud.clouds.opennebula.`**`show_instance`**`(`*name, call=None*`)`

Show the details from OpenNebula concerning a named VM.

name The name of the VM for which to display details.

call Type of call to use with this function such as `function`.

CLI Example:

```
salt-cloud --action show_instance vm_name
salt-cloud -a show_instance vm_name
```

`salt.cloud.clouds.opennebula.`**`start`**(*name*, *call=None*)

Start a VM.

New in version 2016.3.0.

name The name of the VM to start.

CLI Example:

```
salt-cloud -a start my-vm
```

`salt.cloud.clouds.opennebula.`**`stop`**(*name*, *call=None*)

Stop a VM.

New in version 2016.3.0.

name The name of the VM to stop.

CLI Example:

```
salt-cloud -a stop my-vm
```

`salt.cloud.clouds.opennebula.`**`template_allocate`**(*call=None*, *kwargs=None*)

Allocates a new template in OpenNebula.

New in version 2016.3.0.

path The path to a file containing the elements of the template to be allocated. Syntax within the file can be the usual attribute=value or XML. Can be used instead of `data`.

data Contains the elements of the template to be allocated. Syntax can be the usual attribute=value or XML. Can be used instead of `path`.

CLI Example:

```
salt-cloud -f template_allocate opennebula path=/path/to/template_file.txt
salt-cloud -f template_allocate opennebula \
    data='CPU="1.0" DISK=[IMAGE="Ubuntu-14.04"] GRAPHICS=[LISTEN="0.0.0.0",TYPE=
↪"vnc"] \
    MEMORY="1024" NETWORK="yes" NIC=[NETWORK="192net",NETWORK_UNAME="oneadmin"] \
    OS=[ARCH="x86_64"] SUNSTONE_CAPACITY_SELECT="YES" SUNSTONE_NETWORK_SELECT="YES
↪" \
    VCPU="1"'
```

`salt.cloud.clouds.opennebula.`**`template_clone`**(*call=None*, *kwargs=None*)

Clones an existing virtual machine template.

New in version 2016.3.0.

name The name of the new template.

template_id The ID of the template to be cloned. Can be used instead of `template_name`.

template_name The name of the template to be cloned. Can be used instead of `template_id`.

clone_images Optional, defaults to False. Indicates if the images attached to the template should be cloned as well.

CLI Example:

```
salt-cloud -f template_clone opennebula name=my-new-template template_id=0
salt-cloud -f template_clone opennebula name=my-new-template template_name=my-
↪template
```

`salt.cloud.clouds.opennebula.`**`template_delete`**(*call=None*, *kwargs=None*)

Deletes the given template from OpenNebula. Either a name or a template_id must be supplied.

New in version 2016.3.0.

name The name of the template to delete. Can be used instead of `template_id`.

template_id The ID of the template to delete. Can be used instead of `name`.

CLI Example:

```
salt-cloud -f template_delete opennebula name=my-template
salt-cloud --function template_delete opennebula template_id=5
```

`salt.cloud.clouds.opennebula.`**`template_instantiate`**(*call=None*, *kwargs=None*)

Instantiates a new virtual machine from a template.

New in version 2016.3.0.

Note: `template_instantiate` creates a VM on OpenNebula from a template, but it does not install Salt on the new VM. Use the `create` function for that functionality: `salt-cloud -p opennebula-profile vm-name`.

vm_name Name for the new VM instance.

template_id The ID of the template from which the VM will be created. Can be used instead of `template_name`.

template_name The name of the template from which the VM will be created. Can be used instead of `template_id`.

CLI Example:

```
salt-cloud -f template_instantiate opennebula vm_name=my-new-vm template_id=0
```

`salt.cloud.clouds.opennebula.`**`template_update`**(*call=None*, *kwargs=None*)

Replaces the template contents.

New in version 2016.3.0.

template_id The ID of the template to update. Can be used instead of `template_name`.

template_name The name of the template to update. Can be used instead of `template_id`.

path The path to a file containing the elements of the template to be updated. Syntax within the file can be the usual attribute=value or XML. Can be used instead of `data`.

data Contains the elements of the template to be updated. Syntax can be the usual attribute=value or XML. Can be used instead of `path`.

update_type There are two ways to update a template: `replace` the whole template or `merge` the new template with the existing one.

CLI Example:

```
salt-cloud --function template_update opennebula template_id=1 update_
↪type=replace \
    path=/path/to/template_update_file.txt
salt-cloud -f template_update opennebula template_name=my-template update_
↪type=merge \
    data='CPU="1.0" DISK=[IMAGE="Ubuntu-14.04"] GRAPHICS=[LISTEN="0.0.0.0",TYPE=
↪"vnc"] \
    MEMORY="1024" NETWORK="yes" NIC=[NETWORK="192net",NETWORK_UNAME="oneadmin"] \
    OS=[ARCH="x86_64"] SUNSTONE_CAPACITY_SELECT="YES" SUNSTONE_NETWORK_SELECT="YES
↪" \
    VCPU="1"'
```

`salt.cloud.clouds.opennebula.`**`vm_action`**(*name*, *kwargs=None*, *call=None*)

Submits an action to be performed on a given virtual machine.

New in version 2016.3.0.

name The name of the VM to action.

action

The action to be performed on the VM. Available options include:

- boot
- delete
- delete-recreate
- hold
- poweroff
- poweroff-hard
- reboot
- reboot-hard
- release
- resched
- resume
- shutdown
- shutdown-hard
- stop
- suspend
- undeploy
- undeploy-hard
- unresched

CLI Example:

```
salt-cloud -a vm_action my-vm action='release'
```

`salt.cloud.clouds.opennebula.`**`vm_allocate`**(*call=None*, *kwargs=None*)

Allocates a new virtual machine in OpenNebula.

New in version 2016.3.0.

path The path to a file defining the template of the VM to allocate. Syntax within the file can be the usual attribute=value or XML. Can be used instead of `data`.

data Contains the template definitions of the VM to allocate. Syntax can be the usual attribute=value or XML. Can be used instead of `path`.

hold If this parameter is set to `True`, the VM will be created in the `HOLD` state. If not set, the VM is created in the `PENDING` state. Default is `False`.

CLI Example:

```
salt-cloud -f vm_allocate path=/path/to/vm_template.txt
salt-cloud --function vm_allocate path=/path/to/vm_template.txt hold=True
```

`salt.cloud.clouds.opennebula.`**`vm_attach`**(*name*, *kwargs=None*, *call=None*)

Attaches a new disk to the given virtual machine.

New in version 2016.3.0.

name The name of the VM for which to attach the new disk.

path The path to a file containing a single disk vector attribute. Syntax within the file can be the usual attribute=value or XML. Can be used instead of `data`.

data Contains the data needed to attach a single disk vector attribute. Syntax can be the usual attribute=value or XML. Can be used instead of `path`.

CLI Example:

```
salt-cloud -a vm_attach my-vm path=/path/to/disk_file.txt
salt-cloud -a vm_attach my-vm data="DISK=[DISK_ID=1]"
```

`salt.cloud.clouds.opennebula.`**`vm_attach_nic`**(*name*, *kwargs=None*, *call=None*)

Attaches a new network interface to the given virtual machine.

New in version 2016.3.0.

name The name of the VM for which to attach the new network interface.

path The path to a file containing a single NIC vector attribute. Syntax within the file can be the usual attribute=value or XML. Can be used instead of `data`.

data Contains the single NIC vector attribute to attach to the VM. Syntax can be the usual attribute=value or XML. Can be used instead of `path`.

CLI Example:

```
salt-cloud -a vm_attach_nic my-vm path=/path/to/nic_file.txt
salt-cloud -a vm_attach_nic my-vm data="NIC=[NETWORK_ID=1]"
```

`salt.cloud.clouds.opennebula.`**`vm_deploy`**(*name*, *kwargs=None*, *call=None*)

Initiates the instance of the given VM on the target host.

New in version 2016.3.0.

name The name of the VM to deploy.

host_id The ID of the target host where the VM will be deployed. Can be used instead of `host_name`.

host_name The name of the target host where the VM will be deployed. Can be used instead of `host_id`.

capacity_maintained True to enforce the Host capacity is not over-committed. This parameter is only acknowledged for users in the `oneadmin` group. Host capacity will be always enforced for regular users.

datastore_id The ID of the target system data-store where the VM will be deployed. Optional and can be used instead of `datastore_name`. If neither `datastore_id` nor `datastore_name` are set, OpenNebula will choose the data-store.

datastore_name The name of the target system data-store where the VM will be deployed. Optional, and can be used instead of `datastore_id`. If neither `datastore_id` nor `datastore_name` are set, OpenNebula will choose the data-store.

CLI Example:

```
salt-cloud -a vm_deploy my-vm host_id=0
salt-cloud -a vm_deploy my-vm host_id=1 capacity_maintained=False
salt-cloud -a vm_deploy my-vm host_name=host01 datastore_id=1
salt-cloud -a vm_deploy my-vm host_name=host01 datastore_name=default
```

`salt.cloud.clouds.opennebula.`**`vm_detach`**(*name*, *kwargs=None*, *call=None*)

Detaches a disk from a virtual machine.

New in version 2016.3.0.

name The name of the VM from which to detach the disk.

disk_id The ID of the disk to detach.

CLI Example:

```
salt-cloud -a vm_detach my-vm disk_id=1
```

`salt.cloud.clouds.opennebula.`**`vm_detach_nic`**(*name*, *kwargs=None*, *call=None*)

Detaches a disk from a virtual machine.

New in version 2016.3.0.

name The name of the VM from which to detach the network interface.

nic_id The ID of the nic to detach.

CLI Example:

```
salt-cloud -a vm_detach_nic my-vm nic_id=1
```

`salt.cloud.clouds.opennebula.`**`vm_disk_save`**(*name*, *kwargs=None*, *call=None*)

Sets the disk to be saved in the given image.

New in version 2016.3.0.

name The name of the VM containing the disk to save.

disk_id The ID of the disk to save.

image_name The name of the new image where the disk will be saved.

image_type The type for the new image. If not set, then the default `ONED` Configuration will be used. Other valid types include: OS, CDROM, DATABLOCK, KERNEL, RAMDISK, and CONTEXT.

snapshot_id The ID of the snapshot to export. If not set, the current image state will be used.

CLI Example:

```
salt-cloud -a vm_disk_save my-vm disk_id=1 image_name=my-new-image
salt-cloud -a vm_disk_save my-vm disk_id=1 image_name=my-new-image image_
↪type=CONTEXT snapshot_id=10
```

(continues on next page)

(continued from previous page)

`salt.cloud.clouds.opennebula.`**`vm_disk_snapshot_create`**(*name, kwargs=None, call=None*)

Takes a new snapshot of the disk image.

New in version 2016.3.0.

name The name of the VM of which to take the snapshot.

disk_id The ID of the disk to save.

description The description for the snapshot.

CLI Example:

```
salt-cloud -a vm_disk_snapshot_create my-vm disk_id=0 description="My Snapshot Description"
```

`salt.cloud.clouds.opennebula.`**`vm_disk_snapshot_delete`**(*name, kwargs=None, call=None*)

Deletes a disk snapshot based on the given VM and the disk_id.

New in version 2016.3.0.

name The name of the VM containing the snapshot to delete.

disk_id The ID of the disk to save.

snapshot_id The ID of the snapshot to be deleted.

CLI Example:

```
salt-cloud -a vm_disk_snapshot_delete my-vm disk_id=0 snapshot_id=6
```

`salt.cloud.clouds.opennebula.`**`vm_disk_snapshot_revert`**(*name, kwargs=None, call=None*)

Reverts a disk state to a previously taken snapshot.

New in version 2016.3.0.

name The name of the VM containing the snapshot.

disk_id The ID of the disk to revert its state.

snapshot_id The ID of the snapshot to which the snapshot should be reverted.

CLI Example:

```
salt-cloud -a vm_disk_snapshot_revert my-vm disk_id=0 snapshot_id=6
```

`salt.cloud.clouds.opennebula.`**`vm_info`**(*name, call=None*)

Retrieves information for a given virtual machine. A VM name must be supplied.

New in version 2016.3.0.

name The name of the VM for which to gather information.

CLI Example:

```
salt-cloud -a vm_info my-vm
```

`salt.cloud.clouds.opennebula.`**`vm_migrate`**(*name*, *kwargs=None*, *call=None*)

Migrates the specified virtual machine to the specified target host.

New in version 2016.3.0.

name The name of the VM to migrate.

host_id The ID of the host to which the VM will be migrated. Can be used instead of `host_name`.

host_name The name of the host to which the VM will be migrated. Can be used instead of `host_id`.

live_migration If set to `True`, a live-migration will be performed. Default is `False`.

capacity_maintained True to enforce the Host capacity is not over-committed. This parameter is only acknowledged for users in the `oneadmin` group. Host capacity will be always enforced for regular users.

datastore_id The target system data-store ID where the VM will be migrated. Can be used instead of `datastore_name`.

datastore_name The name of the data-store target system where the VM will be migrated. Can be used instead of `datastore_id`.

CLI Example:

```
salt-cloud -a vm_migrate my-vm host_id=0 datastore_id=1
salt-cloud -a vm_migrate my-vm host_id=0 datastore_id=1 live_migration=True
salt-cloud -a vm_migrate my-vm host_name=host01 datastore_name=default
```

`salt.cloud.clouds.opennebula.`**`vm_monitoring`**(*name*, *call=None*)

Returns the monitoring records for a given virtual machine. A VM name must be supplied.

The monitoring information returned is a list of VM elements. Each VM element contains the complete dictionary of the VM with the updated information returned by the poll action.

New in version 2016.3.0.

name The name of the VM for which to gather monitoring records.

CLI Example:

```
salt-cloud -a vm_monitoring my-vm
```

`salt.cloud.clouds.opennebula.`**`vm_resize`**(*name*, *kwargs=None*, *call=None*)

Changes the capacity of the virtual machine.

New in version 2016.3.0.

name The name of the VM to resize.

path The path to a file containing new capacity elements CPU, VCPU, MEMORY. If one of them is not present, or its value is 0, the VM will not be re-sized. Syntax within the file can be the usual attribute=value or XML. Can be used instead of `data`.

data Contains the new capacity elements CPU, VCPU, and MEMORY. If one of them is not present, or its value is 0, the VM will not be re-sized. Can be used instead of `path`.

capacity_maintained True to enforce the Host capacity is not over-committed. This parameter is only acknowledged for users in the `oneadmin` group. Host capacity will be always enforced for regular users.

CLI Example:

```
salt-cloud -a vm_resize my-vm path=/path/to/capacity_template.txt
salt-cloud -a vm_resize my-vm path=/path/to/capacity_template.txt capacity_
↪maintained=False
salt-cloud -a vm_resize my-vm data="CPU=1 VCPU=1 MEMORY=1024"
```

`salt.cloud.clouds.opennebula.`**`vm_snapshot_create`**(*vm_name*, *kwargs=None*, *call=None*)

Creates a new virtual machine snapshot from the provided VM.

New in version 2016.3.0.

vm_name The name of the VM from which to create the snapshot.

snapshot_name The name of the snapshot to be created.

CLI Example:

```
salt-cloud -a vm_snapshot_create my-vm snapshot_name=my-new-snapshot
```

`salt.cloud.clouds.opennebula.`**`vm_snapshot_delete`**(*vm_name*, *kwargs=None*, *call=None*)

Deletes a virtual machine snapshot from the provided VM.

New in version 2016.3.0.

vm_name The name of the VM from which to delete the snapshot.

snapshot_id The ID of the snapshot to be deleted.

CLI Example:

```
salt-cloud -a vm_snapshot_delete my-vm snapshot_id=8
```

`salt.cloud.clouds.opennebula.`**`vm_snapshot_revert`**(*vm_name*, *kwargs=None*, *call=None*)

Reverts a virtual machine to a snapshot

New in version 2016.3.0.

vm_name The name of the VM to revert.

snapshot_id The snapshot ID.

CLI Example:

```
salt-cloud -a vm_snapshot_revert my-vm snapshot_id=42
```

`salt.cloud.clouds.opennebula.`**`vm_update`**(*name*, *kwargs=None*, *call=None*)

Replaces the user template contents.

New in version 2016.3.0.

name The name of the VM to update.

path The path to a file containing new user template contents. Syntax within the file can be the usual attribute=value or XML. Can be used instead of `data`.

data Contains the new user template contents. Syntax can be the usual attribute=value or XML. Can be used instead of `path`.

update_type There are two ways to update a VM: `replace` the whole template or `merge` the new template with the existing one.

CLI Example:

```
salt-cloud -a vm_update my-vm path=/path/to/user_template_file.txt update_type=
↪'replace'
```

`salt.cloud.clouds.opennebula.`**`vn_add_ar`**`(`*call=None*, *kwargs=None*`)`

Adds address ranges to a given virtual network.

New in version 2016.3.0.

vn_id The ID of the virtual network to add the address range. Can be used instead of `vn_name`.

vn_name The name of the virtual network to add the address range. Can be used instead of `vn_id`.

path The path to a file containing the template of the address range to add. Syntax within the file can be the usual attribute=value or XML. Can be used instead of `data`.

data Contains the template of the address range to add. Syntax can be the usual attribute=value or XML. Can be used instead of `path`.

CLI Example:

```
salt-cloud -f vn_add_ar opennebula vn_id=3 path=/path/to/address_range.txt
salt-cloud -f vn_add_ar opennebula vn_name=my-vn \
    data="AR=[TYPE=IP4, IP=192.168.0.5, SIZE=10]"
```

`salt.cloud.clouds.opennebula.`**`vn_allocate`**`(`*call=None*, *kwargs=None*`)`

Allocates a new virtual network in OpenNebula.

New in version 2016.3.0.

path The path to a file containing the template of the virtual network to allocate. Syntax within the file can be the usual attribute=value or XML. Can be used instead of `data`.

data Contains the template of the virtual network to allocate. Syntax can be the usual attribute=value or XML. Can be used instead of `path`.

cluster_id The ID of the cluster for which to add the new virtual network. Can be used instead of `cluster_name`. If neither `cluster_id` nor `cluster_name` are provided, the virtual network won't be added to any cluster.

cluster_name The name of the cluster for which to add the new virtual network. Can be used instead of `cluster_id`. If neither `cluster_name` nor `cluster_id` are provided, the virtual network won't be added to any cluster.

CLI Example:

```
salt-cloud -f vn_allocate opennebula path=/path/to/vn_file.txt
```

`salt.cloud.clouds.opennebula.`**`vn_delete`**`(`*call=None*, *kwargs=None*`)`

Deletes the given virtual network from OpenNebula. Either a name or a vn_id must be supplied.

New in version 2016.3.0.

name The name of the virtual network to delete. Can be used instead of `vn_id`.

vn_id The ID of the virtual network to delete. Can be used instead of `name`.

CLI Example:

```
salt-cloud -f vn_delete opennebula name=my-virtual-network
salt-cloud --function vn_delete opennebula vn_id=3
```

`salt.cloud.clouds.opennebula.`**`vn_free_ar`**`(`*call=None*, *kwargs=None*`)`

Frees a reserved address range from a virtual network.

New in version 2016.3.0.

vn_id The ID of the virtual network from which to free an address range. Can be used instead of `vn_name`.

vn_name The name of the virtual network from which to free an address range. Can be used instead of `vn_id`.

ar_id The ID of the address range to free.

CLI Example:

```
salt-cloud -f vn_free_ar opennebula vn_id=3 ar_id=1
salt-cloud -f vn_free_ar opennebula vn_name=my-vn ar_id=1
```

`salt.cloud.clouds.opennebula.`**`vn_hold`**(*call=None*, *kwargs=None*)

Holds a virtual network lease as used.

New in version 2016.3.0.

vn_id The ID of the virtual network from which to hold the lease. Can be used instead of `vn_name`.

vn_name The name of the virtual network from which to hold the lease. Can be used instead of `vn_id`.

path The path to a file defining the template of the lease to hold. Syntax within the file can be the usual attribute=value or XML. Can be used instead of `data`.

data Contains the template of the lease to hold. Syntax can be the usual attribute=value or XML. Can be used instead of `path`.

CLI Example:

```
salt-cloud -f vn_hold opennebula vn_id=3 path=/path/to/vn_hold_file.txt
salt-cloud -f vn_hold opennebula vn_name=my-vn data="LEASES=[IP=192.168.0.5]"
```

`salt.cloud.clouds.opennebula.`**`vn_info`**(*call=None*, *kwargs=None*)

Retrieves information for the virtual network.

New in version 2016.3.0.

name The name of the virtual network for which to gather information. Can be used instead of `vn_id`.

vn_id The ID of the virtual network for which to gather information. Can be used instead of `name`.

CLI Example:

```
salt-cloud -f vn_info opennebula vn_id=3
salt-cloud --function vn_info opennebula name=public
```

`salt.cloud.clouds.opennebula.`**`vn_release`**(*call=None*, *kwargs=None*)

Releases a virtual network lease that was previously on hold.

New in version 2016.3.0.

vn_id The ID of the virtual network from which to release the lease. Can be used instead of `vn_name`.

vn_name The name of the virtual network from which to release the lease. Can be used instead of `vn_id`.

path The path to a file defining the template of the lease to release. Syntax within the file can be the usual attribute=value or XML. Can be used instead of `data`.

data Contains the template defining the lease to release. Syntax can be the usual attribute=value or XML. Can be used instead of `path`.

CLI Example:

```
salt-cloud -f vn_release opennebula vn_id=3 path=/path/to/vn_release_file.txt
salt-cloud =f vn_release opennebula vn_name=my-vn data="LEASES=[IP=192.168.0.5]"
```

`salt.cloud.clouds.opennebula.`**`vn_reserve`**(*call=None*, *kwargs=None*)

Reserve network addresses.

New in version 2016.3.0.

vn_id The ID of the virtual network from which to reserve addresses. Can be used instead of vn_name.

vn_name The name of the virtual network from which to reserve addresses. Can be used instead of vn_id.

path The path to a file defining the template of the address reservation. Syntax within the file can be the usual attribute=value or XML. Can be used instead of `data`.

data Contains the template defining the address reservation. Syntax can be the usual attribute=value or XML. Data provided must be wrapped in double quotes. Can be used instead of `path`.

CLI Example:

```
salt-cloud -f vn_reserve opennebula vn_id=3 path=/path/to/vn_reserve_file.txt
salt-cloud -f vn_reserve opennebula vn_name=my-vn data="SIZE=10 AR_ID=8 NETWORK_
↪ID=1"
```

21.4.15 salt.cloud.clouds.openstack

Openstack Cloud Driver

depends shade>=1.19.0

OpenStack is an open source project that is in use by a number a cloud providers, each of which have their own ways of using it.

This OpenStack driver uses a the shade python module which is managed by the OpenStack Infra team. This module is written to handle all the different versions of different OpenStack tools for salt, so most commands are just passed over to the module to handle everything.

Provider

There are two ways to configure providers for this driver. The first one is to just let shade handle everything, and configure using os-client-config and setting up */etc/openstack/clouds.yml*.

```
clouds:
  democloud:
    region_name: RegionOne
    auth:
      username: 'demo'
      password: secret
      project_name: 'demo'
      auth_url: 'http://openstack/identity'
```

And then this can be referenced in the salt provider based on the *democloud* name.

```
myopenstack:
  driver: openstack
  cloud: democloud
  region_name: RegionOne
```

This allows for just using one configuration for salt-cloud and for any other openstack tools which are all using */etc/openstack/clouds.yml*

The other method allows for specifying everything in the provider config, instead of using the extra configuration file. This will allow for passing salt-cloud configs only through pillars for minions without having to write a clouds.yml file on each minion.abs

```
myopenstack:
  driver: openstack
  region_name: RegionOne
  auth:
    username: 'demo'
    password: secret
    project_name: 'demo'
    user_domain_name: default,
    project_domain_name: default,
    auth_url: 'http://openstack/identity'
```

Or if you need to use a profile to setup some extra stuff, it can be passed as a *profile* to use any of the vendor config options.

```
myrackspace:
  driver: openstack
  profile: rackspace
  auth:
    username: rackusername
    api_key: myapikey
  region_name: ORD
  auth_type: rackspace_apikey
```

And this will pull in the profile for rackspace and setup all the correct options for the auth_url and different api versions for services.

Profile

Most of the options for building servers are just passed on to the create_server function from shade.

The salt specific ones are:

- ssh_key_file: The path to the ssh key that should be used to login to the machine to bootstrap it
- ssh_key_file: The name of the keypair in openstack
- userdata_template: The renderer to use if the userdata is a file that is templated. Default: False
- ssh_interface: The interface to use to login for bootstrapping: public_ips, private_ips, floating_ips, fixed_ips

```
centos:
  provider: myopenstack
  image: CentOS 7
  size: ds1G
  ssh_key_name: mykey
  ssh_key_file: /root/.ssh/id_rsa
```

This is the minimum setup required.

If metadata is set to make sure that the host has finished setting up the *wait_for_metadata* can be set.

```
centos:
  provider: myopenstack
  image: CentOS 7
```

(continues on next page)

(continued from previous page)

```
size: ds1G
ssh_key_name: mykey
ssh_key_file: /root/.ssh/id_rsa
meta:
  build_config: rack_user_only
wait_for_metadata:
  rax_service_level_automation: Complete
  rackconnect_automation_status: DEPLOYED
```

Anything else from the create_server docs can be passed through here.

- **image**: **Image dict, name or ID to boot with. image is required** unless boot_volume is given.
- **flavor**: Flavor dict, name or ID to boot onto.
- **auto_ip**: **Whether to take actions to find a routable IP for** the server. (defaults to True)
- **ips**: List of IPs to attach to the server (defaults to None)
- **ip_pool**: **Name of the network or floating IP pool to get an** address from. (defaults to None)
- **root_volume**: **Name or ID of a volume to boot from** (defaults to None - deprecated, use boot_volume)
- **boot_volume**: **Name or ID of a volume to boot from** (defaults to None)
- **terminate_volume**: **If booting from a volume, whether it should** be deleted when the server is destroyed. (defaults to False)
- **volumes**: (optional) A list of volumes to attach to the server
- **meta**: **(optional) A dict of arbitrary key/value metadata to** store for this server. Both keys and values must be <=255 characters.
- **files**: **(optional, deprecated) A dict of files to overwrite** on the server upon boot. Keys are file names (i.e. `/etc/passwd`) and values are the file contents (either as a string or as a file-like object). A maximum of five entries is allowed, and each file must be 10k or less.
- **reservation_id**: a UUID for the set of servers being requested.
- **min_count**: **(optional extension) The minimum number of** servers to launch.
- **max_count**: **(optional extension) The maximum number of** servers to launch.
- **security_groups**: A list of security group names
- **userdata**: **user data to pass to be exposed by the metadata** server this can be a file type object as well or a string.
- **key_name**: **(optional extension) name of previously created** keypair to inject into the instance.
- **availability_zone**: **Name of the availability zone for instance** placement.
- **block_device_mapping**: **(optional) A dict of block** device mappings for this server.
- **block_device_mapping_v2**: **(optional) A dict of block** device mappings for this server.
- **nics**: **(optional extension) an ordered list of nics to be** added to this server, with information about connected networks, fixed IPs, port etc.
- **scheduler_hints**: **(optional extension) arbitrary key-value pairs** specified by the client to help boot an instance
- **config_drive**: **(optional extension) value for config drive** either boolean, or volume-id

- **disk_config: (optional extension) control how the disk is** partitioned when the server is created. possible values are 'AUTO' or 'MANUAL'.
- **admin_pass: (optional extension) add a user supplied admin** password.
- **timeout: (optional) Seconds to wait, defaults to 60**. See the `wait` parameter.
- **reuse_ips: (optional) Whether to attempt to reuse pre-existing** floating ips should a floating IP be needed (defaults to True)
- **network: (optional) Network dict or name or ID to attach the** server to. Mutually exclusive with the nics parameter. Can also be be a list of network names or IDs or network dicts.
- **boot_from_volume: Whether to boot from volume. 'boot_volume'** implies True, but boot_from_volume=True with no boot_volume is valid and will create a volume from the image and use that.
- **volume_size: When booting an image from volume, how big should** the created volume be? Defaults to 50.
- **nat_destination: Which network should a created floating IP** be attached to, if it's not possible to infer from the cloud's configuration. (Optional, defaults to None)
- **group: ServerGroup dict, name or id to boot the server in**. If a group is provided in both scheduler_hints and in the group param, the group param will win. (Optional, defaults to None)

Note: If there is anything added, that is not in this list, it can be added to an *extras* dictionary for the profile, and that will be to the create_server function.

`salt.cloud.clouds.openstack.`**`avail_images`**`(`*conn=None, call=None*`)`

List available images for OpenStack

CLI Example

```
salt-cloud -f avail_images myopenstack
salt-cloud --list-images myopenstack
```

`salt.cloud.clouds.openstack.`**`avail_sizes`**`(`*conn=None, call=None*`)`

List available sizes for OpenStack

CLI Example

```
salt-cloud -f avail_sizes myopenstack
salt-cloud --list-sizes myopenstack
```

`salt.cloud.clouds.openstack.`**`call`**`(`*conn=None, call=None, kwargs=None*`)`

Call function from shade.

func

function to call from shade.openstackcloud library

CLI Example

```
salt-cloud -f call myopenstack func=list_images
t sujksalt-cloud -f call myopenstack func=create_network name=mysubnet
```

`salt.cloud.clouds.openstack.`**`create`**`(`*vm_*`)`

Create a single VM from a data dict

`salt.cloud.clouds.openstack.`**`destroy`**`(`*name, conn=None, call=None*`)`

Delete a single VM

`salt.cloud.clouds.openstack.`**`get_configured_provider`**`()`
Return the first configured instance.

`salt.cloud.clouds.openstack.`**`get_conn`**`()`
Return a conn object for the passed VM data

`salt.cloud.clouds.openstack.`**`get_dependencies`**`()`
Warn if dependencies aren't met.

`salt.cloud.clouds.openstack.`**`list_networks`**`(`*conn=None*, *call=None*`)`
List networks for OpenStack

CLI Example

```
salt-cloud -f list_networks myopenstack
```

`salt.cloud.clouds.openstack.`**`list_nodes`**`(`*conn=None*, *call=None*`)`
Return a list of VMs

CLI Example

```
salt-cloud -f list_nodes myopenstack
```

`salt.cloud.clouds.openstack.`**`list_nodes_full`**`(`*conn=None*, *call=None*`)`
Return a list of VMs with all the information about them

CLI Example

```
salt-cloud -f list_nodes_full myopenstack
```

`salt.cloud.clouds.openstack.`**`list_nodes_min`**`(`*conn=None*, *call=None*`)`
Return a list of VMs with minimal information

CLI Example

```
salt-cloud -f list_nodes_min myopenstack
```

`salt.cloud.clouds.openstack.`**`list_nodes_select`**`(`*conn=None*, *call=None*`)`
Return a list of VMs with the fields from *query.selection*

CLI Example

```
salt-cloud -f list_nodes_full myopenstack
```

`salt.cloud.clouds.openstack.`**`list_subnets`**`(`*conn=None*, *call=None*, *kwargs=None*`)`
List subnets in a virtual network

network network to list subnets of

```
salt-cloud -f list_subnets myopenstack network=salt-net
```

`salt.cloud.clouds.openstack.`**`preferred_ip`**`(`*vm_*, *ips*`)`
Return the preferred Internet protocol. Either 'ipv4' (default) or 'ipv6'.

`salt.cloud.clouds.openstack.`**`request_instance`**`(`*vm_*, *conn=None*, *call=None*`)`
Request an instance to be built

`salt.cloud.clouds.openstack.`**`show_instance`**`(`*name*, *conn=None*, *call=None*`)`
Get VM on this OpenStack account

name

name of the instance

CLI Example

```
salt-cloud -a show_instance myserver
```

`salt.cloud.clouds.openstack.`**`ssh_interface`**(*vm_*)
Return the ssh_interface type to connect to. Either 'public_ips' (default) or 'private_ips'.

21.4.16 salt.cloud.clouds.parallels

Parallels Cloud Module

The Parallels cloud module is used to control access to cloud providers using the Parallels VPS system.

Set up the cloud configuration at `/etc/salt/cloud.providers` or `/etc/salt/cloud.providers.d/parallels.conf`:

```
my-parallels-config:
  # Parallels account information
  user: myuser
  password: mypassword
  url: https://api.cloud.xmission.com:4465/paci/v1.0/
  driver: parallels
```

`salt.cloud.clouds.parallels.`**`avail_images`**(*call=None*)
Return a list of the images that are on the provider

`salt.cloud.clouds.parallels.`**`create`**(*vm_*)
Create a single VM from a data dict

`salt.cloud.clouds.parallels.`**`create_node`**(*vm_*)
Build and submit the XML to create a node

`salt.cloud.clouds.parallels.`**`destroy`**(*name, call=None*)
Destroy a node.

CLI Example:

```
salt-cloud --destroy mymachine
```

`salt.cloud.clouds.parallels.`**`get_configured_provider`**()
Return the first configured instance.

`salt.cloud.clouds.parallels.`**`get_image`**(*vm_*)
Return the image object to use

`salt.cloud.clouds.parallels.`**`list_nodes`**(*call=None*)
Return a list of the VMs that are on the provider

`salt.cloud.clouds.parallels.`**`list_nodes_full`**(*call=None*)
Return a list of the VMs that are on the provider

`salt.cloud.clouds.parallels.`**`list_nodes_select`**(*call=None*)
Return a list of the VMs that are on the provider, with select fields

`salt.cloud.clouds.parallels.`**`query`**(*action=None, command=None, args=None, method='GET', data=None*)
Make a web call to a Parallels provider

`salt.cloud.clouds.parallels.`**`script`**`(`*vm_*`)`
Return the script deployment object

`salt.cloud.clouds.parallels.`**`show_image`**`(`*kwargs, call=None*`)`
Show the details from Parallels concerning an image

`salt.cloud.clouds.parallels.`**`show_instance`**`(`*name, call=None*`)`
Show the details from Parallels concerning an instance

`salt.cloud.clouds.parallels.`**`start`**`(`*name, call=None*`)`
Start a node.

CLI Example:

```
salt-cloud -a start mymachine
```

`salt.cloud.clouds.parallels.`**`stop`**`(`*name, call=None*`)`
Stop a node.

CLI Example:

```
salt-cloud -a stop mymachine
```

`salt.cloud.clouds.parallels.`**`wait_until`**`(`*name, state, timeout=300*`)`
Wait until a specific state has been reached on a node

21.4.17 salt.cloud.clouds.profitbricks

ProfitBricks Cloud Module

The ProfitBricks SaltStack cloud module allows a ProfitBricks server to be automatically deployed and bootstraped with Salt.

depends profitbrick >= 3.1.0

The module requires ProfitBricks credentials to be supplied along with an existing virtual datacenter UUID where the server resources will reside. The server should also be assigned a public LAN, a private LAN, or both along with SSH key pairs. ...

Set up the cloud configuration at `/etc/salt/cloud.providers` or `/etc/salt/cloud.providers.d/profitbricks.conf`:

```
my-profitbricks-config:
  driver: profitbricks
  # The ProfitBricks login username
  username: user@example.com
  # The ProfitBricks login password
  password: secretpassword
  # The ProfitBricks virtual datacenter UUID
  datacenter_id: <UUID>
  # SSH private key filename
  ssh_private_key: /path/to/private.key
  # SSH public key filename
  ssh_public_key: /path/to/public.key
```

```
my-profitbricks-profile:
  provider: my-profitbricks-config
  # Name of a predefined server size.
```

(continues on next page)

(continued from previous page)

```
  size: Micro Instance
  # Assign CPU family to server.
  cpu_family: INTEL_XEON
  # Number of CPU cores to allocate to node (overrides server size).
  cores: 4
  # Amount of RAM in multiples of 256 MB (overrides server size).
  ram: 4096
  # The server availability zone.
  availability_zone: ZONE_1
  # Name or UUID of the HDD image to use.
  image: <UUID>
  # Image alias could be provided instead of image.
  # Example 'ubuntu:latest'
  #image_alias: <IMAGE_ALIAS>
  # Size of the node disk in GB (overrides server size).
  disk_size: 40
  # Type of disk (HDD or SSD).
  disk_type: SSD
  # Storage availability zone to use.
  disk_availability_zone: ZONE_2
  # Assign the server to the specified public LAN.
  public_lan: <ID>
  # Assign firewall rules to the network interface.
  public_firewall_rules:
    SSH:
      protocol: TCP
      port_range_start: 22
      port_range_end: 22
  # Assign the server to the specified private LAN.
  private_lan: <ID>
  # Enable NAT on the private NIC.
  nat: true
  # Assign additional volumes to the server.
  volumes:
    data-volume:
      disk_size: 500
      disk_availability_zone: ZONE_3
    log-volume:
      disk_size: 50
      disk_type: SSD
```

To use a private IP for connecting and bootstrapping node:

```
my-profitbricks-profile:
  ssh_interface: private_lan
```

Set `deploy` to False if Salt should not be installed on the node.

```
my-profitbricks-profile:
  deploy: False
```

`salt.cloud.clouds.profitbricks.`**`avail_images`**(*call=None*)

Return a list of the images that are on the provider

`salt.cloud.clouds.profitbricks.`**`avail_locations`**(*call=None*)

Return a dict of all available VM locations on the cloud provider with relevant data

`salt.cloud.clouds.profitbricks.`**`avail_sizes`**(*call=None*)

Return a dict of all available VM sizes on the cloud provider with relevant data. Latest version can be found at:

`salt.cloud.clouds.profitbricks.`**`create`**`(`*vm_*`)`
Create a single VM from a data dict

`salt.cloud.clouds.profitbricks.`**`create_datacenter`**`(`*call=None, kwargs=None*`)`
Creates a virtual datacenter based on supplied parameters.

CLI Example:

```
salt-cloud -f create_datacenter profitbricks name=mydatacenter
location=us/las description="my description"
```

`salt.cloud.clouds.profitbricks.`**`create_loadbalancer`**`(`*call=None, kwargs=None*`)`
Creates a loadbalancer within the datacenter from the provider config.

CLI Example:

```
salt-cloud -f create_loadbalancer profitbricks name=mylb
```

`salt.cloud.clouds.profitbricks.`**`destroy`**`(`*name, call=None*`)`
destroy a machine by name

Parameters

- **`name`** – name given to the machine
- **`call`** – call value in this case is 'action'

Returns array of booleans , true if successfully stopped and true if successfully removed

CLI Example:

```
salt-cloud -d vm_name
```

`salt.cloud.clouds.profitbricks.`**`get_configured_provider`**`()`
Return the first configured instance.

`salt.cloud.clouds.profitbricks.`**`get_conn`**`()`
Return a conn object for the passed VM data

`salt.cloud.clouds.profitbricks.`**`get_datacenter`**`(`*conn*`)`
Return the datacenter from the config provider datacenter ID

`salt.cloud.clouds.profitbricks.`**`get_datacenter_id`**`()`
Return datacenter ID from provider configuration

`salt.cloud.clouds.profitbricks.`**`get_dependencies`**`()`
Warn if dependencies are not met.

`salt.cloud.clouds.profitbricks.`**`get_disk_type`**`(`*vm_*`)`
Return the type of disk to use. Either 'HDD' (default) or 'SSD'.

`salt.cloud.clouds.profitbricks.`**`get_image`**`(`*vm_*`)`
Return the image object to use

`salt.cloud.clouds.profitbricks.`**`get_key_filename`**`(`*vm_*`)`
Check SSH private key file and return absolute path if exists.

`salt.cloud.clouds.profitbricks.`**`get_node`**`(`*conn, name*`)`
Return a node for the named VM

`salt.cloud.clouds.profitbricks.`**`get_public_keys`**`(`*vm_*`)`

Retrieve list of SSH public keys.

`salt.cloud.clouds.profitbricks.`**`get_size`**`(`*vm_*`)`

Return the VM's size object

`salt.cloud.clouds.profitbricks.`**`get_wait_timeout`**`(`*vm_*`)`

Return the wait_for_timeout for resource provisioning.

`salt.cloud.clouds.profitbricks.`**`list_datacenters`**`(`*conn=None, call=None*`)`

List all the data centers

CLI Example:

```
salt-cloud -f list_datacenters my-profitbricks-config
```

`salt.cloud.clouds.profitbricks.`**`list_images`**`(`*call=None, kwargs=None*`)`

List all the images with alias by location

CLI Example:

```
salt-cloud -f list_images my-profitbricks-config location=us/las
```

`salt.cloud.clouds.profitbricks.`**`list_loadbalancers`**`(`*call=None*`)`

Return a list of the loadbalancers that are on the provider

`salt.cloud.clouds.profitbricks.`**`list_nodes`**`(`*conn=None, call=None*`)`

Return a list of VMs that are on the provider

`salt.cloud.clouds.profitbricks.`**`list_nodes_full`**`(`*conn=None, call=None*`)`

Return a list of the VMs that are on the provider, with all fields

`salt.cloud.clouds.profitbricks.`**`reboot`**`(`*name, call=None*`)`

reboot a machine by name :param name: name given to the machine :param call: call value in this case is 'action' :return: true if successful

CLI Example:

```
salt-cloud -a reboot vm_name
```

`salt.cloud.clouds.profitbricks.`**`reserve_ipblock`**`(`*call=None, kwargs=None*`)`

Reserve the IP Block

`salt.cloud.clouds.profitbricks.`**`set_public_lan`**`(`*lan_id*`)`

Enables public Internet access for the specified public_lan. If no public LAN is available, then a new public LAN is created.

`salt.cloud.clouds.profitbricks.`**`show_instance`**`(`*name, call=None*`)`

Show the details from the provider concerning an instance

`salt.cloud.clouds.profitbricks.`**`signal_event`**`(`*vm_, event, description*`)`

`salt.cloud.clouds.profitbricks.`**`ssh_interface`**`(`*vm_*`)`

Return the ssh_interface type to connect to. Either 'public_ips' (default) or 'private_ips'.

`salt.cloud.clouds.profitbricks.`**`start`**`(`*name, call=None*`)`

start a machine by name :param name: name given to the machine :param call: call value in this case is 'action' :return: true if successful

CLI Example:

```
salt-cloud -a start vm_name
```

`salt.cloud.clouds.profitbricks.`**`stop`**(*name*, *call=None*)

stop a machine by name :param name: name given to the machine :param call: call value in this case is 'action' :return: true if successful

CLI Example:

```
salt-cloud -a stop vm_name
```

`salt.cloud.clouds.profitbricks.`**`version_compatible`**(*version*)

Checks profitbricks version

21.4.18 salt.cloud.clouds.proxmox

Proxmox Cloud Module

New in version 2014.7.0.

The Proxmox cloud module is used to control access to cloud providers using the Proxmox system (KVM / OpenVZ / LXC).

Set up the cloud configuration at `/etc/salt/cloud.providers` or `/etc/salt/cloud.providers.d/proxmox.conf`:

```
my-proxmox-config:
  # Proxmox account information
  user: myuser@pam or myuser@pve
  password: mypassword
  url: hypervisor.domain.tld
  driver: proxmox
  verify_ssl: True
```

maintainer Frank Klaassen <frank@cloudright.nl>

depends requests >= 2.2.1

depends IPy >= 0.81

`salt.cloud.clouds.proxmox.`**`avail_images`**(*call=None*, *location='local'*)

Return a list of the images that are on the provider

CLI Example:

```
salt-cloud --list-images my-proxmox-config
```

`salt.cloud.clouds.proxmox.`**`avail_locations`**(*call=None*)

Return a list of the hypervisors (nodes) which this Proxmox PVE machine manages

CLI Example:

```
salt-cloud --list-locations my-proxmox-config
```

`salt.cloud.clouds.proxmox.`**`create`**(*vm_*)

Create a single VM from a data dict

CLI Example:

```
salt-cloud -p proxmox-ubuntu vmhostname
```

salt.cloud.clouds.proxmox.**create_node**(*vm_*, *newid*)
Build and submit the requestdata to create a new node

salt.cloud.clouds.proxmox.**destroy**(*name*, *call=None*)
Destroy a node.

CLI Example:

```
salt-cloud --destroy mymachine
```

salt.cloud.clouds.proxmox.**get_configured_provider**()
Return the first configured instance.

salt.cloud.clouds.proxmox.**get_dependencies**()
Warn if dependencies aren't met.

salt.cloud.clouds.proxmox.**get_resources_nodes**(*call=None*, *resFilter=None*)
Retrieve all hypervisors (nodes) available on this environment CLI Example:

```
salt-cloud -f get_resources_nodes my-proxmox-config
```

salt.cloud.clouds.proxmox.**get_resources_vms**(*call=None*, *resFilter=None*, *includeConfig=True*)
Retrieve all VMs available on this environment

CLI Example:

```
salt-cloud -f get_resources_vms my-proxmox-config
```

salt.cloud.clouds.proxmox.**get_vm_status**(*vmid=None*, *name=None*)
Get the status for a VM, either via the ID or the hostname

salt.cloud.clouds.proxmox.**get_vmconfig**(*vmid*, *node=None*, *node_type='openvz'*)
Get VM configuration

salt.cloud.clouds.proxmox.**list_nodes**(*call=None*)
Return a list of the VMs that are managed by the provider

CLI Example:

```
salt-cloud -Q my-proxmox-config
```

salt.cloud.clouds.proxmox.**list_nodes_full**(*call=None*)
Return a list of the VMs that are on the provider

CLI Example:

```
salt-cloud -F my-proxmox-config
```

salt.cloud.clouds.proxmox.**list_nodes_select**(*call=None*)
Return a list of the VMs that are on the provider, with select fields

CLI Example:

```
salt-cloud -S my-proxmox-config
```

salt.cloud.clouds.proxmox.**query**(*conn_type*, *option*, *post_data=None*)
Execute the HTTP request to the API

`salt.cloud.clouds.proxmox.`**`script`**(*vm_*)
Return the script deployment object

`salt.cloud.clouds.proxmox.`**`set_vm_status`**(*status*, *name=None*, *vmid=None*)
Convenience function for setting VM status

`salt.cloud.clouds.proxmox.`**`show_instance`**(*name*, *call=None*)
Show the details from Proxmox concerning an instance

`salt.cloud.clouds.proxmox.`**`shutdown`**(*name=None*, *vmid=None*, *call=None*)
Shutdown a node via ACPI.

CLI Example:

```
salt-cloud -a shutdown mymachine
```

`salt.cloud.clouds.proxmox.`**`start`**(*name*, *vmid=None*, *call=None*)
Start a node.

CLI Example:

```
salt-cloud -a start mymachine
```

`salt.cloud.clouds.proxmox.`**`stop`**(*name*, *vmid=None*, *call=None*)
Stop a node ("pulling the plug").

CLI Example:

```
salt-cloud -a stop mymachine
```

`salt.cloud.clouds.proxmox.`**`wait_for_created`**(*upid*, *timeout=300*)
Wait until a the vm has been created successfully

`salt.cloud.clouds.proxmox.`**`wait_for_state`**(*vmid*, *state*, *timeout=300*)
Wait until a specific state has been reached on a node

21.4.19 salt.cloud.clouds.pyrax

Pyrax Cloud Module

PLEASE NOTE: This module is currently in early development, and considered to be experimental and unstable. It is not recommended for production use. Unless you are actively developing code in this module, you should use the OpenStack module instead.

`salt.cloud.clouds.pyrax.`**`get_configured_provider`**()
Return the first configured instance.

`salt.cloud.clouds.pyrax.`**`get_conn`**(*conn_type*)
Return a conn object for the passed VM data

`salt.cloud.clouds.pyrax.`**`get_dependencies`**()
Warn if dependencies aren't met.

`salt.cloud.clouds.pyrax.`**`queues_create`**(*call*, *kwargs*)

`salt.cloud.clouds.pyrax.`**`queues_delete`**(*call*, *kwargs*)

`salt.cloud.clouds.pyrax.`**`queues_exists`**(*call*, *kwargs*)

`salt.cloud.clouds.pyrax.`**`queues_show`**(*call*, *kwargs*)

21.4.20 salt.cloud.clouds.qingcloud

QingCloud Cloud Module

New in version 2015.8.0.

The QingCloud cloud module is used to control access to the QingCloud. http://www.qingcloud.com/

Use of this module requires the `access_key_id`, `secret_access_key`, `zone` and `key_filename` parameter to be set.

Set up the cloud configuration at `/etc/salt/cloud.providers` or `/etc/salt/cloud.providers.d/qingcloud.conf`:

```
my-qingcloud:
  driver: qingcloud
  access_key_id: AKIDMRTGYONNLTFFRBQJ
  secret_access_key: clYwH21U5UOmcov4aNV2V2XocaHCG3JZGcxEczFu
  zone: pek2
  key_filename: /path/to/your.pem
```

depends requests

`salt.cloud.clouds.qingcloud.`**`avail_images`**(*kwargs=None*, *call=None*)
Return a list of the images that are on the provider.

CLI Examples:

```
salt-cloud --list-images my-qingcloud
salt-cloud -f avail_images my-qingcloud zone=gd1
```

`salt.cloud.clouds.qingcloud.`**`avail_locations`**(*call=None*)
Return a dict of all available locations on the provider with relevant data.

CLI Examples:

```
salt-cloud --list-locations my-qingcloud
```

`salt.cloud.clouds.qingcloud.`**`avail_sizes`**(*kwargs=None*, *call=None*)
Return a list of the instance sizes that are on the provider.

CLI Examples:

```
salt-cloud --list-sizes my-qingcloud
salt-cloud -f avail_sizes my-qingcloud zone=pek2
```

`salt.cloud.clouds.qingcloud.`**`create`**(*vm_*)
Create a single instance from a data dict.

CLI Examples:

```
salt-cloud -p qingcloud-ubuntu-c1m1 hostname1
salt-cloud -m /path/to/mymap.sls -P
```

`salt.cloud.clouds.qingcloud.`**`destroy`**(*instance_id*, *call=None*)
Destroy an instance.

CLI Example:

```
salt-cloud -a destroy i-2f733r5n
salt-cloud -d i-2f733r5n
```

`salt.cloud.clouds.qingcloud.`**`get_configured_provider`**`()`
Return the first configured instance.

`salt.cloud.clouds.qingcloud.`**`get_dependencies`**`()`
Warn if dependencies aren't met.

`salt.cloud.clouds.qingcloud.`**`list_nodes`**`(`*call=None*`)`
Return a list of the instances that are on the provider.

CLI Examples:

```
salt-cloud -Q my-qingcloud
```

`salt.cloud.clouds.qingcloud.`**`list_nodes_full`**`(`*call=None*`)`
Return a list of the instances that are on the provider.

CLI Examples:

```
salt-cloud -F my-qingcloud
```

`salt.cloud.clouds.qingcloud.`**`list_nodes_min`**`(`*call=None*`)`
Return a list of the instances that are on the provider. Only a list of instances names, and their state, is returned.

CLI Examples:

```
salt-cloud -f list_nodes_min my-qingcloud
```

`salt.cloud.clouds.qingcloud.`**`list_nodes_select`**`(`*call=None*`)`
Return a list of the instances that are on the provider, with selected fields.

CLI Examples:

```
salt-cloud -S my-qingcloud
```

`salt.cloud.clouds.qingcloud.`**`query`**`(`*params=None*`)`
Make a web call to QingCloud IaaS API.

`salt.cloud.clouds.qingcloud.`**`reboot`**`(`*instance_id, call=None*`)`
Reboot an instance.

CLI Examples:

```
salt-cloud -a reboot i-2f733r5n
```

`salt.cloud.clouds.qingcloud.`**`script`**`(`*vm_*`)`
Return the script deployment object.

`salt.cloud.clouds.qingcloud.`**`show_image`**`(`*kwargs, call=None*`)`
Show the details from QingCloud concerning an image.

CLI Examples:

```
salt-cloud -f show_image my-qingcloud image=trustysrvx64c
salt-cloud -f show_image my-qingcloud image=trustysrvx64c,coreos4
salt-cloud -f show_image my-qingcloud image=trustysrvx64c zone=ap1
```

`salt.cloud.clouds.qingcloud.`**`show_instance`**`(`*instance_id, call=None, kwargs=None*`)`
Show the details from QingCloud concerning an instance.

CLI Examples:

```
salt-cloud -a show_instance i-2f733r5n
```

`salt.cloud.clouds.qingcloud.`**`start`**`(`*instance_id, call=None*`)`
Start an instance.

CLI Examples:

```
salt-cloud -a start i-2f733r5n
```

`salt.cloud.clouds.qingcloud.`**`stop`**`(`*instance_id, force=False, call=None*`)`
Stop an instance.

CLI Examples:

```
salt-cloud -a stop i-2f733r5n
salt-cloud -a stop i-2f733r5n force=True
```

21.4.21 salt.cloud.clouds.saltify

Saltify Module

The Saltify module is designed to install Salt on a remote machine, virtual or bare metal, using SSH. This module is useful for provisioning machines which are already installed, but not Salted.

Changed in version 2018.3.0: The wake_on_lan capability, and actions destroy, reboot, and query functions were added.

Use of this module requires some configuration in cloud profile and provider files as described in the *Gettting Started with Saltify* documentation.

`salt.cloud.clouds.saltify.`**`avail_images`**`(`*call=None*`)`
This function returns a list of images available for this cloud provider.

```
salt-cloud --list-images saltify
```

returns a list of available profiles.

..versionadded:: 2018.3.0

`salt.cloud.clouds.saltify.`**`avail_locations`**`(`*call=None*`)`
This function returns a list of locations available.

```
salt-cloud --list-locations my-cloud-provider
```

[saltify will always return an empty dictionary]

`salt.cloud.clouds.saltify.`**`avail_sizes`**`(`*call=None*`)`
This function returns a list of sizes available for this cloud provider.

```
salt-cloud --list-sizes saltify
```

[saltify always returns an empty dictionary] `salt.cloud.clouds.saltify.`**`create`**`(`*vm_*`)`

if configuration parameter `deploy` is `True`,

> Provision a single machine, adding its keys to the salt master

else,

> Test ssh connections to the machine

Configuration parameters:

- deploy: (see above)
- provider: name of entry in `salt/cloud.providers.d/???` file
- ssh_host: IP address or DNS name of the new machine
- ssh_username: name used to log in to the new machine
- ssh_password: password to log in (unless key_filename is used)
- key_filename: (optional) SSH private key for passwordless login
- ssh_port: (default=22) TCP port for SSH connection
- wake_on_lan_mac: (optional) hardware (MAC) address for wake on lan
- wol_sender_node: (optional) salt minion to send wake on lan command
- wol_boot_wait: (default=30) seconds to delay while client boots
- force_minion_config: (optional) replace the minion configuration files on the new machine

See also *Miscellaneous Salt Cloud Options* and *Getting Started with Saltify*

CLI Example:

```
salt-cloud -p mymachine my_new_id
```

`salt.cloud.clouds.saltify.`**`destroy`**`(`*name*, *call=None*`)`

Destroy a node.

New in version 2018.3.0.

Disconnect a minion from the master, and remove its keys.

Optionally, (if `remove_config_on_destroy` is `True`), disables salt-minion from running on the minion, and erases the Salt configuration files from it.

Optionally, (if `shutdown_on_destroy` is `True`), orders the minion to halt.

CLI Example:

```
salt-cloud --destroy mymachine
```

`salt.cloud.clouds.saltify.`**`get_configured_provider`**`()`

Return the first configured instance.

`salt.cloud.clouds.saltify.`**`list_nodes`**`(`*call=None*`)`

List the nodes which have salt-cloud:driver:saltify grains.

```
salt-cloud -Q
```

returns a list of dictionaries of defined standard fields.

..versionadded:: 2018.3.0

`salt.cloud.clouds.saltify.`**`list_nodes_full`**(*call=None*)
Lists complete information for all nodes.

```
salt-cloud -F
```

returns a list of dictionaries.

for 'saltify' minions, returns dict of grains (enhanced).

..versionadded:: 2018.3.0

`salt.cloud.clouds.saltify.`**`list_nodes_select`**(*call=None*)
Return a list of the minions that have salt-cloud grains, with select fields.

`salt.cloud.clouds.saltify.`**`reboot`**(*name*, *call=None*)
Reboot a saltify minion.

..versionadded:: 2018.3.0

name The name of the VM to reboot.

CLI Example:

```
salt-cloud -a reboot vm_name
```

`salt.cloud.clouds.saltify.`**`show_instance`**(*name*, *call=None*)
List the a single node, return dict of grains.

21.4.22 salt.cloud.clouds.scaleway

Scaleway Cloud Module

New in version 2015.8.0.

The Scaleway cloud module is used to interact with your Scaleway BareMetal Servers.

Use of this module only requires the `api_key` parameter to be set. Set up the cloud configuration at `/etc/salt/cloud.providers` or `/etc/salt/cloud.providers.d/scaleway.conf`:

```
scaleway-config:
  # Scaleway organization and token
  access_key: 0e604a2c-aea6-4081-acb2-e1d1258ef95c
  token: be8fd96b-04eb-4d39-b6ba-a9edbcf17f12
  driver: scaleway
```

`salt.cloud.clouds.scaleway.`**`avail_images`**(*call=None*)
Return a list of the images that are on the provider.

`salt.cloud.clouds.scaleway.`**`create`**(*server_*)
Create a single BareMetal server from a data dict.

`salt.cloud.clouds.scaleway.`**`create_node`**(*args*)
Create a node.

`salt.cloud.clouds.scaleway.`**`destroy`**(*name*, *call=None*)
Destroy a node. Will check termination protection and warn if enabled.

CLI Example: .. code-block:: bash

> salt-cloud –destroy mymachine

`salt.cloud.clouds.scaleway.`**`get_configured_provider`**`()`
Return the first configured instance.

`salt.cloud.clouds.scaleway.`**`get_image`**`(`*server_*`)`
Return the image object to use.

`salt.cloud.clouds.scaleway.`**`list_nodes`**`(`*call=None*`)`
Return a list of the BareMetal servers that are on the provider.

`salt.cloud.clouds.scaleway.`**`list_nodes_full`**`(`*call=None*`)`
Return a list of the BareMetal servers that are on the provider.

`salt.cloud.clouds.scaleway.`**`list_nodes_select`**`(`*call=None*`)`
Return a list of the BareMetal servers that are on the provider, with select fields.

`salt.cloud.clouds.scaleway.`**`query`**`(`*method='servers', server_id=None, command=None, args=None, http_method='get'*`)`
Make a call to the Scaleway API.

`salt.cloud.clouds.scaleway.`**`script`**`(`*server_*`)`
Return the script deployment object.

`salt.cloud.clouds.scaleway.`**`show_instance`**`(`*name, call=None*`)`
Show the details from a Scaleway BareMetal server.

21.4.23 salt.cloud.clouds.softlayer

SoftLayer Cloud Module

The SoftLayer cloud module is used to control access to the SoftLayer VPS system.

Use of this module only requires the `apikey` parameter. Set up the cloud configuration at:

`/etc/salt/cloud.providers` or `/etc/salt/cloud.providers.d/softlayer.conf`:

```
my-softlayer-config:
  # SoftLayer account api key
  user: MYLOGIN
  apikey: JVkbSJDGHSDKUKSDJfhsdklfjgsjdkflhjlsdfffhgdgjkenrtuinv
  driver: softlayer
```

The SoftLayer Python Library needs to be installed in order to use the SoftLayer salt.cloud modules. See: https://pypi.python.org/pypi/SoftLayer

depends softlayer

`salt.cloud.clouds.softlayer.`**`avail_images`**`(`*call=None*`)`
Return a dict of all available VM images on the cloud provider.

`salt.cloud.clouds.softlayer.`**`avail_locations`**`(`*call=None*`)`
List all available locations

`salt.cloud.clouds.softlayer.`**`avail_sizes`**`(`*call=None*`)`
Return a dict of all available VM sizes on the cloud provider with relevant data. This data is provided in three dicts.

`salt.cloud.clouds.softlayer.`**`create`**`(`*vm_*`)`
Create a single VM from a data dict

`salt.cloud.clouds.softlayer.`**`destroy`**`(`*name, call=None*`)`
Destroy a node.

CLI Example:

```
salt-cloud --destroy mymachine
```

`salt.cloud.clouds.softlayer.`**`get_configured_provider`**`()`
Return the first configured instance.

`salt.cloud.clouds.softlayer.`**`get_conn`**`(`*service='SoftLayer_Virtual_Guest'*`)`
Return a conn object for the passed VM data

`salt.cloud.clouds.softlayer.`**`get_dependencies`**`()`
Warn if dependencies aren't met.

`salt.cloud.clouds.softlayer.`**`get_location`**`(`*vm_=None*`)`

Return the location to use, in this order:

- CLI parameter
- VM parameter
- Cloud profile setting

`salt.cloud.clouds.softlayer.`**`list_custom_images`**`(`*call=None*`)`
Return a dict of all custom VM images on the cloud provider.

`salt.cloud.clouds.softlayer.`**`list_nodes`**`(`*call=None*`)`
Return a list of the VMs that are on the provider

`salt.cloud.clouds.softlayer.`**`list_nodes_full`**`(`*mask='mask[id]', call=None*`)`
Return a list of the VMs that are on the provider

`salt.cloud.clouds.softlayer.`**`list_nodes_select`**`(`*call=None*`)`
Return a list of the VMs that are on the provider, with select fields

`salt.cloud.clouds.softlayer.`**`list_vlans`**`(`*call=None*`)`
List all VLANs associated with the account

`salt.cloud.clouds.softlayer.`**`script`**`(`*vm_*`)`
Return the script deployment object

`salt.cloud.clouds.softlayer.`**`show_instance`**`(`*name, call=None*`)`
Show the details from SoftLayer concerning a guest

21.4.24 salt.cloud.clouds.softlayer_hw

SoftLayer HW Cloud Module

The SoftLayer HW cloud module is used to control access to the SoftLayer hardware cloud system

Use of this module only requires the `apikey` parameter. Set up the cloud configuration at:

`/etc/salt/cloud.providers` or `/etc/salt/cloud.providers.d/softlayer.conf`:

```
my-softlayer-config:
  # SoftLayer account api key
  user: MYLOGIN
  apikey: JVkbSJDGHSDKUKSDJfhsdklfjgsjdkflhjlsdfffhgdgjkenrtuinv
  driver: softlayer_hw
```

The SoftLayer Python Library needs to be installed in order to use the SoftLayer salt.cloud modules. See: https://pypi.python.org/pypi/SoftLayer

depends softlayer

`salt.cloud.clouds.softlayer_hw.`**`avail_images`**`(`*call=None*`)`
Return a dict of all available VM images on the cloud provider.

`salt.cloud.clouds.softlayer_hw.`**`avail_locations`**`(`*call=None*`)`
List all available locations

`salt.cloud.clouds.softlayer_hw.`**`avail_sizes`**`(`*call=None*`)`
Return a dict of all available VM sizes on the cloud provider with relevant data. This data is provided in three dicts.

`salt.cloud.clouds.softlayer_hw.`**`create`**`(`*vm_*`)`
Create a single VM from a data dict

`salt.cloud.clouds.softlayer_hw.`**`destroy`**`(`*name*, *call=None*`)`
Destroy a node.

CLI Example:

```
salt-cloud --destroy mymachine
```

`salt.cloud.clouds.softlayer_hw.`**`get_configured_provider`**`()`
Return the first configured instance.

`salt.cloud.clouds.softlayer_hw.`**`get_conn`**`(`*service='SoftLayer_Hardware'*`)`
Return a conn object for the passed VM data

`salt.cloud.clouds.softlayer_hw.`**`get_dependencies`**`()`
Warn if dependencies aren't met.

`salt.cloud.clouds.softlayer_hw.`**`get_location`**`(`*vm_=None*`)`

Return the location to use, in this order:

- CLI parameter
- VM parameter
- Cloud profile setting

`salt.cloud.clouds.softlayer_hw.`**`list_nodes`**`(`*call=None*`)`
Return a list of the VMs that are on the provider

`salt.cloud.clouds.softlayer_hw.`**`list_nodes_full`**`(`*mask='mask[id, hostname, primaryIpAddress, primaryBackendIpAddress, processorPhysicalCoreAmount, memoryCount]', call=None*`)`
Return a list of the VMs that are on the provider

`salt.cloud.clouds.softlayer_hw.`**`list_nodes_select`**`(`*call=None*`)`
Return a list of the VMs that are on the provider, with select fields

`salt.cloud.clouds.softlayer_hw.`**`list_vlans`**`(`*call=None*`)`
List all VLANs associated with the account

`salt.cloud.clouds.softlayer_hw.`**`script`**`(`*vm_*`)`
Return the script deployment object

`salt.cloud.clouds.softlayer_hw.`**`show_all_categories`**`(`*call=None*`)`
Return a dict of all available categories on the cloud provider.

New in version 2016.3.0.

`salt.cloud.clouds.softlayer_hw.`**`show_all_prices`**`(`*call=None, kwargs=None*`)`
Return a dict of all prices on the cloud provider.

`salt.cloud.clouds.softlayer_hw.`**`show_instance`**`(`*name, call=None*`)`
Show the details from SoftLayer concerning a guest

`salt.cloud.clouds.softlayer_hw.`**`show_pricing`**`(`*kwargs=None, call=None*`)`
Show pricing for a particular profile. This is only an estimate, based on unofficial pricing sources.

CLI Examples:

```
salt-cloud -f show_pricing my-softlayerhw-config profile=my-profile
```

If pricing sources have not been cached, they will be downloaded. Once they have been cached, they will not be updated automatically. To manually update all prices, use the following command:

```
salt-cloud -f update_pricing <provider>
```

New in version 2015.8.0.

21.4.25 salt.cloud.clouds.tencentcloud

Tencent Cloud Cloud Module

New in version 3000.

The Tencent Cloud Cloud Module is used to control access to the Tencent Cloud instance. https://intl.cloud.tencent.com/

To use this module, set up the cloud configuration at `/etc/salt/cloud.providers` or `/etc/salt/cloud.providers.d/*.conf`:

```
my-tencentcloud-config:
  driver: tencentcloud
  # Tencent Cloud Secret Id
  id: AKIDA64pOio9BMemkApzevX0HS169S4b750A
  # Tencent Cloud Secret Key
  key: 8r2xmPn0C5FDvRAlmcJimiTZKVRsk260
  # Tencent Cloud Region
  location: ap-guangzhou
```

depends tencentcloud-sdk-python

`salt.cloud.clouds.tencentcloud.`**`avail_images`**`(`*call=None*`)`
Return Tencent Cloud available image

CLI Example:

```
salt-cloud --list-images my-tencentcloud-config
salt-cloud -f avail_images my-tencentcloud-config
```

`salt.cloud.clouds.tencentcloud.`**`avail_locations`**`(`*call=None*`)`
Return Tencent Cloud available region

CLI Example:

```
salt-cloud --list-locations my-tencentcloud-config
salt-cloud -f avail_locations my-tencentcloud-config
```

`salt.cloud.clouds.tencentcloud.`**`avail_sizes`**`(`*call=None*`)`

Return Tencent Cloud available instance type

CLI Example:

```
salt-cloud --list-sizes my-tencentcloud-config
salt-cloud -f avail_sizes my-tencentcloud-config
```

`salt.cloud.clouds.tencentcloud.`**`create`**`(`*vm_*`)`

Create a single Tencent Cloud instance from a data dict.

Tencent Cloud profiles require a `provider`, `availability_zone`, `image` and `size`. Set up profile at `/etc/salt/cloud.profiles` or `/etc/salt/cloud.profiles.d/*.conf`:

```
tencentcloud-guangzhou-s1sm1:
    provider: my-tencentcloud-config
    availability_zone: ap-guangzhou-3
    image: img-31tjrtph
    size: S1.SMALL1
    allocate_public_ip: True
    internet_max_bandwidth_out: 1
    password: '153e41ec96140152'
    securitygroups:
        - sg-5e90804b
```

CLI Examples:

```
salt-cloud -p tencentcloud-guangzhou-s1 myinstance
```

`salt.cloud.clouds.tencentcloud.`**`destroy`**`(`*name, call=None*`)`

Destroy a Tencent Cloud instance

CLI Example:

```
salt-cloud -a destroy myinstance
salt-cloud -d myinstance
```

`salt.cloud.clouds.tencentcloud.`**`get_configured_provider`**`()`

Return the first configured instance.

`salt.cloud.clouds.tencentcloud.`**`get_dependencies`**`()`

Warn if dependencies aren't met.

`salt.cloud.clouds.tencentcloud.`**`get_provider_client`**`(`*name=None*`)`

Return a new provider client

`salt.cloud.clouds.tencentcloud.`**`list_availability_zones`**`(`*call=None*`)`

Return all Tencent Cloud availability zones in current region

CLI Example:

```
salt-cloud -f list_availability_zones my-tencentcloud-config
```

`salt.cloud.clouds.tencentcloud.`**`list_custom_images`**`(`*call=None*`)`

Return all Tencent Cloud images in current region

CLI Example:

```
salt-cloud -f list_custom_images my-tencentcloud-config
```

`salt.cloud.clouds.tencentcloud.`**`list_nodes`**(*call=None*)
Return a list of instances that are on the provider

CLI Examples:

```
salt-cloud -Q
```

`salt.cloud.clouds.tencentcloud.`**`list_nodes_full`**(*call=None*)
Return a list of instances that are on the provider, with full details

CLI Examples:

```
salt-cloud -F
```

`salt.cloud.clouds.tencentcloud.`**`list_nodes_min`**(*call=None*)
Return a list of instances that are on the provider, Only names, and their state, is returned.

CLI Examples:

```
salt-cloud -f list_nodes_min my-tencentcloud-config
```

`salt.cloud.clouds.tencentcloud.`**`list_nodes_select`**(*call=None*)
Return a list of instances that are on the provider, with select fields

CLI Examples:

```
salt-cloud -S
```

`salt.cloud.clouds.tencentcloud.`**`list_securitygroups`**(*call=None*)
Return all Tencent Cloud security groups in current region

CLI Example:

```
salt-cloud -f list_securitygroups my-tencentcloud-config
```

`salt.cloud.clouds.tencentcloud.`**`reboot`**(*name*, *call=None*)
Reboot a Tencent Cloud instance

CLI Examples:

```
salt-cloud -a reboot myinstance
```

`salt.cloud.clouds.tencentcloud.`**`script`**(*vm_*)
Return the script deployment object

`salt.cloud.clouds.tencentcloud.`**`show_disk`**(*name*, *call=None*)
Show the disk details of Tencent Cloud instance

CLI Examples:

```
salt-cloud -a show_disk myinstance
```

`salt.cloud.clouds.tencentcloud.`**`show_image`**(*kwargs*, *call=None*)
Show the details of Tencent Cloud image

CLI Examples:

```
salt-cloud -f show_image tencentcloud image=img-31tjrtph
```

`salt.cloud.clouds.tencentcloud.`**`show_instance`**`(`*name*, *call=None*`)`

Show the details of Tencent Cloud instance

CLI Examples:

```
salt-cloud -a show_instance myinstance
```

`salt.cloud.clouds.tencentcloud.`**`start`**`(`*name*, *call=None*`)`

Start a Tencent Cloud instance Notice: the instance state must be stopped

CLI Examples:

```
salt-cloud -a start myinstance
```

`salt.cloud.clouds.tencentcloud.`**`stop`**`(`*name*, *force=False*, *call=None*`)`

Stop a Tencent Cloud running instance Note: use *force=True* to make force stop

CLI Examples:

```
salt-cloud -a stop myinstance
salt-cloud -a stop myinstance force=True
```

21.4.26 salt.cloud.clouds.vagrant

Vagrant Cloud Driver

The Vagrant cloud is designed to ”vagrant up” a virtual machine as a Salt minion.

Use of this module requires some configuration in cloud profile and provider files as described in the *Getting Started with Vagrant* documentation.

New in version 2018.3.0.

`salt.cloud.clouds.vagrant.`**`avail_images`**`(`*call=None*`)`

This function returns a list of images available for this cloud provider. vagrant will return a list of profiles. salt-cloud –list-images my-cloud-provider

`salt.cloud.clouds.vagrant.`**`avail_locations`**`(`*call=None*`)`

This function returns a list of locations available.

CLI Example:

```
salt-cloud --list-locations my-cloud-provider

# \[ vagrant will always returns an empty dictionary \]
```

`salt.cloud.clouds.vagrant.`**`avail_sizes`**`(`*call=None*`)`

This function returns a list of sizes available for this cloud provider.

CLI Example:

```
salt-cloud --list-sizes my-cloud-provider

# \[ vagrant always returns an empty dictionary \]
```

`salt.cloud.clouds.vagrant.`**`create`**`(`*vm_*`)`

Provision a single machine

CLI Example:

```
salt-cloud -p my_profile new_node_1
```

`salt.cloud.clouds.vagrant.`**`destroy`**`(`*name*, *call=None*`)`
Destroy a node.

CLI Example:

```
salt-cloud --destroy mymachine
```

`salt.cloud.clouds.vagrant.`**`get_configured_provider`**`()`
Return the first configured instance.

`salt.cloud.clouds.vagrant.`**`list_nodes`**`(`*call=None*`)`
List the nodes which have salt-cloud:driver:vagrant grains.

CLI Example:

```
salt-cloud -Q
```

`salt.cloud.clouds.vagrant.`**`list_nodes_full`**`(`*call=None*`)`
List the nodes, ask all 'vagrant' minions, return dict of grains (enhanced).

CLI Example:

```
salt-call -F
```

`salt.cloud.clouds.vagrant.`**`list_nodes_select`**`(`*call=None*`)`
Return a list of the minions that have salt-cloud grains, with select fields.

`salt.cloud.clouds.vagrant.`**`reboot`**`(`*name*, *call=None*`)`
Reboot a vagrant minion.

name The name of the VM to reboot.

CLI Example:

```
salt-cloud -a reboot vm_name
```

`salt.cloud.clouds.vagrant.`**`show_instance`**`(`*name*, *call=None*`)`
List the a single node, return dict of grains.

21.4.27 salt.cloud.clouds.virtualbox

A salt cloud provider that lets you use virtualbox on your machine and act as a cloud.

depends vboxapi

For now this will only clone existing VMs. It's best to create a template from which we will clone.

Followed https://docs.saltstack.com/en/latest/topics/cloud/cloud.html#non-libcloud-based-modules to create this.

Dicts provided by salt:

__opts__ [contains the options used to run Salt Cloud,] as well as a set of configuration and environment variables

`salt.cloud.clouds.virtualbox.`**`create`**`(`*vm_info*`)`
Creates a virtual machine from the given VM information

This is what is used to request a virtual machine to be created by the cloud provider, wait for it to become available, and then (optionally) log in and install Salt on it.

Events fired:

This function fires the event `salt/cloud/vm_name/creating`, with the payload containing the names of the VM, profile, and provider.

@param vm_info

```
{
    name: <str>
    profile: <dict>
    driver: <provider>:<profile>
    clonefrom: <vm_name>
    clonemode: <mode> (default: state, choices: state, child, all)
}
```

@type vm_info dict @return dict of resulting vm. !!!Passwords can and should be included!!!

`salt.cloud.clouds.virtualbox.`**`destroy`**`(`*name*, *call=None*`)`

This function irreversibly destroys a virtual machine on the cloud provider. Before doing so, it should fire an event on the Salt event bus.

The tag for this event is *salt/cloud/<vm name>/destroying*. Once the virtual machine has been destroyed, another event is fired. The tag for that event is *salt/cloud/<vm name>/destroyed*.

Dependencies: list_nodes

@param name: @type name: str @param call: @type call: @return: True if all went well, otherwise an error message @rtype: bool|str

`salt.cloud.clouds.virtualbox.`**`list_nodes`**`(`*kwargs=None*, *call=None*`)`

This function returns a list of nodes available on this cloud provider, using the following fields:

id (str) image (str) size (str) state (str) private_ips (list) public_ips (list)

No other fields should be returned in this function, and all of these fields should be returned, even if empty. The private_ips and public_ips fields should always be of a list type, even if empty, and the other fields should always be of a str type. This function is normally called with the -Q option:

```
salt-cloud -Q
```

@param kwargs: @type kwargs: @param call: @type call: @return: @rtype:

`salt.cloud.clouds.virtualbox.`**`list_nodes_full`**`(`*kwargs=None*, *call=None*`)`

All information available about all nodes should be returned in this function. The fields in the list_nodes() function should also be returned, even if they would not normally be provided by the cloud provider.

This is because some functions both within Salt and 3rd party will break if an expected field is not present. This function is normally called with the -F option:

```
salt-cloud -F
```

@param kwargs: @type kwargs: @param call: @type call: @return: @rtype:

`salt.cloud.clouds.virtualbox.`**`list_nodes_select`**`(`*call=None*`)`

Return a list of the VMs that are on the provider, with select fields

`salt.cloud.clouds.virtualbox.`**`map_clonemode`**`(`*vm_info*`)`

Convert the virtualbox config file values for clone_mode into the integers the API requires

`salt.cloud.clouds.virtualbox.`**`show_image`**`(`*kwargs*, *call=None*`)`

Show the details of an image

`salt.cloud.clouds.virtualbox.`**`start`**(*name*, *call=None*)

Start a machine. @param name: Machine to start @type name: str @param call: Must be "action" @type call: str

`salt.cloud.clouds.virtualbox.`**`stop`**(*name*, *call=None*)

Stop a running machine. @param name: Machine to stop @type name: str @param call: Must be "action" @type call: str

21.4.28 salt.cloud.clouds.vmware

VMware Cloud Module

New in version 2015.5.4.

The VMware cloud module allows you to manage VMware ESX, ESXi, and vCenter.

See *Getting started with VMware* to get started.

codeauthor Nitin Madhok <nmadhok@clemson.edu>

Dependencies

- pyVmomi Python Module

pyVmomi

PyVmomi can be installed via pip:

```
pip install pyVmomi
```

Note: Version 6.0 of pyVmomi has some problems with SSL error handling on certain versions of Python. If using version 6.0 of pyVmomi, Python 2.6, Python 2.7.9, or newer must be present. This is due to an upstream dependency in pyVmomi 6.0 that is not supported in Python versions 2.7 to 2.7.8. If the version of Python is not in the supported range, you will need to install an earlier version of pyVmomi. See Issue #29537 for more information.

Based on the note above, to install an earlier version of pyVmomi than the version currently listed in PyPi, run the following:

```
pip install pyVmomi==5.5.0.2014.1.1
```

The 5.5.0.2014.1.1 is a known stable version that this original VMware cloud driver was developed against.

Note: Ensure python pyVmomi module is installed by running following one-liner check. The output should be 0.

```
python -c "import pyVmomi" ; echo $?
```

Configuration

To use this module, set up the vCenter or ESX/ESXi URL, username and password in the cloud configuration at `/etc/salt/cloud.providers` or `/etc/salt/cloud.providers.d/vmware.conf`:

```
my-vmware-config:
  driver: vmware
  user: 'DOMAIN\user'
  password: 'verybadpass'
  url: '10.20.30.40'

vcenter01:
  driver: vmware
  user: 'DOMAIN\user'
  password: 'verybadpass'
  url: 'vcenter01.domain.com'
  protocol: 'https'
  port: 443

vcenter02:
  driver: vmware
  user: 'DOMAIN\user'
  password: 'verybadpass'
  url: 'vcenter02.domain.com'
  protocol: 'http'
  port: 80

esx01:
  driver: vmware
  user: 'admin'
  password: 'verybadpass'
  url: 'esx01.domain.com'
```

Note: Optionally, `protocol` and `port` can be specified if the vCenter server is not using the defaults. Default is `protocol: https` and `port: 443`.

Note: Changed in version 2015.8.0.

The `provider` parameter in cloud provider configuration was renamed to `driver`. This change was made to avoid confusion with the `provider` parameter that is used in cloud profile configuration. Cloud provider configuration now uses `driver` to refer to the salt-cloud driver that provides the underlying functionality to connect to a cloud provider, while cloud profile configuration continues to use `provider` to refer to the cloud provider configuration that you define.

To test the connection for `my-vmware-config` specified in the cloud configuration, run *`test_vcenter_connection()`*

`salt.cloud.clouds.vmware.`**`add_host`**(*kwargs=None*, *call=None*)

Add a host system to the specified cluster or datacenter in this VMware environment

Note: To use this function, you need to specify `esxi_host_user` and `esxi_host_password` under your provider configuration set up at `/etc/salt/cloud.providers` or `/etc/salt/cloud.providers.d/vmware.conf`:

```
vcenter01:
  driver: vmware
  user: 'DOMAIN\user'
  password: 'verybadpass'
```

(continues on next page)

(continued from previous page)

```
  url: 'vcenter01.domain.com'

  # Required when adding a host system
  esxi_host_user: 'root'
  esxi_host_password: 'myhostpassword'
  # Optional fields that can be specified when adding a host system
  esxi_host_ssl_thumbprint:
↪'12:A3:45:B6:CD:7E:F8:90:A1:BC:23:45:D6:78:9E:FA:01:2B:34:CD'
```

The SSL thumbprint of the host system can be optionally specified by setting `esxi_host_ssl_thumbprint` under your provider configuration. To get the SSL thumbprint of the host system, execute the following command from a remote server:

```
echo -n | openssl s_client -connect <YOUR-HOSTSYSTEM-DNS/IP>:443 2>/dev/null |
↪openssl x509 -noout -fingerprint -sha1
```

CLI Example:

```
salt-cloud -f add_host my-vmware-config host="myHostSystemName" cluster=
↪"myClusterName"
salt-cloud -f add_host my-vmware-config host="myHostSystemName" datacenter=
↪"myDatacenterName"
```

salt.cloud.clouds.vmware.**avail_images**(*call=None*)

Return a list of all the templates present in this VMware environment with basic details

CLI Example:

```
salt-cloud --list-images my-vmware-config
```

salt.cloud.clouds.vmware.**avail_locations**(*call=None*)

Return a list of all the available locations/datacenters in this VMware environment

CLI Example:

```
salt-cloud --list-locations my-vmware-config
```

salt.cloud.clouds.vmware.**avail_sizes**(*call=None*)

Return a list of all the available sizes in this VMware environment.

CLI Example:

```
salt-cloud --list-sizes my-vmware-config
```

Note: Since sizes are built into templates, this function will return an empty dictionary.

salt.cloud.clouds.vmware.**build_clonespec**(*config_spec, object_ref, reloc_spec, template*)

Returns the clone spec

salt.cloud.clouds.vmware.**connect_host**(*kwargs=None, call=None*)

Connect the specified host system in this VMware environment

CLI Example:

```
salt-cloud -f connect_host my-vmware-config host="myHostSystemName"
```

`salt.cloud.clouds.vmware.`**`convert_to_template`**(*name, kwargs=None, call=None*)

Convert the specified virtual machine to template.

CLI Example:

```
salt-cloud -a convert_to_template vmname
```

`salt.cloud.clouds.vmware.`**`create`**(*vm_*)

To create a single VM in the VMware environment.

Sample profile and arguments that can be specified in it can be found *here*.

CLI Example:

```
salt-cloud -p vmware-centos6.5 vmname
```

`salt.cloud.clouds.vmware.`**`create_cluster`**(*kwargs=None, call=None*)

Create a new cluster under the specified datacenter in this VMware environment

CLI Example:

```
salt-cloud -f create_cluster my-vmware-config name="myNewCluster" datacenter=
↪"datacenterName"
```

`salt.cloud.clouds.vmware.`**`create_datacenter`**(*kwargs=None, call=None*)

Create a new data center in this VMware environment

CLI Example:

```
salt-cloud -f create_datacenter my-vmware-config name="MyNewDatacenter"
```

`salt.cloud.clouds.vmware.`**`create_datastore_cluster`**(*kwargs=None, call=None*)

Create a new datastore cluster for the specified datacenter in this VMware environment

CLI Example:

```
salt-cloud -f create_datastore_cluster my-vmware-config name="datastoreClusterName
↪" datacenter="datacenterName"
```

`salt.cloud.clouds.vmware.`**`create_folder`**(*kwargs=None, call=None*)

Create the specified folder path in this VMware environment

Note: To create a Host and Cluster Folder under a Datacenter, specify `path="/yourDatacenterName/host/yourFolderName"`

To create a Network Folder under a Datacenter, specify `path="/yourDatacenterName/network/yourFolderName"`

To create a Storage Folder under a Datacenter, specify `path="/yourDatacenterName/datastore/yourFolderName"`

To create a VM and Template Folder under a Datacenter, specify `path="/yourDatacenterName/vm/yourFolderName"`

CLI Example:

```
salt-cloud -f create_folder my-vmware-config path="/Local/a/b/c"
salt-cloud -f create_folder my-vmware-config path="/MyDatacenter/vm/MyVMFolder"
salt-cloud -f create_folder my-vmware-config path="/MyDatacenter/host/MyHostFolder
↪"
salt-cloud -f create_folder my-vmware-config path="/MyDatacenter/network/
↪MyNetworkFolder"
salt-cloud -f create_folder my-vmware-config path="/MyDatacenter/storage/
↪MyStorageFolder"
```

`salt.cloud.clouds.vmware.`**`create_snapshot`**(*name*, *kwargs=None*, *call=None*)

Create a snapshot of the specified virtual machine in this VMware environment

---

Note: If the VM is powered on, the internal state of the VM (memory dump) is included in the snapshot by default which will also set the power state of the snapshot to "powered on". You can set `memdump=False` to override this. This field is ignored if the virtual machine is powered off or if the VM does not support snapshots with memory dumps. Default is `memdump=True`

Note: If the VM is powered on when the snapshot is taken, VMware Tools can be used to quiesce the file system in the virtual machine by setting `quiesce=True`. This field is ignored if the virtual machine is powered off; if VMware Tools are not available or if `memdump=True`. Default is `quiesce=False`

CLI Example:

```
salt-cloud -a create_snapshot vmname snapshot_name="mySnapshot"
salt-cloud -a create_snapshot vmname snapshot_name="mySnapshot" [description="My
↪snapshot"] [memdump=False] [quiesce=True]
```

`salt.cloud.clouds.vmware.`**`destroy`**(*name*, *call=None*)

To destroy a VM from the VMware environment

CLI Example:

```
salt-cloud -d vmname
salt-cloud --destroy vmname
salt-cloud -a destroy vmname
```

`salt.cloud.clouds.vmware.`**`disconnect_host`**(*kwargs=None*, *call=None*)

Disconnect the specified host system in this VMware environment

CLI Example:

```
salt-cloud -f disconnect_host my-vmware-config host="myHostSystemName"
```

`salt.cloud.clouds.vmware.`**`enter_maintenance_mode`**(*kwargs=None*, *call=None*)

To put the specified host system in maintenance mode in this VMware environment

CLI Example:

```
salt-cloud -f enter_maintenance_mode my-vmware-config host="myHostSystemName"
```

`salt.cloud.clouds.vmware.`**`exit_maintenance_mode`**(*kwargs=None*, *call=None*)

To take the specified host system out of maintenance mode in this VMware environment

CLI Example:

```
salt-cloud -f exit_maintenance_mode my-vmware-config host="myHostSystemName"
```

`salt.cloud.clouds.vmware.`**`get_clonespec_for_valid_snapshot`**(*config_spec*, *object_ref*, *reloc_spec*, *template*, *vm_*)

return clonespec only if values are valid

`salt.cloud.clouds.vmware.`**`get_vcenter_version`**(*kwargs=None*, *call=None*)

Show the vCenter Server version with build number.

CLI Example:

```
salt-cloud -f get_vcenter_version my-vmware-config
```

`salt.cloud.clouds.vmware.`**`handle_snapshot`**(*config_spec*, *object_ref*, *reloc_spec*, *template*, *vm_*)

Returns a clone spec for cloning from shapshots :rtype vim.vm.CloneSpec

`salt.cloud.clouds.vmware.`**`list_clusters`**(*kwargs=None*, *call=None*)

List all the clusters for this VMware environment

CLI Example:

```
salt-cloud -f list_clusters my-vmware-config
```

`salt.cloud.clouds.vmware.`**`list_clusters_by_datacenter`**(*kwargs=None*, *call=None*)

List clusters for each datacenter; or clusters for a specified datacenter in this VMware environment

To list clusters for each datacenter:

CLI Example:

```
salt-cloud -f list_clusters_by_datacenter my-vmware-config
```

To list clusters for a specified datacenter:

CLI Example:

```
salt-cloud -f list_clusters_by_datacenter my-vmware-config datacenter=
↪"datacenterName"
```

`salt.cloud.clouds.vmware.`**`list_datacenters`**(*kwargs=None*, *call=None*)

List all the data centers for this VMware environment

CLI Example:

```
salt-cloud -f list_datacenters my-vmware-config
```

`salt.cloud.clouds.vmware.`**`list_datastore_clusters`**(*kwargs=None*, *call=None*)

List all the datastore clusters for this VMware environment

CLI Example:

```
salt-cloud -f list_datastore_clusters my-vmware-config
```

`salt.cloud.clouds.vmware.`**`list_datastores`**(*kwargs=None*, *call=None*)

List all the datastores for this VMware environment

CLI Example:

```
salt-cloud -f list_datastores my-vmware-config
```

`salt.cloud.clouds.vmware.`**`list_dvs`**`(`*kwargs=None, call=None*`)`

List all the distributed virtual switches for this VMware environment

CLI Example:

```
salt-cloud -f list_dvs my-vmware-config
```

`salt.cloud.clouds.vmware.`**`list_folders`**`(`*kwargs=None, call=None*`)`

List all the folders for this VMware environment

CLI Example:

```
salt-cloud -f list_folders my-vmware-config
```

`salt.cloud.clouds.vmware.`**`list_hbas`**`(`*kwargs=None, call=None*`)`

List all HBAs for each host system; or all HBAs for a specified host system; or HBAs of specified type for each host system; or HBAs of specified type for a specified host system in this VMware environment

Note: You can specify type as either `parallel`, `iscsi`, `block` or `fibre`.

To list all HBAs for each host system:

CLI Example:

```
salt-cloud -f list_hbas my-vmware-config
```

To list all HBAs for a specified host system:

CLI Example:

```
salt-cloud -f list_hbas my-vmware-config host="hostSystemName"
```

To list HBAs of specified type for each host system:

CLI Example:

```
salt-cloud -f list_hbas my-vmware-config type="HBAType"
```

To list HBAs of specified type for a specified host system:

CLI Example:

```
salt-cloud -f list_hbas my-vmware-config host="hostSystemName" type="HBAtype"
```

`salt.cloud.clouds.vmware.`**`list_hosts`**`(`*kwargs=None, call=None*`)`

List all the hosts for this VMware environment

CLI Example:

```
salt-cloud -f list_hosts my-vmware-config
```

`salt.cloud.clouds.vmware.`**`list_hosts_by_cluster`**`(`*kwargs=None, call=None*`)`

List hosts for each cluster; or hosts for a specified cluster in this VMware environment

To list hosts for each cluster:

CLI Example:

```
salt-cloud -f list_hosts_by_cluster my-vmware-config
```

To list hosts for a specified cluster:

CLI Example:

```
salt-cloud -f list_hosts_by_cluster my-vmware-config cluster="clusterName"
```

salt.cloud.clouds.vmware.**list_hosts_by_datacenter**(*kwargs=None*, *call=None*)

List hosts for each datacenter; or hosts for a specified datacenter in this VMware environment

To list hosts for each datacenter:

CLI Example:

```
salt-cloud -f list_hosts_by_datacenter my-vmware-config
```

To list hosts for a specified datacenter:

CLI Example:

```
salt-cloud -f list_hosts_by_datacenter my-vmware-config datacenter="datacenterName
↪"
```

salt.cloud.clouds.vmware.**list_networks**(*kwargs=None*, *call=None*)

List all the standard networks for this VMware environment

CLI Example:

```
salt-cloud -f list_networks my-vmware-config
```

salt.cloud.clouds.vmware.**list_nodes**(*kwargs=None*, *call=None*)

Return a list of all VMs and templates that are on the specified provider, with basic fields

CLI Example:

```
salt-cloud -f list_nodes my-vmware-config
```

To return a list of all VMs and templates present on ALL configured providers, with basic fields:

CLI Example:

```
salt-cloud -Q
```

salt.cloud.clouds.vmware.**list_nodes_full**(*kwargs=None*, *call=None*)

Return a list of all VMs and templates that are on the specified provider, with full details

CLI Example:

```
salt-cloud -f list_nodes_full my-vmware-config
```

To return a list of all VMs and templates present on ALL configured providers, with full details:

CLI Example:

```
salt-cloud -F
```

salt.cloud.clouds.vmware.**list_nodes_min**(*kwargs=None*, *call=None*)

Return a list of all VMs and templates that are on the specified provider, with no details

CLI Example:

```
salt-cloud -f list_nodes_min my-vmware-config
```

`salt.cloud.clouds.vmware.`**`list_nodes_select`**(*call=None*)

Return a list of all VMs and templates that are on the specified provider, with fields specified under `query.selection` in `/etc/salt/cloud`

CLI Example:

```
salt-cloud -f list_nodes_select my-vmware-config
```

To return a list of all VMs and templates present on ALL configured providers, with fields specified under `query.selection` in `/etc/salt/cloud`:

CLI Example:

```
salt-cloud -S
```

`salt.cloud.clouds.vmware.`**`list_portgroups`**(*kwargs=None*, *call=None*)

List all the distributed virtual portgroups for this VMware environment

CLI Example:

```
salt-cloud -f list_portgroups my-vmware-config
```

`salt.cloud.clouds.vmware.`**`list_resourcepools`**(*kwargs=None*, *call=None*)

List all the resource pools for this VMware environment

CLI Example:

```
salt-cloud -f list_resourcepools my-vmware-config
```

`salt.cloud.clouds.vmware.`**`list_snapshots`**(*kwargs=None*, *call=None*)

List snapshots either for all VMs and templates or for a specific VM/template in this VMware environment

To list snapshots for all VMs and templates:

CLI Example:

```
salt-cloud -f list_snapshots my-vmware-config
```

To list snapshots for a specific VM/template:

CLI Example:

```
salt-cloud -f list_snapshots my-vmware-config name="vmname"
```

`salt.cloud.clouds.vmware.`**`list_templates`**(*kwargs=None*, *call=None*)

List all the templates present in this VMware environment

CLI Example:

```
salt-cloud -f list_templates my-vmware-config
```

`salt.cloud.clouds.vmware.`**`list_vapps`**(*kwargs=None*, *call=None*)

List all the vApps for this VMware environment

CLI Example:

```
salt-cloud -f list_vapps my-vmware-config
```

`salt.cloud.clouds.vmware.`**`reboot_host`**`(`*kwargs=None, call=None*`)`

Reboot the specified host system in this VMware environment

Note: If the host system is not in maintenance mode, it will not be rebooted. If you want to reboot the host system regardless of whether it is in maintenance mode, set `force=True`. Default is `force=False`.

CLI Example:

```
salt-cloud -f reboot_host my-vmware-config host="myHostSystemName" [force=True]
```

`salt.cloud.clouds.vmware.`**`remove_all_snapshots`**`(`*name, kwargs=None, call=None*`)`

Remove all the snapshots present for the specified virtual machine.

Note: All the snapshots higher up in the hierarchy of the current snapshot tree are consolidated and their virtual disks are merged. To override this behavior and only remove all snapshots, set `merge_snapshots=False`. Default is `merge_snapshots=True`

CLI Example:

```
salt-cloud -a remove_all_snapshots vmname [merge_snapshots=False]
```

`salt.cloud.clouds.vmware.`**`remove_host`**`(`*kwargs=None, call=None*`)`

Remove the specified host system from this VMware environment

CLI Example:

```
salt-cloud -f remove_host my-vmware-config host="myHostSystemName"
```

`salt.cloud.clouds.vmware.`**`remove_snapshot`**`(`*name, kwargs=None, call=None*`)`

Remove a snapshot of the specified virtual machine in this VMware environment

CLI Example:

```
salt-cloud -a remove_snapshot vmname snapshot_name="mySnapshot"
salt-cloud -a remove_snapshot vmname snapshot_name="mySnapshot" [remove_children=
↪"True"]
```

`salt.cloud.clouds.vmware.`**`rescan_hba`**`(`*kwargs=None, call=None*`)`

To rescan a specified HBA or all the HBAs on the Host System

CLI Example:

```
salt-cloud -f rescan_hba my-vmware-config host="hostSystemName"
salt-cloud -f rescan_hba my-vmware-config hba="hbaDeviceName" host="hostSystemName
↪"
```

`salt.cloud.clouds.vmware.`**`reset`**`(`*name, soft=False, call=None*`)`

To reset a VM using its name

Note: If `soft=True` then issues a command to the guest operating system asking it to perform a reboot. Otherwise hypervisor will terminate VM and start it again. Default is soft=False

For `soft=True` vmtools should be installed on guest system.

CLI Example:

```
salt-cloud -a reset vmname
salt-cloud -a reset vmname soft=True
```

`salt.cloud.clouds.vmware.`**`revert_to_snapshot`**`(`*name, kwargs=None, call=None*`)`

Revert virtual machine to it's current snapshot. If no snapshot exists, the state of the virtual machine remains unchanged

---

Note: The virtual machine will be powered on if the power state of the snapshot when it was created was set to "Powered On". Set `power_off=True` so that the virtual machine stays powered off regardless of the power state of the snapshot when it was created. Default is `power_off=False`.

If the power state of the snapshot when it was created was "Powered On" and if `power_off=True`, the VM will be put in suspended state after it has been reverted to the snapshot.

CLI Example:

```
salt-cloud -a revert_to_snapshot vmame [power_off=True]
salt-cloud -a revert_to_snapshot vmame snapshot_name="selectedSnapshot" [power_
↪off=True]
```

`salt.cloud.clouds.vmware.`**`show_instance`**`(`*name, call=None*`)`

List all available details of the specified VM

CLI Example:

```
salt-cloud -a show_instance vmname
```

`salt.cloud.clouds.vmware.`**`start`**`(`*name, call=None*`)`

To start/power on a VM using its name

CLI Example:

```
salt-cloud -a start vmname
```

`salt.cloud.clouds.vmware.`**`stop`**`(`*name, soft=False, call=None*`)`

To stop/power off a VM using its name

Note: If `soft=True` then issues a command to the guest operating system asking it to perform a clean shutdown of all services. Default is soft=False

For `soft=True` vmtools should be installed on guest system.

CLI Example:

```
salt-cloud -a stop vmname
salt-cloud -a stop vmname soft=True
```

`salt.cloud.clouds.vmware.`**`suspend`**`(`*name, call=None*`)`

To suspend a VM using its name

CLI Example:

```
salt-cloud -a suspend vmname
```

`salt.cloud.clouds.vmware.`**`terminate`**`(`*name, call=None*`)`

To do an immediate power off of a VM using its name. A `SIGKILL` is issued to the vmx process of the VM

CLI Example:

```
salt-cloud -a terminate vmname
```

`salt.cloud.clouds.vmware.`**`test_vcenter_connection`**(*kwargs=None*, *call=None*)

Test if the connection can be made to the vCenter server using the specified credentials inside `/etc/salt/cloud.providers` or `/etc/salt/cloud.providers.d/vmware.conf`

CLI Example:

```
salt-cloud -f test_vcenter_connection my-vmware-config
```

`salt.cloud.clouds.vmware.`**`upgrade_tools`**(*name*, *reboot=False*, *call=None*)

To upgrade VMware Tools on a specified virtual machine.

Note: If the virtual machine is running Windows OS, use `reboot=True` to reboot the virtual machine after VMware tools upgrade. Default is `reboot=False`

CLI Example:

```
salt-cloud -a upgrade_tools vmname
salt-cloud -a upgrade_tools vmname reboot=True
```

`salt.cloud.clouds.vmware.`**`upgrade_tools_all`**(*call=None*)

To upgrade VMware Tools on all virtual machines present in the specified provider

Note: If the virtual machine is running Windows OS, this function will attempt to suppress the automatic reboot caused by a VMware Tools upgrade.

CLI Example:

```
salt-cloud -f upgrade_tools_all my-vmware-config
```

21.4.29 salt.cloud.clouds.vultrpy

Vultr Cloud Module using python-vultr bindings

New in version 2016.3.0.

The Vultr cloud module is used to control access to the Vultr VPS system.

Use of this module only requires the `api_key` parameter.

Set up the cloud configuration at `/etc/salt/cloud.providers` or `/etc/salt/cloud.providers.d/vultr.conf`:

```
my-vultr-config:
  # Vultr account api key
  api_key: <supersecretapi_key>
  driver: vultr
```

Set up the cloud profile at `/etc/salt/cloud.profiles` or `/etc/salt/cloud.profiles.d/vultr.conf`:

```
nyc-4gb-4cpu-ubuntu-14-04:
  location: 1
  provider: my-vultr-config
  image: 160
  size: 95
  enable_private_network: True
```

This driver also supports Vultr's *startup script* feature. You can list startup scripts in your account with

```
salt-cloud -f list_scripts <name of vultr provider>
```

That list will include the IDs of the scripts in your account. Thus, if you have a script called 'setup-networking' with an ID of 493234 you can specify that startup script in a profile like so:

```
nyc-2gb-1cpu-ubuntu-17-04:
  location: 1
  provider: my-vultr-config
  image: 223
  size: 13
  startup_script_id: 493234
```

`salt.cloud.clouds.vultrpy.`**`avail_images`**`(`*conn=None*`)`
Return available images

`salt.cloud.clouds.vultrpy.`**`avail_locations`**`(`*conn=None*`)`
return available datacenter locations

`salt.cloud.clouds.vultrpy.`**`avail_scripts`**`(`*conn=None*`)`
return available startup scripts

`salt.cloud.clouds.vultrpy.`**`avail_sizes`**`(`*conn=None*`)`
Return available sizes ("plans" in VultrSpeak)

`salt.cloud.clouds.vultrpy.`**`create`**`(`*vm_*`)`
Create a single VM from a data dict

`salt.cloud.clouds.vultrpy.`**`destroy`**`(`*name*`)`
Remove a node from Vultr

`salt.cloud.clouds.vultrpy.`**`get_configured_provider`**`()`
Return the first configured instance

`salt.cloud.clouds.vultrpy.`**`list_nodes`**`(`***kwargs*`)`
Return basic data on nodes

`salt.cloud.clouds.vultrpy.`**`list_nodes_full`**`(`***kwargs*`)`
Return all data on nodes

`salt.cloud.clouds.vultrpy.`**`list_nodes_select`**`(`*conn=None, call=None*`)`
Return a list of the VMs that are on the provider, with select fields

`salt.cloud.clouds.vultrpy.`**`list_scripts`**`(`*conn=None, call=None*`)`
return list of Startup Scripts

`salt.cloud.clouds.vultrpy.`**`show_instance`**`(`*name, call=None*`)`
Show the details from the provider concerning an instance

`salt.cloud.clouds.vultrpy.`**`start`**`(`**args, **kwargs*`)`
Execute a "start" action on a VM

`salt.cloud.clouds.vultrpy.`**`stop`**`(`**args, **kwargs*`)`

Execute a "stop" action on a VM

21.4.30 salt.cloud.clouds.xen

XenServer Cloud Driver

The XenServer driver is designed to work with a Citrix XenServer.

Requires XenServer SDK (can be downloaded from https://www.citrix.com/downloads/xenserver/product-software/)

Place a copy of the XenAPI.py in the Python site-packages folder.

depends XenAPI

Example provider configuration:

```
# /etc/salt/cloud.providers.d/myxen.conf
myxen:
  driver: xen
  url: http://10.0.0.120
  user: root
  password: p@ssw0rd
```

Example profile configuration:

```
# /etc/salt/cloud.profiles.d/myxen.conf
suse:
  provider: myxen
  user: root
  password: p@ssw0rd
  image: opensuseleap42_2-template
  storage_repo: 'Local storage'
  resource_pool: default_pool
  clone: True
  minion:
    master: 10.0.0.18
sles:
  provider: myxen
  user: root
  clone: False
  image: sles12sp2-template
  deploy: False
w2k12:
  provider: myxen
  image: w2k12svr-template
  clone: True
  userdata_file: /srv/salt/win/files/windows-firewall.ps1
  win_installer: /srv/salt/win/files/Salt-Minion-2016.11.3-AMD64-Setup.exe
  win_username: Administrator
  win_password: p@ssw0rd
  use_winrm: False
  ipv4_cidr: 10.0.0.215/24
  ipv4_gw: 10.0.0.1
```

`salt.cloud.clouds.xen.`**`avail_images`**`(`*call=None*`)`

Get a list of images from Xen

If called with the *–list-images* then it returns images with all details.

```
salt-cloud --list-images myxen
```

`salt.cloud.clouds.xen.`**`avail_locations`**`(`*session=None*, *call=None*`)`

Return available Xen locations (not implemented)

```
salt-cloud --list-locations myxen
```

`salt.cloud.clouds.xen.`**`avail_sizes`**`(`*session=None*, *call=None*`)`

Return a list of Xen template definitions

```
salt-cloud --list-sizes myxen
```

`salt.cloud.clouds.xen.`**`create`**`(`*vm_*`)`

Create a VM in Xen

The configuration for this function is read from the profile settings.

```
salt-cloud -p some_profile xenvm01
```

`salt.cloud.clouds.xen.`**`destroy`**`(`*name=None*, *call=None*`)`

Destroy Xen VM or template instance

```
salt-cloud -d xenvm01
```

`salt.cloud.clouds.xen.`**`destroy_template`**`(`*name=None*, *call=None*, *kwargs=None*`)`

Destroy Xen VM or template instance

```
salt-cloud -f destroy_template myxen name=testvm2
```

`salt.cloud.clouds.xen.`**`destroy_vm_vdis`**`(`*name=None*, *session=None*, *call=None*`)`

Get virtual block devices on VM

```
salt-cloud -a destroy_vm_vdis  xenvm01
```

`salt.cloud.clouds.xen.`**`get_configured_provider`**`()`

Return the first configured instance.

`salt.cloud.clouds.xen.`**`get_pv_args`**`(`*name*, *session=None*, *call=None*`)`

Get PV arguments for a VM

```
salt-cloud -a get_pv_args xenvm01
```

`salt.cloud.clouds.xen.`**`get_vm_ip`**`(`*name=None*, *session=None*, *call=None*`)`

Get the IP address of the VM

```
salt-cloud -a get_vm_ip xenvm01
```

Note: Requires xen guest tools to be installed in VM

`salt.cloud.clouds.xen.`**`host_list`**`(`*call=None*`)`

Get a list of Xen Servers

```
salt-cloud -f host_list myxen
```

`salt.cloud.clouds.xen.`**`list_nodes`**`()`

List virtual machines

```
salt-cloud -Q
```

`salt.cloud.clouds.xen.`**`list_nodes_full`**`(`*session=None*`)`

List full virtual machines

```
salt-cloud -F
```

`salt.cloud.clouds.xen.`**`list_nodes_select`**`(`*call=None*`)`

Perform a select query on Xen VM instances

```
salt-cloud -S
```

`salt.cloud.clouds.xen.`**`pause`**`(`*name, call=None, session=None*`)`

Pause a vm

```
salt-cloud -a pause xenvm01
```

`salt.cloud.clouds.xen.`**`pif_list`**`(`*call=None*`)`

Get a list of Resource Pools

```
salt-cloud -f pool_list myxen
```

`salt.cloud.clouds.xen.`**`pool_list`**`(`*call=None*`)`

Get a list of Resource Pools

```
salt-cloud -f pool_list myxen
```

`salt.cloud.clouds.xen.`**`reboot`**`(`*name, call=None, session=None*`)`

Reboot a vm

```
salt-cloud -a reboot xenvm01
```

`salt.cloud.clouds.xen.`**`resume`**`(`*name, call=None, session=None*`)`

Resume a vm from disk

```
salt-cloud -a resume xenvm01
```

`salt.cloud.clouds.xen.`**`set_pv_args`**`(`*name, kwargs=None, session=None, call=None*`)`

Set PV arguments for a VM

```
salt-cloud -a set_pv_args xenvm01 pv_args="utf-8 graphical"
```

`salt.cloud.clouds.xen.`**`set_vm_ip`**`(`*name=None, ipv4_cidr=None, ipv4_gw=None, session=None, call=None*`)`

Set the IP address on a virtual interface (vif)

`salt.cloud.clouds.xen.`**`show_instance`**`(`*name, session=None, call=None*`)`

Show information about a specific VM or template

```
salt-cloud -a show_instance xenvm01
```

Note: memory is memory_dynamic_max

`salt.cloud.clouds.xen.`**`shutdown`**(*name, call=None, session=None*)

Shutdown a vm

```
salt-cloud -a shutdown xenvm01
```

`salt.cloud.clouds.xen.`**`sr_list`**(*call=None*)

Geta list of storage repositories

```
salt-cloud -f sr_list myxen
```

`salt.cloud.clouds.xen.`**`start`**(*name, call=None, session=None*)

Start a vm

```
salt-cloud -a start xenvm01
```

`salt.cloud.clouds.xen.`**`stop`**(*name, call=None, session=None*)

Stop a vm

```
salt-cloud -a stop xenvm01
```

`salt.cloud.clouds.xen.`**`suspend`**(*name, call=None, session=None*)

Suspend a vm to disk

```
salt-cloud -a suspend xenvm01
```

`salt.cloud.clouds.xen.`**`template_list`**(*call=None*)

Return available Xen template information.

This returns the details of each template to show number cores, memory sizes, etc..

```
salt-cloud -f template_list myxen
```

`salt.cloud.clouds.xen.`**`unpause`**(*name, call=None, session=None*)

UnPause a vm

```
salt-cloud -a unpause xenvm01
```

`salt.cloud.clouds.xen.`**`vbd_list`**(*name=None, call=None*)

Get a list of VBDs on a VM

requires: the name of the vm with the vbd definition

```
salt-cloud -a vbd_list xenvm01
```

`salt.cloud.clouds.xen.`**`vdi_list`**(*call=None, kwargs=None*)

Return available Xen VDI images

If this function is called with the `-f` or `--function` then it can return a list with minimal deatil using the `terse=True` keyword argument.

```
salt-cloud -f vdi_list myxen terse=True
```

`salt.cloud.clouds.xen.`**`vif_list`**(*name, call=None, kwargs=None*)

Get a list of virtual network interfaces on a VM

requires: the name of the vm with the vbd definition

```
salt-cloud -a vif_list xenvm01
```

21.5 engine modules

docker_events	Send events from Docker events :Depends: Docker API >= 1.22
fluent	An engine that reads messages from the salt event bus and pushes them onto a fluent endpoint.
http_logstash	HTTP Logstash engine
ircbot	IRC Bot engine
junos_syslog	Junos Syslog Engine
logentries	An engine that sends events to the Logentries logging service.
logstash_engine	An engine that reads messages from the salt event bus and pushes them onto a logstash endpoint.
napalm_syslog	NAPALM syslog engine
reactor	Setup Reactor
redis_sentinel	An engine that reads messages from the redis sentinel pubsub and sends reactor events based on the channels they are subscribed to.
slack	An engine that reads messages from Slack and can act on them
sqs_events	An engine that continuously reads messages from SQS and fires them as events.
stalekey	An engine that uses presence detection to keep track of which minions have been recently connected and remove their keys if they have not been connected for a certain period of time.
test	A simple test engine, not intended for real use but as an example
thorium	Manage the Thorium complex event reaction system
webhook	Send events from webhook api

21.5.1 salt.engines.docker_events module

Send events from Docker events :Depends: Docker API >= 1.22

`salt.engines.docker_events.`**`start`**(*docker_url='unix://var/run/docker.sock', timeout=60, tag='salt/engines/docker_events', filters=None*)

Scan for Docker events and fire events

Example Config

```
engines:
  - docker_events:
      docker_url: unix://var/run/docker.sock
      filters:
        event:
        - start
        - stop
        - die
        - oom
```

The config above sets up engines to listen for events from the Docker daemon and publish them to the Salt event bus.

For filter reference, see https://docs.docker.com/engine/reference/commandline/events/

21.5.2 salt.engines.fluent

An engine that reads messages from the salt event bus and pushes them onto a fluent endpoint.

New in version 3000.

Configuration

All arguments are optional

Example configuration of default settings

```
engines:
  - fluent:
      host: localhost
      port: 24224
      app: engine
```

Example fluentd configuration

```
<source>
    @type forward
    port 24224
</source>

<match saltstack.**>
    @type file
    path /var/log/td-agent/saltstack
</match>
```

depends fluent-logger

`salt.engines.fluent.`**`start`**(*host='localhost'*, *port=24224*, *app='engine'*)

Listen to salt events and forward them to fluent

Parameters

- **host** (*str*) – Host running fluentd agent. Default is localhost
- **port** (*int*) – Port of fluentd agent. Default is 24224
- **app** (*str*) – Text sent as fluentd tag. Default is "engine". This text is appended to "saltstack." to form a fluentd tag, ex: "saltstack.engine"

21.5.3 salt.engines.http_logstash

HTTP Logstash engine

An engine that reads messages from the salt event bus and pushes them onto a logstash endpoint via HTTP requests.

Changed in version 2018.3.0.

Note: By default, this engine take everything from the Salt bus and exports into Logstash. For a better selection of the events that you want to publish, you can use the `tags` and `funs` options.

configuration Example configuration

```
engines:
  - http_logstash:
      url: http://blabla.com/salt-stuff
      tags:
          - salt/job/*/new
          - salt/job/*/ret/*
      funs:
          - probes.results
          - bgp.config
```

`salt.engines.http_logstash.`**`start`**(*url*, *funs=None*, *tags=None*)

Listen to salt events and forward them to logstash.

url The Logstash endpoint.

funs: `None` A list of functions to be compared against, looking into the `fun` field from the event data. This option helps to select the events generated by one or more functions. If an event does not have the `fun` field in the data section, it will be published. For a better selection, consider using the `tags` option. By default, this option accepts any event to be submitted to Logstash.

tags: `None` A list of pattern to compare the event tag against. By default, this option accepts any event to be submitted to Logstash.

21.5.4 salt.engines.ircbot

IRC Bot engine

New in version 2017.7.0.

Example Configuration

```
engines:
  - ircbot:
      nick: <nick>
      username: <username>
      password: <password>
      host: chat.freenode.net
      port: 7000
      channels:
        - salt-test
        - '##something'
      use_ssl: True
      use_sasl: True
      disable_query: True
      allow_hosts:
        - salt/engineer/.*
      allow_nicks:
        - gtmanfred
```

Available commands on irc are:

ping return pong

echo <stuff> return <stuff> targeted at the user who sent the commands

event <tag> [<extra>, <data>] fire event on the master or minion event stream with the tag *salt/engines/ircbot/<tag>* and a data object with a list of everything else sent in the message

Example of usage

```
08:33:57 @gtmanfred > !ping
08:33:57   gtmanbot > gtmanfred: pong
08:34:02 @gtmanfred > !echo ping
08:34:02   gtmanbot > ping
08:34:17 @gtmanfred > !event test/tag/ircbot irc is useful
08:34:17   gtmanbot > gtmanfred: TaDa!
```

```
[DEBUG   ] Sending event: tag = salt/engines/ircbot/test/tag/ircbot; data = {'_stamp
↪': '2016-11-28T14:34:16.633623', 'data': ['irc', 'is', 'useful']}
```

class salt.engines.ircbot.**Event**(*source*, *code*, *line*)

code
Alias for field number 1

line
Alias for field number 2

source
Alias for field number 0

class salt.engines.ircbot.**IRCClient**(*nick*, *host*, *port=6667*, *username=None*, *password=None*, *channels=None*, *use_ssl=False*, *use_sasl=False*, *char='!'*, *allow_hosts=False*, *allow_nicks=False*, *disable_query=True*)

join_channel(*channel*)

on_closed()

on_connect()

read_messages()

send_message(*line*)

class salt.engines.ircbot.**PrivEvent**(*source*, *nick*, *user*, *host*, *code*, *channel*, *command*, *line*)

channel
Alias for field number 5

code
Alias for field number 4

command
Alias for field number 6

host
Alias for field number 3

line
Alias for field number 7

nick
Alias for field number 1

source
Alias for field number 0

user
Alias for field number 2

`salt.engines.ircbot.`**`start`**`(`*nick*, *host*, *port=6667*, *username=None*, *password=None*, *channels=None*, *use_ssl=False*, *use_sasl=False*, *char='!'*, *allow_hosts=False*, *allow_nicks=False*, *disable_query=True*`)`

IRC Bot for interacting with salt.

nick Nickname of the connected Bot.

host irc server (example - chat.freenode.net).

port irc port. Default: 6667

password password for authenticating. If not provided, user will not authenticate on the irc server.

channels channels to join.

use_ssl connect to server using ssl. Default: False

use_sasl authenticate using sasl, instead of messaging NickServ. Default: False

> **Note:** This will allow the bot user to be fully authenticated before joining any channels

char command character to look for. Default: !

allow_hosts hostmasks allowed to use commands on the bot. Default: False True to allow all False to allow none List of regexes to allow matching

allow_nicks Nicks that are allowed to use commands on the bot. Default: False True to allow all False to allow none List of regexes to allow matching

disable_query Disable commands from being sent through private queries. Require they be sent to a channel, so that all communication can be controlled by access to the channel. Default: True

> **Warning:** Unauthenticated Access to event stream
>
> This engine sends events calls to the event stream without authenticating them in salt. Authentication will need to be configured and enforced on the irc server or enforced in the irc channel. The engine only accepts commands from channels, so non authenticated users could be banned or quieted in the channel.
>
> /mode +q $~a # quiet all users who are not authenticated /mode +r # do not allow unauthenticated users into the channel
>
> It would also be possible to add a password to the irc channel, or only allow invited users to join.

21.5.5 salt.engines.junos_syslog module

Junos Syslog Engine

New in version 2017.7.0.

depends pyparsing, twisted

An engine that listens to syslog message from Junos devices, extract event information and generate message on SaltStack bus.

The event topic sent to salt is dynamically generated according to the topic title specified by the user. The incoming event data (from the junos device) consists of the following fields:

1. hostname
2. hostip

3. daemon
4. event
5. severity
6. priority
7. timestamp
8. message
9. pid
10. raw (the raw event data forwarded from the device)

The topic title can consist of any of the combination of above fields, but the topic has to start with `jnpr/syslog`. Here are a couple example combinations:

- jnpr/syslog/hostip/daemon/event
- jnpr/syslog/daemon/severity

The corresponding dynamic topic sent on salt event bus would look something like:

- jnpr/syslog/1.1.1.1/mgd/UI_COMMIT_COMPLETED
- jnpr/syslog/sshd/7

The default topic title is `jnpr/syslog/hostname/event`.

One can choose the type of data they want from the event bus. For example, if one wants only events pertaining to a particular daemon, this can be specified in the configuration file:

```
daemon: mgd
```

One can even have a list of daemons:

```
daemon:
  - mgd
  - sshd
```

Example configuration (to be written in master config file)

```
engines:
  - junos_syslog:
      port: 9999
      topic: jnpr/syslog/hostip/daemon/event
      daemon:
        - mgd
        - sshd
```

For junos_syslog engine to receive events, syslog must be set on the junos device. This can be done via following configuration:

```
set system syslog host <ip-of-the-salt-device> port 516 any any
```

Below is a sample syslog event which is received from the junos device:

```
<30>May 29 05:18:12 bng-ui-vm-9 mspd[1492]: No chassis configuration found
```

The source for parsing the syslog messages is taken from: https://gist.github.com/leandrosilva/3651640#file-xlog-py `salt.engines.junos_syslog.`**`start`**(*port=516*, ***kwargs*)

21.5.6 salt.engines.logentries

An engine that sends events to the Logentries logging service.

maintainer Jimmy Tang (jimmy_tang@rapid7.com)

maturity New

depends ssl, certifi

platform all

To enable this engine the master and/or minion will need the following python libraries

ssl certifi

If you are running a new enough version of python then the ssl library will be present already.

You will also need the following values configured in the minion or master config.

configuration Example configuration

```
engines:
  - logentries:
      endpoint: data.logentries.com
      port: 10000
      token: 057af3e2-1c05-47c5-882a-5cd644655dbf
```

The 'token' can be obtained from the Logentries service.

To test this engine

```
salt '*' test.ping cmd.run uptime
```

class `salt.engines.logentries.`**`PlainTextSocketAppender`**(*verbose=True*, *LE_API='data.logentries.com'*, *LE_PORT=80*, *LE_TLS_PORT=443*)

`close_connection`()

`open_connection`()

`put`(*data*)

`reopen_connection`()

`salt.engines.logentries.`**`SocketAppender`**

alias of *`salt.engines.logentries.TLSSocketAppender`*

class `salt.engines.logentries.`**`TLSSocketAppender`**(*verbose=True*, *LE_API='data.logentries.com'*, *LE_PORT=80*, *LE_TLS_PORT=443*)

`open_connection`()

`salt.engines.logentries.`**`event_bus_context`**(*opts*)

`salt.engines.logentries.`**`start`**(*endpoint='data.logentries.com'*, *port=10000*, *token=None*, *tag='salt/engines/logentries'*)

Listen to salt events and forward them to Logentries

21.5.7 salt.engines.logstash_engine

An engine that reads messages from the salt event bus and pushes them onto a logstash endpoint.

configuration Example configuration

```
engines:
  - logstash:
      host: log.my_network.com
      port: 5959
      proto: tcp
```

depends logstash

`salt.engines.logstash_engine.`**`event_bus_context`**`(`*opts*`)`

`salt.engines.logstash_engine.`**`start`**`(`*host*, *port=5959*, *tag='salt/engine/logstash'*, *proto='udp'*`)`
Listen to salt events and forward them to logstash

21.5.8 salt.engines.napalm_syslog

NAPALM syslog engine

New in version 2017.7.0.

An engine that takes syslog messages structured in OpenConfig or IETF format and fires Salt events.

As there can be many messages pushed into the event bus, the user is able to filter based on the object structure.

Requirements

- napalm-logs

This engine transfers objects from the napalm-logs library into the event bus. The top dictionary has the following keys:

- `ip`
- `host`
- `timestamp`
- `os`: the network OS identified
- `model_name`: the OpenConfig or IETF model name
- `error`: the error name (consult the documentation)
- `message_details`: details extracted from the syslog message
- `open_config`: the OpenConfig model

The napalm-logs transfers the messages via widely used transport mechanisms such as: ZeroMQ (default), Kafka, etc.

The user can select the right transport using the `transport` option in the configuration.

configuration Example configuration

```
engines:
  - napalm_syslog:
      transport: zmq
      address: 1.2.3.4
      port: 49018
```

configuration Configuration example, excluding messages from IOS-XR devices:

```
engines:
  - napalm_syslog:
      transport: kafka
      address: 1.2.3.4
      port: 49018
      os_blacklist:
        - iosxr
```

Event example:

```
{
    "_stamp": "2017-05-26T10:03:18.653045",
    "error": "BGP_PREFIX_THRESH_EXCEEDED",
    "host": "vmx01",
    "ip": "192.168.140.252",
    "message_details": {
        "date": "May 25",
        "host": "vmx01",
        "message": "192.168.140.254 (External AS 65001): Configured maximum prefix-
↪limit threshold(22) exceeded for inet-unicast nlri: 28 (instance master)",
        "pri": "28",
        "processId": "2957",
        "processName": "rpd",
        "tag": "BGP_PREFIX_THRESH_EXCEEDED",
        "time": "20:50:41"
    },
    "model_name": "openconfig_bgp",
    "open_config": {
        "bgp": {
            "neighbors": {
                "neighbor": {
                    "192.168.140.254": {
                        "afi_safis": {
                            "afi_safi": {
                                "inet": {
                                    "afi_safi_name": "inet",
                                    "ipv4_unicast": {
                                        "prefix_limit": {
                                            "state": {
                                                "max_prefixes": 22
                                            }
                                        }
                                    },
                                    "state": {
                                        "prefixes": {
                                            "received": 28
                                        }
                                    }
                                }
```

(continues on next page)

(continued from previous page)

```
                            }
                        },
                        "neighbor_address": "192.168.140.254",
                        "state": {
                            "peer_as": 65001
                        }
                    }
                }
            }
        }
    },
    "os": "junos",
    "timestamp": "1495741841"
}
```

To consume the events and eventually react and deploy a configuration changes on the device(s) firing the event, one is able to identify the minion ID, using one of the following alternatives, but not limited to:

- `Host grains` to match the event tag
- `Host DNS grain` to match the IP address in the event data
- `Hostname grains` to match the event tag
- *Define static grains*
- *Write a grains module*
- *Targeting minions using pillar data* - The user can configure certain information in the Pillar data and then use it to identify minions

Master configuration example, to match the event and react:

```
reactor:
  - 'napalm/syslog/*/BGP_PREFIX_THRESH_EXCEEDED/*':
    - salt://increase_prefix_limit_on_thresh_exceeded.sls
```

Which matches the events having the error code `BGP_PREFIX_THRESH_EXCEEDED` from any network operating system, from any host and reacts, executing the `increase_prefix_limit_on_thresh_exceeded.sls` reactor, found under one of the `file_roots` paths.

Reactor example:

```
increase_prefix_limit_on_thresh_exceeded:
  local.net.load_template:
    - tgt: "hostname:{{ data['host'] }}"
    - tgt_type: grain
    - kwarg:
        template_name: salt://increase_prefix_limit.jinja
        openconfig_structure: {{ data['open_config'] }}
```

The reactor in the example increases the BGP prefix limit when triggered by an event as above. The minion is matched using the `host` field from the `data` (which is the body of the event), compared to the `hostname grain` field. When the event occurs, the reactor will execute the `net.load_template` function, sending as arguments the template `salt://increase_prefix_limit.jinja` defined by the user in their environment and the complete OpenConfig object under the variable name `openconfig_structure`. Inside the Jinja template, the user can process the object from `openconfig_structure` and define the bussiness logic as required.

`salt.engines.napalm_syslog.`**`start`**(*transport='zmq', address='0.0.0.0', port=49017, auth_address='0.0.0.0', auth_port=49018, disable_security=False, certificate=None, os_whitelist=None, os_blacklist=None, error_whitelist=None, error_blacklist=None, host_whitelist=None, host_blacklist=None*)

Listen to napalm-logs and publish events into the Salt event bus.

transport: **`zmq`** Choose the desired transport.

> **Note**: Currently `zmq` is the only valid option.

address: **`0.0.0.0`** The address of the publisher, as configured on napalm-logs.

port: **`49017`** The port of the publisher, as configured on napalm-logs.

auth_address: **`0.0.0.0`** The address used for authentication when security is not disabled.

auth_port: **`49018`** Port used for authentication.

disable_security: **`False`** Trust unencrypted messages. Strongly discouraged in production.

certificate: **`None`** Absolute path to the SSL certificate.

os_whitelist: **`None`** List of operating systems allowed. By default everything is allowed.

os_blacklist: **`None`** List of operating system to be ignored. Nothing ignored by default.

error_whitelist: **`None`** List of errors allowed.

error_blacklist: **`None`** List of errors ignored.

host_whitelist: **`None`** List of hosts or IPs to be allowed.

host_blacklist: **`None`** List of hosts of IPs to be ignored.

21.5.9 salt.engines.reactor module

Setup Reactor

Example Config in Master or Minion config

```yaml
engines:
  - reactor:
      refresh_interval: 60
      worker_threads: 10
      worker_hwm: 10000

reactor:
  - 'salt/cloud/*/destroyed':
    - /srv/reactor/destroy/*.sls
```

`salt.engines.reactor.`**`start`**(*refresh_interval=None, worker_threads=None, worker_hwm=None*)

21.5.10 salt.engines.redis_sentinel module

An engine that reads messages from the redis sentinel pubsub and sends reactor events based on the channels they are subscribed to.

configuration Example configuration

```
engines:
  - redis_sentinel:
      hosts:
        matching: 'board*'
        port: 26379
        interface: eth2
      channels:
        - '+switch-master'
        - '+odown'
        - '-odown'
```

depends redis

class `salt.engines.redis_sentinel.`**`Listener`**(*host=None, port=None, channels=None, tag=None*)

`run`()

`work`(*item*)

`salt.engines.redis_sentinel.`**`start`**(*hosts, channels, tag=None*)

21.5.11 salt.engines.slack module

An engine that reads messages from Slack and can act on them

depends slackclient Python module

Important: This engine requires a bot user. To create a bot user, first go to the **Custom Integrations** page in your Slack Workspace. Copy and paste the following URL, and replace `myworkspace` with the proper value for your workspace:

`https://myworkspace.slack.com/apps/manage/custom-integrations`

Next, click on the `Bots` integration and request installation. Once approved by an admin, you will be able to proceed with adding the bot user. Once the bot user has been added, you can configure it by adding an avatar, setting the display name, etc. You will also at this time have access to your API token, which will be needed to configure this engine.

Finally, add this bot user to a channel by switching to the channel and using `/invite @mybotuser`. Keep in mind that this engine will process messages from each channel in which the bot is a member, so it is recommended to narrowly define the commands which can be executed, and the Slack users which are allowed to run commands.

This engine has two boolean configuration parameters that toggle specific features (both default to `False`):

1. `control` - If set to `True`, then any message which starts with the trigger string (which defaults to `!` and can be overridden by setting the `trigger` option in the engine configuration) will be interpreted as a Salt CLI command and the engine will attempt to run it. The permissions defined in the various `groups` will determine if the Slack user is allowed to run the command. The `targets` and `default_target` options can be used to set targets for a given command, but the engine can also read the following two keyword arguments:

   - `target` - The target expression to use for the command
   - `tgt_type` - The match type, can be one of `glob`, `list`, `pcre`, `grain`, `grain_pcre`, `pillar`, `nodegroup`, `range`, `ipcidr`, or `compound`. The default value is `glob`.

   Here are a few examples:

```
!test.ping target=*
!state.apply foo target=os:CentOS tgt_type=grain
!pkg.version mypkg target=role:database tgt_type=pillar
```

2. `fire_all` - If set to `True`, all messages which are not prefixed with the trigger string will fired as events onto Salt's ref:*event bus <event-system>*. The tag for these veents will be prefixed with the string specified by the `tag` config option (default: `salt/engines/slack`).

The `groups_pillar_name` config option can be used to pull group configuration from the specified pillar key.

Note: In order to use `groups_pillar_name`, the engine must be running as a minion running on the master, so that the `Caller` client can be used to retrieve that minions pillar data, because the master process does not have pillar data.

Configuration Examples

Changed in version 2017.7.0: Access control group support added

This example uses a single group called `default`. In addition, other groups are being loaded from pillar data. The group names do not have any significance, it is the users and commands defined within them that are used to determine whether the Slack user has permission to run the desired command.

```
engines:
  - slack:
      token: 'xoxb-xxxxxxxxxx-xxxxxxxxxxxxxxxxxxxxxxxx'
      control: True
      fire_all: False
      groups_pillar_name: 'slack_engine:groups_pillar'
      groups:
        default:
          users:
            - '*'
          commands:
            - test.ping
            - cmd.run
            - list_jobs
            - list_commands
          aliases:
            list_jobs:
              cmd: jobs.list_jobs
            list_commands:
              cmd: 'pillar.get salt:engines:slack:valid_commands target=saltmaster
↪tgt_type=list'
          default_target:
            target: saltmaster
            tgt_type: glob
          targets:
            test.ping:
              target: '*'
              tgt_type: glob
            cmd.run:
              target: saltmaster
              tgt_type: list
```

This example shows multiple groups applying to different users, with all users having access to run test.ping. Keep in mind that when using `*`, the value must be quoted, or else PyYAML will fail to load the configuration.

```
engines:
  - slack:
      groups_pillar: slack_engine_pillar
      token: 'xoxb-xxxxxxxxxx-xxxxxxxxxxxxxxxxxxxxxxxx'
      control: True
      fire_all: True
      tag: salt/engines/slack
      groups_pillar_name: 'slack_engine:groups_pillar'
      groups:
        default:
          users:
            - '*'
          commands:
            - test.ping
          aliases:
            list_jobs:
              cmd: jobs.list_jobs
            list_commands:
              cmd: 'pillar.get salt:engines:slack:valid_commands target=saltmaster
↪tgt_type=list'
        gods:
          users:
            - garethgreenaway
          commands:
            - '*'
```

class salt.engines.slack.**SlackClient**(*token*)

can_user_run(*user, command, groups*)

Break out the permissions into the following:

Check whether a user is in any group, including whether a group has the '*' membership

Parameters

- **user** (*str*) – The username being checked against
- **command** (*str*) – The command that is being invoked (e.g. test.ping)
- **groups** (*dict*) – the dictionary with groups permissions structure.

Return type tuple

Returns

On a successful permitting match, returns 2-element tuple that contains the name of the group that successfully matched, and a dictionary containing the configuration of the group so it can be referenced.

On failure it returns an empty tuple

commandline_to_list(*cmdline_str, trigger_string*)

cmdline_str is the string of the command line trigger_string is the trigger string, to be removed

control_message_target(*slack_user_name, text, loaded_groups, trigger_string*)

Returns a tuple of (target, cmdline,) for the response

Raises IndexError if a user can't be looked up from all_slack_users

Returns (False, False) if the user doesn't have permission

These are returned together because the commandline and the targeting interact with the group config (specifically aliases and targeting configuration) so taking care of them together works out.

The cmdline that is returned is the actual list that should be processed by salt, and not the alias.

fire(*tag*, *msg*)
This replaces a function in main called 'fire'

It fires an event into the salt bus.

format_return_text(*data*, *function*, ***kwargs*)
Print out YAML using the block mode

generate_triggered_messages(*token*, *trigger_string*, *groups*, *groups_pillar_name*)
slack_token = string trigger_string = string input_valid_users = set input_valid_commands = set

When the trigger_string prefixes the message text, yields a dictionary of:

```
{
    'message_data': m_data,
    'cmdline': cmdline_list, # this is a list
    'channel': channel,
    'user': m_data['user'],
    'slack_client': sc
}
```

else yields {'message_data': m_data} and the caller can handle that

When encountering an error (e.g. invalid message), yields {}, the caller can proceed to the next message

When the websocket being read from has given up all its messages, yields {'done': True} to indicate that the caller has read all of the relevant data for now, and should continue its own processing and check back for more data later.

This relies on the caller sleeping between checks, otherwise this could flood

get_config_groups(*groups_conf*, *groups_pillar_name*)
get info from groups in config, and from the named pillar

todo: add specification for the minion to use to recover pillar

get_jobs_from_runner(*outstanding_jids*)
Given a list of job_ids, return a dictionary of those job_ids that have completed and their results.

Query the salt event bus via the jobs runner. jobs.list_job will show a job in progress, jobs.lookup_jid will return a job that has completed.

returns a dictionary of job id: result

get_slack_channels(*token*)
Get all channel names from Slack

get_slack_users(*token*)
Get all users from Slack

get_target(*permitted_group*, *cmdline*, *alias_cmdline*)
When we are permitted to run a command on a target, look to see what the default targeting is for that group, and for that specific command (if provided).

It's possible for None or False to be the result of either, which means that it's expected that the caller provide a specific target.

If no configured target is provided, the command line will be parsed for target=foo and tgt_type=bar

Test for this:

```
h = {'aliases': {}, 'commands': {'cmd.run', 'pillar.get'},
    'default_target': {'target': '*', 'tgt_type': 'glob'},
    'targets': {'pillar.get': {'target': 'you_momma', 'tgt_type': 'list'}},
    'users': {'dmangot', 'jmickle', 'pcn'}}
f = {'aliases': {}, 'commands': {'cmd.run', 'pillar.get'},
     'default_target': {}, 'targets': {},'users': {'dmangot', 'jmickle', 'pcn
↪'}}

g = {'aliases': {}, 'commands': {'cmd.run', 'pillar.get'},
     'default_target': {'target': '*', 'tgt_type': 'glob'},
     'targets': {}, 'users': {'dmangot', 'jmickle', 'pcn'}}
```

Run each of them through `get_configured_target(('foo', f), 'pillar.get')` and confirm a valid target

message_text(*m_data*)
Raises ValueError if a value doesn't work out, and TypeError if this isn't a message type

parse_args_and_kwargs(*cmdline*)
cmdline: list

returns tuple of: args (list), kwargs (dict)

run_command_async(*msg*)

Parameters

- **message_generator** (*generator of dict*) – Generates messages from slack that should be run
- **fire_all** (*bool*) – Whether to also fire messages to the event bus
- **tag** (*str*) – The tag to send to use to send to the event bus
- **interval** (*int*) – time to wait between ending a loop and beginning the next

run_commands_from_slack_async(*message_generator, fire_all, tag, control, interval=1*)
Pull any pending messages from the message_generator, sending each one to either the event bus, the command_async or both, depending on the values of fire_all and command

`salt.engines.slack.`**start**(*token, control=False, trigger='!', groups=None, groups_pillar_name=None, fire_all=False, tag='salt/engines/slack'*)
Listen to slack events and forward them to salt, new version

21.5.12 salt.engines.sqs_events

An engine that continuously reads messages from SQS and fires them as events.

Note that long polling is utilized to avoid excessive CPU usage.

New in version 2015.8.0.

depends boto

Configuration

This engine can be run on the master or on a minion.

Example Config:

```
sqs.keyid: GKTADJGHEIQSXMKKRBJ08H
sqs.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
sqs.message_format: json
```

Explicit sqs credentials are accepted but this engine can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More Information available at:

```
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-ec2.html
```

If IAM roles are not (or for `boto` version < 2.5.1) used you need to specify them either in a pillar or in the config file of the master or minion, as appropriate:

To deserialize the message from json:

```
sqs.message_format: json
```

It's also possible to specify key, keyid and region via a profile:

```
sqs.keyid: GKTADJGHEIQSXMKKRBJ08H
sqs.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

A region may also be specified in the configuration:

```
sqs.region: us-east-1
```

If a region is not specified, the default is us-east-1.

It's also possible to specify key, keyid and region via a profile:

```
myprofile:
  keyid: GKTADJGHEIQSXMKKRBJ08H
  key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
  region: us-east-1
```

Additionally you can define cross account sqs:

```
engines:
  - sqs_events:
      queue: prod
      owner_acct_id: 111111111111
```

`salt.engines.sqs_events.`**`start`**`(`*queue*, *profile=None*, *tag='salt/engine/sqs'*, *owner_acct_id=None*`)`

Listen to sqs and fire message on event bus

21.5.13 salt.engines.stalekey module

An engine that uses presence detection to keep track of which minions have been recently connected and remove their keys if they have not been connected for a certain period of time.

Requires that the *`minion_data_cache`* option be enabled.

configuration Example configuration:

```
engines:
  - stalekey:
```

(continues on next page)

(continued from previous page)

```
        interval: 3600
        expire: 86400
```

`salt.engines.stalekey.`**`start`**`(`*interval=3600, expire=604800*`)`

21.5.14 salt.engines.test

A simple test engine, not intended for real use but as an example

`salt.engines.test.`**`event_bus_context`**`(`*opts*`)`

`salt.engines.test.`**`start`**`()`
Listen to events and write them to a log file

21.5.15 salt.engines.thorium module

Manage the Thorium complex event reaction system

`salt.engines.thorium.`**`start`**`(`*grains=False, grain_keys=None, pillar=False, pillar_keys=None*`)`
Execute the Thorium runtime

21.5.16 salt.engines.webhook

Send events from webhook api

`salt.engines.webhook.`**`start`**`(`*address=None, port=5000, ssl_crt=None, ssl_key=None*`)`
Api to listen for webhooks to send to the reactor.

Implement the webhook behavior in an engine. `rest_cherrypy Webhook docs`

Unlike the rest_cherrypy Webhook, this is only an unauthenticated webhook endpoint. If an authenticated webhook endpoint is needed, use the salt-api webhook which runs on the master and authenticates through eauth.

Warning: Unauthenticated endpoint

This engine sends webhook calls to the event stream. If the engine is running on a minion with *file_client: local* the event is sent to the minion event stream. Otherwise it is sent to the master event stream.

Example Config

```
engines:
  - webhook: {}
```

```
engines:
  - webhook:
      port: 8000
      address: 10.128.1.145
      ssl_crt: /etc/pki/tls/certs/localhost.crt
      ssl_key: /etc/pki/tls/certs/localhost.key
```

21.6 executors modules

direct_call	Direct call executor module
docker	Docker executor module
splay	Splay function calls across targeted minions
sudo	Sudo executor module

21.6.1 salt.executors.direct_call module

Direct call executor module

`salt.executors.direct_call.`**`execute`**`(`*opts, data, func, args, kwargs*`)`
Directly calls the given function with arguments

21.6.2 salt.executors.docker module

Docker executor module

Used with the docker proxy minion.

`salt.executors.docker.`**`allow_missing_func`**`(`*function*`)`
Allow all calls to be passed through to docker container.

The docker call will use direct_call, which will return back if the module was unable to be run.

`salt.executors.docker.`**`execute`**`(`*opts, data, func, args, kwargs*`)`
Directly calls the given function with arguments

21.6.3 salt.executors.splay module

Splay function calls across targeted minions

`salt.executors.splay.`**`execute`**`(`*opts, data, func, args, kwargs*`)`
Splay a salt function call execution time across minions over a number of seconds (default: 300)

Note: You *probably* want to use –async here and look up the job results later. If you're dead set on getting the output from the CLI command, then make sure to set the timeout (with the -t flag) to something greater than the splaytime (max splaytime + time to execute job). Otherwise, it's very likely that the cli will time out before the job returns.

CLI Example:

```
# With default splaytime
salt --async --module-executors='[splay, direct_call]' '*' pkg.install cowsay
↪version=3.03-8.el6
```

```
# With specified splaytime (5 minutes) and timeout with 10 second buffer
salt -t 310 --module-executors='[splay, direct_call]' --executor-opts='
↪{splaytime: 300}' '*' pkg.version cowsay
```

21.6.4 salt.executors.sudo module

Sudo executor module

`salt.executors.sudo.`**`execute`**`(`*opts*, *data*, *func*, *args*, *kwargs*`)`

Allow for the calling of execution modules via sudo.

This module is invoked by the minion if the `sudo_user` minion config is present.

Example minion config:

```
sudo_user: saltdev
```

Once this setting is made, any execution module call done by the minion will be run under `sudo -u <sudo_user> salt-call`. For example, with the above minion config,

```
salt sudo_minion cmd.run 'cat /etc/sudoers'
```

is equivalent to

```
sudo -u saltdev salt-call cmd.run 'cat /etc/sudoers'
```

being run on `sudo_minion`.

21.7 fileserver modules

azurefs	The backend for serving files from the Azure blob storage service.
gitfs	Git Fileserver Backend
hgfs	Mercurial Fileserver Backend
minionfs	Fileserver backend which serves files pushed to the Master
roots	The default file server backend
s3fs	Amazon S3 Fileserver Backend
svnfs	Subversion Fileserver Backend

21.7.1 salt.fileserver.azurefs

The backend for serving files from the Azure blob storage service.

New in version 2015.8.0.

To enable, add `azurefs` to the *`fileserver_backend`* option in the Master config file.

```
fileserver_backend:
  - azurefs
```

Starting in Salt 2018.3.0, this fileserver requires the standalone Azure Storage SDK for Python. Theoretically any version >= v0.20.0 should work, but it was developed against the v0.33.0 version.

Each storage container will be mapped to an environment. By default, containers will be mapped to the `base` environment. You can override this behavior with the `saltenv` configuration option. You can have an unlimited number of storage containers, and can have a storage container serve multiple environments, or have multiple storage containers mapped to the same environment. Normal first-found rules apply, and storage containers are searched in the order they are defined.

You must have either an account_key or a sas_token defined for each container, if it is private. If you use a sas_token, it must have READ and LIST permissions.

```
azurefs:
  - account_name: my_storage
    account_key:
↪'fNH9cRp0+qVIVYZ+5rnZAhHc9ycOUcJnHtzpfOr0W0sxrtL2KVLuMe1xDfLwmfed+JJInZaEdWVCPHD4d/
↪oqeA=='
    container_name: my_container
  - account_name: my_storage
    sas_token: 'ss=b&sp=&sv=2015-07-08&sig=cohxXabx8FQdXsSEHyUXMjsSfNH2tZ2OB97Ou44pkRE
↪%3D&srt=co&se=2017-04-18T21%3A38%3A01Z'
    container_name: my_dev_container
    saltenv: dev
  - account_name: my_storage
    container_name: my_public_container
```

Note: Do not include the leading ? for sas_token if generated from the web

21.7.2 salt.fileserver.gitfs

Git Fileserver Backend

With this backend, branches and tags in a remote git repository are exposed to salt as different environments.

To enable, add `gitfs` to the *`fileserver_backend`* option in the Master config file.

```
fileserver_backend:
  - gitfs
```

Note: `git` also works here. Prior to the 2018.3.0 release, *only* `git` would work.

The Git fileserver backend supports both pygit2 and GitPython, to provide the Python interface to git. If both are present, the order of preference for which one will be chosen is the same as the order in which they were listed: pygit2, then GitPython.

An optional master config parameter (*`gitfs_provider`*) can be used to specify which provider should be used, in the event that compatible versions of both pygit2 and GitPython are installed.

More detailed information on how to use GitFS can be found in the *GitFS Walkthrough*.

Note: Minimum requirements

To use pygit2 for GitFS requires a minimum pygit2 version of 0.20.3. pygit2 0.20.3 requires libgit2 0.20.0. pygit2 and libgit2 are developed alongside one another, so it is recommended to keep them both at the same major release to avoid unexpected behavior. For example, pygit2 0.21.x requires libgit2 0.21.x, pygit2 0.22.x will require libgit2 0.22.x, etc.

To use GitPython for GitFS requires a minimum GitPython version of 0.3.0, as well as the git CLI utility. Instructions for installing GitPython can be found *here*.

To clear stale refs the git CLI utility must also be installed.

21.7.3 salt.fileserver.hgfs

Mercurial Fileserver Backend

To enable, add `hgfs` to the *`fileserver_backend`* option in the Master config file.

```
fileserver_backend:
  - hgfs
```

Note: `hg` also works here. Prior to the 2018.3.0 release, *only* `hg` would work.

After enabling this backend, branches, bookmarks, and tags in a remote mercurial repository are exposed to salt as different environments. This feature is managed by the *`fileserver_backend`* option in the salt master config file.

This fileserver has an additional option *`hgfs_branch_method`* that will set the desired branch method. Possible values are: `branches`, `bookmarks`, or `mixed`. If using `branches` or `mixed`, the `default` branch will be mapped to `base`.

Changed in version 2014.1.0: The *`hgfs_base`* master config parameter was added, allowing for a branch other than `default` to be used for the `base` environment, and allowing for a `base` environment to be specified when using an *`hgfs_branch_method`* of `bookmarks`.

depends

- mercurial
- python bindings for mercurial (`python-hglib`)

21.7.4 salt.fileserver.minionfs

Fileserver backend which serves files pushed to the Master

The *`cp.push`* function allows Minions to push files up to the Master. Using this backend, these pushed files are exposed to other Minions via the Salt fileserver.

To enable minionfs, *`file_recv`* needs to be set to `True` in the master config file (otherwise *`cp.push`* will not be allowed to push files to the Master), and `minionfs` must be added to the `fileserver_backends` list.

```
fileserver_backend:
  - minionfs
```

Note: `minion` also works here. Prior to the 2018.3.0 release, *only* `minion` would work.

Other minionfs settings include: *`minionfs_whitelist`*, *`minionfs_blacklist`*, *`minionfs_mountpoint`*, and *`minionfs_env`*.

See also:

MinionFS Backend Walkthrough

21.7.5 salt.fileserver.roots

The default file server backend

This fileserver backend serves files from the Master's local filesystem. If *fileserver_backend* is not defined in the Master config file, then this backend is enabled by default. If it *is* defined then `roots` must be in the *fileserver_backend* list to enable this backend.

```
fileserver_backend:
  - roots
```

Fileserver environments are defined using the *file_roots* configuration option.

21.7.6 salt.fileserver.s3fs

Amazon S3 Fileserver Backend

New in version 0.16.0.

This backend exposes directories in S3 buckets as Salt environments. To enable this backend, add `s3fs` to the *fileserver_backend* option in the Master config file.

```
fileserver_backend:
  - s3fs
```

S3 credentials must also be set in the master config file:

```
s3.keyid: GKTADJGHEIQSXMKKRBJ08H
s3.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

Alternatively, if on EC2 these credentials can be automatically loaded from instance metadata.

This fileserver supports two modes of operation for the buckets:

1. **A single bucket per environment**

   ```
   s3.buckets:
     production:
       - bucket1
       - bucket2
     staging:
       - bucket3
       - bucket4
   ```

2. **Multiple environments per bucket**

   ```
   s3.buckets:
     - bucket1
     - bucket2
     - bucket3
     - bucket4
   ```

Note that bucket names must be all lowercase both in the AWS console and in Salt, otherwise you may encounter `SignatureDoesNotMatch` errors.

A multiple-environment bucket must adhere to the following root directory structure:

```
s3://<bucket name>/<environment>/<files>
```

Note: This fileserver back-end requires the use of the MD5 hashing algorithm. MD5 may not be compliant with all security policies.

Note: This fileserver back-end is only compatible with MD5 ETag hashes in the S3 metadata. This means that you must use SSE-S3 or plaintext for bucket encryption, and that you must not use multipart upload when uploading to your bucket. More information here: https://docs.aws.amazon.com/AmazonS3/latest/API/RESTCommonResponseHeaders.html

Objects without an MD5 ETag will be fetched on every fileserver update.

If you deal with objects greater than 8MB, then you should use the following AWS CLI config to avoid mutipart upload:

```
s3 =
  multipart_threshold = 1024MB
```

More info here: https://docs.aws.amazon.com/cli/latest/topic/s3-config.html

---

21.7.7 salt.fileserver.svnfs

Subversion Fileserver Backend

After enabling this backend, branches and tags in a remote subversion repository are exposed to salt as different environments. To enable this backend, add `svnfs` to the *`fileserver_backend`* option in the Master config file.

```
fileserver_backend:
  - svnfs
```

Note: `svn` also works here. Prior to the 2018.3.0 release, *only* `svn` would work.

This backend assumes a standard svn layout with directories for `branches`, `tags`, and `trunk`, at the repository root.

depends

- subversion
- pysvn

Changed in version 2014.7.0: The paths to the trunk, branches, and tags have been made configurable, via the config options *`svnfs_trunk`*, *`svnfs_branches`*, and *`svnfs_tags`*. *`svnfs_mountpoint`* was also added. Finally, support for per-remote configuration parameters was added. See the *`documentation`* for more information.

21.8 grains modules

`cimc`	Generate baseline proxy minion grains for cimc hosts.
`chronos`	Generate chronos proxy minion grains.
`core`	The static grains, these are the core, or built in grains.
`disks`	Detect disks
`esxi`	Generate baseline proxy minion grains for ESXi hosts.
`extra`	
`fx2`	Generate baseline proxy minion grains for Dell FX2 chassis.

Continued on next page

Table 8 – continued from previous page

junos	Grains for junos.
marathon	Generate marathon proxy minion grains.
mdadm	Detect MDADM RAIDs
metadata	Grains from cloud metadata servers at 169.254.169.254
napalm	NAPALM Grains
opts	Simple grain to merge the opts into the grains directly if the grain_opts configuration value is set
panos	Generate baseline proxy minion grains for panos hosts.
philips_hue	Static grains for the Philips HUE lamps
rest_sample	Generate baseline proxy minion grains

21.8.1 salt.grains.cimc

Generate baseline proxy minion grains for cimc hosts.

`salt.grains.cimc.`**`cimc`**`(`*proxy=None*`)`

21.8.2 salt.grains.chronos

Generate chronos proxy minion grains.

New in version 2015.8.2.

`salt.grains.chronos.`**`kernel`**`()`

`salt.grains.chronos.`**`os`**`()`

`salt.grains.chronos.`**`os_data`**`()`

`salt.grains.chronos.`**`os_family`**`()`

21.8.3 salt.grains.core

The static grains, these are the core, or built in grains.

When grains are loaded they are not loaded in the same way that modules are loaded, grain functions are detected and executed, the functions MUST return a dict which will be applied to the main grains dict. This module will always be executed first, so that any grains loaded here in the core module can be overwritten just by returning dict keys with the same value as those returned here

`salt.grains.core.`**`append_domain`**`()`
Return append_domain if set

`salt.grains.core.`**`cwd`**`()`
Current working directory

`salt.grains.core.`**`default_gateway`**`()`
Populates grains which describe whether a server has a default gateway configured or not. Uses *ip -4 route show* and *ip -6 route show* and greps for a *default* at the beginning of any line. Assuming the standard *default via <ip>* format for default gateways, it will also parse out the ip address of the default gateway, and put it in ip4_gw or ip6_gw.

If the *ip* command is unavailable, no grains will be populated.

Currently does not support multiple default gateways. The grains will be set to the first default gateway found.

List of grains:

ip4_gw: True # ip/True/False if default ipv4 gateway ip6_gw: True # ip/True/False if default ipv6 gateway ip_gw: True # True if either of the above is True, False otherwise

`salt.grains.core.`**`dns`**`()`
Parse the resolver configuration file

New in version 2016.3.0.

`salt.grains.core.`**`fqdns`**`()`
Return all known FQDNs for the system by enumerating all interfaces and then trying to reverse resolve them (excluding 'lo' interface).

`salt.grains.core.`**`get_machine_id`**`()`
Provide the machine-id for machine/virtualization combination

`salt.grains.core.`**`get_master`**`()`
Provides the minion with the name of its master. This is useful in states to target other services running on the master.

`salt.grains.core.`**`get_server_id`**`()`
Provides an integer based on the FQDN of a machine. Useful as server-id in MySQL replication or anywhere else you'll need an ID like this.

`salt.grains.core.`**`hostname`**`()`
Return fqdn, hostname, domainname

Note: On Windows the `domain` grain may refer to the dns entry for the host instead of the Windows domain to which the host is joined. It may also be empty if not a part of any domain. Refer to the `windowsdomain` grain instead

`salt.grains.core.`**`hwaddr_interfaces`**`()`
Provide a dict of the connected interfaces and their hw addresses (Mac Address)

`salt.grains.core.`**`id_`**`()`
Return the id

`salt.grains.core.`**`ip4_interfaces`**`()`
Provide a dict of the connected interfaces and their ip4 addresses The addresses will be passed as a list for each interface

`salt.grains.core.`**`ip6_interfaces`**`()`
Provide a dict of the connected interfaces and their ip6 addresses The addresses will be passed as a list for each interface

`salt.grains.core.`**`ip_fqdn`**`()`
Return ip address and FQDN grains

`salt.grains.core.`**`ip_interfaces`**`()`
Provide a dict of the connected interfaces and their ip addresses The addresses will be passed as a list for each interface

`salt.grains.core.`**`linux_distribution`**`(`*`**kwargs`*`)`

`salt.grains.core.`**`locale_info`**`()`

Provides defaultlanguage defaultencoding

`salt.grains.core.`**`os_data`**`()`
Return grains pertaining to the operating system

`salt.grains.core.`**`path`**`()`
Return the path

`salt.grains.core.`**`pythonexecutable`**`()`
Return the python executable in use

`salt.grains.core.`**`pythonpath`**`()`
Return the Python path

`salt.grains.core.`**`pythonversion`**`()`
Return the Python version

`salt.grains.core.`**`saltpath`**`()`
Return the path of the salt module

`salt.grains.core.`**`saltversion`**`()`
Return the version of salt

`salt.grains.core.`**`saltversioninfo`**`()`
Return the version_info of salt

New in version 0.17.0.

`salt.grains.core.`**`zmqversion`**`()`
Return the zeromq version

21.8.4 salt.grains.disks

Detect disks

`salt.grains.disks.`**`disks`**`()`
Return list of disk devices

21.8.5 salt.grains.esxi

Generate baseline proxy minion grains for ESXi hosts.

New in version 2015.8.4.

`salt.grains.esxi.`**`esxi`**`()`

`salt.grains.esxi.`**`kernel`**`()`

`salt.grains.esxi.`**`os`**`()`

`salt.grains.esxi.`**`os_family`**`()`

21.8.6 salt.grains.extra

`salt.grains.extra.`**`config`**`()`
Return the grains set in the grains file

`salt.grains.extra.`**`shell`**`()`
Return the default shell to use on this system

21.8.7 salt.grains.fx2

Generate baseline proxy minion grains for Dell FX2 chassis. The challenge is that most of Salt isn't bootstrapped yet, so we need to repeat a bunch of things that would normally happen in proxy/fx2.py–just enough to get data from the chassis to include in grains.

`salt.grains.fx2.`**`fx2`**`()`

`salt.grains.fx2.`**`kernel`**`()`

`salt.grains.fx2.`**`location`**`()`

`salt.grains.fx2.`**`os_data`**`()`

`salt.grains.fx2.`**`os_family`**`()`

21.8.8 salt.grains.junos

Grains for junos. NOTE this is a little complicated–junos can only be accessed via salt-proxy-minion.Thus, some grains make sense to get them from the minion (PYTHONPATH), but others don't (ip_interfaces)

`salt.grains.junos.`**`defaults`**`()`

`salt.grains.junos.`**`facts`**`(`*proxy=None*`)`

`salt.grains.junos.`**`os_family`**`()`

21.8.9 salt.grains.marathon

Generate marathon proxy minion grains.

New in version 2015.8.2.

`salt.grains.marathon.`**`kernel`**`()`

`salt.grains.marathon.`**`marathon`**`()`

`salt.grains.marathon.`**`os`**`()`

`salt.grains.marathon.`**`os_data`**`()`

`salt.grains.marathon.`**`os_family`**`()`

21.8.10 salt.grains.mdadm

Detect MDADM RAIDs

`salt.grains.mdadm.`**`mdadm`**`()`
Return list of mdadm devices

21.8.11 salt.grains.metadata

Grains from cloud metadata servers at 169.254.169.254

New in version 2017.7.0.

depends requests

To enable these grains that pull from the http://169.254.169.254/latest metadata server set *metadata_server_grains: True*.

```
metadata_server_grains: True
```

`salt.grains.metadata.`**`metadata`**`()`

21.8.12 salt.grains.napalm

NAPALM Grains

codeauthor Mircea Ulinic <ping@mirceaulinic.net>

maturity new

depends napalm

platform unix

Dependencies

- `NAPALM proxy module`

New in version 2016.11.0.

`salt.grains.napalm.`**`getos`**`(`*proxy=None*`)`

Returns the Operating System name running on the network device.

Example: junos, iosxr, eos, ios etc.

CLI Example - select all network devices running JunOS:

```
salt -G 'os:junos' test.ping
```

`salt.grains.napalm.`**`host`**`(`*proxy=None*`)`

This grain is set by the NAPALM grain module only when running in a proxy minion. When Salt is installed directly on the network device, thus running a regular minion, the `host` grain provides the physical hostname of the network device, as it would be on an ordinary minion server. When running in a proxy minion, `host` points to the value configured in the pillar: *`NAPALM proxy module`*.

Note: The diference between `host` and `hostname` is that `host` provides the physical location - either domain name or IP address, while `hostname` provides the hostname as configured on the device. They are not necessarily the same.

New in version 2017.7.0.

CLI Example:

```
salt 'device*' grains.get host
```

Output:

```
device1:
    ip-172-31-13-136.us-east-2.compute.internal
device2:
```

(continues on next page)

(continued from previous page)

```
    ip-172-31-11-193.us-east-2.compute.internal
device3:
    ip-172-31-2-181.us-east-2.compute.internal
```

`salt.grains.napalm.`**`host_dns`**(*proxy=None*)

Return the DNS information of the host. This grain is a dictionary having two keys:

- A
- AAAA

Note: This grain is disabled by default, as the proxy startup may be slower when the lookup fails. The user can enable it using the `napalm_host_dns_grain` option (in the pillar or proxy configuration file):

```
napalm_host_dns_grain: true
```

New in version 2017.7.0.

CLI Example:

```
salt 'device*' grains.get host_dns
```

Output:

```
device1:
    A:
        - 172.31.9.153
    AAAA:
        - fd52:188c:c068::1
device2:
    A:
        - 172.31.46.249
    AAAA:
        - fdca:3b17:31ab::17
device3:
    A:
        - 172.31.8.167
    AAAA:
        - fd0f:9fd6:5fab::1
```

`salt.grains.napalm.`**`hostname`**(*proxy=None*)

Return the hostname as configured on the network device.

CLI Example:

```
salt 'device*' grains.get hostname
```

Output:

```
device1:
    edge01.yyz01
device2:
    edge01.bjm01
device3:
    edge01.flw01
```

`salt.grains.napalm.`**`interfaces`**`(`*proxy=None*`)`

Returns the complete interfaces list of the network device.

Example: ['lc-0/0/0', 'pfe-0/0/0', 'xe-1/3/0', 'lo0', 'irb', 'demux0', 'fxp0']

CLI Example - select all devices that have a certain interface, e.g.: xe-1/1/1:

```
salt -G 'interfaces:xe-1/1/1' test.ping
```

Output:

```
edge01.yyz01:
    True
edge01.maa01:
    True
edge01.syd01:
    True
edge01.del01:
    True
edge01.dus01:
    True
edge01.kix01:
    True
```

`salt.grains.napalm.`**`model`**`(`*proxy=None*`)`

Returns the network device chassis model.

Example: MX480, ASR-9904-AC etc.

CLI Example - select all Juniper MX480 routers and execute traceroute to 8.8.8.8:

```
salt -G 'model:MX480' net.traceroute 8.8.8.8
```

`salt.grains.napalm.`**`optional_args`**`(`*proxy=None*`)`

Return the connection optional args.

Note: Sensible data will not be returned.

New in version 2017.7.0.

CLI Example - select all devices connecting via port 1234:

```
salt -G 'optional_args:port:1234' test.ping
```

Output:

```
device1:
    True
device2:
    True
```

`salt.grains.napalm.`**`serial`**`(`*proxy=None*`)`

Returns the chassis serial number.

Example: FOX1234W00F

CLI Example - select all devices whose serial number begins with *FOX* and display the serial number value:

```
salt -G 'serial:FOX*' grains.get serial
```

Output:

```
edge01.icn01:
    FOXW00F001
edge01.del01:
    FOXW00F002
edge01.yyz01:
    FOXW00F003
edge01.mrs01:
    FOXW00F004
```

`salt.grains.napalm.`**`uptime`**(*proxy=None*)

Returns the uptime in seconds.

CLI Example - select all devices started/restarted within the last hour:

```
salt -G 'uptime<3600' test.ping
```

`salt.grains.napalm.`**`username`**(*proxy=None*)

Return the username.

New in version 2017.7.0.

CLI Example - select all devices using *foobar* as username for connection:

```
salt -G 'username:foobar' test.ping
```

Output:

`salt.grains.napalm.`**`vendor`**(*proxy=None*)

Returns the network device vendor.

Example: juniper, cisco, arista etc.

CLI Example - select all devices produced by Cisco and shutdown:

```
salt -G 'vendor:cisco' net.cli "shut"
```

`salt.grains.napalm.`**`version`**(*proxy=None*)

Returns the OS version.

Example: 13.3R6.5, 6.0.2 etc.

CLI Example - select all network devices running JunOS 13.3R6.5 and return the model:

```
salt -G 'os:junos and version:13.3R6.5' grains.get model
```

Output:

```
edge01.bjm01:
    MX2000
edge01.sjc01:
    MX960
edge01.mrs01:
    MX480
edge01.muc01:
    MX240
```

21.8.13 salt.grains.opts

Simple grain to merge the opts into the grains directly if the grain_opts configuration value is set

`salt.grains.opts.`**`opts`**`()`

Return the minion configuration settings

21.8.14 salt.grains.panos

Generate baseline proxy minion grains for panos hosts.

`salt.grains.panos.`**`panos`**`(`*proxy=None*`)`

21.8.15 salt.grains.philips_hue

Static grains for the Philips HUE lamps

New in version 2015.8.3.

`salt.grains.philips_hue.`**`kernel`**`()`

`salt.grains.philips_hue.`**`os`**`()`

`salt.grains.philips_hue.`**`os_family`**`()`

`salt.grains.philips_hue.`**`product`**`()`

`salt.grains.philips_hue.`**`vendor`**`()`

21.8.16 salt.grains.rest_sample

Generate baseline proxy minion grains

`salt.grains.rest_sample.`**`kernel`**`()`

`salt.grains.rest_sample.`**`location`**`()`

`salt.grains.rest_sample.`**`os`**`()`

`salt.grains.rest_sample.`**`os_data`**`()`

`salt.grains.rest_sample.`**`os_family`**`()`

`salt.grains.rest_sample.`**`proxy_functions`**`(`*proxy*`)`

The loader will execute functions with one argument and pass a reference to the proxymodules LazyLoader object. However, grains sometimes get called before the LazyLoader object is setup so *proxy* might be None.

21.9 execution modules

Virtual modules

21.9.1 salt.modules.group

`group` is a virtual module that is fulfilled by one of the following modules:

Execution Module	Used for
groupadd	Linux, NetBSD, and OpenBSD systems using `groupadd(8)`, `groupdel(8)`, and `groupmod(8)`
pw_group	FreeBSD-based OSes using `pw(8)`
solaris_group	Solaris-based OSes using `groupadd(1M)`, `groupdel(1M)`, and `groupmod(1M)`
win_groupadd	Windows

21.9.2 salt.modules.kernelpkg

`kernelpkg` is a virtual module that is fulfilled by one of the following modules:

Execution Module	Used for
kernelpkg_linux_apt	Debian/Ubuntu-based distros which use `apt-get` for package management
kernelpkg_linux_yum	RedHat-based distros and derivatives using `yum` or `dnf`

21.9.3 salt.modules.pkg

`pkg` is a virtual module that is fulfilled by one of the following modules:

Execution Module	Used for
aixpkg	AIX OS using `installp` and `rpm`
aptpkg	Debian/Ubuntu-based distros which use `apt-get(8)` for package management
mac_brew_pkg	Mac OS software management using Homebrew
ebuildpkg	Gentoo-based systems (utilizes the `portage` python module as well as `emerge(1)`)
freebsdpkg	FreeBSD-based OSes using `pkg_add(1)`
openbsdpkg	OpenBSD-based OSes using `pkg_add(1)`
pacmanpkg	Arch Linux-based distros using `pacman(8)`
pkgin	NetBSD-based OSes using `pkgin(1)`
pkgng	FreeBSD-based OSes using `pkg(8)`
pkgutil	Solaris-based OSes using OpenCSW's `pkgutil(1)`
solarispkg	Solaris-based OSes using `pkgadd(1M)`
solarisipspkg	Solaris-based OSes using IPS `pkg(1)`
win_pkg	Salt's *Windows Package Manager*
yumpkg	RedHat-based distros and derivatives using `yum(8)` or `dnf(8)`
zypperpkg	SUSE-based distros using `zypper(8)`

21.9.4 salt.modules.service

`service` is a virtual module that is fulfilled by one of the following modules:

Execution Module	Used for
debian_service	Debian Wheezy and earlier
freebsdservice	FreeBSD-based OSes using `service(8)`
gentoo_service	Gentoo Linux using `sysvinit` and `rc-update(8)`
mac_service	Mac OS hosts using `launchctl(1)`
netbsdservice	NetBSD-based OSes
openbsdservice	OpenBSD-based OSes
rh_service	RedHat-based distros and derivatives using `service(8)` and `chkconfig(8)`. Supports both pure sysvinit and mixed sysvinit/upstart systems.
service	Fallback which simply wraps sysvinit scripts
smf_service	Solaris-based OSes which use SMF
systemd_service	Linux distros which use systemd
upstart_service	Ubuntu-based distros using upstart
win_service	Windows

21.9.5 salt.modules.shadow

`shadow` is a virtual module that is fulfilled by one of the following modules:

Execution Module	Used for
shadow	Linux
bsd_shadow	FreeBSD, OpenBSD, NetBSD
solaris_shadow	Solaris-based OSes
win_shadow	Windows

21.9.6 salt.modules.user

`user` is a virtual module that is fulfilled by one of the following modules:

Execution Module	Used for
useradd	Linux, NetBSD, and OpenBSD systems using `useradd(8)`, `userdel(8)`, and `usermod(8)`
pw_user	FreeBSD-based OSes using `pw(8)`
solaris_user	Solaris-based OSes using `useradd(1M)`, `userdel(1M)`, and `usermod(1M)`
mac_user	MacOS
win_useradd	Windows

---

acme	ACME / Let's Encrypt module
aix_group	Manage groups on Solaris
aixpkg	Package support for AIX
aliases	Manage the information in the aliases file
alternatives	Support for Alternatives system
ansiblegate	Ansible Support
apache	Support for Apache

Continued on next page

Table 9 – continued from previous page

apcups	Module for apcupsd
apf	Support for Advanced Policy Firewall (APF)
apkpkg	Support for apk
aptpkg	Support for APT (Advanced Packaging Tool)
archive	A module to wrap (non-Windows) archive calls
arista_pyeapi	Arista pyeapi
artifactory	Module for fetching artifacts from Artifactory
at	Wrapper module for at(1)
at_solaris	Wrapper for at(1) on Solaris-like systems
augeas_cfg	Manages configuration files via augeas
aws_sqs	Support for the Amazon Simple Queue Service.
azurearm_compute	Azure (ARM) Compute Execution Module
azurearm_network	Azure (ARM) Network Execution Module
azurearm_resource	Azure (ARM) Resource Execution Module
bamboohr	Support for BambooHR
bcache	Module for managing BCache sets
beacons	Module for managing the Salt beacons on a minion
bigip	An execution module which can manipulate an f5 bigip via iControl REST
bluez_bluetooth	Support for Bluetooth (using BlueZ in Linux).
boto3_elasticache	Execution module for Amazon Elasticache using boto3
boto3_elasticsearch	Connection module for Amazon Elasticsearch Service
boto3_route53	Execution module for Amazon Route53 written against Boto 3
boto_apigateway	Connection module for Amazon APIGateway
boto_asg	Connection module for Amazon Autoscale Groups
boto_cfn	Connection module for Amazon Cloud Formation
boto_cloudfront	Connection module for Amazon CloudFront
boto_cloudtrail	Connection module for Amazon CloudTrail
boto_cloudwatch	Connection module for Amazon CloudWatch
boto_cloudwatch_event	Connection module for Amazon CloudWatch Events
boto_cognitoidentity	Connection module for Amazon CognitoIdentity
boto_datapipeline	Connection module for Amazon Data Pipeline
boto_dynamodb	Connection module for Amazon DynamoDB
boto_ec2	Connection module for Amazon EC2
boto_efs	Connection module for Amazon EFS
boto_elasticache	Connection module for Amazon Elasticache
boto_elasticsearch_domain	Connection module for Amazon Elasticsearch Service
boto_elb	Connection module for Amazon ELB
boto_elbv2	Connection module for Amazon ALB
boto_iam	Connection module for Amazon IAM
boto_iot	Connection module for Amazon IoT
boto_kinesis	Connection module for Amazon Kinesis
boto_kms	Connection module for Amazon KMS
boto_lambda	Connection module for Amazon Lambda
boto_rds	Connection module for Amazon RDS
boto_route53	Connection module for Amazon Route53
boto_s3_bucket	Connection module for Amazon S3 Buckets
boto_secgroup	Connection module for Amazon Security Groups
boto_sns	Connection module for Amazon SNS

Continued on next page

Table 9 – continued from previous page

boto_sqs	Connection module for Amazon SQS
boto_vpc	Connection module for Amazon VPC
bower	Manage and query Bower packages
bridge	Module for gathering and managing bridging information
bsd_shadow	Manage the password database on BSD systems
btrfs	Module for managing BTRFS file systems.
cabal	Manage and query Cabal packages
capirca_acl	Capirca ACL
cassandra	Cassandra NoSQL Database Module
cassandra_cql	Cassandra Database Module
celery	Support for scheduling celery tasks.
ceph	Module to provide ceph control with salt.
chassis	Glue execution module to link to the *fx2 proxymodule*.
chef	Execute chef in server or solo mode
chocolatey	A dead simple module wrapping calls to the Chocolatey package manager (http://chocolatey.org)
chronos	Module providing a simple management interface to a chronos cluster.
cimc	Module to provide Cisco UCS compatibility to Salt
ciscoconfparse_mod	Execution module for ciscoconfparse
cisconso	Execution module for Cisco Network Services Orchestrator Proxy minions
cloud	Salt-specific interface for calling Salt Cloud directly
cmdmod	A module for shelling out.
composer	Use composer to install PHP dependencies for a directory
config	Return config information
consul	Interact with Consul
container_resource	Common resources for LXC and systemd-nspawn containers
cp	Minion side functions for salt-cp
cpan	Manage Perl modules using CPAN
cron	Work with cron
csf	Support for Config Server Firewall (CSF)
cyg	Manage cygwin packages.
daemontools	daemontools service module.
data	Manage a local persistent data structure that can hold any arbitrary data specific to the minion
datadog_api	An execution module that interacts with the Datadog API
ddns	Support for RFC 2136 dynamic DNS updates.
deb_apache	Support for Apache
deb_postgres	Module to provide Postgres compatibility to salt for debian family specific tools.
debuild_pkgbuild	Debian Package builder system
debconfmod	Support for Debconf
debian_ip	The networking module for Debian-based distros
debian_service	Service support for Debian systems (uses update-rc.d and /sbin/service)

Continued on next page

Table 9 – continued from previous page

defaults	Module to work with salt formula defaults files
devmap	Device-Mapper module
dig	Compendium of generic DNS utilities.
disk	Module for managing disks and blockdevices
djangomod	Manage Django sites
dnsmasq	Module for managing dnsmasq
dnsutil	Compendium of generic DNS utilities.
dockercompose	Module to import docker-compose via saltstack
dockermod	Management of Docker Containers
dpkg_lowpkg	Support for DEB packages
drac	Manage Dell DRAC
dracr	Manage Dell DRAC.
drbd	DRBD administration module
dummyproxy_pkg	Package support for the dummy proxy used by the test suite
dummyproxy_service	Provide the service module for the dummy proxy used in integration tests
ebuildpkg	Support for Portage
eix	Support for Eix
elasticsearch	Elasticsearch - A distributed RESTful search and analytics server
environ	Support for getting and setting the environment variables of the current salt process.
eselect	Support for eselect, Gentoo's configuration and management tool.
esxi	Glues the VMware vSphere Execution Module to the VMware ESXi Proxy Minions to the *esxi proxymodule*.
etcd_mod	Execution module to work with etcd
ethtool	Module for running ethtool command
event	Use the *Salt Event System* to fire events from the master to the minion and vice-versa.
extfs	Module for managing ext2/3/4 file systems
file	Manage information about regular files, directories, and special files on the minion, set/read user, group, mode, and data
firewalld	Support for firewalld.
freebsd_sysctl	Module for viewing and modifying sysctl parameters
freebsd_update	Support for freebsd-update utility on FreeBSD.
freebsdjail	The jail module for FreeBSD
freebsdkmod	Module to manage FreeBSD kernel modules
freebsdpkg	Remote package support using `pkg_add(1)`
freebsdports	Install software from the FreeBSD `ports(7)` system
freebsdservice	The service module for FreeBSD
gem	Manage ruby gems.
genesis	Module for managing container and VM images
gentoo_service	Top level package command wrapper, used to translate the os detected by grains to the correct service manager
gentoolkitmod	Support for Gentoolkit
git	Support for the Git SCM
github	Module for interacting with the GitHub v3 API.

Continued on next page

Table 9 – continued from previous page

glance	Module for handling openstack glance calls.
glanceng	Glance module for interacting with OpenStack Glance
glusterfs	Manage a glusterfs pool
gnomedesktop	GNOME implementations
google_chat	Module for sending messages to google chat.
gpg	Manage a GPG keychains, add keys, create keys, retrieve keys from keyservers.
grafana4	Module for working with the Grafana v4 API
grains	Return/control aspects of the grains data
groupadd	Manage groups on Linux, OpenBSD and NetBSD
grub_legacy	Support for GRUB Legacy
guestfs	Interact with virtual machine images via libguestfs
hadoop	Support for hadoop
haproxyconn	Support for haproxy
hashutil	A collection of hashing and encoding functions
heat	Module for handling OpenStack Heat calls
hg	Support for the Mercurial SCM
hosts	Manage the information in the hosts file
http	Module for making various web calls.
ifttt	Support for IFTTT
ilo	Manage HP ILO
icinga2	Module to provide icinga2 compatibility to salt.
incron	Work with incron
influxdbmod	InfluxDB - A distributed time series database
influxdb08mod	InfluxDB - A distributed time series database
infoblox	This module have been tested on infoblox API v1.2.1, other versions of the API are likly workable.
ini_manage	Edit ini files
inspectlib	
inspectlib.collector	
inspectlib.dbhandle	
inspectlib.entities	
inspectlib.exceptions	
inspectlib.fsdb	**codeauthor** Bo Maryniuk <bo@suse.de>
inspectlib.kiwiproc	
inspectlib.query	
inspector	Module for full system inspection.
introspect	Functions to perform introspection on a minion, and return data in a format usable by Salt States
iosconfig	Cisco IOS configuration manipulation helpers
ipmi	Support IPMI commands over LAN.
ipset	Support for ipset
iptables	Support for iptables
iwtools	Support for Wireless Tools for Linux
jboss7	Module for managing JBoss AS 7 through the CLI interface.
jboss7_cli	Module for low-level interaction with JbossAS7 through CLI.

Continued on next page

Table 9 – continued from previous page

jenkinsmod	Module for controlling Jenkins
jinja	Module for checking jinja maps and verifying the result of loading JSON/YAML files
jira_mod	JIRA Execution module
junos	Module to interact with Junos devices.
k8s	Salt module to manage Kubernetes cluster
kapacitor	Kapacitor execution module.
kerberos	Manage Kerberos KDC
kernelpkg_linux_apt	Manage Linux kernel packages on APT-based systems
kernelpkg_linux_yum	Manage Linux kernel packages on YUM-based systems
key	Functions to view the minion's public key information
keyboard	Module for managing keyboards on supported POSIX-like systems using systemd, or such as Redhat, Debian and Gentoo.
keystone	Module for handling openstack keystone calls.
keystoneng	Keystone module for interacting with OpenStack Keystone
keystore	Module to interact with keystores
kmod	Module to manage Linux kernel modules
kubernetesmod	Module for handling kubernetes calls.
launchctl_service	Module for the management of MacOS systems that use launchd/launchctl
layman	Support for Layman
ldap3	Query and modify an LDAP database (alternative interface)
ldapmod	Salt interface to LDAP commands
libcloud_compute	Apache Libcloud Compute Management
libcloud_dns	Apache Libcloud DNS Management
libcloud_loadbalancer	Apache Libcloud Load Balancer Management
libcloud_storage	Apache Libcloud Storage Management
linux_acl	Support for Linux File Access Control Lists
linux_ip	The networking module for Non-RH/Deb Linux distros
linux_lvm	Support for Linux LVM2
linux_sysctl	Module for viewing and modifying sysctl parameters
localemod	Module for managing locales on POSIX-like systems.
locate	Module for using the locate utilities
logadm	Module for managing Solaris logadm based log rotations.
logmod	On-demand logging
logrotate	Module for managing logrotate.
lvs	Support for LVS (Linux Virtual Server)
lxc	Control Linux Containers via Salt
lxd	Module for managing the LXD daemon and its containers.
mac_assistive	This module allows you to manage assistive access on macOS minions with 10.9+
mac_brew_pkg	Homebrew for macOS
macdefaults	Set defaults on Mac OS
mac_desktop	macOS implementations of various commands in the "desktop" interface
mac_group	Manage groups on Mac OS 10.7+

Continued on next page

Table 9 – continued from previous page

mac_keychain	Install certificates into the keychain on Mac OS
macpackage	Install pkg, dmg and .app applications on macOS minions.
mac_pkgutil	Installer support for macOS.
mac_portspkg	Support for MacPorts under macOS.
mac_power	Module for editing power settings on macOS
mac_service	The service module for macOS
mac_shadow	Manage macOS local directory passwords and policies
mac_softwareupdate	Support for the softwareupdate command on MacOS.
mac_sysctl	Module for viewing and modifying sysctl parameters
mac_system	System module for sleeping, restarting, and shutting down the system on Mac OS X
mac_timezone	Module for editing date/time settings on macOS
mac_user	Manage users on Mac OS 10.7+
mac_xattr	This module allows you to manage extended attributes on files or directories
makeconf	Support for modifying make.conf under Gentoo
mandrill	Mandrill
marathon	Module providing a simple management interface to a marathon cluster.
match	The match module allows for match routines to be run and determine target specs
mattermost	Module for sending messages to Mattermost
mdadm_raid	Salt module to manage RAID arrays with mdadm
mdata	Module for managaging metadata in SmartOS Zones
memcached	Module for Management of Memcached Keys
mine	The function cache system allows for data to be stored on the master so it can be easily read by other minions
minion	Module to provide information about minions
mod_random	Provides access to randomness generators.
modjk	Control Modjk via the Apache Tomcat "Status" worker (http://tomcat.apache.org/connectors-doc/reference/status.html)
mongodb	Module to provide MongoDB functionality to Salt
monit	Monit service module.
moosefs	Module for gathering and managing information about MooseFS
mount	Salt module to manage Unix mounts and the fstab file
mssql	Module to provide MS SQL Server compatibility to salt.
msteams	Module for sending messages to MS Teams
munin	Run munin plugins/checks from salt and format the output as data.
mysql	Module to provide MySQL compatibility to salt.
nacl	This module helps include encrypted passwords in pillars, grains and salt state files.
nagios	Run nagios plugins/checks from salt and get the return as data.
nagios_rpc	Check Host & Service status from Nagios via JSON RPC.
namecheap_domains_dns	Namecheap DNS Management
namecheap_domains	Namecheap Domain Management

Continued on next page

Table 9 – continued from previous page

namecheap_domains_ns	Namecheap Nameserver Management
namecheap_ssl	Namecheap SSL Certificate Management
namecheap_users	Namecheap User Management
napalm_mod	NAPALM helpers
napalm_netacl	NAPALM ACL
napalm_bgp	NAPALM BGP
napalm_formula	NAPALM Formula helpers
napalm_mod	NAPALM helpers
napalm_network	NAPALM Network
napalm_ntp	NAPALM NTP
napalm_probes	NAPALM Probes
napalm_route	NAPALM Route
napalm_snmp	NAPALM SNMP
napalm_users	NAPALM Users
napalm_yang_mod	NAPALM YANG
netaddress	Module for getting information about network addresses.
netbox	NetBox
netbsd_sysctl	Module for viewing and modifying sysctl parameters
netbsdservice	The service module for NetBSD
netmiko_mod	Netmiko Execution Module
netscaler	Module to provide Citrix Netscaler compatibility to Salt (compatible with netscaler 9.2+)
network	Module for gathering and managing network information
neutron	Module for handling OpenStack Neutron calls
neutronng	Neutron module for interacting with OpenStack Neutron
nfs3	Module for managing NFS version 3.
nftables	Support for nftables
nginx	Support for nginx
nilrt_ip	The networking module for NI Linux Real-Time distro
nix	Work with Nix packages
nova	Module for handling OpenStack Nova calls
npm	Manage and query NPM packages.
nspawn	Manage nspawn containers
nxos	Execution module for Cisco NX OS Switches Proxy minions
nxos_api	Execution module to manage Cisco Nexus Switches (NX-OS) over the NX-API
omapi	This module interacts with an ISC DHCP Server via OMAPI.
openbsd_sysctl	Module for viewing and modifying OpenBSD sysctl parameters
openbsdpkg	Package support for OpenBSD
openbsdrcctl_service	The rcctl service module for OpenBSD
openbsdservice	The service module for OpenBSD
openscap	Module for OpenSCAP Management
openstack_config	Modify, retrieve, or delete values from OpenStack configuration files.
openstack_mng	Module for OpenStack Management

Continued on next page

Table 9 – continued from previous page

openvswitch	Support for Open vSwitch - module with basic Open vSwitch commands.
opkg	Support for Opkg
opsgenie	Module for sending data to OpsGenie
oracle	Oracle DataBase connection module
osquery	Support for OSQuery - https://osquery.io.
out	Output Module
pacmanpkg	A module to wrap pacman calls, since Arch is the best (https://wiki.archlinux.org/index.php/Arch_is_the_best)
pagerduty	Module for Firing Events via PagerDuty
pagerduty_util	Module for manageing PagerDuty resource
pam	Support for pam
panos	Module to provide Palo Alto compatibility to Salt
parallels	Manage Parallels Desktop VMs with `prlctl` and `prlsrvctl`.
parted_partition	Module for managing partitions on POSIX-like systems.
pcs	Configure a Pacemaker/Corosync cluster with PCS
pdbedit	Manage accounts in Samba's passdb using pdbedit
pecl	Manage PHP pecl extensions.
peeringdb	PeeringDB Module
pf	Control the OpenBSD packet filter (PF).
philips_hue	Philips HUE lamps module for proxy.
pillar	Extract the pillar data for this minion
pip	Install Python packages with pip to either the system or a virtualenv
pkg_resource	Resources needed by pkg providers
pkgin	Package support for pkgin based systems, inspired from freebsdpkg module
pkgng	Support for `pkgng`, the new package manager for FreeBSD
pkgutil	Pkgutil support for Solaris
portage_config	Configure `portage(5)`
postfix	Support for Postfix
postgres	Module to provide Postgres compatibility to salt.
poudriere	Support for poudriere
powerpath	powerpath support.
proxy	This module allows you to manage proxy settings
ps	A salt interface to psutil, a system and process library.
publish	Publish a command from a minion to a target
puppet	Execute puppet routines
purefa	Management of Pure Storage FlashArray
purefb	Management of Pure Storage FlashBlade
pushbullet	Module for sending messages to Pushbullet (https://www.pushbullet.com)
pushover_notify	Module for sending messages to Pushover (https://www.pushover.net)
pw_group	Manage groups on FreeBSD
pw_user	Manage users with the pw command
pyenv	Manage python installations with pyenv.

Continued on next page

Table 9 – continued from previous page

qemu_img	Qemu-img Command Wrapper
qemu_nbd	Qemu Command Wrapper
quota	Module for managing quotas on POSIX-like systems.
rabbitmq	Module to provide RabbitMQ compatibility to Salt.
rallydev	Support for RallyDev
random_org	Module for retrieving random information from Random.org
rbac_solaris	Module for Solaris' Role-Based Access Control
rbenv	Manage ruby installations with rbenv.
rdp	Manage RDP Service on Windows servers
redismod	Module to provide redis functionality to Salt
reg	Manage the Windows registry
rest_pkg	Package support for the REST example
rest_sample_utils	Utility functions for the rest_sample
rest_service	Provide the service module for the proxy-minion REST sample
restartcheck	checkrestart functionality for Debian and Red Hat Based systems
ret	Module to integrate with the returner system and retrieve data sent to a salt returner
rh_ip	The networking module for RHEL/Fedora based distros
rh_service	Service support for RHEL-based systems, including support for both upstart and sysvinit
riak	Riak Salt Module
rpm_lowpkg	Support for rpm
rpmbuild_pkgbuild	RPM Package builder system
rsync	Wrapper for rsync
runit	runit service module (http://smarden.org/runit)
rvm	Manage ruby installations and gemsets with RVM, the Ruby Version Manager.
s3	Connection module for Amazon S3
s6	s6 service module
salt_proxy	Salt proxy module
salt_version	Access Salt's elemental release code-names.
saltcheck	A module for testing the logic of states and highstates
saltcloudmod	Control a salt cloud system
saltutil	The Saltutil module is used to manage the state of the salt minion itself.
schedule	Module for managing the Salt schedule on a minion
scp_mod	SCP Module
scsi	SCSI administration module
sdb	Module for Manipulating Data via the Salt DB API
seed	Virtual machine image management tools
selinux	Execute calls on selinux
sensehat	Module for controlling the LED matrix or reading environment data on the SenseHat of a Raspberry Pi.
sensors	Read lm-sensors
serverdensity_device	Wrapper around Server Density API
service	If Salt's OS detection does not identify a different virtual service module, the minion will fall back to using this basic module, which simply wraps sysvinit scripts.

Continued on next page

Table 9 – continued from previous page

`servicenow`	Module for execution of ServiceNow CI (configuration items)
`shadow`	Manage the shadow file on Linux systems
`slack_notify`	Module for sending messages to Slack
`slsutil`	Utility functions for use with or in SLS files
`smartos_imgadm`	Module for running imgadm command on SmartOS
`smartos_nictagadm`	Module for running nictagadm command on SmartOS :maintainer: Jorge Schrauwen <sjorge@blackdot.be> :maturity: new :depends: nictagadm binary, dladm binary :platform: smartos
`smartos_virt`	virst compatibility module for managing VMs on SmartOS
`smartos_vmadm`	Module for running vmadm command on SmartOS
`smbios`	Interface to SMBIOS/DMI
`smf_service`	Service support for Solaris 10 and 11, should work with other systems that use SMF also.
`smtp`	Module for Sending Messages via SMTP
`solaris_fmadm`	Module for running fmadm and fmdump on Solaris
`solaris_group`	Manage groups on Solaris
`solaris_shadow`	Manage the password database on Solaris systems
`solaris_system`	Support for reboot, shutdown, etc
`solaris_user`	Manage users with the useradd command
`solarisipspkg`	IPS pkg support for Solaris
`solarispkg`	Package support for Solaris
`solr`	Apache Solr Salt Module
`solrcloud`	Module for solrcloud configuration
`splunk`	Module for interop with the Splunk API
`splunk_search`	Module for interop with the Splunk API
`sqlite3`	Support for SQLite3
`ssh`	Manage client ssh components
`ssh_pkg`	Service support for the REST example
`ssh_service`	Provide the service module for the proxy-minion SSH sample ..
`snapper`	Module to manage filesystem snapshots with snapper
`state`	Control the state system on the minion.
`status`	Module for returning various status data about a minion.
`statuspage`	StatusPage
`supervisord`	Provide the service module for system supervisord or supervisord in a virtualenv
`suse_apache`	Support for Apache
`svn`	Subversion SCM
`swarm`	Docker Swarm Module using Docker's Python SDK
`swift`	Module for handling OpenStack Swift calls Author: Anthony Stanton <anthony.stanton@gmail.com>
`sysbench`	The 'sysbench' module is used to analyze the performance of the minions, right from the master! It measures various system parameters such as CPU, Memory, File I/O, Threads and Mutex.
`sysfs`	Module for interfacing with SysFS
`syslog_ng`	Module for getting information about syslog-ng

Continued on next page

Table 9 – continued from previous page

sysmod	The sys module provides information about the available functions on the minion
sysrc	sysrc module for FreeBSD
system	Support for reboot, shutdown, etc on POSIX-like systems.
system_profiler	System Profiler Module
systemd_service	Provides the service module for systemd
telegram	Module for sending messages via Telegram.
telemetry	Connection module for Telemetry
temp	Simple module for creating temporary directories and files
test	Module for running arbitrary tests
testinframod	This module exposes the functionality of the TestInfra library for use with SaltStack in order to verify the state of your minions.
test_virtual	Module for running arbitrary tests with a __virtual__ function
textfsm_mod	TextFSM
timezone	Module for managing timezone on POSIX-like systems.
tls	A salt module for SSL/TLS.
tomcat	Support for Tomcat
trafficserver	Apache Traffic Server execution module.
travisci	Commands for working with travisci.
tuned	Interface to Red Hat tuned-adm module
twilio_notify	Module for notifications via Twilio
udev	Manage and query udev info
upstart_service	Module for the management of upstart systems.
uptime	Wrapper around uptime API
useradd	Manage users with the useradd command
uwsgi	uWSGI stats server https://uwsgi-docs.readthedocs.io/en/latest/StatsServer.html
vagrant	Work with virtual machines managed by Vagrant.
varnish	Support for Varnish
vault	Functions to interact with Hashicorp Vault.
vbox_guest	VirtualBox Guest Additions installer
vboxmanage	Support for VirtualBox using the VBoxManage command
victorops	Support for VictorOps
virt	Work with virtual machines managed by libvirt
virtualenv_mod	Create virtualenv environments.
vmctl	Manage vms running on the OpenBSD VMM hypervisor using vmctl(8).
vsphere	Manage VMware vCenter servers and ESXi hosts.
webutil	Support for htpasswd command.
win_autoruns	Module for listing programs that automatically run on startup (very alpha...not tested on anything but my Win 7x64)
win_certutil	This module allows you to install certificates into the windows certificate manager.
win_dacl	Manage DACLs on Windows
win_disk	Module for gathering disk information on Windows

Continued on next page

Table 9 – continued from previous page

win_dism	Install features/packages for Windows using DISM, which is useful for minions not running server versions of Windows.
win_dns_client	Module for configuring DNS Client on Windows systems
win_dsc	Module for working with Windows PowerShell DSC (Desired State Configuration)
win_file	Manage information about files on the minion, set/read user, group data, modify the ACL of files/directories
win_firewall	Module for configuring Windows Firewall using `netsh`
win_groupadd	Manage groups on Windows
win_iis	Microsoft IIS site management via WebAdministration powershell module
win_ip	The networking module for Windows based systems
win_lgpo	Manage Local Policy on Windows
win_license	This module allows you to manage windows licensing via slmgr.vbs
win_network	Module for gathering and managing network information
win_ntp	Management of NTP servers on Windows
win_path	Manage the Windows System PATH
win_pkg	A module to manage software on Windows
win_pki	Microsoft certificate management via the PKI Client PowerShell module.
win_powercfg	This module allows you to control the power settings of a windows minion via powercfg.
win_psget	Module for managing PowerShell through PowerShell-Get (PSGet)
winrepo	Module to manage Windows software repo on a Standalone Minion
win_servermanager	Manage Windows features via the ServerManager powershell module.
win_service	Windows Service module.
win_shadow	Manage the shadow file
win_smtp_server	Module for managing IIS SMTP server configuration on Windows servers.
win_snmp	Module for managing SNMP service settings on Windows servers.
win_status	Module for returning various status data about a minion.
win_system	Module for managing windows systems.
win_task	Windows Task Scheduler Module ..
win_timezone	Module for managing timezone on Windows systems.
win_useradd	Module for managing Windows Users
win_wua	Module for managing Windows Updates using the Windows Update Agent.
win_wusa	Microsoft Update files management via wusa.exe
x509	Manage X509 certificates
xapi_virt	This module (mostly) uses the XenAPI to manage Xen virtual machines.

Continued on next page

Table 9 – continued from previous page

xbpspkg	Package support for XBPS package manager (used by VoidLinux)
xfs	Module for managing XFS file systems.
xml	XML file manager
xmpp	Module for Sending Messages via XMPP (a.k.a.
yumpkg	Support for YUM/DNF
zabbix	Support for Zabbix
zcbuildout	Management of zc.buildout
zenoss	Module for working with the Zenoss API
zfs	Module for running ZFS command
zk_concurrency	Concurrency controls in zookeeper
znc	znc - An advanced IRC bouncer
zoneadm	Module for Solaris 10's zoneadm
zonecfg	Module for Solaris 10's zonecfg
zookeeper	Zookeeper Module
zpool	Module for running ZFS zpool command
zypperpkg	Package support for openSUSE via the zypper package manager

21.9.7 salt.modules.acme module

ACME / Let's Encrypt module

This module currently looks for certbot script in the $PATH as - certbot, - lestsencrypt, - certbot-auto, - letsencrypt-auto eventually falls back to /opt/letsencrypt/letsencrypt-auto

Note: Installation & configuration of the Let's Encrypt client can for example be done using https://github.com/saltstack-formulas/letsencrypt-formula

Warning: Be sure to set at least accept-tos = True in cli.ini!

Most parameters will fall back to cli.ini defaults if None is given.

DNS plugins

This module currently supports the CloudFlare certbot DNS plugin. The DNS plugin credentials file needs to be passed in using the `dns_plugin_credentials` argument.

Make sure the appropriate certbot plugin for the wanted DNS provider is installed before using this module.

`salt.modules.acme.`**`cert`**(*name, aliases=None, email=None, webroot=None, test_cert=False, renew=None, keysize=None, server=None, owner='root', group='root', mode='0640', certname=None, preferred_challenges=None, tls_sni_01_port=None, tls_sni_01_address=None, http_01_port=None, http_01_address=None, dns_plugin=None, dns_plugin_credentials=None*)

Obtain/renew a certificate from an ACME CA, probably Let's Encrypt.

Parameters

- **name** – Common Name of the certificate (DNS name of certificate)
- **aliases** – subjectAltNames (Additional DNS names on certificate)
- **email** – e-mail address for interaction with ACME provider
- **webroot** – True or a full path to use to use webroot. Otherwise use standalone mode
- **test_cert** – Request a certificate from the Happy Hacker Fake CA (mutually exclusive with 'server')
- **renew** – True/'force' to force a renewal, or a window of renewal before expiry in days
- **keysize** – RSA key bits
- **server** – API endpoint to talk to
- **owner** – owner of the private key file
- **group** – group of the private key file
- **mode** – mode of the private key file
- **certname** – Name of the certificate to save
- **preferred_challenges** – A sorted, comma delimited list of the preferred challenge to use during authorization with the most preferred challenge listed first.
- **tls_sni_01_port** – Port used during tls-sni-01 challenge. This only affects the port Certbot listens on. A conforming ACME server will still attempt to connect on port 443.
- **tls_sni_01_address** – The address the server listens to during tls-sni-01 challenge.
- **http_01_port** – Port used in the http-01 challenge. This only affects the port Certbot listens on. A conforming ACME server will still attempt to connect on port 80.
- **https_01_address** – The address the server listens to during http-01 challenge.
- **dns_plugin** – Name of a DNS plugin to use (currently only 'cloudflare' or 'digitalocean')
- **dns_plugin_credentials** – Path to the credentials file if required by the specified DNS plugin
- **dns_plugin_propagate_seconds** – Number of seconds to wait for DNS propogations before asking ACME servers to verify the DNS record. (default 10)

Return type dict

Returns Dictionary with 'result' True/False/None, 'comment' and certificate's expiry date ('not_after')

CLI example:

```
salt 'gitlab.example.com' acme.cert dev.example.com "[gitlab.example.com]" test_
↪cert=True         renew=14 webroot=/opt/gitlab/embedded/service/gitlab-rails/
↪public
```

salt.modules.acme.**certs**()

Return a list of active certificates

CLI example:

```
salt 'vhost.example.com' acme.certs
```

salt.modules.acme.**expires**(*name*)

The expiry date of a certificate in ISO format

Parameters **name** (*str*) – CommonName of certificate

Return type str

Returns Expiry date in ISO format.

CLI example:

```
salt 'gitlab.example.com' acme.expires dev.example.com
```

`salt.modules.acme.`**`has`**`(`*name*`)`

Test if a certificate is in the Let's Encrypt Live directory

Parameters **name** (*str*) – CommonName of certificate

Return type bool

Code example:

```
if __salt__['acme.has']('dev.example.com'):
    log.info('That is one nice certificate you have there!')
```

`salt.modules.acme.`**`info`**`(`*name*`)`

Return information about a certificate

Parameters **name** (*str*) – CommonName of certificate

Return type dict

Returns Dictionary with information about the certificate. If neither the `tls` nor the `x509` module can be used to determine the certificate information, the information will be retrieved as one big text block under the key `text` using the openssl cli.

CLI example:

```
salt 'gitlab.example.com' acme.info dev.example.com
```

`salt.modules.acme.`**`needs_renewal`**`(`*name, window=None*`)`

Check if a certificate needs renewal

Parameters

- **name** (*str*) – CommonName of certificate
- **window** (*bool/str/int*) – Window in days to renew earlier or True/force to just return True

Return type bool

Returns Whether or not the certificate needs to be renewed.

Code example:

```
if __salt__['acme.needs_renewal']('dev.example.com'):
    __salt__['acme.cert']('dev.example.com', **kwargs)
else:
    log.info('Your certificate is still good')
```

`salt.modules.acme.`**`renew_by`**`(`*name, window=None*`)`

Date in ISO format when a certificate should first be renewed

Parameters

- **name** (*str*) – CommonName of certificate

- **window** (*int*) – number of days before expiry when renewal should take place

Return type str

Returns Date of certificate renewal in ISO format.

21.9.8 salt.modules.aix_group module

Manage groups on Solaris

Important: If you feel that Salt should be using this module to manage groups on a minion, and it is using a different module (or gives an error similar to *'group.info' is not available*), see *here*.

`salt.modules.aix_group.`**`add`**`(`*name, gid=None, system=False, root=None*`)`

Add the specified group

CLI Example:

```
salt '*' group.add foo 3456
```

`salt.modules.aix_group.`**`adduser`**`(`*name, username, root=None*`)`

Add a user in the group.

CLI Example:

```
salt '*' group.adduser foo bar
```

Verifies if a valid username 'bar' as a member of an existing group 'foo', if not then adds it.

`salt.modules.aix_group.`**`chgid`**`(`*name, gid*`)`

Change the gid for a named group

CLI Example:

```
salt '*' group.chgid foo 4376
```

`salt.modules.aix_group.`**`delete`**`(`*name*`)`

Remove the named group

CLI Example:

```
salt '*' group.delete foo
```

`salt.modules.aix_group.`**`deluser`**`(`*name, username, root=None*`)`

Remove a user from the group.

CLI Example:

```
salt '*' group.deluser foo bar
```

Removes a member user 'bar' from a group 'foo'. If group is not present then returns True.

`salt.modules.aix_group.`**`getent`**`(`*refresh=False*`)`

Return info on all groups

CLI Example:

```
salt '*' group.getent
```

`salt.modules.aix_group.`**`info`**`(`*name*`)`

Return information about a group

CLI Example:

```
salt '*' group.info foo
```

`salt.modules.aix_group.`**`members`**`(`*name*, *members_list*, *root=None*`)`

Replaces members of the group with a provided list.

CLI Example:

> salt '*' group.members foo 'user1,user2,user3,…'

Replaces a membership list for a local group 'foo'. foo:x:1234:user1,user2,user3,…

21.9.9 salt.modules.aixpkg

Package support for AIX

Important: If you feel that Salt should be using this module to manage filesets or rpm packages on a minion, and it is using a different module (or gives an error similar to *'pkg.install' is not available*), see *here*.

`salt.modules.aixpkg.`**`available_version`**`(`**names*, ***kwargs*`)`

This function is an alias of `latest_version`.

> Return the latest version of the named fileset/rpm package available for upgrade or installation. If more than one fileset/rpm package name is specified, a dict of name/version pairs is returned.
>
> If the latest version of a given fileset/rpm package is already installed, an empty string will be returned for that package.
>
> CLI Example:
>
> ```
> salt '*' pkg.latest_version <package name>
> salt '*' pkg.latest_version <package1> <package2> <package3> ...
> ```
>
> NOTE: Repositories are not presently supported for AIX. This function will always return an empty string for a given fileset/rpm package.

`salt.modules.aixpkg.`**`install`**`(`*name=None*, *refresh=False*, *pkgs=None*, *version=None*, *test=False*, ***kwargs*`)`

Install the named fileset(s)/rpm package(s).

name The name of the fileset or rpm package to be installed.

refresh Whether or not to update the yum database before executing.

Multiple Package Installation Options:

pkgs A list of filesets and/or rpm packages to install. Must be passed as a python list. The `name` parameter will be ignored if this option is passed.

version Install a specific version of a fileset/rpm package. (Unused at present).

test Verify that command functions correctly:

Returns a dict containing the new fileset(s)/rpm package(s) names and versions:

> **{'<package>': {'old': '<old-version>',** 'new': '<new-version>'}}

CLI Example:

```
salt '*' pkg.install /stage/middleware/AIX/bash-4.2-3.aix6.1.ppc.rpm
salt '*' pkg.install /stage/middleware/AIX/bash-4.2-3.aix6.1.ppc.rpm refresh=True
salt '*' pkg.install /stage/middleware/AIX/VIOS2211_update/tpc_4.1.1.85.bff
salt '*' pkg.install /stage/middleware/AIX/Xlc/usr/sys/inst.images/xlC.rte
salt '*' pkg.install /stage/middleware/AIX/Firefox/ppc-AIX53/Firefox.base
salt '*' pkg.install pkgs='["foo", "bar"]'
```

`salt.modules.aixpkg.`**`latest_version`**`(`**names, **kwargs*`)`

Return the latest version of the named fileset/rpm package available for upgrade or installation. If more than one fileset/rpm package name is specified, a dict of name/version pairs is returned.

If the latest version of a given fileset/rpm package is already installed, an empty string will be returned for that package.

CLI Example:

```
salt '*' pkg.latest_version <package name>
salt '*' pkg.latest_version <package1> <package2> <package3> ...
```

NOTE: Repositories are not presently supported for AIX. This function will always return an empty string for a given fileset/rpm package.

`salt.modules.aixpkg.`**`list_pkgs`**`(`*versions_as_list=False, **kwargs*`)`

List the filesets/rpm packages currently installed as a dict:

```
{'<package_name>': '<version>'}
```

CLI Example:

```
salt '*' pkg.list_pkgs
```

`salt.modules.aixpkg.`**`remove`**`(`*name=None, pkgs=None, **kwargs*`)`

Remove specified fileset(s)/rpm package(s).

name The name of the fileset or rpm package to be deleted.

Multiple Package Options:

pkgs A list of filesets and/or rpm packages to delete. Must be passed as a python list. The `name` parameter will be ignored if this option is passed.

Returns a list containing the removed packages.

CLI Example:

```
salt '*' pkg.remove <fileset/rpm package name>
salt '*' pkg.remove tcsh
salt '*' pkg.remove xlC.rte
salt '*' pkg.remove Firefox.base.adt
salt '*' pkg.remove pkgs='["foo", "bar"]'
```

`salt.modules.aixpkg.`**`upgrade_available`**`(`*name*`)`

Check whether or not an upgrade is available for a given package

CLI Example:

```
salt '*' pkg.upgrade_available <package name>
```

`salt.modules.aixpkg.`**`version`**`(`**names, **kwargs*`)`

Common interface for obtaining the version of installed fileset/rpm package.

CLI Example:

```
salt '*' pkg.version vim
salt '*' pkg.version foo bar baz
```

21.9.10 salt.modules.aliases

Manage the information in the aliases file

`salt.modules.aliases.`**`get_target`**`(`*alias*`)`

Return the target associated with an alias

CLI Example:

```
salt '*' aliases.get_target alias
```

`salt.modules.aliases.`**`has_target`**`(`*alias, target*`)`

Return true if the alias/target is set

CLI Example:

```
salt '*' aliases.has_target alias target
```

`salt.modules.aliases.`**`list_aliases`**`()`

Return the aliases found in the aliases file in this format:

```
{'alias': 'target'}
```

CLI Example:

```
salt '*' aliases.list_aliases
```

`salt.modules.aliases.`**`rm_alias`**`(`*alias*`)`

Remove an entry from the aliases file

CLI Example:

```
salt '*' aliases.rm_alias alias
```

`salt.modules.aliases.`**`set_target`**`(`*alias, target*`)`

Set the entry in the aliases file for the given alias, this will overwrite any previous entry for the given alias or create a new one if it does not exist.

CLI Example:

```
salt '*' aliases.set_target alias target
```

21.9.11 salt.modules.alternatives

Support for Alternatives system

codeauthor Radek Rada <radek.rada@gmail.com>

`salt.modules.alternatives.`**`auto`**`(`*name*`)`

Trigger alternatives to set the path for <name> as specified by priority.

CLI Example:

```
salt '*' alternatives.auto name
```

`salt.modules.alternatives.`**`check_exists`**`(`*name*, *path*`)`

Check if the given path is an alternative for a name.

New in version 2015.8.4.

CLI Example:

```
salt '*' alternatives.check_exists name path
```

`salt.modules.alternatives.`**`check_installed`**`(`*name*, *path*`)`

Check if the current highest-priority match for a given alternatives link is set to the desired path

CLI Example:

```
salt '*' alternatives.check_installed name path
```

`salt.modules.alternatives.`**`display`**`(`*name*`)`

Display alternatives settings for defined command name

CLI Example:

```
salt '*' alternatives.display editor
```

`salt.modules.alternatives.`**`install`**`(`*name*, *link*, *path*, *priority*`)`

Install symbolic links determining default commands

CLI Example:

```
salt '*' alternatives.install editor /usr/bin/editor /usr/bin/emacs23 50
```

`salt.modules.alternatives.`**`remove`**`(`*name*, *path*`)`

Remove symbolic links determining the default commands.

CLI Example:

```
salt '*' alternatives.remove name path
```

`salt.modules.alternatives.`**`set_`**`(`*name*, *path*`)`

Manually set the alternative <path> for <name>.

CLI Example:

```
salt '*' alternatives.set name path
```

`salt.modules.alternatives.`**`show_current`**`(`*name*`)`

Display the current highest-priority alternative for a given alternatives link

CLI Example:

```
salt '*' alternatives.show_current editor
```

`salt.modules.alternatives.`**`show_link`**`(`*name*`)`

Display master link for the alternative

New in version 2015.8.13,2016.3.4,2016.11.0.

CLI Example:

```
salt '*' alternatives.show_link editor
```

21.9.12 salt.modules.ansiblegate

Ansible Support

This module can have an optional minion-level configuration in /etc/salt/minion.d/ as follows:

ansible_timeout: 1200

The timeout is how many seconds Salt should wait for any Ansible module to respond.

class `salt.modules.ansiblegate.`**`AnsibleModuleCaller`**(*resolver*)

`DEFAULT_TIMEOUT = 1200`

`OPT_TIMEOUT_KEY = 'ansible_timeout'`

`call`(*module*, **args*, ***kwargs*)

Call an Ansible module by invoking it. :param module: the name of the module. :param args: Arguments to the module :param kwargs: keywords to the module :return:

class `salt.modules.ansiblegate.`**`AnsibleModuleResolver`**(*opts*)

This class is to resolve all available modules in Ansible.

`get_modules_list`(*pattern=None*)

Return module map references. :return:

`install`()

`load_module`(*module*)

Introspect Ansible module.

Parameters **`module`** –

Returns

`resolve`()

`salt.modules.ansiblegate.`**`help`**(*module=None*, **args*)

Display help on Ansible standard module.

Parameters **`module`** –

Returns

`salt.modules.ansiblegate.`**`list`**(*pattern=None*)

Lists available modules. :return:

`salt.modules.ansiblegate.`**`playbooks`**(*playbook*, *rundir=None*, *check=False*, *diff=False*, *extra_vars=None*, *flush_cache=False*, *forks=5*, *inventory=None*, *limit=None*, *list_hosts=False*, *list_tags=False*, *list_tasks=False*, *module_path=None*, *skip_tags=None*, *start_at_task=None*, *syntax_check=False*, *tags=None*, *playbook_kwargs=None*)

Run Ansible Playbooks

Parameters

- **playbook** – Which playbook to run.
- **rundir** – Directory to run *ansible-playbook* in. (Default: None)
- **check** – don't make any changes; instead, try to predict some of the changes that may occur (Default: False)
- **diff** – when changing (small) files and templates, show the differences in those files; works great with –check (default: False)
- **extra_vars** – set additional variables as key=value or YAML/JSON, if filename prepend with @, (default: None)
- **flush_cache** – clear the fact cache for every host in inventory (default: False)
- **forks** – specify number of parallel processes to use (Default: 5)
- **inventory** – specify inventory host path or comma separated host list. (Default: None) (Ansible's default is /etc/ansible/hosts)
- **limit** – further limit selected hosts to an additional pattern (Default: None)
- **list_hosts** – outputs a list of matching hosts; does not execute anything else (Default: False)
- **list_tags** – list all available tags (Default: False)
- **list_tasks** – list all tasks that would be executed (Default: False)
- **module_path** – prepend colon-separated path(s) to module library. (Default: None)
- **skip_tags** – only run plays and tasks whose tags do not match these values (Default: False)
- **start_at_task** – start the playbook at the task matching this name (Default: None)
- **tags** – only run plays and tasks tagged with these values (Default: None)

Param syntax_check: perform a syntax check on the playbook, but do not execute it (Default: False)

Returns Playbook return

CLI Example:

```
salt 'ansiblehost'  ansible.playbook playbook=/srv/playbooks/play.yml
```

21.9.13 salt.modules.apache

Support for Apache

Note: The functions in here are generic functions designed to work with all implementations of Apache. Debian-specific functions have been moved into deb_apache.py, but will still load under the `apache` namespace when a Debian-based system is detected.

`salt.modules.apache.`**`config`**`(`*name, config, edit=True*`)`

Create VirtualHost configuration files

name File for the virtual host

config VirtualHost configurations

Note: This function is not meant to be used from the command line. Config is meant to be an ordered dict of all of the apache configs.

CLI Example:

```
salt '*' apache.config /etc/httpd/conf.d/ports.conf config="[{'Listen': '22'}]"
```

salt.modules.apache.**directives**()

Return list of directives together with expected arguments and places where the directive is valid (`apachectl -L`)

CLI Example:

```
salt '*' apache.directives
```

salt.modules.apache.**fullversion**()

Return server version (`apachectl -V`)

CLI Example:

```
salt '*' apache.fullversion
```

salt.modules.apache.**modules**()

Return list of static and shared modules (`apachectl -M`)

CLI Example:

```
salt '*' apache.modules
```

salt.modules.apache.**server_status**(*profile='default'*)

Get Information from the Apache server-status handler

Note: The server-status handler is disabled by default. In order for this function to work it needs to be enabled. See http://httpd.apache.org/docs/2.2/mod/mod_status.html

---

The following configuration needs to exists in pillar/grains. Each entry nested in `apache.server-status` is a profile of a vhost/server. This would give support for multiple apache servers/vhosts.

```
apache.server-status:
  default:
    url: http://localhost/server-status
    user: someuser
    pass: password
    realm: 'authentication realm for digest passwords'
    timeout: 5
```

CLI Examples:

```
salt '*' apache.server_status
salt '*' apache.server_status other-profile
```

salt.modules.apache.**servermods**()

Return list of modules compiled into the server (`apachectl -l`)

CLI Example:

```
salt '*' apache.servermods
```

`salt.modules.apache.`**`signal`**`(`*signal=None*`)`

Signals httpd to start, restart, or stop.

CLI Example:

```
salt '*' apache.signal restart
```

`salt.modules.apache.`**`useradd`**`(`*pwfile, user, password, opts=''*`)`

Add HTTP user using the `htpasswd` command. If the `htpasswd` file does not exist, it will be created. Valid options that can be passed are:

```
n  Don't update file; display results on stdout.
m  Force MD5 hashing of the password (default).
d  Force CRYPT(3) hashing of the password.
p  Do not hash the password (plaintext).
s  Force SHA1 hashing of the password.
```

CLI Examples:

```
salt '*' apache.useradd /etc/httpd/htpasswd larry badpassword
salt '*' apache.useradd /etc/httpd/htpasswd larry badpass opts=ns
```

`salt.modules.apache.`**`userdel`**`(`*pwfile, user*`)`

Delete HTTP user from the specified `htpasswd` file.

CLI Example:

```
salt '*' apache.userdel /etc/httpd/htpasswd larry
```

`salt.modules.apache.`**`version`**`()`

Return server version (`apachectl -v`)

CLI Example:

```
salt '*' apache.version
```

`salt.modules.apache.`**`vhosts`**`()`

Show the settings as parsed from the config file (currently only shows the virtualhost settings) (`apachectl -S`). Because each additional virtual host adds to the execution time, this command may require a long timeout be specified by using `-t 10`.

CLI Example:

```
salt -t 10 '*' apache.vhosts
```

21.9.14 salt.modules.apcups

Module for apcupsd

`salt.modules.apcups.`**`status`**`()`

Return apcaccess output

CLI Example:

```
salt '*' apcups.status
```

salt.modules.apcups.**status_battery**()

Return true if running on battery power

CLI Example:

```
salt '*' apcups.status_battery
```

salt.modules.apcups.**status_charge**()

Return battery charge

CLI Example:

```
salt '*' apcups.status_charge
```

salt.modules.apcups.**status_load**()

Return load

CLI Example:

```
salt '*' apcups.status_load
```

21.9.15 salt.modules.apf

Support for Advanced Policy Firewall (APF)

maintainer Mostafa Hussein <mostafa.hussein91@gmail.com>

maturity new

depends python-iptables

platform Linux

salt.modules.apf.**allow**(*ip*, *port=None*)

Add host (IP/FQDN) to allow_hosts.rules and immediately load new rule into firewall CLI Example:

```
salt '*' apf.allow 127.0.0.1
```

salt.modules.apf.**deny**(*ip*)

Add host (IP/FQDN) to deny_hosts.rules and immediately load new rule into firewall CLI Example:

```
salt '*' apf.deny 1.2.3.4
```

salt.modules.apf.**disable**()

Stop (flush) all firewall rules CLI Example:

```
salt '*' apf.disable
```

salt.modules.apf.**enable**()

Load all firewall rules CLI Example:

```
salt '*' apf.enable
```

salt.modules.apf.**refresh**()

Refresh & resolve dns names in trust rules CLI Example:

```
salt '*' apf.refresh
```

`salt.modules.apf.`**`reload`**`()`
Stop (flush) & reload firewall rules CLI Example:

```
salt '*' apf.reload
```

`salt.modules.apf.`**`remove`**`(`*ip*`)`
Remove host from [glob]*_hosts.rules and immediately remove rule from firewall CLI Example:

```
salt '*' apf.remove 1.2.3.4
```

`salt.modules.apf.`**`running`**`()`
Check apf status CLI Example:

```
salt '*' apf.running
```

21.9.16 salt.modules.apkpkg

Support for apk

Important: If you feel that Salt should be using this module to manage packages on a minion, and it is using a different module (or gives an error similar to *'pkg.install' is not available*), see *here*.

`salt.modules.apkpkg.`**`file_dict`**`(`**packages*`)`
List the files that belong to a package, grouped by package. Not specifying any packages will return a list of _every_ file on the system's package database (not generally recommended).

CLI Examples:

```
salt '*' pkg.file_list httpd
salt '*' pkg.file_list httpd postfix
salt '*' pkg.file_list
```

`salt.modules.apkpkg.`**`file_list`**`(`**packages*`)`
List the files that belong to a package. Not specifying any packages will return a list of _every_ file on the system's package database (not generally recommended).

CLI Examples:

```
salt '*' pkg.file_list httpd
salt '*' pkg.file_list httpd postfix
salt '*' pkg.file_list
```

`salt.modules.apkpkg.`**`install`**`(`*name=None, refresh=False, pkgs=None, sources=None, **kwargs*`)`
Install the passed package, add refresh=True to update the apk database.

name The name of the package to be installed. Note that this parameter is ignored if either "pkgs" or "sources" is passed. Additionally, please note that this option can only be used to install packages from a software repository. To install a package file manually, use the "sources" option.

CLI Example:

```
salt '*' pkg.install <package name>
```

refresh Whether or not to refresh the package database before installing.

Multiple Package Installation Options:

pkgs A list of packages to install from a software repository. Must be passed as a python list.

CLI Example:

```
salt '*' pkg.install pkgs='["foo", "bar"]'
```

sources A list of IPK packages to install. Must be passed as a list of dicts, with the keys being package names, and the values being the source URI or local path to the package. Dependencies are automatically resolved and marked as auto-installed.

CLI Example:

```
salt '*' pkg.install sources='[{"foo": "salt://foo.deb"},{"bar": "salt://bar.
↪deb"}]'
```

install_recommends Whether to install the packages marked as recommended. Default is True.

Returns a dict containing the new package names and versions:

```
{'<package>': {'old': '<old-version>',
               'new': '<new-version>'}}
```

`salt.modules.apkpkg.`**`latest_version`**(**names*, ***kwargs*)

Return the latest version of the named package available for upgrade or installation. If more than one package name is specified, a dict of name/version pairs is returned.

If the latest version of a given package is already installed, an empty string will be returned for that package.

CLI Example:

```
salt '*' pkg.latest_version <package name>
salt '*' pkg.latest_version <package name>
salt '*' pkg.latest_version <package1> <package2> <package3> ...
```

`salt.modules.apkpkg.`**`list_pkgs`**(*versions_as_list=False*, ***kwargs*)

List the packages currently installed in a dict:

```
{'<package_name>': '<version>'}
```

CLI Example:

```
salt '*' pkg.list_pkgs
salt '*' pkg.list_pkgs versions_as_list=True
```

`salt.modules.apkpkg.`**`list_upgrades`**(*refresh=True*)

List all available package upgrades.

CLI Example:

```
salt '*' pkg.list_upgrades
```

`salt.modules.apkpkg.`**`owner`**(**paths*)

Return the name of the package that owns the file. Multiple file paths can be passed. Like `pkg.version <salt.modules.apk.version`, if a single path is passed, a string will be returned, and if multiple paths are passed, a dictionary of file/package name pairs will be returned.

If the file is not owned by a package, or is not present on the minion, then an empty string will be returned for that path.

CLI Example:

salt '*' pkg.owns /usr/bin/apachectl salt '*' pkg.owns /usr/bin/apachectl /usr/bin/basename

`salt.modules.apkpkg.`**`purge`**`(`*name=None*, *pkgs=None*, ***kwargs*`)`
Alias to remove

`salt.modules.apkpkg.`**`refresh_db`**`()`
Updates the package list

- `True`: Database updated successfully
- `False`: Problem updating database

CLI Example:

```
salt '*' pkg.refresh_db
```

`salt.modules.apkpkg.`**`remove`**`(`*name=None*, *pkgs=None*, *purge=False*, ***kwargs*`)`
Remove packages using `apk del`.

name The name of the package to be deleted.

Multiple Package Options:

pkgs A list of packages to delete. Must be passed as a python list. The `name` parameter will be ignored if this option is passed.

Returns a dict containing the changes.

CLI Example:

```
salt '*' pkg.remove <package name>
salt '*' pkg.remove <package1>,<package2>,<package3>
salt '*' pkg.remove pkgs='["foo", "bar"]'
```

`salt.modules.apkpkg.`**`upgrade`**`(`*name=None*, *pkgs=None*, *refresh=True*`)`
Upgrades all packages via `apk upgrade` or a specific package if name or pkgs is specified. Name is ignored if pkgs is specified

Returns a dict containing the changes.

{'<package>': {'old': '<old-version>', 'new': '<new-version>'}}

CLI Example:

```
salt '*' pkg.upgrade
```

`salt.modules.apkpkg.`**`version`**`(`**names*, ***kwargs*`)`
Returns a string representing the package version or an empty string if not installed. If more than one package name is specified, a dict of name/version pairs is returned.

CLI Example:

```
salt '*' pkg.version <package name>
salt '*' pkg.version <package1> <package2> <package3> ...
```

21.9.17 salt.modules.aptpkg

Support for APT (Advanced Packaging Tool)

Important: If you feel that Salt should be using this module to manage packages on a minion, and it is using a different module (or gives an error similar to *'pkg.install' is not available*), see *here*.

For repository management, the `python-apt` package must be installed.

`salt.modules.aptpkg.`**`add_repo_key`**(*path=None, text=None, keyserver=None, keyid=None, saltenv='base'*)

New in version 2017.7.0.

Add a repo key using `apt-key add`.

Parameters

- **`path`** (*`str`*) – The path of the key file to import.
- **`text`** (*`str`*) – The key data to import, in string form.
- **`keyserver`** (*`str`*) – The server to download the repo key specified by the keyid.
- **`keyid`** (*`str`*) – The key id of the repo key to add.
- **`saltenv`** (*`str`*) – The environment the key file resides in.

Returns A boolean representing whether the repo key was added.

Return type bool

CLI Examples:

```
salt '*' pkg.add_repo_key 'salt://apt/sources/test.key'

salt '*' pkg.add_repo_key text="'$KEY1'"

salt '*' pkg.add_repo_key keyserver='keyserver.example' keyid='0000AAAA'
```

`salt.modules.aptpkg.`**`autoremove`**(*list_only=False, purge=False*)

New in version 2015.5.0.

Remove packages not required by another package using `apt-get autoremove`.

list_only [False] Only retrieve the list of packages to be auto-removed, do not actually perform the auto-removal.

purge [False] Also remove package config data when autoremoving packages.

New in version 2015.8.0.

CLI Example:

```
salt '*' pkg.autoremove
salt '*' pkg.autoremove list_only=True
salt '*' pkg.autoremove purge=True
```

`salt.modules.aptpkg.`**`del_repo`**(*repo, **kwargs*)

Delete a repo from the sources.list / sources.list.d

If the .list file is in the sources.list.d directory and the file that the repo exists in does not contain any other repo configuration, the file itself will be deleted.

The repo passed in must be a fully formed repository definition string.

CLI Examples:

```
salt '*' pkg.del_repo "myrepo definition"
```

`salt.modules.aptpkg.`**`del_repo_key`**(*name=None*, ***kwargs*)

New in version 2015.8.0.

Remove a repo key using `apt-key del`

name Repo from which to remove the key. Unnecessary if `keyid` is passed.

keyid The KeyID of the GPG key to remove

keyid_ppa [False] If set to `True`, the repo's GPG key ID will be looked up from ppa.launchpad.net and removed.

> **Note:** Setting this option to `True` requires that the `name` param also be passed.

CLI Examples:

```
salt '*' pkg.del_repo_key keyid=0123ABCD
salt '*' pkg.del_repo_key name='ppa:foo/bar' keyid_ppa=True
```

`salt.modules.aptpkg.`**`expand_repo_def`**(***kwargs*)

Take a repository definition and expand it to the full pkg repository dict that can be used for comparison. This is a helper function to make the Debian/Ubuntu apt sources sane for comparison in the pkgrepo states.

This is designed to be called from pkgrepo states and will have little use being called on the CLI.

`salt.modules.aptpkg.`**`file_dict`**(**packages*)

List the files that belong to a package, grouped by package. Not specifying any packages will return a list of _every_ file on the system's package database (not generally recommended).

CLI Examples:

```
salt '*' pkg.file_dict httpd
salt '*' pkg.file_dict httpd postfix
salt '*' pkg.file_dict
```

`salt.modules.aptpkg.`**`file_list`**(**packages*)

List the files that belong to a package. Not specifying any packages will return a list of _every_ file on the system's package database (not generally recommended).

CLI Examples:

```
salt '*' pkg.file_list httpd
salt '*' pkg.file_list httpd postfix
salt '*' pkg.file_list
```

`salt.modules.aptpkg.`**`get_repo`**(*repo*, ***kwargs*)

Display a repo from the sources.list / sources.list.d

The repo passed in needs to be a complete repo entry.

CLI Examples:

```
salt '*' pkg.get_repo "myrepo definition"
```

`salt.modules.aptpkg.`**`get_repo_keys`**`()`

New in version 2017.7.0.

List known repo key details.

Returns A dictionary containing the repo keys.

Return type dict

CLI Examples:

```
salt '*' pkg.get_repo_keys
```

`salt.modules.aptpkg.`**`get_selections`**`(`*pattern=None, state=None*`)`

View package state from the dpkg database.

Returns a dict of dicts containing the state, and package names:

```
{'<host>':
    {'<state>': ['pkg1',
                 ...
                ]
    },
    ...
}
```

CLI Example:

```
salt '*' pkg.get_selections
salt '*' pkg.get_selections 'python-*'
salt '*' pkg.get_selections state=hold
salt '*' pkg.get_selections 'openssh*' state=hold
```

`salt.modules.aptpkg.`**`hold`**`(`*name=None, pkgs=None, sources=None, **kwargs*`)`

New in version 2014.7.0.

Set package in 'hold' state, meaning it will not be upgraded.

name The name of the package, e.g., 'tmux'

CLI Example:

```
salt '*' pkg.hold <package name>
```

pkgs A list of packages to hold. Must be passed as a python list.

CLI Example:

```
salt '*' pkg.hold pkgs='["foo", "bar"]'
```

`salt.modules.aptpkg.`**`info_installed`**`(`**names, **kwargs*`)`

Return the information of the named package(s) installed on the system.

New in version 2015.8.1.

names The names of the packages for which to return information.

failhard Whether to throw an exception if none of the packages are installed. Defaults to True.

New in version 2016.11.3.

CLI example:

```
salt '*' pkg.info_installed <package1>
salt '*' pkg.info_installed <package1> <package2> <package3> ...
salt '*' pkg.info_installed <package1> failhard=false
```

`salt.modules.aptpkg.install`(*name=None, refresh=False, fromrepo=None, skip_verify=False, debconf=None, pkgs=None, sources=None, reinstall=False, downloadonly=False, ignore_epoch=False, **kwargs*)

Changed in version 2015.8.12,2016.3.3,2016.11.0: On minions running systemd>=205, systemd-run(1) is now used to isolate commands which modify installed packages from the `salt-minion` daemon's control group. This is done to keep systemd from killing any apt-get/dpkg commands spawned by Salt when the `salt-minion` service is restarted. (see `KillMode` in the systemd.kill(5) manpage for more information). If desired, usage of systemd-run(1) can be suppressed by setting a *config option* called `systemd.scope`, with a value of `False` (no quotes).

Install the passed package, add refresh=True to update the dpkg database.

name The name of the package to be installed. Note that this parameter is ignored if either "pkgs" or "sources" is passed. Additionally, please note that this option can only be used to install packages from a software repository. To install a package file manually, use the "sources" option.

32-bit packages can be installed on 64-bit systems by appending the architecture designation (`:i386`, etc.) to the end of the package name.

CLI Example:

```
salt '*' pkg.install <package name>
```

refresh Whether or not to refresh the package database before installing.

cache_valid_time

New in version 2016.11.0.

Skip refreshing the package database if refresh has already occurred within <value> seconds

fromrepo Specify a package repository to install from (e.g., `apt-get -t unstable install somepackage`)

skip_verify Skip the GPG verification check (e.g., `--allow-unauthenticated`, or `--force-bad-verify` for install from package file).

debconf Provide the path to a debconf answers file, processed before installation.

version Install a specific version of the package, e.g. 1.2.3~0ubuntu0. Ignored if "pkgs" or "sources" is passed.

Changed in version 2018.3.0: version can now contain comparison operators (e.g. `>1.2.3`, `<=2.0`, etc.)

reinstall [False] Specifying reinstall=True will use `apt-get install --reinstall` rather than simply `apt-get install` for requested packages that are already installed.

If a version is specified with the requested package, then `apt-get install --reinstall` will only be used if the installed version matches the requested version.

New in version 2015.8.0.

ignore_epoch [False] Only used when the version of a package is specified using a comparison operator (e.g. `>4.1`). If set to `True`, then the epoch will be ignored when comparing the currently-installed version to the desired version.

New in version 2018.3.0.

Multiple Package Installation Options:

pkgs A list of packages to install from a software repository. Must be passed as a python list.

CLI Example:

```
salt '*' pkg.install pkgs='["foo", "bar"]'
salt '*' pkg.install pkgs='["foo", {"bar": "1.2.3-0ubuntu0"}]'
```

sources A list of DEB packages to install. Must be passed as a list of dicts, with the keys being package names, and the values being the source URI or local path to the package. Dependencies are automatically resolved and marked as auto-installed.

32-bit packages can be installed on 64-bit systems by appending the architecture designation (`:i386`, etc.) to the end of the package name.

Changed in version 2014.7.0.

CLI Example:

```
salt '*' pkg.install sources='[{"foo": "salt://foo.deb"},{"bar": "salt://bar.
↪deb"}]'
```

force_yes Passes `--force-yes` to the apt-get command. Don't use this unless you know what you're doing.

New in version 0.17.4.

install_recommends Whether to install the packages marked as recommended. Default is True.

New in version 2015.5.0.

only_upgrade Only upgrade the packages, if they are already installed. Default is False.

New in version 2015.5.0.

force_conf_new Always install the new version of any configuration files.

New in version 2015.8.0.

Returns a dict containing the new package names and versions:

```
{'<package>': {'old': '<old-version>',
               'new': '<new-version>'}}
```

`salt.modules.aptpkg.`**`latest_version`**(**names*, ***kwargs*)

Return the latest version of the named package available for upgrade or installation. If more than one package name is specified, a dict of name/version pairs is returned.

If the latest version of a given package is already installed, an empty string will be returned for that package.

A specific repo can be requested using the `fromrepo` keyword argument.

cache_valid_time

New in version 2016.11.0.

Skip refreshing the package database if refresh has already occurred within <value> seconds

CLI Example:

```
salt '*' pkg.latest_version <package name>
salt '*' pkg.latest_version <package name> fromrepo=unstable
salt '*' pkg.latest_version <package1> <package2> <package3> ...
```

`salt.modules.aptpkg.`**`list_downloaded`**(*root=None*, ***kwargs*)

New in version 3000?.

List prefetched packages downloaded by apt in the local disk.

root operate on a different root directory.

CLI example:

```
salt '*' pkg.list_downloaded
```

`salt.modules.aptpkg.`**`list_pkgs`**`(`*versions_as_list=False, removed=False, purge_desired=False, **kwargs*`)`

List the packages currently installed in a dict:

```
{'<package_name>': '<version>'}
```

removed If `True`, then only packages which have been removed (but not purged) will be returned.

purge_desired If `True`, then only packages which have been marked to be purged, but can't be purged due to their status as dependencies for other installed packages, will be returned. Note that these packages will appear in installed

Changed in version 2014.1.1: Packages in this state now correctly show up in the output of this function.

CLI Example:

```
salt '*' pkg.list_pkgs
salt '*' pkg.list_pkgs versions_as_list=True
```

`salt.modules.aptpkg.`**`list_repo_pkgs`**`(`**args, **kwargs*`)`

New in version 2017.7.0.

Returns all available packages. Optionally, package names (and name globs) can be passed and the results will be filtered to packages matching those names.

This function can be helpful in discovering the version or repo to specify in a *`pkg.installed`* state.

The return data will be a dictionary mapping package names to a list of version numbers, ordered from newest to oldest. For example:

```
{
    'bash': ['4.3-14ubuntu1.1',
             '4.3-14ubuntu1'],
    'nginx': ['1.10.0-0ubuntu0.16.04.4',
              '1.9.15-0ubuntu1']
}
```

CLI Examples:

```
salt '*' pkg.list_repo_pkgs
salt '*' pkg.list_repo_pkgs foo bar baz
```

`salt.modules.aptpkg.`**`list_repos`**`()`

Lists all repos in the sources.list (and sources.lists.d) files

CLI Example:

```
salt '*' pkg.list_repos
salt '*' pkg.list_repos disabled=True
```

`salt.modules.aptpkg.`**`list_upgrades`**`(`*refresh=True, dist_upgrade=True, **kwargs*`)`

List all available package upgrades.

refresh Whether to refresh the package database before listing upgrades. Default: True.

cache_valid_time

New in version 2016.11.0.

Skip refreshing the package database if refresh has already occurred within <value> seconds

dist_upgrade Whether to list the upgrades using dist-upgrade vs upgrade. Default is to use dist-upgrade.

CLI Example:

```
salt '*' pkg.list_upgrades
```

`salt.modules.aptpkg.`**`mod_repo`**(*repo*, *saltenv='base'*, ***kwargs*)

Modify one or more values for a repo. If the repo does not exist, it will be created, so long as the definition is well formed. For Ubuntu the `ppa:<project>/repo` format is acceptable. `ppa:` format can only be used to create a new repository.

The following options are available to modify a repo definition:

architectures A comma-separated list of supported architectures, e.g. `amd64` If this option is not set, all architectures (configured in the system) will be used.

comps A comma separated list of components for the repo, e.g. `main`

file A file name to be used

keyserver Keyserver to get gpg key from

keyid Key ID or a list of key IDs to load with the `keyserver` argument

key_url URL to a GPG key to add to the APT GPG keyring

key_text GPG key in string form to add to the APT GPG keyring

New in version 2018.3.0.

consolidate [False] If `True`, will attempt to de-duplicate and consolidate sources

comments Sometimes you want to supply additional information, but not as enabled configuration. All comments provided here will be joined into a single string and appended to the repo configuration with a comment marker (#) before it.

New in version 2015.8.9.

refresh [True] Enable or disable (True or False) refreshing of the apt package database. The previous `refresh_db` argument was deprecated in favor of `refresh``. The `refresh_db` argument will still continue to work to ensure backwards compatibility, but please change to using the preferred `refresh`.

Note: Due to the way keys are stored for APT, there is a known issue where the key won't be updated unless another change is made at the same time. Keys should be properly added on initial configuration.

CLI Examples:

```
salt '*' pkg.mod_repo 'myrepo definition' uri=http://new/uri
salt '*' pkg.mod_repo 'myrepo definition' comps=main,universe
```

`salt.modules.aptpkg.`**`normalize_name`**(*name*)

Strips the architecture from the specified package name, if necessary.

CLI Example:

```
salt '*' pkg.normalize_name zsh:amd64
```

salt.modules.aptpkg.**owner**(**paths*)

New in version 2014.7.0.

Return the name of the package that owns the file. Multiple file paths can be passed. Like `pkg.version`, if a single path is passed, a string will be returned, and if multiple paths are passed, a dictionary of file/package name pairs will be returned.

If the file is not owned by a package, or is not present on the minion, then an empty string will be returned for that path.

CLI Example:

```
salt '*' pkg.owner /usr/bin/apachectl
salt '*' pkg.owner /usr/bin/apachectl /usr/bin/basename
```

salt.modules.aptpkg.**parse_arch**(*name*)

Parse name and architecture from the specified package name.

CLI Example:

```
salt '*' pkg.parse_arch zsh:amd64
```

salt.modules.aptpkg.**purge**(*name=None*, *pkgs=None*, ***kwargs*)

Changed in version 2015.8.12,2016.3.3,2016.11.0: On minions running systemd>=205, systemd-run(1) is now used to isolate commands which modify installed packages from the `salt-minion` daemon's control group. This is done to keep systemd from killing any apt-get/dpkg commands spawned by Salt when the `salt-minion` service is restarted. (see `KillMode` in the systemd.kill(5) manpage for more information). If desired, usage of systemd-run(1) can be suppressed by setting a `config option` called `systemd.scope`, with a value of `False` (no quotes).

Remove packages via `apt-get purge` along with all configuration files.

name The name of the package to be deleted.

Multiple Package Options:

pkgs A list of packages to delete. Must be passed as a python list. The `name` parameter will be ignored if this option is passed.

New in version 0.16.0.

Returns a dict containing the changes.

CLI Example:

```
salt '*' pkg.purge <package name>
salt '*' pkg.purge <package1>,<package2>,<package3>
salt '*' pkg.purge pkgs='["foo", "bar"]'
```

salt.modules.aptpkg.**refresh_db**(*cache_valid_time=0*, *failhard=False*)

Updates the APT database to latest packages based upon repositories

Returns a dict, with the keys being package databases and the values being the result of the update attempt. Values can be one of the following:

- `True`: Database updated successfully
- `False`: Problem updating database
- `None`: Database already up-to-date

cache_valid_time

New in version 2016.11.0.

Skip refreshing the package database if refresh has already occurred within <value> seconds

failhard

If False, return results of Err lines as `False` for the package database that encountered the error. If True, raise an error with a list of the package databases that encountered errors.

CLI Example:

```
salt '*' pkg.refresh_db
```

`salt.modules.aptpkg.`**`remove`**(*name=None*, *pkgs=None*, ***kwargs*)

Changed in version 2015.8.12,2016.3.3,2016.11.0: On minions running systemd>=205, systemd-run(1) is now used to isolate commands which modify installed packages from the `salt-minion` daemon's control group. This is done to keep systemd from killing any apt-get/dpkg commands spawned by Salt when the `salt-minion` service is restarted. (see `KillMode` in the systemd.kill(5) manpage for more information). If desired, usage of systemd-run(1) can be suppressed by setting a *`config option`* called `systemd.scope`, with a value of `False` (no quotes).

Remove packages using `apt-get remove`.

name The name of the package to be deleted.

Multiple Package Options:

pkgs A list of packages to delete. Must be passed as a python list. The `name` parameter will be ignored if this option is passed.

New in version 0.16.0.

Returns a dict containing the changes.

CLI Example:

```
salt '*' pkg.remove <package name>
salt '*' pkg.remove <package1>,<package2>,<package3>
salt '*' pkg.remove pkgs='["foo", "bar"]'
```

`salt.modules.aptpkg.`**`set_selections`**(*path=None*, *selection=None*, *clear=False*, *saltenv='base'*)

Change package state in the dpkg database.

The state can be any one of, documented in `dpkg(1)`:

- install
- hold
- deinstall
- purge

This command is commonly used to mark specific packages to be held from being upgraded, that is, to be kept at a certain version. When a state is changed to anything but being held, then it is typically followed by `apt-get -u dselect-upgrade`.

Note: Be careful with the `clear` argument, since it will start with setting all packages to deinstall state.

Returns a dict of dicts containing the package names, and the new and old versions:

```
{'<host>':
    {'<package>': {'new': '<new-state>',
                   'old': '<old-state>'}
    },
    ...
}
```

CLI Example:

```
salt '*' pkg.set_selections selection='{"install": ["netcat"]}'
salt '*' pkg.set_selections selection='{"hold": ["openssh-server", "openssh-client
↪"]}'
salt '*' pkg.set_selections salt://path/to/file
salt '*' pkg.set_selections salt://path/to/file clear=True
```

`salt.modules.aptpkg.`**`show`**`(`**names, **kwargs*`)`

New in version 2019.2.0.

Runs an `apt-cache show` on the passed package names, and returns the results in a nested dictionary. The top level of the return data will be the package name, with each package name mapping to a dictionary of version numbers to any additional information returned by `apt-cache show`.

filter An optional comma-separated list (or quoted Python list) of case-insensitive keys on which to filter. This allows one to restrict the information returned for each package to a smaller selection of pertinent items.

refresh [False] If `True`, the apt cache will be refreshed first. By default, no refresh is performed.

CLI Examples:

```
salt myminion pkg.show gawk
salt myminion pkg.show 'nginx-*'
salt myminion pkg.show 'nginx-*' filter=description,provides
```

`salt.modules.aptpkg.`**`unhold`**`(`*name=None, pkgs=None, sources=None, **kwargs*`)`

New in version 2014.7.0.

Set package current in 'hold' state to install state, meaning it will be upgraded.

name The name of the package, e.g., 'tmux'

CLI Example:

```
salt '*' pkg.unhold <package name>
```

pkgs A list of packages to unhold. Must be passed as a python list.

CLI Example:

```
salt '*' pkg.unhold pkgs='["foo", "bar"]'
```

`salt.modules.aptpkg.`**`upgrade`**`(`*refresh=True, dist_upgrade=False, **kwargs*`)`

Changed in version 2015.8.12,2016.3.3,2016.11.0: On minions running systemd>=205, systemd-run(1) is now used to isolate commands which modify installed packages from the `salt-minion` daemon's control group. This is done to keep systemd from killing any apt-get/dpkg commands spawned by Salt when the `salt-minion` service is restarted. (see `KillMode` in the systemd.kill(5) manpage for more information). If desired, usage of systemd-run(1) can be suppressed by setting a *`config option`* called `systemd.scope`, with a value of `False` (no quotes).

Upgrades all packages via `apt-get upgrade` or `apt-get dist-upgrade` if `dist_upgrade` is `True`.

Returns a dictionary containing the changes:

```
{'<package>':  {'old': '<old-version>',
                'new': '<new-version>'}}
```

dist_upgrade Whether to perform the upgrade using dist-upgrade vs upgrade. Default is to use upgrade.

New in version 2014.7.0.

cache_valid_time

New in version 2016.11.0.

Skip refreshing the package database if refresh has already occurred within <value> seconds

download_only (or downloadonly) Only download the packages, don't unpack or install them. Use downloadonly to be in line with yum and zypper module.

New in version 2018.3.0.

force_conf_new Always install the new version of any configuration files.

New in version 2015.8.0.

CLI Example:

```
salt '*' pkg.upgrade
```

`salt.modules.aptpkg.`**`upgrade_available`**`(`*name*`)`

Check whether or not an upgrade is available for a given package

CLI Example:

```
salt '*' pkg.upgrade_available <package name>
```

`salt.modules.aptpkg.`**`version`**`(`**names, **kwargs*`)`

Returns a string representing the package version or an empty string if not installed. If more than one package name is specified, a dict of name/version pairs is returned.

CLI Example:

```
salt '*' pkg.version <package name>
salt '*' pkg.version <package1> <package2> <package3> ...
```

`salt.modules.aptpkg.`**`version_cmp`**`(`*pkg1, pkg2, ignore_epoch=False*`)`

Do a cmp-style comparison on two packages. Return -1 if pkg1 < pkg2, 0 if pkg1 == pkg2, and 1 if pkg1 > pkg2. Return None if there was a problem making the comparison.

ignore_epoch [False] Set to `True` to ignore the epoch when comparing versions

New in version 2015.8.10,2016.3.2.

CLI Example:

```
salt '*' pkg.version_cmp '0.2.4-0ubuntu1' '0.2.4.1-0ubuntu1'
```

21.9.18 salt.modules.archive

A module to wrap (non-Windows) archive calls

New in version 2014.1.0.

`salt.modules.archive.`**`cmd_unzip`**(*zip_file*, *dest*, *excludes=None*, *options=None*, *template=None*, *runas=None*, *trim_output=False*, *password=None*)

New in version 2015.5.0: In versions 2014.7.x and earlier, this function was known as `archive.unzip`.

Uses the `unzip` command to unpack zip files. This command is part of the Info-ZIP suite of tools, and is typically packaged as simply `unzip`.

zip_file Path of zip file to be unpacked

dest The destination directory into which the file should be unpacked

excludes [None] Comma-separated list of files not to unpack. Can also be passed in a Python list.

template [None] Can be set to 'jinja' or another supported template engine to render the command arguments before execution:

```
salt '*' archive.cmd_unzip template=jinja /tmp/zipfile.zip '/tmp/{{grains.id}}
↪' excludes=file_1,file_2
```

options Optional when using `zip` archives, ignored when usign other archives files. This is mostly used to overwrite existing files with `o`. This options are only used when `unzip` binary is used.

New in version 2016.3.1.

runas [None] Unpack the zip file as the specified user. Defaults to the user under which the minion is running.

New in version 2015.5.0.

trim_output [False] The number of files we should output on success before the rest are trimmed, if this is set to True then it will default to 100

password Password to use with password protected zip files

> **Note:** This is not considered secure. It is recommended to instead use *`archive.unzip`* for password-protected ZIP files. If a password is used here, then the unzip command run to extract the ZIP file will not show up in the minion log like most shell commands Salt runs do. However, the password will still be present in the events logged to the minion log at the `debug` log level. If the minion is logging at `debug` (or more verbose), then be advised that the password will appear in the log.

New in version 2016.11.0.

CLI Example:

```
salt '*' archive.cmd_unzip /tmp/zipfile.zip /home/strongbad/ excludes=file_1,file_
↪2
```

`salt.modules.archive.`**`cmd_zip`**(*zip_file*, *sources*, *template=None*, *cwd=None*, *runas=None*)

New in version 2015.5.0: In versions 2014.7.x and earlier, this function was known as `archive.zip`.

Uses the `zip` command to create zip files. This command is part of the Info-ZIP suite of tools, and is typically packaged as simply `zip`.

zip_file Path of zip file to be created

sources Comma-separated list of sources to include in the zip file. Sources can also be passed in a Python list.

Changed in version 2017.7.0: Globbing is now supported for this argument

template [None] Can be set to 'jinja' or another supported template engine to render the command arguments before execution:

```
salt '*' archive.cmd_zip template=jinja /tmp/zipfile.zip /tmp/sourcefile1,/
↪tmp/{{grains.id}}.txt
```

cwd [None] Use this argument along with relative paths in `sources` to create zip files which do not contain the leading directories. If not specified, the zip file will be created as if the cwd was `/`, and creating a zip file of `/foo/bar/baz.txt` will contain the parent directories `foo` and `bar`. To create a zip file containing just `baz.txt`, the following command would be used:

```
salt '*' archive.cmd_zip /tmp/baz.zip baz.txt cwd=/foo/bar
```

New in version 2014.7.1.

runas [None] Create the zip file as the specified user. Defaults to the user under which the minion is running.

New in version 2015.5.0.

CLI Example:

```
salt '*' archive.cmd_zip /tmp/zipfile.zip /tmp/sourcefile1,/tmp/sourcefile2
# Globbing for sources (2017.7.0 and later)
salt '*' archive.cmd_zip /tmp/zipfile.zip '/tmp/sourcefile*'
```

`salt.modules.archive.`**`gunzip`**(*gzipfile*, *template=None*, *runas=None*, *options=None*)

Uses the gunzip command to unpack gzip files

template [None] Can be set to 'jinja' or another supported template engine to render the command arguments before execution:

```
salt '*' archive.gunzip template=jinja /tmp/{{grains.id}}.txt.gz
```

runas [None] The user with which to run the gzip command line

options [None] Pass any additional arguments to gzip

New in version 2016.3.4.

CLI Example:

```
# Create /tmp/sourcefile.txt
salt '*' archive.gunzip /tmp/sourcefile.txt.gz
salt '*' archive.gunzip /tmp/sourcefile.txt options='--verbose'
```

`salt.modules.archive.`**`gzip`**(*sourcefile*, *template=None*, *runas=None*, *options=None*)

Uses the gzip command to create gzip files

template [None] Can be set to 'jinja' or another supported template engine to render the command arguments before execution:

```
salt '*' archive.gzip template=jinja /tmp/{{grains.id}}.txt
```

runas [None] The user with which to run the gzip command line

options [None] Pass any additional arguments to gzip

New in version 2016.3.4.

CLI Example:

```
# Create /tmp/sourcefile.txt.gz
salt '*' archive.gzip /tmp/sourcefile.txt
salt '*' archive.gzip /tmp/sourcefile.txt options='-9 --verbose'
```

`salt.modules.archive.`**`is_encrypted`**`(`*name*, *clean=False*, *saltenv='base'*, *source_hash=None*`)`

New in version 2016.11.0.

Returns `True` if the zip archive is password-protected, `False` if not. If the specified file is not a ZIP archive, an error will be raised.

name The path / URL of the archive to check.

clean [False] Set this value to `True` to delete the path referred to by `name` once the contents have been listed. This option should be used with care.

> **Note:** If there is an error listing the archive's contents, the cached file will not be removed, to allow for troubleshooting.

saltenv [base] Specifies the fileserver environment from which to retrieve `archive`. This is only applicable when `archive` is a file from the `salt://` fileserver.

source_hash If `name` is an http(s)/ftp URL and the file exists in the minion's file cache, this option can be passed to keep the minion from re-downloading the archive if the cached copy matches the specified hash.

New in version 2018.3.0.

CLI Examples:

```
salt '*' archive.is_encrypted /path/to/myfile.zip
salt '*' archive.is_encrypted salt://foo.zip
salt '*' archive.is_encrypted salt://foo.zip saltenv=dev
salt '*' archive.is_encrypted https://domain.tld/myfile.zip clean=True
salt '*' archive.is_encrypted https://domain.tld/myfile.zip source_
↪hash=f1d2d2f924e986ac86fdf7b36c94bcdf32beec15
salt '*' archive.is_encrypted ftp://10.1.2.3/foo.zip
```

`salt.modules.archive.`**`list_`**`(`*name*, *archive_format=None*, *options=None*, *strip_components=None*, *clean=False*, *verbose=False*, *saltenv='base'*, *source_hash=None*`)`

New in version 2016.11.0.

Changed in version 2016.11.2: The rarfile Python module is now supported for listing the contents of rar archives. This is necessary on minions with older releases of the `rar` CLI tool, which do not support listing the contents in a parsable format.

List the files and directories in an tar, zip, or rar archive.

> **Note:** This function will only provide results for XZ-compressed archives if the xz CLI command is available, as Python does not at this time natively support XZ compression in its tarfile module. Keep in mind however that most Linux distros ship with xz already installed.
>
> To check if a given minion has xz, the following Salt command can be run:
>
> ```
> salt minion_id cmd.which xz
> ```
>
> If `None` is returned, then xz is not present and must be installed. It is widely available and should be packaged as either `xz` or `xz-utils`.

name Path/URL of archive

archive_format Specify the format of the archive (`tar`, `zip`, or `rar`). If this argument is omitted, the archive format will be guessed based on the value of the `name` parameter.

options **For tar archives only**. This function will, by default, try to use the tarfile module from the Python standard library to get a list of files/directories. If this method fails, then it will fall back to using the shell to decompress the archive to stdout and pipe the results to `tar -tf -` to produce a list of filenames. XZ-compressed archives are already supported automatically, but in the event that the tar archive uses a different sort of compression not supported natively by tarfile, this option can be used to specify a command that will decompress the archive to stdout. For example:

```
salt minion_id archive.list /path/to/foo.tar.gz options='gzip --decompress --
↪stdout'
```

Note: It is not necessary to manually specify options for gzip'ed archives, as gzip compression is natively supported by tarfile.

strip_components This argument specifies a number of top-level directories to strip from the results. This is similar to the paths that would be extracted if `--strip-components` (or `--strip`) were used when extracting tar archives.

New in version 2016.11.2.

clean [False] Set this value to `True` to delete the path referred to by `name` once the contents have been listed. This option should be used with care.

Note: If there is an error listing the archive's contents, the cached file will not be removed, to allow for troubleshooting.

verbose [False] If `False`, this function will return a list of files/dirs in the archive. If `True`, it will return a dictionary categorizing the paths into separate keys containing the directory names, file names, and also directories/files present in the top level of the archive.

Changed in version 2016.11.2: This option now includes symlinks in their own list. Before, they were included with files.

saltenv [base] Specifies the fileserver environment from which to retrieve `archive`. This is only applicable when `archive` is a file from the `salt://` fileserver.

source_hash If `name` is an http(s)/ftp URL and the file exists in the minion's file cache, this option can be passed to keep the minion from re-downloading the archive if the cached copy matches the specified hash.

New in version 2018.3.0.

CLI Examples:

```
salt '*' archive.list /path/to/myfile.tar.gz
salt '*' archive.list /path/to/myfile.tar.gz strip_components=1
salt '*' archive.list salt://foo.tar.gz
salt '*' archive.list https://domain.tld/myfile.zip
salt '*' archive.list https://domain.tld/myfile.zip source_
↪hash=f1d2d2f924e986ac86fdf7b36c94bcdf32beec15
salt '*' archive.list ftp://10.1.2.3/foo.rar
```

`salt.modules.archive.`**`rar`**(*rarfile, sources, template=None, cwd=None, runas=None*)

Uses rar for Linux to create rar files

rarfile Path of rar file to be created

sources Comma-separated list of sources to include in the rar file. Sources can also be passed in a Python list.

Changed in version 2017.7.0: Globbing is now supported for this argument

cwd [None] Run the rar command from the specified directory. Use this argument along with relative file paths to create rar files which do not contain the leading directories. If not specified, this will default to the home directory of the user under which the salt minion process is running.

New in version 2014.7.1.

template [None] Can be set to 'jinja' or another supported template engine to render the command arguments before execution:

```
salt '*' archive.rar template=jinja /tmp/rarfile.rar '/tmp/sourcefile1,/tmp/{
↪{grains.id}}.txt'
```

CLI Example:

```
salt '*' archive.rar /tmp/rarfile.rar /tmp/sourcefile1,/tmp/sourcefile2
# Globbing for sources (2017.7.0 and later)
salt '*' archive.rar /tmp/rarfile.rar '/tmp/sourcefile*'
```

`salt.modules.archive.`**`tar`**(*options*, *tarfile*, *sources=None*, *dest=None*, *cwd=None*, *template=None*, *runas=None*)

Note: This function has changed for version 0.17.0. In prior versions, the `cwd` and `template` arguments must be specified, with the source directories/files coming as a space-separated list at the end of the command. Beginning with 0.17.0, `sources` must be a comma-separated list, and the `cwd` and `template` arguments are optional.

Uses the tar command to pack, unpack, etc. tar files

options Options to pass to the tar command

Changed in version 2015.8.0: The mandatory - prefixing has been removed. An options string beginning with a *–long-option*, would have uncharacteristically needed its first - removed under the former scheme.

Also, tar will parse its options differently if short options are used with or without a preceding -, so it is better to not confuse the user into thinking they're using the non-- format, when really they are using the with-- format.

tarfile The filename of the tar archive to pack/unpack

sources Comma delimited list of files to **pack** into the tarfile. Can also be passed as a Python list.

Changed in version 2017.7.0: Globbing is now supported for this argument

dest The destination directory into which to **unpack** the tarfile

cwd [None] The directory in which the tar command should be executed. If not specified, will default to the home directory of the user under which the salt minion process is running.

template [None] Can be set to 'jinja' or another supported template engine to render the command arguments before execution:

```
salt '*' archive.tar cjvf /tmp/salt.tar.bz2 {{grains.saltpath}} template=jinja
```

CLI Examples:

```
# Create a tarfile
salt '*' archive.tar cjvf /tmp/tarfile.tar.bz2 /tmp/file_1,/tmp/file_2
# Create a tarfile using globbing (2017.7.0 and later)
salt '*' archive.tar cjvf /tmp/tarfile.tar.bz2 '/tmp/file_*'
# Unpack a tarfile
salt '*' archive.tar xf foo.tar dest=/target/directory
```

`salt.modules.archive.`**`unrar`**(*rarfile*, *dest*, *excludes=None*, *template=None*, *runas=None*, *trim_output=False*)

Uses rar for Linux to unpack rar files

rarfile Name of rar file to be unpacked

dest The destination directory into which to **unpack** the rar file

template [None] Can be set to 'jinja' or another supported template engine to render the command arguments before execution:

```
salt '*' archive.unrar template=jinja /tmp/rarfile.rar /tmp/{{grains.id}}/↩
↪excludes=file_1,file_2
```

trim_output [False] The number of files we should output on success before the rest are trimmed, if this is set to True then it will default to 100

CLI Example:

```
salt '*' archive.unrar /tmp/rarfile.rar /home/strongbad/ excludes=file_1,file_2
```

`salt.modules.archive.`**`unzip`**(*zip_file*, *dest*, *excludes=None*, *options=None*, *template=None*, *runas=None*, *trim_output=False*, *password=None*, *extract_perms=True*)

Uses the `zipfile` Python module to unpack zip files

Changed in version 2015.5.0: This function was rewritten to use Python's native zip file support. The old functionality has been preserved in the new function *`archive.cmd_unzip`*. For versions 2014.7.x and earlier, see the *`archive.cmd_zip`* documentation.

zip_file Path of zip file to be unpacked

dest The destination directory into which the file should be unpacked

excludes [None] Comma-separated list of files not to unpack. Can also be passed in a Python list.

options This options are only used when `unzip` binary is used. In this function is ignored.

New in version 2016.3.1.

template [None] Can be set to 'jinja' or another supported template engine to render the command arguments before execution:

```
salt '*' archive.unzip template=jinja /tmp/zipfile.zip /tmp/{{grains.id}}/↩
↪excludes=file_1,file_2
```

runas [None] Unpack the zip file as the specified user. Defaults to the user under which the minion is running.

trim_output [False] The number of files we should output on success before the rest are trimmed, if this is set to True then it will default to 100

CLI Example:

```
salt '*' archive.unzip /tmp/zipfile.zip /home/strongbad/ excludes=file_1,file_2
```

password Password to use with password protected zip files

Note: The password will be present in the events logged to the minion log file at the `debug` log level. If the minion is logging at `debug` (or more verbose), then be advised that the password will appear in the log.

New in version 2016.3.0.

extract_perms [True] The Python zipfile module does not extract file/directory attributes by default. When this argument is set to `True`, Salt will attempt to apply the file permission attributes to the extracted files/folders.

On Windows, only the read-only flag will be extracted as set within the zip file, other attributes (i.e. user/group permissions) are ignored.

Set this argument to `False` to disable this behavior.

New in version 2016.11.0.

CLI Example:

```
salt '*' archive.unzip /tmp/zipfile.zip /home/strongbad/ password='BadPassword'
```

`salt.modules.archive.`**`zip_`**(*zip_file*, *sources*, *template=None*, *cwd=None*, *runas=None*, *zip64=False*)

Uses the `zipfile` Python module to create zip files

Changed in version 2015.5.0: This function was rewritten to use Python's native zip file support. The old functionality has been preserved in the new function *`archive.cmd_zip`*. For versions 2014.7.x and earlier, see the *`archive.cmd_zip`* documentation.

zip_file Path of zip file to be created

sources Comma-separated list of sources to include in the zip file. Sources can also be passed in a Python list.

Changed in version 2017.7.0: Globbing is now supported for this argument

template [None] Can be set to 'jinja' or another supported template engine to render the command arguments before execution:

```
salt '*' archive.zip template=jinja /tmp/zipfile.zip /tmp/sourcefile1,/tmp/{
↪{grains.id}}.txt
```

cwd [None] Use this argument along with relative paths in `sources` to create zip files which do not contain the leading directories. If not specified, the zip file will be created as if the cwd was `/`, and creating a zip file of `/foo/bar/baz.txt` will contain the parent directories `foo` and `bar`. To create a zip file containing just `baz.txt`, the following command would be used:

```
salt '*' archive.zip /tmp/baz.zip baz.txt cwd=/foo/bar
```

runas [None] Create the zip file as the specified user. Defaults to the user under which the minion is running.

zip64 [False] Used to enable ZIP64 support, necessary to create archives larger than 4 GByte in size. If true, will create ZIP file with the ZIPp64 extension when the zipfile is larger than 2 GB. ZIP64 extension is disabled by default in the Python native zip support because the default zip and unzip commands on Unix (the InfoZIP utilities) don't support these extensions.

CLI Example:

```
salt '*' archive.zip /tmp/zipfile.zip /tmp/sourcefile1,/tmp/sourcefile2
# Globbing for sources (2017.7.0 and later)
salt '*' archive.zip /tmp/zipfile.zip '/tmp/sourcefile*'
```

21.9.19 salt.modules.arista_pyeapi

Arista pyeapi

New in version 2019.2.0.

Execution module to interface the connection with Arista switches, connecting to the remote network device using the pyeapi library. It is flexible enough to execute the commands both when running under an Arista Proxy Minion, as well as running under a Regular Minion by specifying the connection arguments, i.e., `device_type`, `host`, `username`, `password` etc.

codeauthor Mircea Ulinic <ping@mirceaulinic.net>

maturity new

depends pyeapi

platform unix

Note: To understand how to correctly enable the eAPI on your switch, please check https://eos.arista.com/arista-eapi-101/.

Dependencies

The `pyeapi` Execution module requires the Python Client for eAPI (pyeapi) to be installed: `pip install pyeapi`.

Usage

This module can equally be used via the *`pyeapi`* Proxy module or directly from an arbitrary (Proxy) Minion that is running on a machine having access to the network device API, and the `pyeapi` library is installed.

When running outside of the *`pyeapi Proxy`* (i.e., from another Proxy Minion type, or regular Minion), the pyeapi connection arguments can be either specified from the CLI when executing the command, or in a configuration block under the `pyeapi` key in the configuration opts (i.e., (Proxy) Minion configuration file), or Pillar. The module supports these simultaneously. These fields are the exact same supported by the `pyeapi` Proxy Module:

transport: `https` Specifies the type of connection transport to use. Valid values for the connection are `socket`, `http_local`, `http`, and `https`.

host: `localhost` The IP address or DNS host name of the connection device.

username: `admin` The username to pass to the device to authenticate the eAPI connection.

password The password to pass to the device to authenticate the eAPI connection.

port The TCP port of the endpoint for the eAPI connection. If this keyword is not specified, the default value is automatically determined by the transport type (`80` for `http`, or `443` for `https`).

enablepwd The enable mode password if required by the destination node.

Example (when not running in a `pyeapi` Proxy Minion):

```
pyeapi:
  username: test
  password: test
```

In case the `username` and `password` are the same on any device you are targeting, the block above (besides other parameters specific to your environment you might need) should suffice to be able to execute commands from outside a `pyeapi` Proxy, e.g.:

```
salt '*' pyeapi.send_commands 'show version' 'show interfaces'
salt '*' pyeapi.config 'ntp server 1.2.3.4'
```

Note: Remember that the above applies only when not running in a `pyeapi` Proxy Minion. If you want to use the *`pyeapi Proxy`*, please follow the documentation notes for a proper setup.

`salt.modules.arista_pyeapi.`**`call`**`(`*method*, **args*, ***kwargs*`)`

Invoke an arbitrary pyeapi method.

method The name of the pyeapi method to invoke.

args A list of arguments to send to the method invoked.

kwargs Key-value dictionary to send to the method invoked.

transport: `https` Specifies the type of connection transport to use. Valid values for the connection are `socket`, `http_local`, `http`, and `https`.

Note: This argument does not need to be specified when running in a *`pyeapi`* Proxy Minion.

host: `localhost` The IP address or DNS host name of the connection device.

Note: This argument does not need to be specified when running in a *`pyeapi`* Proxy Minion.

username: `admin` The username to pass to the device to authenticate the eAPI connection.

Note: This argument does not need to be specified when running in a *`pyeapi`* Proxy Minion.

password The password to pass to the device to authenticate the eAPI connection.

Note: This argument does not need to be specified when running in a *`pyeapi`* Proxy Minion.

port The TCP port of the endpoint for the eAPI connection. If this keyword is not specified, the default value is automatically determined by the transport type (`80` for `http`, or `443` for `https`).

Note: This argument does not need to be specified when running in a *`pyeapi`* Proxy Minion.

enablepwd The enable mode password if required by the destination node.

Note: This argument does not need to be specified when running in a *pyeapi* Proxy Minion.

CLI Example:

```
salt '*' pyeapi.call run_commands "['show version']"
```

`salt.modules.arista_pyeapi.`**`config`**(*commands=None*, *config_file=None*, *template_engine='jinja'*, *context=None*, *defaults=None*, *saltenv='base'*, ***kwargs*)

Configures the node with the specified commands.

This method is used to send configuration commands to the node. It will take either a string or a list and prepend the necessary commands to put the session into config mode.

Returns the diff after the configuration commands are loaded.

config_file The source file with the configuration commands to be sent to the device.

The file can also be a template that can be rendered using the template engine of choice.

This can be specified using the absolute path to the file, or using one of the following URL schemes:

- `salt://`, to fetch the file from the Salt fileserver.
- `http://` or `https://`
- `ftp://`
- `s3://`
- `swift://`

commands The commands to send to the node in config mode. If the commands argument is a string it will be cast to a list. The list of commands will also be prepended with the necessary commands to put the session in config mode.

Note: This argument is ignored when `config_file` is specified.

template_engine: **`jinja`** The template engine to use when rendering the source file. Default: `jinja`. To simply fetch the file without attempting to render, set this argument to `None`.

context Variables to add to the template context.

defaults Default values of the `context` dict.

transport: **`https`** Specifies the type of connection transport to use. Valid values for the connection are `socket`, `http_local`, `http`, and `https`.

Note: This argument does not need to be specified when running in a *pyeapi* Proxy Minion.

host: **`localhost`** The IP address or DNS host name of the connection device.

Note: This argument does not need to be specified when running in a *pyeapi* Proxy Minion.

username: **`admin`** The username to pass to the device to authenticate the eAPI connection.

Note: This argument does not need to be specified when running in a *pyeapi* Proxy Minion.

password The password to pass to the device to authenticate the eAPI connection.

Note: This argument does not need to be specified when running in a *pyeapi* Proxy Minion.

port The TCP port of the endpoint for the eAPI connection. If this keyword is not specified, the default value is automatically determined by the transport type (`80` for `http`, or `443` for `https`).

Note: This argument does not need to be specified when running in a *pyeapi* Proxy Minion.

enablepwd The enable mode password if required by the destination node.

Note: This argument does not need to be specified when running in a *pyeapi* Proxy Minion.

CLI Example:

```
salt '*' pyeapi.config commands="['ntp server 1.2.3.4', 'ntp server 5.6.7.8']"
salt '*' pyeapi.config config_file=salt://config.txt
salt '*' pyeapi.config config_file=https://bit.ly/2LGLcDy context="{'servers': [
↪'1.2.3.4']}"
```

`salt.modules.arista_pyeapi.`**`get_config`**(*config='running-config'*, *params=None*, *as_string=False*, ***kwargs*)

Retrieves the config from the device.

This method will retrieve the config from the node as either a string or a list object. The config to retrieve can be specified as either the startup-config or the running-config.

config: `running-config` Specifies to return either the nodes `startup-config` or `running-config`. The default value is the `running-config`.

params A string of keywords to append to the command for retrieving the config.

as_string: `False` Flag that determines the response. If `True`, then the configuration is returned as a raw string. If `False`, then the configuration is returned as a list. The default value is `False`.

transport: `https` Specifies the type of connection transport to use. Valid values for the connection are `socket`, `http_local`, `http`, and `https`.

Note: This argument does not need to be specified when running in a *pyeapi* Proxy Minion.

host: `localhost` The IP address or DNS host name of the connection device.

Note: This argument does not need to be specified when running in a *pyeapi* Proxy Minion.

username: `admin` The username to pass to the device to authenticate the eAPI connection.

Note: This argument does not need to be specified when running in a *pyeapi* Proxy Minion.

password The password to pass to the device to authenticate the eAPI connection.

> **Note:** This argument does not need to be specified when running in a *pyeapi* Proxy Minion.

port The TCP port of the endpoint for the eAPI connection. If this keyword is not specified, the default value is automatically determined by the transport type (`80` for `http`, or `443` for `https`).

> **Note:** This argument does not need to be specified when running in a *pyeapi* Proxy Minion.

enablepwd The enable mode password if required by the destination node.

> **Note:** This argument does not need to be specified when running in a *pyeapi* Proxy Minion.

CLI Example:

```
salt '*' pyeapi.get_config
salt '*' pyeapi.get_config params='section snmp-server'
salt '*' pyeapi.get_config config='startup-config'
```

`salt.modules.arista_pyeapi.`**`get_connection`**(***kwargs*)

Return the connection object to the pyeapi Node.

> **Warning:** This function returns an unserializable object, hence it is not meant to be used on the CLI. This should mainly be used when invoked from other modules for the low level connection with the network device.

kwargs Key-value dictionary with the authentication details.

USAGE Example:

```
conn = __salt__['pyeapi.get_connection'](host='router1.example.com',
                                          username='example',
                                          password='example')
show_ver = conn.run_commands(['show version', 'show interfaces'])
```

`salt.modules.arista_pyeapi.`**`run_commands`**(**commands*, ***kwargs*)

Sends the commands over the transport to the device.

This function sends the commands to the device using the nodes transport. This is a lower layer function that shouldn't normally need to be used, preferring instead to use `config()` or `enable()`.

transport: `https` Specifies the type of connection transport to use. Valid values for the connection are `socket`, `http_local`, `http`, and `https`.

> **Note:** This argument does not need to be specified when running in a *pyeapi* Proxy Minion.

host: `localhost` The IP address or DNS host name of the connection device.

> **Note:** This argument does not need to be specified when running in a *pyeapi* Proxy Minion.

username: `admin` The username to pass to the device to authenticate the eAPI connection.

> **Note:** This argument does not need to be specified when running in a *pyeapi* Proxy Minion.

password The password to pass to the device to authenticate the eAPI connection.

> **Note:** This argument does not need to be specified when running in a *pyeapi* Proxy Minion.

port The TCP port of the endpoint for the eAPI connection. If this keyword is not specified, the default value is automatically determined by the transport type (`80` for `http`, or `443` for `https`).

> **Note:** This argument does not need to be specified when running in a *pyeapi* Proxy Minion.

enablepwd The enable mode password if required by the destination node.

> **Note:** This argument does not need to be specified when running in a *pyeapi* Proxy Minion.

CLI Example:

```
salt '*' pyeapi.run_commands 'show version'
salt '*' pyeapi.run_commands 'show version' encoding=text
salt '*' pyeapi.run_commands 'show version' encoding=text host=cr1.thn.lon
↪username=example password=weak
```

Output example:

```
veos1:
    |_
      ----------
      architecture:
          i386
      bootupTimestamp:
          1527541728.53
      hardwareRevision:
      internalBuildId:
          63d2e89a-220d-4b8a-a9b3-0524fa8f9c5f
      internalVersion:
          4.18.1F-4591672.4181F
      isIntlVersion:
          False
      memFree:
          501468
      memTotal:
          1893316
      modelName:
          vEOS
      serialNumber:
      systemMacAddress:
          52:54:00:3f:e6:d0
      version:
          4.18.1F
```

`salt.modules.arista_pyeapi.`**`section`**(*regex*, *config='running-config'*, ***kwargs*)

Return a section of the config.

regex A valid regular expression used to select sections of configuration to return.

config: **`running-config`** The configuration to return. Valid values for config are `running-config` or `startup-config`. The default value is `running-config`.

transport: **`https`** Specifies the type of connection transport to use. Valid values for the connection are `socket`, `http_local`, `http`, and `https`.

Note: This argument does not need to be specified when running in a *`pyeapi`* Proxy Minion.

host: **`localhost`** The IP address or DNS host name of the connection device.

Note: This argument does not need to be specified when running in a *`pyeapi`* Proxy Minion.

username: **`admin`** The username to pass to the device to authenticate the eAPI connection.

Note: This argument does not need to be specified when running in a *`pyeapi`* Proxy Minion.

password The password to pass to the device to authenticate the eAPI connection.

Note: This argument does not need to be specified when running in a *`pyeapi`* Proxy Minion.

port The TCP port of the endpoint for the eAPI connection. If this keyword is not specified, the default value is automatically determined by the transport type (`80` for `http`, or `443` for `https`).

Note: This argument does not need to be specified when running in a *`pyeapi`* Proxy Minion.

enablepwd The enable mode password if required by the destination node.

Note: This argument does not need to be specified when running in a *`pyeapi`* Proxy Minion.

CLI Example:

```
salt '*'
```

21.9.20 salt.modules.artifactory

Module for fetching artifacts from Artifactory

`exception` `salt.modules.artifactory.`**`ArtifactoryError`**(*value*)

`salt.modules.artifactory.`**`get_latest_release`**(*artifactory_url*, *repository*, *group_id*, *artifact_id*, *packaging*, *target_dir='/tmp'*, *target_file=None*, *classifier=None*, *username=None*, *password=None*, *use_literal_group_id=False*)

Gets the latest release of the artifact

artifactory_url URL of artifactory instance

repository Release repository in artifactory to retrieve artifact from, for example: libs-releases

group_id Group Id of the artifact

artifact_id Artifact Id of the artifact

packaging Packaging type (jar,war,ear,etc)

target_dir Target directory to download artifact to (default: /tmp)

target_file Target file to download artifact to (by default it is target_dir/artifact_id-version.packaging)

classifier Artifact classifier name (ex: sources,javadoc,etc). Optional parameter.

username Artifactory username. Optional parameter.

password Artifactory password. Optional parameter.

`salt.modules.artifactory.`**`get_latest_snapshot`**(*artifactory_url*, *repository*, *group_id*, *artifact_id*, *packaging*, *target_dir='/tmp'*, *target_file=None*, *classifier=None*, *username=None*, *password=None*, *use_literal_group_id=False*)

Gets latest snapshot of the given artifact

artifactory_url URL of artifactory instance

repository Snapshot repository in artifactory to retrieve artifact from, for example: libs-snapshots

group_id Group Id of the artifact

artifact_id Artifact Id of the artifact

packaging Packaging type (jar,war,ear,etc)

target_dir Target directory to download artifact to (default: /tmp)

target_file Target file to download artifact to (by default it is target_dir/artifact_id-snapshot_version.packaging)

classifier Artifact classifier name (ex: sources,javadoc,etc). Optional parameter.

username Artifactory username. Optional parameter.

password Artifactory password. Optional parameter.

`salt.modules.artifactory.`**`get_release`**(*artifactory_url*, *repository*, *group_id*, *artifact_id*, *packaging*, *version*, *target_dir='/tmp'*, *target_file=None*, *classifier=None*, *username=None*, *password=None*, *use_literal_group_id=False*)

Gets the specified release of the artifact

artifactory_url URL of artifactory instance

repository Release repository in artifactory to retrieve artifact from, for example: libs-releases

group_id Group Id of the artifact

artifact_id Artifact Id of the artifact

packaging Packaging type (jar,war,ear,etc)

version Version of the artifact

target_dir Target directory to download artifact to (default: /tmp)

target_file Target file to download artifact to (by default it is target_dir/artifact_id-version.packaging)

classifier Artifact classifier name (ex: sources,javadoc,etc). Optional parameter.

username Artifactory username. Optional parameter.

password Artifactory password. Optional parameter.

`salt.modules.artifactory.`**`get_snapshot`**(*artifactory_url, repository, group_id, artifact_id, packaging, version, snapshot_version=None, target_dir='/tmp', target_file=None, classifier=None, username=None, password=None, use_literal_group_id=False*)

Gets snapshot of the desired version of the artifact

artifactory_url URL of artifactory instance

repository Snapshot repository in artifactory to retrieve artifact from, for example: libs-snapshots

group_id Group Id of the artifact

artifact_id Artifact Id of the artifact

packaging Packaging type (jar,war,ear,etc)

version Version of the artifact

target_dir Target directory to download artifact to (default: /tmp)

target_file Target file to download artifact to (by default it is target_dir/artifact_id-snapshot_version.packaging)

classifier Artifact classifier name (ex: sources,javadoc,etc). Optional parameter.

username Artifactory username. Optional parameter.

password Artifactory password. Optional parameter.

21.9.21 salt.modules.at

Wrapper module for at(1)

Also, a 'tag' feature has been added to more easily tag jobs.

platform linux,openbsd,freebsd

Changed in version 2017.7.0.

`salt.modules.at.`**`at`**(**args, **kwargs*)

Add a job to the queue.

The 'timespec' follows the format documented in the at(1) manpage.

CLI Example:

```
salt '*' at.at <timespec> <cmd> [tag=<tag>] [runas=<user>]
salt '*' at.at 12:05am '/sbin/reboot' tag=reboot
salt '*' at.at '3:05am +3 days' 'bin/myscript' tag=nightly runas=jim
```

`salt.modules.at.`**`atc`**(*jobid*)

Print the at(1) script that will run for the passed job id. This is mostly for debugging so the output will just be text.

CLI Example:

```
salt '*' at.atc <jobid>
```

`salt.modules.at.`**`atq`**`(`*tag=None*`)`

List all queued and running jobs or only those with an optional 'tag'.

CLI Example:

```
salt '*' at.atq
salt '*' at.atq [tag]
salt '*' at.atq [job number]
```

`salt.modules.at.`**`atrm`**`(`**args*`)`

Remove jobs from the queue.

CLI Example:

```
salt '*' at.atrm <jobid> <jobid> .. <jobid>
salt '*' at.atrm all
salt '*' at.atrm all [tag]
```

`salt.modules.at.`**`jobcheck`**`(`***kwargs*`)`

Check the job from queue. The kwargs dict include 'hour minute day month year tag runas' Other parameters will be ignored.

CLI Example:

```
salt '*' at.jobcheck runas=jam day=13
salt '*' at.jobcheck day=13 month=12 year=13 tag=rose
```

21.9.22 salt.modules.at_solaris

Wrapper for at(1) on Solaris-like systems

Note: we try to mirror the generic at module where possible

maintainer jorge schrauwen <sjorge@blackdot.be>

maturity new

platform solaris,illumos,smartso

New in version 2017.7.0.

`salt.modules.at_solaris.`**`at`**`(`**args, **kwargs*`)`

Add a job to the queue.

The 'timespec' follows the format documented in the at(1) manpage.

CLI Example:

```
salt '*' at.at <timespec> <cmd> [tag=<tag>] [runas=<user>]
salt '*' at.at 12:05am '/sbin/reboot' tag=reboot
salt '*' at.at '3:05am +3 days' 'bin/myscript' tag=nightly runas=jim
```

`salt.modules.at_solaris.`**`atc`**`(`*jobid*`)`

Print the at(1) script that will run for the passed job id. This is mostly for debugging so the output will just be text.

CLI Example:

```
salt '*' at.atc <jobid>
```

`salt.modules.at_solaris.`**`atq`**(*tag=None*)

List all queued and running jobs or only those with an optional 'tag'.

CLI Example:

```
salt '*' at.atq
salt '*' at.atq [tag]
salt '*' at.atq [job number]
```

`salt.modules.at_solaris.`**`atrm`**(**args*)

Remove jobs from the queue.

CLI Example:

```
salt '*' at.atrm <jobid> <jobid> .. <jobid>
salt '*' at.atrm all
salt '*' at.atrm all [tag]
```

`salt.modules.at_solaris.`**`jobcheck`**(***kwargs*)

Check the job from queue. The kwargs dict include 'hour minute day month year tag runas' Other parameters will be ignored.

CLI Example:

```
salt '*' at.jobcheck runas=jam day=13
salt '*' at.jobcheck day=13 month=12 year=13 tag=rose
```

21.9.23 salt.modules.augeas_cfg

Manages configuration files via augeas

This module requires the `augeas` Python module.

Warning: Minimal installations of Debian and Ubuntu have been seen to have packaging bugs with python-augeas, causing the augeas module to fail to import. If the minion has the augeas module installed, but the functions in this execution module fail to run due to being unavailable, first restart the salt-minion service. If the problem persists past that, the following command can be run from the master to determine what is causing the import to fail:

```
salt minion-id cmd.run 'python -c "from augeas import Augeas"'
```

For affected Debian/Ubuntu hosts, installing `libpython2.7` has been known to resolve the issue.

`salt.modules.augeas_cfg.`**`execute`**(*context=None, lens=None, commands=(), load_path=None*)

Execute Augeas commands

New in version 2014.7.0.

CLI Example:

```
salt '*' augeas.execute /files/etc/redis/redis.conf \
commands='["set bind 0.0.0.0", "set maxmemory 1G"]'
```

context The Augeas context

lens The Augeas lens to use

commands The Augeas commands to execute

New in version 2016.3.0.

load_path A colon-spearated list of directories that modules should be searched in. This is in addition to the standard load path and the directories in AUGEAS_LENS_LIB.

`salt.modules.augeas_cfg.`**`get`**(*path*, *value=''*, *load_path=None*)

Get a value for a specific augeas path

CLI Example:

```
salt '*' augeas.get /files/etc/hosts/1/ ipaddr
```

path The path to get the value of

value The optional value to get

New in version 2016.3.0.

load_path A colon-spearated list of directories that modules should be searched in. This is in addition to the standard load path and the directories in AUGEAS_LENS_LIB.

`salt.modules.augeas_cfg.`**`ls`**(*path*, *load_path=None*)

List the direct children of a node

CLI Example:

```
salt '*' augeas.ls /files/etc/passwd
```

path The path to list

New in version 2016.3.0.

load_path A colon-spearated list of directories that modules should be searched in. This is in addition to the standard load path and the directories in AUGEAS_LENS_LIB.

`salt.modules.augeas_cfg.`**`match`**(*path*, *value=''*, *load_path=None*)

Get matches for path expression

CLI Example:

```
salt '*' augeas.match /files/etc/services/service-name ssh
```

path The path to match

value The value to match on

New in version 2016.3.0.

load_path A colon-spearated list of directories that modules should be searched in. This is in addition to the standard load path and the directories in AUGEAS_LENS_LIB.

`salt.modules.augeas_cfg.`**`remove`**(*path*, *load_path=None*)

Get matches for path expression

CLI Example:

```
salt '*' augeas.remove \
/files/etc/sysctl.conf/net.ipv4.conf.all.log_martians
```

path The path to remove

New in version 2016.3.0.

load_path A colon-spearated list of directories that modules should be searched in. This is in addition to the standard load path and the directories in AUGEAS_LENS_LIB.

`salt.modules.augeas_cfg.`**`setvalue`**`(`**args*`)`

Set a value for a specific augeas path

CLI Example:

```
salt '*' augeas.setvalue /files/etc/hosts/1/canonical localhost
```

This will set the first entry in /etc/hosts to localhost

CLI Example:

```
salt '*' augeas.setvalue /files/etc/hosts/01/ipaddr 192.168.1.1 \
                         /files/etc/hosts/01/canonical test
```

Adds a new host to /etc/hosts the ip address 192.168.1.1 and hostname test

CLI Example:

```
salt '*' augeas.setvalue prefix=/files/etc/sudoers/ \
         "spec[user = '%wheel']/user" "%wheel" \
         "spec[user = '%wheel']/host_group/host" 'ALL' \
         "spec[user = '%wheel']/host_group/command[1]" 'ALL' \
         "spec[user = '%wheel']/host_group/command[1]/tag" 'PASSWD' \
         "spec[user = '%wheel']/host_group/command[2]" '/usr/bin/apt-get' \
         "spec[user = '%wheel']/host_group/command[2]/tag" NOPASSWD
```

Ensures that the following line is present in /etc/sudoers:

```
%wheel ALL = PASSWD : ALL , NOPASSWD : /usr/bin/apt-get , /usr/bin/aptitude
```

`salt.modules.augeas_cfg.`**`tree`**`(`*path, load_path=None*`)`

Returns recursively the complete tree of a node

CLI Example:

```
salt '*' augeas.tree /files/etc/
```

path The base of the recursive listing

New in version 2016.3.0.

load_path A colon-spearated list of directories that modules should be searched in. This is in addition to the standard load path and the directories in AUGEAS_LENS_LIB.

21.9.24 salt.modules.aws_sqs

Support for the Amazon Simple Queue Service.

`salt.modules.aws_sqs.`**`create_queue`**`(`*name*, *region*, *opts=None*, *user=None*`)`

Creates a queue with the correct name.

name Name of the SQS queue to create

region Region to create the SQS queue in

opts [None] Any additional options to add to the command line

user [None] Run hg as a user other than what the minion runs as

CLI Example:

salt '*' aws_sqs.create_queue <sqs queue> <region>

`salt.modules.aws_sqs.`**`delete_message`**`(`*queue*, *region*, *receipthandle*, *opts=None*, *user=None*`)`

Delete one or more messages from a queue in a region

queue The name of the queue to delete messages from

region Region where SQS queues exists

receipthandle The ReceiptHandle of the message to delete. The ReceiptHandle is obtained in the return from receive_message

opts [None] Any additional options to add to the command line

user [None] Run as a user other than what the minion runs as

CLI Example:

```
salt '*' aws_sqs.delete_message <sqs queue> <region> receipthandle='<sqs
↪ReceiptHandle>'
```

New in version 2014.7.0.

`salt.modules.aws_sqs.`**`delete_queue`**`(`*name*, *region*, *opts=None*, *user=None*`)`

Deletes a queue in the region.

name Name of the SQS queue to deletes

region Name of the region to delete the queue from

opts [None] Any additional options to add to the command line

user [None] Run hg as a user other than what the minion runs as

CLI Example:

salt '*' aws_sqs.delete_queue <sqs queue> <region>

`salt.modules.aws_sqs.`**`list_queues`**`(`*region*, *opts=None*, *user=None*`)`

List the queues in the selected region.

region Region to list SQS queues for

opts [None] Any additional options to add to the command line

user [None] Run hg as a user other than what the minion runs as

CLI Example:

salt '*' aws_sqs.list_queues <region>

`salt.modules.aws_sqs.`**`queue_exists`**`(`*name*, *region*, *opts=None*, *user=None*`)`

Returns True or False on whether the queue exists in the region

name Name of the SQS queue to search for

region Name of the region to search for the queue in

opts [None] Any additional options to add to the command line

user [None] Run hg as a user other than what the minion runs as

CLI Example:

salt ’*’ aws_sqs.queue_exists <sqs queue> <region>

`salt.modules.aws_sqs.`**`receive_message`**`(`*queue, region, num=1, opts=None, user=None*`)`

Receive one or more messages from a queue in a region

queue The name of the queue to receive messages from

region Region where SQS queues exists

num [1] The max number of messages to receive

opts [None] Any additional options to add to the command line

user [None] Run as a user other than what the minion runs as

CLI Example:

```
salt '*' aws_sqs.receive_message <sqs queue> <region>
salt '*' aws_sqs.receive_message <sqs queue> <region> num=10
```

New in version 2014.7.0.

21.9.25 salt.modules.azurearm_compute

Azure (ARM) Compute Execution Module

New in version 2019.2.0.

maintainer <devops@decisionlab.io>

maturity new

depends

- azure >= 2.0.0
- azure-common >= 1.1.8
- azure-mgmt >= 1.0.0
- azure-mgmt-compute >= 1.0.0
- azure-mgmt-network >= 1.7.1
- azure-mgmt-resource >= 1.1.0
- azure-mgmt-storage >= 1.0.0
- azure-mgmt-web >= 0.32.0
- azure-storage >= 0.34.3
- msrestazure >= 0.4.21

platform linux

configuration This module requires Azure Resource Manager credentials to be passed as keyword arguments to every function in order to work properly.

Required provider parameters:

if using username and password:

- `subscription_id`
- `username`
- `password`

if using a service principal:

- `subscription_id`
- `tenant`
- `client_id`
- `secret`

Optional provider parameters:

cloud_environment: Used to point the cloud driver to different API endpoints, such as Azure GovCloud.

Possible values:

- `AZURE_PUBLIC_CLOUD` (default)
- `AZURE_CHINA_CLOUD`
- `AZURE_US_GOV_CLOUD`
- `AZURE_GERMAN_CLOUD`

`salt.modules.azurearm_compute.`**`availability_set_create_or_update`**`(`*name, resource_group, **kwargs*`)`

New in version 2019.2.0.

Create or update an availability set.

Parameters

- **`name`** – The availability set to create.
- **`resource_group`** – The resource group name assigned to the availability set.

CLI Example:

```
salt-call azurearm_compute.availability_set_create_or_update testset testgroup
```

`salt.modules.azurearm_compute.`**`availability_set_delete`**`(`*name, resource_group, **kwargs*`)`

New in version 2019.2.0.

Delete an availability set.

Parameters

- **`name`** – The availability set to delete.
- **`resource_group`** – The resource group name assigned to the availability set.

CLI Example:

```
salt-call azurearm_compute.availability_set_delete testset testgroup
```

`salt.modules.azurearm_compute.`**`availability_set_get`**`(`*name*, *resource_group*, ***kwargs*`)`

New in version 2019.2.0.

Get a dictionary representing an availability set's properties.

Parameters

- **`name`** – The availability set to get.
- **`resource_group`** – The resource group name assigned to the availability set.

CLI Example:

```
salt-call azurearm_compute.availability_set_get testset testgroup
```

`salt.modules.azurearm_compute.`**`availability_sets_list`**`(`*resource_group*, ***kwargs*`)`

New in version 2019.2.0.

List all availability sets within a resource group.

Parameters **`resource_group`** – The resource group name to list availability sets within.

CLI Example:

```
salt-call azurearm_compute.availability_sets_list testgroup
```

`salt.modules.azurearm_compute.`**`availability_sets_list_available_sizes`**`(`*name*, *resource_group*, ***kwargs*`)`

New in version 2019.2.0.

List all available virtual machine sizes that can be used to to create a new virtual machine in an existing availability set.

Parameters

- **`name`** – The availability set name to list available virtual machine sizes within.
- **`resource_group`** – The resource group name to list available availability set sizes within.

CLI Example:

```
salt-call azurearm_compute.availability_sets_list_available_sizes testset
↪testgroup
```

`salt.modules.azurearm_compute.`**`virtual_machine_capture`**`(`*name*, *destination_name*, *resource_group*, *prefix='capture-'*, *overwrite=False*, ***kwargs*`)`

New in version 2019.2.0.

Captures the VM by copying virtual hard disks of the VM and outputs a template that can be used to create similar VMs.

Parameters

- **`name`** – The name of the virtual machine.
- **`destination_name`** – The destination container name.
- **`resource_group`** – The resource group name assigned to the virtual machine.

- **prefix** – (Default: 'capture-') The captured virtual hard disk's name prefix.
- **overwrite** – (Default: False) Overwrite the destination disk in case of conflict.

CLI Example:

```
salt-call azurearm_compute.virtual_machine_capture testvm testcontainer testgroup
```

`salt.modules.azurearm_compute.`**`virtual_machine_convert_to_managed_disks`**`(`*name, resource_group, **kwargs*`)`

New in version 2019.2.0.

Converts virtual machine disks from blob-based to managed disks. Virtual machine must be stop-deallocated before invoking this operation.

Parameters

- **name** – The name of the virtual machine to convert.
- **resource_group** – The resource group name assigned to the virtual machine.

CLI Example:

```
salt-call azurearm_compute.virtual_machine_convert_to_managed_disks testvm␣
↪testgroup
```

`salt.modules.azurearm_compute.`**`virtual_machine_deallocate`**`(`*name, resource_group, **kwargs*`)`

New in version 2019.2.0.

Power off a virtual machine and deallocate compute resources.

Parameters

- **name** – The name of the virtual machine to deallocate.
- **resource_group** – The resource group name assigned to the virtual machine.

CLI Example:

```
salt-call azurearm_compute.virtual_machine_deallocate testvm testgroup
```

`salt.modules.azurearm_compute.`**`virtual_machine_generalize`**`(`*name, resource_group, **kwargs*`)`

New in version 2019.2.0.

Set the state of a virtual machine to 'generalized'.

Parameters

- **name** – The name of the virtual machine.
- **resource_group** – The resource group name assigned to the virtual machine.

CLI Example:

```
salt-call azurearm_compute.virtual_machine_generalize testvm testgroup
```

`salt.modules.azurearm_compute.`**`virtual_machine_get`**`(`*name, resource_group, **kwargs*`)`

New in version 2019.2.0.

Retrieves information about the model view or the instance view of a virtual machine.

Parameters

- **name** – The name of the virtual machine.
- **resource_group** – The resource group name assigned to the virtual machine.

CLI Example:

```
salt-call azurearm_compute.virtual_machine_get testvm testgroup
```

salt.modules.azurearm_compute.**virtual_machine_power_off**(*name*, *resource_group*, ***kwargs*)

New in version 2019.2.0.

Power off (stop) a virtual machine.

Parameters

- **name** – The name of the virtual machine to stop.
- **resource_group** – The resource group name assigned to the virtual machine.

CLI Example:

```
salt-call azurearm_compute.virtual_machine_power_off testvm testgroup
```

salt.modules.azurearm_compute.**virtual_machine_redeploy**(*name*, *resource_group*, ***kwargs*)

New in version 2019.2.0.

Redeploy a virtual machine.

Parameters

- **name** – The name of the virtual machine to redeploy.
- **resource_group** – The resource group name assigned to the virtual machine.

CLI Example:

```
salt-call azurearm_compute.virtual_machine_redeploy testvm testgroup
```

salt.modules.azurearm_compute.**virtual_machine_restart**(*name*, *resource_group*, ***kwargs*)

New in version 2019.2.0.

Restart a virtual machine.

Parameters

- **name** – The name of the virtual machine to restart.
- **resource_group** – The resource group name assigned to the virtual machine.

CLI Example:

```
salt-call azurearm_compute.virtual_machine_restart testvm testgroup
```

salt.modules.azurearm_compute.**virtual_machine_start**(*name*, *resource_group*, ***kwargs*)

New in version 2019.2.0.

Power on (start) a virtual machine.

Parameters

- **`name`** – The name of the virtual machine to start.
- **`resource_group`** – The resource group name assigned to the virtual machine.

CLI Example:

```
salt-call azurearm_compute.virtual_machine_start testvm testgroup
```

`salt.modules.azurearm_compute.`**`virtual_machines_list`**(*resource_group*, ***kwargs*)

New in version 2019.2.0.

List all virtual machines within a resource group.

Parameters **`resource_group`** – The resource group name to list virtual machines within.

CLI Example:

```
salt-call azurearm_compute.virtual_machines_list testgroup
```

`salt.modules.azurearm_compute.`**`virtual_machines_list_all`**(***kwargs*)

New in version 2019.2.0.

List all virtual machines within a subscription.

CLI Example:

```
salt-call azurearm_compute.virtual_machines_list_all
```

`salt.modules.azurearm_compute.`**`virtual_machines_list_available_sizes`**(*name*, *resource_group*, ***kwargs*)

New in version 2019.2.0.

Lists all available virtual machine sizes to which the specified virtual machine can be resized.

Parameters

- **`name`** – The name of the virtual machine.
- **`resource_group`** – The resource group name assigned to the virtual machine.

CLI Example:

```
salt-call azurearm_compute.virtual_machines_list_available_sizes testvm testgroup
```

21.9.26 salt.modules.azurearm_network

Azure (ARM) Network Execution Module

New in version 2019.2.0.

maintainer <devops@decisionlab.io>

maturity new

depends

- azure >= 2.0.0
- azure-common >= 1.1.8
- azure-mgmt >= 1.0.0
- azure-mgmt-compute >= 1.0.0

- azure-mgmt-network >= 1.7.1
- azure-mgmt-resource >= 1.1.0
- azure-mgmt-storage >= 1.0.0
- azure-mgmt-web >= 0.32.0
- azure-storage >= 0.34.3
- msrestazure >= 0.4.21

platform linux

configuration This module requires Azure Resource Manager credentials to be passed as keyword arguments to every function in order to work properly.

Required provider parameters:

if using username and password:

- `subscription_id`
- `username`
- `password`

if using a service principal:

- `subscription_id`
- `tenant`
- `client_id`
- `secret`

Optional provider parameters:

cloud_environment: Used to point the cloud driver to different API endpoints, such as Azure GovCloud.

Possible values:

- `AZURE_PUBLIC_CLOUD` (default)
- `AZURE_CHINA_CLOUD`
- `AZURE_US_GOV_CLOUD`
- `AZURE_GERMAN_CLOUD`

`salt.modules.azurearm_network.`**`check_dns_name_availability`**`(`*name*, *region*, ***kwargs*`)`

New in version 2019.2.0.

Check whether a domain name in the current zone is available for use.

Parameters

- **`name`** – The DNS name to query.
- **`region`** – The region to query for the DNS name in question.

CLI Example:

```
salt-call azurearm_network.check_dns_name_availability testdnsname westus
```

`salt.modules.azurearm_network.`**`check_ip_address_availability`**(*ip_address*, *virtual_network*, *resource_group*, ***kwargs*)

New in version 2019.2.0.

Check that a private ip address is available within the specified virtual network.

Parameters

- **ip_address** – The ip_address to query.
- **virtual_network** – The virtual network to query for the IP address in question.
- **resource_group** – The resource group name assigned to the virtual network.

CLI Example:

```
salt-call azurearm_network.check_ip_address_availability 10.0.0.4 testnet␣
↪testgroup
```

`salt.modules.azurearm_network.`**`default_security_rule_get`**(*name*, *security_group*, *resource_group*, ***kwargs*)

New in version 2019.2.0.

Get details about a default security rule within a security group.

Parameters

- **name** – The name of the security rule to query.
- **security_group** – The network security group containing the security rule.
- **resource_group** – The resource group name assigned to the network security group.

CLI Example:

```
salt-call azurearm_network.default_security_rule_get DenyAllOutBound testnsg␣
↪testgroup
```

`salt.modules.azurearm_network.`**`default_security_rules_list`**(*security_group*, *resource_group*, ***kwargs*)

New in version 2019.2.0.

List default security rules within a security group.

Parameters

- **security_group** – The network security group to query.
- **resource_group** – The resource group name assigned to the network security group.

CLI Example:

```
salt-call azurearm_network.default_security_rules_list testnsg testgroup
```

`salt.modules.azurearm_network.`**`get_virtual_machine_scale_set_network_interface`**(*name*, *scale_set*, *vm_index*, *resource_group*, ***kwargs*)

New in version 2019.2.0.

Get information about a specfic network interface within a scale set.

Parameters

- **`name`** – The name of the network interface to query.
- **`scale_set`** – The name of the scale set containing the interface.
- **`vm_index`** – The virtual machine index.
- **`resource_group`** – The resource group name assigned to the scale set.

CLI Example:

```
salt-call azurearm_network.get_virtual_machine_scale_set_network_interface test-
↪iface0 testset testvm testgroup
```

`salt.modules.azurearm_network.`**`list_virtual_machine_scale_set_network_interfaces`**(*scale_set*, *resource_group*, ***kwargs*)

New in version 2019.2.0.

Get information about all network interfaces within a scale set.

Parameters

- **`scale_set`** – The name of the scale set to query.
- **`resource_group`** – The resource group name assigned to the scale set.

CLI Example:

```
salt-call azurearm_network.list_virtual_machine_scale_set_vm_network_interfaces␣
↪testset testgroup
```

`salt.modules.azurearm_network.`**`list_virtual_machine_scale_set_vm_network_interfaces`**(*scale_set*, *vm_index*, *re-source_gro*
***kwargs*)

New in version 2019.2.0.

Get information about all network interfaces in a specific virtual machine within a scale set.

Parameters

- **`scale_set`** – The name of the scale set to query.
- **`vm_index`** – The virtual machine index.
- **`resource_group`** – The resource group name assigned to the scale set.

CLI Example:

```
salt-call azurearm_network.list_virtual_machine_scale_set_vm_network_interfaces␣
↪testset testvm testgroup
```

`salt.modules.azurearm_network.`**`load_balancer_create_or_update`**(*name*, *resource_group*, ***kwargs*)

New in version 2019.2.0.

Create or update a load balancer within a specified resource group.

Parameters

- **name** – The name of the load balancer to create.
- **resource_group** – The resource group name assigned to the load balancer.

CLI Example:

```
salt-call azurearm_network.load_balancer_create_or_update testlb testgroup
```

salt.modules.azurearm_network.**load_balancer_delete**(*name*, *resource_group*, ***kwargs*)
New in version 2019.2.0.

Delete a load balancer.

Parameters

- **name** – The name of the load balancer to delete.
- **resource_group** – The resource group name assigned to the load balancer.

CLI Example:

```
salt-call azurearm_network.load_balancer_delete testlb testgroup
```

salt.modules.azurearm_network.**load_balancer_get**(*name*, *resource_group*, ***kwargs*)
New in version 2019.2.0.

Get details about a specific load balancer.

Parameters

- **name** – The name of the load balancer to query.
- **resource_group** – The resource group name assigned to the load balancer.

CLI Example:

```
salt-call azurearm_network.load_balancer_get testlb testgroup
```

salt.modules.azurearm_network.**load_balancers_list**(*resource_group*, ***kwargs*)
New in version 2019.2.0.

List all load balancers within a resource group.

Parameters **resource_group** – The resource group name to list load balancers within.

CLI Example:

```
salt-call azurearm_network.load_balancers_list testgroup
```

salt.modules.azurearm_network.**load_balancers_list_all**(***kwargs*)
New in version 2019.2.0.

List all load balancers within a subscription.

CLI Example:

```
salt-call azurearm_network.load_balancers_list_all
```

`salt.modules.azurearm_network.`**`network_interface_create_or_update`**`(`*name, ip_configurations, subnet, virtual_network, resource_group, **kwargs*`)`

New in version 2019.2.0.

Create or update a network interface within a specified resource group.

Parameters

- **`name`** – The name of the network interface to create.
- **`ip_configurations`** – A list of dictionaries representing valid NetworkInterfaceIPConfiguration objects. The 'name' key is required at minimum. At least one IP Configuration must be present.
- **`subnet`** – The name of the subnet assigned to the network interface.
- **`virtual_network`** – The name of the virtual network assigned to the subnet.
- **`resource_group`** – The resource group name assigned to the virtual network.

CLI Example:

```
salt-call azurearm_network.network_interface_create_or_update test-iface0 [{'name
↪': 'testipconfig1'}]                          testsubnet testnet testgroup
```

`salt.modules.azurearm_network.`**`network_interface_delete`**`(`*name, resource_group, **kwargs*`)`

New in version 2019.2.0.

Delete a network interface.

Parameters

- **`name`** – The name of the network interface to delete.
- **`resource_group`** – The resource group name assigned to the network interface.

CLI Example:

```
salt-call azurearm_network.network_interface_delete test-iface0 testgroup
```

`salt.modules.azurearm_network.`**`network_interface_get`**`(`*name, resource_group, **kwargs*`)`

New in version 2019.2.0.

Get details about a specific network interface.

Parameters

- **`name`** – The name of the network interface to query.
- **`resource_group`** – The resource group name assigned to the network interface.

CLI Example:

```
salt-call azurearm_network.network_interface_get test-iface0 testgroup
```

salt.modules.azurearm_network.**network_interface_get_effective_route_table**(*name, resource_group, **kwargs*)

New in version 2019.2.0.

Get all route tables for a specific network interface.

Parameters

- **name** – The name of the network interface to query.
- **resource_group** – The resource group name assigned to the network interface.

CLI Example:

```
salt-call azurearm_network.network_interface_get_effective_route_table test-
↪iface0 testgroup
```

salt.modules.azurearm_network.**network_interface_list_effective_network_security_groups**(*nam re- sourc **kw*

New in version 2019.2.0.

Get all network security groups applied to a specific network interface.

Parameters

- **name** – The name of the network interface to query.
- **resource_group** – The resource group name assigned to the network interface.

CLI Example:

```
salt-call azurearm_network.network_interface_list_effective_network_security_
↪groups test-iface0 testgroup
```

salt.modules.azurearm_network.**network_interfaces_list**(*resource_group, **kwargs*)

New in version 2019.2.0.

List all network interfaces within a resource group.

Parameters **resource_group** – The resource group name to list network interfaces within.

CLI Example:

```
salt-call azurearm_network.network_interfaces_list testgroup
```

salt.modules.azurearm_network.**network_interfaces_list_all**(***kwargs*)

New in version 2019.2.0.

List all network interfaces within a subscription.

CLI Example:

```
salt-call azurearm_network.network_interfaces_list_all
```

salt.modules.azurearm_network.**network_security_group_create_or_update**(*name, resource_group, **kwargs*)

New in version 2019.2.0.

Create or update a network security group.

Parameters

- **name** – The name of the network security group to create.
- **resource_group** – The resource group name assigned to the network security group.

CLI Example:

```
salt-call azurearm_network.network_security_group_create_or_update testnsg␣
↪testgroup
```

salt.modules.azurearm_network.**network_security_group_delete**(*name*, *resource_group*, ***kwargs*)

New in version 2019.2.0.

Delete a network security group within a resource group.

Parameters

- **name** – The name of the network security group to delete.
- **resource_group** – The resource group name assigned to the network security group.

CLI Example:

```
salt-call azurearm_network.network_security_group_delete testnsg testgroup
```

salt.modules.azurearm_network.**network_security_group_get**(*name*, *resource_group*, ***kwargs*)

New in version 2019.2.0.

Get details about a network security group within a resource group.

Parameters

- **name** – The name of the network security group to query.
- **resource_group** – The resource group name assigned to the network security group.

CLI Example:

```
salt-call azurearm_network.network_security_group_get testnsg testgroup
```

salt.modules.azurearm_network.**network_security_groups_list**(*resource_group*, ***kwargs*)

New in version 2019.2.0.

List all network security groups within a resource group.

Parameters **resource_group** – The resource group name to list network security groups within.

CLI Example:

```
salt-call azurearm_network.network_security_groups_list testgroup
```

salt.modules.azurearm_network.**network_security_groups_list_all**(***kwargs*)

New in version 2019.2.0.

List all network security groups within a subscription.

CLI Example:

```
salt-call azurearm_network.network_security_groups_list_all
```

salt.modules.azurearm_network.**public_ip_address_create_or_update**(*name*, *resource_group*, ***kwargs*)

New in version 2019.2.0.

Create or update a public IP address within a specified resource group.

Parameters

- **name** – The name of the public IP address to create.
- **resource_group** – The resource group name assigned to the public IP address.

CLI Example:

```
salt-call azurearm_network.public_ip_address_create_or_update test-ip-0 testgroup
```

salt.modules.azurearm_network.**public_ip_address_delete**(*name*, *resource_group*, ***kwargs*)

New in version 2019.2.0.

Delete a public IP address.

Parameters

- **name** – The name of the public IP address to delete.
- **resource_group** – The resource group name assigned to the public IP address.

CLI Example:

```
salt-call azurearm_network.public_ip_address_delete test-pub-ip testgroup
```

salt.modules.azurearm_network.**public_ip_address_get**(*name*, *resource_group*, ***kwargs*)

New in version 2019.2.0.

Get details about a specific public IP address.

Parameters

- **name** – The name of the public IP address to query.
- **resource_group** – The resource group name assigned to the public IP address.

CLI Example:

```
salt-call azurearm_network.public_ip_address_get test-pub-ip testgroup
```

salt.modules.azurearm_network.**public_ip_addresses_list**(*resource_group*, ***kwargs*)

New in version 2019.2.0.

List all public IP addresses within a resource group.

Parameters **resource_group** – The resource group name to list public IP addresses within.

CLI Example:

```
salt-call azurearm_network.public_ip_addresses_list testgroup
```

`salt.modules.azurearm_network.`**`public_ip_addresses_list_all`**`(`***kwargs*`)`

New in version 2019.2.0.

List all public IP addresses within a subscription.

CLI Example:

```
salt-call azurearm_network.public_ip_addresses_list_all
```

`salt.modules.azurearm_network.`**`route_create_or_update`**`(`*name, address_prefix, next_hop_type, route_table, resource_group, next_hop_ip_address=None, **kwargs*`)`

New in version 2019.2.0.

Create or update a route within a specified route table.

Parameters

- **`name`** – The name of the route to create.
- **`address_prefix`** – The destination CIDR to which the route applies.
- **`next_hop_type`** – The type of Azure hop the packet should be sent to. Possible values are: 'VirtualNetworkGateway', 'VnetLocal', 'Internet', 'VirtualAppliance', and 'None'.
- **`next_hop_ip_address`** – Optional IP address to which packets should be forwarded. Next hop values are only allowed in routes where the next_hop_type is 'VirtualAppliance'.
- **`route_table`** – The name of the route table containing the route.
- **`resource_group`** – The resource group name assigned to the route table.

CLI Example:

```
salt-call azurearm_network.route_create_or_update test-rt '10.0.0.0/8' test-rt-
↪table testgroup
```

`salt.modules.azurearm_network.`**`route_delete`**`(`*name, route_table, resource_group, **kwargs*`)`

New in version 2019.2.0.

Delete a route from a route table.

Parameters

- **`name`** – The route to delete.
- **`route_table`** – The route table containing the route.
- **`resource_group`** – The resource group name assigned to the route table.

CLI Example:

```
salt-call azurearm_network.route_delete test-rt test-rt-table testgroup
```

`salt.modules.azurearm_network.`**`route_filter_create_or_update`**`(`*name, resource_group, **kwargs*`)`

New in version 2019.2.0.

Create or update a route filter within a specified resource group.

Parameters

- **name** – The name of the route filter to create.
- **resource_group** – The resource group name assigned to the route filter.

CLI Example:

```
salt-call azurearm_network.route_filter_create_or_update test-filter testgroup
```

`salt.modules.azurearm_network.`**`route_filter_delete`**`(`*name*, *resource_group*, ***kwargs*`)`

New in version 2019.2.0.

Delete a route filter.

Parameters

- **name** – The name of the route filter to delete.
- **resource_group** – The resource group name assigned to the route filter.

CLI Example:

```
salt-call azurearm_network.route_filter_delete test-filter testgroup
```

`salt.modules.azurearm_network.`**`route_filter_get`**`(`*name*, *resource_group*, ***kwargs*`)`

New in version 2019.2.0.

Get details about a specific route filter.

Parameters

- **name** – The name of the route table to query.
- **resource_group** – The resource group name assigned to the route filter.

CLI Example:

```
salt-call azurearm_network.route_filter_get test-filter testgroup
```

`salt.modules.azurearm_network.`**`route_filter_rule_create_or_update`**`(`*name*, *access*, *communities*, *route_filter*, *resource_group*, ***kwargs*`)`

New in version 2019.2.0.

Create or update a rule within a specified route filter.

Parameters

- **name** – The name of the rule to create.
- **access** – The access type of the rule. Valid values are 'Allow' and 'Deny'.
- **communities** – A list of BGP communities to filter on.
- **route_filter** – The name of the route filter containing the rule.
- **resource_group** – The resource group name assigned to the route filter.

CLI Example:

```
salt-call azurearm_network.route_filter_rule_create_or_update ␣
↪test-rule allow "['12076:51006']" test-filter testgroup
```

`salt.modules.azurearm_network.`**`route_filter_rule_delete`**`(`*name, route_filter, resource_group, **kwargs*`)`

New in version 2019.2.0.

Delete a route filter rule.

Parameters

- **`name`** – The route filter rule to delete.
- **`route_filter`** – The route filter containing the rule.
- **`resource_group`** – The resource group name assigned to the route filter.

CLI Example:

```
salt-call azurearm_network.route_filter_rule_delete test-rule test-filter␣
↪testgroup
```

`salt.modules.azurearm_network.`**`route_filter_rule_get`**`(`*name, route_filter, resource_group, **kwargs*`)`

New in version 2019.2.0.

Get details about a specific route filter rule.

Parameters

- **`name`** – The route filter rule to query.
- **`route_filter`** – The route filter containing the rule.
- **`resource_group`** – The resource group name assigned to the route filter.

CLI Example:

```
salt-call azurearm_network.route_filter_rule_get test-rule test-filter testgroup
```

`salt.modules.azurearm_network.`**`route_filter_rules_list`**`(`*route_filter, resource_group, **kwargs*`)`

New in version 2019.2.0.

List all routes within a route filter.

Parameters

- **`route_filter`** – The route filter to query.
- **`resource_group`** – The resource group name assigned to the route filter.

CLI Example:

```
salt-call azurearm_network.route_filter_rules_list test-filter testgroup
```

`salt.modules.azurearm_network.`**`route_filters_list`**`(`*resource_group, **kwargs*`)`

New in version 2019.2.0.

List all route filters within a resource group.

Parameters **`resource_group`** – The resource group name to list route filters within.

CLI Example:

```
salt-call azurearm_network.route_filters_list testgroup
```

salt.modules.azurearm_network.**route_filters_list_all**(***kwargs*)

New in version 2019.2.0.

List all route filters within a subscription.

CLI Example:

```
salt-call azurearm_network.route_filters_list_all
```

salt.modules.azurearm_network.**route_get**(*name*, *route_table*, *resource_group*, ***kwargs*)

New in version 2019.2.0.

Get details about a specific route.

Parameters

- **name** – The route to query.
- **route_table** – The route table containing the route.
- **resource_group** – The resource group name assigned to the route table.

CLI Example:

```
salt-call azurearm_network.route_get test-rt test-rt-table testgroup
```

salt.modules.azurearm_network.**route_table_create_or_update**(*name*, *resource_group*, ***kwargs*)

New in version 2019.2.0.

Create or update a route table within a specified resource group.

Parameters

- **name** – The name of the route table to create.
- **resource_group** – The resource group name assigned to the route table.

CLI Example:

```
salt-call azurearm_network.route_table_create_or_update test-rt-table testgroup
```

salt.modules.azurearm_network.**route_table_delete**(*name*, *resource_group*, ***kwargs*)

New in version 2019.2.0.

Delete a route table.

Parameters

- **name** – The name of the route table to delete.
- **resource_group** – The resource group name assigned to the route table.

CLI Example:

```
salt-call azurearm_network.route_table_delete test-rt-table testgroup
```

salt.modules.azurearm_network.**route_table_get**(*name*, *resource_group*, ***kwargs*)

New in version 2019.2.0.

Get details about a specific route table.

Parameters

- **name** – The name of the route table to query.

- **resource_group** – The resource group name assigned to the route table.

CLI Example:

```
salt-call azurearm_network.route_table_get test-rt-table testgroup
```

`salt.modules.azurearm_network.`**`route_tables_list`**`(`*resource_group*, ***kwargs*`)`

New in version 2019.2.0.

List all route tables within a resource group.

Parameters **resource_group** – The resource group name to list route tables within.

CLI Example:

```
salt-call azurearm_network.route_tables_list testgroup
```

`salt.modules.azurearm_network.`**`route_tables_list_all`**`(`***kwargs*`)`

New in version 2019.2.0.

List all route tables within a subscription.

CLI Example:

```
salt-call azurearm_network.route_tables_list_all
```

`salt.modules.azurearm_network.`**`routes_list`**`(`*route_table*, *resource_group*, ***kwargs*`)`

New in version 2019.2.0.

List all routes within a route table.

Parameters

- **route_table** – The route table to query.
- **resource_group** – The resource group name assigned to the route table.

CLI Example:

```
salt-call azurearm_network.routes_list test-rt-table testgroup
```

`salt.modules.azurearm_network.`**`security_rule_create_or_update`**`(`*name*, *access*, *direction*, *priority*, *protocol*, *security_group*, *resource_group*, *source_address_prefix=None*, *destination_address_prefix=None*, *source_port_range=None*, *destination_port_range=None*, *source_address_prefixes=None*, *destination_address_prefixes=None*, *source_port_ranges=None*, *destination_port_ranges=None*, ***kwargs*`)`

New in version 2019.2.0.

Create or update a security rule within a specified network security group.

Parameters

- **`name`** – The name of the security rule to create.
- **`access`** – 'allow' or 'deny'
- **`direction`** – 'inbound' or 'outbound'
- **`priority`** – Integer between 100 and 4096 used for ordering rule application.
- **`protocol`** – 'tcp', 'udp', or '*'
- **`destination_address_prefix`** – The CIDR or destination IP range. Asterix '*' can also be used to match all destination IPs. Default tags such as 'VirtualNetwork', 'AzureLoadBalancer' and 'Internet' can also be used. If this is an ingress rule, specifies where network traffic originates from.
- **`destination_port_range`** – The destination port or range. Integer or range between 0 and 65535. Asterix '*' can also be used to match all ports.
- **`source_address_prefix`** – The CIDR or source IP range. Asterix '*' can also be used to match all source IPs. Default tags such as 'VirtualNetwork', 'AzureLoadBalancer' and 'Internet' can also be used. If this is an ingress rule, specifies where network traffic originates from.
- **`source_port_range`** – The source port or range. Integer or range between 0 and 65535. Asterix '*' can also be used to match all ports.
- **`destination_address_prefixes`** – A list of destination_address_prefix values. This parameter overrides destination_address_prefix and will cause any value entered there to be ignored.
- **`destination_port_ranges`** – A list of destination_port_range values. This parameter overrides destination_port_range and will cause any value entered there to be ignored.
- **`source_address_prefixes`** – A list of source_address_prefix values. This parameter overrides source_address_prefix and will cause any value entered there to be ignored.
- **`source_port_ranges`** – A list of source_port_range values. This parameter overrides source_port_range and will cause any value entered there to be ignored.
- **`security_group`** – The network security group containing the security rule.
- **`resource_group`** – The resource group name assigned to the network security group.

CLI Example:

```
salt-call azurearm_network.security_rule_create_or_update testrule1 allow⊠
↪outbound 101 tcp testnsg testgroup                     source_address_prefix='*'⊠
↪destination_address_prefix=internet source_port_range='*'                  ⊠
↪destination_port_range='1-1024'
```

`salt.modules.azurearm_network.`**`security_rule_delete`**(*security_rule*, *security_group*, *resource_group*, ***kwargs*)

New in version 2019.2.0.

Delete a security rule within a specified security group.

Parameters

- **`name`** – The name of the security rule to delete.
- **`security_group`** – The network security group containing the security rule.

- **`resource_group`** – The resource group name assigned to the network security group.

CLI Example:

```
salt-call azurearm_network.security_rule_delete testrule1 testnsg testgroup
```

`salt.modules.azurearm_network.`**`security_rule_get`**`(`*security_rule, security_group, resource_group, **kwargs*`)`

New in version 2019.2.0.

Get a security rule within a specified network security group.

Parameters

- **`name`** – The name of the security rule to query.
- **`security_group`** – The network security group containing the security rule.
- **`resource_group`** – The resource group name assigned to the network security group.

CLI Example:

```
salt-call azurearm_network.security_rule_get testrule1 testnsg testgroup
```

`salt.modules.azurearm_network.`**`security_rules_list`**`(`*security_group, resource_group, **kwargs*`)`

New in version 2019.2.0.

List security rules within a network security group.

Parameters

- **`security_group`** – The network security group to query.
- **`resource_group`** – The resource group name assigned to the network security group.

CLI Example:

```
salt-call azurearm_network.security_rules_list testnsg testgroup
```

`salt.modules.azurearm_network.`**`subnet_create_or_update`**`(`*name, address_prefix, virtual_network, resource_group, **kwargs*`)`

New in version 2019.2.0.

Create or update a subnet.

Parameters

- **`name`** – The name assigned to the subnet being created or updated.
- **`address_prefix`** – A valid CIDR block within the virtual network.
- **`virtual_network`** – The virtual network name containing the subnet.
- **`resource_group`** – The resource group name assigned to the virtual network.

CLI Example:

```
salt-call azurearm_network.subnet_create_or_update testsubnet
↪'10.0.0.0/24' testnet testgroup
```

`salt.modules.azurearm_network.`**`subnet_delete`**`(`*name, virtual_network, resource_group, **kwargs*`)`

New in version 2019.2.0.

Delete a subnet.

Parameters

- **`name`** – The name of the subnet to delete.
- **`virtual_network`** – The virtual network name containing the subnet.
- **`resource_group`** – The resource group name assigned to the virtual network.

CLI Example:

```
salt-call azurearm_network.subnet_delete testsubnet testnet testgroup
```

`salt.modules.azurearm_network.`**`subnet_get`**`(`*name*, *virtual_network*, *resource_group*, ***kwargs*`)`

New in version 2019.2.0.

Get details about a specific subnet.

Parameters

- **`name`** – The name of the subnet to query.
- **`virtual_network`** – The virtual network name containing the subnet.
- **`resource_group`** – The resource group name assigned to the virtual network.

CLI Example:

```
salt-call azurearm_network.subnet_get testsubnet testnet testgroup
```

`salt.modules.azurearm_network.`**`subnets_list`**`(`*virtual_network*, *resource_group*, ***kwargs*`)`

New in version 2019.2.0.

List all subnets within a virtual network.

Parameters

- **`virtual_network`** – The virtual network name to list subnets within.
- **`resource_group`** – The resource group name assigned to the virtual network.

CLI Example:

```
salt-call azurearm_network.subnets_list testnet testgroup
```

`salt.modules.azurearm_network.`**`usages_list`**`(`*location*, ***kwargs*`)`

New in version 2019.2.0.

List subscription network usage for a location.

Parameters **`location`** – The Azure location to query for network usage.

CLI Example:

```
salt-call azurearm_network.usages_list westus
```

`salt.modules.azurearm_network.`**`virtual_network_create_or_update`**`(`*name*, *address_prefixes*, *resource_group*, ***kwargs*`)`

New in version 2019.2.0.

Create or update a virtual network.

Parameters

- **name** – The name assigned to the virtual network being created or updated.
- **address_prefixes** – A list of CIDR blocks which can be used by subnets within the virtual network.
- **resource_group** – The resource group name assigned to the virtual network.

CLI Example:

```
salt-call azurearm_network.virtual_network_create_or_update
↪testnet ['10.0.0.0/16'] testgroup
```

salt.modules.azurearm_network.**virtual_network_delete**(*name*, *resource_group*, ***kwargs*)

New in version 2019.2.0.

Delete a virtual network.

Parameters

- **name** – The name of the virtual network to delete.
- **resource_group** – The resource group name assigned to the virtual network

CLI Example:

```
salt-call azurearm_network.virtual_network_delete testnet testgroup
```

salt.modules.azurearm_network.**virtual_network_get**(*name*, *resource_group*, ***kwargs*)

New in version 2019.2.0.

Get details about a specific virtual network.

Parameters

- **name** – The name of the virtual network to query.
- **resource_group** – The resource group name assigned to the virtual network.

CLI Example:

```
salt-call azurearm_network.virtual_network_get testnet testgroup
```

salt.modules.azurearm_network.**virtual_networks_list**(*resource_group*, ***kwargs*)

New in version 2019.2.0.

List all virtual networks within a resource group.

Parameters **resource_group** – The resource group name to list virtual networks within.

CLI Example:

```
salt-call azurearm_network.virtual_networks_list testgroup
```

salt.modules.azurearm_network.**virtual_networks_list_all**(***kwargs*)

New in version 2019.2.0.

List all virtual networks within a subscription.

CLI Example:

```
salt-call azurearm_network.virtual_networks_list_all
```

21.9.27 salt.modules.azurearm_resource

Azure (ARM) Resource Execution Module

New in version 2019.2.0.

maintainer <devops@decisionlab.io>

maturity new

depends

- azure >= 2.0.0
- azure-common >= 1.1.8
- azure-mgmt >= 1.0.0
- azure-mgmt-compute >= 1.0.0
- azure-mgmt-network >= 1.7.1
- azure-mgmt-resource >= 1.1.0
- azure-mgmt-storage >= 1.0.0
- azure-mgmt-web >= 0.32.0
- azure-storage >= 0.34.3
- msrestazure >= 0.4.21

platform linux

configuration This module requires Azure Resource Manager credentials to be passed as keyword arguments to every function in order to work properly.

Required provider parameters:

if using username and password:

- `subscription_id`
- `username`
- `password`

if using a service principal:

- `subscription_id`
- `tenant`
- `client_id`
- `secret`

Optional provider parameters:

cloud_environment: **Used to point the cloud driver to different API endpoints, such as Azure GovCloud.**

Possible values:

- `AZURE_PUBLIC_CLOUD` (default)
- `AZURE_CHINA_CLOUD`
- `AZURE_US_GOV_CLOUD`
- `AZURE_GERMAN_CLOUD`

`salt.modules.azurearm_resource.`**`deployment_cancel`**`(`*name*, *resource_group*, ***kwargs*`)`

New in version 2019.2.0.

Cancel a deployment if in 'Accepted' or 'Running' state.

Parameters

- **`name`** – The name of the deployment to cancel.
- **`resource_group`** – The resource group name assigned to the deployment.

CLI Example:

```
salt-call azurearm_resource.deployment_cancel testdeploy testgroup
```

`salt.modules.azurearm_resource.`**`deployment_check_existence`**`(`*name*, *resource_group*, ***kwargs*`)`

New in version 2019.2.0.

Check the existence of a deployment.

Parameters

- **`name`** – The name of the deployment to query.
- **`resource_group`** – The resource group name assigned to the deployment.

CLI Example:

```
salt-call azurearm_resource.deployment_check_existence testdeploy testgroup
```

`salt.modules.azurearm_resource.`**`deployment_create_or_update`**`(`*name*, *resource_group*, *deploy_mode='incremental'*, *debug_setting='none'*, *deploy_params=None*, *parameters_link=None*, *deploy_template=None*, *template_link=None*, ***kwargs*`)`

New in version 2019.2.0.

Deploys resources to a resource group.

Parameters

- **`name`** – The name of the deployment to create or update.
- **`resource_group`** – The resource group name assigned to the deployment.
- **`deploy_mode`** – The mode that is used to deploy resources. This value can be either 'incremental' or 'complete'. In Incremental mode, resources are deployed without deleting existing resources that are not included in the template. In Complete mode, resources are deployed and existing resources in the resource group that are not included in the template are deleted. Be careful when using Complete mode as you may unintentionally delete resources.
- **`debug_setting`** – The debug setting of the deployment. The permitted values are 'none', 'requestContent', 'responseContent', or 'requestContent,responseContent'. By logging information about the request or response, you could potentially expose sensitive data that is retrieved through the deployment operations.

- **deploy_params** – JSON string containing name and value pairs that define the deployment parameters for the template. You use this element when you want to provide the parameter values directly in the request rather than link to an existing parameter file. Use either the parameters_link property or the deploy_params property, but not both.
- **parameters_link** – The URI of a parameters file. You use this element to link to an existing parameters file. Use either the parameters_link property or the deploy_params property, but not both.
- **deploy_template** – JSON string of template content. You use this element when you want to pass the template syntax directly in the request rather than link to an existing template. Use either the template_link property or the deploy_template property, but not both.
- **template_link** – The URI of the template. Use either the template_link property or the deploy_template property, but not both.

CLI Example:

```
salt-call azurearm_resource.deployment_create_or_update testdeploy testgroup
```

`salt.modules.azurearm_resource.`**`deployment_delete`**`(`*name*, *resource_group*, ***kwargs*`)`

New in version 2019.2.0.

Delete a deployment.

Parameters

- **name** – The name of the deployment to delete.
- **resource_group** – The resource group name assigned to the deployment.

CLI Example:

```
salt-call azurearm_resource.deployment_delete testdeploy testgroup
```

`salt.modules.azurearm_resource.`**`deployment_export_template`**`(`*name*, *resource_group*, ***kwargs*`)`

New in version 2019.2.0.

Exports the template used for the specified deployment.

Parameters

- **name** – The name of the deployment to query.
- **resource_group** – The resource group name assigned to the deployment.

CLI Example:

```
salt-call azurearm_resource.deployment_export_template testdeploy testgroup
```

`salt.modules.azurearm_resource.`**`deployment_get`**`(`*name*, *resource_group*, ***kwargs*`)`

New in version 2019.2.0.

Get details about a specific deployment.

Parameters

- **name** – The name of the deployment to query.
- **resource_group** – The resource group name assigned to the deployment.

CLI Example:

```
salt-call azurearm_resource.deployment_get testdeploy testgroup
```

`salt.modules.azurearm_resource.`**`deployment_operation_get`**(*operation*, *deployment*, *resource_group*, ***kwargs*)

New in version 2019.2.0.

Get a deployment operation within a deployment.

Parameters

- **`operation`** – The operation ID of the operation within the deployment.
- **`deployment`** – The name of the deployment containing the operation.
- **`resource_group`** – The resource group name assigned to the deployment.

CLI Example:

```
salt-call azurearm_resource.deployment_operation_get XXXXX testdeploy testgroup
```

`salt.modules.azurearm_resource.`**`deployment_operations_list`**(*name*, *resource_group*, *result_limit=10*, ***kwargs*)

New in version 2019.2.0.

List all deployment operations within a deployment.

Parameters

- **`name`** – The name of the deployment to query.
- **`resource_group`** – The resource group name assigned to the deployment.
- **`result_limit`** – (Default: 10) The limit on the list of deployment operations.

CLI Example:

```
salt-call azurearm_resource.deployment_operations_list testdeploy testgroup
```

`salt.modules.azurearm_resource.`**`deployment_validate`**(*name*, *resource_group*, *deploy_mode=None*, *debug_setting=None*, *deploy_params=None*, *parameters_link=None*, *deploy_template=None*, *template_link=None*, ***kwargs*)

New in version 2019.2.0.

Validates whether the specified template is syntactically correct and will be accepted by Azure Resource Manager.

Parameters

- **`name`** – The name of the deployment to validate.
- **`resource_group`** – The resource group name assigned to the deployment.
- **`deploy_mode`** – The mode that is used to deploy resources. This value can be either 'incremental' or 'complete'. In Incremental mode, resources are deployed without deleting existing resources that are not included in the template. In Complete mode, resources are deployed and existing resources in the resource group that are not included in the template are deleted. Be careful when using Complete mode as you may unintentionally delete resources.

- **debug_setting** – The debug setting of the deployment. The permitted values are 'none', 'requestContent', 'responseContent', or 'requestContent,responseContent'. By logging information about the request or response, you could potentially expose sensitive data that is retrieved through the deployment operations.
- **deploy_params** – JSON string containing name and value pairs that define the deployment parameters for the template. You use this element when you want to provide the parameter values directly in the request rather than link to an existing parameter file. Use either the parameters_link property or the deploy_params property, but not both.
- **parameters_link** – The URI of a parameters file. You use this element to link to an existing parameters file. Use either the parameters_link property or the deploy_params property, but not both.
- **deploy_template** – JSON string of template content. You use this element when you want to pass the template syntax directly in the request rather than link to an existing template. Use either the template_link property or the deploy_template property, but not both.
- **template_link** – The URI of the template. Use either the template_link property or the deploy_template property, but not both.

CLI Example:

```
salt-call azurearm_resource.deployment_validate testdeploy testgroup
```

`salt.modules.azurearm_resource.`**`deployments_list`**(*resource_group*, ***kwargs*)

New in version 2019.2.0.

List all deployments within a resource group.

CLI Example:

```
salt-call azurearm_resource.deployments_list testgroup
```

`salt.modules.azurearm_resource.`**`policy_assignment_create`**(*name*, *scope*, *definition_name*, ***kwargs*)

New in version 2019.2.0.

Create a policy assignment.

Parameters

- **name** – The name of the policy assignment to create.
- **scope** – The scope of the policy assignment.
- **definition_name** – The name of the policy definition to assign.

CLI Example:

```
salt-call azurearm_resource.policy_assignment_create testassign        /
↪subscriptions/bc75htn-a0fhsi-349b-56gh-4fghti-f84852 testpolicy
```

`salt.modules.azurearm_resource.`**`policy_assignment_delete`**(*name*, *scope*, ***kwargs*)

New in version 2019.2.0.

Delete a policy assignment.

Parameters

- **name** – The name of the policy assignment to delete.
- **scope** – The scope of the policy assignment.

CLI Example:

```
salt-call azurearm_resource.policy_assignment_delete testassign          /
↪subscriptions/bc75htn-a0fhsi-349b-56gh-4fghti-f84852
```

`salt.modules.azurearm_resource.`**`policy_assignment_get`**(*name*, *scope*, ***kwargs*)

New in version 2019.2.0.

Get details about a specific policy assignment.

Parameters

- **name** – The name of the policy assignment to query.
- **scope** – The scope of the policy assignment.

CLI Example:

```
salt-call azurearm_resource.policy_assignment_get testassign          /
↪subscriptions/bc75htn-a0fhsi-349b-56gh-4fghti-f84852
```

`salt.modules.azurearm_resource.`**`policy_assignments_list`**(***kwargs*)

New in version 2019.2.0.

List all policy assignments for a subscription.

CLI Example:

```
salt-call azurearm_resource.policy_assignments_list
```

`salt.modules.azurearm_resource.`**`policy_assignments_list_for_resource_group`**(*resource_group*, ***kwargs*)

New in version 2019.2.0.

List all policy assignments for a resource group.

Parameters **resource_group** – The resource group name to list policy assignments within.

CLI Example:

```
salt-call azurearm_resource.policy_assignments_list_for_resource_group testgroup
```

`salt.modules.azurearm_resource.`**`policy_definition_create_or_update`**(*name*, *policy_rule*, ***kwargs*)

New in version 2019.2.0.

Create or update a policy definition.

Parameters

- **name** – The name of the policy definition to create or update.
- **policy_rule** – A dictionary defining the policy rule.

CLI Example:

```
salt-call azurearm_resource.policy_definition_create_or_update testpolicy '{...
↪rule definition..}'
```

`salt.modules.azurearm_resource.`**`policy_definition_delete`**(*name*, ***kwargs*)

New in version 2019.2.0.

Delete a policy definition.

Parameters **name** – The name of the policy definition to delete.

CLI Example:

```
salt-call azurearm_resource.policy_definition_delete testpolicy
```

salt.modules.azurearm_resource.**policy_definition_get**(*name*, ***kwargs*)
New in version 2019.2.0.

Get details about a specific policy definition.

Parameters **name** – The name of the policy definition to query.

CLI Example:

```
salt-call azurearm_resource.policy_definition_get testpolicy
```

salt.modules.azurearm_resource.**policy_definitions_list**(*hide_builtin=False*, ***kwargs*)
New in version 2019.2.0.

List all policy definitions for a subscription.

Parameters **hide_builtin** – Boolean which will filter out BuiltIn policy definitions from the result.

CLI Example:

```
salt-call azurearm_resource.policy_definitions_list
```

salt.modules.azurearm_resource.**resource_group_check_existence**(*name*, ***kwargs*)
New in version 2019.2.0.

Check for the existence of a named resource group in the current subscription.

Parameters **name** – The resource group name to check.

CLI Example:

```
salt-call azurearm_resource.resource_group_check_existence testgroup
```

salt.modules.azurearm_resource.**resource_group_create_or_update**(*name*, *location*, ***kwargs*)
New in version 2019.2.0.

Create or update a resource group in a given location.

Parameters

- **name** – The name of the resource group to create or update.
- **location** – The location of the resource group. This value is not able to be updated once the resource group is created.

CLI Example:

```
salt-call azurearm_resource.resource_group_create_or_update testgroup westus
```

salt.modules.azurearm_resource.**resource_group_delete**(*name*, ***kwargs*)
New in version 2019.2.0.

Delete a resource group from the subscription.

Parameters **name** – The resource group name to delete.

CLI Example:

```
salt-call azurearm_resource.resource_group_delete testgroup
```

`salt.modules.azurearm_resource.`**`resource_group_get`**`(`*name*, ***kwargs*`)`

New in version 2019.2.0.

Get a dictionary representing a resource group's properties.

Parameters **`name`** – The resource group name to get.

CLI Example:

```
salt-call azurearm_resource.resource_group_get testgroup
```

`salt.modules.azurearm_resource.`**`resource_groups_list`**`(`***kwargs*`)`

New in version 2019.2.0.

List all resource groups within a subscription.

CLI Example:

```
salt-call azurearm_resource.resource_groups_list
```

`salt.modules.azurearm_resource.`**`subscription_get`**`(`*subscription_id=None*, ***kwargs*`)`

New in version 2019.2.0.

Get details about a subscription.

Parameters **`subscription_id`** – The ID of the subscription to query.

CLI Example:

```
salt-call azurearm_resource.subscription_get XXXXXXXX
```

`salt.modules.azurearm_resource.`**`subscriptions_list`**`(`***kwargs*`)`

New in version 2019.2.0.

List all subscriptions for a tenant.

CLI Example:

```
salt-call azurearm_resource.subscriptions_list
```

`salt.modules.azurearm_resource.`**`subscriptions_list_locations`**`(`*subscription_id=None*, ***kwargs*`)`

New in version 2019.2.0.

List all locations for a subscription.

Parameters **`subscription_id`** – The ID of the subscription to query.

CLI Example:

```
salt-call azurearm_resource.subscriptions_list_locations XXXXXXXX
```

`salt.modules.azurearm_resource.`**`tenants_list`**`(`***kwargs*`)`

New in version 2019.2.0.

List all tenants for your account.

CLI Example:

```
salt-call azurearm_resource.tenants_list
```

21.9.28 salt.modules.bamboohr

Support for BambooHR

New in version 2015.8.0.

Requires a `subdomain` and an `apikey` in `/etc/salt/minion`:

```
bamboohr:
  apikey: 012345678901234567890
  subdomain: mycompany
```

`salt.modules.bamboohr.`**`list_employees`**`(`*order_by='id'*`)`

Show all employees for this company.

CLI Example:

salt myminion bamboohr.list_employees

By default, the return data will be keyed by ID. However, it can be ordered by any other field. Keep in mind that if the field that is chosen contains duplicate values (i.e., location is used, for a company which only has one location), then each duplicate value will be overwritten by the previous. Therefore, it is advisable to only sort by fields that are guaranteed to be unique.

CLI Examples:

salt myminion bamboohr.list_employees order_by=id salt myminion bamboohr.list_employees order_by=displayName salt myminion bamboohr.list_employees order_by=workEmail

`salt.modules.bamboohr.`**`list_meta_fields`**`()`

Show all meta data fields for this company.

CLI Example:

salt myminion bamboohr.list_meta_fields

`salt.modules.bamboohr.`**`list_users`**`(`*order_by='id'*`)`

Show all users for this company.

CLI Example:

salt myminion bamboohr.list_users

By default, the return data will be keyed by ID. However, it can be ordered by any other field. Keep in mind that if the field that is chosen contains duplicate values (i.e., location is used, for a company which only has one location), then each duplicate value will be overwritten by the previous. Therefore, it is advisable to only sort by fields that are guaranteed to be unique.

CLI Examples:

salt myminion bamboohr.list_users order_by=id salt myminion bamboohr.list_users order_by=email

`salt.modules.bamboohr.`**`show_employee`**`(`*emp_id, fields=None*`)`

Show all employees for this company.

CLI Example:

salt myminion bamboohr.show_employee 1138

By default, the fields normally returned from bamboohr.list_employees are returned. These fields are:

- canUploadPhoto
- department
- displayName
- firstName
- id
- jobTitle
- lastName
- location
- mobilePhone
- nickname
- photoUploaded
- photoUrl
- workEmail
- workPhone
- workPhoneExtension

If needed, a different set of fields may be specified, separated by commas:

CLI Example:

> salt myminion bamboohr.show_employee 1138 displayName,dateOfBirth

A list of available fields can be found at http://www.bamboohr.com/api/documentation/employees.php

`salt.modules.bamboohr.`**`update_employee`**(*emp_id*, *key=None*, *value=None*, *items=None*)

Update one or more items for this employee. Specifying an empty value will clear it for that employee.

CLI Examples:

> salt myminion bamboohr.update_employee 1138 nickname Curly salt myminion bamboohr.update_employee 1138 nickname ” salt myminion bamboohr.update_employee 1138 items=’{”nickname”: ”Curly”} salt myminion bamboohr.update_employee 1138 items=’{”nickname”: ””}

21.9.29 salt.modules.bcache module

Module for managing BCache sets

BCache is a block-level caching mechanism similar to ZFS L2ARC/ZIL, dm-cache and fscache. It works by formatting one block device as a cache set, then adding backend devices (which need to be formatted as such) to the set and activating them.

It’s available in Linux mainline kernel since 3.10

https://www.kernel.org/doc/Documentation/bcache.txt

This module needs the bcache userspace tools to function.

`salt.modules.bcache.`**`attach_`**(*dev=None*)

Attach a backing devices to a cache set If no dev is given, all backing devices will be attached.

CLI example:

```
salt '*' bcache.attach sdc
salt '*' bcache.attach /dev/bcache1
```

Returns bool or None if nuttin' happened

`salt.modules.bcache.`**`back_make`**(*dev*, *cache_mode='writeback'*, *force=False*, *attach=True*, *bucket_size=None*)

Create a backing device for attachment to a set. Because the block size must be the same, a cache set already needs to exist.

CLI example:

```
salt '*' bcache.back_make sdc cache_mode=writeback attach=True
```

Parameters

- **`cache_mode`** – writethrough, writeback, writearound or none.
- **`force`** – Overwrite existing bcaches
- **`attach`** – Immediately attach the backing device to the set
- **`bucket_size`** – Size of a bucket (see kernel doc)

`salt.modules.bcache.`**`cache_make`**(*dev*, *reserved=None*, *force=False*, *block_size=None*, *bucket_size=None*, *attach=True*)

Create BCache cache on a block device. If blkdiscard is available the entire device will be properly cleared in advance.

CLI example:

```
salt '*' bcache.cache_make sdb reserved=10% block_size=4096
```

Parameters

- **`reserved`** – if dev is a full device, create a partition table with this size empty.

  Note: this increases the amount of reserved space available to SSD garbage collectors, potentially (vastly) increasing performance

- **`block_size`** – Block size of the cache; defaults to devices' logical block size
- **`force`** – Overwrite existing BCache sets
- **`attach`** – Attach all existing backend devices immediately

`salt.modules.bcache.`**`config_`**(*dev=None*, ***kwargs*)

Show or update config of a bcache device.

If no device is given, operate on the cache set itself.

CLI example:

```
salt '*' bcache.config
salt '*' bcache.config bcache1
salt '*' bcache.config errors=panic journal_delay_ms=150
salt '*' bcache.config bcache1 cache_mode=writeback writeback_percent=15
```

Returns config or True/False

`salt.modules.bcache.`**`detach`**`(`*dev=None*`)`

Detach a backing device(s) from a cache set If no dev is given, all backing devices will be attached.

Detaching a backing device will flush it's write cache. This should leave the underlying device in a consistent state, but might take a while.

CLI example:

```
salt '*' bcache.detach sdc
salt '*' bcache.detach bcache1
```

`salt.modules.bcache.`**`device`**`(`*dev*, *stats=False*, *config=False*, *internals=False*, *superblock=False*`)`

Check the state of a single bcache device

CLI example:

```
salt '*' bcache.device bcache0
salt '*' bcache.device /dev/sdc stats=True
```

Parameters

- **`stats`** – include statistics
- **`settings`** – include all settings
- **`internals`** – include all internals
- **`superblock`** – include superblock info

`salt.modules.bcache.`**`start`**`()`

Trigger a start of the full bcache system through udev.

CLI example:

```
salt '*' bcache.start
```

`salt.modules.bcache.`**`status`**`(`*stats=False*, *config=False*, *internals=False*, *superblock=False*, *alldevs=False*`)`

Show the full status of the BCache system and optionally all it's involved devices

CLI example:

```
salt '*' bcache.status
salt '*' bcache.status stats=True
salt '*' bcache.status internals=True alldevs=True
```

Parameters

- **`stats`** – include statistics
- **`config`** – include settings
- **`internals`** – include internals
- **`superblock`** – include superblock

`salt.modules.bcache.`**`stop`**`(`*dev=None*`)`

Stop a bcache device If no device is given, all backing devices will be detached from the cache, which will subsequently be stopped.

> **Warning:** 'Stop' on an individual backing device means hard-stop; no attempt at flushing will be done and the bcache device will seemingly 'disappear' from the device lists

CLI example:

```
salt '*' bcache.stop
```

`salt.modules.bcache.`**`super_`**(*dev*)

Read out BCache SuperBlock

CLI example:

```
salt '*' bcache.device bcache0
salt '*' bcache.device /dev/sdc
```

`salt.modules.bcache.`**`uuid`**(*dev=None*)

Return the bcache UUID of a block device. If no device is given, the Cache UUID is returned.

CLI example:

```
salt '*' bcache.uuid
salt '*' bcache.uuid /dev/sda
salt '*' bcache.uuid bcache0
```

21.9.30 salt.modules.beacons

Module for managing the Salt beacons on a minion

New in version 2015.8.0.

`salt.modules.beacons.`**`add`**(*name*, *beacon_data*, ***kwargs*)

Add a beacon on the minion

Parameters

- **`name`** – Name of the beacon to configure
- **`beacon_data`** – Dictionary or list containing configuration for beacon.

Returns Boolean and status message on success or failure of add.

CLI Example:

```
salt '*' beacons.add ps "[{'processes': {'salt-master': 'stopped', 'apache2':
↪'stopped'}}]"
```

`salt.modules.beacons.`**`delete`**(*name*, ***kwargs*)

Delete a beacon item

Parameters **`name`** – Name of the beacon to delete

Returns Boolean and status message on success or failure of delete.

CLI Example:

```
salt '*' beacons.delete ps

salt '*' beacons.delete load
```

salt.modules.beacons.**disable**(***kwargs*)

Disable all beacons jobs on the minion

Returns Boolean and status message on success or failure of disable.

CLI Example:

```
salt '*' beacons.disable
```

salt.modules.beacons.**disable_beacon**(*name*, ***kwargs*)

Disable beacon on the minion

Name Name of the beacon to disable.

Returns Boolean and status message on success or failure of disable.

CLI Example:

```
salt '*' beacons.disable_beacon ps
```

salt.modules.beacons.**enable**(***kwargs*)

Enable all beacons on the minion

Returns Boolean and status message on success or failure of enable.

Return type bool

CLI Example:

```
salt '*' beacons.enable
```

salt.modules.beacons.**enable_beacon**(*name*, ***kwargs*)

Enable beacon on the minion

Name Name of the beacon to enable.

Returns Boolean and status message on success or failure of enable.

CLI Example:

```
salt '*' beacons.enable_beacon ps
```

salt.modules.beacons.**list_**(*return_yaml=True*, *include_pillar=True*, *include_opts=True*, ***kwargs*)

List the beacons currently configured on the minion

Parameters

- **`return_yaml`** – Whether to return YAML formatted output, default `True`
- **`include_pillar`** – Whether to include beacons that are configured in pillar, default is `True`.
- **`include_opts`** – Whether to include beacons that are configured in opts, default is `True`.

Returns List of currently configured Beacons.

CLI Example:

```
salt '*' beacons.list
```

salt.modules.beacons.**list_available**(*return_yaml=True*, ***kwargs*)

List the beacons currently available on the minion

Parameters **`return_yaml`** – Whether to return YAML formatted output, default `True`

Returns List of currently configured Beacons.

CLI Example:

```
salt '*' beacons.list_available
```

`salt.modules.beacons.`**`modify`**`(`*name*, *beacon_data*, ***kwargs*`)`

Modify an existing beacon

Parameters

- **`name`** – Name of the beacon to configure
- **`beacon_data`** – Dictionary or list containing updated configuration for beacon.

Returns Boolean and status message on success or failure of modify.

CLI Example:

```
salt '*' beacons.modify ps "[{'salt-master': 'stopped'}, {'apache2': 'stopped'}]"
```

`salt.modules.beacons.`**`reset`**`(`***kwargs*`)`

Resest beacon configuration on the minion

CLI Example:

```
salt '*' beacons.reset
```

`salt.modules.beacons.`**`save`**`(`***kwargs*`)`

Save all beacons on the minion

Returns Boolean and status message on success or failure of save.

CLI Example:

```
salt '*' beacons.save
```

21.9.31 salt.modules.bigip

An execution module which can manipulate an f5 bigip via iControl REST

maturity develop

platform f5_bigip_11.6

`salt.modules.bigip.`**`add_pool_member`**`(`*hostname*, *username*, *password*, *name*, *member*`)`

A function to connect to a bigip device and add a new member to an existing pool.

hostname The host/address of the bigip device

username The iControl REST username

password The iControl REST password

name The name of the pool to modify

member The name of the member to add i.e. 10.1.1.2:80

CLI Example:

```
salt '*' bigip.add_pool_members bigip admin admin my-pool 10.2.2.1:80
```

`salt.modules.bigip.`**`commit_transaction`**(*hostname, username, password, label*)

A function to connect to a bigip device and commit an existing transaction.

hostname The host/address of the bigip device

username The iControl REST username

password The iControl REST password

label the label of this transaction stored within the grain: `bigip_f5_trans:<label>`

CLI Example:

```
salt '*' bigip.commit_transaction bigip admin admin my_transaction
```

`salt.modules.bigip.`**`create_monitor`**(*hostname, username, password, monitor_type, name, **kwargs*)

A function to connect to a bigip device and create a monitor.

hostname The host/address of the bigip device

username The iControl REST username

password The iControl REST password

monitor_type The type of monitor to create

name The name of the monitor to create

kwargs Consult F5 BIGIP user guide for specific options for each monitor type. Typically, tmsh arg names are used.

CLI Example:

```
salt '*' bigip.create_monitor bigip admin admin http my-http-monitor timeout=10
↪interval=5
```

`salt.modules.bigip.`**`create_node`**(*hostname, username, password, name, address, trans_label=None*)

A function to connect to a bigip device and create a node.

hostname The host/address of the bigip device

username The iControl REST username

password The iControl REST password

name The name of the node

address The address of the node

trans_label The label of the transaction stored within the grain: `bigip_f5_trans:<label>`

CLI Example:

```
salt '*' bigip.create_node bigip admin admin 10.1.1.2
```

`salt.modules.bigip.`**`create_pool`**`(`*hostname, username, password, name, members=None, allow_nat=None, allow_snat=None, description=None, gateway_failsafe_device=None, ignore_persisted_weight=None, ip_tos_to_client=None, ip_tos_to_server=None, link_qos_to_client=None, link_qos_to_server=None, load_balancing_mode=None, min_active_members=None, min_up_members=None, min_up_members_action=None, min_up_members_checking=None, monitor=None, profiles=None, queue_depth_limit=None, queue_on_connection_limit=None, queue_time_limit=None, reselect_tries=None, service_down_action=None, slow_ramp_time=None*`)`

A function to connect to a bigip device and create a pool.

hostname The host/address of the bigip device

username The iControl REST username

password The iControl REST password

name The name of the pool to create.

members List of comma delimited pool members to add to the pool. i.e. 10.1.1.1:80,10.1.1.2:80,10.1.1.3:80

allow_nat [yes | no]

allow_snat [yes | no]

description [string]

gateway_failsafe_device [string]

ignore_persisted_weight [enabled | disabled]

ip_tos_to_client [pass-through | [integer]]

ip_tos_to_server [pass-through | [integer]]

link_qos_to_client [pass-through | [integer]]

link_qos_to_server [pass-through | [integer]]

load_balancing_mode [dynamic-ratio-member | dynamic-ratio-node | fastest-app-response | fastest-node | least-connections-members | least-connections-node | least-sessions | observed-member | observed-node | predictive-member | predictive-node | ratio-least-connections-member | ratio-least-connections-node | ratio-member | ratio-node | ratio-session | round-robin | weighted-least-connections-member | weighted-least-connections-node]

min_active_members [integer]

min_up_members [integer]

min_up_members_action [failover | reboot | restart-all]

min_up_members_checking [enabled | disabled]

monitor [name]

profiles [none | profile_name]

queue_depth_limit [integer]

queue_on_connection_limit [enabled | disabled]

queue_time_limit [integer]

reselect_tries [integer]

service_down_action [drop | none | reselect | reset]

slow_ramp_time [integer]

CLI Example:

```
salt '*' bigip.create_pool bigip admin admin my-pool 10.1.1.1:80,10.1.1.2:80,10.1.
↪1.3:80 monitor=http
```

`salt.modules.bigip.`**`create_profile`**(*hostname*, *username*, *password*, *profile_type*, *name*, ***kwargs*)

A function to connect to a bigip device and create a profile.

hostname The host/address of the bigip device

username The iControl REST username

password The iControl REST password

profile_type The type of profile to create

name The name of the profile to create

kwargs `[ arg=val ] ... [arg=key1:val1,key2:val2] ...`

Consult F5 BIGIP user guide for specific options for each monitor type. Typically, tmsh arg names are used.

Creating Complex Args Profiles can get pretty complicated in terms of the amount of possible config options. Use the following shorthand to create complex arguments such as lists, dictionaries, and lists of dictionaries. An option is also provided to pass raw json as well.

lists `[i,i,i]`: `param='item1,item2,item3'`

Dictionary `[k:v,k:v,k,v]`: `param='key-1:val-1,key-2:val2,key-3:va-3'`

List of Dictionaries `[k:v,k:v|k:v,k:v|k:v,k:v]`: `param='key-1:val-1, key-2:val-2|key-1:val-1,key-2:val-2|key-1:val-1,key-2:val-2'`

JSON: `'j{ ... }j'`: `cert-key-chain='j{ "default": { "cert": "default.crt", "chain": "default.crt", "key": "default.key" } }j'`

Escaping Delimiters: Use `\,` or `\:` or `\|` to escape characters which shouldn't be treated as delimiters i.e. `ciphers='DEFAULT\:!SSLv3'`

CLI Examples:

```
salt '*' bigip.create_profile bigip admin admin http my-http-profile defaultsFrom=
↪'/Common/http'
salt '*' bigip.create_profile bigip admin admin http my-http-profile defaultsFrom=
↪'/Common/http' \
    enforcement=maxHeaderCount:3200,maxRequests:10
```

`salt.modules.bigip.`**`create_virtual`**`(`*hostname, username, password, name, destination, pool=None, address_status=None, auto_lasthop=None, bwc_policy=None, cmp_enabled=None, connection_limit=None, dhcp_relay=None, description=None, fallback_persistence=None, flow_eviction_policy=None, gtm_score=None, ip_forward=None, ip_protocol=None, internal=None, twelve_forward=None, last_hop_pool=None, mask=None, mirror=None, nat64=None, persist=None, profiles=None, policies=None, rate_class=None, rate_limit=None, rate_limit_mode=None, rate_limit_dst=None, rate_limit_src=None, rules=None, related_rules=None, reject=None, source=None, source_address_translation=None, source_port=None, state=None, traffic_classes=None, translate_address=None, translate_port=None, vlans=None*`)`

A function to connect to a bigip device and create a virtual server.

hostname The host/address of the bigip device

username The iControl REST username

password The iControl REST password

name The name of the virtual to create

destination [[virtual_address_name:port] | [ipv4:port] | [ipv6.port]]

pool [[pool_name] | none]

address_status [yes | no]

auto_lasthop [default | enabled | disabled]

bwc_policy [none] | string]

cmp_enabled [yes | no]

dhcp_relay [yes | no]

connection_limit [integer]

description [string]

state [disabled | enabled]

fallback_persistence [none | [profile name]]

flow_eviction_policy [none | [eviction policy name]]

gtm_score [integer]

ip_forward [yes | no]

ip_protocol [any | protocol]

internal [yes | no]

twelve_forward (12-forward) [yes | no]

last_hop-pool [[pool_name] | none]

mask { [ipv4] | [ipv6] }

mirror { [disabled | enabled | none] }

nat64 [enabled | disabled]

persist [none | profile1,profile2,profile3 …]

profiles [none | default | profile1,profile2,profile3 ...]

policies [none | default | policy1,policy2,policy3 ...]

rate_class [name]

rate_limit [integer]

rate_limit_mode [destination | object | object-destination | object-source | object-source-destination | source | source-destination]

rate_limit_dst [integer]

rate_limitçsrc [integer]

rules [none | [rule_one,rule_two ...]]

related_rules [none | [rule_one,rule_two ...]]

reject [yes | no]

source { [ipv4[/prefixlen]] | [ipv6[/prefixlen]] }

source_address_translation [none | snat:pool_name | lsn | automap]

source_port [change | preserve | preserve-strict]

state [enabled | disabled]

traffic_classes [none | default | class_one,class_two ...]

translate_address [enabled | disabled]

translate_port [enabled | disabled]

vlans [none | default | [enabled|disabled]:vlan1,vlan2,vlan3 ...]

CLI Examples:

```
salt '*' bigip.create_virtual bigip admin admin my-virtual-3 26.2.2.5:80 \
    pool=my-http-pool-http profiles=http,tcp

salt '*' bigip.create_virtual bigip admin admin my-virtual-3 43.2.2.5:80 \
    pool=test-http-pool-http profiles=http,websecurity persist=cookie,hash \
    policies=asm_auto_l7_policy__http-virtual \
    rules=_sys_APM_ExchangeSupport_helper,_sys_https_redirect \
    related_rules=_sys_APM_activesync,_sys_APM_ExchangeSupport_helper \
    source_address_translation=snat:my-snat-pool \
    translate_address=enabled translate_port=enabled \
    traffic_classes=my-class,other-class \
    vlans=enabled:external,internal
```

`salt.modules.bigip.`**`delete_monitor`**(*hostname, username, password, monitor_type, name*)

A function to connect to a bigip device and delete an existing monitor.

hostname The host/address of the bigip device

username The iControl REST username

password The iControl REST password

monitor_type The type of monitor to delete

name The name of the monitor to delete

CLI Example:

```
salt '*' bigip.delete_monitor bigip admin admin http my-http-monitor
```

`salt.modules.bigip.`**`delete_node`**(*hostname, username, password, name, trans_label=None*)

A function to connect to a bigip device and delete a specific node.

hostname The host/address of the bigip device

username The iControl REST username

password The iControl REST password

name The name of the node which will be deleted.

trans_label The label of the transaction stored within the grain: `bigip_f5_trans:<label>`

CLI Example:

```
salt '*' bigip.delete_node bigip admin admin my-node
```

`salt.modules.bigip.`**`delete_pool`**(*hostname, username, password, name*)

A function to connect to a bigip device and delete a specific pool.

hostname The host/address of the bigip device

username The iControl REST username

password The iControl REST password

name The name of the pool which will be deleted

CLI Example:

```
salt '*' bigip.delete_node bigip admin admin my-pool
```

`salt.modules.bigip.`**`delete_pool_member`**(*hostname, username, password, name, member*)

A function to connect to a bigip device and delete a specific pool.

hostname The host/address of the bigip device

username The iControl REST username

password The iControl REST password

name The name of the pool to modify

member The name of the pool member to delete

CLI Example:

```
salt '*' bigip.delete_pool_member bigip admin admin my-pool 10.2.2.2:80
```

`salt.modules.bigip.`**`delete_profile`**(*hostname, username, password, profile_type, name*)

A function to connect to a bigip device and delete an existing profile.

hostname The host/address of the bigip device

username The iControl REST username

password The iControl REST password

profile_type The type of profile to delete

name The name of the profile to delete

CLI Example:

```
salt '*' bigip.delete_profile bigip admin admin http my-http-profile
```

salt.modules.bigip.**delete_transaction**(*hostname*, *username*, *password*, *label*)
A function to connect to a bigip device and delete an existing transaction.

hostname The host/address of the bigip device

username The iControl REST username

password The iControl REST password

label The label of this transaction stored within the grain: `bigip_f5_trans:<label>`

CLI Example:

```
salt '*' bigip.delete_transaction bigip admin admin my_transaction
```

salt.modules.bigip.**delete_virtual**(*hostname*, *username*, *password*, *name*)
A function to connect to a bigip device and delete a specific virtual.

hostname The host/address of the bigip device

username The iControl REST username

password The iControl REST password

name The name of the virtual to delete

CLI Example:

```
salt '*' bigip.delete_virtual bigip admin admin my-virtual
```

salt.modules.bigip.**list_monitor**(*hostname*, *username*, *password*, *monitor_type*, *name=None*)
A function to connect to a bigip device and list an existing monitor. If no name is provided than all monitors of the specified type will be listed.

hostname The host/address of the bigip device

username The iControl REST username

password The iControl REST password

monitor_type The type of monitor(s) to list

name The name of the monitor to list

CLI Example:

```
salt '*' bigip.list_monitor bigip admin admin http my-http-monitor
```

salt.modules.bigip.**list_node**(*hostname*, *username*, *password*, *name=None*, *trans_label=None*)
A function to connect to a bigip device and list all nodes or a specific node.

hostname The host/address of the bigip device

username The iControl REST username

password The iControl REST password

name The name of the node to list. If no name is specified than all nodes will be listed.

trans_label The label of the transaction stored within the grain: `bigip_f5_trans:<label>`

CLI Example:

```
salt '*' bigip.list_node bigip admin admin my-node
```

`salt.modules.bigip.`**`list_pool`**(*hostname*, *username*, *password*, *name=None*)

A function to connect to a bigip device and list all pools or a specific pool.

hostname The host/address of the bigip device

username The iControl REST username

password The iControl REST password

name The name of the pool to list. If no name is specified then all pools will be listed.

CLI Example:

```
salt '*' bigip.list_pool bigip admin admin my-pool
```

`salt.modules.bigip.`**`list_profile`**(*hostname*, *username*, *password*, *profile_type*, *name=None*)

A function to connect to a bigip device and list an existing profile. If no name is provided than all profiles of the specified type will be listed.

hostname The host/address of the bigip device

username The iControl REST username

password The iControl REST password

profile_type The type of profile(s) to list

name The name of the profile to list

CLI Example:

```
salt '*' bigip.list_profile bigip admin admin http my-http-profile
```

`salt.modules.bigip.`**`list_transaction`**(*hostname*, *username*, *password*, *label*)

A function to connect to a bigip device and list an existing transaction.

hostname The host/address of the bigip device

username The iControl REST username

password The iControl REST password

label the label of this transaction stored within the grain: `bigip_f5_trans:<label>`

CLI Example:

```
salt '*' bigip.list_transaction bigip admin admin my_transaction
```

`salt.modules.bigip.`**`list_virtual`**(*hostname*, *username*, *password*, *name=None*)

A function to connect to a bigip device and list all virtuals or a specific virtual.

hostname The host/address of the bigip device

username The iControl REST username

password The iControl REST password

name The name of the virtual to list. If no name is specified than all virtuals will be listed.

CLI Example:

```
salt '*' bigip.list_virtual bigip admin admin my-virtual
```

`salt.modules.bigip.`**`modify_monitor`**`(`*hostname, username, password, monitor_type, name, **kwargs*`)`

A function to connect to a bigip device and modify an existing monitor.

hostname The host/address of the bigip device

username The iControl REST username

password The iControl REST password

monitor_type The type of monitor to modify

name The name of the monitor to modify

kwargs Consult F5 BIGIP user guide for specific options for each monitor type. Typically, tmsh arg names are used.

CLI Example:

```
salt '*' bigip.modify_monitor bigip admin admin http my-http-monitor  timout=16␣
↪interval=6
```

`salt.modules.bigip.`**`modify_node`**`(`*hostname, username, password, name, connection_limit=None, description=None, dynamic_ratio=None, logging=None, monitor=None, rate_limit=None, ratio=None, session=None, state=None, trans_label=None*`)`

A function to connect to a bigip device and modify an existing node.

hostname The host/address of the bigip device

username The iControl REST username

password The iControl REST password

name The name of the node to modify

connection_limit [integer]

description [string]

dynamic_ratio [integer]

logging [enabled | disabled]

monitor [[name] | none | default]

rate_limit [integer]

ratio [integer]

session [user-enabled | user-disabled]

state [user-down | user-up]

trans_label The label of the transaction stored within the grain: `bigip_f5_trans:<label>`

CLI Example:

```
salt '*' bigip.modify_node bigip admin admin 10.1.1.2 ratio=2 logging=enabled
```

`salt.modules.bigip.`**`modify_pool`**`(`*hostname, username, password, name, allow_nat=None, allow_snat=None, description=None, gateway_failsafe_device=None, ignore_persisted_weight=None, ip_tos_to_client=None, ip_tos_to_server=None, link_qos_to_client=None, link_qos_to_server=None, load_balancing_mode=None, min_active_members=None, min_up_members=None, min_up_members_action=None, min_up_members_checking=None, monitor=None, profiles=None, queue_depth_limit=None, queue_on_connection_limit=None, queue_time_limit=None, reselect_tries=None, service_down_action=None, slow_ramp_time=None*`)`

A function to connect to a bigip device and modify an existing pool.

hostname The host/address of the bigip device

username The iControl REST username

password The iControl REST password

name The name of the pool to modify.

allow_nat [yes | no]

allow_snat [yes | no]

description [string]

gateway_failsafe_device [string]

ignore_persisted_weight [yes | no]

ip_tos_to_client [pass-through | [integer]]

ip_tos_to_server [pass-through | [integer]]

link_qos_to_client [pass-through | [integer]]

link_qos_to_server [pass-through | [integer]]

load_balancing_mode [dynamic-ratio-member | dynamic-ratio-node | fastest-app-response | fastest-node | least-connections-members | least-connections-node | least-sessions | observed-member | observed-node | predictive-member | predictive-node | ratio-least-connections-member | ratio-least-connections-node | ratio-member | ratio-node | ratio-session | round-robin | weighted-least-connections-member | weighted-least-connections-node]

min_active_members [integer]

min_up_members [integer]

min_up_members_action [failover | reboot | restart-all]

min_up_members_checking [enabled | disabled]

monitor [name]

profiles [none | profile_name]

queue_on_connection_limit [enabled | disabled]

queue_depth_limit [integer]

queue_time_limit [integer]

reselect_tries [integer]

service_down_action [drop | none | reselect | reset]

slow_ramp_time [integer]

CLI Example:

```
salt '*' bigip.modify_pool bigip admin admin my-pool 10.1.1.1:80,10.1.1.2:80,10.1.
↪1.3:80 min_active_members=1
```

`salt.modules.bigip.`**`modify_pool_member`**(*hostname, username, password, name, member, connection_limit=None, description=None, dynamic_ratio=None, inherit_profile=None, logging=None, monitor=None, priority_group=None, profiles=None, rate_limit=None, ratio=None, session=None, state=None*)

A function to connect to a bigip device and modify an existing member of a pool.

hostname The host/address of the bigip device

username The iControl REST username

password The iControl REST password

name The name of the pool to modify

member The name of the member to modify i.e. 10.1.1.2:80

connection_limit [integer]

description [string]

dynamic_ratio [integer]

inherit_profile [enabled | disabled]

logging [enabled | disabled]

monitor [name]

priority_group [integer]

profiles [none | profile_name]

rate_limit [integer]

ratio [integer]

session [user-enabled | user-disabled]

state [user-up | user-down]

CLI Example:

```
salt '*' bigip.modify_pool_member bigip admin admin my-pool 10.2.2.1:80 state=use-
↪down session=user-disabled
```

`salt.modules.bigip.`**`modify_profile`**(*hostname, username, password, profile_type, name, **kwargs*)

A function to connect to a bigip device and create a profile.

A function to connect to a bigip device and create a profile.

hostname The host/address of the bigip device

username The iControl REST username

password The iControl REST password

profile_type The type of profile to create

name The name of the profile to create

kwargs `[ arg=val ] ... [arg=key1:val1,key2:val2] ...`

Consult F5 BIGIP user guide for specific options for each monitor type. Typically, tmsh arg names are used.

Creating Complex Args

Profiles can get pretty complicated in terms of the amount of possible config options. Use the following shorthand to create complex arguments such as lists, dictionaries, and lists of dictionaries. An option is also provided to pass raw json as well.

lists `[i,i,i]`: `param='item1,item2,item3'`

Dictionary `[k:v,k:v,k,v]`: `param='key-1:val-1,key-2:val2,key-3:va-3'`

List of Dictionaries `[k:v,k:v|k:v,k:v|k:v,k:v]`: `param='key-1:val-1, key-2:val-2|key-1:val-1,key-2:val-2|key-1:val-1,key-2:val-2'`

JSON: `'j{ ... }j'`: `cert-key-chain='j{ "default": { "cert": "default.crt", "chain": "default.crt", "key": "default.key" } }j'`

Escaping Delimiters: Use `\,` or `\:` or `\|` to escape characters which shouldn't be treated as delimiters i.e. `ciphers='DEFAULT\:!SSLv3'`

CLI Examples:

```
salt '*' bigip.modify_profile bigip admin admin http my-http-profile defaultsFrom=
↪'/Common/http'

salt '*' bigip.modify_profile bigip admin admin http my-http-profile defaultsFrom=
↪'/Common/http' \
    enforcement=maxHeaderCount:3200,maxRequests:10

salt '*' bigip.modify_profile bigip admin admin client-ssl my-client-ssl-1⊠
↪retainCertificate=false \
    ciphers='DEFAULT\:!SSLv3'
    cert_key_chain='j{ "default": { "cert": "default.crt", "chain": "default.crt",
↪ "key": "default.key" } }j'
```

`salt.modules.bigip.`**`modify_virtual`**(*hostname, username, password, name, destination=None, pool=None, address_status=None, auto_lasthop=None, bwc_policy=None, cmp_enabled=None, connection_limit=None, dhcp_relay=None, description=None, fallback_persistence=None, flow_eviction_policy=None, gtm_score=None, ip_forward=None, ip_protocol=None, internal=None, twelve_forward=None, last_hop_pool=None, mask=None, mirror=None, nat64=None, persist=None, profiles=None, policies=None, rate_class=None, rate_limit=None, rate_limit_mode=None, rate_limit_dst=None, rate_limit_src=None, rules=None, related_rules=None, reject=None, source=None, source_address_translation=None, source_port=None, state=None, traffic_classes=None, translate_address=None, translate_port=None, vlans=None*)

A function to connect to a bigip device and modify an existing virtual server.

hostname The host/address of the bigip device

username The iControl REST username

password The iControl REST password

name The name of the virtual to modify

destination [[virtual_address_name:port] | [ipv4:port] | [ipv6.port]]

pool [[pool_name] | none]

address_status [yes | no]

auto_lasthop [default | enabled | disabled]

bwc_policy [none] | string]

cmp_enabled [yes | no]

dhcp_relay [yes | no}

connection_limit [integer]

description [string]

state [disabled | enabled]

fallback_persistence [none | [profile name]]

flow_eviction_policy [none | [eviction policy name]]

gtm_score [integer]

ip_forward [yes | no]

ip_protocol [any | protocol]

internal [yes | no]

twelve_forward (12-forward) [yes | no]

last_hop-pool [[pool_name] | none]

mask { [ipv4] | [ipv6] }

mirror { [disabled | enabled | none] }

nat64 [enabled | disabled]

persist [none | profile1,profile2,profile3 …]

profiles [none | default | profile1,profile2,profile3 …]

policies [none | default | policy1,policy2,policy3 …]

rate_class [name]

rate_limit [integer]

rate_limitr_mode [destination | object | object-destination | object-source | object-source-destination | source | source-destination]

rate_limit_dst [integer]

rate_limit_src [integer]

rules [none | [rule_one,rule_two …]]

related_rules [none | [rule_one,rule_two …]]

reject [yes | no]

source { [ipv4[/prefixlen]] | [ipv6[/prefixlen]] }

source_address_translation [none | snat:pool_name | lsn | automap]

source_port [change | preserve | preserve-strict]

state [enabled | disable]

traffic_classes [none | default | class_one,class_two …]

translate_address [enabled | disabled]

translate_port [enabled | disabled]

vlans [none | default | [enabled|disabled]:vlan1,vlan2,vlan3 …]

CLI Example:

```
salt '*' bigip.modify_virtual bigip admin admin my-virtual source_address_
↪translation=none
salt '*' bigip.modify_virtual bigip admin admin my-virtual rules=my-rule,my-other-
↪rule
```

`salt.modules.bigip.`**`replace_pool_members`**(*hostname, username, password, name, members*)

A function to connect to a bigip device and replace members of an existing pool with new members.

hostname The host/address of the bigip device

username The iControl REST username

password The iControl REST password

name The name of the pool to modify

members List of comma delimited pool members to replace existing members with. i.e. 10.1.1.1:80,10.1.1.2:80,10.1.1.3:80

CLI Example:

```
salt '*' bigip.replace_pool_members bigip admin admin my-pool 10.2.2.1:80,10.2.2.
↪2:80,10.2.2.3:80
```

`salt.modules.bigip.`**`start_transaction`**(*hostname, username, password, label*)

A function to connect to a bigip device and start a new transaction.

hostname The host/address of the bigip device

username The iControl REST username

password The iControl REST password

label The name / alias for this transaction. The actual transaction id will be stored within a grain called `bigip_f5_trans:<label>`

CLI Example:

```
salt '*' bigip.start_transaction bigip admin admin my_transaction
```

21.9.32 salt.modules.bluez_bluetooth

Support for Bluetooth (using BlueZ in Linux).

The following packages are required packages for this module:

bluez >= 5.7 bluez-libs >= 5.7 bluez-utils >= 5.7 pybluez >= 0.18

`salt.modules.bluez_bluetooth.`**`address_`**`()`

Get the many addresses of the Bluetooth adapter

CLI Example:

```
salt '*' bluetooth.address
```

`salt.modules.bluez_bluetooth.`**`block`**`(`*bdaddr*`)`

Block a specific bluetooth device by BD Address

CLI Example:

```
salt '*' bluetooth.block DE:AD:BE:EF:CA:FE
```

`salt.modules.bluez_bluetooth.`**`discoverable`**`(`*dev*`)`

Enable this bluetooth device to be discoverable.

CLI Example:

```
salt '*' bluetooth.discoverable hci0
```

`salt.modules.bluez_bluetooth.`**`noscan`**`(`*dev*`)`

Turn off scanning modes on this device.

CLI Example:

```
salt '*' bluetooth.noscan hci0
```

`salt.modules.bluez_bluetooth.`**`pair`**`(`*address, key*`)`

Pair the bluetooth adapter with a device

CLI Example:

```
salt '*' bluetooth.pair DE:AD:BE:EF:CA:FE 1234
```

Where DE:AD:BE:EF:CA:FE is the address of the device to pair with, and 1234 is the passphrase.

TODO: This function is currently broken, as the bluez-simple-agent program no longer ships with BlueZ >= 5.0. It needs to be refactored.

`salt.modules.bluez_bluetooth.`**`power`**`(`*dev, mode*`)`

Power a bluetooth device on or off

CLI Examples:

```
salt '*' bluetooth.power hci0 on
salt '*' bluetooth.power hci0 off
```

`salt.modules.bluez_bluetooth.`**`scan`**`()`

Scan for bluetooth devices in the area

CLI Example:

```
salt '*' bluetooth.scan
```

`salt.modules.bluez_bluetooth.`**`start`**`()`

Start the bluetooth service.

CLI Example:

```
salt '*' bluetooth.start
```

salt.modules.bluez_bluetooth.**stop**()

Stop the bluetooth service.

CLI Example:

```
salt '*' bluetooth.stop
```

salt.modules.bluez_bluetooth.**unblock**(*bdaddr*)

Unblock a specific bluetooth device by BD Address

CLI Example:

```
salt '*' bluetooth.unblock DE:AD:BE:EF:CA:FE
```

salt.modules.bluez_bluetooth.**unpair**(*address*)

Unpair the bluetooth adapter from a device

CLI Example:

```
salt '*' bluetooth.unpair DE:AD:BE:EF:CA:FE
```

Where DE:AD:BE:EF:CA:FE is the address of the device to unpair.

TODO: This function is currently broken, as the bluez-simple-agent program no longer ships with BlueZ >= 5.0. It needs to be refactored.

salt.modules.bluez_bluetooth.**version**()

Return Bluez version from bluetoothd -v

CLI Example:

```
salt '*' bluetoothd.version
```

21.9.33 salt.modules.boto3_elasticache module

Execution module for Amazon Elasticache using boto3

New in version 2017.7.0.

configuration This module accepts explicit elasticache credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More Information available at:

```
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-
↪ec2.html
```

If IAM roles are not used you need to specify them either in a pillar or in the minion's config file:

```
elasticache.keyid: GKTADJGHEIQSXMKKRBJ08H
elasticache.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

A region may also be specified in the configuration:

```
elasticache.region: us-east-1
```

If a region is not specified, the default is us-east-1.

It's also possible to specify key, keyid and region via a profile, either as a passed in dict, or as a string to pull from pillars or minion config:

```
myprofile:
    keyid: GKTADJGHEIQSXMKKRBJ08H
    key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
    region: us-east-1
```

depends boto3

`salt.modules.boto3_elasticache.`**`add_tags_to_resource`**(*name*, *region=None*, *key=None*, *keyid=None*, *profile=None*, ***args*)

Add tags to an Elasticache resource.

Note that this function is essentially useless as it requires a full AWS ARN for the resource being operated on, but there is no provided API or programmatic way to find the ARN for a given object from its name or ID alone. It requires specific knowledge about the account number, AWS partition, and other magic details to generate.

If you happen to have those at hand though, feel free to utilize this function…

Example:

```
salt myminion boto3_elasticache.add_tags_to_resource                 name
↪'=arn:aws:elasticache:us-west-2:0123456789:snapshot:mySnapshot'                ⊠
↪Tags="[{'Key': 'TeamOwner', 'Value': 'infrastructure'}]"
```

`salt.modules.boto3_elasticache.`**`authorize_cache_security_group_ingress`**(*name*, *region=None*, *key=None*, *keyid=None*, *profile=None*, ***args*)

Authorize network ingress from an ec2 security group to a cache security group.

Example:

```
salt myminion boto3_elasticache.authorize_cache_security_group_ingress          ⊠
↪                              mycachesecgrp                                     ⊠
↪   EC2SecurityGroupName=someEC2sg                                         ⊠
↪EC2SecurityGroupOwnerId=SOMEOWNERID
```

`salt.modules.boto3_elasticache.`**`cache_cluster_exists`**(*name*, *conn=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Check to see if a cache cluster exists.

Example:

```
salt myminion boto3_elasticache.cache_cluster_exists myelasticache
```

`salt.modules.boto3_elasticache.`**`cache_security_group_exists`**(*name*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Check to see if an ElastiCache security group exists.

Example:

```
salt myminion boto3_elasticache.cache_security_group_exists mysecuritygroup
```

`salt.modules.boto3_elasticache.`**`cache_subnet_group_exists`**(*name*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Check to see if an ElastiCache subnet group exists.

Example:

```
salt myminion boto3_elasticache.cache_subnet_group_exists my-subnet-group
```

`salt.modules.boto3_elasticache.`**`copy_snapshot`**(*name*, *region=None*, *key=None*, *keyid=None*, *profile=None*, ***args*)

Make a copy of an existing snapshot.

Example:

```
salt myminion boto3_elasticache.copy_snapshot name=mySnapshot                     ⊠
↪                                   TargetSnapshotName=copyOfMySnapshot
```

`salt.modules.boto3_elasticache.`**`create_cache_cluster`**(*name*, *wait=600*, *security_groups=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*, ***args*)

Create a cache cluster.

Example:

```
salt myminion boto3_elasticache.create_cache_cluster name=myCacheCluster          ⊠
↪                                                     Engine=redis                ⊠
↪                                               CacheNodeType=cache.t2.micro      ⊠
↪                                                         NumCacheNodes=1         ⊠
↪                                                      SecurityGroupIds='[sg-
↪11223344]'                                                              ⊠
↪CacheSubnetGroupName=myCacheSubnetGroup
```

`salt.modules.boto3_elasticache.`**`create_cache_parameter_group`**(*name*, *region=None*, *key=None*, *keyid=None*, *profile=None*, ***args*)

Create a cache parameter group.

Example:

```
salt myminion boto3_elasticache.create_cache_parameter_group                 ⊠
↪name=myParamGroup                  CacheParameterGroupFamily=redis2.8            ⊠
↪      Description="My Parameter Group"
```

`salt.modules.boto3_elasticache.`**`create_cache_security_group`**(*name*, *region=None*, *key=None*, *keyid=None*, *profile=None*, ***args*)

Create a cache security group.

Example:

```
salt myminion boto3_elasticache.create_cache_security_group mycachesecgrp
↪Description='My Cache Security Group'
```

`salt.modules.boto3_elasticache.`**`create_cache_subnet_group`**(*name, subnets=None, region=None, key=None, keyid=None, profile=None, **args*)

Create an ElastiCache subnet group

Example:

```
salt myminion boto3_elasticache.create_cache_subnet_group name=my-subnet-group
↪                                             CacheSubnetGroupDescription=
↪"description"                                                 subnets=
↪'[myVPCSubnet1,myVPCSubnet2]'
```

`salt.modules.boto3_elasticache.`**`create_replication_group`**(*name, wait=600, security_groups=None, region=None, key=None, keyid=None, profile=None, **args*)

Create a replication group. Params are extensive and variable - see http://boto3.readthedocs.io/en/latest/reference/services/elasticache.html?#ElastiCache.Client.create_replication_group for in-depth usage documentation.

Example:

```
salt myminion boto3_elasticache.create_replication_group
↪                         name=myelasticache
↪                ReplicationGroupDescription=description
```

`salt.modules.boto3_elasticache.`**`delete_cache_cluster`**(*name, wait=600, region=None, key=None, keyid=None, profile=None, **args*)

Delete a cache cluster.

Example:

```
salt myminion boto3_elasticache.delete myelasticache
```

`salt.modules.boto3_elasticache.`**`delete_cache_parameter_group`**(*name, region=None, key=None, keyid=None, profile=None, **args*)

Delete a cache parameter group.

Example:

```
salt myminion boto3_elasticache.delete_cache_parameter_group myParamGroup
```

`salt.modules.boto3_elasticache.`**`delete_cache_security_group`**(*name, region=None, key=None, keyid=None, profile=None, **args*)

Delete a cache security group.

Example:

```
salt myminion boto3_elasticache.delete_cache_security_group myelasticachesg
```

`salt.modules.boto3_elasticache.`**`delete_cache_subnet_group`**`(`*name, region=None, key=None, keyid=None, profile=None, **args*`)`

Delete an ElastiCache subnet group.

Example:

```
salt myminion boto3_elasticache.delete_subnet_group my-subnet-group region=us-
↪east-1
```

`salt.modules.boto3_elasticache.`**`delete_replication_group`**`(`*name, wait=600, region=None, key=None, keyid=None, profile=None, **args*`)`

Delete an ElastiCache replication group, optionally taking a snapshot first.

Example:

```
salt myminion boto3_elasticache.delete_replication_group my-replication-group
```

`salt.modules.boto3_elasticache.`**`describe_cache_clusters`**`(`*name=None, conn=None, region=None, key=None, keyid=None, profile=None, **args*`)`

Return details about all (or just one) Elasticache cache clusters.

Example:

```
salt myminion boto3_elasticache.describe_cache_clusters
salt myminion boto3_elasticache.describe_cache_clusters myelasticache
```

`salt.modules.boto3_elasticache.`**`describe_cache_parameter_groups`**`(`*name=None, conn=None, region=None, key=None, keyid=None, profile=None*`)`

Return details about all (or just one) Elasticache cache clusters.

Example:

```
salt myminion boto3_elasticache.describe_cache_parameter_groups
salt myminion boto3_elasticache.describe_cache_parameter_groups myParameterGroup
```

`salt.modules.boto3_elasticache.`**`describe_cache_security_groups`**`(`*name=None, conn=None, region=None, key=None, keyid=None, profile=None*`)`

Return details about all (or just one) Elasticache cache clusters.

Example:

```
salt myminion boto3_elasticache.describe_cache_security_groups
salt myminion boto3_elasticache.describe_cache_security_groups mycachesecgrp
```

`salt.modules.boto3_elasticache.`**`describe_cache_subnet_groups`**(*name=None, conn=None, region=None, key=None, keyid=None, profile=None*)

Return details about all (or just one) Elasticache replication groups.

Example:

```
salt myminion boto3_elasticache.describe_cache_subnet_groups region=us-east-1
```

`salt.modules.boto3_elasticache.`**`describe_replication_groups`**(*name=None, conn=None, region=None, key=None, keyid=None, profile=None*)

Return details about all (or just one) Elasticache replication groups.

Example:

```
salt myminion boto3_elasticache.describe_replication_groups
salt myminion boto3_elasticache.describe_replication_groups myelasticache
```

`salt.modules.boto3_elasticache.`**`list_cache_subnet_groups`**(*region=None, key=None, keyid=None, profile=None*)

Return a list of all cache subnet group names

Example:

```
salt myminion boto3_elasticache.list_cache_subnet_groups region=us-east-1
```

`salt.modules.boto3_elasticache.`**`list_tags_for_resource`**(*name, region=None, key=None, keyid=None, profile=None, **args*)

List tags on an Elasticache resource.

Note that this function is essentially useless as it requires a full AWS ARN for the resource being operated on, but there is no provided API or programmatic way to find the ARN for a given object from its name or ID alone. It requires specific knowledge about the account number, AWS partition, and other magic details to generate.

If you happen to have those handy, feel free to utilize this however...

Example:

```
salt myminion boto3_elasticache.list_tags_for_resource                  name
↪'=arn:aws:elasticache:us-west-2:0123456789:snapshot:mySnapshot'
```

`salt.modules.boto3_elasticache.`**`modify_cache_cluster`**(*name, wait=600, security_groups=None, region=None, key=None, keyid=None, profile=None, **args*)

Update a cache cluster in place.

Notes: {ApplyImmediately: False} is pretty danged silly in the context of salt. You can pass it, but for fairly obvious reasons the results over multiple runs will be undefined and probably contrary to your desired state. Reducing the number of nodes requires an EXPLICIT CacheNodeIdsToRemove be passed, which until a reasonable heuristic for programmatically deciding which nodes to remove has been established, MUST be decided and populated intentionally before a state call, and removed again before the next. In practice this is not particularly useful and should probably be avoided.

Example:

```
salt myminion boto3_elasticache.create_cache_cluster name=myCacheCluster ⊠
↪                                                   ⊠
↪NotificationTopicStatus=inactive
```

`salt.modules.boto3_elasticache.`**`modify_cache_subnet_group`**(*name, subnets=None, region=None, key=None, keyid=None, profile=None, **args*)

Modify an ElastiCache subnet group

Example:

```
salt myminion boto3_elasticache.modify_cache_subnet_group                   ⊠
↪                      name=my-subnet-group                                 ⊠
↪         subnets='[myVPCSubnet3]'
```

`salt.modules.boto3_elasticache.`**`modify_replication_group`**(*name, wait=600, security_groups=None, region=None, key=None, keyid=None, profile=None, **args*)

Modify a replication group.

Example:

```
salt myminion boto3_elasticache.modify_replication_group                    ⊠
↪                         name=myelasticache                                ⊠
↪              ReplicationGroupDescription=newDescription
```

`salt.modules.boto3_elasticache.`**`remove_tags_from_resource`**(*name, region=None, key=None, keyid=None, profile=None, **args*)

Remove tags from an Elasticache resource.

Note that this function is essentially useless as it requires a full AWS ARN for the resource being operated on, but there is no provided API or programmatic way to find the ARN for a given object from its name or ID alone. It requires specific knowledge about the account number, AWS partition, and other magic details to generate.

If you happen to have those at hand though, feel free to utilize this function…

Example:

```
salt myminion boto3_elasticache.remove_tags_from_resource                 name
↪'=arn:aws:elasticache:us-west-2:0123456789:snapshot:mySnapshot'            ⊠
↪TagKeys="['TeamOwner']"
```

`salt.modules.boto3_elasticache.`**`replication_group_exists`**(*name, region=None, key=None, keyid=None, profile=None*)

Check to see if a replication group exists.

Example:

```
salt myminion boto3_elasticache.replication_group_exists myelasticache
```

salt.modules.boto3_elasticache.**revoke_cache_security_group_ingress**(*name*, *region=None*, *key=None*, *keyid=None*, *profile=None*, ***args*)

Revoke network ingress from an ec2 security group to a cache security group.

Example:

```
salt myminion boto3_elasticache.revoke_cache_security_group_ingress                  ↩
↪                           mycachesecgrp                                            ↩
↪EC2SecurityGroupName=someEC2sg                                    ↩
↪EC2SecurityGroupOwnerId=SOMEOWNERID
```

21.9.34 salt.modules.boto3_elasticsearch module

Connection module for Amazon Elasticsearch Service

New in version Natrium.

configuration This module accepts explicit IAM credentials but can also utilize IAM roles assigned to the instance trough Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More Information available at:

```
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-
↪ec2.html
```

If IAM roles are not used you need to specify them either in a pillar or in the minion's config file:

```
es.keyid: GKTADJGHEIQSXMKKRBJ08H
es.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

A region may also be specified in the configuration:

```
es.region: us-east-1
```

If a region is not specified, the default is us-east-1.

It's also possible to specify key, keyid and region via a profile, either as a passed in dict, or as a string to pull from pillars or minion config:

```
myprofile:
    keyid: GKTADJGHEIQSXMKKRBJ08H
    key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
    region: us-east-1
```

All methods return a dict with: 'result' key containing a boolean indicating success or failure, 'error' key containing the errormessage returned by boto on error, 'response' key containing the data of the response returned by boto on success.

codeauthor Herbert Buurman <herbert.buurman@ogd.nl>

depends boto3

`salt.modules.boto3_elasticsearch.`**`add_tags`**(*domain_name=None, arn=None, tags=None, region=None, key=None, keyid=None, profile=None*)

Attaches tags to an existing Elasticsearch domain. Tags are a set of case-sensitive key value pairs. An Elasticsearch domain may have up to 10 tags.

Parameters

- **domain_name** (*str*) – The name of the Elasticsearch domain you want to add tags to.
- **arn** (*str*) – The ARN of the Elasticsearch domain you want to add tags to. Specifying this overrides `domain_name`.
- **tags** (*dict*) – The dict of tags to add to the Elasticsearch domain.

Return type dict

Returns Dictionary with key 'result' and as value a boolean denoting success or failure. Upon failure, also contains a key 'error' with the error message as value.

New in version Natrium.

CLI Example:

```
salt myminion boto3_elasticsearch.add_tags domain_name=mydomain tags='{"foo": "bar
↪", "baz": "qux"}'
```

`salt.modules.boto3_elasticsearch.`**`cancel_elasticsearch_service_software_update`**(*domain_name, region=None, keyid=None, key=None, profile=None*)

Cancels a scheduled service software update for an Amazon ES domain. You can only perform this operation before the AutomatedUpdateDate and when the UpdateStatus is in the PENDING_UPDATE state.

Parameters **domain_name** (*str*) – The name of the domain that you want to stop the latest service software update on.

Return type dict

Returns Dictionary with key 'result' and as value a boolean denoting success or failure. Upon success, also contains a key 'reponse' with the current service software options. Upon failure, also contains a key 'error' with the error message as value.

New in version Natrium.

`salt.modules.boto3_elasticsearch.`**`check_upgrade_eligibility`**(*domain_name, elasticsearch_version, region=None, keyid=None, key=None, profile=None*)

Helper function to determine in one call if an Elasticsearch domain can be upgraded to the specified Elasticsearch version.

This assumes that the Elasticsearch domain is at rest at the moment this function is called. I.e. The domain is not in the process of :

- being created.
- being updated.

- another upgrade running, or a check thereof.
- being deleted.

Behind the scenes, this does 3 things:

- Check if `elasticsearch_version` is among the compatible elasticsearch versions.
- Perform a check if the Elasticsearch domain is eligible for the upgrade.
- Check the result of the check and return the result as a boolean.

Parameters

- **`name`** (*str*) – The Elasticsearch domain name to check.
- **`elasticsearch_version`** (*str*) – The Elasticsearch version to upgrade to.

Return type dict

Returns Dictionary with key 'result' and as value a boolean denoting success or failure. Upon success, also contains a key 'reponse' with boolean result of the check. Upon failure, also contains a key 'error' with the error message as value.

New in version Natrium.

CLI Example:

```
salt myminion boto3_elasticsearch.check_upgrade_eligibility mydomain '6.7'
```

`salt.modules.boto3_elasticsearch.`**`create_elasticsearch_domain`**(*domain_name, elasticsearch_version=None, elasticsearch_cluster_config=None, ebs_options=None, access_policies=None, snapshot_options=None, vpc_options=None, cognito_options=None, encryption_at_rest_options=None, node_to_node_encryption_options=None, advanced_options=None, log_publishing_options=None, blocking=False, region=None, key=None, keyid=None, profile=None*)

Given a valid config, create a domain.

Parameters

- **`domain_name`** (*str*) – The name of the Elasticsearch domain that you are creating. Domain names are unique across the domains owned by an account within an AWS region.

Domain names must start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen).

- **elasticsearch_version** (*str*) – String of format X.Y to specify version for the Elasticsearch domain eg. "1.5" or "2.3".
- **elasticsearch_cluster_config** (*dict*) – Dictionary specifying the configuration options for an Elasticsearch domain. Keys (case sensitive) in here are:
  - InstanceType (str): The instance type for an Elasticsearch cluster.
  - InstanceCount (int): The instance type for an Elasticsearch cluster.
  - DedicatedMasterEnabled (bool): Indicate whether a dedicated master node is enabled.
  - ZoneAwarenessEnabled (bool): Indicate whether zone awareness is enabled. If this is not enabled, the Elasticsearch domain will only be in one availability zone.
  - ZoneAwarenessConfig (dict): Specifies the zone awareness configuration for a domain when zone awareness is enabled. Keys (case sensitive) in here are:
    - AvailabilityZoneCount (int): An integer value to indicate the number of availability zones for a domain when zone awareness is enabled. This should be equal to number of subnets if VPC endpoints is enabled. Allowed values: 2, 3
  - DedicatedMasterType (str): The instance type for a dedicated master node.
  - DedicatedMasterCount (int): Total number of dedicated master nodes, active and on standby, for the cluster.
- **ebs_options** (*dict*) – Dict specifying the options to enable or disable and specifying the type and size of EBS storage volumes. Keys (case sensitive) in here are:
  - EBSEnabled (bool): Specifies whether EBS-based storage is enabled.
  - VolumeType (str): Specifies the volume type for EBS-based storage.
  - VolumeSize (int): Integer to specify the size of an EBS volume.
  - Iops (int): Specifies the IOPD for a Provisioned IOPS EBS volume (SSD).
- **access_policies** (*str or dict*) – Dict or JSON string with the IAM access policy.
- **snapshot_options** (*dict*) – Dict specifying the snapshot options. Keys (case sensitive) in here are:
  - AutomatedSnapshotStartHour (int): Specifies the time, in UTC format, when the service takes a daily automated snapshot of the specified Elasticsearch domain. Default value is 0 hours.
- **vpc_options** (*dict*) – Dict with the options to specify the subnets and security groups for the VPC endpoint. Keys (case sensitive) in here are:
  - SubnetIds (list): The list of subnets for the VPC endpoint.
  - SecurityGroupIds (list): The list of security groups for the VPC endpoint.
- **cognito_options** (*dict*) – Dict with options to specify the cognito user and identity pools for Kibana authentication. Keys (case sensitive) in here are:
  - Enabled (bool): Specifies the option to enable Cognito for Kibana authentication.
  - UserPoolId (str): Specifies the Cognito user pool ID for Kibana authentication.
  - IdentityPoolId (str): Specifies the Cognito identity pool ID for Kibana authentication.

  - RoleArn (str): Specifies the role ARN that provides Elasticsearch permissions for accessing Cognito resources.
- **encryption_at_rest_options** (*dict*) – Dict specifying the encryption at rest options. Keys (case sensitive) in here are:
  - Enabled (bool): Specifies the option to enable Encryption At Rest.
  - KmsKeyId (str): Specifies the KMS Key ID for Encryption At Rest options.
- **node_to_node_encryption_options** (*dict*) – Dict specifying the node to node encryption options. Keys (case sensitive) in here are:
  - Enabled (bool): Specify True to enable node-to-node encryption.
- **advanced_options** (*dict*) – Dict with option to allow references to indices in an HTTP request body. Must be False when configuring access to individual sub-resources. By default, the value is True. See http://docs.aws.amazon.com/elasticsearch-service/latest/developerguide /es-createupdatedomains.html#es-createdomain-configure-advanced-options for more information.
- **log_publishing_options** (*dict*) – Dict with options for various type of logs. The keys denote the type of log file and can be one of the following:
  - INDEX_SLOW_LOGS
  - SEARCH_SLOW_LOGS
  - ES_APPLICATION_LOGS

  The value assigned to each key is a dict with the following case sensitive keys:
  - CloudWatchLogsLogGroupArn (str): The ARN of the Cloudwatch log group to which the log needs to be published.
  - Enabled (bool): Specifies whether given log publishing option is enabled or not.
- **blocking** (*bool*) – Whether or not to wait (block) until the Elasticsearch domain has been created.

Note: Not all instance types allow enabling encryption at rest. See https://docs.aws.amazon.com /elasticsearch-service/latest/developerguide/aes-supported-instance-types.html

Return type dict

Returns Dictionary with key 'result' and as value a boolean denoting success or failure. Upon success, also contains a key 'reponse' with the domain status configuration. Upon failure, also contains a key 'error' with the error message as value.

New in version Natrium.

CLI Example:

```
salt myminion boto3_elasticsearch.create_elasticsearch_domain mydomain \
elasticsearch_cluster_config='{ \
  "InstanceType": "t2.micro.elasticsearch", \
  "InstanceCount": 1, \
  "DedicatedMasterEnabled": False, \
  "ZoneAwarenessEnabled": False}' \
ebs_options='{ \
  "EBSEnabled": True, \
  "VolumeType": "gp2", \
  "VolumeSize": 10, \
  "Iops": 0}' \
```

(continues on next page)

(continued from previous page)

```
access_policies='{ \
  "Version": "2012-10-17", \
  "Statement": [ \
    {"Effect": "Allow", \
     "Principal": {"AWS": "*"}, \
     "Action": "es:*", \
     "Resource": "arn:aws:es:us-east-1:111111111111:domain/mydomain/*", \
     "Condition": {"IpAddress": {"aws:SourceIp": ["127.0.0.1"]}}}]}' \
snapshot_options='{"AutomatedSnapshotStartHour": 0}' \
advanced_options='{"rest.action.multi.allow_explicit_index": "true"}'
```

`salt.modules.boto3_elasticsearch.`**`delete_elasticsearch_domain`**(*domain_name*, *blocking=False*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Permanently deletes the specified Elasticsearch domain and all of its data. Once a domain is deleted, it cannot be recovered.

Parameters

- **domain_name** (*str*) – The name of the domain to delete.
- **blocking** (*bool*) – Whether or not to wait (block) until the Elasticsearch domain has been deleted.

Return type dict

Returns Dictionary with key 'result' and as value a boolean denoting success or failure. Upon failure, also contains a key 'error' with the error message as value.

New in version Natrium.

`salt.modules.boto3_elasticsearch.`**`delete_elasticsearch_service_role`**(*region=None*, *keyid=None*, *key=None*, *profile=None*)

Deletes the service-linked role that Elasticsearch Service uses to manage and maintain VPC domains. Role deletion will fail if any existing VPC domains use the role. You must delete any such Elasticsearch domains before deleting the role.

Return type dict

Returns Dictionary with key 'result' and as value a boolean denoting success or failure. Upon failure, also contains a key 'error' with the error message as value.

New in version Natrium.

`salt.modules.boto3_elasticsearch.`**`describe_elasticsearch_domain`**(*domain_name*, *region=None*, *keyid=None*, *key=None*, *profile=None*)

Given a domain name gets its status description.

Parameters **domain_name** (*str*) – The name of the domain to get the status of.

Return type dict

Returns Dictionary ith key 'result' and as value a boolean denoting success or failure. Upon success, also contains a key 'reponse' with the domain status information. Upon failure, also contains a key 'error' with the error message as value.

New in version Natrium.

`salt.modules.boto3_elasticsearch.`**`describe_elasticsearch_domain_config`**`(`*domain_name, region=None, keyid=None, key=None, profile=None*`)`

Provides cluster configuration information about the specified Elasticsearch domain, such as the state, creation date, update version, and update date for cluster options.

Parameters **`domain_name`** (*`str`*) – The name of the domain to describe.

Return type dict

Returns Dictionary with key 'result' and as value a boolean denoting success or failure. Upon success, also contains a key 'reponse' with the current configuration information. Upon failure, also contains a key 'error' with the error message as value.

New in version Natrium.

`salt.modules.boto3_elasticsearch.`**`describe_elasticsearch_domains`**`(`*domain_names, region=None, keyid=None, key=None, profile=None*`)`

Returns domain configuration information about the specified Elasticsearch domains, including the domain ID, domain endpoint, and domain ARN.

Parameters **`domain_names`** (*`list`*) – List of domain names to get information for.

Return type dict

Returns Dictionary with key 'result' and as value a boolean denoting success or failure. Upon success, also contains a key 'reponse' with the list of domain status information. Upon failure, also contains a key 'error' with the error message as value.

New in version Natrium.

CLI Example:

```
salt myminion boto3_elasticsearch.describe_elasticsearch_domains '["domain_a",
↪"domain_b"]'
```

`salt.modules.boto3_elasticsearch.`**`describe_elasticsearch_instance_type_limits`**`(`*instance_type, elasticsearch_version, domain_name=None, region=None, keyid=None, key=None, profile=None*`)`

Describe Elasticsearch Limits for a given InstanceType and ElasticsearchVersion. When modifying existing Domain, specify the “ DomainName “ to know what Limits are supported for modifying.

Parameters

- **`instance_type`** (*str*) – The instance type for an Elasticsearch cluster for which Elasticsearch `Limits` are needed.
- **`elasticsearch_version`** (*str*) – Version of Elasticsearch for which `Limits` are needed.
- **`domain_name`** (*str*) – Represents the name of the Domain that we are trying to modify. This should be present only if we are querying for Elasticsearch `Limits` for existing domain.

Return type dict

Returns Dictionary with key ’result’ and as value a boolean denoting success or failure. Upon success, also contains a key ’reponse’ with the limits information. Upon failure, also contains a key ’error’ with the error message as value.

New in version Natrium.

CLI Example:

```
salt myminion boto3_elasticsearch.describe_elasticsearch_instance_type_limits \
  instance_type=r3.8xlarge.elasticsearch \
  elasticsearch_version='6.2'
```

`salt.modules.boto3_elasticsearch.`**`describe_reserved_elasticsearch_instance_offerings`**(*reserved_ region=Non keyid=No key=None profile=None*

Lists available reserved Elasticsearch instance offerings.

Parameters **`reserved_elasticsearch_instance_offering_id`** (*str*) – The offering identifier filter value. Use this parameter to show only the available offering that matches the specified reservation identifier.

Return type dict

Returns Dictionary with key ’result’ and as value a boolean denoting success or failure. Upon success, also contains a key ’reponse’ with the list of offerings information. Upon failure, also contains a key ’error’ with the error message as value.

New in version Natrium.

`salt.modules.boto3_elasticsearch.`**`describe_reserved_elasticsearch_instances`**(*reserved_elasticsearch_ region=None, keyid=None, key=None, profile=None*)

Returns information about reserved Elasticsearch instances for this account.

Parameters **`reserved_elasticsearch_instance_id`** (*str*) – The reserved instance identifier filter value. Use this parameter to show only the reservation that matches the specified reserved Elasticsearch instance ID.

Return type dict

Returns Dictionary with key 'result' and as value a boolean denoting success or failure. Upon success, also contains a key 'reponse' with a list of information on reserved instances. Upon failure, also contains a key 'error' with the error message as value.

Note Version 1.9.174 of boto3 has a bug in that reserved_elasticsearch_instance_id is considered a required argument, even though the documentation says otherwise.

New in version Natrium.

`salt.modules.boto3_elasticsearch.`**`exists`**(*domain_name*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Given a domain name, check to see if the given domain exists.

Parameters **`domain_name`** (*`str`*) – The name of the domain to check.

Return type dict

Returns Dictionary with key 'result' and as value a boolean denoting success or failure. Upon failure, also contains a key 'error' with the error message as value.

New in version Natrium.

`salt.modules.boto3_elasticsearch.`**`get_compatible_elasticsearch_versions`**(*domain_name=None*, *region=None*, *keyid=None*, *key=None*, *profile=None*)

Returns a list of upgrade compatible Elastisearch versions. You can optionally pass a `domain_name` to get all upgrade compatible Elasticsearch versions for that specific domain.

Parameters **`domain_name`** (*`str`*) – The name of an Elasticsearch domain.

Return type dict

Returns Dictionary with key 'result' and as value a boolean denoting success or failure. Upon success, also contains a key 'reponse' with a list of compatible versions. Upon failure, also contains a key 'error' with the error message as value.

New in version Natrium.

`salt.modules.boto3_elasticsearch.`**`get_upgrade_history`**(*domain_name*, *region=None*, *keyid=None*, *key=None*, *profile=None*)

Retrieves the complete history of the last 10 upgrades that were performed on the domain.

Parameters **`domain_name`** (*`str`*) – The name of an Elasticsearch domain. Domain names are unique across the domains owned by an account within an AWS region. Domain names start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen).

Return type dict

Returns Dictionary with key 'result' and as value a boolean denoting success or failure. Upon success, also contains a key 'reponse' with a list of upgrade histories. Upon failure, also contains a key 'error' with the error message as value.

New in version Natrium.

`salt.modules.boto3_elasticsearch.`**`get_upgrade_status`**(*domain_name*, *region=None*, *keyid=None*, *key=None*, *profile=None*)

Retrieves the latest status of the last upgrade or upgrade eligibility check that was performed on the domain.

Parameters **`domain_name`** (*`str`*) – The name of an Elasticsearch domain. Domain names are unique across the domains owned by an account within an AWS region. Domain names start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen).

Return type dict

Returns Dictionary with key 'result' and as value a boolean denoting success or failure. Upon success, also contains a key 'reponse' with upgrade status information. Upon failure, also contains a key 'error' with the error message as value.

New in version Natrium.

`salt.modules.boto3_elasticsearch.`**`list_domain_names`**(*region=None*, *keyid=None*, *key=None*, *profile=None*)

Returns the name of all Elasticsearch domains owned by the current user's account.

Return type dict

Returns Dictionary with key 'result' and as value a boolean denoting success or failure. Upon success, also contains a key 'reponse' with a list of domain names. Upon failure, also contains a key 'error' with the error message as value.

New in version Natrium.

`salt.modules.boto3_elasticsearch.`**`list_elasticsearch_instance_types`**(*elasticsearch_version*, *domain_name=None*, *region=None*, *keyid=None*, *key=None*, *profile=None*)

List all Elasticsearch instance types that are supported for given ElasticsearchVersion.

Parameters

- **`elasticsearch_version`** (*`str`*) – Version of Elasticsearch for which list of supported elasticsearch instance types are needed.
- **`domain_name`** (*`str`*) – DomainName represents the name of the Domain that we are trying to modify. This should be present only if we are querying for list of available Elasticsearch instance types when modifying existing domain.

Return type dict

Returns Dictionary with key 'result' and as value a boolean denoting success or failure. Upon success, also contains a key 'reponse' with a list of Elasticsearch instance types. Upon failure, also contains a key 'error' with the error message as value.

New in version Natrium.

`salt.modules.boto3_elasticsearch.`**`list_elasticsearch_versions`**(*region=None*, *keyid=None*, *key=None*, *profile=None*)

List all supported Elasticsearch versions.

Return type dict

Returns Dictionary with key 'result' and as value a boolean denoting success or failure. Upon success, also contains a key 'reponse' with a list of Elasticsearch versions. Upon failure, also contains a key 'error' with the error message as value.

New in version Natrium.

`salt.modules.boto3_elasticsearch.`**`list_tags`**(*domain_name=None, arn=None, region=None, key=None, keyid=None, profile=None*)

Returns all tags for the given Elasticsearch domain.

Return type dict

Returns Dictionary with key 'result' and as value a boolean denoting success or failure. Upon success, also contains a key 'reponse' with a dict of tags. Upon failure, also contains a key 'error' with the error message as value.

New in version Natrium.

`salt.modules.boto3_elasticsearch.`**`purchase_reserved_elasticsearch_instance_offering`**(*reserved_el reservation_name, instance_cou region=None, keyid=Non key=None, profile=None*)

Allows you to purchase reserved Elasticsearch instances.

Parameters

- **`reserved_elasticsearch_instance_offering_id`** (*`str`*) – The ID of the reserved Elasticsearch instance offering to purchase.
- **`reservation_name`** (*`str`*) – A customer-specified identifier to track this reservation.
- **`instance_count`** (*`int`*) – The number of Elasticsearch instances to reserve.

Return type dict

Returns Dictionary with key 'result' and as value a boolean denoting success or failure. Upon success, also contains a key 'reponse' with purchase information. Upon failure, also contains a key 'error' with the error message as value.

New in version Natrium.

`salt.modules.boto3_elasticsearch.`**`remove_tags`**(*tag_keys, domain_name=None, arn=None, region=None, key=None, keyid=None, profile=None*)

Removes the specified set of tags from the specified Elasticsearch domain.

Parameters

- **`tag_keys`** (*`list`*) – List with tag keys you want to remove from the Elasticsearch domain.

- **domain_name** (*str*) – The name of the Elasticsearch domain you want to remove tags from.
- **arn** (*str*) – The ARN of the Elasticsearch domain you want to remove tags from. Specifying this overrides `domain_name`.

Return type dict

Returns Dictionary with key 'result' and as value a boolean denoting success or failure. Upon failure, also contains a key 'error' with the error message as value.

New in version Natrium.

CLI Example:

```
salt myminion boto3_elasticsearch.remove_tags '["foo", "bar"]' domain_name=my_
↪domain
```

`salt.modules.boto3_elasticsearch.`**`start_elasticsearch_service_software_update`**(*domain_name, region=None, keyid=None, key=None, profile=None*)

Schedules a service software update for an Amazon ES domain.

Parameters **domain_name** (*str*) – The name of the domain that you want to update to the latest service software.

Return type dict

Returns Dictionary with key 'result' and as value a boolean denoting success or failure. Upon success, also contains a key 'reponse' with service software information. Upon failure, also contains a key 'error' with the error message as value.

New in version Natrium.

`salt.modules.boto3_elasticsearch.`**`update_elasticsearch_domain_config`**(*domain_name, elasticsearch_cluster_config=None, ebs_options=None, vpc_options=None, access_policies=None, snapshot_options=None, cognito_options=None, advanced_options=None, log_publishing_options=None, blocking=False, region=None, key=None, keyid=None, profile=None*)

Modifies the cluster configuration of the specified Elasticsearch domain, for example setting the instance type and the number of instances.

Parameters

- **`domain_name`** (*`str`*) – The name of the Elasticsearch domain that you are creating. Domain names are unique across the domains owned by an account within an AWS region. Domain names must start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen).
- **`elasticsearch_cluster_config`** (*`dict`*) – Dictionary specifying the configuration options for an Elasticsearch domain. Keys (case sensitive) in here are:
  - InstanceType (str): The instance type for an Elasticsearch cluster.
  - InstanceCount (int): The instance type for an Elasticsearch cluster.
  - DedicatedMasterEnabled (bool): Indicate whether a dedicated master node is enabled.
  - ZoneAwarenessEnabled (bool): Indicate whether zone awareness is enabled.
  - ZoneAwarenessConfig (dict): Specifies the zone awareness configuration for a domain when zone awareness is enabled. Keys (case sensitive) in here are:
    - AvailabilityZoneCount (int): An integer value to indicate the number of availability zones for a domain when zone awareness is enabled. This should be equal to number of subnets if VPC endpoints is enabled.
  - DedicatedMasterType (str): The instance type for a dedicated master node.
  - DedicatedMasterCount (int): Total number of dedicated master nodes, active and on standby, for the cluster.
- **`ebs_options`** (*`dict`*) – Dict specifying the options to enable or disable and specifying the type and size of EBS storage volumes. Keys (case sensitive) in here are:
  - EBSEnabled (bool): Specifies whether EBS-based storage is enabled.
  - VolumeType (str): Specifies the volume type for EBS-based storage.
  - VolumeSize (int): Integer to specify the size of an EBS volume.
  - Iops (int): Specifies the IOPD for a Provisioned IOPS EBS volume (SSD).
- **`snapshot_options`** (*`dict`*) – Dict specifying the snapshot options. Keys (case sensitive) in here are:
  - AutomatedSnapshotStartHour (int): Specifies the time, in UTC format, when the service takes a daily automated snapshot of the specified Elasticsearch domain. Default value is 0 hours.
- **`vpc_options`** (*`dict`*) – Dict with the options to specify the subnets and security groups for the VPC endpoint. Keys (case sensitive) in here are:
  - SubnetIds (list): The list of subnets for the VPC endpoint.
  - SecurityGroupIds (list): The list of security groups for the VPC endpoint.
- **`cognito_options`** (*`dict`*) – Dict with options to specify the cognito user and identity pools for Kibana authentication. Keys (case sensitive) in here are:
  - Enabled (bool): Specifies the option to enable Cognito for Kibana authentication.
  - UserPoolId (str): Specifies the Cognito user pool ID for Kibana authentication.
  - IdentityPoolId (str): Specifies the Cognito identity pool ID for Kibana authentication.

  - RoleArn (str): Specifies the role ARN that provides Elasticsearch permissions for accessing Cognito resources.

- **advanced_options** (*dict*) – Dict with option to allow references to indices in an HTTP request body. Must be False when configuring access to individual sub-resources. By default, the value is True. See http://docs.aws.amazon.com/elasticsearch-service/latest/developerguide /es-createupdatedomains.html#es-createdomain-configure-advanced-options for more information.
- **access_policies** (*str/dict*) – Dict or JSON string with the IAM access policy.
- **log_publishing_options** (*dict*) – Dict with options for various type of logs. The keys denote the type of log file and can be one of the following:

  INDEX_SLOW_LOGS, SEARCH_SLOW_LOGS, ES_APPLICATION_LOGS.

  The value assigned to each key is a dict with the following case sensitive keys:

  - CloudWatchLogsLogGroupArn (str): The ARN of the Cloudwatch log group to which the log needs to be published.
  - Enabled (bool): Specifies whether given log publishing option is enabled or not.
- **blocking** (*bool*) – Whether or not to wait (block) until the Elasticsearch domain has been updated.

Return type dict

Returns Dictionary with key 'result' and as value a boolean denoting success or failure. Upon success, also contains a key 'reponse' with the domain configuration. Upon failure, also contains a key 'error' with the error message as value.

New in version Natrium.

CLI Example:

```
salt myminion boto3_elasticsearch.update_elasticsearch_domain_config mydomain \
  elasticsearch_cluster_config='{\
    "InstanceType": "t2.micro.elasticsearch", \
    "InstanceCount": 1, \
    "DedicatedMasterEnabled": false,
    "ZoneAwarenessEnabled": false}' \
  ebs_options='{\
    "EBSEnabled": true, \
    "VolumeType": "gp2", \
    "VolumeSize": 10, \
    "Iops": 0}' \
  access_policies='{"Version": "2012-10-17", "Statement": [{\
    "Effect": "Allow", "Principal": {"AWS": "*"}, "Action": "es:*", \
    "Resource": "arn:aws:es:us-east-1:111111111111:domain/mydomain/*", \
    "Condition": {"IpAddress": {"aws:SourceIp": ["127.0.0.1"]}}}]}' \
  snapshot_options='{"AutomatedSnapshotStartHour": 0}' \
  advanced_options='{"rest.action.multi.allow_explicit_index": "true"}'
```

salt.modules.boto3_elasticsearch.**upgrade_elasticsearch_domain**(*domain_name, target_version, perform_check_only=None, blocking=False, region=None, keyid=None, key=None, profile=None*)

Allows you to either upgrade your domain or perform an Upgrade eligibility check to a compatible Elasticsearch version.

Parameters

- **domain_name** (*str*) – The name of an Elasticsearch domain. Domain names are unique across the domains owned by an account within an AWS region. Domain names start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen).
- **target_version** (*str*) – The version of Elasticsearch that you intend to upgrade the domain to.
- **perform_check_only** (*bool*) – This flag, when set to True, indicates that an Upgrade Eligibility Check needs to be performed. This will not actually perform the Upgrade.
- **blocking** (*bool*) – Whether or not to wait (block) until the Elasticsearch domain has been upgraded.

Return type dict

Returns Dictionary with key 'result' and as value a boolean denoting success or failure. Upon success, also contains a key 'reponse' with the domain configuration. Upon failure, also contains a key 'error' with the error message as value.

New in version Natrium.

CLI Example:

```
salt myminion boto3_elasticsearch.upgrade_elasticsearch_domain mydomain \
target_version='6.7' \
perform_check_only=True
```

salt.modules.boto3_elasticsearch.**wait_for_upgrade**(*domain_name, region=None, keyid=None, key=None, profile=None*)

Block until an upgrade-in-progress for domain `name` is finished.

Parameters **name** (*str*) – The name of the domain to wait for.

Rtype dict

Returns Dictionary with key 'result' and as value a boolean denoting success or failure. Upon failure, also contains a key 'error' with the error message as value.

New in version Natrium.

21.9.35 salt.modules.boto3_route53 module

Execution module for Amazon Route53 written against Boto 3

New in version 2017.7.0.

configuration This module accepts explicit route53 credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More Information available at:

```
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-
↪ec2.html
```

If IAM roles are not used you need to specify them either in a pillar or in the minion's config file:

```
route53.keyid: GKTADJGHEIQSXMKKRBJ08H
route53.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

A region may also be specified in the configuration:

```
route53.region: us-east-1
```

It's also possible to specify key, keyid and region via a profile, either as a passed in dict, or as a string to pull from pillars or minion config:

```
myprofile:
  keyid: GKTADJGHEIQSXMKKRBJ08H
  key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
  region: us-east-1
```

Note that Route53 essentially ignores all (valid) settings for 'region', since there is only one Endpoint (in us-east-1 if you care) and any (valid) region setting will just send you there. It is entirely safe to set it to None as well.

depends boto3

`salt.modules.boto3_route53.`**`associate_vpc_with_hosted_zone`**`(`*HostedZoneId=None, Name=None, VPCId=None, VPCName=None, VPCRegion=None, Comment=None, region=None, key=None, keyid=None, profile=None*`)`

Associates an Amazon VPC with a private hosted zone.

To perform the association, the VPC and the private hosted zone must already exist. You can't convert a public hosted zone into a private hosted zone. If you want to associate a VPC from one AWS account with a zone from a another, the AWS account owning the hosted zone must first submit a CreateVPCAssociationAuthorization (using create_vpc_association_authorization() or by other means, such as the AWS console). With that done, the account owning the VPC can then call associate_vpc_with_hosted_zone() to create the association.

Note that if both sides happen to be within the same account, associate_vpc_with_hosted_zone() is enough on its own, and there is no need for the CreateVPCAssociationAuthorization step.

Also note that looking up hosted zones by name (e.g. using the Name parameter) only works within a single account - if you're associating a VPC to a zone in a different account, as outlined above, you unfortunately MUST use the HostedZoneId parameter exclusively.

HostedZoneId The unique Zone Identifier for the Hosted Zone.

Name The domain name associated with the Hosted Zone(s).

VPCId When working with a private hosted zone, either the VPC ID or VPC Name to associate with is required. Exclusive with VPCName.

VPCName When working with a private hosted zone, either the VPC ID or VPC Name to associate with is required. Exclusive with VPCId.

VPCRegion When working with a private hosted zone, the region of the associated VPC is required. If not provided, an effort will be made to determine it from VPCId or VPCName, if possible. If this fails, you'll need to provide an explicit value for VPCRegion.

Comment Any comments you want to include about the change being made.

CLI Example:

```
salt myminion boto3_route53.associate_vpc_with_hosted_zone                    ☒
↪Name=example.org. VPCName=myVPC                    VPCRegion=us-east-1 Comment=
↪"Whoo-hoo!  I added another VPC."
```

`salt.modules.boto3_route53.`**`aws_encode`**`(`*x*`)`

An implementation of the encoding required to suport AWS's domain name rules defined here:

While AWS's documentation specifies individual ASCII characters which need to be encoded, we instead just try to force the string to one of escaped unicode or idna depending on whether there are non-ASCII characters present.

This means that we support things like ☒☒☒☒.☒☒☒ as a domain name string.

More information about IDNA encoding in python is found here:

`salt.modules.boto3_route53.`**`change_resource_record_sets`**`(`*HostedZoneId=None, Name=None, PrivateZone=None, ChangeBatch=None, region=None, key=None, keyid=None, profile=None*`)`

See the AWS Route53 API docs as well as the Boto3 documentation for all the details…

The syntax for a ChangeBatch parameter is as follows, but note that the permutations of allowed parameters and combinations thereof are quite varied, so perusal of the above linked docs is highly recommended for any non-trival configurations.

```
{
    "Comment": "string",
    "Changes": [
        {
            "Action": "CREATE"|"DELETE"|"UPSERT",
            "ResourceRecordSet": {
                "Name": "string",
                "Type": "SOA"|"A"|"TXT"|"NS"|"CNAME"|"MX"|"NAPTR"|"PTR"|"SRV"|"SPF
↪"|"AAAA",
                "SetIdentifier": "string",
                "Weight": 123,
                "Region": "us-east-1"|"us-east-2"|"us-west-1"|"us-west-2"|"ca-
↪central-1"|"eu-west-1"|"eu-west-2"|"eu-central-1"|"ap-southeast-1"|"ap-
↪southeast-2"|"ap-northeast-1"|"ap-northeast-2"|"sa-east-1"|"cn-north-1"|"ap-
↪south-1",
                "GeoLocation": {
                    "ContinentCode": "string",
                    "CountryCode": "string",
                    "SubdivisionCode": "string"
                },
                "Failover": "PRIMARY"|"SECONDARY",
                "TTL": 123,
```

(continues on next page)

(continued from previous page)

```
                "ResourceRecords": [
                    {
                        "Value": "string"
                    },
                ],
                "AliasTarget": {
                    "HostedZoneId": "string",
                    "DNSName": "string",
                    "EvaluateTargetHealth": True|False
                },
                "HealthCheckId": "string",
                "TrafficPolicyInstanceId": "string"
            }
        },
    ]
}
```

CLI Example:

```
foo='{
       "Name": "my-cname.example.org.",
       "TTL": 600,
       "Type": "CNAME",
       "ResourceRecords": [
         {
           "Value": "my-host.example.org"
         }
       ]
     }'
foo=`echo $foo`  # Remove newlines
salt myminion boto3_route53.change_resource_record_sets DomainName=example.org.
↪                keyid=A1234567890ABCDEF123 key=xblahblahblah
↪ChangeBatch="{'Changes': [{'Action': 'UPSERT', 'ResourceRecordSet': $foo}]}"
```

`salt.modules.boto3_route53.`**`create_hosted_zone`**(*Name, VPCId=None, VPCName=None, VPCRegion=None, CallerReference=None, Comment=”, PrivateZone=False, DelegationSetId=None, region=None, key=None, keyid=None, profile=None*)

Create a new Route53 Hosted Zone. Returns a Python data structure with information about the newly created Hosted Zone.

Name The name of the domain. This should be a fully-specified domain, and should terminate with a period. This is the name you have registered with your DNS registrar. It is also the name you will delegate from your registrar to the Amazon Route 53 delegation servers returned in response to this request.

VPCId When creating a private hosted zone, either the VPC ID or VPC Name to associate with is required. Exclusive with VPCName. Ignored if passed for a non-private zone.

VPCName When creating a private hosted zone, either the VPC ID or VPC Name to associate with is required. Exclusive with VPCId. Ignored if passed for a non-private zone.

VPCRegion When creating a private hosted zone, the region of the associated VPC is required. If not provided, an effort will be made to determine it from VPCId or VPCName, if possible. If this fails, you'll need to provide an explicit value for this option. Ignored if passed for a non-private zone.

CallerReference A unique string that identifies the request and that allows create_hosted_zone() calls to be retried without the risk of executing the operation twice. This is a required parameter when creating new Hosted Zones. Maximum length of 128.

Comment Any comments you want to include about the hosted zone.

PrivateZone Boolean - Set to True if creating a private hosted zone.

DelegationSetId If you want to associate a reusable delegation set with this hosted zone, the ID that Amazon Route 53 assigned to the reusable delegation set when you created it. Note that XXX TODO create_delegation_set() is not yet implemented, so you'd need to manually create any delegation sets before utilizing this.

region Region endpoint to connect to.

key AWS key to bind with.

keyid AWS keyid to bind with.

profile Dict, or pillar key pointing to a dict, containing AWS region/key/keyid.

CLI Example:

```
salt myminion boto3_route53.create_hosted_zone example.org.
```

`salt.modules.boto3_route53.`**`delete_hosted_zone`**(*Id*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Delete a Route53 hosted zone.

CLI Example:

```
salt myminion boto3_route53.delete_hosted_zone Z1234567890
```

`salt.modules.boto3_route53.`**`delete_hosted_zone_by_domain`**(*Name*, *PrivateZone=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Delete a Route53 hosted zone by domain name, and PrivateZone status if provided.

CLI Example:

```
salt myminion boto3_route53.delete_hosted_zone_by_domain example.org.
```

`salt.modules.boto3_route53.`**`disassociate_vpc_from_hosted_zone`**(*HostedZoneId=None*, *Name=None*, *VPCId=None*, *VPCName=None*, *VPCRegion=None*, *Comment=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Disassociates an Amazon VPC from a private hosted zone.

You can't disassociate the last VPC from a private hosted zone. You also can't convert a private hosted zone into a public hosted zone.

Note that looking up hosted zones by name (e.g. using the Name parameter) only works XXX FACTCHECK within a single AWS account - if you're disassociating a VPC in one account from a hosted zone in a different account you unfortunately MUST use the HostedZoneId parameter exclusively. XXX FIXME DOCU

HostedZoneId The unique Zone Identifier for the Hosted Zone.

Name The domain name associated with the Hosted Zone(s).

VPCId When working with a private hosted zone, either the VPC ID or VPC Name to associate with is required. Exclusive with VPCName.

VPCName When working with a private hosted zone, either the VPC ID or VPC Name to associate with is required. Exclusive with VPCId.

VPCRegion When working with a private hosted zone, the region of the associated VPC is required. If not provided, an effort will be made to determine it from VPCId or VPCName, if possible. If this fails, you'll need to provide an explicit value for VPCRegion.

Comment Any comments you want to include about the change being made.

CLI Example:

```
salt myminion boto3_route53.disassociate_vpc_from_hosted_zone
↪Name=example.org. VPCName=myVPC                          VPCRegion=us-east-1 Comment=
↪"Whoops!  Don't wanna talk to this-here zone no more."
```

`salt.modules.boto3_route53.`**`find_hosted_zone`**(*Id=None, Name=None, PrivateZone=None, region=None, key=None, keyid=None, profile=None*)

Find a hosted zone with the given characteristics.

Id The unique Zone Identifier for the Hosted Zone. Exclusive with Name.

Name The domain name associated with the Hosted Zone. Exclusive with Id. Note this has the potential to match more then one hosted zone (e.g. a public and a private if both exist) which will raise an error unless PrivateZone has also been passed in order split the different.

PrivateZone Boolean - Set to True if searching for a private hosted zone.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile Dict, or pillar key pointing to a dict, containing AWS region/key/keyid.

CLI Example:

```
salt myminion boto3_route53.find_hosted_zone Name=salt.org.
↪profile='{"region": "us-east-1", "keyid": "A12345678AB", "key": "xblahblahblah"}
↪'
```

`salt.modules.boto3_route53.`**`get_hosted_zone`**(*Id, region=None, key=None, keyid=None, profile=None*)

Return detailed info about the given zone.

Id The unique Zone Identifier for the Hosted Zone.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile Dict, or pillar key pointing to a dict, containing AWS region/key/keyid.

CLI Example:

```
salt myminion boto3_route53.get_hosted_zone Z1234567690                  profile='{
↪"region": "us-east-1", "keyid": "A12345678AB", "key": "xblahblahblah"}'
```

`salt.modules.boto3_route53.`**`get_hosted_zones_by_domain`**`(`*Name, region=None, key=None, keyid=None, profile=None*`)`

Find any zones with the given domain name and return detailed info about them. Note that this can return multiple Route53 zones, since a domain name can be used in both public and private zones.

Name The domain name associated with the Hosted Zone(s).

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile Dict, or pillar key pointing to a dict, containing AWS region/key/keyid.

CLI Example:

```
salt myminion boto3_route53.get_hosted_zones_by_domain salt.org.
↪profile='{"region": "us-east-1", "keyid": "A12345678AB", "key": "xblahblahblah"}
↪'
```

`salt.modules.boto3_route53.`**`get_resource_records`**`(`*HostedZoneId=None, Name=None, StartRecordName=None, StartRecordType=None, PrivateZone=None, region=None, key=None, keyid=None, profile=None*`)`

Get all resource records from a given zone matching the provided StartRecordName (if given) or all records in the zone (if not), optionally filtered by a specific StartRecordType. This will return any and all RRs matching, regardless of their special AWS flavors (weighted, geolocation, alias, etc.) so your code should be prepared for potentially large numbers of records back from this function - for example, if you've created a complex geolocation mapping with lots of entries all over the world providing the same server name to many different regional clients.

If you want EXACTLY ONE record to operate on, you'll need to implement any logic required to pick the specific RR you care about from those returned.

Note that if you pass in Name without providing a value for PrivateZone (either True or False), CommandExecutionError can be raised in the case of both public and private zones matching the domain. XXX FIXME DOCU

CLI example:

```
salt myminion boto3_route53.get_records test.example.org example.org A
```

`salt.modules.boto3_route53.`**`list_hosted_zones`**`(`*DelegationSetId=None, region=None, key=None, keyid=None, profile=None*`)`

Return detailed info about all zones in the bound account.

DelegationSetId If you're using reusable delegation sets and you want to list all of the hosted zones that are associated with a reusable delegation set, specify the ID of that delegation set.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile Dict, or pillar key pointing to a dict, containing AWS region/key/keyid.

CLI Example:

```
salt myminion boto3_route53.describe_hosted_zones                profile='{
↪"region": "us-east-1", "keyid": "A12345678AB", "key": "xblahblahblah"}'
```

`salt.modules.boto3_route53.`**`update_hosted_zone_comment`**`(`*Id=None, Name=None, Comment=None, PrivateZone=None, region=None, key=None, keyid=None, profile=None*`)`

Update the comment on an existing Route 53 hosted zone.

Id The unique Zone Identifier for the Hosted Zone.

Name The domain name associated with the Hosted Zone(s).

Comment Any comments you want to include about the hosted zone.

PrivateZone Boolean - Set to True if changing a private hosted zone.

CLI Example:

```
salt myminion boto3_route53.update_hosted_zone_comment Name=example.org.
↪         Comment="This is an example comment for an example zone"
```

21.9.36 salt.modules.boto_apigateway module

Connection module for Amazon APIGateway

New in version 2016.11.0.

depends

- boto >= 2.8.0
- boto3 >= 1.2.1
- botocore >= 1.4.49

configuration This module accepts explicit Lambda credentials but can also utilize IAM roles assigned to the instance trough Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More Information available at:

```
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-
↪ec2.html
```

If IAM roles are not used you need to specify them either in a pillar or in the minion's config file:

```
apigateway.keyid: GKTADJGHEIQSXMKKRBJ08H
apigateway.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

A region may also be specified in the configuration:

```
apigateway.region: us-west-2
```

If a region is not specified, the default is us-east-1.

It's also possible to specify key, keyid and region via a profile, either as a passed in dict, or as a string to pull from pillars or minion config:

```
myprofile:
    keyid: GKTADJGHEIQSXMKKRBJ08H
    key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
    region: us-west-2
```

Changed in version 2015.8.0: All methods now return a dictionary. Create and delete methods return:

```
created: true
```

or

```
created: false
error:
  message: error message
```

Request methods (e.g., `describe_apigateway`) return:

```
apigateway:
  - {...}
  - {...}
```

or

```
error:
  message: error message
```

`salt.modules.boto_apigateway.`**`activate_api_deployment`**(*restApiId, stageName, deploymentId, region=None, key=None, keyid=None, profile=None*)

Activates previously deployed deployment for a given stage

CLI Example:

```
salt myminion boto_apigateway.activate_api_deployent restApiId stagename␣
↪deploymentId
```

`salt.modules.boto_apigateway.`**`api_exists`**(*name, description=None, region=None, key=None, keyid=None, profile=None*)

Check to see if the given Rest API Name and optionally description exists.

CLI Example:

```
salt myminion boto_apigateway.exists myapi_name
```

`salt.modules.boto_apigateway.`**`api_model_exists`**(*restApiId, modelName, region=None, key=None, keyid=None, profile=None*)

Check to see if the given modelName exists in the given restApiId

CLI Example:

```
salt myminion boto_apigateway.api_model_exists restApiId modelName
```

`salt.modules.boto_apigateway.`**`associate_api_key_stagekeys`**(*apiKey, stagekeyslist, region=None, key=None, keyid=None, profile=None*)

associate the given stagekeyslist to the given apiKey.

CLI Example:

```
salt myminion boto_apigateway.associate_stagekeys_api_key \
        api_key '["restapi id/stage name", ...]'
```

salt.modules.boto_apigateway.**attach_usage_plan_to_apis**(*plan_id, apis, region=None, key=None, keyid=None, profile=None*)

Attaches given usage plan to each of the apis provided in a list of apiId and stage values

New in version 2017.7.0.

apis a list of dictionaries, where each dictionary contains the following:

apiId a string, which is the id of the created API in AWS ApiGateway

stage a string, which is the stage that the created API is deployed to.

CLI Example:

```
salt myminion boto_apigateway.attach_usage_plan_to_apis plan_id='usage plan id'␣
↪apis='[{"apiId": "some id 1", "stage": "some stage 1"}]'
```

salt.modules.boto_apigateway.**create_api**(*name, description, cloneFrom=None, region=None, key=None, keyid=None, profile=None*)

Create a new REST API Service with the given name

Returns {created: True} if the rest api was created and returns {created: False} if the rest api was not created.

CLI Example:

```
salt myminion boto_apigateway.create_api myapi_name api_description
```

salt.modules.boto_apigateway.**create_api_deployment**(*restApiId, stageName, stageDescription=", description=", cacheClusterEnabled=False, cacheClusterSize='0.5', variables=None, region=None, key=None, keyid=None, profile=None*)

Creates a new API deployment.

CLI Example:

```
salt myminion boto_apigateway.create_api_deployent restApiId stagename␣
↪stageDescription='' \
description='' cacheClusterEnabled=True|False cacheClusterSize=0.5 variables='{
↪"name": "value"}'
```

salt.modules.boto_apigateway.**create_api_integration**(*restApiId, resourcePath, httpMethod, integrationType, integrationHttpMethod, uri, credentials, requestParameters=None, requestTemplates=None, region=None, key=None, keyid=None, profile=None*)

Creates an integration for a given method in a given API. If integrationType is MOCK, uri and credential parameters will be ignored.

uri is in the form of (substitute APIGATEWAY_REGION and LAMBDA_FUNC_ARN) ”arn:aws:apigateway:APIGATEWAY_REGION:lambda:path/2015-03-31/functions/LAMBDA_FUNC_ARN/invocations”

credentials is in the form of an iam role name or role arn.

CLI Example:

```
salt myminion boto_apigateway.create_api_integration restApiId resourcePath⊠
↪httpMethod \
                     integrationType integrationHttpMethod uri credentials ['{}' [
↪'{}']]
```

salt.modules.boto_apigateway.**create_api_integration_response**(*restApiId, resourcePath, httpMethod, statusCode, selectionPattern, responseParameters=None, responseTemplates=None, region=None, key=None, keyid=None, profile=None*)

Creates an integration response for a given method in a given API

CLI Example:

```
salt myminion boto_apigateway.create_api_integration_response restApiId⊠
↪resourcePath httpMethod \
                    statusCode selectionPattern ['{}' ['{}']]
```

salt.modules.boto_apigateway.**create_api_key**(*name, description, enabled=True, stageKeys=None, region=None, key=None, keyid=None, profile=None*)

Create an API key given name and description.

An optional enabled argument can be provided. If provided, the valid values are True|False. This argument defaults to True.

An optional stageKeys argument can be provided in the form of list of dictionary with ’restApiId’ and ’stageName’ as keys.

CLI Example:

```
salt myminion boto_apigateway.create_api_key name description

salt myminion boto_apigateway.create_api_key name description enabled=False

salt myminion boto_apigateway.create_api_key name description \
     stageKeys='[{"restApiId": "id", "stageName": "stagename"}]'
```

salt.modules.boto_apigateway.**create_api_method**(*restApiId, resourcePath, httpMethod, authorizationType, apiKeyRequired=False, requestParameters=None, requestModels=None, region=None, key=None, keyid=None, profile=None*)

Creates API method for a resource in the given API

CLI Example:

```
salt myminion boto_apigateway.create_api_method restApiId resourcePath,
↪httpMethod, authorizationType, \
    apiKeyRequired=False, requestParameters='{"name", "value"}', requestModels='{
↪"content-type", "value"}'
```

salt.modules.boto_apigateway.**create_api_method_response**(*restApiId, resourcePath, httpMethod, statusCode, responseParameters=None, responseModels=None, region=None, key=None, keyid=None, profile=None*)

Create API method response for a method on a given resource in the given API

CLI Example:

```
salt myminion boto_apigateway.create_api_method_response restApiId resourcePath
↪httpMethod \
       statusCode responseParameters='{"name", "True|False"}' responseModels='{
↪"content-type", "model"}'
```

salt.modules.boto_apigateway.**create_api_model**(*restApiId, modelName, modelDescription, schema, contentType='application/json', region=None, key=None, keyid=None, profile=None*)

Create a new model in a given API with a given schema, currently only contentType supported is 'application/json'

CLI Example:

```
salt myminion boto_apigateway.create_api_model restApiId modelName
↪modelDescription '<schema>' 'content-type'
```

salt.modules.boto_apigateway.**create_api_resources**(*restApiId, path, region=None, key=None, keyid=None, profile=None*)

Given rest api id, and an absolute resource path, create all the resources and return all resources in the resourcepath, returns False on failure.

CLI Example:

```
salt myminion boto_apigateway.create_api_resources myapi_id resource_path
```

salt.modules.boto_apigateway.**create_api_stage**(*restApiId, stageName, deploymentId, description='', cacheClusterEnabled=False, cacheClusterSize='0.5', variables=None, region=None, key=None, keyid=None, profile=None*)

Creates a new API stage for a given restApiId and deploymentId.

CLI Example:

```
salt myminion boto_apigateway.create_api_stage restApiId stagename deploymentId \
    description='' cacheClusterEnabled=True|False cacheClusterSize='0.5'
↪variables='{"name": "value"}'
```

salt.modules.boto_apigateway.**create_usage_plan**(*name, description=None, throttle=None, quota=None, region=None, key=None, keyid=None, profile=None*)

Creates a new usage plan with throttling and quotas optionally applied

New in version 2017.7.0.

name Name of the usage plan

throttle A dictionary consisting of the following keys:

- **rateLimit** requests per second at steady rate, float
- **burstLimit** maximum number of requests per second, integer

quota A dictionary consisting of the following keys:

- **limit** number of allowed requests per specified quota period [required if quota parameter is present]
- **offset** number of requests to be subtracted from limit at the beginning of the period [optional]
- **period** quota period, must be one of DAY, WEEK, or MONTH. [required if quota parameter is present

CLI Example:

```
salt myminion boto_apigateway.create_usage_plan name='usage plan name' throttle='{
↪"rateLimit": 10.0, "burstLimit": 10}'
```

`salt.modules.boto_apigateway.`**`delete_api`**(*name, description=None, region=None, key=None, keyid=None, profile=None*)

Delete all REST API Service with the given name and an optional API description

Returns {deleted: True, count: deleted_count} if apis were deleted, and returns {deleted: False} if error or not found.

CLI Example:

```
salt myminion boto_apigateway.delete_api myapi_name

salt myminion boto_apigateway.delete_api myapi_name description='api description'
```

`salt.modules.boto_apigateway.`**`delete_api_deployment`**(*restApiId, deploymentId, region=None, key=None, keyid=None, profile=None*)

Deletes API deployment for a given restApiId and deploymentID

CLI Example:

```
salt myminion boto_apigateway.delete_api_deployent restApiId deploymentId
```

`salt.modules.boto_apigateway.`**`delete_api_integration`**(*restApiId, resourcePath, httpMethod, region=None, key=None, keyid=None, profile=None*)

Deletes an integration for a given method in a given API

CLI Example:

```
salt myminion boto_apigateway.delete_api_integration restApiId resourcePath
↪httpMethod
```

salt.modules.boto_apigateway.**delete_api_integration_response**(*restApiId*, *resourcePath*, *httpMethod*, *statusCode*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Deletes an integration response for a given method in a given API

CLI Example:

```
salt myminion boto_apigateway.delete_api_integration_response restApiId
↪resourcePath httpMethod statusCode
```

salt.modules.boto_apigateway.**delete_api_key**(*apiKey*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Deletes a given apiKey

CLI Example:

```
salt myminion boto_apigateway.delete_api_key apikeystring
```

salt.modules.boto_apigateway.**delete_api_method**(*restApiId*, *resourcePath*, *httpMethod*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Delete API method for a resource in the given API

CLI Example:

```
salt myminion boto_apigateway.delete_api_method restApiId resourcePath httpMethod
```

salt.modules.boto_apigateway.**delete_api_method_response**(*restApiId*, *resourcePath*, *httpMethod*, *statusCode*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Delete API method response for a resource in the given API

CLI Example:

```
salt myminion boto_apigateway.delete_api_method_response restApiId resourcePath
↪httpMethod statusCode
```

salt.modules.boto_apigateway.**delete_api_model**(*restApiId*, *modelName*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Delete a model identified by name in a given API

CLI Example:

```
salt myminion boto_apigateway.delete_api_model restApiId modelName
```

salt.modules.boto_apigateway.**delete_api_resources**(*restApiId*, *path*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Given restApiId and an absolute resource path, delete the resources starting from the absolute resource path. If resourcepath is the root resource '/', the function will return False. Returns False on failure.

CLI Example:

```
salt myminion boto_apigateway.delete_api_resources myapi_id, resource_path
```

`salt.modules.boto_apigateway.`**`delete_api_stage`**(*restApiId*, *stageName*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Deletes stage identified by stageName from API identified by restApiId

CLI Example:

```
salt myminion boto_apigateway.delete_api_stage restApiId stageName
```

`salt.modules.boto_apigateway.`**`delete_usage_plan`**(*plan_id*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Deletes usage plan identified by plan_id

New in version 2017.7.0.

CLI Example:

```
salt myminion boto_apigateway.delete_usage_plan plan_id='usage plan id'
```

`salt.modules.boto_apigateway.`**`describe_api_deployment`**(*restApiId*, *deploymentId*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Get API deployment for a given restApiId and deploymentId.

CLI Example:

```
salt myminion boto_apigateway.describe_api_deployent restApiId deploymentId
```

`salt.modules.boto_apigateway.`**`describe_api_deployments`**(*restApiId*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Gets information about the defined API Deployments. Return list of api deployments.

CLI Example:

```
salt myminion boto_apigateway.describe_api_deployments restApiId
```

`salt.modules.boto_apigateway.`**`describe_api_integration`**(*restApiId*, *resourcePath*, *httpMethod*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Get an integration for a given method in a given API

CLI Example:

```
salt myminion boto_apigateway.describe_api_integration restApiId resourcePath␣
↪httpMethod
```

`salt.modules.boto_apigateway.`**`describe_api_integration_response`**(*restApiId*, *resourcePath*, *httpMethod*, *statusCode*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Get an integration response for a given method in a given API

CLI Example:

```
salt myminion boto_apigateway.describe_api_integration_response restApiId␣
↪resourcePath httpMethod statusCode
```

`salt.modules.boto_apigateway.`**`describe_api_key`**`(`*apiKey, region=None, key=None, keyid=None, profile=None*`)`

Gets info about the given api key

CLI Example:

```
salt myminion boto_apigateway.describe_api_key apigw_api_key
```

`salt.modules.boto_apigateway.`**`describe_api_keys`**`(`*region=None, key=None, keyid=None, profile=None*`)`

Gets information about the defined API Keys. Return list of apiKeys.

CLI Example:

```
salt myminion boto_apigateway.describe_api_keys
```

`salt.modules.boto_apigateway.`**`describe_api_method`**`(`*restApiId, resourcePath, httpMethod, region=None, key=None, keyid=None, profile=None*`)`

Get API method for a resource in the given API

CLI Example:

```
salt myminion boto_apigateway.describe_api_method restApiId resourcePath␣
↪httpMethod
```

`salt.modules.boto_apigateway.`**`describe_api_method_response`**`(`*restApiId, resourcePath, httpMethod, statusCode, region=None, key=None, keyid=None, profile=None*`)`

Get API method response for a resource in the given API

CLI Example:

```
salt myminion boto_apigateway.describe_api_method_response restApiId resourcePath␣
↪httpMethod statusCode
```

`salt.modules.boto_apigateway.`**`describe_api_model`**`(`*restApiId, modelName, flatten=True, region=None, key=None, keyid=None, profile=None*`)`

Get a model by name for a given API

CLI Example:

```
salt myminion boto_apigateway.describe_api_model restApiId modelName [True]
```

`salt.modules.boto_apigateway.`**`describe_api_models`**`(`*restApiId, region=None, key=None, keyid=None, profile=None*`)`

Get all models for a given API

CLI Example:

```
salt myminion boto_apigateway.describe_api_models restApiId
```

salt.modules.boto_apigateway.**describe_api_resource**(*restApiId, path, region=None, key=None, keyid=None, profile=None*)

Given rest api id, and an absolute resource path, returns the resource id for the given path.

CLI Example:

```
salt myminion boto_apigateway.describe_api_resource myapi_id resource_path
```

salt.modules.boto_apigateway.**describe_api_resource_method**(*restApiId, resourcePath, httpMethod, region=None, key=None, keyid=None, profile=None*)

Given rest api id, resource path, and http method (must be one of DELETE, GET, HEAD, OPTIONS, PATCH, POST, PUT), return the method for the api/resource path if defined. Return False if method is not defined.

CLI Example:

```
salt myminion boto_apigateway.describe_api_resource_method myapi_id resource_path
↪httpmethod
```

salt.modules.boto_apigateway.**describe_api_resources**(*restApiId, region=None, key=None, keyid=None, profile=None*)

Given rest api id, return all resources for this api.

CLI Example:

```
salt myminion boto_apigateway.describe_api_resources myapi_id
```

salt.modules.boto_apigateway.**describe_api_stage**(*restApiId, stageName, region=None, key=None, keyid=None, profile=None*)

Get API stage for a given apiID and stage name

CLI Example:

```
salt myminion boto_apigateway.describe_api_stage restApiId stageName
```

salt.modules.boto_apigateway.**describe_api_stages**(*restApiId, deploymentId, region=None, key=None, keyid=None, profile=None*)

Get all API stages for a given apiID and deploymentID

CLI Example:

```
salt myminion boto_apigateway.describe_api_stages restApiId deploymentId
```

salt.modules.boto_apigateway.**describe_apis**(*name=None, description=None, region=None, key=None, keyid=None, profile=None*)

Returns all rest apis in the defined region. If optional parameter name is included, returns all rest apis matching the name in the defined region.

CLI Example:

```
salt myminion boto_apigateway.describe_apis

salt myminion boto_apigateway.describe_apis name='api name'

salt myminion boto_apigateway.describe_apis name='api name' description='desc str'
```

salt.modules.boto_apigateway.**describe_usage_plans**(*name=None*, *plan_id=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Returns a list of existing usage plans, optionally filtered to match a given plan name

New in version 2017.7.0.

CLI Example:

```
salt myminion boto_apigateway.describe_usage_plans
salt myminion boto_apigateway.describe_usage_plans name='usage plan name'
salt myminion boto_apigateway.describe_usage_plans plan_id='usage plan id'
```

salt.modules.boto_apigateway.**detach_usage_plan_from_apis**(*plan_id*, *apis*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Detaches given usage plan from each of the apis provided in a list of apiId and stage value

New in version 2017.7.0.

apis a list of dictionaries, where each dictionary contains the following:

apiId a string, which is the id of the created API in AWS ApiGateway

stage a string, which is the stage that the created API is deployed to.

CLI Example:

```
salt myminion boto_apigateway.detach_usage_plan_to_apis plan_id='usage plan id'
↪apis='[{"apiId": "some id 1", "stage": "some stage 1"}]'
```

salt.modules.boto_apigateway.**disable_api_key**(*apiKey*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

disable the given apiKey.

CLI Example:

```
salt myminion boto_apigateway.enable_api_key api_key
```

salt.modules.boto_apigateway.**disassociate_api_key_stagekeys**(*apiKey*, *stagekeyslist*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

disassociate the given stagekeyslist to the given apiKey.

CLI Example:

```
salt myminion boto_apigateway.disassociate_stagekeys_api_key \
        api_key '["restapi id/stage name", ...]'
```

salt.modules.boto_apigateway.**enable_api_key**(*apiKey*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

enable the given apiKey.

CLI Example:

```
salt myminion boto_apigateway.enable_api_key api_key
```

`salt.modules.boto_apigateway.`**`flush_api_stage_cache`**(*restApiId, stageName, region=None, key=None, keyid=None, profile=None*)

Flushes cache for the stage identified by stageName from API identified by restApiId

CLI Example:

```
salt myminion boto_apigateway.flush_api_stage_cache restApiId stageName
```

`salt.modules.boto_apigateway.`**`overwrite_api_stage_variables`**(*restApiId, stageName, variables, region=None, key=None, keyid=None, profile=None*)

Overwrite the stage variables for the given restApiId and stage name with the given variables, variables must be in the form of a dictionary. Overwrite will always remove all the existing stage variables associated with the given restApiId and stage name, follow by the adding of all the variables specified in the variables dictionary

CLI Example:

```
salt myminion boto_apigateway.overwrite_api_stage_variables restApiId stageName
↪variables='{"name": "value"}'
```

`salt.modules.boto_apigateway.`**`update_api_key_description`**(*apiKey, description, region=None, key=None, keyid=None, profile=None*)

update the given apiKey with the given description.

CLI Example:

```
salt myminion boto_apigateway.update_api_key_description api_key description
```

`salt.modules.boto_apigateway.`**`update_api_model_schema`**(*restApiId, modelName, schema, region=None, key=None, keyid=None, profile=None*)

update the schema (in python dictionary format) for the given model in the given restApiId

CLI Example:

```
salt myminion boto_apigateway.update_api_model_schema restApiId modelName schema
```

`salt.modules.boto_apigateway.`**`update_usage_plan`**(*plan_id, throttle=None, quota=None, region=None, key=None, keyid=None, profile=None*)

Updates an existing usage plan with throttling and quotas

New in version 2017.7.0.

plan_id Id of the created usage plan

throttle A dictionary consisting of the following keys:

- **rateLimit** requests per second at steady rate, float
- **burstLimit** maximum number of requests per second, integer

quota A dictionary consisting of the following keys:

- **limit** number of allowed requests per specified quota period [required if quota parameter is present]
- **offset** number of requests to be subtracted from limit at the beginning of the period [optional]

period quota period, must be one of DAY, WEEK, or MONTH. [required if quota parameter is present

CLI Example:

```
salt myminion boto_apigateway.update_usage_plan plan_id='usage plan id' throttle='
↪{"rateLimit": 10.0, "burstLimit": 10}'
```

21.9.37 salt.modules.boto_asg

Connection module for Amazon Autoscale Groups

New in version 2014.7.0.

configuration This module accepts explicit autoscale credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More Information available at:

```
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-
↪ec2.html
```

If IAM roles are not used you need to specify them either in a pillar or in the minion's config file:

```
asg.keyid: GKTADJGHEIQSXMKKRBJ08H
asg.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

A region may also be specified in the configuration:

```
asg.region: us-east-1
```

If a region is not specified, the default is us-east-1.

It's also possible to specify key, keyid and region via a profile, either as a passed in dict, or as a string to pull from pillars or minion config:

```
myprofile:
    keyid: GKTADJGHEIQSXMKKRBJ08H
    key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
    region: us-east-1
```

depends boto

depends boto3

`salt.modules.boto_asg.`**`create`**(*name, launch_config_name, availability_zones, min_size, max_size, desired_capacity=None, load_balancers=None, default_cooldown=None, health_check_type=None, health_check_period=None, placement_group=None, vpc_zone_identifier=None, tags=None, termination_policies=None, suspended_processes=None, scaling_policies=None, scheduled_actions=None, region=None, notification_arn=None, notification_types=None, key=None, keyid=None, profile=None*)

Create an autoscale group.

CLI example:

```
salt myminion boto_asg.create myasg mylc '["us-east-1a", "us-east-1e"]' 1 10 load_
↪balancers='["myelb", "myelb2"]' tags='[{"key": "Name", value="myasg",
↪"propagate_at_launch": True}]'
```

`salt.modules.boto_asg.`**`create_launch_configuration`**(*name, image_id, key_name=None, vpc_id=None, vpc_name=None, security_groups=None, user_data=None, instance_type='m1.small', kernel_id=None, ramdisk_id=None, block_device_mappings=None, instance_monitoring=False, spot_price=None, instance_profile_name=None, ebs_optimized=False, associate_public_ip_address=None, volume_type=None, delete_on_termination=True, iops=None, use_block_device_types=False, region=None, key=None, keyid=None, profile=None*)

Create a launch configuration.

CLI example:

```
salt myminion boto_asg.create_launch_configuration mylc image_id=ami-0b9c9f62 key_
↪name='mykey' security_groups='["mygroup"]' instance_type='c3.2xlarge'
```

`salt.modules.boto_asg.`**`delete`**(*name, force=False, region=None, key=None, keyid=None, profile=None*)

Delete an autoscale group.

CLI example:

```
salt myminion boto_asg.delete myasg region=us-east-1
```

`salt.modules.boto_asg.`**`delete_launch_configuration`**(*name, region=None, key=None, keyid=None, profile=None*)

Delete a launch configuration.

CLI example:

```
salt myminion boto_asg.delete_launch_configuration mylc
```

`salt.modules.boto_asg.`**`describe_launch_configuration`**(*name, region=None, key=None, keyid=None, profile=None*)

Dump details of a given launch configuration.

CLI example:

```
salt myminion boto_asg.describe_launch_configuration mylc
```

`salt.modules.boto_asg.`**`enter_standby`**(*name, instance_ids, should_decrement_desired_capacity=False, region=None, key=None, keyid=None, profile=None*)

Switch desired instances to StandBy mode

New in version 2016.11.0.

CLI example:

```
salt-call boto_asg.enter_standby my_autoscale_group_name '["i-xxxxxx"]'
```

`salt.modules.boto_asg.`**`exists`**(*name, region=None, key=None, keyid=None, profile=None*)

Check to see if an autoscale group exists.

CLI example:

```
salt myminion boto_asg.exists myasg region=us-east-1
```

`salt.modules.boto_asg.`**`exit_standby`**(*name, instance_ids, should_decrement_desired_capacity=False, region=None, key=None, keyid=None, profile=None*)

Exit desired instances from StandBy mode

New in version 2016.11.0.

CLI example:

```
salt-call boto_asg.exit_standby my_autoscale_group_name '["i-xxxxxx"]'
```

`salt.modules.boto_asg.`**`get_all_groups`**(*region=None, key=None, keyid=None, profile=None*)

Return all AutoScale Groups visible in the account (as a list of boto.ec2.autoscale.group.AutoScalingGroup).

New in version 2016.11.0.

CLI example:

```
salt-call boto_asg.get_all_groups region=us-east-1 --output yaml
```

`salt.modules.boto_asg.`**`get_all_launch_configurations`**(*region=None, key=None, keyid=None, profile=None*)

Fetch and return all Launch Configuration with details.

CLI example:

```
salt myminion boto_asg.get_all_launch_configurations
```

`salt.modules.boto_asg.`**`get_cloud_init_mime`**(*cloud_init*)

Get a mime multipart encoded string from a cloud-init dict. Currently supports boothooks, scripts and cloud-config.

CLI Example:

```
salt myminion boto.get_cloud_init_mime <cloud init>
```

`salt.modules.boto_asg.`**`get_config`**(*name, region=None, key=None, keyid=None, profile=None*)

Get the configuration for an autoscale group.

CLI example:

```
salt myminion boto_asg.get_config myasg region=us-east-1
```

`salt.modules.boto_asg.`**`get_instances`**(*name, lifecycle_state='InService', health_status='Healthy', attribute='private_ip_address', attributes=None, region=None, key=None, keyid=None, profile=None*)

return attribute of all instances in the named autoscale group.

CLI example:

```
salt-call boto_asg.get_instances my_autoscale_group_name
```

`salt.modules.boto_asg.`**`get_scaling_policy_arn`**(*as_group*, *scaling_policy_name*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Return the arn for a scaling policy in a specific autoscale group or None if not found. Mainly used as a helper method for boto_cloudwatch_alarm, for linking alarms to scaling policies.

CLI Example:

```
salt '*' boto_asg.get_scaling_policy_arn mygroup mypolicy
```

`salt.modules.boto_asg.`**`launch_configuration_exists`**(*name*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Check for a launch configuration's existence.

CLI example:

```
salt myminion boto_asg.launch_configuration_exists mylc
```

`salt.modules.boto_asg.`**`list_groups`**(*region=None*, *key=None*, *keyid=None*, *profile=None*)

Return all AutoScale Groups visible in the account (as a list of names).

New in version 2016.11.0.

CLI example:

```
salt-call boto_asg.list_groups region=us-east-1
```

`salt.modules.boto_asg.`**`list_launch_configurations`**(*region=None*, *key=None*, *keyid=None*, *profile=None*)

List all Launch Configurations.

CLI example:

```
salt myminion boto_asg.list_launch_configurations
```

`salt.modules.boto_asg.`**`update`**(*name*, *launch_config_name*, *availability_zones*, *min_size*, *max_size*, *desired_capacity=None*, *load_balancers=None*, *default_cooldown=None*, *health_check_type=None*, *health_check_period=None*, *placement_group=None*, *vpc_zone_identifier=None*, *tags=None*, *termination_policies=None*, *suspended_processes=None*, *scaling_policies=None*, *scheduled_actions=None*, *notification_arn=None*, *notification_types=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Update an autoscale group.

CLI example:

```
salt myminion boto_asg.update myasg mylc '["us-east-1a", "us-east-1e"]' 1 10 load_
↪balancers='["myelb", "myelb2"]' tags='[{"key": "Name", value="myasg",
↪"propagate_at_launch": True}]'
```

21.9.38 salt.modules.boto_cfn

Connection module for Amazon Cloud Formation

New in version 2015.5.0.

configuration This module accepts explicit AWS credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More Information available at:

```
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-
↪ec2.html
```

If IAM roles are not used you need to specify them either in a pillar or in the minion's config file:

```
cfn.keyid: GKTADJGHEIQSXMKKRBJ08H
cfn.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

A region may also be specified in the configuration:

```
cfn.region: us-east-1
```

depends boto

`salt.modules.boto_cfn.`**`create`**(*name, template_body=None, template_url=None, parameters=None, notification_arns=None, disable_rollback=None, timeout_in_minutes=None, capabilities=None, tags=None, on_failure=None, stack_policy_body=None, stack_policy_url=None, region=None, key=None, keyid=None, profile=None*)

Create a CFN stack.

CLI Example:

```
salt myminion boto_cfn.create mystack template_url='https://s3.amazonaws.com/
↪bucket/template.cft'          region=us-east-1
```

`salt.modules.boto_cfn.`**`delete`**(*name, region=None, key=None, keyid=None, profile=None*)

Delete a CFN stack.

CLI Example:

```
salt myminion boto_cfn.delete mystack region=us-east-1
```

`salt.modules.boto_cfn.`**`describe`**(*name, region=None, key=None, keyid=None, profile=None*)

Describe a stack.

New in version 2015.8.0.

CLI Example:

```
salt myminion boto_cfn.describe mystack region=us-east-1
```

`salt.modules.boto_cfn.`**`exists`**(*name, region=None, key=None, keyid=None, profile=None*)

Check to see if a stack exists.

CLI Example:

```
salt myminion boto_cfn.exists mystack region=us-east-1
```

`salt.modules.boto_cfn.`**`get_template`**(*name, region=None, key=None, keyid=None, profile=None*)

Check to see if attributes are set on a CFN stack.

CLI Example:

```
salt myminion boto_cfn.get_template mystack
```

`salt.modules.boto_cfn.`**`update_stack`**`(`*name, template_body=None, template_url=None, parameters=None, notification_arns=None, disable_rollback=False, timeout_in_minutes=None, capabilities=None, tags=None, use_previous_template=None, stack_policy_during_update_body=None, stack_policy_during_update_url=None, stack_policy_body=None, stack_policy_url=None, region=None, key=None, keyid=None, profile=None*`)`

Update a CFN stack.

New in version 2015.8.0.

CLI Example:

```
salt myminion boto_cfn.update_stack mystack template_url='https://s3.amazonaws.
↪com/bucket/template.cft'         region=us-east-1
```

`salt.modules.boto_cfn.`**`validate_template`**`(`*template_body=None, template_url=None, region=None, key=None, keyid=None, profile=None*`)`

Validate cloudformation template

New in version 2015.8.0.

CLI Example:

```
salt myminion boto_cfn.validate_template mystack-template
```

21.9.39 salt.modules.boto_cloudfront

Connection module for Amazon CloudFront

New in version 2018.3.0.

depends boto3

configuration This module accepts explicit AWS credentials but can also utilize IAM roles assigned to the instance through Instance Profiles or it can read them from the ~/.aws/credentials file or from these environment variables: AWS_ACCESS_KEY_ID, AWS_SECRET_ACCESS_KEY. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More information available at:

```
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/
    iam-roles-for-amazon-ec2.html

http://boto3.readthedocs.io/en/latest/guide/
    configuration.html#guide-configuration
```

If IAM roles are not used you need to specify them either in a pillar or in the minion's config file:

```
cloudfront.keyid: GKTADJGHEIQSXMKKRBJ08H
cloudfront.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

A region may also be specified in the configuration:

```
cloudfront.region: us-east-1
```

If a region is not specified, the default is us-east-1.

It's also possible to specify key, keyid and region via a profile, either as a passed in dict, or as a string to pull from pillars or minion config:

```
myprofile:
    keyid: GKTADJGHEIQSXMKKRBJ08H
    key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
    region: us-east-1
```

`salt.modules.boto_cloudfront.`**`create_distribution`**(*name, config, tags=None, region=None, key=None, keyid=None, profile=None*)

Create a CloudFront distribution with the given name, config, and (optionally) tags.

name Name for the CloudFront distribution

config Configuration for the distribution

tags Tags to associate with the distribution

region Region to connect to

key Secret key to use

keyid Access key to use

profile A dict with region, key, and keyid, or a pillar key (string) that contains such a dict.

CLI Example:

```
salt myminion boto_cloudfront.create_distribution name=mydistribution⊠
↪profile=awsprofile           config='{"Comment":"partial configuration",
↪"Enabled":true}'
```

`salt.modules.boto_cloudfront.`**`export_distributions`**(*region=None, key=None, keyid=None, profile=None*)

Get details of all CloudFront distributions. Produces results that can be used to create an SLS file.

CLI Example:

```
salt-call boto_cloudfront.export_distributions --out=txt |            sed "s/
↪local: //" > cloudfront_distributions.sls
```

`salt.modules.boto_cloudfront.`**`get_distribution`**(*name, region=None, key=None, keyid=None, profile=None*)

Get information about a CloudFront distribution (configuration, tags) with a given name.

name Name of the CloudFront distribution

region Region to connect to

key Secret key to use

keyid Access key to use

profile A dict with region, key, and keyid, or a pillar key (string) that contains such a dict.

CLI Example:

```
salt myminion boto_cloudfront.get_distribution name=mydistribution⊠
↪profile=awsprofile
```

`salt.modules.boto_cloudfront.`**`update_distribution`**`(`*name, config, tags=None, region=None, key=None, keyid=None, profile=None*`)`

Update the config (and optionally tags) for the CloudFront distribution with the given name.

name Name of the CloudFront distribution

config Configuration for the distribution

tags Tags to associate with the distribution

region Region to connect to

key Secret key to use

keyid Access key to use

profile A dict with region, key, and keyid, or a pillar key (string) that contains such a dict.

CLI Example:

```
salt myminion boto_cloudfront.update_distribution name=mydistribution
↪profile=awsprofile             config='{"Comment":"partial configuration",
↪"Enabled":true}'
```

21.9.40 salt.modules.boto_cloudtrail module

Connection module for Amazon CloudTrail

New in version 2016.3.0.

depends

- boto
- boto3

The dependencies listed above can be installed via package or pip.

configuration This module accepts explicit Lambda credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More Information available at:

```
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-
↪ec2.html
```

If IAM roles are not used you need to specify them either in a pillar or in the minion's config file:

```
cloudtrail.keyid: GKTADJGHEIQSXMKKRBJ08H
cloudtrail.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

A region may also be specified in the configuration:

```
cloudtrail.region: us-east-1
```

If a region is not specified, the default is us-east-1.

It's also possible to specify key, keyid and region via a profile, either as a passed in dict, or as a string to pull from pillars or minion config:

```
myprofile:
    keyid: GKTADJGHEIQSXMKKRBJ08H
    key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
    region: us-east-1
```

`salt.modules.boto_cloudtrail.`**`add_tags`**`(`*Name, region=None, key=None, keyid=None, profile=None, **kwargs*`)`

Add tags to a trail

Returns {tagged: true} if the trail was tagged and returns {tagged: False} if the trail was not tagged.

CLI Example:

```
salt myminion boto_cloudtrail.add_tags my_trail tag_a=tag_value tag_b=tag_value
```

`salt.modules.boto_cloudtrail.`**`create`**`(`*Name, S3BucketName, S3KeyPrefix=None, SnsTopicName=None, IncludeGlobalServiceEvents=None, IsMultiRegionTrail=None, EnableLogFileValidation=None, CloudWatchLogsLogGroupArn=None, CloudWatchLogsRoleArn=None, KmsKeyId=None, region=None, key=None, keyid=None, profile=None*`)`

Given a valid config, create a trail.

Returns {created: true} if the trail was created and returns {created: False} if the trail was not created.

CLI Example:

```
salt myminion boto_cloudtrail.create my_trail my_bucket
```

`salt.modules.boto_cloudtrail.`**`delete`**`(`*Name, region=None, key=None, keyid=None, profile=None*`)`

Given a trail name, delete it.

Returns {deleted: true} if the trail was deleted and returns {deleted: false} if the trail was not deleted.

CLI Example:

```
salt myminion boto_cloudtrail.delete mytrail
```

`salt.modules.boto_cloudtrail.`**`describe`**`(`*Name, region=None, key=None, keyid=None, profile=None*`)`

Given a trail name describe its properties.

Returns a dictionary of interesting properties.

CLI Example:

```
salt myminion boto_cloudtrail.describe mytrail
```

`salt.modules.boto_cloudtrail.`**`exists`**`(`*Name, region=None, key=None, keyid=None, profile=None*`)`

Given a trail name, check to see if the given trail exists.

Returns True if the given trail exists and returns False if the given trail does not exist.

CLI Example:

```
salt myminion boto_cloudtrail.exists mytrail
```

`salt.modules.boto_cloudtrail.`**`list`**`(`*region=None, key=None, keyid=None, profile=None*`)`

List all trails

Returns list of trails

CLI Example:

```
policies:
  - {...}
  - {...}
```

`salt.modules.boto_cloudtrail.`**`list_tags`**(*Name*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

List tags of a trail

Returns

- {…}
- {…}

Return type tags

CLI Example:

```
salt myminion boto_cloudtrail.list_tags my_trail
```

`salt.modules.boto_cloudtrail.`**`remove_tags`**(*Name*, *region=None*, *key=None*, *keyid=None*, *profile=None*, ***kwargs*)

Remove tags from a trail

Returns {tagged: true} if the trail was tagged and returns {tagged: False} if the trail was not tagged.

CLI Example:

```
salt myminion boto_cloudtrail.remove_tags my_trail tag_a=tag_value tag_b=tag_value
```

`salt.modules.boto_cloudtrail.`**`start_logging`**(*Name*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Start logging for a trail

Returns {started: true} if the trail was started and returns {started: False} if the trail was not started.

CLI Example:

```
salt myminion boto_cloudtrail.start_logging my_trail
```

`salt.modules.boto_cloudtrail.`**`status`**(*Name*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Given a trail name describe its properties.

Returns a dictionary of interesting properties.

CLI Example:

```
salt myminion boto_cloudtrail.describe mytrail
```

`salt.modules.boto_cloudtrail.`**`stop_logging`**(*Name*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Stop logging for a trail

Returns {stopped: true} if the trail was stopped and returns {stopped: False} if the trail was not stopped.

CLI Example:

```
salt myminion boto_cloudtrail.stop_logging my_trail
```

`salt.modules.boto_cloudtrail.`**`update`**(*Name, S3BucketName, S3KeyPrefix=None, SnsTopicName=None, IncludeGlobalServiceEvents=None, IsMultiRegionTrail=None, EnableLogFileValidation=None, CloudWatchLogsLogGroupArn=None, CloudWatchLogsRoleArn=None, KmsKeyId=None, region=None, key=None, keyid=None, profile=None*)

Given a valid config, update a trail.

Returns {created: true} if the trail was created and returns {created: False} if the trail was not created.

CLI Example:

```
salt myminion boto_cloudtrail.update my_trail my_bucket
```

21.9.41 salt.modules.boto_cloudwatch

Connection module for Amazon CloudWatch

New in version 2014.7.0.

configuration This module accepts explicit credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More Information available at:

```
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-
↪ec2.html
```

If IAM roles are not used you need to specify them either in a pillar or in the minion's config file:

```
cloudwatch.keyid: GKTADJGHEIQSXMKKRBJ08H
cloudwatch.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

A region may also be specified in the configuration:

```
cloudwatch.region: us-east-1
```

If a region is not specified, the default is us-east-1.

It's also possible to specify key, keyid and region via a profile, either as a passed in dict, or as a string to pull from pillars or minion config:

```
myprofile:
    keyid: GKTADJGHEIQSXMKKRBJ08H
    key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
    region: us-east-1
```

depends boto

`salt.modules.boto_cloudwatch.`**`convert_to_arn`**(*arns, region=None, key=None, keyid=None, profile=None*)

Convert a list of strings into actual arns. Converts convenience names such as 'scaling_policy:...'

CLI Example:

```
salt '*' convert_to_arn 'scaling_policy:'
```

salt.modules.boto_cloudwatch.**create_or_update_alarm**(*connection=None, name=None, metric=None, namespace=None, statistic=None, comparison=None, threshold=None, period=None, evaluation_periods=None, unit=None, description='', dimensions=None, alarm_actions=None, insufficient_data_actions=None, ok_actions=None, region=None, key=None, keyid=None, profile=None*)

Create or update a cloudwatch alarm.

Params are the same as: https://boto.readthedocs.io/en/latest/ref/cloudwatch.html#boto.ec2.cloudwatch.alarm.MetricAlarm.

Dimensions must be a dict. If the value of Dimensions is a string, it will be json decoded to produce a dict. alarm_actions, insufficient_data_actions, and ok_actions must be lists of string. If the passed-in value is a string, it will be split on "," to produce a list. The strings themselves for alarm_actions, insufficient_data_actions, and ok_actions must be Amazon resource names (ARN's); however, this method also supports an arn lookup notation, as follows:

> arn:aws:.... ARN as per http://docs.aws.amazon.com/general/latest/gr/aws-arns-and-namespaces.html scaling_policy:<as_name>:<scaling_policy_name> The named autoscale group scaling policy, for the named group (e.g. scaling_policy:my-asg:ScaleDown)

This is convenient for setting up autoscaling as follows. First specify a boto_asg.present state for an ASG with scaling_policies, and then set up boto_cloudwatch_alarm.present states which have alarm_actions that reference the scaling_policy.

CLI example:

> salt myminion boto_cloudwatch.create_alarm name=myalarm ... region=us-east-1

salt.modules.boto_cloudwatch.**delete_alarm**(*name, region=None, key=None, keyid=None, profile=None*)

Delete a cloudwatch alarm

CLI example to delete a queue:

```
salt myminion boto_cloudwatch.delete_alarm myalarm region=us-east-1
```

salt.modules.boto_cloudwatch.**get_alarm**(*name, region=None, key=None, keyid=None, profile=None*)

Get alarm details. Also can be used to check to see if an alarm exists.

CLI example:

```
salt myminion boto_cloudwatch.get_alarm myalarm region=us-east-1
```

salt.modules.boto_cloudwatch.**get_all_alarms**(*region=None, prefix=None, key=None, keyid=None, profile=None*)

Get all alarm details. Produces results that can be used to create an sls file.

If prefix parameter is given, alarm names in the output will be prepended with the prefix; alarms that have the prefix will be skipped. This can be used to convert existing alarms to be managed by salt, as follows:

1. **Make a "backup" of all existing alarms** $ salt-call boto_cloudwatch.get_all_alarms –out=txt | sed "s/local: //" > legacy_alarms.sls

2. **Get all alarms with new prefixed names** $ salt-call boto_cloudwatch.get_all_alarms "prefix=**MANAGED BY SALT** " –out=txt | sed "s/local: //" > managed_alarms.sls
3. **Insert the managed alarms into cloudwatch** $ salt-call state.template managed_alarms.sls
4. Manually verify that the new alarms look right
5. Delete the original alarms $ sed s/present/absent/ legacy_alarms.sls > remove_legacy_alarms.sls $ salt-call state.template remove_legacy_alarms.sls
6. Get all alarms again, verify no changes $ salt-call boto_cloudwatch.get_all_alarms –out=txt | sed "s/local: //" > final_alarms.sls $ diff final_alarms.sls managed_alarms.sls

CLI example:

```
salt myminion boto_cloudwatch.get_all_alarms region=us-east-1 --out=txt
```

21.9.42 salt.modules.boto_cloudwatch_event module

Connection module for Amazon CloudWatch Events

New in version 2016.11.0.

configuration This module accepts explicit credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More Information available at:

```
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-
↪ec2.html
```

If IAM roles are not used you need to specify them either in a pillar or in the minion's config file:

```
cloudwatch_event.keyid: GKTADJGHEIQSXMKKRBJ08H
cloudwatch_event.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

A region may also be specified in the configuration:

```
cloudwatch_event.region: us-east-1
```

If a region is not specified, the default is us-east-1.

It's also possible to specify key, keyid and region via a profile, either as a passed in dict, or as a string to pull from pillars or minion config:

```
myprofile:
    keyid: GKTADJGHEIQSXMKKRBJ08H
    key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
    region: us-east-1
```

depends boto3

`salt.modules.boto_cloudwatch_event.`**`create_or_update`**(*Name, ScheduleExpression=None, EventPattern=None, Description=None, RoleArn=None, State=None, region=None, key=None, keyid=None, profile=None*)

Given a valid config, create an event rule.

Returns {created: true} if the rule was created and returns {created: False} if the rule was not created.

CLI Example:

```
salt myminion boto_cloudwatch_event.create_or_update my_rule
```

`salt.modules.boto_cloudwatch_event.`**`delete`**(*Name, region=None, key=None, keyid=None, profile=None*)

Given a rule name, delete it.

Returns {deleted: true} if the rule was deleted and returns {deleted: false} if the rule was not deleted.

CLI Example:

```
salt myminion boto_cloudwatch_event.delete myrule
```

`salt.modules.boto_cloudwatch_event.`**`describe`**(*Name, region=None, key=None, keyid=None, profile=None*)

Given a rule name describe its properties.

Returns a dictionary of interesting properties.

CLI Example:

```
salt myminion boto_cloudwatch_event.describe myrule
```

`salt.modules.boto_cloudwatch_event.`**`exists`**(*Name, region=None, key=None, keyid=None, profile=None*)

Given a rule name, check to see if the given rule exists.

Returns True if the given rule exists and returns False if the given rule does not exist.

CLI example:

```
salt myminion boto_cloudwatch_event.exists myevent region=us-east-1
```

`salt.modules.boto_cloudwatch_event.`**`list_rules`**(*region=None, key=None, keyid=None, profile=None*)

List, with details, all Cloudwatch Event rules visible in the current scope.

CLI example:

```
salt myminion boto_cloudwatch_event.list_rules region=us-east-1
```

`salt.modules.boto_cloudwatch_event.`**`list_targets`**(*Rule, region=None, key=None, keyid=None, profile=None*)

Given a rule name list the targets of that rule.

Returns a dictionary of interesting properties.

CLI Example:

```
salt myminion boto_cloudwatch_event.list_targets myrule
```

`salt.modules.boto_cloudwatch_event.`**`put_targets`**(*Rule, Targets, region=None, key=None, keyid=None, profile=None*)

Add the given targets to the given rule

Returns a dictionary describing any failures.

CLI Example:

```
salt myminion boto_cloudwatch_event.put_targets myrule [{'Id': 'target1', 'Arn':
↪'arn:***'}]
```

`salt.modules.boto_cloudwatch_event.`**`remove_targets`**(*Rule, Ids, region=None, key=None, keyid=None, profile=None*)

Given a rule name remove the named targets from the target list

Returns a dictionary describing any failures.

CLI Example:

```
salt myminion boto_cloudwatch_event.remove_targets myrule ['Target1']
```

21.9.43 salt.modules.boto_cognitoidentity module

Connection module for Amazon CognitoIdentity

New in version 2016.11.0.

configuration This module accepts explicit CognitoIdentity credentials but can also utilize IAM roles assigned to the instance trough Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More Information available at:

```
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-
↪ec2.html
```

If IAM roles are not used you need to specify them either in a pillar or in the minion's config file:

```
cognitoidentity.keyid: GKTADJGHEIQSXMKKRBJ08H
cognitoidentity.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

A region may also be specified in the configuration:

```
cognitoidentity.region: us-east-1
```

If a region is not specified, the default is us-east-1.

It's also possible to specify key, keyid and region via a profile, either as a passed in dict, or as a string to pull from pillars or minion config:

```
myprofile:
    keyid: GKTADJGHEIQSXMKKRBJ08H
    key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
    region: us-east-1
```

Changed in version 2015.8.0: All methods now return a dictionary. Create, delete, set, and update methods return:

```
created: true
```

or

```
created: false
error:
  message: error message
```

Request methods (e.g., *describe_identity_pools*) return:

```
identity_pools:
  - {...}
  - {...}
```

or

```
error:
  message: error message
```

depends boto3

`salt.modules.boto_cognitoidentity.`**`create_identity_pool`**(*IdentityPoolName, AllowUnauthenticatedIdentities=False, SupportedLoginProviders=None, DeveloperProviderName=None, OpenIdConnectProviderARNs=None, region=None, key=None, keyid=None, profile=None*)

Creates a new identity pool. All parameters except for IdentityPoolName is optional. SupportedLoginProviders should be a dictionary mapping provider names to provider app IDs. OpenIdConnectProviderARNs should be a list of OpenID Connect provider ARNs.

Returns the created identity pool if successful

CLI Example:

```
salt myminion boto_cognitoidentity.create_identity_pool my_id_pool_name
↪                    DeveloperProviderName=custom_developer_provider
```

`salt.modules.boto_cognitoidentity.`**`delete_identity_pools`**(*IdentityPoolName, IdentityPoolId=None, region=None, key=None, keyid=None, profile=None*)

Given an identity pool name, (optionally if an identity pool id is given, the given name will be ignored)

Deletes all identity pools matching the given name, or the specific identity pool with the given identity pool id.

CLI Example:

```
salt myminion boto_cognitoidentity.delete_identity_pools my_id_pool_name
salt myminion boto_cognitoidentity.delete_identity_pools '' IdentityPoolId=my_id_
↪pool_id
```

`salt.modules.boto_cognitoidentity.`**`describe_identity_pools`**(*IdentityPoolName, IdentityPoolId=None, region=None, key=None, keyid=None, profile=None*)

Given an identity pool name, (optionally if an identity pool id is given, the given name will be ignored)

Returns a list of matched identity pool name's pool properties

CLI Example:

```
salt myminion boto_cognitoidentity.describe_identity_pools my_id_pool_name
salt myminion boto_cognitoidentity.describe_identity_pools '' IdentityPoolId=my_
↪id_pool_id
```

salt.modules.boto_cognitoidentity.**get_identity_pool_roles**(*IdentityPoolName, IdentityPoolId=None, region=None, key=None, keyid=None, profile=None*)

Given an identity pool name, (optionally if an identity pool id if given, the given name will be ignored)

Returns a list of matched identity pool name's associated roles

CLI Example:

```
salt myminion boto_cognitoidentity.get_identity_pool_roles my_id_pool_name
salt myminion boto_cognitoidentity.get_identity_pool_roles '' IdentityPoolId=my_
↪id_pool_id
```

salt.modules.boto_cognitoidentity.**set_identity_pool_roles**(*IdentityPoolId, AuthenticatedRole=None, UnauthenticatedRole=None, region=None, key=None, keyid=None, profile=None*)

Given an identity pool id, set the given AuthenticatedRole and UnauthenticatedRole (the Role can be an iam arn, or a role name) If AuthenticatedRole or UnauthenticatedRole is not given, the authenticated and/or the unauthenticated role associated previously with the pool will be cleared.

Returns set True if successful, set False if unsuccessful with the associated errors.

CLI Example:

```
salt myminion boto_cognitoidentity.set_identity_pool_roles my_id_pool_roles  #
↪this clears the roles
salt myminion boto_cognitoidentity.set_identity_pool_roles my_id_pool_id        
↪   AuthenticatedRole=my_auth_role UnauthenticatedRole=my_unauth_role  # this
↪set both roles
salt myminion boto_cognitoidentity.set_identity_pool_roles my_id_pool_id        
↪   AuthenticatedRole=my_auth_role  # this will set the auth role and clear the
↪unauth role
salt myminion boto_cognitoidentity.set_identity_pool_roles my_id_pool_id        
↪   UnauthenticatedRole=my_unauth_role  # this will set the unauth role and
↪clear the auth role
```

salt.modules.boto_cognitoidentity.**update_identity_pool**(*IdentityPoolId, IdentityPoolName=None, AllowUnauthenticatedIdentities=False, SupportedLoginProviders=None, DeveloperProviderName=None, OpenIdConnectProviderARNs=None, region=None, key=None, keyid=None, profile=None*)

Updates the given IdentityPoolId's properties. All parameters except for IdentityPoolId, is optional. SupportedLoginProviders should be a dictionary mapping provider names to provider app IDs. OpenIdConnectProviderARNs should be a list of OpenID Connect provider ARNs.

To clear SupportedLoginProviders pass '{}'

To clear OpenIdConnectProviderARNs pass '[]'

boto3 api prevents DeveloperProviderName to be updated after it has been set for the first time.

Returns the updated identity pool if successful

CLI Example:

```
salt myminion boto_cognitoidentity.update_identity_pool my_id_pool_id my_id_pool_
↪name                              DeveloperProviderName=custom_developer_
↪provider
```

21.9.44 salt.modules.boto_datapipeline module

Connection module for Amazon Data Pipeline

New in version 2016.3.0.

depends boto3

salt.modules.boto_datapipeline.**activate_pipeline**(*pipeline_id*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Start processing pipeline tasks. This function is idempotent.

CLI example:

```
salt myminion boto_datapipeline.activate_pipeline my_pipeline_id
```

salt.modules.boto_datapipeline.**create_pipeline**(*name*, *unique_id*, *description=''*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Create a new, empty pipeline. This function is idempotent.

CLI example:

```
salt myminion boto_datapipeline.create_pipeline my_name my_unique_id
```

salt.modules.boto_datapipeline.**delete_pipeline**(*pipeline_id*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Delete a pipeline, its pipeline definition, and its run history. This function is idempotent.

CLI example:

```
salt myminion boto_datapipeline.delete_pipeline my_pipeline_id
```

salt.modules.boto_datapipeline.**describe_pipelines**(*pipeline_ids*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Retrieve metadata about one or more pipelines.

CLI example:

```
salt myminion boto_datapipeline.describe_pipelines ['my_pipeline_id']
```

salt.modules.boto_datapipeline.**get_pipeline_definition**(*pipeline_id*, *version='latest'*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Get the definition of the specified pipeline.

CLI example:

```
salt myminion boto_datapipeline.get_pipeline_definition my_pipeline_id
```

`salt.modules.boto_datapipeline.`**`list_pipelines`**`(`*region=None*, *key=None*, *keyid=None*, *profile=None*`)`

Get a list of pipeline ids and names for all pipelines.

CLI Example:

```
salt myminion boto_datapipeline.list_pipelines profile=myprofile
```

`salt.modules.boto_datapipeline.`**`pipeline_id_from_name`**`(`*name*, *region=None*, *key=None*, *keyid=None*, *profile=None*`)`

Get the pipeline id, if it exists, for the given name.

CLI example:

```
salt myminion boto_datapipeline.pipeline_id_from_name my_pipeline_name
```

`salt.modules.boto_datapipeline.`**`put_pipeline_definition`**`(`*pipeline_id*, *pipeline_objects*, *parameter_objects=None*, *parameter_values=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*`)`

Add tasks, schedules, and preconditions to the specified pipeline. This function is idempotent and will replace an existing definition.

CLI example:

```
salt myminion boto_datapipeline.put_pipeline_definition my_pipeline_id my_
↪pipeline_objects
```

21.9.45 salt.modules.boto_dynamodb

Connection module for Amazon DynamoDB

New in version 2015.5.0.

configuration This module accepts explicit DynamoDB credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More Information available at:

```
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-
↪ec2.html
```

If IAM roles are not used you need to specify them either in a pillar or in the minion's config file:

```
keyid: GKTADJGHEIQSXMKKRBJ08H
key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

A region may also be specified in the configuration:

```
region: us-east-1
```

If a region is not specified, the default is us-east-1.

It's also possible to specify key, keyid and region via a profile, either as a passed in dict, or as a string to pull from pillars or minion config:

```
myprofile:
    keyid: GKTADJGHEIQSXMKKRBJ08H
    key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
    region: us-east-1
```

depends boto

`salt.modules.boto_dynamodb.`**`create_global_secondary_index`**(*table_name, global_index, region=None, key=None, keyid=None, profile=None*)

Creates a single global secondary index on a DynamoDB table.

CLI Example: .. code-block:: bash

salt myminion boto_dynamodb.create_global_secondary_index table_name / index_name

`salt.modules.boto_dynamodb.`**`create_table`**(*table_name, region=None, key=None, keyid=None, profile=None, read_capacity_units=None, write_capacity_units=None, hash_key=None, hash_key_data_type=None, range_key=None, range_key_data_type=None, local_indexes=None, global_indexes=None*)

Creates a DynamoDB table.

CLI Example:

```
salt myminion boto_dynamodb.create_table table_name /
region=us-east-1 /
hash_key=id /
hash_key_data_type=N /
range_key=created_at /
range_key_data_type=N /
read_capacity_units=1 /
write_capacity_units=1
```

`salt.modules.boto_dynamodb.`**`delete`**(*table_name, region=None, key=None, keyid=None, profile=None*)

Delete a DynamoDB table.

CLI Example:

```
salt myminion boto_dynamodb.delete table_name region=us-east-1
```

`salt.modules.boto_dynamodb.`**`describe`**(*table_name, region=None, key=None, keyid=None, profile=None*)

Describe a DynamoDB table.

CLI example:

```
salt myminion boto_dynamodb.describe table_name region=us-east-1
```

`salt.modules.boto_dynamodb.`**`exists`**(*table_name, region=None, key=None, keyid=None, profile=None*)

Check to see if a table exists.

CLI Example:

```
salt myminion boto_dynamodb.exists table_name region=us-east-1
```

`salt.modules.boto_dynamodb.`**`extract_index`**(*index_data*, *global_index=False*)

Instantiates and returns an AllIndex object given a valid index configuration

CLI Example: salt myminion boto_dynamodb.extract_index index

`salt.modules.boto_dynamodb.`**`update`**(*table_name*, *throughput=None*, *global_indexes=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Update a DynamoDB table.

CLI example:

```
salt myminion boto_dynamodb.update table_name region=us-east-1
```

`salt.modules.boto_dynamodb.`**`update_global_secondary_index`**(*table_name*, *global_indexes*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Updates the throughput of the given global secondary indexes.

CLI Example: .. code-block:: bash

salt myminion boto_dynamodb.update_global_secondary_index table_name / indexes

21.9.46 salt.modules.boto_ec2

Connection module for Amazon EC2

New in version 2015.8.0.

configuration This module accepts explicit EC2 credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More Information available here.

If IAM roles are not used you need to specify them either in a pillar or in the minion's config file:

```
ec2.keyid: GKTADJGHEIQSXMKKRBJ08H
ec2.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

A region may also be specified in the configuration:

```
ec2.region: us-east-1
```

If a region is not specified, the default is us-east-1.

It's also possible to specify key, keyid, and region via a profile, either as a passed in dict, or as a string to pull from pillars or minion config:

```
myprofile:
  keyid: GKTADJGHEIQSXMKKRBJ08H
  key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
  region: us-east-1
```

depends boto

`salt.modules.boto_ec2.`**`allocate_eip_address`**(*domain=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Allocate a new Elastic IP address and associate it with your account.

domain (string) Optional param - if set to exactly 'vpc', the address will be allocated to the VPC. The default simply maps the EIP to your account container.

returns (dict) dict of 'interesting' information about the newly allocated EIP, with probably the most interesting keys being 'public_ip'; and 'allocation_id' iff 'domain=vpc' was passed.

CLI Example:

```
salt-call boto_ec2.allocate_eip_address domain=vpc
```

New in version 2016.3.0.

`salt.modules.boto_ec2.`**`assign_private_ip_addresses`**(*network_interface_name=None, network_interface_id=None, private_ip_addresses=None, secondary_private_ip_address_count=None, allow_reassignment=False, region=None, key=None, keyid=None, profile=None*)

Assigns one or more secondary private IP addresses to a network interface.

network_interface_id (string) - ID of the network interface to associate the IP with (exclusive with 'network_interface_name')

network_interface_name (string) - Name of the network interface to associate the IP with (exclusive with 'network_interface_id')

private_ip_addresses (list) - Assigns the specified IP addresses as secondary IP addresses to the network interface (exclusive with 'secondary_private_ip_address_count')

secondary_private_ip_address_count (int) - The number of secondary IP addresses to assign to the network interface. (exclusive with 'private_ip_addresses')

allow_reassociation (bool) – Allow a currently associated EIP to be re-associated with the new instance or interface.

returns (bool) - True on success, False on failure.

CLI Example:

```
salt myminion boto_ec2.assign_private_ip_addresses network_interface_name=my_eni␣
↪private_ip_addresses=private_ip
salt myminion boto_ec2.assign_private_ip_addresses network_interface_name=my_eni␣
↪secondary_private_ip_address_count=2
```

New in version 2017.7.0.

`salt.modules.boto_ec2.`**`associate_eip_address`**(*instance_id=None, instance_name=None, public_ip=None, allocation_id=None, network_interface_id=None, network_interface_name=None, private_ip_address=None, allow_reassociation=False, region=None, key=None, keyid=None, profile=None*)

Associate an Elastic IP address with a currently running instance or a network interface. This requires exactly one of either 'public_ip' or 'allocation_id', depending on whether you're associating a VPC address or a plain EC2 address.

instance_id (string) – ID of the instance to associate with (exclusive with 'instance_name')

instance_name (string) – Name tag of the instance to associate with (exclusive with 'instance_id')

public_ip (string) – Public IP address, for standard EC2 based allocations.

allocation_id (string) – Allocation ID for a VPC-based EIP.

network_interface_id (string) - ID of the network interface to associate the EIP with

network_interface_name (string) - Name of the network interface to associate the EIP with

private_ip_address (string) – The primary or secondary private IP address to associate with the Elastic IP address.

allow_reassociation (bool) – Allow a currently associated EIP to be re-associated with the new instance or interface.

returns (bool) - True on success, False on failure.

CLI Example:

```
salt myminion boto_ec2.associate_eip_address instance_name=bubba.ho.tep␣
↪allocation_id=eipalloc-ef382c8a
```

New in version 2016.3.0.

`salt.modules.boto_ec2.`**`attach_network_interface`**(*device_index, name=None, network_interface_id=None, instance_name=None, instance_id=None, region=None, key=None, keyid=None, profile=None*)

Attach an Elastic Network Interface.

New in version 2016.3.0.

CLI Example:

```
salt myminion boto_ec2.attach_network_interface my_eni instance_name=salt-master␣
↪device_index=0
```

`salt.modules.boto_ec2.`**`attach_volume`**(*volume_id, instance_id, device, region=None, key=None, keyid=None, profile=None*)

Attach an EBS volume to an EC2 instance. ..

volume_id (string) – The ID of the EBS volume to be attached.

instance_id (string) – The ID of the EC2 instance to attach the volume to.

device (string) – The device on the instance through which the volume is exposed (e.g. /dev/sdh)

returns (bool) - True on success, False on failure.

CLI Example:

```
salt-call boto_ec2.attach_volume vol-12345678 i-87654321 /dev/sdh
```

`salt.modules.boto_ec2.`**`create_image`**(*ami_name, instance_id=None, instance_name=None, tags=None, region=None, key=None, keyid=None, profile=None, description=None, no_reboot=False, dry_run=False, filters=None*)

Given instance properties that define exactly one instance, create AMI and return AMI-id.

CLI Examples:

```
salt myminion boto_ec2.create_image ami_name instance_name=myinstance
salt myminion boto_ec2.create_image another_ami_name tags='{"mytag": "value"}'␣
↪description='this is my ami'
```

(continues on next page)

(continued from previous page)

```

```

`salt.modules.boto_ec2.`**`create_key`**`(`*key_name, save_path, region=None, key=None, keyid=None, profile=None*`)`

Creates a key and saves it to a given path. Returns the private key.

CLI Example:

```
salt myminion boto_ec2.create_key mykey /root/
```

`salt.modules.boto_ec2.`**`create_network_interface`**`(`*name, subnet_id=None, subnet_name=None, private_ip_address=None, description=None, groups=None, region=None, key=None, keyid=None, profile=None*`)`

Create an Elastic Network Interface.

New in version 2016.3.0.

CLI Example:

```
salt myminion boto_ec2.create_network_interface my_eni subnet-12345␣
↪description=my_eni groups=['my_group']
```

`salt.modules.boto_ec2.`**`create_tags`**`(`*resource_ids, tags, region=None, key=None, keyid=None, profile=None*`)`

Create new metadata tags for the specified resource ids.

New in version 2016.11.0.

resource_ids (string) or (list) – List of resource IDs. A plain string will be converted to a list of one element.

tags (dict) – Dictionary of name/value pairs. To create only a tag name, pass ” as the value.

returns (bool) - True on success, False on failure.

CLI Example:

```
salt-call boto_ec2.create_tags vol-12345678 '{"Name": "myVolume01"}'
```

`salt.modules.boto_ec2.`**`create_volume`**`(`*zone_name, size=None, snapshot_id=None, volume_type=None, iops=None, encrypted=False, kms_key_id=None, wait_for_creation=False, region=None, key=None, keyid=None, profile=None*`)`

Create an EBS volume to an availability zone.

zone_name (string) – The Availability zone name of the EBS volume to be created.

size

(int) – The size of the new volume, in GiB. If you're creating the volume from a snapshot and don't specify a volume size, the default is the snapshot size.

snapshot_id (string) – The snapshot ID from which the new volume will be created.

volume_type

(string) - The type of the volume. Valid volume types for AWS can be found here: http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSVolumeTypes.html

iops (int) - The provisioned IOPS you want to associate with this volume.

encrypted (bool) - Specifies whether the volume should be encrypted.

kms_key_id

(string) - If encrypted is True, this KMS Key ID may be specified to encrypt volume with this key e.g.: arn:aws:kms:us-east-1:012345678910:key/abcd1234-a123-456a-a12b-a123b4cd56ef

wait_for_creation (bool) - Whether or not to wait for volume creation to complete.

returns (string) - created volume id on success, error message on failure.

CLI Example:

```
salt-call boto_ec2.create_volume us-east-1a size=10
salt-call boto_ec2.create_volume us-east-1a snapshot_id=snap-0123abcd
```

`salt.modules.boto_ec2.`**`delete_key`**(*key_name, region=None, key=None, keyid=None, profile=None*)

Deletes a key. Always returns True

CLI Example:

```
salt myminion boto_ec2.delete_key mykey
```

`salt.modules.boto_ec2.`**`delete_network_interface`**(*name=None, network_interface_id=None, region=None, key=None, keyid=None, profile=None*)

Create an Elastic Network Interface.

New in version 2016.3.0.

CLI Example:

```
salt myminion boto_ec2.create_network_interface my_eni subnet-12345
↪description=my_eni groups=['my_group']
```

`salt.modules.boto_ec2.`**`delete_tags`**(*resource_ids, tags, region=None, key=None, keyid=None, profile=None*)

Delete metadata tags for the specified resource ids.

New in version 2016.11.0.

resource_ids (string) or (list) – List of resource IDs. A plain string will be converted to a list of one element.

tags

(dict) or (list) – Either a dictionary containing name/value pairs or a list containing just tag names. If you pass in a dictionary, the values must match the actual tag values or the tag will not be deleted. If you pass in a value of None for the tag value, all tags with that name will be deleted.

returns (bool) - True on success, False on failure.

CLI Example:

```
salt-call boto_ec2.delete_tags vol-12345678 '{"Name": "myVolume01"}'
salt-call boto_ec2.delete_tags vol-12345678 '["Name","MountPoint"]'
```

`salt.modules.boto_ec2.`**`delete_volume`**(*volume_id, instance_id=None, device=None, force=False, region=None, key=None, keyid=None, profile=None*)

Detach an EBS volume from an EC2 instance.

New in version 2016.11.0.

volume_id (string) – The ID of the EBS volume to be deleted.

force (bool) – Forces deletion even if the device has not yet been detached from its instance.

returns (bool) - True on success, False on failure.

CLI Example:

```
salt-call boto_ec2.delete_volume vol-12345678
```

`salt.modules.boto_ec2.`**`detach_network_interface`**`(`*name=None, network_interface_id=None, attachment_id=None, force=False, region=None, key=None, keyid=None, profile=None*`)`

Detach an Elastic Network Interface.

New in version 2016.3.0.

CLI Example:

```
salt myminion boto_ec2.detach_network_interface my_eni
```

`salt.modules.boto_ec2.`**`detach_volume`**`(`*volume_id, instance_id=None, device=None, force=False, wait_for_detachement=False, region=None, key=None, keyid=None, profile=None*`)`

Detach an EBS volume from an EC2 instance.

New in version 2016.11.0.

volume_id (string) – The ID of the EBS volume to be detached.

instance_id (string) – The ID of the EC2 instance from which it will be detached.

device (string) – The device on the instance through which the volume is exposted (e.g. /dev/sdh)

force

(bool) – Forces detachment if the previous detachment attempt did not occur cleanly. This option can lead to data loss or a corrupted file system. Use this option only as a last resort to detach a volume from a failed instance. The instance will not have an opportunity to flush file system caches nor file system meta data. If you use this option, you must perform file system check and repair procedures.

wait_for_detachement (bool) - Whether or not to wait for volume detachement to complete.

returns (bool) - True on success, False on failure.

CLI Example:

```
salt-call boto_ec2.detach_volume vol-12345678 i-87654321
```

`salt.modules.boto_ec2.`**`disassociate_eip_address`**`(`*public_ip=None, association_id=None, region=None, key=None, keyid=None, profile=None*`)`

Disassociate an Elastic IP address from a currently running instance. This requires exactly one of either 'association_id' or 'public_ip', depending on whether you're dealing with a VPC or EC2 Classic address.

public_ip (string) – Public IP address, for EC2 Classic allocations.

association_id (string) – Association ID for a VPC-bound EIP.

returns (bool) - True on success, False on failure.

CLI Example:

```
salt myminion boto_ec2.disassociate_eip_address association_id=eipassoc-e3ba2d16
```

New in version 2016.3.0.

`salt.modules.boto_ec2.`**`exists`**(*instance_id=None, name=None, tags=None, region=None, key=None, keyid=None, profile=None, in_states=None, filters=None*)

Given an instance id, check to see if the given instance id exists.

Returns True if the given instance with the given id, name, or tags exists; otherwise, False is returned.

CLI Example:

```
salt myminion boto_ec2.exists myinstance
```

`salt.modules.boto_ec2.`**`find_images`**(*ami_name=None, executable_by=None, owners=None, image_ids=None, tags=None, region=None, key=None, keyid=None, profile=None, return_objs=False*)

Given image properties, find and return matching AMI ids

CLI Examples:

```
salt myminion boto_ec2.find_images tags='{"mytag": "value"}'
```

`salt.modules.boto_ec2.`**`find_instances`**(*instance_id=None, name=None, tags=None, region=None, key=None, keyid=None, profile=None, return_objs=False, in_states=None, filters=None*)

Given instance properties, find and return matching instance ids

CLI Examples:

```
salt myminion boto_ec2.find_instances # Lists all instances
salt myminion boto_ec2.find_instances name=myinstance
salt myminion boto_ec2.find_instances tags='{"mytag": "value"}'
salt myminion boto_ec2.find_instances filters='{"vpc-id": "vpc-12345678"}'
```

`salt.modules.boto_ec2.`**`get_all_eip_addresses`**(*addresses=None, allocation_ids=None, region=None, key=None, keyid=None, profile=None*)

Get public addresses of some, or all EIPs associated with the current account.

addresses (list) - Optional list of addresses. If provided, only the addresses associated with those in the list will be returned.

allocation_ids (list) - Optional list of allocation IDs. If provided, only the addresses associated with the given allocation IDs will be returned.

returns (list) - A list of the requested EIP addresses

CLI Example:

```
salt-call boto_ec2.get_all_eip_addresses
```

New in version 2016.3.0.

`salt.modules.boto_ec2.`**`get_all_tags`**(*filters=None, region=None, key=None, keyid=None, profile=None*)

Describe all tags matching the filter criteria, or all tags in the account otherwise.

New in version 2018.3.0.

filters (dict) - Additional constraints on which volumes to return. Note that valid filters vary extensively depending on the resource type. When in doubt, search first without a filter and then use the returned data to help fine-tune your search. You can generally garner the resource type from its ID (e.g. *vol-XXXXX* is a volume, *i-XXXXX* is an instance, etc.

CLI Example:

```
salt-call boto_ec2.get_all_tags '{"tag:Name": myInstanceNameTag, resource-type:
↪instance}'
```

salt.modules.boto_ec2.**get_all_volumes**(*volume_ids=None, filters=None, return_objs=False, region=None, key=None, keyid=None, profile=None*)

Get a list of all EBS volumes, optionally filtered by provided 'filters' param

New in version 2016.11.0.

volume_ids (list) - Optional list of volume_ids. If provided, only the volumes associated with those in the list will be returned.

filters (dict) - Additional constraints on which volumes to return. Valid filters are:

- attachment.attach-time - The time stamp when the attachment initiated.
- attachment.delete-on-termination - Whether the volume is deleted on instance termination.
- attachment.device - The device name that is exposed to the instance (for example, /dev/sda1).
- attachment.instance-id - The ID of the instance the volume is attached to.
- attachment.status - The attachment state (attaching | attached | detaching | detached).
- availability-zone - The Availability Zone in which the volume was created.
- create-time - The time stamp when the volume was created.
- encrypted - The encryption status of the volume.
- size - The size of the volume, in GiB.
- snapshot-id - The snapshot from which the volume was created.
- status - The status of the volume (creating | available | in-use | deleting | deleted | error).
- tag:key=value - The key/value combination of a tag assigned to the resource.
- volume-id - The volume ID.
- volume-type - The Amazon EBS volume type. This can be `gp2` for General Purpose SSD, `io1` for Provisioned IOPS SSD, `st1` for Throughput Optimized HDD, `sc1` for Cold HDD, or `standard` for Magnetic volumes.

return_objs (bool) - Changes the return type from list of volume IDs to list of boto.ec2.volume.Volume objects

returns (list) - A list of the requested values: Either the volume IDs or, if return_objs is `True`, boto.ec2.volume.Volume objects.

CLI Example:

```
salt-call boto_ec2.get_all_volumes filters='{"tag:Name": "myVolume01"}'
```

salt.modules.boto_ec2.**get_attribute**(*attribute, instance_name=None, instance_id=None, region=None, key=None, keyid=None, profile=None, filters=None*)

Get an EC2 instance attribute.

CLI Example:

```
salt myminion boto_ec2.get_attribute sourceDestCheck instance_name=my_instance
```

Available attributes:

- instanceType
- kernel
- ramdisk
- userData
- disableApiTermination
- instanceInitiatedShutdownBehavior
- rootDeviceName
- blockDeviceMapping
- productCodes
- sourceDestCheck
- groupSet
- ebsOptimized
- sriovNetSupport

`salt.modules.boto_ec2.`**`get_eip_address_info`**(*addresses=None, allocation_ids=None, region=None, key=None, keyid=None, profile=None*)

Get 'interesting' info about some, or all EIPs associated with the current account.

addresses (list) - Optional list of addresses. If provided, only the addresses associated with those in the list will be returned.

allocation_ids (list) - Optional list of allocation IDs. If provided, only the addresses associated with the given allocation IDs will be returned.

returns (list of dicts) - A list of dicts, each containing the info for one of the requested EIPs.

CLI Example:

```
salt-call boto_ec2.get_eip_address_info addresses=52.4.2.15
```

New in version 2016.3.0.

`salt.modules.boto_ec2.`**`get_id`**(*name=None, tags=None, region=None, key=None, keyid=None, profile=None, in_states=None, filters=None*)

Given instance properties, return the instance id if it exists.

CLI Example:

```
salt myminion boto_ec2.get_id myinstance
```

`salt.modules.boto_ec2.`**`get_key`**(*key_name, region=None, key=None, keyid=None, profile=None*)

Check to see if a key exists. Returns fingerprint and name if it does and False if it doesn't CLI Example:

```
salt myminion boto_ec2.get_key mykey
```

`salt.modules.boto_ec2.`**`get_keys`**(*keynames=None, filters=None, region=None, key=None, keyid=None, profile=None*)

Gets all keys or filters them by name and returns a list. keynames (list):: A list of the names of keypairs to retrieve. If not provided, all key pairs will be returned. filters (dict) :: Optional filters that can be used to limit the results returned. Filters are provided in the form of a dictionary consisting of filter names as the key and filter values as the value. The set of allowable filter names/values is dependent on the request being performed. Check the EC2 API guide for details.

CLI Example:

```
salt myminion boto_ec2.get_keys
```

`salt.modules.boto_ec2.`**`get_network_interface`**`(`*name=None, network_interface_id=None, region=None, key=None, keyid=None, profile=None*`)`

Get an Elastic Network Interface.

New in version 2016.3.0.

CLI Example:

```
salt myminion boto_ec2.get_network_interface name=my_eni
```

`salt.modules.boto_ec2.`**`get_network_interface_id`**`(`*name, region=None, key=None, keyid=None, profile=None*`)`

Get an Elastic Network Interface id from its name tag.

New in version 2016.3.0.

CLI Example:

```
salt myminion boto_ec2.get_network_interface_id name=my_eni
```

`salt.modules.boto_ec2.`**`get_tags`**`(`*instance_id=None, keyid=None, key=None, profile=None, region=None*`)`

Given an instance_id, return a list of tags associated with that instance.

returns (list) - list of tags as key/value pairs

CLI Example:

```
salt myminion boto_ec2.get_tags instance_id
```

`salt.modules.boto_ec2.`**`get_unassociated_eip_address`**`(`*domain='standard', region=None, key=None, keyid=None, profile=None*`)`

Return the first unassociated EIP

domain Indicates whether the address is an EC2 address or a VPC address (standard|vpc).

CLI Example:

```
salt-call boto_ec2.get_unassociated_eip_address
```

New in version 2016.3.0.

`salt.modules.boto_ec2.`**`get_zones`**`(`*region=None, key=None, keyid=None, profile=None*`)`

Get a list of AZs for the configured region.

CLI Example:

```
salt myminion boto_ec2.get_zones
```

`salt.modules.boto_ec2.`**`import_key`**`(`*key_name, public_key_material, region=None, key=None, keyid=None, profile=None*`)`

Imports the public key from an RSA key pair that you created with a third-party tool. Supported formats: - OpenSSH public key format (e.g., the format in ~/.ssh/authorized_keys) - Base64 encoded DER format - SSH public key file format as specified in RFC4716 - DSA keys are not supported. Make sure your key generator is set up to create RSA keys. Supported lengths: 1024, 2048, and 4096.

CLI Example:

```
salt myminion boto_ec2.import mykey publickey
```

`salt.modules.boto_ec2.`**`modify_network_interface_attribute`**(*name=None, network_interface_id=None, attr=None, value=None, region=None, key=None, keyid=None, profile=None*)

Modify an attribute of an Elastic Network Interface.

New in version 2016.3.0.

CLI Example:

```
salt myminion boto_ec2.modify_network_interface_attribute my_eni attr=description␣
↪value='example description'
```

`salt.modules.boto_ec2.`**`release_eip_address`**(*public_ip=None, allocation_id=None, region=None, key=None, keyid=None, profile=None*)

Free an Elastic IP address. Pass either a public IP address to release an EC2 Classic EIP, or an AllocationId to release a VPC EIP.

public_ip (string) - The public IP address - for EC2 elastic IPs.

allocation_id (string) - The Allocation ID - for VPC elastic IPs.

returns (bool) - True on success, False on failure

CLI Example:

```
salt myminion boto_ec2.release_eip_address allocation_id=eipalloc-ef382c8a
```

New in version 2016.3.0.

`salt.modules.boto_ec2.`**`run`**(*image_id, name=None, tags=None, key_name=None, security_groups=None, user_data=None, instance_type='m1.small', placement=None, kernel_id=None, ramdisk_id=None, monitoring_enabled=None, vpc_id=None, vpc_name=None, subnet_id=None, subnet_name=None, private_ip_address=None, block_device_map=None, disable_api_termination=None, instance_initiated_shutdown_behavior=None, placement_group=None, client_token=None, security_group_ids=None, security_group_names=None, additional_info=None, tenancy=None, instance_profile_arn=None, instance_profile_name=None, ebs_optimized=None, network_interface_id=None, network_interface_name=None, region=None, key=None, keyid=None, profile=None, network_interfaces=None*)

Create and start an EC2 instance.

Returns True if the instance was created; otherwise False.

CLI Example:

```
salt myminion boto_ec2.run ami-b80c2b87 name=myinstance
```

image_id (string) – The ID of the image to run.

name (string) - The name of the instance.

tags (dict of key: value pairs) - tags to apply to the instance.

key_name (string) – The name of the key pair with which to launch instances.

security_groups (list of strings) – The names of the EC2 classic security groups with which to associate instances

user_data (string) – The Base64-encoded MIME user data to be made available to the instance(s) in this reservation.

instance_type (string) – The type of instance to run. Note that some image types (e.g. hvm) only run on some instance types.

placement (string) – The Availability Zone to launch the instance into.

kernel_id (string) – The ID of the kernel with which to launch the instances.

ramdisk_id (string) – The ID of the RAM disk with which to launch the instances.

monitoring_enabled (bool) – Enable detailed CloudWatch monitoring on the instance.

vpc_id (string) - ID of a VPC to bind the instance to. Exclusive with vpc_name.

vpc_name (string) - Name of a VPC to bind the instance to. Exclusive with vpc_id.

subnet_id (string) – The subnet ID within which to launch the instances for VPC.

subnet_name (string) – The name of a subnet within which to launch the instances for VPC.

private_ip_address (string) – If you're using VPC, you can optionally use this parameter to assign the instance a specific available IP address from the subnet (e.g. 10.0.0.25).

block_device_map (boto.ec2.blockdevicemapping.BlockDeviceMapping) – A BlockDeviceMapping data structure describing the EBS volumes associated with the Image. (string) - A string representation of a BlockDeviceMapping structure (dict) - A dict describing a BlockDeviceMapping structure

YAML example:

```
device-maps:
    /dev/sdb:
        ephemeral_name: ephemeral0
    /dev/sdc:
        ephemeral_name: ephemeral1
    /dev/sdd:
        ephemeral_name: ephemeral2
    /dev/sde:
        ephemeral_name: ephemeral3
    /dev/sdf:
        size: 20
        volume_type: gp2
```

disable_api_termination (bool) – If True, the instances will be locked and will not be able to be terminated via the API.

instance_initiated_shutdown_behavior (string) – Specifies whether the instance stops or terminates on instance-initiated shutdown. Valid values are: stop, terminate

placement_group (string) – If specified, this is the name of the placement group in which the instance(s) will be launched.

client_token (string) – Unique, case-sensitive identifier you provide to ensure idempotency of the request. Maximum 64 ASCII characters.

security_group_ids (list of strings) – The ID(s) of the VPC security groups with which to associate instances.

security_group_names (list of strings) – The name(s) of the VPC security groups with which to associate instances.

additional_info (string) – Specifies additional information to make available to the instance(s).

tenancy (string) – The tenancy of the instance you want to launch. An instance with a tenancy of 'dedicated' runs on single-tenant hardware and can only be launched into a VPC. Valid values are:"default" or "dedicated". NOTE: To use dedicated tenancy you MUST specify a VPC subnet-ID as well.

instance_profile_arn (string) – The Amazon resource name (ARN) of the IAM Instance Profile (IIP) to associate with the instances.

instance_profile_name (string) – The name of the IAM Instance Profile (IIP) to associate with the instances.

ebs_optimized (bool) – Whether the instance is optimized for EBS I/O. This optimization provides dedicated throughput to Amazon EBS and an optimized configuration stack to provide optimal EBS I/O performance. This optimization isn't available with all instance types.

network_interfaces (boto.ec2.networkinterface.NetworkInterfaceCollection) – A NetworkInterfaceCollection data structure containing the ENI specifications for the instance.

network_interface_id (string) - ID of the network interface to attach to the instance

network_interface_name (string) - Name of the network interface to attach to the instance

`salt.modules.boto_ec2.`**`set_attribute`**(*attribute*, *attribute_value*, *instance_name=None*, *instance_id=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*, *filters=None*)

Set an EC2 instance attribute. Returns whether the operation succeeded or not.

CLI Example:

```
salt myminion boto_ec2.set_attribute sourceDestCheck False instance_name=my_
↪instance
```

Available attributes:

- instanceType
- kernel
- ramdisk
- userData
- disableApiTermination
- instanceInitiatedShutdownBehavior
- rootDeviceName
- blockDeviceMapping
- productCodes
- sourceDestCheck

- groupSet
- ebsOptimized
- sriovNetSupport

`salt.modules.boto_ec2.`**`set_volumes_tags`**(*tag_maps, authoritative=False, dry_run=False, region=None, key=None, keyid=None, profile=None*)

New in version 2016.11.0.

tag_maps (list) List of dicts of filters and tags, where 'filters' is a dict suitable for passing to the 'filters' argument of get_all_volumes() above, and 'tags' is a dict of tags to be set on volumes (via create_tags/delete_tags) as matched by the given filters. The filter syntax is extended to permit passing either a list of volume_ids or an instance_name (with instance_name being the Name tag of the instance to which the desired volumes are mapped). Each mapping in the list is applied separately, so multiple sets of volumes can be all tagged differently with one call to this function. If filtering by instance Name, You may additionally limit the instances matched by passing in a list of desired instance states. The default set of states is ('pending', 'rebooting', 'running', 'stopping', 'stopped').

YAML example fragment:

```
- filters:
    attachment.instance_id: i-abcdef12
  tags:
    Name: dev-int-abcdef12.aws-foo.com
- filters:
    attachment.device: /dev/sdf
  tags:
    ManagedSnapshots: true
    BillingGroup: bubba.hotep@aws-foo.com
  in_states:
  - stopped
  - terminated
- filters:
    instance_name: prd-foo-01.aws-foo.com
  tags:
    Name: prd-foo-01.aws-foo.com
    BillingGroup: infra-team@aws-foo.com
- filters:
    volume_ids: [ vol-12345689, vol-abcdef12 ]
  tags:
    BillingGroup: infra-team@aws-foo.com
```

authoritative (bool) If true, any existing tags on the matched volumes, and not explicitly requested here, will be removed.

dry_run (bool) If true, don't change anything, just return a dictionary describing any changes which would have been applied.

returns (dict) A dict describing status and any changes.

`salt.modules.boto_ec2.`**`terminate`**(*instance_id=None, name=None, region=None, key=None, keyid=None, profile=None, filters=None*)

Terminate the instance described by instance_id or name.

CLI Example:

```
salt myminion boto_ec2.terminate name=myinstance
salt myminion boto_ec2.terminate instance_id=i-a46b9f
```

`salt.modules.boto_ec2.`**`unassign_private_ip_addresses`**(*network_interface_name=None, network_interface_id=None, private_ip_addresses=None, region=None, key=None, keyid=None, profile=None*)

Unassigns one or more secondary private IP addresses from a network interface

network_interface_id (string) - ID of the network interface to associate the IP with (exclusive with 'network_interface_name')

network_interface_name (string) - Name of the network interface to associate the IP with (exclusive with 'network_interface_id')

private_ip_addresses (list) - Assigns the specified IP addresses as secondary IP addresses to the network interface.

returns (bool) - True on success, False on failure.

CLI Example:

```
salt myminion boto_ec2.unassign_private_ip_addresses network_interface_name=my_
↪eni private_ip_addresses=private_ip
```

New in version 2017.7.0.

21.9.47 salt.modules.boto_efs module

Connection module for Amazon EFS

New in version 2017.7.0.

configuration This module accepts explicit EFS credentials but can also utilize IAM roles assigned to the instance through Instance Profiles or it can read them from the ~/.aws/credentials file or from these environment variables: AWS_ACCESS_KEY_ID, AWS_SECRET_ACCESS_KEY. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More information available at:

```
http://docs.aws.amazon.com/efs/latest/ug/
    access-control-managing-permissions.html

http://boto3.readthedocs.io/en/latest/guide/
    configuration.html#guide-configuration
```

If IAM roles are not used you need to specify them either in a pillar or in the minion's config file

```
efs.keyid: GKTADJGHEIQSXMKKRBJ08H
efs.key: askd+ghsdfjkghWupU/asdflkdfklgjsdfjajkghs
```

A region may also be specified in the configuration

```
efs.region: us-east-1
```

If a region is not specified, the default is us-east-1.

It's also possible to speficy key, keyid, and region via a profile, either as a passed in dict, or as a string to pull from pillars or minion config:

```
myprofile:
  keyid: GKTADJGHEIQSXMKKRBJ08H
  key: askd+ghsdfjkghWupU/asdflkdfklgjsdfjajkghs
  region: us-east-1
```

depends boto3

`salt.modules.boto_efs.`**`create_file_system`**`(`*name, performance_mode='generalPurpose', keyid=None, key=None, profile=None, region=None, creation_token=None, **kwargs*`)`

Creates a new, empty file system.

name (string) - The name for the new file system

performance_mode (string) - The PerformanceMode of the file system. Can be either generalPurpose or maxIO

creation_token (string) - A unique name to be used as reference when creating an EFS. This will ensure idempotency. Set to name if not specified otherwise

returns (dict) - A dict of the data for the elastic file system

CLI Example:

```
salt 'my-minion' boto_efs.create_file_system efs-name generalPurpose
```

`salt.modules.boto_efs.`**`create_mount_target`**`(`*filesystemid, subnetid, ipaddress=None, securitygroups=None, keyid=None, key=None, profile=None, region=None, **kwargs*`)`

Creates a mount target for a file system. You can then mount the file system on EC2 instances via the mount target.

You can create one mount target in each Availability Zone in your VPC. All EC2 instances in a VPC within a given Availability Zone share a single mount target for a given file system.

If you have multiple subnets in an Availability Zone, you create a mount target in one of the subnets. EC2 instances do not need to be in the same subnet as the mount target in order to access their file system.

filesystemid (string) - ID of the file system for which to create the mount target.

subnetid (string) - ID of the subnet to add the mount target in.

ipaddress

(string) - Valid IPv4 address within the address range of the specified subnet.

securitygroups

(list[string]) - Up to five VPC security group IDs, of the form sg-xxxxxxxx. These must be for the same VPC as subnet specified.

returns (dict) - A dict of the response data

CLI Example:

```
salt 'my-minion' boto_efs.create_mount_target filesystemid subnetid
```

`salt.modules.boto_efs.`**`create_tags`**`(`*filesystemid, tags, keyid=None, key=None, profile=None, region=None, **kwargs*`)`

Creates or overwrites tags associated with a file system. Each tag is a key-value pair. If a tag key specified in the request already exists on the file system, this operation overwrites its value with the value provided in the request.

filesystemid (string) - ID of the file system for whose tags will be modified.

tags (dict) - The tags to add to the file system

CLI Example:

```
salt 'my-minion' boto_efs.create_tags
```

salt.modules.boto_efs.**delete_file_system**(*filesystemid, keyid=None, key=None, profile=None, region=None, **kwargs*)

Deletes a file system, permanently severing access to its contents. Upon return, the file system no longer exists and you can't access any contents of the deleted file system. You can't delete a file system that is in use. That is, if the file system has any mount targets, you must first delete them.

filesystemid (string) - ID of the file system to delete.

CLI Example:

```
salt 'my-minion' boto_efs.delete_file_system filesystemid
```

salt.modules.boto_efs.**delete_mount_target**(*mounttargetid, keyid=None, key=None, profile=None, region=None, **kwargs*)

Deletes the specified mount target.

This operation forcibly breaks any mounts of the file system via the mount target that is being deleted, which might disrupt instances or applications using those mounts. To avoid applications getting cut off abruptly, you might consider unmounting any mounts of the mount target, if feasible. The operation also deletes the associated network interface. Uncommitted writes may be lost, but breaking a mount target using this operation does not corrupt the file system itself. The file system you created remains. You can mount an EC2 instance in your VPC via another mount target.

mounttargetid (string) - ID of the mount target to delete

CLI Example:

```
salt 'my-minion' boto_efs.delete_mount_target mounttargetid
```

salt.modules.boto_efs.**delete_tags**(*filesystemid, tags, keyid=None, key=None, profile=None, region=None, **kwargs*)

Deletes the specified tags from a file system.

filesystemid (string) - ID of the file system for whose tags will be removed.

tags (list[string]) - The tag keys to delete to the file system

CLI Example:

```
salt 'my-minion' boto_efs.delete_tags
```

salt.modules.boto_efs.**get_file_systems**(*filesystemid=None, keyid=None, key=None, profile=None, region=None, creation_token=None, **kwargs*)

Get all EFS properties or a specific instance property if filesystemid is specified

filesystemid (string) - ID of the file system to retrieve properties

creation_token (string) - A unique token that identifies an EFS. If fileysystem created via create_file_system this would either be explictitly passed in or set to name. You can limit your search with this.

returns (list[dict]) - list of all elastic file system properties

CLI Example:

```
salt 'my-minion' boto_efs.get_file_systems efs-id
```

salt.modules.boto_efs.**get_mount_targets**(*filesystemid=None*, *mounttargetid=None*, *keyid=None*, *key=None*, *profile=None*, *region=None*, ***kwargs*)

Get all the EFS mount point properties for a specific filesystemid or the properties for a specific mounttargetid. One or the other must be specified

filesystemid

(string) - ID of the file system whose mount targets to list Must be specified if mounttargetid is not

mounttargetid

(string) - ID of the mount target to have its properties returned Must be specified if filesystemid is not

returns (list[dict]) - list of all mount point properties

CLI Example:

```
salt 'my-minion' boto_efs.get_mount_targets
```

salt.modules.boto_efs.**get_tags**(*filesystemid*, *keyid=None*, *key=None*, *profile=None*, *region=None*, ***kwargs*)

Return the tags associated with an EFS instance.

filesystemid (string) - ID of the file system whose tags to list

returns (list) - list of tags as key/value pairs

CLI Example:

```
salt 'my-minion' boto_efs.get_tags efs-id
```

salt.modules.boto_efs.**set_security_groups**(*mounttargetid*, *securitygroup*, *keyid=None*, *key=None*, *profile=None*, *region=None*, ***kwargs*)

Modifies the set of security groups in effect for a mount target

mounttargetid (string) - ID of the mount target whose security groups will be modified

securitygroups (list[string]) - list of no more than 5 VPC security group IDs.

CLI Example:

```
salt 'my-minion' boto_efs.set_security_groups my-mount-target-id my-sec-group
```

21.9.48 salt.modules.boto_elasticache

Connection module for Amazon Elasticache

New in version 2014.7.0.

configuration This module accepts explicit elasticache credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More Information available at:

```
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-
↪ec2.html
```

If IAM roles are not used you need to specify them either in a pillar or in the minion's config file:

```
elasticache.keyid: GKTADJGHEIQSXMKKRBJ08H
elasticache.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

A region may also be specified in the configuration:

```
elasticache.region: us-east-1
```

If a region is not specified, the default is us-east-1.

It's also possible to specify key, keyid and region via a profile, either as a passed in dict, or as a string to pull from pillars or minion config:

```
myprofile:
    keyid: GKTADJGHEIQSXMKKRBJ08H
    key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
    region: us-east-1
```

depends boto

`salt.modules.boto_elasticache.`**`authorize_cache_security_group_ingress`**`(`*name, ec2_security_group_name, ec2_security_group_owner_id, region=None, key=None, keyid=None, profile=None*`)`

Authorize network ingress from an ec2 security group to a cache security group.

CLI example:

```
salt myminion boto_elasticache.authorize_cache_security_group_ingress␣
↪myelasticachesg myec2sg 879879
```

`salt.modules.boto_elasticache.`**`create`**`(`*name, num_cache_nodes=None, engine=None, cache_node_type=None, replication_group_id=None, engine_version=None, cache_parameter_group_name=None, cache_subnet_group_name=None, cache_security_group_names=None, security_group_ids=None, snapshot_arns=None, preferred_availability_zone=None, preferred_maintenance_window=None, port=None, notification_topic_arn=None, auto_minor_version_upgrade=None, wait=None, region=None, key=None, keyid=None, profile=None*`)`

Create a cache cluster.

CLI example:

```
salt myminion boto_elasticache.create myelasticache 1 redis cache.t1.micro
cache_security_group_names='["myelasticachesg"]'
```

`salt.modules.boto_elasticache.`**`create_cache_security_group`**`(`*name, description, region=None, key=None, keyid=None, profile=None*`)`

Create a cache security group.

CLI example:

```
salt myminion boto_elasticache.create_cache_security_group myelasticachesg 'My⊠
↪Cache Security Group'
```

`salt.modules.boto_elasticache.`**`create_replication_group`**(*name, primary_cluster_id, replication_group_description, wait=None, region=None, key=None, keyid=None, profile=None*)

Create replication group.

CLI example:

```
salt myminion boto_elasticache.create_replication_group myelasticache⊠
↪myprimarycluster description
```

`salt.modules.boto_elasticache.`**`create_subnet_group`**(*name, description, subnet_ids=None, subnet_names=None, tags=None, region=None, key=None, keyid=None, profile=None*)

Create an ElastiCache subnet group

CLI example to create an ElastiCache subnet group:

```
salt myminion boto_elasticache.create_subnet_group my-subnet-group                ⊠
↪"group description" subnet_ids='[subnet-12345678, subnet-87654321]'
↪region=us-east-1
```

`salt.modules.boto_elasticache.`**`delete`**(*name, wait=False, region=None, key=None, keyid=None, profile=None*)

Delete a cache cluster.

CLI example:

```
salt myminion boto_elasticache.delete myelasticache
```

`salt.modules.boto_elasticache.`**`delete_cache_security_group`**(*name, region=None, key=None, keyid=None, profile=None*)

Delete a cache security group.

CLI example:

```
salt myminion boto_elasticache.delete_cache_security_group myelasticachesg 'My⊠
↪Cache Security Group'
```

`salt.modules.boto_elasticache.`**`delete_replication_group`**(*name, region=None, key=None, keyid=None, profile=None*)

Delete an ElastiCache replication group.

CLI example:

```
salt myminion boto_elasticache.delete_replication_group my-replication-group      ⊠
↪           region=us-east-1
```

salt.modules.boto_elasticache.**delete_subnet_group**(*name, region=None, key=None, keyid=None, profile=None*)

Delete an ElastiCache subnet group.

CLI example:

```
salt myminion boto_elasticache.delete_subnet_group my-subnet-group
↪ region=us-east-1
```

salt.modules.boto_elasticache.**describe_replication_group**(*name, region=None, key=None, keyid=None, profile=None, parameter=None*)

Get replication group information.

CLI example:

```
salt myminion boto_elasticache.describe_replication_group mygroup
```

salt.modules.boto_elasticache.**exists**(*name, region=None, key=None, keyid=None, profile=None*)

Check to see if a cache cluster exists.

CLI example:

```
salt myminion boto_elasticache.exists myelasticache
```

salt.modules.boto_elasticache.**get_all_cache_subnet_groups**(*name=None, region=None, key=None, keyid=None, profile=None*)

Return a list of all cache subnet groups with details

CLI example:

```
salt myminion boto_elasticache.get_all_subnet_groups region=us-east-1
```

salt.modules.boto_elasticache.**get_cache_subnet_group**(*name, region=None, key=None, keyid=None, profile=None*)

Get information about a cache subnet group.

CLI example:

```
salt myminion boto_elasticache.get_cache_subnet_group mycache_subnet_group
```

salt.modules.boto_elasticache.**get_config**(*name, region=None, key=None, keyid=None, profile=None*)

Get the configuration for a cache cluster.

CLI example:

```
salt myminion boto_elasticache.get_config myelasticache
```

salt.modules.boto_elasticache.**get_group_host**(*name, region=None, key=None, keyid=None, profile=None*)

Get hostname from replication cache group

CLI example:

```
salt myminion boto_elasticache.get_group_host myelasticachegroup
```

`salt.modules.boto_elasticache.`**`get_node_host`**`(`*name*, *region=None*, *key=None*, *keyid=None*, *profile=None*`)`

Get hostname from cache node

CLI example:

```
salt myminion boto_elasticache.get_node_host myelasticache
```

`salt.modules.boto_elasticache.`**`group_exists`**`(`*name*, *region=None*, *key=None*, *keyid=None*, *profile=None*`)`

Check to see if a replication group exists.

CLI example:

```
salt myminion boto_elasticache.group_exists myelasticache
```

`salt.modules.boto_elasticache.`**`list_cache_subnet_groups`**`(`*name=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*`)`

Return a list of all cache subnet group names

CLI example:

```
salt myminion boto_elasticache.list_subnet_groups region=us-east-1
```

`salt.modules.boto_elasticache.`**`revoke_cache_security_group_ingress`**`(`*name*, *ec2_security_group_name*, *ec2_security_group_owner_id*, *region=None*, *key=None*, *keyid=None*, *profile=None*`)`

Revoke network ingress from an ec2 security group to a cache security group.

CLI example:

```
salt myminion boto_elasticache.revoke_cache_security_group_ingress␣
↪myelasticachesg myec2sg 879879
```

`salt.modules.boto_elasticache.`**`subnet_group_exists`**`(`*name*, *tags=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*`)`

Check to see if an ElastiCache subnet group exists.

CLI example:

```
salt myminion boto_elasticache.subnet_group_exists my-param-group          ␣
↪region=us-east-1
```

21.9.49 salt.modules.boto_elasticsearch_domain module

Connection module for Amazon Elasticsearch Service

New in version 2016.11.0.

configuration This module accepts explicit AWS credentials but can also utilize IAM roles assigned to the instance trough Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More Information available at:

```
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-
↪ec2.html
```

If IAM roles are not used you need to specify them either in a pillar or in the minion's config file:

```
lambda.keyid: GKTADJGHEIQSXMKKRBJ08H
lambda.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

A region may also be specified in the configuration:

```
lambda.region: us-east-1
```

If a region is not specified, the default is us-east-1.

It's also possible to specify key, keyid and region via a profile, either as a passed in dict, or as a string to pull from pillars or minion config:

```
myprofile:
    keyid: GKTADJGHEIQSXMKKRBJ08H
    key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
    region: us-east-1
```

Create and delete methods return:

```
created: true
```

or

```
created: false
error:
  message: error message
```

Request methods (e.g., *describe_function*) return:

```
domain:
  - {...}
  - {...}
```

or

```
error:
  message: error message
```

depends boto3

`salt.modules.boto_elasticsearch_domain.`**`add_tags`**(*DomainName=None, ARN=None, region=None, key=None, keyid=None, profile=None, **kwargs*)

Add tags to a domain

Returns {tagged: true} if the domain was tagged and returns {tagged: False} if the domain was not tagged.

CLI Example:

```
salt myminion boto_elasticsearch_domain.add_tags mydomain tag_a=tag_value tag_
↪b=tag_value
```

`salt.modules.boto_elasticsearch_domain.`**`create`**(*DomainName, ElasticsearchClusterConfig=None, EBSOptions=None, AccessPolicies=None, SnapshotOptions=None, AdvancedOptions=None, region=None, key=None, keyid=None, profile=None, ElasticsearchVersion=None*)

Given a valid config, create a domain.

Returns {created: true} if the domain was created and returns {created: False} if the domain was not created.

CLI Example:

```
salt myminion boto_elasticsearch_domain.create mydomain \
      {'InstanceType': 't2.micro.elasticsearch', 'InstanceCount': 1, \
      'DedicatedMasterEnabled': false, 'ZoneAwarenessEnabled': false} \
      {'EBSEnabled': true, 'VolumeType': 'gp2', 'VolumeSize': 10, \
      'Iops': 0} \
      {"Version": "2012-10-17", "Statement": [{"Effect": "Allow", "Principal": {
↪"AWS": "*"}, "Action": "es:*", \
       "Resource": "arn:aws:es:us-east-1:111111111111:domain/mydomain/*", \
       "Condition": {"IpAddress": {"aws:SourceIp": ["127.0.0.1"]}}}]} \
      {"AutomatedSnapshotStartHour": 0} \
      {"rest.action.multi.allow_explicit_index": "true"}
```

`salt.modules.boto_elasticsearch_domain.`**`delete`**(*DomainName, region=None, key=None, keyid=None, profile=None*)

Given a domain name, delete it.

Returns {deleted: true} if the domain was deleted and returns {deleted: false} if the domain was not deleted.

CLI Example:

```
salt myminion boto_elasticsearch_domain.delete mydomain
```

`salt.modules.boto_elasticsearch_domain.`**`describe`**(*DomainName, region=None, key=None, keyid=None, profile=None*)

Given a domain name describe its properties.

Returns a dictionary of interesting properties.

CLI Example:

```
salt myminion boto_elasticsearch_domain.describe mydomain
```

`salt.modules.boto_elasticsearch_domain.`**`exists`**(*DomainName, region=None, key=None, keyid=None, profile=None*)

Given a domain name, check to see if the given domain exists.

Returns True if the given domain exists and returns False if the given function does not exist.

CLI Example:

```
salt myminion boto_elasticsearch_domain.exists mydomain
```

`salt.modules.boto_elasticsearch_domain.`**`list_tags`**(*DomainName=None, ARN=None, region=None, key=None, keyid=None, profile=None*)

List tags of a trail

Returns

- {...}
- {...}

Return type tags

CLI Example:

```
salt myminion boto_cloudtrail.list_tags my_trail
```

salt.modules.boto_elasticsearch_domain.**remove_tags**(*TagKeys, DomainName=None, ARN=None, region=None, key=None, keyid=None, profile=None*)

Remove tags from a trail

Returns {tagged: true} if the trail was tagged and returns {tagged: False} if the trail was not tagged.

CLI Example:

```
salt myminion boto_cloudtrail.remove_tags my_trail tag_a=tag_value tag_b=tag_value
```

salt.modules.boto_elasticsearch_domain.**status**(*DomainName, region=None, key=None, keyid=None, profile=None*)

Given a domain name describe its status.

Returns a dictionary of interesting properties.

CLI Example:

```
salt myminion boto_elasticsearch_domain.status mydomain
```

salt.modules.boto_elasticsearch_domain.**update**(*DomainName, ElasticsearchClusterConfig=None, EBSOptions=None, AccessPolicies=None, SnapshotOptions=None, AdvancedOptions=None, region=None, key=None, keyid=None, profile=None*)

Update the named domain to the configuration.

Returns {updated: true} if the domain was updated and returns {updated: False} if the domain was not updated.

CLI Example:

```
salt myminion boto_elasticsearch_domain.update mydomain \
      {'InstanceType': 't2.micro.elasticsearch', 'InstanceCount': 1, \
      'DedicatedMasterEnabled': false, 'ZoneAwarenessEnabled': false} \
      {'EBSEnabled': true, 'VolumeType': 'gp2', 'VolumeSize': 10, \
      'Iops': 0} \
      {"Version": "2012-10-17", "Statement": [{"Effect": "Allow", "Principal": {
↪"AWS": "*"}, "Action": "es:*", \
       "Resource": "arn:aws:es:us-east-1:111111111111:domain/mydomain/*", \
       "Condition": {"IpAddress": {"aws:SourceIp": ["127.0.0.1"]}}}]} \
      {"AutomatedSnapshotStartHour": 0} \
      {"rest.action.multi.allow_explicit_index": "true"}
```

21.9.50 salt.modules.boto_elb

Connection module for Amazon ELB

New in version 2014.7.0.

configuration This module accepts explicit elb credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More Information available at:

```
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-
↪ec2.html
```

If IAM roles are not used you need to specify them either in a pillar or in the minion's config file:

```
elb.keyid: GKTADJGHEIQSXMKKRBJ08H
elb.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

A region may also be specified in the configuration:

```
elb.region: us-east-1
```

If a region is not specified, the default is us-east-1.

It's also possible to specify key, keyid and region via a profile, either as a passed in dict, or as a string to pull from pillars or minion config:

```
myprofile:
    keyid: GKTADJGHEIQSXMKKRBJ08H
    key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
    region: us-east-1
```

depends boto >= 2.33.0

`salt.modules.boto_elb.`**`apply_security_groups`**(*name, security_groups, region=None, key=None, keyid=None, profile=None*)

Apply security groups to ELB.

CLI example:

```
salt myminion boto_elb.apply_security_groups myelb '["mysecgroup1"]'
```

`salt.modules.boto_elb.`**`attach_subnets`**(*name, subnets, region=None, key=None, keyid=None, profile=None*)

Attach ELB to subnets.

CLI example:

```
salt myminion boto_elb.attach_subnets myelb '["mysubnet"]'
```

`salt.modules.boto_elb.`**`create`**(*name, availability_zones, listeners, subnets=None, security_groups=None, scheme='internet-facing', region=None, key=None, keyid=None, profile=None*)

Create an ELB

CLI example to create an ELB:

```
salt myminion boto_elb.create myelb '["us-east-1a", "us-east-1e"]' '{"elb_port":␣
↪443, "elb_protocol": "HTTPS", ...}' region=us-east-1
```

salt.modules.boto_elb.**create_listeners**(*name, listeners, region=None, key=None, keyid=None, profile=None*)

Create listeners on an ELB.

CLI example:

```
salt myminion boto_elb.create_listeners myelb '[["HTTPS", "HTTP", 443, 80,
↪"arn:aws:iam::11  11111:server-certificate/mycert"]]'
```

salt.modules.boto_elb.**create_policy**(*name, policy_name, policy_type, policy, region=None, key=None, keyid=None, profile=None*)

Create an ELB policy.

New in version 2016.3.0.

CLI example:

```
salt myminion boto_elb.create_policy myelb mypolicy LBCookieStickinessPolicyType '
↪{"CookieExpirationPeriod": 3600}'
```

salt.modules.boto_elb.**delete**(*name, region=None, key=None, keyid=None, profile=None*)

Delete an ELB.

CLI example to delete an ELB:

```
salt myminion boto_elb.delete myelb region=us-east-1
```

salt.modules.boto_elb.**delete_listeners**(*name, ports, region=None, key=None, keyid=None, profile=None*)

Delete listeners on an ELB.

CLI example:

```
salt myminion boto_elb.delete_listeners myelb '[80,443]'
```

salt.modules.boto_elb.**delete_policy**(*name, policy_name, region=None, key=None, keyid=None, profile=None*)

Delete an ELB policy.

New in version 2016.3.0.

CLI example:

```
salt myminion boto_elb.delete_policy myelb mypolicy
```

salt.modules.boto_elb.**delete_tags**(*name, tags, region=None, key=None, keyid=None, profile=None*)

Add the tags on an ELB

name name of the ELB

tags list of tags to remove

CLI Example:

```
salt myminion boto_elb.delete_tags my-elb-name ['TagToRemove1', 'TagToRemove2']
```

salt.modules.boto_elb.**deregister_instances**(*name, instances, region=None, key=None, keyid=None, profile=None*)

Deregister instances with an ELB. Instances is either a string instance id or a list of string instance id's.

Returns:

- `True`: instance(s) deregistered successfully
- `False`: instance(s) failed to be deregistered
- `None`: instance(s) not valid or not registered, no action taken

CLI example:

```
salt myminion boto_elb.deregister_instances myelb instance_id
salt myminion boto_elb.deregister_instances myelb "[instance_id, instance_id]"
```

`salt.modules.boto_elb.`**`detach_subnets`**(*name, subnets, region=None, key=None, keyid=None, profile=None*)

Detach ELB from subnets.

CLI example:

```
salt myminion boto_elb.detach_subnets myelb '["mysubnet"]'
```

`salt.modules.boto_elb.`**`disable_availability_zones`**(*name, availability_zones, region=None, key=None, keyid=None, profile=None*)

Disable availability zones for ELB.

CLI example:

```
salt myminion boto_elb.disable_availability_zones myelb '["us-east-1a"]'
```

`salt.modules.boto_elb.`**`enable_availability_zones`**(*name, availability_zones, region=None, key=None, keyid=None, profile=None*)

Enable availability zones for ELB.

CLI example:

```
salt myminion boto_elb.enable_availability_zones myelb '["us-east-1a"]'
```

`salt.modules.boto_elb.`**`exists`**(*name, region=None, key=None, keyid=None, profile=None*)

Check to see if an ELB exists.

CLI example:

```
salt myminion boto_elb.exists myelb region=us-east-1
```

`salt.modules.boto_elb.`**`get_all_elbs`**(*region=None, key=None, keyid=None, profile=None*)

Return all load balancers associated with an account

CLI example:

```
salt myminion boto_elb.get_all_elbs region=us-east-1
```

`salt.modules.boto_elb.`**`get_attributes`**(*name, region=None, key=None, keyid=None, profile=None*)

Check to see if attributes are set on an ELB.

CLI example:

```
salt myminion boto_elb.get_attributes myelb
```

`salt.modules.boto_elb.`**`get_elb_config`**(*name, region=None, key=None, keyid=None, profile=None*)

Get an ELB configuration.

CLI example:

```
salt myminion boto_elb.exists myelb region=us-east-1
```

`salt.modules.boto_elb.`**`get_health_check`**(*name, region=None, key=None, keyid=None, profile=None*)

Get the health check configured for this ELB.

CLI example:

```
salt myminion boto_elb.get_health_check myelb
```

`salt.modules.boto_elb.`**`get_instance_health`**(*name, region=None, key=None, keyid=None, profile=None, instances=None*)

Get a list of instances and their health state

CLI example:

```
salt myminion boto_elb.get_instance_health myelb
salt myminion boto_elb.get_instance_health myelb region=us-east-1 instances=
↪"[instance_id,instance_id]"
```

`salt.modules.boto_elb.`**`list_elbs`**(*region=None, key=None, keyid=None, profile=None*)

Return names of all load balancers associated with an account

CLI example:

```
salt myminion boto_elb.list_elbs region=us-east-1
```

`salt.modules.boto_elb.`**`listener_dict_to_tuple`**(*listener*)

Convert an ELB listener dict into a listener tuple used by certain parts of the AWS ELB API.

CLI example:

```
salt myminion boto_elb.listener_dict_to_tuple '{"elb_port":80,"instance_port":80,
↪"elb_protocol":"HTTP"}'
```

`salt.modules.boto_elb.`**`register_instances`**(*name, instances, region=None, key=None, keyid=None, profile=None*)

Register instances with an ELB. Instances is either a string instance id or a list of string instance id's.

Returns:

- `True`: instance(s) registered successfully
- `False`: instance(s) failed to be registered

CLI example:

```
salt myminion boto_elb.register_instances myelb instance_id
salt myminion boto_elb.register_instances myelb "[instance_id,instance_id]"
```

`salt.modules.boto_elb.`**`set_attributes`**(*name, attributes, region=None, key=None, keyid=None, profile=None*)

Set attributes on an ELB.

name (string) Name of the ELB instance to set attributes for

attributes A dict of attributes to set.

Valid attributes are:

access_log (dict)

enabled (bool) Enable storage of access logs.

s3_bucket_name (string) The name of the S3 bucket to place logs.

s3_bucket_prefix (string) Prefix for the log file name.

emit_interval (int) Interval for storing logs in S3 in minutes. Valid values are 5 and 60.

connection_draining (dict)

enabled (bool) Enable connection draining.

timeout (int) Maximum allowed time in seconds for sending existing connections to an instance that is deregistering or unhealthy. Default is 300.

cross_zone_load_balancing (dict)

enabled (bool) Enable cross-zone load balancing.

CLI example to set attributes on an ELB:

```
salt myminion boto_elb.set_attributes myelb '{"access_log": {"enabled": "true",
↪"s3_bucket_name": "mybucket", "s3_bucket_prefix": "mylogs/", "emit_interval": "5
↪"}}' region=us-east-1
```

`salt.modules.boto_elb.`**`set_backend_policy`**(*name, port, policies=None, region=None, key=None, keyid=None, profile=None*)

Set the policies of an ELB backend server.

CLI example:

salt myminion boto_elb.set_backend_policy myelb 443 ”[policy1,policy2]”

`salt.modules.boto_elb.`**`set_health_check`**(*name, health_check, region=None, key=None, keyid=None, profile=None*)

Set attributes on an ELB.

CLI example to set attributes on an ELB:

```
salt myminion boto_elb.set_health_check myelb '{"target": "HTTP:80/"}'
```

`salt.modules.boto_elb.`**`set_instances`**(*name, instances, test=False, region=None, key=None, keyid=None, profile=None*)

Set the instances assigned to an ELB to exactly the list given

CLI example:

```
salt myminion boto_elb.set_instances myelb region=us-east-1 instances="[instance_
↪id,instance_id]"
```

`salt.modules.boto_elb.`**`set_listener_policy`**(*name, port, policies=None, region=None, key=None, keyid=None, profile=None*)

Set the policies of an ELB listener.

New in version 2016.3.0.

CLI example:

```
salt myminion boto_elb.set_listener_policy myelb 443 "[policy1,policy2]"
```

`salt.modules.boto_elb.`**`set_tags`**(*name, tags, region=None, key=None, keyid=None, profile=None*)

Add the tags on an ELB

New in version 2016.3.0.

name name of the ELB

tags dict of name/value pair tags

CLI Example:

```
salt myminion boto_elb.set_tags my-elb-name "{'Tag1': 'Value', 'Tag2': 'Another␣
↪Value'}"
```

21.9.51 salt.modules.boto_elbv2 module

Connection module for Amazon ALB

New in version 2017.7.0.

configuration This module accepts explicit elb credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More Information available at:

```
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-
↪ec2.html
```

If IAM roles are not used you need to specify them either in a pillar or in the minion's config file:

```
elbv2.keyid: GKTADJGHEIQSXMKKRBJ08H
elbv2.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
elbv2.region: us-west-2
```

If a region is not specified, the default is us-east-1.

It's also possible to specify key, keyid and region via a profile, either as a passed in dict, or as a string to pull from pillars or minion config:

```
myprofile:
    keyid: GKTADJGHEIQSXMKKRBJ08H
    key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
    region: us-east-1
```

depends boto3

`salt.modules.boto_elbv2.`**`create_target_group`**`(`*name, protocol, port, vpc_id, region=None, key=None, keyid=None, profile=None, health_check_protocol='HTTP', health_check_port='traffic-port', health_check_path='/', health_check_interval_seconds=30, health_check_timeout_seconds=5, healthy_threshold_count=5, unhealthy_threshold_count=2*`)`

Create target group if not present.

name (string) - The name of the target group.

protocol (string) - The protocol to use for routing traffic to the targets

port (int) - The port on which the targets receive traffic. This port is used unless you specify a port override when registering the traffic.

vpc_id (string) - The identifier of the virtual private cloud (VPC).

health_check_protocol (string) - The protocol the load balancer uses when performing health check on targets. The default is the HTTP protocol.

health_check_port (string) - The port the load balancer uses when performing health checks on targets. The default is 'traffic-port', which indicates the port on which each target receives traffic from the load balancer.

health_check_path (string) - The ping path that is the destination on the targets for health checks. The default is /.

health_check_interval_seconds (integer) - The approximate amount of time, in seconds, between health checks of an individual target. The default is 30 seconds.

health_check_timeout_seconds (integer) - The amount of time, in seconds, during which no response from a target means a failed health check. The default is 5 seconds.

healthy_threshold_count (integer) - The number of consecutive health checks successes required before considering an unhealthy target healthy. The default is 5.

unhealthy_threshold_count (integer) - The number of consecutive health check failures required before considering a target unhealthy. The default is 2.

returns (bool) - True on success, False on failure.

CLI example: .. code-block:: bash

salt myminion boto_elbv2.create_target_group learn1give1 protocol=HTTP port=54006 vpc_id=vpc-deadbeef

salt.modules.boto_elbv2.**delete_target_group**(*name*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Delete target group.

name (string) - Target Group Name or Amazon Resource Name (ARN).

returns (bool) - True on success, False on failure.

CLI example:

```
salt myminion boto_elbv2.delete_target_group arn:aws:elasticloadbalancing:us-west-
↪2:644138682826:targetgroup/learn1give1-api/414788a16b5cf163
```

salt.modules.boto_elbv2.**deregister_targets**(*name*, *targets*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Deregister targets to a target froup of an ALB. `targets` is either a instance id string or a list of instance id's.

Returns:

- `True`: instance(s) deregistered successfully
- `False`: instance(s) failed to be deregistered

CLI example:

```
salt myminion boto_elbv2.deregister_targets myelb instance_id
salt myminion boto_elbv2.deregister_targets myelb "[instance_id,instance_id]"
```

salt.modules.boto_elbv2.**describe_target_health**(*name*, *targets=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Get the curret health check status for targets in a target group.

CLI example:

```
salt myminion boto_elbv2.describe_target_health arn:aws:elasticloadbalancing:us-
↪west-2:644138682826:targetgroup/learn1give1-api/414788a16b5cf163 targets=["i-
↪isdf23ifjf"]
```

`salt.modules.boto_elbv2.`**`register_targets`**(*name, targets, region=None, key=None, keyid=None, profile=None*)

Register targets to a target froup of an ALB. `targets` is either a instance id string or a list of instance id's.

Returns:

- `True`: instance(s) registered successfully
- `False`: instance(s) failed to be registered

CLI example:

```
salt myminion boto_elbv2.register_targets myelb instance_id
salt myminion boto_elbv2.register_targets myelb "[instance_id,instance_id]"
```

`salt.modules.boto_elbv2.`**`target_group_exists`**(*name, region=None, key=None, keyid=None, profile=None*)

Check to see if an target group exists.

CLI example:

```
salt myminion boto_elbv2.target_group_exists arn:aws:elasticloadbalancing:us-west-
↪2:644138682826:targetgroup/learn1give1-api/414788a16b5cf163
```

21.9.52 salt.modules.boto_iam

Connection module for Amazon IAM

New in version 2014.7.0.

configuration This module accepts explicit iam credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More Information available at:

```
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-
↪ec2.html
```

If IAM roles are not used you need to specify them either in a pillar or in the minion's config file:

```
iam.keyid: GKTADJGHEIQSXMKKRBJ08H
iam.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
iam.region: us-east-1
```

It's also possible to specify key, keyid and region via a profile, either as a passed in dict, or as a string to pull from pillars or minion config:

```
myprofile:
    keyid: GKTADJGHEIQSXMKKRBJ08H
    key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
    region: us-east-1
```

depends boto

`salt.modules.boto_iam.`**`add_user_to_group`**`(`*user_name, group_name, region=None, key=None, keyid=None, profile=None*`)`

Add user to group.

New in version 2015.8.0.

CLI Example:

```
salt myminion boto_iam.add_user_to_group myuser mygroup
```

`salt.modules.boto_iam.`**`associate_profile_to_role`**`(`*profile_name, role_name, region=None, key=None, keyid=None, profile=None*`)`

Associate an instance profile with an IAM role.

CLI Example:

```
salt myminion boto_iam.associate_profile_to_role myirole myiprofile
```

`salt.modules.boto_iam.`**`attach_group_policy`**`(`*policy_name, group_name, region=None, key=None, keyid=None, profile=None*`)`

Attach a managed policy to a group.

CLI Example:

```
salt myminion boto_iam.attach_group_policy mypolicy mygroup
```

`salt.modules.boto_iam.`**`attach_role_policy`**`(`*policy_name, role_name, region=None, key=None, keyid=None, profile=None*`)`

Attach a managed policy to a role.

CLI Example:

```
salt myminion boto_iam.attach_role_policy mypolicy myrole
```

`salt.modules.boto_iam.`**`attach_user_policy`**`(`*policy_name, user_name, region=None, key=None, keyid=None, profile=None*`)`

Attach a managed policy to a user.

CLI Example:

```
salt myminion boto_iam.attach_user_policy mypolicy myuser
```

`salt.modules.boto_iam.`**`build_policy`**`(`*region=None, key=None, keyid=None, profile=None*`)`

Build a default assume role policy.

New in version 2015.8.0.

CLI Example:

```
salt myminion boto_iam.build_policy
```

`salt.modules.boto_iam.`**`create_access_key`**`(`*user_name, region=None, key=None, keyid=None, profile=None*`)`

Create access key id for a user.

New in version 2015.8.0.

CLI Example:

```
salt myminion boto_iam.create_access_key myuser
```

salt.modules.boto_iam.**create_group**(*group_name, path=None, region=None, key=None, keyid=None, profile=None*)

Create a group.

New in version 2015.8.0.

CLI Example:

```
salt myminion boto_iam.create_group group
```

salt.modules.boto_iam.**create_instance_profile**(*name, region=None, key=None, keyid=None, profile=None*)

Create an instance profile.

CLI Example:

```
salt myminion boto_iam.create_instance_profile myiprofile
```

salt.modules.boto_iam.**create_login_profile**(*user_name, password, region=None, key=None, keyid=None, profile=None*)

Creates a login profile for the specified user, give the user the ability to access AWS services and the AWS Management Console.

New in version 2015.8.0.

CLI Example:

```
salt myminion boto_iam.create_login_profile user_name password
```

salt.modules.boto_iam.**create_policy**(*policy_name, policy_document, path=None, description=None, region=None, key=None, keyid=None, profile=None*)

Create a policy.

CLI Example:

```
salt myminios boto_iam.create_policy mypolicy '{"Version": "2012-10-17",
↪"Statement": [{ "Effect": "Allow", "Action": ["s3:Get*", "s3:List*"], "Resource
↪": ["arn:aws:s3:::my-bucket/shared/*"]},]}'
```

salt.modules.boto_iam.**create_policy_version**(*policy_name, policy_document, set_as_default=None, region=None, key=None, keyid=None, profile=None*)

Create a policy version.

CLI Example:

```
salt myminios boto_iam.create_policy_version mypolicy '{"Version": "2012-10-17",
↪"Statement": [{ "Effect": "Allow", "Action": ["s3:Get*", "s3:List*"], "Resource
↪": ["arn:aws:s3:::my-bucket/shared/*"]},]}'
```

salt.modules.boto_iam.**create_role**(*name, policy_document=None, path=None, region=None, key=None, keyid=None, profile=None*)

Create an instance role.

CLI Example:

```
salt myminion boto_iam.create_role myrole
```

`salt.modules.boto_iam.`**`create_role_policy`**(*role_name, policy_name, policy, region=None, key=None, keyid=None, profile=None*)

Create or modify a role policy.

CLI Example:

```
salt myminion boto_iam.create_role_policy myirole mypolicy '{"MyPolicy":
↪"Statement": [{"Action": ["sqs:*"], "Effect": "Allow", "Resource": [
↪"arn:aws:sqs:*:*:*"], "Sid": "MyPolicySqs1"}]}'
```

`salt.modules.boto_iam.`**`create_saml_provider`**(*name, saml_metadata_document, region=None, key=None, keyid=None, profile=None*)

Create SAML provider

CLI Example:

```
salt myminion boto_iam.create_saml_provider my_saml_provider_name saml_metadata_
↪document
```

`salt.modules.boto_iam.`**`create_user`**(*user_name, path=None, region=None, key=None, keyid=None, profile=None*)

Create a user.

New in version 2015.8.0.

CLI Example:

```
salt myminion boto_iam.create_user myuser
```

`salt.modules.boto_iam.`**`deactivate_mfa_device`**(*user_name, serial, region=None, key=None, keyid=None, profile=None*)

Deactivates the specified MFA device and removes it from association with the user.

New in version 2016.3.0.

CLI Example:

```
salt myminion boto_iam.deactivate_mfa_device user_name serial_num
```

`salt.modules.boto_iam.`**`delete_access_key`**(*access_key_id, user_name=None, region=None, key=None, keyid=None, profile=None*)

Delete access key id from a user.

New in version 2015.8.0.

CLI Example:

```
salt myminion boto_iam.delete_access_key myuser
```

`salt.modules.boto_iam.`**`delete_group`**(*group_name, region=None, key=None, keyid=None, profile=None*)

Delete a group policy.

CLI Example:

```
.. code-block:: bash
```

salt myminion boto_iam.delete_group mygroup `salt.modules.boto_iam.`**`delete_group_policy`**(*group_name*, *policy_name*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Delete a group policy.

CLI Example:

```
.. code-block:: bash
```

salt myminion boto_iam.delete_group_policy mygroup mypolicy

`salt.modules.boto_iam.`**`delete_instance_profile`**(*name*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Delete an instance profile.

CLI Example:

```
salt myminion boto_iam.delete_instance_profile myiprofile
```

`salt.modules.boto_iam.`**`delete_login_profile`**(*user_name*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Deletes a login profile for the specified user.

New in version 2016.3.0.

CLI Example:

```
salt myminion boto_iam.delete_login_profile user_name
```

`salt.modules.boto_iam.`**`delete_policy`**(*policy_name*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Delete a policy.

CLI Example:

```
salt myminion boto_iam.delete_policy mypolicy
```

`salt.modules.boto_iam.`**`delete_policy_version`**(*policy_name*, *version_id*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Delete a policy version.

CLI Example:

```
salt myminion boto_iam.delete_policy_version mypolicy v1
```

`salt.modules.boto_iam.`**`delete_role`**(*name*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Delete an IAM role.

CLI Example:

```
salt myminion boto_iam.delete_role myirole
```

`salt.modules.boto_iam.`**`delete_role_policy`**(*role_name*, *policy_name*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Delete a role policy.

CLI Example:

```
salt myminion boto_iam.delete_role_policy myirole mypolicy
```

`salt.modules.boto_iam.`**`delete_saml_provider`**`(`*name, region=None, key=None, keyid=None, profile=None*`)`

Delete SAML provider

CLI Example:

```
salt myminion boto_iam.delete_saml_provider my_saml_provider_name
```

`salt.modules.boto_iam.`**`delete_server_cert`**`(`*cert_name, region=None, key=None, keyid=None, profile=None*`)`

Deletes a certificate from Amazon.

New in version 2015.8.0.

CLI Example:

```
salt myminion boto_iam.delete_server_cert mycert_name
```

`salt.modules.boto_iam.`**`delete_user`**`(`*user_name, region=None, key=None, keyid=None, profile=None*`)`

Delete a user.

New in version 2015.8.0.

CLI Example:

```
salt myminion boto_iam.delete_user myuser
```

`salt.modules.boto_iam.`**`delete_user_policy`**`(`*user_name, policy_name, region=None, key=None, keyid=None, profile=None*`)`

Delete a user policy.

CLI Example:

```
salt myminion boto_iam.delete_user_policy myuser mypolicy
```

`salt.modules.boto_iam.`**`delete_virtual_mfa_device`**`(`*serial, region=None, key=None, keyid=None, profile=None*`)`

Deletes the specified virtual MFA device.

CLI Example:

```
salt myminion boto_iam.delete_virtual_mfa_device serial_num
```

`salt.modules.boto_iam.`**`describe_role`**`(`*name, region=None, key=None, keyid=None, profile=None*`)`

Get information for a role.

CLI Example:

```
salt myminion boto_iam.describe_role myirole
```

`salt.modules.boto_iam.`**`detach_group_policy`**`(`*policy_name, group_name, region=None, key=None, keyid=None, profile=None*`)`

Detach a managed policy to a group.

CLI Example:

```
salt myminion boto_iam.detach_group_policy mypolicy mygroup
```

`salt.modules.boto_iam.`**`detach_role_policy`**(*policy_name*, *role_name*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Detach a managed policy to a role.

CLI Example:

```
salt myminion boto_iam.detach_role_policy mypolicy myrole
```

`salt.modules.boto_iam.`**`detach_user_policy`**(*policy_name*, *user_name*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Detach a managed policy to a user.

CLI Example:

```
salt myminion boto_iam.detach_user_policy mypolicy myuser
```

`salt.modules.boto_iam.`**`disassociate_profile_from_role`**(*profile_name*, *role_name*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Disassociate an instance profile from an IAM role.

CLI Example:

```
salt myminion boto_iam.disassociate_profile_from_role myirole myiprofile
```

`salt.modules.boto_iam.`**`export_roles`**(*path_prefix='/'*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Get all IAM role details. Produces results that can be used to create an sls file.

CLI Example:

> salt-call boto_iam.export_roles –out=txt | sed "s/local: //" > iam_roles.sls

`salt.modules.boto_iam.`**`export_users`**(*path_prefix='/'*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Get all IAM user details. Produces results that can be used to create an sls file.

New in version 2016.3.0.

CLI Example:

> salt-call boto_iam.export_users –out=txt | sed "s/local: //" > iam_users.sls

`salt.modules.boto_iam.`**`get_account_id`**(*region=None*, *key=None*, *keyid=None*, *profile=None*)

Get a the AWS account id associated with the used credentials.

CLI Example:

```
salt myminion boto_iam.get_account_id
```

`salt.modules.boto_iam.`**`get_account_policy`**(*region=None*, *key=None*, *keyid=None*, *profile=None*)

Get account policy for the AWS account.

New in version 2015.8.0.

CLI Example:

```
salt myminion boto_iam.get_account_policy
```

`salt.modules.boto_iam.`**`get_all_access_keys`**`(`*user_name, marker=None, max_items=None, region=None, key=None, keyid=None, profile=None*`)`

Get all access keys from a user.

New in version 2015.8.0.

CLI Example:

```
salt myminion boto_iam.get_all_access_keys myuser
```

`salt.modules.boto_iam.`**`get_all_group_policies`**`(`*group_name, region=None, key=None, keyid=None, profile=None*`)`

Get a list of policy names from a group.

CLI Example:

```
salt myminion boto_iam.get_all_group_policies mygroup
```

`salt.modules.boto_iam.`**`get_all_groups`**`(`*path_prefix='/', region=None, key=None, keyid=None, profile=None*`)`

Get and return all IAM group details, starting at the optional path.

New in version 2016.3.0.

CLI Example:

> salt-call boto_iam.get_all_groups

`salt.modules.boto_iam.`**`get_all_instance_profiles`**`(`*path_prefix='/', region=None, key=None, keyid=None, profile=None*`)`

Get and return all IAM instance profiles, starting at the optional path.

New in version 2016.11.0.

CLI Example:

> salt-call boto_iam.get_all_instance_profiles

`salt.modules.boto_iam.`**`get_all_mfa_devices`**`(`*user_name, region=None, key=None, keyid=None, profile=None*`)`

Get all MFA devices associated with an IAM user.

New in version 2016.3.0.

CLI Example:

```
salt myminion boto_iam.get_all_mfa_devices user_name
```

`salt.modules.boto_iam.`**`get_all_roles`**`(`*path_prefix=None, region=None, key=None, keyid=None, profile=None*`)`

Get and return all IAM role details, starting at the optional path.

New in version 2016.3.0.

CLI Example:

> salt-call boto_iam.get_all_roles

`salt.modules.boto_iam.`**`get_all_user_policies`**`(`*user_name, marker=None, max_items=None, region=None, key=None, keyid=None, profile=None*`)`

Get all user policies.

New in version 2015.8.0.

CLI Example:

```
salt myminion boto_iam.get_all_user_policies myuser
```

`salt.modules.boto_iam.`**`get_all_users`**`(`*path_prefix='/', region=None, key=None, keyid=None, profile=None*`)`

Get and return all IAM user details, starting at the optional path.

New in version 2016.3.0.

CLI Example:

salt-call boto_iam.get_all_users

`salt.modules.boto_iam.`**`get_group`**`(`*group_name, region=None, key=None, keyid=None, profile=None*`)`

Get group information.

New in version 2015.8.0.

CLI Example:

```
salt myminion boto_iam.get_group mygroup
```

`salt.modules.boto_iam.`**`get_group_members`**`(`*group_name, region=None, key=None, keyid=None, profile=None*`)`

Get group information.

New in version 2016.3.0.

CLI Example:

```
salt myminion boto_iam.get_group mygroup
```

`salt.modules.boto_iam.`**`get_group_policy`**`(`*group_name, policy_name, region=None, key=None, keyid=None, profile=None*`)`

Retrieves the specified policy document for the specified group.

New in version 2015.8.0.

CLI Example:

```
salt myminion boto_iam.get_group_policy mygroup policyname
```

`salt.modules.boto_iam.`**`get_policy`**`(`*policy_name, region=None, key=None, keyid=None, profile=None*`)`

Check to see if policy exists.

CLI Example:

```
salt myminion boto_iam.instance_profile_exists myiprofile
```

`salt.modules.boto_iam.`**`get_policy_version`**`(`*policy_name, version_id, region=None, key=None, keyid=None, profile=None*`)`

Check to see if policy exists.

CLI Example:

```
salt myminion boto_iam.instance_profile_exists myiprofile
```

`salt.modules.boto_iam.`**`get_role_policy`**`(`*role_name, policy_name, region=None, key=None, keyid=None, profile=None*`)`

Get a role policy.

CLI Example:

```
salt myminion boto_iam.get_role_policy myirole mypolicy
```

salt.modules.boto_iam.**get_saml_provider**(*name*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Get SAML provider document.

CLI Example:

```
salt myminion boto_iam.get_saml_provider arn
```

salt.modules.boto_iam.**get_saml_provider_arn**(*name*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Get SAML provider

CLI Example:

```
salt myminion boto_iam.get_saml_provider_arn my_saml_provider_name
```

salt.modules.boto_iam.**get_server_certificate**(*cert_name*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Returns certificate information from Amazon

New in version 2015.8.0.

CLI Example:

```
salt myminion boto_iam.get_server_certificate mycert_name
```

salt.modules.boto_iam.**get_user**(*user_name=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Get user information.

New in version 2015.8.0.

CLI Example:

```
salt myminion boto_iam.get_user myuser
```

salt.modules.boto_iam.**get_user_policy**(*user_name*, *policy_name*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Retrieves the specified policy document for the specified user.

New in version 2015.8.0.

CLI Example:

```
salt myminion boto_iam.get_user_policy myuser mypolicyname
```

salt.modules.boto_iam.**instance_profile_exists**(*name*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Check to see if an instance profile exists.

CLI Example:

```
salt myminion boto_iam.instance_profile_exists myiprofile
```

`salt.modules.boto_iam.`**`list_attached_group_policies`**`(`*group_name, path_prefix=None, entity_filter=None, region=None, key=None, keyid=None, profile=None*`)`

List entities attached to the given group.

CLI Example:

```
salt myminion boto_iam.list_entities_for_policy mypolicy
```

`salt.modules.boto_iam.`**`list_attached_role_policies`**`(`*role_name, path_prefix=None, entity_filter=None, region=None, key=None, keyid=None, profile=None*`)`

List entities attached to the given role.

CLI Example:

```
salt myminion boto_iam.list_entities_for_policy mypolicy
```

`salt.modules.boto_iam.`**`list_attached_user_policies`**`(`*user_name, path_prefix=None, entity_filter=None, region=None, key=None, keyid=None, profile=None*`)`

List entities attached to the given user.

CLI Example:

```
salt myminion boto_iam.list_entities_for_policy mypolicy
```

`salt.modules.boto_iam.`**`list_entities_for_policy`**`(`*policy_name, path_prefix=None, entity_filter=None, region=None, key=None, keyid=None, profile=None*`)`

List entities that a policy is attached to.

CLI Example:

```
salt myminion boto_iam.list_entities_for_policy mypolicy
```

`salt.modules.boto_iam.`**`list_instance_profiles`**`(`*path_prefix='/', region=None, key=None, keyid=None, profile=None*`)`

List all IAM instance profiles, starting at the optional path.

New in version 2016.11.0.

CLI Example:

> salt-call boto_iam.list_instance_profiles

`salt.modules.boto_iam.`**`list_policies`**`(`*region=None, key=None, keyid=None, profile=None*`)`

List policies.

CLI Example:

```
salt myminion boto_iam.list_policies
```

`salt.modules.boto_iam.`**`list_policy_versions`**`(`*policy_name, region=None, key=None, keyid=None, profile=None*`)`

List versions of a policy.

CLI Example:

```
salt myminion boto_iam.list_policy_versions mypolicy
```

`salt.modules.boto_iam.`**`list_role_policies`**(*role_name, region=None, key=None, keyid=None, profile=None*)

Get a list of policy names from a role.

CLI Example:

```
salt myminion boto_iam.list_role_policies myirole
```

`salt.modules.boto_iam.`**`list_saml_providers`**(*region=None, key=None, keyid=None, profile=None*)

List SAML providers.

CLI Example:

```
salt myminion boto_iam.list_saml_providers
```

`salt.modules.boto_iam.`**`policy_exists`**(*policy_name, region=None, key=None, keyid=None, profile=None*)

Check to see if policy exists.

CLI Example:

```
salt myminion boto_iam.instance_profile_exists myiprofile
```

`salt.modules.boto_iam.`**`policy_version_exists`**(*policy_name, version_id, region=None, key=None, keyid=None, profile=None*)

Check to see if policy exists.

CLI Example:

```
salt myminion boto_iam.instance_profile_exists myiprofile
```

`salt.modules.boto_iam.`**`profile_associated`**(*role_name, profile_name, region, key, keyid, profile*)

Check to see if an instance profile is associated with an IAM role.

CLI Example:

```
salt myminion boto_iam.profile_associated myirole myiprofile
```

`salt.modules.boto_iam.`**`put_group_policy`**(*group_name, policy_name, policy_json, region=None, key=None, keyid=None, profile=None*)

Adds or updates the specified policy document for the specified group.

New in version 2015.8.0.

CLI Example:

```
salt myminion boto_iam.put_group_policy mygroup policyname policyrules
```

`salt.modules.boto_iam.`**`put_user_policy`**(*user_name, policy_name, policy_json, region=None, key=None, keyid=None, profile=None*)

Adds or updates the specified policy document for the specified user.

New in version 2015.8.0.

CLI Example:

```
salt myminion boto_iam.put_user_policy myuser policyname policyrules
```

`salt.modules.boto_iam.`**`remove_user_from_group`**`(`*group_name*, *user_name*, *region=None*, *key=None*, *keyid=None*, *profile=None*`)`

Remove user from group.

New in version 2015.8.0.

CLI Example:

```
salt myminion boto_iam.remove_user_from_group mygroup myuser
```

`salt.modules.boto_iam.`**`role_exists`**`(`*name*, *region=None*, *key=None*, *keyid=None*, *profile=None*`)`

Check to see if an IAM role exists.

CLI Example:

```
salt myminion boto_iam.role_exists myirole
```

`salt.modules.boto_iam.`**`set_default_policy_version`**`(`*policy_name*, *version_id*, *region=None*, *key=None*, *keyid=None*, *profile=None*`)`

Set the default version of a policy.

CLI Example:

```
salt myminion boto_iam.set_default_policy_version mypolicy v1
```

`salt.modules.boto_iam.`**`update_account_password_policy`**`(`*allow_users_to_change_password=None*, *hard_expiry=None*, *max_password_age=None*, *minimum_password_length=None*, *password_reuse_prevention=None*, *require_lowercase_characters=None*, *require_numbers=None*, *require_symbols=None*, *require_uppercase_characters=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*`)`

Update the password policy for the AWS account.

New in version 2015.8.0.

CLI Example:

```
salt myminion boto_iam.update_account_password_policy True
```

`salt.modules.boto_iam.`**`update_assume_role_policy`**`(`*role_name*, *policy_document*, *region=None*, *key=None*, *keyid=None*, *profile=None*`)`

Update an assume role policy for a role.

New in version 2015.8.0.

CLI Example:

```
salt myminion boto_iam.update_assume_role_policy myrole '{"Statement":"..."}'
```

salt.modules.boto_iam.**update_saml_provider**(*name*, *saml_metadata_document*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Update SAML provider.

CLI Example:

```
salt myminion boto_iam.update_saml_provider my_saml_provider_name saml_metadata_
↪document
```

salt.modules.boto_iam.**upload_server_cert**(*cert_name*, *cert_body*, *private_key*, *cert_chain=None*, *path=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Upload a certificate to Amazon.

New in version 2015.8.0.

CLI Example:

```
salt myminion boto_iam.upload_server_cert mycert_name crt priv_key
```

Parameters

- **cert_name** – The name for the server certificate. Do not include the path in this value.
- **cert_body** – The contents of the public key certificate in PEM-encoded format.
- **private_key** – The contents of the private key in PEM-encoded format.
- **cert_chain** – The contents of the certificate chain. This is typically a concatenation of the PEM-encoded public key certificates of the chain.
- **path** – The path for the server certificate.
- **region** – The name of the region to connect to.
- **key** – The key to be used in order to connect
- **keyid** – The keyid to be used in order to connect
- **profile** – The profile that contains a dict of region, key, keyid

Returns True / False

salt.modules.boto_iam.**user_exists_in_group**(*user_name*, *group_name*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Check if user exists in group.

New in version 2015.8.0.

CLI Example:

```
salt myminion boto_iam.user_exists_in_group myuser mygroup
```

21.9.53 salt.modules.boto_iot module

Connection module for Amazon IoT

New in version 2016.3.0.

depends

- boto
- boto3

The dependencies listed above can be installed via package or pip.

configuration This module accepts explicit Lambda credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More Information available at:

```
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-
↪ec2.html
```

If IAM roles are not used you need to specify them either in a pillar or in the minion's config file:

```
iot.keyid: GKTADJGHEIQSXMKKRBJ08H
iot.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

A region may also be specified in the configuration:

```
iot.region: us-east-1
```

If a region is not specified, the default is us-east-1.

It's also possible to specify key, keyid and region via a profile, either as a passed in dict, or as a string to pull from pillars or minion config:

```
myprofile:
  keyid: GKTADJGHEIQSXMKKRBJ08H
  key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
  region: us-east-1
```

`salt.modules.boto_iot.`**`attach_principal_policy`**(*policyName, principal, region=None, key=None, keyid=None, profile=None*)

Attach the specified policy to the specified principal (certificate or other credential.)

Returns {attached: true} if the policy was attached {attached: False} if the policy was not attached.

CLI Example:

```
salt myminion boto_iot.attach_principal_policy mypolicy mycognitoID
```

`salt.modules.boto_iot.`**`create_policy`**(*policyName, policyDocument, region=None, key=None, keyid=None, profile=None*)

Given a valid config, create a policy.

Returns {created: true} if the policy was created and returns {created: False} if the policy was not created.

CLI Example:

```
salt myminion boto_iot.create_policy my_policy \
      '{"Version":"2015-12-12",\
      "Statement":[{"Effect":"Allow",\
                    "Action":["iot:Publish"],\
                    "Resource":["arn:::::topic/foo/bar"]}]}'
```

`salt.modules.boto_iot.`**`create_policy_version`**(*policyName, policyDocument, setAsDefault=False, region=None, key=None, keyid=None, profile=None*)

Given a valid config, create a new version of a policy.

Returns {created: true} if the policy version was created and returns {created: False} if the policy version was not created.

CLI Example:

```
salt myminion boto_iot.create_policy_version my_policy \
       '{"Statement":[{"Effect":"Allow","Action":["iot:Publish"],"Resource":[
↪"arn::::topic/foo/bar"]}]}'
```

`salt.modules.boto_iot.`**`create_thing_type`**(*thingTypeName, thingTypeDescription, searchableAttributesList, region=None, key=None, keyid=None, profile=None*)

Given a valid config, create a thing type.

Returns {created: true} if the thing type was created and returns {created: False} if the thing type was not created.

New in version 2016.11.0.

CLI Example:

```
salt myminion boto_iot.create_thing_type mythingtype \
      thingtype_description_string '["searchable_attr_1", "searchable_attr_2"]'
```

`salt.modules.boto_iot.`**`create_topic_rule`**(*ruleName, sql, actions, description, ruleDisabled=False, region=None, key=None, keyid=None, profile=None*)

Given a valid config, create a topic rule.

Returns {created: true} if the rule was created and returns {created: False} if the rule was not created.

CLI Example:

```
salt myminion boto_iot.create_topic_rule my_rule "SELECT * FROM 'some/thing'" \
    '[{"lambda":{"functionArn":"arn:::::something"}},{"sns":{\
    "targetArn":"arn:::::something","roleArn":"arn:::::something"}}]'
```

`salt.modules.boto_iot.`**`delete_policy`**(*policyName, region=None, key=None, keyid=None, profile=None*)

Given a policy name, delete it.

Returns {deleted: true} if the policy was deleted and returns {deleted: false} if the policy was not deleted.

CLI Example:

```
salt myminion boto_iot.delete_policy mypolicy
```

`salt.modules.boto_iot.`**`delete_policy_version`**(*policyName, policyVersionId, region=None, key=None, keyid=None, profile=None*)

Given a policy name and version, delete it.

Returns {deleted: true} if the policy version was deleted and returns {deleted: false} if the policy version was not deleted.

CLI Example:

```
salt myminion boto_iot.delete_policy_version mypolicy version
```

`salt.modules.boto_iot.`**`delete_thing_type`**(*thingTypeName, region=None, key=None, keyid=None, profile=None*)

Given a thing type name, delete it.

Returns {deleted: true} if the thing type was deleted and returns {deleted: false} if the thing type was not deleted.

New in version 2016.11.0.

CLI Example:

```
salt myminion boto_iot.delete_thing_type mythingtype
```

`salt.modules.boto_iot.`**`delete_topic_rule`**`(`*ruleName*, *region=None*, *key=None*, *keyid=None*, *profile=None*`)`

Given a rule name, delete it.

Returns {deleted: true} if the rule was deleted and returns {deleted: false} if the rule was not deleted.

CLI Example:

```
salt myminion boto_iot.delete_rule myrule
```

`salt.modules.boto_iot.`**`deprecate_thing_type`**`(`*thingTypeName*, *undoDeprecate=False*, *region=None*, *key=None*, *keyid=None*, *profile=None*`)`

Given a thing type name, deprecate it when undoDeprecate is False and undeprecate it when undoDeprecate is True.

Returns {deprecated: true} if the thing type was deprecated and returns {deprecated: false} if the thing type was not deprecated.

New in version 2016.11.0.

CLI Example:

```
salt myminion boto_iot.deprecate_thing_type mythingtype
```

`salt.modules.boto_iot.`**`describe_policy`**`(`*policyName*, *region=None*, *key=None*, *keyid=None*, *profile=None*`)`

Given a policy name describe its properties.

Returns a dictionary of interesting properties.

CLI Example:

```
salt myminion boto_iot.describe_policy mypolicy
```

`salt.modules.boto_iot.`**`describe_policy_version`**`(`*policyName*, *policyVersionId*, *region=None*, *key=None*, *keyid=None*, *profile=None*`)`

Given a policy name and version describe its properties.

Returns a dictionary of interesting properties.

CLI Example:

```
salt myminion boto_iot.describe_policy_version mypolicy version
```

`salt.modules.boto_iot.`**`describe_thing_type`**`(`*thingTypeName*, *region=None*, *key=None*, *keyid=None*, *profile=None*`)`

Given a thing type name describe its properties.

Returns a dictionary of interesting properties.

New in version 2016.11.0.

CLI Example:

```
salt myminion boto_iot.describe_thing_type mythingtype
```

salt.modules.boto_iot.**describe_topic_rule**(*ruleName*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Given a topic rule name describe its properties.

Returns a dictionary of interesting properties.

CLI Example:

```
salt myminion boto_iot.describe_topic_rule myrule
```

salt.modules.boto_iot.**detach_principal_policy**(*policyName*, *principal*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Detach the specified policy from the specified principal (certificate or other credential.)

Returns {detached: true} if the policy was detached {detached: False} if the policy was not detached.

CLI Example:

```
salt myminion boto_iot.detach_principal_policy mypolicy mycognitoID
```

salt.modules.boto_iot.**list_policies**(*region=None*, *key=None*, *keyid=None*, *profile=None*)

List all policies

Returns list of policies

CLI Example:

```
salt myminion boto_iot.list_policies
```

Example Return:

```
policies:
  - {...}
  - {...}
```

salt.modules.boto_iot.**list_policy_versions**(*policyName*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

List the versions available for the given policy.

CLI Example:

```
salt myminion boto_iot.list_policy_versions mypolicy
```

Example Return:

```
policyVersions:
  - {...}
  - {...}
```

salt.modules.boto_iot.**list_principal_policies**(*principal*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

List the policies attached to the given principal.

CLI Example:

```
salt myminion boto_iot.list_principal_policies myprincipal
```

Example Return:

```
policies:
  - {...}
  - {...}
```

`salt.modules.boto_iot.`**`list_topic_rules`**(*topic=None, ruleDisabled=None, region=None, key=None, keyid=None, profile=None*)

List all rules (for a given topic, if specified)

Returns list of rules

CLI Example:

```
salt myminion boto_iot.list_topic_rules
```

Example Return:

```
rules:
  - {...}
  - {...}
```

`salt.modules.boto_iot.`**`policy_exists`**(*policyName, region=None, key=None, keyid=None, profile=None*)

Given a policy name, check to see if the given policy exists.

Returns True if the given policy exists and returns False if the given policy does not exist.

CLI Example:

```
salt myminion boto_iot.policy_exists mypolicy
```

`salt.modules.boto_iot.`**`policy_version_exists`**(*policyName, policyVersionId, region=None, key=None, keyid=None, profile=None*)

Given a policy name and version ID, check to see if the given policy version exists.

Returns True if the given policy version exists and returns False if the given policy version does not exist.

CLI Example:

```
salt myminion boto_iot.policy_version_exists mypolicy versionid
```

`salt.modules.boto_iot.`**`replace_topic_rule`**(*ruleName, sql, actions, description, ruleDisabled=False, region=None, key=None, keyid=None, profile=None*)

Given a valid config, replace a topic rule with the new values.

Returns {created: true} if the rule was created and returns {created: False} if the rule was not created.

CLI Example:

```
salt myminion boto_iot.replace_topic_rule my_rule 'SELECT * FROM some.thing' \
    '[{"lambda":{"functionArn":"arn:::::something"}},{"sns":{\
    "targetArn":"arn:::::something","roleArn":"arn:::::something"}}]'
```

`salt.modules.boto_iot.`**`set_default_policy_version`**(*policyName, policyVersionId, region=None, key=None, keyid=None, profile=None*)

Sets the specified version of the specified policy as the policy's default (operative) version. This action affects all certificates that the policy is attached to.

Returns {changed: true} if the policy version was set {changed: False} if the policy version was not set.

CLI Example:

```
salt myminion boto_iot.set_default_policy_version mypolicy versionid
```

`salt.modules.boto_iot.`**`thing_type_exists`**(*thingTypeName, region=None, key=None, keyid=None, profile=None*)

Given a thing type name, check to see if the given thing type exists

Returns True if the given thing type exists and returns False if the given thing type does not exist.

New in version 2016.11.0.

CLI Example:

```
salt myminion boto_iot.thing_type_exists mythingtype
```

`salt.modules.boto_iot.`**`topic_rule_exists`**(*ruleName, region=None, key=None, keyid=None, profile=None*)

Given a rule name, check to see if the given rule exists.

Returns True if the given rule exists and returns False if the given rule does not exist.

CLI Example:

```
salt myminion boto_iot.topic_rule_exists myrule
```

21.9.54 salt.modules.boto_kinesis module

Connection module for Amazon Kinesis

New in version 2017.7.0.

configuration This module accepts explicit Kinesis credentials but can also utilize IAM roles assigned to the instance trough Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More Information available at:

```
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-
↪ec2.html
```

If IAM roles are not used you need to specify them either in a pillar or in the minion's config file:

```
kinesis.keyid: GKTADJGHEIQSXMKKRBJ08H
kinesis.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

A region may also be specified in the configuration:

```
kinesis.region: us-east-1
```

If a region is not specified, the default is us-east-1.

It's also possible to specify key, keyid and region via a profile, either as a passed in dict, or as a string to pull from pillars or minion config:

```
myprofile:
    keyid: GKTADJGHEIQSXMKKRBJ08H
    key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
    region: us-east-1
```

depends boto3

`salt.modules.boto_kinesis.`**`create_stream`**`(`*stream_name, num_shards, region=None, key=None, keyid=None, profile=None*`)`

Create a stream with name stream_name and initial number of shards num_shards.

CLI example:

```
salt myminion boto_kinesis.create_stream my_stream N region=us-east-1
```

`salt.modules.boto_kinesis.`**`decrease_stream_retention_period`**`(`*stream_name, retention_hours, region=None, key=None, keyid=None, profile=None*`)`

Decrease stream retention period to retention_hours

CLI example:

```
salt myminion boto_kinesis.decrease_stream_retention_period my_stream N region=us-
↪east-1
```

`salt.modules.boto_kinesis.`**`delete_stream`**`(`*stream_name, region=None, key=None, keyid=None, profile=None*`)`

Delete the stream with name stream_name. This cannot be undone! All data will be lost!!

CLI example:

```
salt myminion boto_kinesis.delete_stream my_stream region=us-east-1
```

`salt.modules.boto_kinesis.`**`disable_enhanced_monitoring`**`(`*stream_name, metrics, region=None, key=None, keyid=None, profile=None*`)`

Disable enhanced monitoring for the specified shard-level metrics on stream stream_name

CLI example:

```
salt myminion boto_kinesis.disable_enhanced_monitoring my_stream ["metrics", "to",
↪ "disable"] region=us-east-1
```

`salt.modules.boto_kinesis.`**`enable_enhanced_monitoring`**`(`*stream_name, metrics, region=None, key=None, keyid=None, profile=None*`)`

Enable enhanced monitoring for the specified shard-level metrics on stream stream_name

CLI example:

```
salt myminion boto_kinesis.enable_enhanced_monitoring my_stream ["metrics", "to",
↪"enable"] region=us-east-1
```

`salt.modules.boto_kinesis.`**`exists`**`(`*stream_name, region=None, key=None, keyid=None, profile=None*`)`

Check if the stream exists. Returns False and the error if it does not.

CLI example:

```
salt myminion boto_kinesis.exists my_stream region=us-east-1
```

`salt.modules.boto_kinesis.`**`get_info_for_reshard`**`(`*stream_details*`)`

Collect some data: number of open shards, key range, etc. Modifies stream_details to add a sorted list of OpenShards. Returns (min_hash_key, max_hash_key, stream_details)

CLI example:

```
salt myminion boto_kinesis.get_info_for_reshard existing_stream_details
```

`salt.modules.boto_kinesis.`**`get_stream_when_active`**(*stream_name*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Get complete stream info from AWS, returning only when the stream is in the ACTIVE state. Continues to retry when stream is updating or creating. If the stream is deleted during retries, the loop will catch the error and break.

CLI example:

```
salt myminion boto_kinesis.get_stream_when_active my_stream region=us-east-1
```

`salt.modules.boto_kinesis.`**`increase_stream_retention_period`**(*stream_name*, *retention_hours*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Increase stream retention period to retention_hours

CLI example:

```
salt myminion boto_kinesis.increase_stream_retention_period my_stream N region=us-
↪east-1
```

`salt.modules.boto_kinesis.`**`list_streams`**(*region=None*, *key=None*, *keyid=None*, *profile=None*)

Return a list of all streams visible to the current account

CLI example:

```
salt myminion boto_kinesis.list_streams
```

`salt.modules.boto_kinesis.`**`long_int`**(*hash_key*)

The hash key is a 128-bit int, sent as a string. It's necessary to convert to int/long for comparison operations. This helper method handles python 2/3 incompatibility

CLI example:

```
salt myminion boto_kinesis.long_int some_MD5_hash_as_string
```

Returns long object if python 2.X, int object if python 3.X

`salt.modules.boto_kinesis.`**`reshard`**(*stream_name*, *desired_size*, *force=False*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Reshard a kinesis stream. Each call to this function will wait until the stream is ACTIVE, then make a single split or merge operation. This function decides where to split or merge with the assumption that the ultimate goal is a balanced partition space.

For safety, user must past in force=True; otherwise, the function will dry run.

CLI example:

```
salt myminion boto_kinesis.reshard my_stream N True region=us-east-1
```

Returns True if a split or merge was found/performed, False if nothing is needed

21.9.55 salt.modules.boto_kms

Connection module for Amazon KMS

New in version 2015.8.0.

configuration This module accepts explicit kms credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More Information available at:

```
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-
↪ec2.html
```

If IAM roles are not used you need to specify them either in a pillar or in the minion's config file:

```
kms.keyid: GKTADJGHEIQSXMKKRBJ08H
kms.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

A region may also be specified in the configuration:

```
kms.region: us-east-1
```

If a region is not specified, the default is us-east-1.

It's also possible to specify key, keyid and region via a profile, either as a passed in dict, or as a string to pull from pillars or minion config:

myprofile: keyid: GKTADJGHEIQSXMKKRBJ08H key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs region: us-east-1

depends boto

`salt.modules.boto_kms.`**`create_alias`**(*alias_name, target_key_id, region=None, key=None, keyid=None, profile=None*)

Create a display name for a key.

CLI example:

```
salt myminion boto_kms.create_alias 'alias/mykey' key_id
```

`salt.modules.boto_kms.`**`create_grant`**(*key_id, grantee_principal, retiring_principal=None, operations=None, constraints=None, grant_tokens=None, region=None, key=None, keyid=None, profile=None*)

Adds a grant to a key to specify who can access the key and under what conditions.

CLI example:

```
salt myminion boto_kms.create_grant 'alias/mykey' 'arn:aws:iam::1111111:/role/
↪myrole' operations='["Encrypt","Decrypt"]'
```

`salt.modules.boto_kms.`**`create_key`**(*policy=None, description=None, key_usage=None, region=None, key=None, keyid=None, profile=None*)

Creates a master key.

CLI example:

```
salt myminion boto_kms.create_key '{"Statement":...}' "My master key"
```

`salt.modules.boto_kms.`**`decrypt`**(*ciphertext_blob, encryption_context=None, grant_tokens=None, region=None, key=None, keyid=None, profile=None*)

Decrypt ciphertext.

CLI example:

```
salt myminion boto_kms.decrypt encrypted_ciphertext
```

`salt.modules.boto_kms.`**`describe_key`**(*key_id*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Get detailed information about a key.

CLI example:

```
salt myminion boto_kms.describe_key 'alias/mykey'
```

`salt.modules.boto_kms.`**`disable_key`**(*key_id*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Mark key as disabled.

CLI example:

```
salt myminion boto_kms.disable_key 'alias/mykey'
```

`salt.modules.boto_kms.`**`disable_key_rotation`**(*key_id*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Disable key rotation for specified key.

CLI example:

```
salt myminion boto_kms.disable_key_rotation 'alias/mykey'
```

`salt.modules.boto_kms.`**`enable_key`**(*key_id*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Mark key as enabled.

CLI example:

```
salt myminion boto_kms.enable_key 'alias/mykey'
```

`salt.modules.boto_kms.`**`enable_key_rotation`**(*key_id*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Disable key rotation for specified key.

CLI example:

```
salt myminion boto_kms.enable_key_rotation 'alias/mykey'
```

`salt.modules.boto_kms.`**`encrypt`**(*key_id*, *plaintext*, *encryption_context=None*, *grant_tokens=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Encrypt plaintext into cipher text using specified key.

CLI example:

```
salt myminion boto_kms.encrypt 'alias/mykey' 'myplaindata' '{"aws:username":
↪"myuser"}'
```

`salt.modules.boto_kms.`**`generate_data_key`**(*key_id*, *encryption_context=None*, *number_of_bytes=None*, *key_spec=None*, *grant_tokens=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Generate a secure data key.

CLI example:

```
salt myminion boto_kms.generate_data_key 'alias/mykey' number_of_bytes=1024 key_
↪spec=AES_128
```

`salt.modules.boto_kms.`**`generate_data_key_without_plaintext`**(*key_id, encryption_context=None, number_of_bytes=None, key_spec=None, grant_tokens=None, region=None, key=None, keyid=None, profile=None*)

Generate a secure data key without a plaintext copy of the key.

CLI example:

```
salt myminion boto_kms.generate_data_key_without_plaintext 'alias/mykey' number_
↪of_bytes=1024 key_spec=AES_128
```

`salt.modules.boto_kms.`**`generate_random`**(*number_of_bytes=None, region=None, key=None, keyid=None, profile=None*)

Generate a random string.

CLI example:

```
salt myminion boto_kms.generate_random number_of_bytes=1024
```

`salt.modules.boto_kms.`**`get_key_policy`**(*key_id, policy_name, region=None, key=None, keyid=None, profile=None*)

Get the policy for the specified key.

CLI example:

```
salt myminion boto_kms.get_key_policy 'alias/mykey' mypolicy
```

`salt.modules.boto_kms.`**`get_key_rotation_status`**(*key_id, region=None, key=None, keyid=None, profile=None*)

Get status of whether or not key rotation is enabled for a key.

CLI example:

```
salt myminion boto_kms.get_key_rotation_status 'alias/mykey'
```

`salt.modules.boto_kms.`**`key_exists`**(*key_id, region=None, key=None, keyid=None, profile=None*)

Check for the existence of a key.

CLI example:

```
salt myminion boto_kms.key_exists 'alias/mykey'
```

`salt.modules.boto_kms.`**`list_grants`**(*key_id, limit=None, marker=None, region=None, key=None, keyid=None, profile=None*)

List grants for the specified key.

CLI example:

```
salt myminion boto_kms.list_grants 'alias/mykey'
```

salt.modules.boto_kms.**list_key_policies**(*key_id*, *limit=None*, *marker=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

List key_policies for the specified key.

CLI example:

```
salt myminion boto_kms.list_key_policies 'alias/mykey'
```

salt.modules.boto_kms.**put_key_policy**(*key_id*, *policy_name*, *policy*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Attach a key policy to the specified key.

CLI example:

```
salt myminion boto_kms.put_key_policy 'alias/mykey' default '{"Statement":...}'
```

salt.modules.boto_kms.**re_encrypt**(*ciphertext_blob*, *destination_key_id*, *source_encryption_context=None*, *destination_encryption_context=None*, *grant_tokens=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Reencrypt encrypted data with a new master key.

CLI example:

```
salt myminion boto_kms.re_encrypt 'encrypted_data' 'alias/mynewkey' default '{
↪"Statement":...}'
```

salt.modules.boto_kms.**revoke_grant**(*key_id*, *grant_id*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Revoke a grant from a key.

CLI example:

```
salt myminion boto_kms.revoke_grant 'alias/mykey' 8u89hf-j09j...
```

salt.modules.boto_kms.**update_key_description**(*key_id*, *description*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Update a key's description.

CLI example:

```
salt myminion boto_kms.update_key_description 'alias/mykey' 'My key'
```

21.9.56 salt.modules.boto_lambda module

Connection module for Amazon Lambda

New in version 2016.3.0.

depends

- boto
- boto3

The dependencies listed above can be installed via package or pip.

configuration This module accepts explicit Lambda credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More Information available here.

If IAM roles are not used you need to specify them either in a pillar or in the minion's config file:

```
lambda.keyid: GKTADJGHEIQSXMKKRBJ08H
lambda.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

A region may also be specified in the configuration:

```
lambda.region: us-east-1
```

If a region is not specified, the default is us-east-1.

It's also possible to specify key, keyid and region via a profile, either as a passed in dict, or as a string to pull from pillars or minion config:

```
myprofile:
    keyid: GKTADJGHEIQSXMKKRBJ08H
    key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
    region: us-east-1
```

Changed in version 2015.8.0: All methods now return a dictionary. Create and delete methods return:

```
created: true
```

or

```
created: false
error:
  message: error message
```

Request methods (e.g., *describe_function*) return:

```
function:
  - {...}
  - {...}
```

or

```
error:
  message: error message
```

`salt.modules.boto_lambda.`**`add_permission`**(*FunctionName, StatementId, Action, Principal, SourceArn=None, SourceAccount=None, Qualifier=None, region=None, key=None, keyid=None, profile=None*)

Add a permission to a lambda function.

Returns {added: true} if the permission was added and returns {added: False} if the permission was not added.

CLI Example:

```
salt myminion boto_lamba.add_permission my_function my_id "lambda:*" \
                   s3.amazonaws.com aws:arn::::bucket-name \
                   aws-account-id
```

`salt.modules.boto_lambda.`**`alias_exists`**(*FunctionName, Name, region=None, key=None, keyid=None, profile=None*)

Given a function name and alias name, check to see if the given alias exists.

Returns True if the given alias exists and returns False if the given alias does not exist.

CLI Example:

```
salt myminion boto_lambda.alias_exists myfunction myalias
```

`salt.modules.boto_lambda.`**`create_alias`**(*FunctionName, Name, FunctionVersion, Description=”, region=None, key=None, keyid=None, profile=None*)

Given a valid config, create an alias to a function.

Returns {created: true} if the alias was created and returns {created: False} if the alias was not created.

CLI Example:

```
salt myminion boto_lamba.create_alias my_function my_alias $LATEST "An alias"
```

`salt.modules.boto_lambda.`**`create_event_source_mapping`**(*EventSourceArn, FunctionName, StartingPosition, Enabled=True, BatchSize=100, region=None, key=None, keyid=None, profile=None*)

Identifies a stream as an event source for a Lambda function. It can be either an Amazon Kinesis stream or an Amazon DynamoDB stream. AWS Lambda invokes the specified function when records are posted to the stream.

Returns {created: true} if the event source mapping was created and returns {created: False} if the event source mapping was not created.

CLI Example:

```
salt myminion boto_lamba.create_event_source_mapping arn::::eventsource⊠
↪myfunction LATEST
```

`salt.modules.boto_lambda.`**`create_function`**(*FunctionName, Runtime, Role, Handler, ZipFile=None, S3Bucket=None, S3Key=None, S3ObjectVersion=None, Description=”, Timeout=3, MemorySize=128, Publish=False, WaitForRole=False, RoleRetries=5, region=None, key=None, keyid=None, profile=None, VpcConfig=None, Environment=None*)

New in version 2017.7.0.

Given a valid config, create a function.

Environment The parent object that contains your environment's configuration settings. This is a dictionary of the form:

```
{
    'Variables': {
        'VariableName': 'VariableValue'
    }
}
```

Returns `{'created': True}` if the function was created and `{created: False}` if the function was not created.

CLI Example:

```
salt myminion boto_lamba.create_function my_function python2.7 my_role my_file.my_
↪function my_function.zip
```

salt.modules.boto_lambda.**delete_alias**(*FunctionName*, *Name*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Given a function name and alias name, delete the alias.

Returns {deleted: true} if the alias was deleted and returns {deleted: false} if the alias was not deleted.

CLI Example:

```
salt myminion boto_lambda.delete_alias myfunction myalias
```

salt.modules.boto_lambda.**delete_event_source_mapping**(*UUID=None*, *EventSourceArn=None*, *FunctionName=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Given an event source mapping ID or an event source ARN and FunctionName, delete the event source mapping

Returns {deleted: true} if the mapping was deleted and returns {deleted: false} if the mapping was not deleted.

CLI Example:

```
salt myminion boto_lambda.delete_event_source_mapping 260c423d-e8b5-4443-8d6a-
↪5e91b9ecd0fa
```

salt.modules.boto_lambda.**delete_function**(*FunctionName*, *Qualifier=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Given a function name and optional version qualifier, delete it.

Returns {deleted: true} if the function was deleted and returns {deleted: false} if the function was not deleted.

CLI Example:

```
salt myminion boto_lambda.delete_function myfunction
```

salt.modules.boto_lambda.**describe_alias**(*FunctionName*, *Name*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Given a function name and alias name describe the properties of the alias.

Returns a dictionary of interesting properties.

CLI Example:

```
salt myminion boto_lambda.describe_alias myalias
```

salt.modules.boto_lambda.**describe_event_source_mapping**(*UUID=None*, *EventSourceArn=None*, *FunctionName=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Given an event source mapping ID or an event source ARN and FunctionName, obtain the current settings of that mapping.

Returns a dictionary of interesting properties.

CLI Example:

```
salt myminion boto_lambda.describe_event_source_mapping uuid
```

salt.modules.boto_lambda.**describe_function**(*FunctionName*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Given a function name describe its properties.

Returns a dictionary of interesting properties.

CLI Example:

```
salt myminion boto_lambda.describe_function myfunction
```

salt.modules.boto_lambda.**event_source_mapping_exists**(*UUID=None*, *EventSourceArn=None*, *FunctionName=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Given an event source mapping ID or an event source ARN and FunctionName, check whether the mapping exists.

Returns True if the given alias exists and returns False if the given alias does not exist.

CLI Example:

```
salt myminion boto_lambda.alias_exists myfunction myalias
```

salt.modules.boto_lambda.**function_exists**(*FunctionName*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Given a function name, check to see if the given function name exists.

Returns True if the given function exists and returns False if the given function does not exist.

CLI Example:

```
salt myminion boto_lambda.function_exists myfunction
```

salt.modules.boto_lambda.**get_event_source_mapping_ids**(*EventSourceArn*, *FunctionName*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Given an event source and function name, return a list of mapping IDs

CLI Example:

```
salt myminion boto_lambda.get_event_source_mapping_ids arn:::: myfunction
```

salt.modules.boto_lambda.**get_permissions**(*FunctionName*, *Qualifier=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Get resource permissions for the given lambda function

Returns dictionary of permissions, by statement ID

CLI Example:

```
salt myminion boto_lamba.get_permissions my_function

permissions: {...}
```

salt.modules.boto_lambda.**list_function_versions**(*FunctionName*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

List the versions available for the given function.

Returns list of function versions

CLI Example:

```
versions:
  - {...}
  - {...}
```

`salt.modules.boto_lambda.`**`list_functions`**(*region=None*, *key=None*, *keyid=None*, *profile=None*)

List all Lambda functions visible in the current scope.

CLI Example:

```
salt myminion boto_lambda.list_functions
```

`salt.modules.boto_lambda.`**`remove_permission`**(*FunctionName*, *StatementId*, *Qualifier=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Remove a permission from a lambda function.

Returns {removed: true} if the permission was removed and returns {removed: False} if the permission was not removed.

CLI Example:

```
salt myminion boto_lamba.remove_permission my_function my_id
```

`salt.modules.boto_lambda.`**`update_alias`**(*FunctionName*, *Name*, *FunctionVersion=None*, *Description=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Update the named alias to the configuration.

Returns {updated: true} if the alias was updated and returns {updated: False} if the alias was not updated.

CLI Example:

```
salt myminion boto_lamba.update_alias my_lambda my_alias $LATEST
```

`salt.modules.boto_lambda.`**`update_event_source_mapping`**(*UUID*, *FunctionName=None*, *Enabled=None*, *BatchSize=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Update the event source mapping identified by the UUID.

Returns {updated: true} if the alias was updated and returns {updated: False} if the alias was not updated.

CLI Example:

```
salt myminion boto_lamba.update_event_source_mapping uuid FunctionName=new_
↪function
```

`salt.modules.boto_lambda.`**`update_function_code`**(*FunctionName*, *ZipFile=None*, *S3Bucket=None*, *S3Key=None*, *S3ObjectVersion=None*, *Publish=False*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Upload the given code to the named lambda function.

Returns {updated: true} if the function was updated and returns {updated: False} if the function was not updated.

CLI Example:

```
salt myminion boto_lamba.update_function_code my_function ZipFile=function.zip
```

`salt.modules.boto_lambda.`**`update_function_config`**(*FunctionName, Role=None, Handler=None, Description=None, Timeout=None, MemorySize=None, region=None, key=None, keyid=None, profile=None, VpcConfig=None, WaitForRole=False, RoleRetries=5, Environment=None*)

New in version 2017.7.0.

Update the named lambda function to the configuration.

Environment The parent object that contains your environment's configuration settings. This is a dictionary of the form:

```
{
    'Variables': {
        'VariableName': 'VariableValue'
    }
}
```

Returns `{'updated': True}` if the function was updated, and `{'updated': False}` if the function was not updated.

CLI Example:

```
salt myminion boto_lamba.update_function_config my_function my_role my_file.my_
↪function "my lambda function"
```

21.9.57 salt.modules.boto_rds

Connection module for Amazon RDS

New in version 2015.8.0.

configuration This module accepts explicit rds credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More Information available at:

```
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-
↪ec2.html
```

If IAM roles are not used you need to specify them either in a pillar or in the minion's config file:

```
rds.keyid: GKTADJGHEIQSXMKKRBJ08H
rds.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

A region may also be specified in the configuration:

```
rds.region: us-east-1
```

If a region is not specified, the default is us-east-1.

It's also possible to specify key, keyid and region via a profile, either as a passed in dict, or as a string to pull from pillars or minion config:

```
myprofile:
    keyid: GKTADJGHEIQSXMKKRBJ08H
    key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
    region: us-east-1
```

depends boto3

`salt.modules.boto_rds.`**`create`**(*name, allocated_storage, db_instance_class, engine, master_username, master_user_password, db_name=None, db_security_groups=None, vpc_security_group_ids=None, vpc_security_groups=None, availability_zone=None, db_subnet_group_name=None, preferred_maintenance_window=None, db_parameter_group_name=None, backup_retention_period=None, preferred_backup_window=None, port=None, multi_az=None, engine_version=None, auto_minor_version_upgrade=None, license_model=None, iops=None, option_group_name=None, character_set_name=None, publicly_accessible=None, wait_status=None, tags=None, db_cluster_identifier=None, storage_type=None, tde_credential_arn=None, tde_credential_password=None, storage_encrypted=None, kms_key_id=None, domain=None, copy_tags_to_snapshot=None, monitoring_interval=None, monitoring_role_arn=None, domain_iam_role_name=None, region=None, promotion_tier=None, key=None, keyid=None, profile=None*)

Create an RDS Instance

CLI example to create an RDS Instance:

```
salt myminion boto_rds.create myrds 10 db.t2.micro MySQL sqlusr sqlpassw
```

`salt.modules.boto_rds.`**`create_option_group`**(*name, engine_name, major_engine_version, option_group_description, tags=None, region=None, key=None, keyid=None, profile=None*)

Create an RDS option group

CLI example to create an RDS option group:

```
salt myminion boto_rds.create_option_group my-opt-group mysql 5.6
↪"group description"
```

`salt.modules.boto_rds.`**`create_parameter_group`**(*name, db_parameter_group_family, description, tags=None, region=None, key=None, keyid=None, profile=None*)

Create an RDS parameter group

CLI example to create an RDS parameter group:

```
salt myminion boto_rds.create_parameter_group my-param-group mysql5.6                    ␣
↪    "group description"
```

salt.modules.boto_rds.**create_read_replica**(*name, source_name, db_instance_class=None, availability_zone=None, port=None, auto_minor_version_upgrade=None, iops=None, option_group_name=None, publicly_accessible=None, tags=None, db_subnet_group_name=None, storage_type=None, copy_tags_to_snapshot=None, monitoring_interval=None, monitoring_role_arn=None, region=None, key=None, keyid=None, profile=None*)

Create an RDS read replica

CLI example to create an RDS read replica:

```
salt myminion boto_rds.create_read_replica replicaname source_name
```

salt.modules.boto_rds.**create_subnet_group**(*name, description, subnet_ids, tags=None, region=None, key=None, keyid=None, profile=None*)

Create an RDS subnet group

CLI example to create an RDS subnet group:

```
salt myminion boto_rds.create_subnet_group my-subnet-group            "group⊠
↪description" '[subnet-12345678, subnet-87654321]'          region=us-east-1
```

salt.modules.boto_rds.**delete**(*name, skip_final_snapshot=None, final_db_snapshot_identifier=None, region=None, key=None, keyid=None, profile=None, tags=None, wait_for_deletion=True, timeout=180*)

Delete an RDS instance.

CLI example:

```
salt myminion boto_rds.delete myrds skip_final_snapshot=True                ⊠
↪region=us-east-1
```

salt.modules.boto_rds.**delete_option_group**(*name, region=None, key=None, keyid=None, profile=None*)

Delete an RDS option group.

CLI example:

```
salt myminion boto_rds.delete_option_group my-opt-group                region=us-
↪east-1
```

salt.modules.boto_rds.**delete_parameter_group**(*name, region=None, key=None, keyid=None, profile=None*)

Delete an RDS parameter group.

CLI example:

```
salt myminion boto_rds.delete_parameter_group my-param-group                ⊠
↪region=us-east-1
```

salt.modules.boto_rds.**delete_subnet_group**(*name, region=None, key=None, keyid=None, profile=None*)

Delete an RDS subnet group.

CLI example:

```
salt myminion boto_rds.delete_subnet_group my-subnet-group                    ⊠
↪region=us-east-1
```

`salt.modules.boto_rds.`**`describe`**(*name*, *tags=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Return RDS instance details.

CLI example:

```
salt myminion boto_rds.describe myrds
```

`salt.modules.boto_rds.`**`describe_db_instances`**(*name=None*, *filters=None*, *jmespath='DBInstances'*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Return a detailed listing of some, or all, DB Instances visible in the current scope. Arbitrary subelements or subsections of the returned dataset can be selected by passing in a valid JMSEPath filter as well.

CLI example:

```
salt myminion boto_rds.describe_db_instances jmespath='DBInstances[*].
↪DBInstanceIdentifier'
```

`salt.modules.boto_rds.`**`describe_db_subnet_groups`**(*name=None*, *filters=None*, *jmespath='DBSubnetGroups'*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Return a detailed listing of some, or all, DB Subnet Groups visible in the current scope. Arbitrary subelements or subsections of the returned dataset can be selected by passing in a valid JMSEPath filter as well.

CLI example:

```
salt myminion boto_rds.describe_db_subnet_groups
```

`salt.modules.boto_rds.`**`describe_parameter_group`**(*name*, *Filters=None*, *MaxRecords=None*, *Marker=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Returns a list of *DBParameterGroup* descriptions. CLI example to description of parameter group:

```
salt myminion boto_rds.describe_parameter_group parametergroupname            ⊠
↪region=us-east-1
```

`salt.modules.boto_rds.`**`describe_parameters`**(*name*, *Source=None*, *MaxRecords=None*, *Marker=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Returns a list of *DBParameterGroup* parameters. CLI example to description of parameters

```
salt myminion boto_rds.describe_parameters parametergroupname                 ⊠
↪region=us-east-1
```

`salt.modules.boto_rds.`**`exists`**(*name*, *tags=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Check to see if an RDS exists.

CLI example:

```
salt myminion boto_rds.exists myrds region=us-east-1
```

`salt.modules.boto_rds.`**`get_endpoint`**(*name*, *tags=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Return the endpoint of an RDS instance.

CLI example:

```
salt myminion boto_rds.get_endpoint myrds
```

`salt.modules.boto_rds.`**`modify_db_instance`**(*name, allocated_storage=None, allow_major_version_upgrade=None, apply_immediately=None, auto_minor_version_upgrade=None, backup_retention_period=None, ca_certificate_identifier=None, character_set_name=None, copy_tags_to_snapshot=None, db_cluster_identifier=None, db_instance_class=None, db_name=None, db_parameter_group_name=None, db_port_number=None, db_security_groups=None, db_subnet_group_name=None, domain=None, domain_iam_role_name=None, engine_version=None, iops=None, kms_key_id=None, license_model=None, master_user_password=None, monitoring_interval=None, monitoring_role_arn=None, multi_az=None, new_db_instance_identifier=None, option_group_name=None, preferred_backup_window=None, preferred_maintenance_window=None, promotion_tier=None, publicly_accessible=None, storage_encrypted=None, storage_type=None, tde_credential_arn=None, tde_credential_password=None, vpc_security_group_ids=None, region=None, key=None, keyid=None, profile=None*)

Modify settings for a DB instance. CLI example to description of parameters

```
salt myminion boto_rds.modify_db_instance db_instance_identifier region=us-east-1
```

`salt.modules.boto_rds.`**`option_group_exists`**(*name, tags=None, region=None, key=None, keyid=None, profile=None*)

Check to see if an RDS option group exists.

CLI example:

```
salt myminion boto_rds.option_group_exists myoptiongr region=us-east-1
```

`salt.modules.boto_rds.`**`parameter_group_exists`**(*name, tags=None, region=None, key=None, keyid=None, profile=None*)

Check to see if an RDS parameter group exists.

CLI example:

```
salt myminion boto_rds.parameter_group_exists myparametergroup ␣
↪region=us-east-1
```

`salt.modules.boto_rds.`**`subnet_group_exists`**(*name, tags=None, region=None, key=None, keyid=None, profile=None*)

Check to see if an RDS subnet group exists.

CLI example:

```
salt myminion boto_rds.subnet_group_exists my-param-group                  ⊠
↪region=us-east-1
```

salt.modules.boto_rds.**update_parameter_group**(*name, parameters, apply_method='pending-reboot', tags=None, region=None, key=None, keyid=None, profile=None*)

Update an RDS parameter group.

CLI example:

```
salt myminion boto_rds.update_parameter_group my-param-group                  ⊠
↪parameters='{"back_log":1, "binlog_cache_size":4096}'                 region=us-
↪east-1
```

21.9.58 salt.modules.boto_route53

Connection module for Amazon Route53

New in version 2014.7.0.

configuration This module accepts explicit route53 credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More Information available at:

```
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-
↪ec2.html
```

If IAM roles are not used you need to specify them either in a pillar or in the minion's config file:

```
route53.keyid: GKTADJGHEIQSXMKKRBJ08H
route53.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

A region may also be specified in the configuration:

```
route53.region: us-east-1
```

If a region is not specified, the default is 'universal', which is what the boto_route53 library expects, rather than None.

It's also possible to specify key, keyid and region via a profile, either as a passed in dict, or as a string to pull from pillars or minion config:

```
myprofile:
  keyid: GKTADJGHEIQSXMKKRBJ08H
  key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
  region: us-east-1
```

depends boto

salt.modules.boto_route53.**add_record**(*name, value, zone, record_type, identifier=None, ttl=None, region=None, key=None, keyid=None, profile=None, wait_for_sync=True, split_dns=False, private_zone=False, retry_on_rate_limit=None, rate_limit_retries=None, retry_on_errors=True, error_retries=5*)

Add a record to a zone.

CLI example:

```
salt myminion boto_route53.add_record test.example.org 1.1.1.1 example.
↪org A
```

retry_on_errors

Continue to query if the zone exists after an error is raised. The previously used argument *retry_on_rate_limit* was deprecated for this argument. Users can still use *retry_on_rate_limit* to ensure backwards compatibility, but please migrate to using the favored *retry_on_errors* argument instead.

error_retries Amount of times to attempt to query if the zone exists. The previously used argument *rate_limit_retries* was deprecated for this arguments. Users can still use *rate_limit_retries* to ensure backwards compatibility, but please migrate to using the favored *error_retries* argument instead.

`salt.modules.boto_route53.`**`create_healthcheck`**`(`*ip_addr=None, fqdn=None, region=None, key=None, keyid=None, profile=None, port=53, hc_type='TCP', resource_path='', string_match=None, request_interval=30, failure_threshold=3, retry_on_errors=True, error_retries=5*`)`

Create a Route53 healthcheck

New in version 2018.3.0.

ip_addr

IP address to check. ip_addr or fqdn is required.

fqdn

Domain name of the endpoint to check. ip_addr or fqdn is required

port

Port to check

hc_type

Healthcheck type. HTTP | HTTPS | HTTP_STR_MATCH | HTTPS_STR_MATCH | TCP

resource_path

Path to check

string_match

If hc_type is HTTP_STR_MATCH or HTTPS_STR_MATCH, the string to search for in the response body from the specified resource

request_interval

The number of seconds between the time that Amazon Route 53 gets a response from your endpoint and the time that it sends the next health-check request.

failure_threshold

The number of consecutive health checks that an endpoint must pass or fail for Amazon Route 53 to change the current status of the endpoint from unhealthy to healthy or vice versa.

region

Region endpoint to connect to

key

AWS key

keyid

AWS keyid

profile

AWS pillar profile

CLI Example:

```
salt myminion boto_route53.create_healthcheck 192.168.0.1
salt myminion boto_route53.create_healthcheck 192.168.0.1 port=443 hc_type=HTTPS ⊠
↪                                                    resource_path=/ fqdn=blog.
↪saltstack.furniture
```

`salt.modules.boto_route53.`**`create_hosted_zone`**(*domain_name*, *caller_ref=None*, *comment=''*, *private_zone=False*, *vpc_id=None*, *vpc_name=None*, *vpc_region=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Create a new Route53 Hosted Zone. Returns a Python data structure with information about the newly created Hosted Zone.

domain_name The name of the domain. This must be fully-qualified, terminating with a period. This is the name you have registered with your domain registrar. It is also the name you will delegate from your registrar to the Amazon Route 53 delegation servers returned in response to this request.

caller_ref A unique string that identifies the request and that allows create_hosted_zone() calls to be retried without the risk of executing the operation twice. It can take several minutes for the change to replicate globally, and change from PENDING to INSYNC status. Thus it's best to provide some value for this where possible, since duplicate calls while the first is in PENDING status will be accepted and can lead to multiple copies of the zone being created. On the other hand, if a zone is created with a given caller_ref, then deleted, a second attempt to create a zone with the same caller_ref will fail until that caller_ref is flushed from the Route53 system, which can take upwards of 24 hours.

comment Any comments you want to include about the hosted zone.

private_zone Set True if creating a private hosted zone.

vpc_id When creating a private hosted zone, either the VPC ID or VPC Name to associate with is required. Exclusive with vpe_name. Ignored when creating a non-private zone.

vpc_name When creating a private hosted zone, either the VPC ID or VPC Name to associate with is required. Exclusive with vpe_id. Ignored when creating a non-private zone.

vpc_region When creating a private hosted zone, the region of the associated VPC is required. If not provided, an effort will be made to determine it from vpc_id or vpc_name, where possible. If this fails, you'll need to provide an explicit value for this option. Ignored when creating a non-private zone.

region Region endpoint to connect to.

key AWS key to bind with.

keyid AWS keyid to bind with.

profile Dict, or pillar key pointing to a dict, containing AWS region/key/keyid.

CLI Example:

```
salt myminion boto_route53.create_hosted_zone example.org
```

`salt.modules.boto_route53.`**`create_zone`**(*zone*, *private=False*, *vpc_id=None*, *vpc_region=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Create a Route53 hosted zone.

New in version 2015.8.0.

zone DNS zone to create

private True/False if the zone will be a private zone

vpc_id VPC ID to associate the zone to (required if private is True)

vpc_region VPC Region (required if private is True)

region region endpoint to connect to

key AWS key

keyid AWS keyid

profile AWS pillar profile

CLI Example:

```
salt myminion boto_route53.create_zone example.org
```

`salt.modules.boto_route53.`**`delete_record`**(*name*, *zone*, *record_type*, *identifier=None*, *all_records=False*, *region=None*, *key=None*, *keyid=None*, *profile=None*, *wait_for_sync=True*, *split_dns=False*, *private_zone=False*, *retry_on_rate_limit=None*, *rate_limit_retries=None*, *retry_on_errors=True*, *error_retries=5*)

Modify a record in a zone.

CLI example:

```
salt myminion boto_route53.delete_record test.example.org example.org A
```

retry_on_errors

Continue to query if the zone exists after an error is raised. The previously used argument *retry_on_rate_limit* was deprecated for this argument. Users can still use *retry_on_rate_limit* to ensure backwards compatibility, but please migrate to using the favored *retry_on_errors* argument instead.

error_retries Amount of times to attempt to query if the zone exists. The previously used argument *rate_limit_retries* was deprecated for this arguments. Users can still use *rate_limit_retries* to ensure backwards compatibility, but please migrate to using the favored *error_retries* argument instead.

`salt.modules.boto_route53.`**`delete_zone`**(*zone*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Delete a Route53 hosted zone.

New in version 2015.8.0.

CLI Example:

```
salt myminion boto_route53.delete_zone example.org
```

`salt.modules.boto_route53.`**`describe_hosted_zones`**(*zone_id=None*, *domain_name=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Return detailed info about one, or all, zones in the bound account. If neither zone_id nor domain_name is provided, return all zones. Note that the return format is slightly different between the 'all' and 'single' description types.

zone_id The unique identifier for the Hosted Zone

domain_name The FQDN of the Hosted Zone (including final period)

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

CLI Example:

```
salt myminion boto_route53.describe_hosted_zones domain_name=foo.bar.com.
↪        profile='{"region": "us-east-1", "keyid": "A12345678AB", "key":
↪"xblahblahblah"}'
```

`salt.modules.boto_route53.`**`get_record`**(*name*, *zone*, *record_type*, *fetch_all=False*, *region=None*, *key=None*, *keyid=None*, *profile=None*, *split_dns=False*, *private_zone=False*, *identifier=None*, *retry_on_rate_limit=None*, *rate_limit_retries=None*, *retry_on_errors=True*, *error_retries=5*)

Get a record from a zone.

CLI example:

```
salt myminion boto_route53.get_record test.example.org example.org A
```

retry_on_errors

Continue to query if the zone exists after an error is raised. The previously used argument *retry_on_rate_limit* was deprecated for this argument. Users can still use *retry_on_rate_limit* to ensure backwards compatibility, but please migrate to using the favored *retry_on_errors* argument instead.

error_retries Amount of times to attempt to query if the zone exists. The previously used argument *rate_limit_retries* was deprecated for this arguments. Users can still use *rate_limit_retries* to ensure backwards compatibility, but please migrate to using the favored *error_retries* argument instead.

`salt.modules.boto_route53.`**`list_all_zones_by_id`**(*region=None*, *key=None*, *keyid=None*, *profile=None*)

List, by their IDs, all hosted zones in the bound account.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

CLI Example:

```
salt myminion boto_route53.list_all_zones_by_id
```

`salt.modules.boto_route53.`**`list_all_zones_by_name`**(*region=None*, *key=None*, *keyid=None*, *profile=None*)

List, by their FQDNs, all hosted zones in the bound account.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

CLI Example:

```
salt myminion boto_route53.list_all_zones_by_name
```

`salt.modules.boto_route53.`**`update_record`**(*name*, *value*, *zone*, *record_type*, *identifier=None*, *ttl=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*, *wait_for_sync=True*, *split_dns=False*, *private_zone=False*, *retry_on_rate_limit=None*, *rate_limit_retries=None*, *retry_on_errors=True*, *error_retries=5*)

Modify a record in a zone.

CLI example:

```
salt myminion boto_route53.modify_record test.example.org 1.1.1.1 example.
↪org A
```

retry_on_errors

Continue to query if the zone exists after an error is raised. The previously used argument *retry_on_rate_limit* was deprecated for this argument. Users can still use *retry_on_rate_limit* to ensure backwards compatibility, but please migrate to using the favored *retry_on_errors* argument instead.

error_retries Amount of times to attempt to query if the zone exists. The previously used argument *rate_limit_retries* was deprecated for this arguments. Users can still use *rate_limit_retries* to ensure backwards compatibility, but please migrate to using the favored *error_retries* argument instead.

`salt.modules.boto_route53.`**`zone_exists`**(*zone*, *region=None*, *key=None*, *keyid=None*, *profile=None*, *retry_on_rate_limit=None*, *rate_limit_retries=None*, *retry_on_errors=True*, *error_retries=5*)

Check for the existence of a Route53 hosted zone.

New in version 2015.8.0.

CLI Example:

```
salt myminion boto_route53.zone_exists example.org
```

retry_on_errors

Continue to query if the zone exists after an error is raised. The previously used argument *retry_on_rate_limit* was deprecated for this argument. Users can still use *retry_on_rate_limit* to ensure backwards compatibility, but please migrate to using the favored *retry_on_errors* argument instead.

error_retries Amount of times to attempt to query if the zone exists. The previously used argument *rate_limit_retries* was deprecated for this arguments. Users can still use *rate_limit_retries* to ensure backwards compatibility, but please migrate to using the favored *error_retries* argument instead.

21.9.59 salt.modules.boto_s3_bucket module

Connection module for Amazon S3 Buckets

New in version 2016.3.0.

depends

- boto
- boto3

The dependencies listed above can be installed via package or pip.

configuration This module accepts explicit Lambda credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More Information available at:

```
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-
↪ec2.html
```

If IAM roles are not used you need to specify them either in a pillar or in the minion's config file:

```
s3.keyid: GKTADJGHEIQSXMKKRBJ08H
s3.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

A region may also be specified in the configuration:

```
s3.region: us-east-1
```

If a region is not specified, the default is us-east-1.

It's also possible to specify key, keyid and region via a profile, either as a passed in dict, or as a string to pull from pillars or minion config:

```
myprofile:
    keyid: GKTADJGHEIQSXMKKRBJ08H
    key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
    region: us-east-1
```

`salt.modules.boto_s3_bucket.`**`create`**(*Bucket*, *ACL=None*, *LocationConstraint=None*, *GrantFullControl=None*, *GrantRead=None*, *GrantReadACP=None*, *GrantWrite=None*, *GrantWriteACP=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Given a valid config, create an S3 Bucket.

Returns {created: true} if the bucket was created and returns {created: False} if the bucket was not created.

CLI Example:

```
salt myminion boto_s3_bucket.create my_bucket \
                 GrantFullControl='emailaddress=example@example.com' \
                 GrantRead='uri="http://acs.amazonaws.com/groups/global/AllUsers"
↪' \
                 GrantReadACP='emailaddress="exampl@example.com",id=
↪"2345678909876432"' \
                 LocationConstraint=us-west-1
```

`salt.modules.boto_s3_bucket.`**`delete`**(*Bucket*, *MFA=None*, *RequestPayer=None*, *Force=False*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Given a bucket name, delete it, optionally emptying it first.

Returns {deleted: true} if the bucket was deleted and returns {deleted: false} if the bucket was not deleted.

CLI Example:

```
salt myminion boto_s3_bucket.delete mybucket
```

`salt.modules.boto_s3_bucket.`**`delete_cors`**(*Bucket*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Delete the CORS configuration for the given bucket

Returns {deleted: true} if CORS was deleted and returns {deleted: False} if CORS was not deleted.

CLI Example:

```
salt myminion boto_s3_bucket.delete_cors my_bucket
```

`salt.modules.boto_s3_bucket.`**`delete_lifecycle_configuration`**(*Bucket*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Delete the lifecycle configuration for the given bucket

Returns {deleted: true} if Lifecycle was deleted and returns {deleted: False} if Lifecycle was not deleted.

CLI Example:

```
salt myminion boto_s3_bucket.delete_lifecycle_configuration my_bucket
```

`salt.modules.boto_s3_bucket.`**`delete_objects`**(*Bucket*, *Delete*, *MFA=None*, *RequestPayer=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Delete objects in a given S3 bucket.

Returns {deleted: true} if all objects were deleted and {deleted: false, failed: [key, …]} otherwise

CLI Example:

```
salt myminion boto_s3_bucket.delete_objects mybucket '{Objects: [Key: myobject]}'
```

`salt.modules.boto_s3_bucket.`**`delete_policy`**(*Bucket*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Delete the policy from the given bucket

Returns {deleted: true} if policy was deleted and returns {deleted: False} if policy was not deleted.

CLI Example:

```
salt myminion boto_s3_bucket.delete_policy my_bucket
```

`salt.modules.boto_s3_bucket.`**`delete_replication`**`(`*Bucket, region=None, key=None, keyid=None, profile=None*`)`

Delete the replication config from the given bucket

Returns {deleted: true} if replication configuration was deleted and returns {deleted: False} if replication configuration was not deleted.

CLI Example:

```
salt myminion boto_s3_bucket.delete_replication my_bucket
```

`salt.modules.boto_s3_bucket.`**`delete_tagging`**`(`*Bucket, region=None, key=None, keyid=None, profile=None*`)`

Delete the tags from the given bucket

Returns {deleted: true} if tags were deleted and returns {deleted: False} if tags were not deleted.

CLI Example:

```
salt myminion boto_s3_bucket.delete_tagging my_bucket
```

`salt.modules.boto_s3_bucket.`**`delete_website`**`(`*Bucket, region=None, key=None, keyid=None, profile=None*`)`

Remove the website configuration from the given bucket

Returns {deleted: true} if website configuration was deleted and returns {deleted: False} if website configuration was not deleted.

CLI Example:

```
salt myminion boto_s3_bucket.delete_website my_bucket
```

`salt.modules.boto_s3_bucket.`**`describe`**`(`*Bucket, region=None, key=None, keyid=None, profile=None*`)`

Given a bucket name describe its properties.

Returns a dictionary of interesting properties.

CLI Example:

```
salt myminion boto_s3_bucket.describe mybucket
```

`salt.modules.boto_s3_bucket.`**`empty`**`(`*Bucket, MFA=None, RequestPayer=None, region=None, key=None, keyid=None, profile=None*`)`

Delete all objects in a given S3 bucket.

Returns {deleted: true} if all objects were deleted and {deleted: false, failed: [key, ...]} otherwise

CLI Example:

```
salt myminion boto_s3_bucket.empty mybucket
```

`salt.modules.boto_s3_bucket.`**`exists`**`(`*Bucket, region=None, key=None, keyid=None, profile=None*`)`

Given a bucket name, check to see if the given bucket exists.

Returns True if the given bucket exists and returns False if the given bucket does not exist.

CLI Example:

```
salt myminion boto_s3_bucket.exists mybucket
```

`salt.modules.boto_s3_bucket.`**`list`**(*region=None, key=None, keyid=None, profile=None*)

List all buckets owned by the authenticated sender of the request.

Returns list of buckets

CLI Example:

```
Owner: {...}
Buckets:
  - {...}
  - {...}
```

`salt.modules.boto_s3_bucket.`**`list_object_versions`**(*Bucket, Delimiter=None, EncodingType=None, Prefix=None, region=None, key=None, keyid=None, profile=None*)

List objects in a given S3 bucket.

Returns a list of objects.

CLI Example:

```
salt myminion boto_s3_bucket.list_object_versions mybucket
```

`salt.modules.boto_s3_bucket.`**`list_objects`**(*Bucket, Delimiter=None, EncodingType=None, Prefix=None, FetchOwner=False, StartAfter=None, region=None, key=None, keyid=None, profile=None*)

List objects in a given S3 bucket.

Returns a list of objects.

CLI Example:

```
salt myminion boto_s3_bucket.list_objects mybucket
```

`salt.modules.boto_s3_bucket.`**`put_acl`**(*Bucket, ACL=None, AccessControlPolicy=None, GrantFullControl=None, GrantRead=None, GrantReadACP=None, GrantWrite=None, GrantWriteACP=None, region=None, key=None, keyid=None, profile=None*)

Given a valid config, update the ACL for a bucket.

Returns {updated: true} if the ACL was updated and returns {updated: False} if the ACL was not updated.

CLI Example:

```
salt myminion boto_s3_bucket.put_acl my_bucket 'public' \
                GrantFullControl='emailaddress=example@example.com' \
                GrantRead='uri="http://acs.amazonaws.com/groups/global/AllUsers"
↪' \
                GrantReadACP='emailaddress="exampl@example.com",id=
↪"2345678909876432"'
```

`salt.modules.boto_s3_bucket.`**`put_cors`**(*Bucket, CORSRules, region=None, key=None, keyid=None, profile=None*)

Given a valid config, update the CORS rules for a bucket.

Returns {updated: true} if CORS was updated and returns {updated: False} if CORS was not updated.

CLI Example:

```
salt myminion boto_s3_bucket.put_cors my_bucket '[{\
      "AllowedHeaders":[],\
      "AllowedMethods":["GET"],\
      "AllowedOrigins":["*"],\
      "ExposeHeaders":[],\
      "MaxAgeSeconds":123,\
}]'
```

`salt.modules.boto_s3_bucket.`**`put_lifecycle_configuration`**(*Bucket, Rules, region=None, key=None, keyid=None, profile=None*)

Given a valid config, update the Lifecycle rules for a bucket.

Returns {updated: true} if Lifecycle was updated and returns {updated: False} if Lifecycle was not updated.

CLI Example:

```
salt myminion boto_s3_bucket.put_lifecycle_configuration my_bucket '[{\
      "Expiration": {...},\
      "ID": "idstring",\
      "Prefix": "prefixstring",\
      "Status": "enabled",\
      "Transitions": [{...},],\
      "NoncurrentVersionTransitions": [{...},],\
      "NoncurrentVersionExpiration": {...},\
}]'
```

`salt.modules.boto_s3_bucket.`**`put_logging`**(*Bucket, TargetBucket=None, TargetPrefix=None, TargetGrants=None, region=None, key=None, keyid=None, profile=None*)

Given a valid config, update the logging parameters for a bucket.

Returns {updated: true} if parameters were updated and returns {updated: False} if parameters were not updated.

CLI Example:

```
salt myminion boto_s3_bucket.put_logging my_bucket log_bucket '[{...}]' prefix
```

`salt.modules.boto_s3_bucket.`**`put_notification_configuration`**(*Bucket, TopicConfigurations=None, QueueConfigurations=None, LambdaFunctionConfigurations=None, region=None, key=None, keyid=None, profile=None*)

Given a valid config, update the notification parameters for a bucket.

Returns {updated: true} if parameters were updated and returns {updated: False} if parameters were not updated.

CLI Example:

```
salt myminion boto_s3_bucket.put_notification_configuration my_bucket
        [{...}] \
        [{...}] \
        [{...}]
```

`salt.modules.boto_s3_bucket.`**`put_policy`**`(`*Bucket, Policy, region=None, key=None, keyid=None, profile=None*`)`

Given a valid config, update the policy for a bucket.

Returns {updated: true} if policy was updated and returns {updated: False} if policy was not updated.

CLI Example:

```
salt myminion boto_s3_bucket.put_policy my_bucket {...}
```

`salt.modules.boto_s3_bucket.`**`put_replication`**`(`*Bucket, Role, Rules, region=None, key=None, keyid=None, profile=None*`)`

Given a valid config, update the replication configuration for a bucket.

Returns {updated: true} if replication configuration was updated and returns {updated: False} if replication configuration was not updated.

CLI Example:

```
salt myminion boto_s3_bucket.put_replication my_bucket my_role [...]
```

`salt.modules.boto_s3_bucket.`**`put_request_payment`**`(`*Bucket, Payer, region=None, key=None, keyid=None, profile=None*`)`

Given a valid config, update the request payment configuration for a bucket.

Returns {updated: true} if request payment configuration was updated and returns {updated: False} if request payment configuration was not updated.

CLI Example:

```
salt myminion boto_s3_bucket.put_request_payment my_bucket Requester
```

`salt.modules.boto_s3_bucket.`**`put_tagging`**`(`*Bucket, region=None, key=None, keyid=None, profile=None, **kwargs*`)`

Given a valid config, update the tags for a bucket.

Returns {updated: true} if tags were updated and returns {updated: False} if tags were not updated.

CLI Example:

```
salt myminion boto_s3_bucket.put_tagging my_bucket my_role [...]
```

`salt.modules.boto_s3_bucket.`**`put_versioning`**`(`*Bucket, Status, MFADelete=None, MFA=None, region=None, key=None, keyid=None, profile=None*`)`

Given a valid config, update the versioning configuration for a bucket.

Returns {updated: true} if versioning configuration was updated and returns {updated: False} if versioning configuration was not updated.

CLI Example:

```
salt myminion boto_s3_bucket.put_versioning my_bucket Enabled
```

`salt.modules.boto_s3_bucket.`**`put_website`**`(`*Bucket, ErrorDocument=None, IndexDocument=None, RedirectAllRequestsTo=None, RoutingRules=None, region=None, key=None, keyid=None, profile=None*`)`

Given a valid config, update the website configuration for a bucket.

Returns {updated: true} if website configuration was updated and returns {updated: False} if website configuration was not updated.

CLI Example:

```
salt myminion boto_s3_bucket.put_website my_bucket IndexDocument='{"Suffix":
↪"index.html"}'
```

21.9.60 salt.modules.boto_secgroup

Connection module for Amazon Security Groups

New in version 2014.7.0.

configuration This module accepts explicit ec2 credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More Information available at:

```
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-
↪ec2.html
```

If IAM roles are not used you need to specify them either in a pillar or in the minion's config file:

```
secgroup.keyid: GKTADJGHEIQSXMKKRBJ08H
secgroup.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

A region may also be specified in the configuration:

```
secgroup.region: us-east-1
```

If a region is not specified, the default is us-east-1.

It's also possible to specify key, keyid and region via a profile, either as a passed in dict, or as a string to pull from pillars or minion config:

```
myprofile:
    keyid: GKTADJGHEIQSXMKKRBJ08H
    key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
    region: us-east-1
```

depends boto

`salt.modules.boto_secgroup.`**`authorize`**(*name=None, source_group_name=None, source_group_owner_id=None, ip_protocol=None, from_port=None, to_port=None, cidr_ip=None, group_id=None, source_group_group_id=None, region=None, key=None, keyid=None, profile=None, vpc_id=None, vpc_name=None, egress=False*)

Add a new rule to an existing security group.

CLI example:

```
salt myminion boto_secgroup.authorize mysecgroup ip_protocol=tcp from_port=80 to_
↪port=80 cidr_ip='['10.0.0.0/8', '192.168.0.0/24']'
```

`salt.modules.boto_secgroup.`**`convert_to_group_ids`**(*groups, vpc_id=None, vpc_name=None, region=None, key=None, keyid=None, profile=None*)

Given a list of security groups and a vpc_id, convert_to_group_ids will convert all list items in the given list to security group ids.

CLI example:

```
salt myminion boto_secgroup.convert_to_group_ids mysecgroup vpc-89yhh7h
```

`salt.modules.boto_secgroup.`**`create`**`(`*name, description, vpc_id=None, vpc_name=None, region=None, key=None, keyid=None, profile=None*`)`

Create a security group.

CLI example:

```
salt myminion boto_secgroup.create mysecgroup 'My Security Group'
```

`salt.modules.boto_secgroup.`**`delete`**`(`*name=None, group_id=None, region=None, key=None, keyid=None, profile=None, vpc_id=None, vpc_name=None*`)`

Delete a security group.

CLI example:

```
salt myminion boto_secgroup.delete mysecgroup
```

`salt.modules.boto_secgroup.`**`delete_tags`**`(`*tags, name=None, group_id=None, vpc_name=None, vpc_id=None, region=None, key=None, keyid=None, profile=None*`)`

Deletes tags from a security group.

New in version 2016.3.0.

tags a list of tags to remove

name the name of the security group

group_id the group id of the security group (in lie of a name/vpc combo)

vpc_name the name of the vpc to search the named group for

vpc_id the id of the vpc, in lieu of the vpc_name

region the amazon region

key amazon key

keyid amazon keyid

profile amazon profile

CLI example:

```
salt myminion boto_secgroup.delete_tags ['TAG_TO_DELETE1','TAG_TO_DELETE2']␣
↪security_group_name vpc_id=vpc-13435 profile=my_aws_profile
```

`salt.modules.boto_secgroup.`**`exists`**`(`*name=None, region=None, key=None, keyid=None, profile=None, vpc_id=None, vpc_name=None, group_id=None*`)`

Check to see if a security group exists.

CLI example:

```
salt myminion boto_secgroup.exists mysecgroup
```

`salt.modules.boto_secgroup.`**`get_all_security_groups`**`(`*groupnames=None, group_ids=None, filters=None, region=None, key=None, keyid=None, profile=None*`)`

Return a list of all Security Groups matching the given criteria and filters.

Note that the `groupnames` argument only functions correctly for EC2 Classic and default VPC Security Groups. To find groups by name in other VPCs you'll want to use the `group-name` filter instead.

The valid keys for the `filters` argument can be found in AWS's API documentation.

CLI example:

```
salt myminion boto_secgroup.get_all_security_groups filters='{group-name: mygroup}
↪'
```

`salt.modules.boto_secgroup.`**`get_config`**(*name=None, group_id=None, region=None, key=None, keyid=None, profile=None, vpc_id=None, vpc_name=None*)

Get the configuration for a security group.

CLI example:

```
salt myminion boto_secgroup.get_config mysecgroup
```

`salt.modules.boto_secgroup.`**`get_group_id`**(*name, vpc_id=None, vpc_name=None, region=None, key=None, keyid=None, profile=None*)

Get a Group ID given a Group Name or Group Name and VPC ID

CLI example:

```
salt myminion boto_secgroup.get_group_id mysecgroup
```

`salt.modules.boto_secgroup.`**`revoke`**(*name=None, source_group_name=None, source_group_owner_id=None, ip_protocol=None, from_port=None, to_port=None, cidr_ip=None, group_id=None, source_group_group_id=None, region=None, key=None, keyid=None, profile=None, vpc_id=None, vpc_name=None, egress=False*)

Remove a rule from an existing security group.

CLI example:

```
salt myminion boto_secgroup.revoke mysecgroup ip_protocol=tcp from_port=80 to_
↪port=80 cidr_ip='10.0.0.0/8'
```

`salt.modules.boto_secgroup.`**`set_tags`**(*tags, name=None, group_id=None, vpc_name=None, vpc_id=None, region=None, key=None, keyid=None, profile=None*)

Sets tags on a security group.

New in version 2016.3.0.

tags a dict of key:value pair of tags to set on the security group

name the name of the security group

group_id the group id of the security group (in lie of a name/vpc combo)

vpc_name the name of the vpc to search the named group for

vpc_id the id of the vpc, in lieu of the vpc_name

region the amazon region

key amazon key

keyid amazon keyid

profile amazon profile

CLI example:

```
salt myminion boto_secgroup.set_tags "{'TAG1': 'Value1', 'TAG2': 'Value2'}"␣
↪security_group_name vpc_id=vpc-13435 profile=my_aws_profile
```

21.9.61 salt.modules.boto_sns

Connection module for Amazon SNS

configuration This module accepts explicit sns credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More Information available at:

```
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-
↪ec2.html
```

If IAM roles are not used you need to specify them either in a pillar or in the minion's config file:

```
sns.keyid: GKTADJGHEIQSXMKKRBJ08H
sns.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

A region may also be specified in the configuration:

```
sns.region: us-east-1
```

If a region is not specified, the default is us-east-1.

It's also possible to specify key, keyid and region via a profile, either as a passed in dict, or as a string to pull from pillars or minion config:

```
myprofile:
    keyid: GKTADJGHEIQSXMKKRBJ08H
    key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
    region: us-east-1
```

depends boto

`salt.modules.boto_sns.`**`create`**(*name*, *region=None*, *key=None*, *keyid=None*, *profile=None*)
Create an SNS topic.

CLI example to create a topic:

```
salt myminion boto_sns.create mytopic region=us-east-1
```

`salt.modules.boto_sns.`**`delete`**(*name*, *region=None*, *key=None*, *keyid=None*, *profile=None*)
Delete an SNS topic.

CLI example to delete a topic:

```
salt myminion boto_sns.delete mytopic region=us-east-1
```

`salt.modules.boto_sns.`**`exists`**(*name*, *region=None*, *key=None*, *keyid=None*, *profile=None*)
Check to see if an SNS topic exists.

CLI example:

```
salt myminion boto_sns.exists mytopic region=us-east-1
```

`salt.modules.boto_sns.`**`get_all_subscriptions_by_topic`**(*name*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Get list of all subscriptions to a specific topic.

CLI example to delete a topic:

```
salt myminion boto_sns.get_all_subscriptions_by_topic mytopic region=us-east-1
```

`salt.modules.boto_sns.`**`get_all_topics`**(*region=None*, *key=None*, *keyid=None*, *profile=None*)

Returns a list of the all topics..

CLI example:

```
salt myminion boto_sns.get_all_topics
```

`salt.modules.boto_sns.`**`get_arn`**(*name*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Returns the full ARN for a given topic name.

CLI example:

```
salt myminion boto_sns.get_arn mytopic
```

`salt.modules.boto_sns.`**`subscribe`**(*topic*, *protocol*, *endpoint*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Subscribe to a Topic.

CLI example to delete a topic:

```
salt myminion boto_sns.subscribe mytopic https https://www.example.com/sns-
↪endpoint region=us-east-1
```

`salt.modules.boto_sns.`**`unsubscribe`**(*topic*, *subscription_arn*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Unsubscribe a specific SubscriptionArn of a topic.

CLI Example:

```
salt myminion boto_sns.unsubscribe my_topic my_subscription_arn region=us-east-1
```

New in version 2016.11.0.

21.9.62 salt.modules.boto_sqs

Connection module for Amazon SQS

New in version 2014.7.0.

configuration This module accepts explicit sqs credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More information available at:

```
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-
↪ec2.html
```

If IAM roles are not used you need to specify them either in a pillar or in the minion's config file:

```
sqs.keyid: GKTADJGHEIQSXMKKRBJ08H
sqs.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

A region may also be specified in the configuration:

```
sqs.region: us-east-1
```

If a region is not specified, the default is us-east-1.

It's also possible to specify key, keyid and region via a profile, either as a passed in dict, or as a string to pull from pillars or minion config:

```
myprofile:
    keyid: GKTADJGHEIQSXMKKRBJ08H
    key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
    region: us-east-1
```

depends boto3

`salt.modules.boto_sqs.`**`create`**(*name, attributes=None, region=None, key=None, keyid=None, profile=None*)

Create an SQS queue.

CLI Example:

```
salt myminion boto_sqs.create myqueue region=us-east-1
```

`salt.modules.boto_sqs.`**`delete`**(*name, region=None, key=None, keyid=None, profile=None*)

Delete an SQS queue.

CLI Example:

```
salt myminion boto_sqs.delete myqueue region=us-east-1
```

`salt.modules.boto_sqs.`**`exists`**(*name, region=None, key=None, keyid=None, profile=None*)

Check to see if a queue exists.

CLI Example:

```
salt myminion boto_sqs.exists myqueue region=us-east-1
```

`salt.modules.boto_sqs.`**`get_attributes`**(*name, region=None, key=None, keyid=None, profile=None*)

Return attributes currently set on an SQS queue.

CLI Example:

```
salt myminion boto_sqs.get_attributes myqueue
```

`salt.modules.boto_sqs.`**`list_`**(*prefix=", region=None, key=None, keyid=None, profile=None*)

Return a list of the names of all visible queues.

New in version 2016.11.0.

CLI Example:

```
salt myminion boto_sqs.list region=us-east-1
```

`salt.modules.boto_sqs.`**`set_attributes`**`(`*name*, *attributes*, *region=None*, *key=None*, *keyid=None*, *profile=None*`)`

Set attributes on an SQS queue.

CLI Example:

```
salt myminion boto_sqs.set_attributes myqueue '{ReceiveMessageWaitTimeSeconds: 20}
↪' region=us-east-1
```

21.9.63 salt.modules.boto_vpc

Connection module for Amazon VPC

New in version 2014.7.0.

depends

- boto >= 2.8.0
- boto3 >= 1.2.6

configuration This module accepts explicit VPC credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More Information available here.

If IAM roles are not used you need to specify them either in a pillar or in the minion's config file:

```
vpc.keyid: GKTADJGHEIQSXMKKRBJ08H
vpc.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

A region may also be specified in the configuration:

```
vpc.region: us-east-1
```

If a region is not specified, the default is us-east-1.

It's also possible to specify key, keyid and region via a profile, either as a passed in dict, or as a string to pull from pillars or minion config:

```
myprofile:
    keyid: GKTADJGHEIQSXMKKRBJ08H
    key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
    region: us-east-1
```

Changed in version 2015.8.0: All methods now return a dictionary. Create and delete methods return:

```
created: true
```

or

```
created: false
error:
  message: error message
```

Request methods (e.g., *describe_vpc*) return:

```
vpcs:
  - {...}
  - {...}
```

or

```
error:
  message: error message
```

New in version 2016.11.0.

Functions to request, accept, delete and describe VPC peering connections. Named VPC peering connections can be requested using these modules. VPC owner accounts can accept VPC peering connections (named or otherwise).

Examples showing creation of VPC peering connection

```
# Create a named VPC peering connection
salt myminion boto_vpc.request_vpc_peering_connection vpc-4a3e622e vpc-be82e9da
↪name=my_vpc_connection
# Without a name
salt myminion boto_vpc.request_vpc_peering_connection vpc-4a3e622e vpc-be82e9da
# Specify a region
salt myminion boto_vpc.request_vpc_peering_connection vpc-4a3e622e vpc-be82e9da
↪region=us-west-2
```

Check to see if VPC peering connection is pending

```
salt myminion boto_vpc.is_peering_connection_pending name=salt-vpc
# Specify a region
salt myminion boto_vpc.is_peering_connection_pending name=salt-vpc region=us-west-2
# specify an id
salt myminion boto_vpc.is_peering_connection_pending conn_id=pcx-8a8939e3
```

Accept VPC peering connection

```
salt myminion boto_vpc.accept_vpc_peering_connection name=salt-vpc
# Specify a region
salt myminion boto_vpc.accept_vpc_peering_connection name=salt-vpc region=us-west-2
# specify an id
salt myminion boto_vpc.accept_vpc_peering_connection conn_id=pcx-8a8939e3
```

Deleting VPC peering connection via this module

```
# Delete a named VPC peering connection
salt myminion boto_vpc.delete_vpc_peering_connection name=salt-vpc
# Specify a region
salt myminion boto_vpc.delete_vpc_peering_connection name=salt-vpc region=us-west-2
# specify an id
salt myminion boto_vpc.delete_vpc_peering_connection conn_id=pcx-8a8939e3
```

`salt.modules.boto_vpc.`**`accept_vpc_peering_connection`**(*conn_id=”, name=”, region=None, key=None, keyid=None, profile=None, dry_run=False*)

Request a VPC peering connection between two VPCs.

New in version 2016.11.0.

Parameters

- **conn_id** – The ID to use. String type.
- **name** – The name of this VPC peering connection. String type.
- **region** – The AWS region to use. Type string.

- **key** – The key to use for this connection. Type string.
- **keyid** – The key id to use.
- **profile** – The profile to use.
- **dry_run** – The dry_run flag to set.

Returns dict

Warning: Please specify either the `vpc_peering_connection_id` or `name` but not both. Specifying both will result in an error!

CLI Example:

```
salt myminion boto_vpc.accept_vpc_peering_connection name=salt-vpc
# Specify a region
salt myminion boto_vpc.accept_vpc_peering_connection name=salt-vpc region=us-west-
↪2
# specify an id
salt myminion boto_vpc.accept_vpc_peering_connection conn_id=pcx-8a8939e3
```

`salt.modules.boto_vpc.`**`associate_dhcp_options_to_vpc`**(*dhcp_options_id, vpc_id=None, vpc_name=None, region=None, key=None, keyid=None, profile=None*)

Given valid DHCP options id and a valid VPC id, associate the DHCP options record with the VPC.

Returns True if the DHCP options record were associated and returns False if the DHCP options record was not associated.

CLI Example:

```
salt myminion boto_vpc.associate_dhcp_options_to_vpc 'dhcp-a0bl34pp' 'vpc-6b1fe402
↪'
```

`salt.modules.boto_vpc.`**`associate_network_acl_to_subnet`**(*network_acl_id=None, subnet_id=None, network_acl_name=None, subnet_name=None, region=None, key=None, keyid=None, profile=None*)

Given a network acl and subnet ids or names, associate a network acl to a subnet.

CLI Example:

```
salt myminion boto_vpc.associate_network_acl_to_subnet \
        network_acl_id='acl-5fb85d36' subnet_id='subnet-6a1fe403'
```

```
salt myminion boto_vpc.associate_network_acl_to_subnet \
        network_acl_id='myacl' subnet_id='mysubnet'
```

`salt.modules.boto_vpc.`**`associate_route_table`**(*route_table_id=None, subnet_id=None, route_table_name=None, subnet_name=None, region=None, key=None, keyid=None, profile=None*)

Given a route table and subnet name or id, associates the route table with the subnet.

CLI Example:

```
salt myminion boto_vpc.associate_route_table 'rtb-1f382e7d' 'subnet-6a1fe403'
```

```
salt myminion boto_vpc.associate_route_table route_table_name='myrtb' \
        subnet_name='mysubnet'
```

`salt.modules.boto_vpc.`**`check_vpc`**(*vpc_id=None, vpc_name=None, region=None, key=None, keyid=None, profile=None*)

Check whether a VPC with the given name or id exists. Returns the vpc_id or None. Raises SaltInvocationError if both vpc_id and vpc_name are None. Optionally raise a CommandExecutionError if the VPC does not exist.

New in version 2016.3.0.

CLI Example:

```
salt myminion boto_vpc.check_vpc vpc_name=myvpc profile=awsprofile
```

`salt.modules.boto_vpc.`**`create`**(*cidr_block, instance_tenancy=None, vpc_name=None, enable_dns_support=None, enable_dns_hostnames=None, tags=None, region=None, key=None, keyid=None, profile=None*)

Given a valid CIDR block, create a VPC.

An optional instance_tenancy argument can be provided. If provided, the valid values are 'default' or 'dedicated'

An optional vpc_name argument can be provided.

Returns {created: true} if the VPC was created and returns {created: False} if the VPC was not created.

CLI Example:

```
salt myminion boto_vpc.create '10.0.0.0/24'
```

`salt.modules.boto_vpc.`**`create_customer_gateway`**(*vpn_connection_type, ip_address, bgp_asn, customer_gateway_name=None, tags=None, region=None, key=None, keyid=None, profile=None*)

Given a valid VPN connection type, a static IP address and a customer gateway's Border Gateway Protocol (BGP) Autonomous System Number, create a customer gateway.

Returns the customer gateway id if the customer gateway was created and returns False if the customer gateway was not created.

CLI Example:

```
salt myminion boto_vpc.create_customer_gateway 'ipsec.1', '12.1.2.3', 65534
```

`salt.modules.boto_vpc.`**`create_dhcp_options`**(*domain_name=None, domain_name_servers=None, ntp_servers=None, netbios_name_servers=None, netbios_node_type=None, dhcp_options_name=None, tags=None, vpc_id=None, vpc_name=None, region=None, key=None, keyid=None, profile=None*)

Given valid DHCP options, create a DHCP options record, optionally associating it with an existing VPC.

Returns True if the DHCP options record was created and returns False if the DHCP options record was not deleted.

Changed in version 2015.8.0: Added vpc_name and vpc_id arguments

CLI Example:

```
salt myminion boto_vpc.create_dhcp_options domain_name='example.com' \
        domain_name_servers='[1.2.3.4]' ntp_servers='[5.6.7.8]' \
        netbios_name_servers='[10.0.0.1]' netbios_node_type=1 \
        vpc_name='myvpc'
```

`salt.modules.boto_vpc.`**`create_internet_gateway`**(*internet_gateway_name=None, vpc_id=None, vpc_name=None, tags=None, region=None, key=None, keyid=None, profile=None*)

Create an Internet Gateway, optionally attaching it to an existing VPC.

Returns the internet gateway id if the internet gateway was created and returns False if the internet gateways was not created.

New in version 2015.8.0.

CLI Example:

```
salt myminion boto_vpc.create_internet_gateway \
        internet_gateway_name=myigw vpc_name=myvpc
```

`salt.modules.boto_vpc.`**`create_nat_gateway`**(*subnet_id=None, subnet_name=None, allocation_id=None, region=None, key=None, keyid=None, profile=None*)

Create a NAT Gateway within an existing subnet. If allocation_id is specified, the elastic IP address it references is associated with the gateway. Otherwise, a new allocation_id is created and used.

This function requires boto3 to be installed.

Returns the nat gateway id if the nat gateway was created and returns False if the nat gateway was not created.

New in version 2016.11.0.

CLI Example:

```
salt myminion boto_vpc.create_nat_gateway subnet_name=mysubnet
```

`salt.modules.boto_vpc.`**`create_network_acl`**(*vpc_id=None, vpc_name=None, network_acl_name=None, subnet_id=None, subnet_name=None, tags=None, region=None, key=None, keyid=None, profile=None*)

Given a vpc_id, creates a network acl.

Returns the network acl id if successful, otherwise returns False.

Changed in version 2015.8.0: Added vpc_name, subnet_id, and subnet_name arguments

CLI Example:

```
salt myminion boto_vpc.create_network_acl 'vpc-6b1fe402'
```

`salt.modules.boto_vpc.`**`create_network_acl_entry`**(*network_acl_id=None, rule_number=None, protocol=None, rule_action=None, cidr_block=None, egress=None, network_acl_name=None, icmp_code=None, icmp_type=None, port_range_from=None, port_range_to=None, region=None, key=None, keyid=None, profile=None*)

Creates a network acl entry.

CLI Example:

```
salt myminion boto_vpc.create_network_acl_entry 'acl-5fb85d36' '32767' \
        'all' 'deny' '0.0.0.0/0' egress=true
```

`salt.modules.boto_vpc.`**`create_route`**(*route_table_id=None, destination_cidr_block=None, route_table_name=None, gateway_id=None, internet_gateway_name=None, instance_id=None, interface_id=None, vpc_peering_connection_id=None, vpc_peering_connection_name=None, region=None, key=None, keyid=None, profile=None, nat_gateway_id=None, nat_gateway_subnet_name=None, nat_gateway_subnet_id=None*)

Creates a route.

If a nat gateway is specified, boto3 must be installed

CLI Example:

```
salt myminion boto_vpc.create_route 'rtb-1f382e7d' '10.0.0.0/16' gateway_id='vgw-
↪a1b2c3'
```

`salt.modules.boto_vpc.`**`create_route_table`**(*vpc_id=None, vpc_name=None, route_table_name=None, tags=None, region=None, key=None, keyid=None, profile=None*)

Creates a route table.

Changed in version 2015.8.0: Added vpc_name argument

CLI Examples:

```
salt myminion boto_vpc.create_route_table vpc_id='vpc-6b1fe402' \
        route_table_name='myroutetable'
salt myminion boto_vpc.create_route_table vpc_name='myvpc' \
        route_table_name='myroutetable'
```

`salt.modules.boto_vpc.`**`create_subnet`**(*vpc_id=None, cidr_block=None, vpc_name=None, availability_zone=None, subnet_name=None, tags=None, region=None, key=None, keyid=None, profile=None, auto_assign_public_ipv4=False*)

Given a valid VPC ID or Name and a CIDR block, create a subnet for the VPC.

An optional availability zone argument can be provided.

Returns True if the VPC subnet was created and returns False if the VPC subnet was not created.

Changed in version 2015.8.0: Added vpc_name argument

CLI Examples:

```
salt myminion boto_vpc.create_subnet vpc_id='vpc-6b1fe402' \
        subnet_name='mysubnet' cidr_block='10.0.0.0/25'
salt myminion boto_vpc.create_subnet vpc_name='myvpc' \
        subnet_name='mysubnet', cidr_block='10.0.0.0/25'
```

`salt.modules.boto_vpc.`**`customer_gateway_exists`**(*customer_gateway_id=None, customer_gateway_name=None, region=None, key=None, keyid=None, profile=None*)

Given a customer gateway ID, check if the customer gateway ID exists.

Returns True if the customer gateway ID exists; Returns False otherwise.

CLI Example:

```
salt myminion boto_vpc.customer_gateway_exists cgw-b6a247df
salt myminion boto_vpc.customer_gateway_exists customer_gatway_name=mycgw
```

`salt.modules.boto_vpc.`**`delete`**(*vpc_id=None, name=None, vpc_name=None, tags=None, region=None, key=None, keyid=None, profile=None*)

Given a VPC ID or VPC name, delete the VPC.

Returns {deleted: true} if the VPC was deleted and returns {deleted: false} if the VPC was not deleted.

CLI Example:

```
salt myminion boto_vpc.delete vpc_id='vpc-6b1fe402'
salt myminion boto_vpc.delete name='myvpc'
```

`salt.modules.boto_vpc.`**`delete_customer_gateway`**(*customer_gateway_id=None, customer_gateway_name=None, region=None, key=None, keyid=None, profile=None*)

Given a customer gateway ID or name, delete the customer gateway.

Returns True if the customer gateway was deleted and returns False if the customer gateway was not deleted.

Changed in version 2015.8.0: Added customer_gateway_name argument

CLI Example:

```
salt myminion boto_vpc.delete_customer_gateway 'cgw-b6a247df'
```

`salt.modules.boto_vpc.`**`delete_dhcp_options`**(*dhcp_options_id=None, dhcp_options_name=None, region=None, key=None, keyid=None, profile=None*)

Delete dhcp options by id or name.

New in version 2015.8.0.

CLI Example:

```
salt mymininon boto_vpc.delete_dhcp_options 'dopt-b6a247df'
```

`salt.modules.boto_vpc.`**`delete_internet_gateway`**(*internet_gateway_id=None, internet_gateway_name=None, detach=False, region=None, key=None, keyid=None, profile=None*)

Delete an internet gateway (by name or id).

Returns True if the internet gateway was deleted and otherwise False.

New in version 2015.8.0.

CLI Examples:

```
salt myminion boto_vpc.delete_internet_gateway internet_gateway_id=igw-1a2b3c
salt myminion boto_vpc.delete_internet_gateway internet_gateway_name=myigw
```

`salt.modules.boto_vpc.`**`delete_nat_gateway`**(*nat_gateway_id, release_eips=False, region=None, key=None, keyid=None, profile=None, wait_for_delete=False, wait_for_delete_retries=5*)

Delete a nat gateway (by id).

Returns True if the internet gateway was deleted and otherwise False.

This function requires boto3 to be installed.

New in version 2016.11.0.

nat_gateway_id Id of the NAT Gateway

releaes_eips whether to release the elastic IPs associated with the given NAT Gateway Id

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

wait_for_delete whether to wait for delete of the NAT gateway to be in failed or deleted state after issuing the delete call.

wait_for_delete_retries NAT gateway may take some time to be go into deleted or failed state. During the deletion process, subsequent release of elastic IPs may fail; this state will automatically retry this number of times to ensure the NAT gateway is in deleted or failed state before proceeding.

CLI Examples:

```
salt myminion boto_vpc.delete_nat_gateway nat_gateway_id=igw-1a2b3c
```

`salt.modules.boto_vpc.`**`delete_network_acl`**(*network_acl_id=None, network_acl_name=None, disassociate=False, region=None, key=None, keyid=None, profile=None*)

Delete a network acl based on the network_acl_id or network_acl_name provided.

CLI Examples:

```
salt myminion boto_vpc.delete_network_acl network_acl_id='acl-5fb85d36' \
        disassociate=false
```

```
salt myminion boto_vpc.delete_network_acl network_acl_name='myacl' \
        disassociate=true
```

`salt.modules.boto_vpc.`**`delete_network_acl_entry`**(*network_acl_id=None, rule_number=None, egress=None, network_acl_name=None, region=None, key=None, keyid=None, profile=None*)

Deletes a network acl entry.

CLI Example:

```
salt myminion boto_vpc.delete_network_acl_entry 'acl-5fb85d36' '32767'
```

`salt.modules.boto_vpc.`**`delete_route`**(*route_table_id=None, destination_cidr_block=None, route_table_name=None, region=None, key=None, keyid=None, profile=None*)

Deletes a route.

CLI Example:

```
salt myminion boto_vpc.delete_route 'rtb-1f382e7d' '10.0.0.0/16'
```

`salt.modules.boto_vpc.`**`delete_route_table`**(*route_table_id=None, route_table_name=None, region=None, key=None, keyid=None, profile=None*)

Deletes a route table.

CLI Examples:

```
salt myminion boto_vpc.delete_route_table route_table_id='rtb-1f382e7d'
salt myminion boto_vpc.delete_route_table route_table_name='myroutetable'
```

`salt.modules.boto_vpc.`**`delete_subnet`**(*subnet_id=None, subnet_name=None, region=None, key=None, keyid=None, profile=None*)

Given a subnet ID or name, delete the subnet.

Returns True if the subnet was deleted and returns False if the subnet was not deleted.

Changed in version 2015.8.0: Added subnet_name argument

CLI Example:

```
salt myminion boto_vpc.delete_subnet 'subnet-6a1fe403'
```

`salt.modules.boto_vpc.`**`delete_vpc_peering_connection`**(*conn_id=None, conn_name=None, region=None, key=None, keyid=None, profile=None, dry_run=False*)

Delete a VPC peering connection.

New in version 2016.11.0.

conn_id The connection ID to check. Exclusive with conn_name.

conn_name The connection name to check. Exclusive with conn_id.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

dry_run If True, skip application and simply return projected status.

CLI Example:

```
# Create a named VPC peering connection
salt myminion boto_vpc.delete_vpc_peering_connection conn_name=salt-vpc
# Specify a region
salt myminion boto_vpc.delete_vpc_peering_connection conn_name=salt-vpc region=us-
↪west-2
# specify an id
salt myminion boto_vpc.delete_vpc_peering_connection conn_id=pcx-8a8939e3
```

`salt.modules.boto_vpc.`**`describe`**(*vpc_id=None, vpc_name=None, region=None, key=None, keyid=None, profile=None*)

Given a VPC ID describe its properties.

Returns a dictionary of interesting properties.

Changed in version 2015.8.0: Added vpc_name argument

CLI Example:

```
salt myminion boto_vpc.describe vpc_id=vpc-123456
salt myminion boto_vpc.describe vpc_name=myvpc
```

`salt.modules.boto_vpc.`**`describe_nat_gateways`**(*nat_gateway_id=None, subnet_id=None, subnet_name=None, vpc_id=None, vpc_name=None, states=('pending', 'available'), region=None, key=None, keyid=None, profile=None*)

Return a description of nat gateways matching the selection criteria

This function requires boto3 to be installed.

CLI Example:

```
salt myminion boto_vpc.describe_nat_gateways nat_gateway_id='nat-03b02643b43216fe7
↪'
salt myminion boto_vpc.describe_nat_gateways subnet_id='subnet-5b05942d'
```

`salt.modules.boto_vpc.`**`describe_route_table`**(*route_table_id=None, route_table_name=None, tags=None, region=None, key=None, keyid=None, profile=None*)

Given route table properties, return route table details if matching table(s) exist.

New in version 2015.8.0.

CLI Example:

```
salt myminion boto_vpc.describe_route_table route_table_id='rtb-1f382e7d'
```

`salt.modules.boto_vpc.`**`describe_route_tables`**(*route_table_id=None, route_table_name=None, vpc_id=None, tags=None, region=None, key=None, keyid=None, profile=None*)

Given route table properties, return details of all matching route tables.

This function requires boto3 to be installed.

New in version 2016.11.0.

CLI Example:

```
salt myminion boto_vpc.describe_route_tables vpc_id='vpc-a6a9efc3'
```

`salt.modules.boto_vpc.`**`describe_subnet`**(*subnet_id=None, subnet_name=None, region=None, key=None, keyid=None, profile=None*)

Given a subnet id or name, describe its properties.

Returns a dictionary of interesting properties.

New in version 2015.8.0.

CLI Examples:

```
salt myminion boto_vpc.describe_subnet subnet_id=subnet-123456
salt myminion boto_vpc.describe_subnet subnet_name=mysubnet
```

`salt.modules.boto_vpc.`**`describe_subnets`**`(`*subnet_ids=None, subnet_names=None, vpc_id=None, cidr=None, region=None, key=None, keyid=None, profile=None*`)`

Given a VPC ID or subnet CIDR, returns a list of associated subnets and their details. Return all subnets if VPC ID or CIDR are not provided. If a subnet id or CIDR is provided, only its associated subnet details will be returned.

New in version 2015.8.0.

CLI Examples:

```
salt myminion boto_vpc.describe_subnets
```

```
salt myminion boto_vpc.describe_subnets subnet_ids=['subnet-ba1987ab', 'subnet-
↪ba1987cd']
```

```
salt myminion boto_vpc.describe_subnets vpc_id=vpc-123456
```

```
salt myminion boto_vpc.describe_subnets cidr=10.0.0.0/21
```

`salt.modules.boto_vpc.`**`describe_vpc_peering_connection`**`(`*name, region=None, key=None, keyid=None, profile=None*`)`

Returns any VPC peering connection id(s) for the given VPC peering connection name.

VPC peering connection ids are only returned for connections that are in the `active`, `pending-acceptance` or `provisioning` state.

New in version 2016.11.0.

Parameters

- **`name`** – The string name for this VPC peering connection
- **`region`** – The aws region to use
- **`key`** – Your aws key
- **`keyid`** – The key id associated with this aws account
- **`profile`** – The profile to use

Returns dict

CLI Example:

```
salt myminion boto_vpc.describe_vpc_peering_connection salt-vpc
# Specify a region
salt myminion boto_vpc.describe_vpc_peering_connection salt-vpc region=us-west-2
```

`salt.modules.boto_vpc.`**`describe_vpcs`**`(`*vpc_id=None, name=None, cidr=None, tags=None, region=None, key=None, keyid=None, profile=None*`)`

Describe all VPCs, matching the filter criteria if provided.

Returns a list of dictionaries with interesting properties.

New in version 2015.8.0.

CLI Example:

```
salt myminion boto_vpc.describe_vpcs
```

`salt.modules.boto_vpc.`**`dhcp_options_exists`**(*dhcp_options_id=None, name=None, dhcp_options_name=None, tags=None, region=None, key=None, keyid=None, profile=None*)

Check if a dhcp option exists.

Returns True if the dhcp option exists; Returns False otherwise.

CLI Example:

```
salt myminion boto_vpc.dhcp_options_exists dhcp_options_id='dhcp-a0bl34pp'
```

`salt.modules.boto_vpc.`**`disassociate_network_acl`**(*subnet_id=None, vpc_id=None, subnet_name=None, vpc_name=None, region=None, key=None, keyid=None, profile=None*)

Given a subnet ID, disassociates a network acl.

CLI Example:

```
salt myminion boto_vpc.disassociate_network_acl 'subnet-6a1fe403'
```

`salt.modules.boto_vpc.`**`disassociate_route_table`**(*association_id, region=None, key=None, keyid=None, profile=None*)

Dissassociates a route table.

association_id The Route Table Association ID to disassociate

CLI Example:

```
salt myminion boto_vpc.disassociate_route_table 'rtbassoc-d8ccddba'
```

`salt.modules.boto_vpc.`**`exists`**(*vpc_id=None, name=None, cidr=None, tags=None, region=None, key=None, keyid=None, profile=None*)

Given a VPC ID, check to see if the given VPC ID exists.

Returns True if the given VPC ID exists and returns False if the given VPC ID does not exist.

CLI Example:

```
salt myminion boto_vpc.exists myvpc
```

`salt.modules.boto_vpc.`**`get_dhcp_options`**(*dhcp_options_name=None, dhcp_options_id=None, region=None, key=None, keyid=None, profile=None*)

Return a dict with the current values of the requested DHCP options set

CLI Example:

```
salt myminion boto_vpc.get_dhcp_options 'myfunnydhcpoptionsname'
```

New in version 2016.3.0.

`salt.modules.boto_vpc.`**`get_id`**(*name=None, cidr=None, tags=None, region=None, key=None, keyid=None, profile=None*)

Given VPC properties, return the VPC id if a match is found.

CLI Example:

```
salt myminion boto_vpc.get_id myvpc
```

`salt.modules.boto_vpc.`**`get_resource_id`**(*resource, name=None, resource_id=None, region=None, key=None, keyid=None, profile=None*)

Get an AWS id for a VPC resource by type and name.

New in version 2015.8.0.

CLI Example:

```
salt myminion boto_vpc.get_resource_id internet_gateway myigw
```

`salt.modules.boto_vpc.`**`get_subnet_association`**(*subnets, region=None, key=None, keyid=None, profile=None*)

Given a subnet (aka: a vpc zone identifier) or list of subnets, returns vpc association.

Returns a VPC ID if the given subnets are associated with the same VPC ID. Returns False on an error or if the given subnets are associated with different VPC IDs.

CLI Examples:

```
salt myminion boto_vpc.get_subnet_association subnet-61b47516
```

```
salt myminion boto_vpc.get_subnet_association ['subnet-61b47516','subnet-2cb9785b
↪']
```

`salt.modules.boto_vpc.`**`is_peering_connection_pending`**(*conn_id=None, conn_name=None, region=None, key=None, keyid=None, profile=None*)

Check if a VPC peering connection is in the pending state.

New in version 2016.11.0.

conn_id The connection ID to check. Exclusive with conn_name.

conn_name The connection name to check. Exclusive with conn_id.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

CLI Example:

```
salt myminion boto_vpc.is_peering_connection_pending conn_name=salt-vpc
# Specify a region
salt myminion boto_vpc.is_peering_connection_pending conn_name=salt-vpc region=us-
↪west-2
# specify an id
salt myminion boto_vpc.is_peering_connection_pending conn_id=pcx-8a8939e3
```

`salt.modules.boto_vpc.`**`nat_gateway_exists`**(*nat_gateway_id=None, subnet_id=None, subnet_name=None, vpc_id=None, vpc_name=None, states=('pending', 'available'), region=None, key=None, keyid=None, profile=None*)

Checks if a nat gateway exists.

This function requires boto3 to be installed.

New in version 2016.11.0.

CLI Example:

```
salt myminion boto_vpc.nat_gateway_exists nat_gateway_id='nat-03b02643b43216fe7'
salt myminion boto_vpc.nat_gateway_exists subnet_id='subnet-5b05942d'
```

`salt.modules.boto_vpc.`**`network_acl_exists`**`(`*network_acl_id=None, name=None, network_acl_name=None, tags=None, region=None, key=None, keyid=None, profile=None*`)`

Checks if a network acl exists.

Returns True if the network acl exists or returns False if it doesn't exist.

CLI Example:

```
salt myminion boto_vpc.network_acl_exists network_acl_id='acl-5fb85d36'
```

`salt.modules.boto_vpc.`**`peering_connection_pending_from_vpc`**`(`*conn_id=None, conn_name=None, vpc_id=None, vpc_name=None, region=None, key=None, keyid=None, profile=None*`)`

Check if a VPC peering connection is in the pending state, and requested from the given VPC.

New in version 2016.11.0.

conn_id The connection ID to check. Exclusive with conn_name.

conn_name The connection name to check. Exclusive with conn_id.

vpc_id Is this the ID of the requesting VPC for this peering connection. Exclusive with vpc_name.

vpc_name Is this the Name of the requesting VPC for this peering connection. Exclusive with vpc_id.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

CLI Example:

```
salt myminion boto_vpc.is_peering_connection_pending name=salt-vpc
```

`salt.modules.boto_vpc.`**`replace_network_acl_entry`**`(`*network_acl_id=None, rule_number=None, protocol=None, rule_action=None, cidr_block=None, egress=None, network_acl_name=None, icmp_code=None, icmp_type=None, port_range_from=None, port_range_to=None, region=None, key=None, keyid=None, profile=None*`)`

Replaces a network acl entry.

CLI Example:

```
salt myminion boto_vpc.replace_network_acl_entry 'acl-5fb85d36' '32767' \
        'all' 'deny' '0.0.0.0/0' egress=true
```

`salt.modules.boto_vpc.`**`replace_route`**(*route_table_id=None, destination_cidr_block=None, route_table_name=None, gateway_id=None, instance_id=None, interface_id=None, region=None, key=None, keyid=None, profile=None, vpc_peering_connection_id=None*)

Replaces a route.

CLI Example:

```
salt myminion boto_vpc.replace_route 'rtb-1f382e7d' '10.0.0.0/16' gateway_id='vgw-
↪a1b2c3'
```

`salt.modules.boto_vpc.`**`replace_route_table_association`**(*association_id, route_table_id, region=None, key=None, keyid=None, profile=None*)

Replaces a route table association.

CLI Example:

```
salt myminion boto_vpc.replace_route_table_association 'rtbassoc-d8ccddba' 'rtb-
↪1f382e7d'
```

`salt.modules.boto_vpc.`**`request_vpc_peering_connection`**(*requester_vpc_id=None, requester_vpc_name=None, peer_vpc_id=None, peer_vpc_name=None, name=None, peer_owner_id=None, region=None, key=None, keyid=None, profile=None, dry_run=False*)

Request a VPC peering connection between two VPCs.

New in version 2016.11.0.

requester_vpc_id ID of the requesting VPC. Exclusive with requester_vpc_name.

requester_vpc_name Name tag of the requesting VPC. Exclusive with requester_vpc_id.

peer_vpc_id ID of the VPC to create VPC peering connection with. This can be a VPC in another account. Exclusive with peer_vpc_name.

peer_vpc_name Name tag of the VPC to create VPC peering connection with. This can only be a VPC in the same account, else resolving it into a vpc ID will almost certainly fail. Exclusive with peer_vpc_id.

name The name to use for this VPC peering connection.

peer_owner_id ID of the owner of the peer VPC. Defaults to your account ID, so a value is required if peering with a VPC in a different account.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

dry_run If True, skip application and return status.

CLI Example:

```
# Create a named VPC peering connection
salt myminion boto_vpc.request_vpc_peering_connection vpc-4a3e622e vpc-be82e9da␣
↪name=my_vpc_connection
# Without a name
salt myminion boto_vpc.request_vpc_peering_connection vpc-4a3e622e vpc-be82e9da
# Specify a region
salt myminion boto_vpc.request_vpc_peering_connection vpc-4a3e622e vpc-be82e9da␣
↪region=us-west-2
```

`salt.modules.boto_vpc.`**`resource_exists`**(*resource, name=None, resource_id=None, tags=None, region=None, key=None, keyid=None, profile=None*)

Given a resource type and name, return {exists: true} if it exists, {exists: false} if it does not exist, or {error: {message: error text} on error.

New in version 2015.8.0.

CLI Example:

```
salt myminion boto_vpc.resource_exists internet_gateway myigw
```

`salt.modules.boto_vpc.`**`route_exists`**(*destination_cidr_block, route_table_name=None, route_table_id=None, gateway_id=None, instance_id=None, interface_id=None, tags=None, region=None, key=None, keyid=None, profile=None, vpc_peering_connection_id=None*)

Checks if a route exists.

New in version 2015.8.0.

CLI Example:

```
salt myminion boto_vpc.route_exists destination_cidr_block='10.0.0.0/20' gateway_
↪id='local' route_table_name='test'
```

`salt.modules.boto_vpc.`**`route_table_exists`**(*route_table_id=None, name=None, route_table_name=None, tags=None, region=None, key=None, keyid=None, profile=None*)

Checks if a route table exists.

CLI Example:

```
salt myminion boto_vpc.route_table_exists route_table_id='rtb-1f382e7d'
```

`salt.modules.boto_vpc.`**`subnet_exists`**(*subnet_id=None, name=None, subnet_name=None, cidr=None, tags=None, zones=None, region=None, key=None, keyid=None, profile=None*)

Check if a subnet exists.

Returns True if the subnet exists, otherwise returns False.

Changed in version 2015.8.0: Added subnet_name argument Deprecated name argument

CLI Example:

```
salt myminion boto_vpc.subnet_exists subnet_id='subnet-6a1fe403'
```

21.9.64 salt.modules.bower

Manage and query Bower packages

This module manages the installed packages using Bower. Note that npm, git and bower must be installed for this module to be available.

`salt.modules.bower.`**`install`**(*pkg*, *dir*, *pkgs=None*, *runas=None*, *env=None*)

Install a Bower package.

If no package is specified, the dependencies (from bower.json) of the package in the given directory will be installed.

pkg A package name in any format accepted by Bower, including a version identifier

dir The target directory in which to install the package

pkgs A list of package names in the same format as the `pkg` parameter

runas The user to run Bower with

env Environment variables to set when invoking Bower. Uses the same `env` format as the *`cmd.run`* execution function.

CLI Example:

```
salt '*' bower.install underscore /path/to/project

salt '*' bower.install jquery#2.0 /path/to/project
```

`salt.modules.bower.`**`list_`**(*dir*, *runas=None*, *env=None*)

List installed Bower packages.

dir The directory whose packages will be listed

runas The user to run Bower with

env Environment variables to set when invoking Bower. Uses the same `env` format as the *`cmd.run`* execution function.

CLI Example:

```
salt '*' bower.list /path/to/project
```

`salt.modules.bower.`**`prune`**(*dir*, *runas=None*, *env=None*)

New in version 2017.7.0.

Remove extraneous local Bower packages, i.e. those not referenced in bower.json

dir The directory whose packages will be pruned

runas The user to run Bower with

env Environment variables to set when invoking Bower. Uses the same `env` format as the *`cmd.run`* execution function.

CLI Example:

```
salt '*' bower.prune /path/to/project
```

`salt.modules.bower.`**`uninstall`**(*pkg*, *dir*, *runas=None*, *env=None*)

Uninstall a Bower package.

pkg A package name in any format accepted by Bower

dir The target directory from which to uninstall the package

runas The user to run Bower with

env Environment variables to set when invoking Bower. Uses the same `env` format as the *cmd.run* execution function.

CLI Example:

```
salt '*' bower.uninstall underscore /path/to/project
```

21.9.65 salt.modules.bridge

Module for gathering and managing bridging information

`salt.modules.bridge.`**`add`**`(`*br=None*`)`

Creates a bridge

CLI Example:

```
salt '*' bridge.add br0
```

`salt.modules.bridge.`**`addif`**`(`*br=None, iface=None*`)`

Adds an interface to a bridge

CLI Example:

```
salt '*' bridge.addif br0 eth0
```

`salt.modules.bridge.`**`delete`**`(`*br=None*`)`

Deletes a bridge

CLI Example:

```
salt '*' bridge.delete br0
```

`salt.modules.bridge.`**`delif`**`(`*br=None, iface=None*`)`

Removes an interface from a bridge

CLI Example:

```
salt '*' bridge.delif br0 eth0
```

`salt.modules.bridge.`**`find_interfaces`**`(`**args*`)`

Returns the bridge to which the interfaces are bond to

CLI Example:

```
salt '*' bridge.find_interfaces eth0 [eth1...]
```

`salt.modules.bridge.`**`interfaces`**`(`*br=None*`)`

Returns interfaces attached to a bridge

CLI Example:

```
salt '*' bridge.interfaces br0
```

`salt.modules.bridge.`**`list_`**`()`

Returns the machine's bridges list

CLI Example:

```
salt '*' bridge.list
```

`salt.modules.bridge.`**`show`**`(`*br=None*`)`

Returns bridges interfaces along with enslaved physical interfaces. If no interface is given, all bridges are shown, else only the specified bridge values are returned.

CLI Example:

```
salt '*' bridge.show
salt '*' bridge.show br0
```

`salt.modules.bridge.`**`stp`**`(`*br=None, state='disable', iface=None*`)`

Sets Spanning Tree Protocol state for a bridge

CLI Example:

```
salt '*' bridge.stp br0 enable
salt '*' bridge.stp br0 disable
```

For BSD-like operating systems, it is required to add the interface on which to enable the STP.

CLI Example:

```
salt '*' bridge.stp bridge0 enable fxp0
salt '*' bridge.stp bridge0 disable fxp0
```

21.9.66 salt.modules.bsd_shadow

Manage the password database on BSD systems

Important: If you feel that Salt should be using this module to manage passwords on a minion, and it is using a different module (or gives an error similar to *'shadow.info' is not available*), see *here*.

`salt.modules.bsd_shadow.`**`default_hash`**`()`

Returns the default hash used for unset passwords

CLI Example:

```
salt '*' shadow.default_hash
```

`salt.modules.bsd_shadow.`**`del_password`**`(`*name*`)`

New in version 2015.8.2.

Delete the password from name user

CLI Example:

```
salt '*' shadow.del_password username
```

`salt.modules.bsd_shadow.`**`info`**`(`*name*`)`

Return information for the specified user

CLI Example:

```
salt '*' shadow.info someuser
```

salt.modules.bsd_shadow.**set_change**(*name*, *change*)

Sets the time at which the password expires (in seconds since the UNIX epoch). See `man 8 usermod` on NetBSD and OpenBSD or `man 8 pw` on FreeBSD.

A value of `0` sets the password to never expire.

CLI Example:

```
salt '*' shadow.set_change username 1419980400
```

salt.modules.bsd_shadow.**set_expire**(*name*, *expire*)

Sets the time at which the account expires (in seconds since the UNIX epoch). See `man 8 usermod` on NetBSD and OpenBSD or `man 8 pw` on FreeBSD.

A value of `0` sets the account to never expire.

CLI Example:

```
salt '*' shadow.set_expire username 1419980400
```

salt.modules.bsd_shadow.**set_password**(*name*, *password*)

Set the password for a named user. The password must be a properly defined hash. The password hash can be generated with this command:

`python -c "import crypt; print crypt.crypt('password', ciphersalt)"`

Note: When constructing the `ciphersalt` string, you must escape any dollar signs, to avoid them being interpolated by the shell.

`'password'` is, of course, the password for which you want to generate a hash.

`ciphersalt` is a combination of a cipher identifier, an optional number of rounds, and the cryptographic salt. The arrangement and format of these fields depends on the cipher and which flavor of BSD you are using. For more information on this, see the manpage for `crpyt(3)`. On NetBSD, additional information is available in `passwd.conf(5)`.

It is important to make sure that a supported cipher is used.

CLI Example:

```
salt '*' shadow.set_password someuser '$1$UYCIxa628.9qXjpQCjM4a..'
```

21.9.67 salt.modules.btrfs

Module for managing BTRFS file systems.

salt.modules.btrfs.**add**(*mountpoint*, **devices*, ***kwargs*)

Add a devices to a BTRFS filesystem.

General options:

- **nodiscard**: Do not perform whole device TRIM
- **force**: Force overwrite existing filesystem on the disk

CLI Example:

```
salt '*' btrfs.add /mountpoint /dev/sda1 /dev/sda2
```

`salt.modules.btrfs.`**`convert`**`(`*device*, *permanent=False*, *keeplf=False*`)`

Convert ext2/3/4 to BTRFS. Device should be mounted.

Filesystem can be converted temporarily so the further processing and rollback is possible, or permanently, where previous extended filesystem image gets deleted. Please note, permanent conversion takes a while as BTRFS filesystem needs to be properly rebalanced afterwards.

General options:

- **permanent**: Specify if the migration should be permanent (false by default)
- **keeplf**: **Keep `lost+found` of the partition (removed by default**, but still in the image, if not permanent migration)

CLI Example:

```
salt '*' btrfs.convert /dev/sda1
salt '*' btrfs.convert /dev/sda1 permanent=True
```

`salt.modules.btrfs.`**`defragment`**`(`*path*`)`

Defragment mounted BTRFS filesystem. In order to defragment a filesystem, device should be properly mounted and writable.

If passed a device name, then defragmented whole filesystem, mounted on in. If passed a moun tpoint of the filesystem, then only this mount point is defragmented.

CLI Example:

```
salt '*' btrfs.defragment /dev/sda1
salt '*' btrfs.defragment /path/on/filesystem
```

`salt.modules.btrfs.`**`delete`**`(`*mountpoint*, **devices*, ***kwargs*`)`

Remove devices from a BTRFS filesystem.

CLI Example:

```
salt '*' btrfs.delete /mountpoint /dev/sda1 /dev/sda2
```

`salt.modules.btrfs.`**`devices`**`()`

Get known BTRFS formatted devices on the system.

CLI Example:

```
salt '*' btrfs.devices
```

`salt.modules.btrfs.`**`features`**`()`

List currently available BTRFS features.

CLI Example:

```
salt '*' btrfs.mkfs_features
```

`salt.modules.btrfs.`**`info`**`(`*device*`)`

Get BTRFS filesystem information.

CLI Example:

```
salt '*' btrfs.info /dev/sda1
```

`salt.modules.btrfs.`**`mkfs`**`(`**devices*, ***kwargs*`)`

Create a file system on the specified device. By default wipes out with force.

General options:

- **allocsize**: Specify the BTRFS offset from the start of the device.
- **bytecount**: Specify the size of the resultant filesystem.
- **nodesize**: Node size.
- **leafsize**: Specify the nodesize, the tree block size in which btrfs stores data.
- **noforce**: Prevent force overwrite when an existing filesystem is detected on the device.
- **sectorsize**: Specify the sectorsize, the minimum data block allocation unit.
- **nodiscard**: Do not perform whole device TRIM operation by default.
- **uuid**: Pass UUID or pass True to generate one.

Options:

- **dto**: **(raid0|raid1|raid5|raid6|raid10|single|dup)** Specify how the data must be spanned across the devices specified.
- **mto**: **(raid0|raid1|raid5|raid6|raid10|single|dup)** Specify how metadata must be spanned across the devices specified.
- **fts**: Features (call `salt <host> btrfs.features` for full list of available features)

See the `mkfs.btrfs(8)` manpage for a more complete description of corresponding options description.

CLI Example:

```
salt '*' btrfs.mkfs /dev/sda1
salt '*' btrfs.mkfs /dev/sda1 noforce=True
```

`salt.modules.btrfs.`**`properties`**(*obj*, *type=None*, *set=None*)

List properties for given btrfs object. The object can be path of BTRFS device, mount point, or any directories/files inside the BTRFS filesystem.

General options:

- **type**: Possible types are s[ubvol], f[ilesystem], i[node] and d[evice].
- **force**: Force overwrite existing filesystem on the disk
- **set**: <key=value,key1=value1...> Options for a filesystem properties.

CLI Example:

```
salt '*' btrfs.properties /mountpoint
salt '*' btrfs.properties /dev/sda1 type=subvol set='ro=false,label="My Storage"'
```

`salt.modules.btrfs.`**`resize`**(*mountpoint*, *size*)

Resize filesystem.

General options:

- **mountpoint**: Specify the BTRFS mountpoint to resize.
- **size**: ([+/-]<newsize>[kKmMgGtTpPeE]|max) Specify the new size of the target.

CLI Example:

```
salt '*' btrfs.resize /mountpoint size=+1g
salt '*' btrfs.resize /dev/sda1 size=max
```

salt.modules.btrfs.**subvolume_create**(*name*, *dest=None*, *qgroupids=None*)

Create subvolume *name* in *dest*.

Return True if the subvolume is created, False is the subvolume is already there.

name Name of the new subvolume

dest If not given, the subvolume will be created in the current directory, if given will be in /dest/name

qgroupids Add the newly created subcolume to a qgroup. This parameter is a list

CLI Example:

```
salt '*' btrfs.subvolume_create var
salt '*' btrfs.subvolume_create var dest=/mnt
salt '*' btrfs.subvolume_create var qgroupids='[200]'
```

salt.modules.btrfs.**subvolume_delete**(*name=None*, *names=None*, *commit=None*)

Delete the subvolume(s) from the filesystem

The user can remove one single subvolume (name) or multiple of then at the same time (names). One of the two parameters needs to specified.

Please, refer to the documentation to understand the implication on the transactions, and when the subvolume is really deleted.

Return True if the subvolume is deleted, False is the subvolume was already missing.

name Name of the subvolume to remove

names List of names of subvolumes to remove

commit

- 'after': Wait for transaction commit at the end
- 'each': Wait for transaction commit after each delete

CLI Example:

```
salt '*' btrfs.subvolume_delete /var/volumes/tmp
salt '*' btrfs.subvolume_delete /var/volumes/tmp commit=after
```

salt.modules.btrfs.**subvolume_exists**(*path*)

Check if a subvolume is present in the filesystem.

path Mount point for the subvolume (full path)

CLI Example:

```
salt '*' btrfs.subvolume_exists /mnt/var
```

salt.modules.btrfs.**subvolume_find_new**(*name*, *last_gen*)

List the recently modified files in a subvolume

name Name of the subvolume

last_gen Last transid marker from where to compare

CLI Example:

```
salt '*' btrfs.subvolume_find_new /var/volumes/tmp 1024
```

salt.modules.btrfs.**subvolume_get_default**(*path*)

Get the default subvolume of the filesystem path

path Mount point for the subvolume

CLI Example:

```
salt '*' btrfs.subvolume_get_default /var/volumes/tmp
```

`salt.modules.btrfs.`**`subvolume_list`**(*path*, *parent_id=False*, *absolute=False*, *ogeneration=False*, *generation=False*, *subvolumes=False*, *uuid=False*, *parent_uuid=False*, *sent_subvolume_uuid=False*, *snapshots=False*, *readonly=False*, *deleted=False*, *generation_cmp=None*, *ogeneration_cmp=None*, *sort=None*)

List the subvolumes present in the filesystem.

path Mount point for the subvolume

parent_id Print parent ID

absolute Print all the subvolumes in the filesystem and distinguish between absolute and relative path with respect to the given <path>

ogeneration Print the ogeneration of the subvolume

generation Print the generation of the subvolume

subvolumes Print only subvolumes below specified <path>

uuid Print the UUID of the subvolume

parent_uuid Print the parent uuid of subvolumes (and snapshots)

sent_subvolume_uuid Print the UUID of the sent subvolume, where the subvolume is the result of a receive operation

snapshots Only snapshot subvolumes in the filesystem will be listed

readonly Only readonly subvolumes in the filesystem will be listed

deleted Only deleted subvolumens that are ye not cleaned

generation_cmp List subvolumes in the filesystem that its generation is >=, <= or = value. '+' means >= value, '-' means <= value, If there is neither '+' nor '-', it means = value

ogeneration_cmp List subvolumes in the filesystem that its ogeneration is >=, <= or = value

sort List subvolumes in order by specified items. Possible values: * rootid * gen * ogen * path You can add '+' or '-' in front of each items, '+' means ascending, '-' means descending. The default is ascending. You can combite it in a list.

CLI Example:

```
salt '*' btrfs.subvolume_list /var/volumes/tmp
salt '*' btrfs.subvolume_list /var/volumes/tmp path=True
salt '*' btrfs.subvolume_list /var/volumes/tmp sort='[-rootid]'
```

`salt.modules.btrfs.`**`subvolume_set_default`**(*subvolid*, *path*)

Set the subvolume as default

subvolid ID of the new default subvolume

path Mount point for the filesystem

CLI Example:

```
salt '*' btrfs.subvolume_set_default 257 /var/volumes/tmp
```

`salt.modules.btrfs.`**`subvolume_show`**`(`*path*`)`

Show information of a given subvolume

path Mount point for the filesystem

CLI Example:

```
salt '*' btrfs.subvolume_show /var/volumes/tmp
```

`salt.modules.btrfs.`**`subvolume_snapshot`**`(`*source, dest=None, name=None, read_only=False*`)`

Create a snapshot of a source subvolume

source Source subvolume from where to create the snapshot

dest If only dest is given, the subvolume will be named as the basename of the source

name Name of the snapshot

read_only Create a read only snapshot

CLI Example:

```
salt '*' btrfs.subvolume_snapshot /var/volumes/tmp dest=/.snapshots
salt '*' btrfs.subvolume_snapshot /var/volumes/tmp name=backup
```

`salt.modules.btrfs.`**`subvolume_sync`**`(`*path, subvolids=None, sleep=None*`)`

Wait until given subvolume are completely removed from the filesystem after deletion.

path Mount point for the filesystem

subvolids List of IDs of subvolumes to wait for

sleep Sleep N seconds betwenn checks (default: 1)

CLI Example:

```
salt '*' btrfs.subvolume_sync /var/volumes/tmp
salt '*' btrfs.subvolume_sync /var/volumes/tmp subvolids='[257]'
```

`salt.modules.btrfs.`**`usage`**`(`*path*`)`

Show in which disk the chunks are allocated.

CLI Example:

```
salt '*' btrfs.usage /your/mountpoint
```

`salt.modules.btrfs.`**`version`**`()`

Return BTRFS version.

CLI Example:

```
salt '*' btrfs.version
```

21.9.68 salt.modules.cabal

Manage and query Cabal packages

New in version 2015.8.0.

`salt.modules.cabal.`**`install`**`(`*pkg=None, pkgs=None, user=None, install_global=False, env=None*`)`

Install a cabal package.

pkg A package name in format accepted by cabal-install. See: https://wiki.haskell.org/Cabal-Install

pkgs A list of packages names in same format as `pkg`

user The user to run cabal install with

install_global Install package globally instead of locally

env Environment variables to set when invoking cabal. Uses the same `env` format as the *`cmd.run`* execution function

CLI Example:

```
salt '*' cabal.install shellcheck
salt '*' cabal.install shellcheck-0.3.5
```

`salt.modules.cabal.`**`list_`**(*pkg=None*, *user=None*, *installed=False*, *env=None*)

List packages matching a search string.

pkg Search string for matching package names

user The user to run cabal list with

installed If True, only return installed packages.

env Environment variables to set when invoking cabal. Uses the same `env` format as the *`cmd.run`* execution function

CLI example:

```
salt '*' cabal.list
salt '*' cabal.list ShellCheck
```

`salt.modules.cabal.`**`uninstall`**(*pkg*, *user=None*, *env=None*)

Uninstall a cabal package.

pkg The package to uninstall

user The user to run ghc-pkg unregister with

env Environment variables to set when invoking cabal. Uses the same `env` format as the *`cmd.run`* execution function

CLI Example:

```
salt '*' cabal.uninstall ShellCheck
```

`salt.modules.cabal.`**`update`**(*user=None*, *env=None*)

Updates list of known packages.

user The user to run cabal update with

env Environment variables to set when invoking cabal. Uses the same `env` format as the *`cmd.run`* execution function.

CLI Example:

```
salt '*' cabal.update
```

21.9.69 salt.modules.capirca_acl module

Capirca ACL

Generate ACL (firewall) configuration for network devices.

New in version 2017.7.0.

codeauthor Mircea Ulinic <ping@mirceaulinic.net> & Robert Ankeny <robankeny@google.com>

maturity new

depends capirca

platform unix

Dependencies

The firewall configuration is generated by Capirca.

To install Capirca, execute: `pip install capirca`.

`salt.modules.capirca_acl.`**`get_filter_config`**(*platform, filter_name, filter_options=None, terms=None, prepend=True, pillar_key='acl', pillarenv=None, saltenv=None, merge_pillar=True, only_lower_merge=False, revision_id=None, revision_no=None, revision_date=True, revision_date_format='%Y/%m/%d'*)

Return the configuration of a policy filter.

platform The name of the Capirca platform.

filter_name The name of the policy filter.

filter_options Additional filter options. These options are platform-specific. See the complete list of options.

terms List of terms for this policy filter. If not specified or empty, will try to load the configuration from the pillar, unless `merge_pillar` is set as `False`.

prepend: `True` When `merge_pillar` is set as `True`, the final list of terms generated by merging the terms from `terms` with those defined in the pillar (if any): new terms are prepended at the beginning, while existing ones will preserve the position. To add the new terms at the end of the list, set this argument to `False`.

pillar_key: `acl` The key in the pillar containing the default attributes values. Default: `acl`.

pillarenv Query the master to generate fresh pillar data on the fly, specifically from the requested pillar environment.

saltenv Included only for compatibility with *`pillarenv_from_saltenv`*, and is otherwise ignored.

merge_pillar: `True` Merge the CLI variables with the pillar. Default: `True`.

only_lower_merge: `False` Specify if it should merge only the terms fields. Otherwise it will try to merge also filters fields. Default: `False`.

revision_id Add a comment in the filter config having the description for the changes applied.

revision_no The revision count.

revision_date: `True` Boolean flag: display the date when the filter configuration was generated. Default: `True`.

revision_date_format: `%Y/%m/%d` The date format to be used when generating the perforce data. Default: `%Y/%m/%d` (<year>/<month>/<day>).

CLI Example:

```
salt '*' capirca.get_filter_config ciscoxr my-filter pillar_key=netacl
```

Output Example:

```
! $Id:$
! $Date:$
! $Revision:$
no ipv4 access-list my-filter
ipv4 access-list my-filter
 remark $Id:$
 remark my-term
 deny ipv4 any eq 1234 any
 deny ipv4 any eq 1235 any
 remark my-other-term
 permit tcp any range 5678 5680 any
exit
```

The filter configuration has been loaded from the pillar, having the following structure:

```
netacl:
  - my-filter:
      terms:
        - my-term:
            source_port: [1234, 1235]
            action: reject
        - my-other-term:
            source_port:
              - [5678, 5680]
            protocol: tcp
            action: accept
```

`salt.modules.capirca_acl.`**`get_filter_pillar`**(*filter_name, pillar_key='acl', pillarenv=None, saltenv=None*)

Helper that can be used inside a state SLS, in order to get the filter configuration given its name.

filter_name The name of the filter.

pillar_key The root key of the whole policy config.

pillarenv Query the master to generate fresh pillar data on the fly, specifically from the requested pillar environment.

saltenv Included only for compatibility with *`pillarenv_from_saltenv`*, and is otherwise ignored.

`salt.modules.capirca_acl.`**`get_policy_config`**(*platform, filters=None, prepend=True, pillar_key='acl', pillarenv=None, saltenv=None, merge_pillar=True, only_lower_merge=False, revision_id=None, revision_no=None, revision_date=True, revision_date_format='%Y/%m/%d'*)

Return the configuration of the whole policy.

platform The name of the Capirca platform.

filters List of filters for this policy. If not specified or empty, will try to load the configuration from the pillar, unless `merge_pillar` is set as `False`.

prepend: `True` When `merge_pillar` is set as `True`, the final list of filters generated by merging the filters from `filters` with those defined in the pillar (if any): new filters are prepended at the beginning, while existing ones will preserve the position. To add the new filters at the end of the list, set this argument to `False`.

pillar_key: `acl` The key in the pillar containing the default attributes values. Default: `acl`.

pillarenv Query the master to generate fresh pillar data on the fly, specifically from the requested pillar environment.

saltenv Included only for compatibility with *`pillarenv_from_saltenv`*, and is otherwise ignored.

merge_pillar: `True` Merge the CLI variables with the pillar. Default: `True`.

only_lower_merge: `False` Specify if it should merge only the filters and terms fields. Otherwise it will try to merge everything at the policy level. Default: `False`.

revision_id Add a comment in the policy config having the description for the changes applied.

revision_no The revision count.

revision_date: `True` Boolean flag: display the date when the policy configuration was generated. Default: `True`.

revision_date_format: `%Y/%m/%d` The date format to be used when generating the perforce data. Default: `%Y/%m/%d` (<year>/<month>/<day>).

CLI Example:

```
salt '*' capirca.get_policy_config juniper pillar_key=netacl
```

Output Example:

```
firewall {
    family inet {
        replace:
        /*
        ** $Id:$
        ** $Date:$
        ** $Revision:$
        **
        */
        filter my-filter {
            term my-term {
                from {
                    source-port [ 1234 1235 ];
                }
                then {
                    reject;
                }
            }
            term my-other-term {
                from {
                    protocol tcp;
                    source-port 5678-5680;
                }
                then accept;
            }
        }
    }
}
firewall {
    family inet {
```

(continues on next page)

(continued from previous page)

```
        replace:
        /*
        ** $Id:$
        ** $Date:$
        ** $Revision:$
        **
        */
        filter my-other-filter {
            interface-specific;
            term dummy-term {
                from {
                    protocol [ tcp udp ];
                }
                then {
                    reject;
                }
            }
        }
    }
}
```

The policy configuration has been loaded from the pillar, having the following structure:

```
netacl:
  - my-filter:
      options:
        - not-interface-specific
      terms:
        - my-term:
            source_port: [1234, 1235]
            action: reject
        - my-other-term:
            source_port:
              - [5678, 5680]
            protocol: tcp
            action: accept
  - my-other-filter:
      terms:
        - dummy-term:
            protocol:
              - tcp
              - udp
            action: reject
```

`salt.modules.capirca_acl.`**`get_term_config`**(*platform, filter_name, term_name, filter_options=None, pillar_key='acl', pillarenv=None, saltenv=None, merge_pillar=True, revision_id=None, revision_no=None, revision_date=True, revision_date_format='%Y/%m/%d', source_service=None, destination_service=None, **term_fields*)

Return the configuration of a single policy term.

platform The name of the Capirca platform.

filter_name The name of the policy filter.

term_name The name of the term.

filter_options Additional filter options. These options are platform-specific. E.g.: `inet6`, `bridge`, `object-group`, See the complete list of options.

pillar_key: **`acl`** The key in the pillar containing the default attributes values. Default: `acl`. If the pillar contains the following structure:

```
firewall:
  - my-filter:
      terms:
        - my-term:
            source_port: 1234
            source_address:
                - 1.2.3.4/32
                - 5.6.7.8/32
```

The `pillar_key` field would be specified as `firewall`.

pillarenv Query the master to generate fresh pillar data on the fly, specifically from the requested pillar environment.

saltenv Included only for compatibility with *`pillarenv_from_saltenv`*, and is otherwise ignored.

merge_pillar: **`True`** Merge the CLI variables with the pillar. Default: `True`.

revision_id Add a comment in the term config having the description for the changes applied.

revision_no The revision count.

revision_date: **`True`** Boolean flag: display the date when the term configuration was generated. Default: `True`.

revision_date_format: **`%Y/%m/%d`** The date format to be used when generating the perforce data. Default: `%Y/%m/%d` (<year>/<month>/<day>).

source_service A special service to choose from. This is a helper so the user is able to select a source just using the name, instead of specifying a source_port and protocol.

As this module is available on Unix platforms only, it reads the IANA port assignment from `/etc/services`.

If the user requires additional shortcuts to be referenced, they can add entries under `/etc/services`, which can be managed using the *`file state`*.

destination_service A special service to choose from. This is a helper so the user is able to select a source just using the name, instead of specifying a destination_port and protocol. Allows the same options as `source_service`.

term_fields Term attributes. To see what fields are supported, please consult the list of supported keywords. Some platforms have few other optional keywords.

Note: The following fields are accepted:

- action
- address
- address_exclude
- comment
- counter

- expiration
- destination_address
- destination_address_exclude
- destination_port
- destination_prefix
- forwarding_class
- forwarding_class_except
- logging
- log_name
- loss_priority
- option
- policer
- port
- precedence
- principals
- protocol
- protocol_except
- qos
- pan_application
- routing_instance
- source_address
- source_address_exclude
- source_port
- source_prefix
- verbatim
- packet_length
- fragment_offset
- hop_limit
- icmp_type
- ether_type
- traffic_class_count
- traffic_type
- translated
- dscp_set
- dscp_match
- dscp_except

- next_ip
- flexible_match_range
- source_prefix_except
- destination_prefix_except
- vpn
- source_tag
- destination_tag
- source_interface
- destination_interface
- flattened
- flattened_addr
- flattened_saddr
- flattened_daddr
- priority

Note: The following fields can be also a single value and a list of values:

- action
- address
- address_exclude
- comment
- destination_address
- destination_address_exclude
- destination_port
- destination_prefix
- forwarding_class
- forwarding_class_except
- logging
- option
- port
- precedence
- principals
- protocol
- protocol_except
- pan_application
- source_address
- source_address_exclude

- source_port
- source_prefix
- verbatim
- icmp_type
- ether_type
- traffic_type
- dscp_match
- dscp_except
- flexible_match_range
- source_prefix_except
- destination_prefix_except
- source_tag
- destination_tag
- source_service
- destination_service

Example: `destination_address` can be either defined as:

```
destination_address: 172.17.17.1/24
```

or as a list of destination IP addresses:

```
destination_address:
    - 172.17.17.1/24
    - 172.17.19.1/24
```

or a list of services to be matched:

```
source_service:
    - ntp
    - snmp
    - ldap
    - bgpd
```

Note: The port fields `source_port` and `destination_port` can be used as above to select either a single value, either a list of values, but also they can select port ranges. Example:

```
source_port:
    - [1000, 2000]
    - [3000, 4000]
```

With the configuration above, the user is able to select the 1000-2000 and 3000-4000 source port ranges.

CLI Example:

```
salt '*' capirca.get_term_config arista filter-name term-name source_address=1.2.
↪3.4 destination_address=5.6.7.8 action=accept
```

Output Example:

```
! $Date: 2017/03/22 $
no ip access-list filter-name
ip access-list filter-name
 remark term-name
 permit ip host 1.2.3.4 host 5.6.7.8
exit
```

`salt.modules.capirca_acl.`**`get_term_pillar`**(*filter_name, term_name, pillar_key='acl', pillarenv=None, saltenv=None*)

Helper that can be used inside a state SLS, in order to get the term configuration given its name, under a certain filter uniquely identified by its name.

filter_name The name of the filter.

term_name The name of the term.

pillar_key: **`acl`** The root key of the whole policy config. Default: `acl`.

pillarenv Query the master to generate fresh pillar data on the fly, specifically from the requested pillar environment.

saltenv Included only for compatibility with *`pillarenv_from_saltenv`*, and is otherwise ignored.

21.9.70 salt.modules.cassandra

Cassandra NoSQL Database Module

depends

- pycassa Cassandra Python adapter

configuration The location of the 'nodetool' command, host, and thrift port needs to be specified via pillar:

```
cassandra.nodetool: /usr/local/bin/nodetool
cassandra.host: localhost
cassandra.thrift_port: 9160
```

`salt.modules.cassandra.`**`column_families`**(*keyspace=None*)

Return existing column families for all keyspaces or just the provided one.

CLI Example:

```
salt '*' cassandra.column_families
salt '*' cassandra.column_families <keyspace>
```

`salt.modules.cassandra.`**`column_family_definition`**(*keyspace, column_family*)

Return a dictionary of column family definitions for the given keyspace/column_family

CLI Example:

```
salt '*' cassandra.column_family_definition <keyspace> <column_family>
```

`salt.modules.cassandra.`**`compactionstats`**()

Return compactionstats info

CLI Example:

```
salt '*' cassandra.compactionstats
```

salt.modules.cassandra.**info**()
Return cassandra node info

CLI Example:

```
salt '*' cassandra.info
```

salt.modules.cassandra.**keyspaces**()
Return existing keyspaces

CLI Example:

```
salt '*' cassandra.keyspaces
```

salt.modules.cassandra.**netstats**()
Return netstats info

CLI Example:

```
salt '*' cassandra.netstats
```

salt.modules.cassandra.**ring**()
Return cassandra ring info

CLI Example:

```
salt '*' cassandra.ring
```

salt.modules.cassandra.**tpstats**()
Return tpstats info

CLI Example:

```
salt '*' cassandra.tpstats
```

salt.modules.cassandra.**version**()
Return the cassandra version

CLI Example:

```
salt '*' cassandra.version
```

21.9.71 salt.modules.cassandra_cql

Cassandra Database Module

New in version 2015.5.0.

This module works with Cassandra v2 and v3 and hence generates queries based on the internal schema of said version.

depends DataStax Python Driver for Apache Cassandra https://github.com/datastax/python-driver pip install cassandra-driver

referenced by Salt's cassandra_cql returner

configuration The Cassandra cluster members and connection port can either be specified in the master or minion config, the minion's pillar or be passed to the module.

Example configuration in the config for a single node:

```
cassandra:
  cluster: 192.168.50.10
  port: 9000
```

Example configuration in the config for a cluster:

```
cassandra:
  cluster:
    - 192.168.50.10
    - 192.168.50.11
    - 192.168.50.12
  port: 9000
  username: cas_admin
```

Changed in version 2016.11.0.

Added support for `ssl_options` and `protocol_version`.

Example configuration with ssl options:

If `ssl_options` are present in cassandra config the cassandra_cql returner will use SSL. SSL isn't used if `ssl_options` isn't specified.

```
cassandra:
  cluster:
    - 192.168.50.10
    - 192.168.50.11
    - 192.168.50.12
  port: 9000
  username: cas_admin

  ssl_options:
    ca_certs: /etc/ssl/certs/ca-bundle.trust.crt

    # SSL version should be one from the ssl module
    # This is an optional parameter
    ssl_version: PROTOCOL_TLSv1
```

Additionally you can also specify the `protocol_version` to use.

```
cassandra:
  cluster:
    - 192.168.50.10
    - 192.168.50.11
    - 192.168.50.12
  port: 9000
  username: cas_admin

  # defaults to 4, if not set
  protocol_version: 3
```

`salt.modules.cassandra_cql.`**`cql_query`**(*query, contact_points=None, port=None, cql_user=None, cql_pass=None*)

Run a query on a Cassandra cluster and return a dictionary.

Parameters

- **query** (*str*) – The query to execute.
- **contact_points** (*str | list[str]*) – The Cassandra cluster addresses, can either be a string or a list of IPs.
- **cql_user** (*str*) – The Cassandra user if authentication is turned on.
- **cql_pass** (*str*) – The Cassandra user password if authentication is turned on.
- **port** (*int*) – The Cassandra cluster port, defaults to None.
- **params** (*str*) – The parameters for the query, optional.

Returns A dictionary from the return values of the query

Return type list[dict]

CLI Example:

```
salt 'cassandra-server' cassandra_cql.cql_query "SELECT * FROM users_by_name
↪WHERE first_name = 'jane'"
```

`salt.modules.cassandra_cql.`**`cql_query_with_prepare`**(*query, statement_name, statement_arguments, callback_errors=None, contact_points=None, port=None, cql_user=None, cql_pass=None, **kwargs*)

Run a query on a Cassandra cluster and return a dictionary.

This function should not be used asynchronously for SELECTs – it will not return anything and we don't currently have a mechanism for handling a future that will return results.

Parameters

- **query** (*str*) – The query to execute.
- **statement_name** (*str*) – Name to assign the prepared statement in the __context__ dictionary
- **statement_arguments** (*list[str]*) – Bind parameters for the SQL statement
- **async** (*bool*) – Run this query in asynchronous mode
- **callback_errors** (*Function callable*) – Function to call after query runs if there is an error
- **contact_points** (*str | list[str]*) – The Cassandra cluster addresses, can either be a string or a list of IPs.
- **cql_user** (*str*) – The Cassandra user if authentication is turned on.
- **cql_pass** (*str*) – The Cassandra user password if authentication is turned on.
- **port** (*int*) – The Cassandra cluster port, defaults to None.
- **params** (*str*) – The parameters for the query, optional.

Returns A dictionary from the return values of the query

Return type list[dict]

CLI Example:

```
# Insert data asynchronously
salt this-node cassandra_cql.cql_query_with_prepare "name_insert" "INSERT INTO␣
↪USERS (first_name, last_name) VALUES (?, ?)"            statement_arguments=[
↪'John','Doe'], asynchronous=True

# Select data, should not be asynchronous because there is not currently a␣
↪facility to return data from a future
salt this-node cassandra_cql.cql_query_with_prepare "name_select" "SELECT * FROM␣
↪USERS WHERE first_name=?"             statement_arguments=['John']
```

`salt.modules.cassandra_cql.`**`create_keyspace`**(*keyspace, replication_strategy='SimpleStrategy', replication_factor=1, replication_datacenters=None, contact_points=None, port=None, cql_user=None, cql_pass=None*)

Create a new keyspace in Cassandra.

Parameters

- **keyspace** (*str*) – The keyspace name
- **replication_strategy** (*str*) – either *SimpleStrategy* or *NetworkTopologyStrategy*
- **replication_factor** (*int*) – number of replicas of data on multiple nodes. not used if using NetworkTopologyStrategy
- **replication_datacenters** (*str | dict[str, int]*) – string or dict of datacenter names to replication factors, required if using NetworkTopologyStrategy (will be a dict if coming from state file).
- **contact_points** (*str | list[str]*) – The Cassandra cluster addresses, can either be a string or a list of IPs.
- **cql_user** (*str*) – The Cassandra user if authentication is turned on.
- **cql_pass** (*str*) – The Cassandra user password if authentication is turned on.
- **port** (*int*) – The Cassandra cluster port, defaults to None.

Returns The info for the keyspace or False if it does not exist.

Return type dict

CLI Example:

```
# CLI Example:
salt 'minion1' cassandra_cql.create_keyspace keyspace=newkeyspace

salt 'minion1' cassandra_cql.create_keyspace keyspace=newkeyspace replication_
↪strategy=NetworkTopologyStrategy          replication_datacenters='{"datacenter_1
↪": 3, "datacenter_2": 2}'
```

`salt.modules.cassandra_cql.`**`create_user`**(*username, password, superuser=False, contact_points=None, port=None, cql_user=None, cql_pass=None*)

Create a new cassandra user with credentials and superuser status.

Parameters

- **username** (*str*) – The name of the new user.
- **password** (*str*) – The password of the new user.

- **superuser** (*bool*) – Is the new user going to be a superuser? default: False
- **contact_points** (*str | list[str]*) – The Cassandra cluster addresses, can either be a string or a list of IPs.
- **cql_user** (*str*) – The Cassandra user if authentication is turned on.
- **cql_pass** (*str*) – The Cassandra user password if authentication is turned on.
- **port** (*int*) – The Cassandra cluster port, defaults to None.

Returns

Return type

CLI Example:

```
salt 'minion1' cassandra_cql.create_user username=joe password=secret

salt 'minion1' cassandra_cql.create_user username=joe password=secret␣
↪superuser=True

salt 'minion1' cassandra_cql.create_user username=joe password=secret␣
↪superuser=True contact_points=minion1
```

`salt.modules.cassandra_cql.`**`drop_keyspace`**(*keyspace, contact_points=None, port=None, cql_user=None, cql_pass=None*)

Drop a keyspace if it exists in a Cassandra cluster.

Parameters

- **keyspace** (*str*) – The keyspace to drop.
- **contact_points** (*str | list[str]*) – The Cassandra cluster addresses, can either be a string or a list of IPs.
- **cql_user** (*str*) – The Cassandra user if authentication is turned on.
- **cql_pass** (*str*) – The Cassandra user password if authentication is turned on.
- **port** (*int*) – The Cassandra cluster port, defaults to None.

Returns The info for the keyspace or False if it does not exist.

Return type dict

CLI Example:

```
salt 'minion1' cassandra_cql.drop_keyspace keyspace=test

salt 'minion1' cassandra_cql.drop_keyspace keyspace=test contact_points=minion1
```

`salt.modules.cassandra_cql.`**`grant_permission`**(*username, resource=None, resource_type='keyspace', permission=None, contact_points=None, port=None, cql_user=None, cql_pass=None*)

Grant permissions to a user.

Parameters

- **username** (*str*) – The name of the user to grant permissions to.
- **resource** (*str*) – The resource (keyspace or table), if None, permissions for all resources are granted.

- **resource_type** (*str*) – The resource_type (keyspace or table), defaults to 'keyspace'.
- **permission** (*str*) – A permission name (e.g. select), if None, all permissions are granted.
- **contact_points** (*str | list[str]*) – The Cassandra cluster addresses, can either be a string or a list of IPs.
- **cql_user** (*str*) – The Cassandra user if authentication is turned on.
- **cql_pass** (*str*) – The Cassandra user password if authentication is turned on.
- **port** (*int*) – The Cassandra cluster port, defaults to None.

Returns

Return type

CLI Example:

```
salt 'minion1' cassandra_cql.grant_permission

salt 'minion1' cassandra_cql.grant_permission username=joe resource=test_keyspace␣
↪permission=select

salt 'minion1' cassandra_cql.grant_permission username=joe resource=test_table␣
↪resource_type=table         permission=select contact_points=minion1
```

`salt.modules.cassandra_cql.`**`info`**(*contact_points=None, port=None, cql_user=None, cql_pass=None*)

Show the Cassandra information for this cluster.

Parameters

- **contact_points** (*str | list[str]*) – The Cassandra cluster addresses, can either be a string or a list of IPs.
- **cql_user** (*str*) – The Cassandra user if authentication is turned on.
- **cql_pass** (*str*) – The Cassandra user password if authentication is turned on.
- **port** (*int*) – The Cassandra cluster port, defaults to None.

Returns The information for this Cassandra cluster.

Return type dict

CLI Example:

```
salt 'minion1' cassandra_cql.info

salt 'minion1' cassandra_cql.info contact_points=minion1
```

`salt.modules.cassandra_cql.`**`keyspace_exists`**(*keyspace, contact_points=None, port=None, cql_user=None, cql_pass=None*)

Check if a keyspace exists in a Cassandra cluster.

:param keyspace The keyspace name to check for. :type keyspace: str :param contact_points: The Cassandra cluster addresses, can either be a string or a list of IPs. :type contact_points: str | list[str] :param cql_user: The Cassandra user if authentication is turned on. :type cql_user: str :param cql_pass: The Cassandra user password if authentication is turned on. :type cql_pass: str :param port: The Cassandra cluster port, defaults to None. :type port: int :return: The info for the keyspace or False if it does not exist. :rtype: dict

CLI Example:

```
salt 'minion1' cassandra_cql.keyspace_exists keyspace=system
```

`salt.modules.cassandra_cql.`**`list_column_families`**(*keyspace=None*, *contact_points=None*, *port=None*, *cql_user=None*, *cql_pass=None*)

List column families in a Cassandra cluster for all keyspaces or just the provided one.

Parameters

- **keyspace** (*str*) – The keyspace to provide the column families for, optional.
- **contact_points** (*str | list[str]*) – The Cassandra cluster addresses, can either be a string or a list of IPs.
- **cql_user** (*str*) – The Cassandra user if authentication is turned on.
- **cql_pass** (*str*) – The Cassandra user password if authentication is turned on.
- **port** (*int*) – The Cassandra cluster port, defaults to None.

Returns The column families in this Cassandra cluster.

Return type list[dict]

CLI Example:

```
salt 'minion1' cassandra_cql.list_column_families

salt 'minion1' cassandra_cql.list_column_families contact_points=minion1

salt 'minion1' cassandra_cql.list_column_families keyspace=system
```

`salt.modules.cassandra_cql.`**`list_keyspaces`**(*contact_points=None*, *port=None*, *cql_user=None*, *cql_pass=None*)

List keyspaces in a Cassandra cluster.

Parameters

- **contact_points** (*str | list[str]*) – The Cassandra cluster addresses, can either be a string or a list of IPs.
- **cql_user** (*str*) – The Cassandra user if authentication is turned on.
- **cql_pass** (*str*) – The Cassandra user password if authentication is turned on.
- **port** (*int*) – The Cassandra cluster port, defaults to None.

Returns The keyspaces in this Cassandra cluster.

Return type list[dict]

CLI Example:

```
salt 'minion1' cassandra_cql.list_keyspaces

salt 'minion1' cassandra_cql.list_keyspaces contact_points=minion1 port=9000
```

`salt.modules.cassandra_cql.`**`list_permissions`**(*username=None*, *resource=None*, *resource_type='keyspace'*, *permission=None*, *contact_points=None*, *port=None*, *cql_user=None*, *cql_pass=None*)

List permissions.

Parameters

- **username** (*str*) – The name of the user to list permissions for.
- **resource** (*str*) – The resource (keyspace or table), if None, permissions for all resources are listed.
- **resource_type** (*str*) – The resource_type (keyspace or table), defaults to 'keyspace'.
- **permission** (*str*) – A permission name (e.g. select), if None, all permissions are listed.
- **contact_points** (*str | list[str]*) – The Cassandra cluster addresses, can either be a string or a list of IPs.
- **cql_user** (*str*) – The Cassandra user if authentication is turned on.
- **cql_pass** (*str*) – The Cassandra user password if authentication is turned on.
- **port** (*int*) – The Cassandra cluster port, defaults to None.

Returns Dictionary of permissions.

Return type dict

CLI Example:

```
salt 'minion1' cassandra_cql.list_permissions

salt 'minion1' cassandra_cql.list_permissions username=joe resource=test_keyspace␣
↪permission=select

salt 'minion1' cassandra_cql.list_permissions username=joe resource=test_table␣
↪resource_type=table            permission=select contact_points=minion1
```

`salt.modules.cassandra_cql.`**`list_users`**(*contact_points=None, port=None, cql_user=None, cql_pass=None*)

List existing users in this Cassandra cluster.

Parameters

- **contact_points** (*str | list[str]*) – The Cassandra cluster addresses, can either be a string or a list of IPs.
- **port** (*int*) – The Cassandra cluster port, defaults to None.
- **cql_user** (*str*) – The Cassandra user if authentication is turned on.
- **cql_pass** (*str*) – The Cassandra user password if authentication is turned on.

Returns The list of existing users.

Return type dict

CLI Example:

```
salt 'minion1' cassandra_cql.list_users

salt 'minion1' cassandra_cql.list_users contact_points=minion1
```

`salt.modules.cassandra_cql.`**`version`**(*contact_points=None, port=None, cql_user=None, cql_pass=None*)

Show the Cassandra version.

Parameters

- **contact_points** (*str | list[str]*) – The Cassandra cluster addresses, can either be a string or a list of IPs.

- **cql_user** (*str*) – The Cassandra user if authentication is turned on.
- **cql_pass** (*str*) – The Cassandra user password if authentication is turned on.
- **port** (*int*) – The Cassandra cluster port, defaults to None.

Returns The version for this Cassandra cluster.

Return type str

CLI Example:

```
salt 'minion1' cassandra_cql.version

salt 'minion1' cassandra_cql.version contact_points=minion1
```

21.9.72 salt.modules.celery module

Support for scheduling celery tasks. The worker is independent of salt and thus can run in a different virtualenv or on a different python version, as long as broker, backend and serializer configurations match. Also note that celery and packages required by the celery broker, e.g. redis must be installed to load the salt celery execution module.

Note: A new app (and thus new connections) is created for each task execution

`salt.modules.celery.`**`run_task`**(*task_name, args=None, kwargs=None, broker=None, backend=None, wait_for_result=False, timeout=None, propagate=True, interval=0.5, no_ack=True, raise_timeout=True, config=None*)

Execute celery tasks. For celery specific parameters see celery documentation.

CLI Example:

```
salt '*' celery.run_task tasks.sleep args=[4] broker=redis://localhost \
backend=redis://localhost wait_for_result=true
```

task_name The task name, e.g. tasks.sleep

args Task arguments as a list

kwargs Task keyword arguments

broker Broker for celeryapp, see celery documentation

backend Result backend for celeryapp, see celery documentation

wait_for_result Wait until task result is read from result backend and return result, Default: False

timeout Timeout waiting for result from celery, see celery AsyncResult.get documentation

propagate Propagate exceptions from celery task, see celery AsyncResult.get documentation, Default: True

interval Interval to check for task result, see celery AsyncResult.get documentation, Default: 0.5

no_ack see celery AsyncResult.get documentation. Default: True

raise_timeout Raise timeout exception if waiting for task result times out. Default: False

config Config dict for celery app, See celery documentation

21.9.73 salt.modules.ceph module

Module to provide ceph control with salt.

depends

- ceph_cfg Python module

New in version 2016.11.0.

`salt.modules.ceph.`**`ceph_version`**`()`

Get the version of ceph installed

CLI Example:

```
salt '*' ceph.ceph_version
```

`salt.modules.ceph.`**`cluster_quorum`**`(`*`**kwargs`*`)`

Get the cluster's quorum status

CLI Example:

```
salt '*' ceph.cluster_quorum \
        'cluster_name'='ceph' \
        'cluster_uuid'='cluster_uuid'
```

cluster_uuid The cluster UUID. Defaults to value found in ceph config file.

cluster_name The cluster name. Defaults to `ceph`.

`salt.modules.ceph.`**`cluster_status`**`(`*`**kwargs`*`)`

Get the cluster status, including health if in quorum

CLI Example:

```
salt '*' ceph.cluster_status \
        'cluster_name'='ceph' \
        'cluster_uuid'='cluster_uuid'
```

cluster_uuid The cluster UUID. Defaults to value found in ceph config file.

cluster_name The cluster name. Defaults to `ceph`.

`salt.modules.ceph.`**`keyring_auth_add`**`(`*`**kwargs`*`)`

Add keyring to authorized list

CLI Example:

```
salt '*' ceph.keyring_auth_add \
        'keyring_type'='admin' \
        'cluster_name'='ceph' \
        'cluster_uuid'='cluster_uuid'
```

keyring_type (required) One of `admin`, `mon`, `osd`, `rgw`, `mds`

cluster_uuid The cluster UUID. Defaults to value found in ceph config file.

cluster_name The cluster name. Defaults to `ceph`.

`salt.modules.ceph.`**`keyring_auth_del`**`(`***kwargs*`)`

Remove keyring from authorised list

CLI Example:

```
salt '*' ceph.keyring_osd_auth_del \
        'keyring_type'='admin' \
        'cluster_name'='ceph' \
        'cluster_uuid'='cluster_uuid'
```

keyring_type (required) One of `admin`, `mon`, `osd`, `rgw`, `mds`

cluster_uuid The cluster UUID. Defaults to value found in ceph config file.

cluster_name The cluster name. Defaults to `ceph`.

`salt.modules.ceph.`**`keyring_auth_list`**`(`***kwargs*`)`

List all cephx authorization keys

CLI Example:

```
salt '*' ceph.keyring_auth_list \
        'cluster_name'='ceph' \
        'cluster_uuid'='cluster_uuid'
```

cluster_name The cluster name. Defaults to `ceph`.

cluster_uuid The cluster UUID. Defaults to value found in ceph config file.

`salt.modules.ceph.`**`keyring_create`**`(`***kwargs*`)`

Create keyring for cluster

CLI Example:

```
salt '*' ceph.keyring_create \
        'keyring_type'='admin' \
        'cluster_name'='ceph' \
        'cluster_uuid'='cluster_uuid'
```

keyring_type (required) One of `admin`, `mon`, `osd`, `rgw`, `mds`

cluster_uuid The cluster UUID. Defaults to value found in ceph config file.

cluster_name The cluster name. Defaults to `ceph`.

`salt.modules.ceph.`**`keyring_present`**`(`***kwargs*`)`

Returns `True` if the keyring is present on disk, otherwise `False`

CLI Example:

```
salt '*' ceph.keyring_present \
        'keyring_type'='admin' \
        'cluster_name'='ceph' \
        'cluster_uuid'='cluster_uuid'
```

keyring_type (required) One of `admin`, `mon`, `osd`, `rgw`, `mds`

cluster_uuid The cluster UUID. Defaults to value found in ceph config file.

cluster_name The cluster name. Defaults to `ceph`.

`salt.modules.ceph.`**`keyring_purge`**`(`*`**kwargs`*`)`

Delete keyring for cluster

CLI Example:

```
salt '*' ceph.keyring_purge \
        'keyring_type'='admin' \
        'cluster_name'='ceph' \
        'cluster_uuid'='cluster_uuid'
```

keyring_type (required) One of `admin`, `mon`, `osd`, `rgw`, `mds`

cluster_uuid The cluster UUID. Defaults to value found in ceph config file.

cluster_name The cluster name. Defaults to `ceph`.

If no ceph config file is found, this command will fail.

`salt.modules.ceph.`**`keyring_save`**`(`*`**kwargs`*`)`

Create save keyring locally

CLI Example:

```
salt '*' ceph.keyring_save \
        'keyring_type'='admin' \
        'cluster_name'='ceph' \
        'cluster_uuid'='cluster_uuid'
```

keyring_type (required) One of `admin`, `mon`, `osd`, `rgw`, `mds`

cluster_uuid The cluster UUID. Defaults to value found in ceph config file.

cluster_name The cluster name. Defaults to `ceph`.

`salt.modules.ceph.`**`mds_create`**`(`*`**kwargs`*`)`

Create a mds

CLI Example:

```
salt '*' ceph.mds_create \
        'name' = 'mds.name' \
        'port' = 1000, \
        'addr' = 'fqdn.example.org' \
        'cluster_name'='ceph' \
        'cluster_uuid'='cluster_uuid'
```

name (required) The MDS name (must start with `mds.`)

port (required) Port to which the MDS will listen

addr (required) Address or IP address for the MDS to listen

cluster_uuid The cluster UUID. Defaults to value found in ceph config file.

cluster_name The cluster name. Defaults to `ceph`.

salt.modules.ceph.**mds_destroy**(***kwargs*)

Remove a mds

CLI Example:

```
salt '*' ceph.mds_destroy \
        'name' = 'mds.name' \
        'cluster_name'='ceph' \
        'cluster_uuid'='cluster_uuid'
```

name (required) The MDS name (must start with `mds.`)

cluster_uuid The cluster UUID. Defaults to value found in ceph config file.

cluster_name The cluster name. Defaults to `ceph`.

salt.modules.ceph.**mon_active**(***kwargs*)

Returns `True` if the mon daemon is running, otherwise `False`

CLI Example:

```
salt '*' ceph.mon_active \
        'cluster_name'='ceph' \
        'cluster_uuid'='cluster_uuid'
```

cluster_uuid The cluster UUID. Defaults to value found in ceph config file.

cluster_name The cluster name. Defaults to `ceph`.

salt.modules.ceph.**mon_create**(***kwargs*)

Create a mon node

CLI Example:

```
salt '*' ceph.mon_create \
        'cluster_name'='ceph' \
        'cluster_uuid'='cluster_uuid'
```

cluster_uuid The cluster UUID. Defaults to value found in ceph config file.

cluster_name The cluster name. Defaults to `ceph`.

salt.modules.ceph.**mon_is**(***kwargs*)

Returns `True` if the target is a mon node, otherwise `False`

CLI Example:

```
salt '*' ceph.mon_is \
        'cluster_name'='ceph' \
        'cluster_uuid'='cluster_uuid'
```

cluster_name The cluster name. Defaults to `ceph`.

cluster_uuid The cluster UUID. Defaults to value found in ceph config file.

salt.modules.ceph.**mon_quorum**(***kwargs*)

Returns `True` if the mon daemon is in the quorum, otherwise `False`

CLI Example:

```
salt '*' ceph.mon_quorum \
        'cluster_name'='ceph' \
        'cluster_uuid'='cluster_uuid'
```

cluster_uuid The cluster UUID. Defaults to value found in ceph config file.

cluster_name The cluster name. Defaults to `ceph`.

salt.modules.ceph.**mon_status**(***kwargs*)

Get status from mon daemon

CLI Example:

```
salt '*' ceph.mon_status \
        'cluster_name'='ceph' \
        'cluster_uuid'='cluster_uuid'
```

cluster_uuid The cluster UUID. Defaults to value found in ceph config file.

cluster_name The cluster name. Defaults to `ceph`.

salt.modules.ceph.**osd_activate**(***kwargs*)

Activate an OSD

CLI Example:

```
salt '*' ceph.osd_activate 'osd_dev'='/dev/vdc'
```

salt.modules.ceph.**osd_discover**()

List all OSD by cluster

CLI Example:

```
salt '*' ceph.osd_discover
```

salt.modules.ceph.**osd_prepare**(***kwargs*)

Prepare an OSD

CLI Example:

```
salt '*' ceph.osd_prepare 'osd_dev'='/dev/vdc' \
        'journal_dev'='device' \
        'cluster_name'='ceph' \
        'cluster_uuid'='cluster_uuid' \
        'osd_fs_type'='xfs' \
        'osd_uuid'='2a143b73-6d85-4389-a9e9-b8a78d9e1e07' \
        'journal_uuid'='4562a5db-ff6f-4268-811d-12fd4a09ae98'
```

cluster_uuid The device to store the osd data on.

journal_dev The journal device. defaults to osd_dev.

cluster_name The cluster name. Defaults to `ceph`.

cluster_uuid The cluster date will be added too. Defaults to the value found in local config.

osd_fs_type set the file system to store OSD data with. Defaults to "xfs".

osd_uuid set the OSD data UUID. If set will return if OSD with data UUID already exists.

journal_uuid set the OSD journal UUID. If set will return if OSD with journal UUID already exists.

`salt.modules.ceph.`**`partition_is`**`(`*dev*`)`

Check whether a given device path is a partition or a full disk.

CLI Example:

```
salt '*' ceph.partition_is /dev/sdc1
```

`salt.modules.ceph.`**`partition_list`**`()`

List partitions by disk

CLI Example:

```
salt '*' ceph.partition_list
```

`salt.modules.ceph.`**`partition_list_journal`**`()`

List all OSD journal partitions by partition

CLI Example:

```
salt '*' ceph.partition_list_journal
```

`salt.modules.ceph.`**`partition_list_osd`**`()`

List all OSD data partitions by partition

CLI Example:

```
salt '*' ceph.partition_list_osd
```

`salt.modules.ceph.`**`pool_add`**`(`*pool_name, **kwargs*`)`

Create a pool

CLI Example:

```
salt '*' ceph.pool_add pool_name \
        'cluster_name'='ceph' \
        'cluster_uuid'='cluster_uuid'
```

cluster_name The cluster name. Defaults to `ceph`.

cluster_uuid The cluster UUID. Defaults to value found in ceph config file.

pg_num Default to 8

pgp_num Default to pg_num

pool_type can take values "replicated" or "erasure"

erasure_code_profile The "erasure_code_profile"

crush_ruleset The crush map rule set

`salt.modules.ceph.`**`pool_del`**`(`*pool_name, **kwargs*`)`

Delete a pool

CLI Example:

```
salt '*' ceph.pool_del pool_name \
        'cluster_name'='ceph' \
        'cluster_uuid'='cluster_uuid'
```

cluster_name The cluster name. Defaults to `ceph`.

cluster_uuid The cluster UUID. Defaults to value found in ceph config file.

`salt.modules.ceph.`**`pool_list`**`(`***kwargs*`)`

List all pools

CLI Example:

```
salt '*' ceph.pool_list \
        'cluster_name'='ceph' \
        'cluster_uuid'='cluster_uuid'
```

cluster_name The cluster name. Defaults to `ceph`.

cluster_uuid The cluster UUID. Defaults to value found in ceph config file.

`salt.modules.ceph.`**`purge`**`(`***kwargs*`)`

purge ceph configuration on the node

CLI Example:

```
salt '*' ceph.purge \
        'cluster_name'='ceph' \
        'cluster_uuid'='cluster_uuid'
```

cluster_name The cluster name. Defaults to `ceph`.

cluster_uuid The cluster UUID. Defaults to value found in ceph config file.

`salt.modules.ceph.`**`rgw_create`**`(`***kwargs*`)`

Create a rgw

CLI Example:

```
salt '*' ceph.rgw_create \
        'name' = 'rgw.name' \
        'cluster_name'='ceph' \
        'cluster_uuid'='cluster_uuid'
```

name (required) The RGW client name. Must start with `rgw.`

cluster_uuid The cluster UUID. Defaults to value found in ceph config file.

cluster_name The cluster name. Defaults to `ceph`.

`salt.modules.ceph.`**`rgw_destroy`**`(`***kwargs*`)`

Remove a rgw

CLI Example:

```
salt '*' ceph.rgw_destroy \
        'name' = 'rgw.name' \
        'cluster_name'='ceph' \
        'cluster_uuid'='cluster_uuid'
```

name (required) The RGW client name (must start with `rgw.`)

cluster_uuid The cluster UUID. Defaults to value found in ceph config file.

cluster_name The cluster name. Defaults to `ceph`.

`salt.modules.ceph.`**`rgw_pools_create`**`(`*`**kwargs`*`)`
Create pools for rgw

CLI Example:

```
salt '*' ceph.rgw_pools_create
```

cluster_uuid The cluster UUID. Defaults to value found in ceph config file.

cluster_name The cluster name. Defaults to `ceph`.

`salt.modules.ceph.`**`rgw_pools_missing`**`(`*`**kwargs`*`)`
Show pools missing for rgw

CLI Example:

```
salt '*' ceph.rgw_pools_missing
```

cluster_uuid The cluster UUID. Defaults to value found in ceph config file.

cluster_name The cluster name. Defaults to `ceph`.

`salt.modules.ceph.`**`zap`**`(`*`target=None, **kwargs`*`)`
Destroy the partition table and content of a given disk.

```
salt '*' ceph.osd_prepare 'dev'='/dev/vdc' \
      'cluster_name'='ceph' \
      'cluster_uuid'='cluster_uuid'
```

dev The block device to format.

cluster_name The cluster name. Defaults to `ceph`.

cluster_uuid The cluster UUID. Defaults to value found in ceph config file.

21.9.74 salt.modules.chassis

Glue execution module to link to the *`fx2 proxymodule`*.

Depends: *`iDRAC Remote execution module (salt.modules.dracr)`*

For documentation on commands that you can direct to a Dell chassis via proxy, look in the documentation for *`salt.modules.dracr`*.

This execution module calls through to a function in the fx2 proxy module called `chconfig`. That function looks up the function passed in the `cmd` parameter in *`salt.modules.dracr`* and calls it.

New in version 2015.8.2.

`salt.modules.chassis.`**`chassis_credentials`**`()`

`salt.modules.chassis.`**`cmd`**`(`*`cmd, *args, **kwargs`*`)`

21.9.75 salt.modules.chef

Execute chef in server or solo mode

`salt.modules.chef.`**`client`**`(`*whyrun=False*, *localmode=False*, *logfile=None*, ***kwargs*`)`

Execute a chef client run and return a dict with the stderr, stdout, return code, and pid.

CLI Example:

```
salt '*' chef.client server=https://localhost
```

server The chef server URL

client_key Set the client key file location

config The configuration file to use

config-file-jail Directory under which config files are allowed to be loaded (no client.rb or knife.rb outside this path will be loaded).

environment Set the Chef Environment on the node

group Group to set privilege to

json-attributes Load attributes from a JSON file or URL

localmode Point chef-client at local repository if True

log_level Set the log level (debug, info, warn, error, fatal)

logfile Set the log file location

node-name The node name for this client

override-runlist Replace current run list with specified items for a single run

pid Set the PID file location, defaults to /tmp/chef-client.pid

run-lock-timeout Set maximum duration to wait for another client run to finish, default is indefinitely.

runlist Permanently replace current run list with specified items

user User to set privilege to

validation_key Set the validation key file location, used for registering new clients

whyrun Enable whyrun mode when set to True

`salt.modules.chef.`**`solo`**`(`*whyrun=False*, *logfile=None*, ***kwargs*`)`

Execute a chef solo run and return a dict with the stderr, stdout, return code, and pid.

CLI Example:

```
salt '*' chef.solo override-runlist=test
```

config The configuration file to use

environment Set the Chef Environment on the node

group Group to set privilege to

json-attributes Load attributes from a JSON file or URL

log_level Set the log level (debug, info, warn, error, fatal)

logfile Set the log file location

node-name The node name for this client

override-runlist Replace current run list with specified items for a single run

recipe-url Pull down a remote gzipped tarball of recipes and untar it to the cookbook cache

run-lock-timeout Set maximum duration to wait for another client run to finish, default is indefinitely.

user User to set privilege to

whyrun Enable whyrun mode when set to True

21.9.76 salt.modules.chocolatey

A dead simple module wrapping calls to the Chocolatey package manager (http://chocolatey.org)

New in version 2014.1.0.

`salt.modules.chocolatey.`**`add_source`**`(`*name*, *source_location*, *username=None*, *password=None*`)`

Instructs Chocolatey to add a source.

name The name of the source to be added as a chocolatey repository.

source Location of the source you want to work with.

username Provide username for chocolatey sources that need authentication credentials.

password Provide password for chocolatey sources that need authentication credentials.

CLI Example:

```
salt '*' chocolatey.add_source <source name> <source_location>
salt '*' chocolatey.add_source <source name> <source_location> user=<user>
↪password=<password>
```

`salt.modules.chocolatey.`**`bootstrap`**`(`*force=False*`)`

Download and install the latest version of the Chocolatey package manager via the official bootstrap.

Chocolatey requires Windows PowerShell and the .NET v4.0 runtime. Depending on the host's version of Windows, chocolatey.bootstrap will attempt to ensure these prerequisites are met by downloading and executing the appropriate installers from Microsoft.

Note that if PowerShell is installed, you may have to restart the host machine for Chocolatey to work.

force Run the bootstrap process even if Chocolatey is found in the path.

CLI Example:

```
salt '*' chocolatey.bootstrap
salt '*' chocolatey.bootstrap force=True
```

`salt.modules.chocolatey.`**`chocolatey_version`**`()`

Returns the version of Chocolatey installed on the minion.

CLI Example:

```
salt '*' chocolatey.chocolatey_version
```

`salt.modules.chocolatey.`**`disable_source`**`(`*name*`)`

Instructs Chocolatey to disable a source.

name Name of the source repository to disable.

CLI Example:

```
salt '*' chocolatey.disable_source <name>
```

`salt.modules.chocolatey.`**`enable_source`**`(`*name*`)`

Instructs Chocolatey to enable a source.

name Name of the source repository to enable.

CLI Example:

```
salt '*' chocolatey.enable_source <name>
```

`salt.modules.chocolatey.`**`install`**`(`*name, version=None, source=None, force=False, pre_versions=False, install_args=None, override_args=False, force_x86=False, package_args=None, allow_multiple=False, execution_timeout=None*`)`

Instructs Chocolatey to install a package.

Parameters

- **`name`** (*`str`*) – The name of the package to be installed. Only accepts a single argument. Required.
- **`version`** (*`str`*) – Install a specific version of the package. Defaults to latest version. Default is None.
- **`source`** (*`str`*) – Chocolatey repository (directory, share or remote URL feed) the package comes from. Defaults to the official Chocolatey feed. Default is None.

  Alternate Sources:

  - cygwin
  - python
  - ruby
  - webpi
  - windowsfeatures
- **`force`** (*`bool`*) – Reinstall the current version of an existing package. Do not use with `allow_multiple`. Default is False.
- **`pre_versions`** (*`bool`*) – Include pre-release packages. Default is False.
- **`install_args`** (*`str`*) – A list of install arguments you want to pass to the installation process i.e product key or feature list. Default is None.
- **`override_args`** (*`bool`*) – Set to true if you want to override the original install arguments (for the native installer) in the package and use your own. When this is set to False install_args will be appended to the end of the default arguments. Default is None.
- **`force_x86`** (*`bool`*) – Force x86 (32bit) installation on 64 bit systems. Default is False.
- **`package_args`** (*`str`*) – Arguments you want to pass to the package. Default is None.
- **`allow_multiple`** (*`bool`*) – Allow multiple versions of the package to be installed. Do not use with `force`. Does not work with all packages. Default is False.

  New in version 2017.7.0.
- **`execution_timeout`** (*`str`*) –
- **`execution timeout value you want to pass to the installation process. Default is None.`** (*`Chocolatey`*) – New in version 2018.3.0.

Returns The output of the `chocolatey` command

Return type str

CLI Example:

```
salt '*' chocolatey.install <package name>
salt '*' chocolatey.install <package name> version=<package version>
salt '*' chocolatey.install <package name> install_args=<args> override_args=True
```

`salt.modules.chocolatey.`**`install_cygwin`**(*name, install_args=None, override_args=False*)

Instructs Chocolatey to install a package via Cygwin.

name The name of the package to be installed. Only accepts a single argument.

install_args A list of install arguments you want to pass to the installation process i.e product key or feature list

override_args Set to true if you want to override the original install arguments (for the native installer) in the package and use your own. When this is set to False install_args will be appended to the end of the default arguments

CLI Example:

```
salt '*' chocolatey.install_cygwin <package name>
salt '*' chocolatey.install_cygwin <package name> install_args=<args> override_
↪args=True
```

`salt.modules.chocolatey.`**`install_gem`**(*name, version=None, install_args=None, override_args=False*)

Instructs Chocolatey to install a package via Ruby's Gems.

name The name of the package to be installed. Only accepts a single argument.

version Install a specific version of the package. Defaults to latest version available.

install_args A list of install arguments you want to pass to the installation process i.e product key or feature list

override_args Set to true if you want to override the original install arguments (for the native installer) in the package and use your own. When this is set to False install_args will be appended to the end of the default arguments

CLI Example:

```
salt '*' chocolatey.install_gem <package name>
salt '*' chocolatey.install_gem <package name> version=<package version>
salt '*' chocolatey.install_gem <package name> install_args=<args> override_
↪args=True
```

`salt.modules.chocolatey.`**`install_missing`**(*name, version=None, source=None*)

Instructs Chocolatey to install a package if it doesn't already exist.

Changed in version 2014.7.0: If the minion has Chocolatey >= 0.9.8.24 installed, this function calls *`chocolatey.install`* instead, as `installmissing` is deprecated as of that version and will be removed in Chocolatey 1.0.

name The name of the package to be installed. Only accepts a single argument.

version Install a specific version of the package. Defaults to latest version available.

source Chocolatey repository (directory, share or remote URL feed) the package comes from. Defaults to the official Chocolatey feed.

CLI Example:

```
salt '*' chocolatey.install_missing <package name>
salt '*' chocolatey.install_missing <package name> version=<package version>
```

salt.modules.chocolatey.**install_python**(*name*, *version=None*, *install_args=None*, *override_args=False*)

Instructs Chocolatey to install a package via Python's easy_install.

name The name of the package to be installed. Only accepts a single argument.

version Install a specific version of the package. Defaults to latest version available.

install_args A list of install arguments you want to pass to the installation process i.e product key or feature list

override_args Set to true if you want to override the original install arguments (for the native installer) in the package and use your own. When this is set to False install_args will be appended to the end of the default arguments

CLI Example:

```
salt '*' chocolatey.install_python <package name>
salt '*' chocolatey.install_python <package name> version=<package version>
salt '*' chocolatey.install_python <package name> install_args=<args> override_
↪args=True
```

salt.modules.chocolatey.**install_webpi**(*name*, *install_args=None*, *override_args=False*)

Instructs Chocolatey to install a package via the Microsoft Web PI service.

name The name of the package to be installed. Only accepts a single argument.

install_args A list of install arguments you want to pass to the installation process i.e product key or feature list

override_args Set to true if you want to override the original install arguments (for the native installer) in the package and use your own. When this is set to False install_args will be appended to the end of the default arguments

CLI Example:

```
salt '*' chocolatey.install_webpi <package name>
salt '*' chocolatey.install_webpi <package name> install_args=<args> override_
↪args=True
```

salt.modules.chocolatey.**install_windowsfeatures**(*name*)

Instructs Chocolatey to install a Windows Feature via the Deployment Image Servicing and Management tool.

name The name of the feature to be installed. Only accepts a single argument.

CLI Example:

```
salt '*' chocolatey.install_windowsfeatures <package name>
```

salt.modules.chocolatey.**list_**(*narrow=None*, *all_versions=False*, *pre_versions=False*, *source=None*, *local_only=False*, *exact=False*)

Instructs Chocolatey to pull a vague package list from the repository.

Parameters

- **narrow** (*str*) – Term used to narrow down results. Searches against name/description/tag. Default is None.

- **`all_versions`** (*bool*) – Display all available package versions in results. Default is False.
- **`pre_versions`** (*bool*) – Display pre-release packages in results. Default is False.
- **`source`** (*str*) – Chocolatey repository (directory, share or remote URL feed) the package comes from. Defaults to the official Chocolatey feed if None is passed. Default is None.
- **`local_only`** (*bool*) – Display packages only installed locally. Default is False.
- **`exact`** (*bool*) – Display only packages that match `narrow` exactly. Default is False.

  New in version 2017.7.0.

Returns A dictionary of results.

Return type dict

CLI Example:

```
salt '*' chocolatey.list <narrow>
salt '*' chocolatey.list <narrow> all_versions=True
```

`salt.modules.chocolatey.`**`list_webpi`**`()`

Instructs Chocolatey to pull a full package list from the Microsoft Web PI repository.

Returns List of webpi packages

Return type str

CLI Example:

```
salt '*' chocolatey.list_webpi
```

`salt.modules.chocolatey.`**`list_windowsfeatures`**`()`

Instructs Chocolatey to pull a full package list from the Windows Features list, via the Deployment Image Servicing and Management tool.

Returns List of Windows Features

Return type str

CLI Example:

```
salt '*' chocolatey.list_windowsfeatures
```

`salt.modules.chocolatey.`**`uninstall`**`(`*name, version=None, uninstall_args=None, override_args=False*`)`

Instructs Chocolatey to uninstall a package.

name The name of the package to be uninstalled. Only accepts a single argument.

version Uninstalls a specific version of the package. Defaults to latest version installed.

uninstall_args A list of uninstall arguments you want to pass to the uninstallation process i.e product key or feature list

override_args Set to true if you want to override the original uninstall arguments (for the native uninstaller) in the package and use your own. When this is set to False uninstall_args will be appended to the end of the default arguments

CLI Example:

```
salt '*' chocolatey.uninstall <package name>
salt '*' chocolatey.uninstall <package name> version=<package version>
salt '*' chocolatey.uninstall <package name> version=<package version> uninstall_
↪args=<args> override_args=True
```

`salt.modules.chocolatey.`**`update`**(*name*, *source=None*, *pre_versions=False*)

Instructs Chocolatey to update packages on the system.

name The name of the package to update, or "all" to update everything installed on the system.

source Chocolatey repository (directory, share or remote URL feed) the package comes from. Defaults to the official Chocolatey feed.

pre_versions Include pre-release packages in comparison. Defaults to False.

CLI Example:

```
salt "*" chocolatey.update all
salt "*" chocolatey.update <package name> pre_versions=True
```

`salt.modules.chocolatey.`**`upgrade`**(*name*, *version=None*, *source=None*, *force=False*, *pre_versions=False*, *install_args=None*, *override_args=False*, *force_x86=False*, *package_args=None*)

New in version 2016.3.4.

Instructs Chocolatey to upgrade packages on the system. (update is being deprecated). This command will install the package if not installed.

Parameters

- **`name`** (*`str`*) – The name of the package to update, or "all" to update everything installed on the system.
- **`version`** (*`str`*) – Install a specific version of the package. Defaults to latest version.
- **`source`** (*`str`*) – Chocolatey repository (directory, share or remote URL feed) the package comes from. Defaults to the official Chocolatey feed.
- **`force`** (*`bool`*) – Reinstall the **same** version already installed
- **`pre_versions`** (*`bool`*) – Include pre-release packages in comparison. Defaults to False.
- **`install_args`** (*`str`*) – A list of install arguments you want to pass to the installation process i.e product key or feature list
- **`override_args`** (*`str`*) – Set to true if you want to override the original install arguments (for the native installer) in the package and use your own. When this is set to False install_args will be appended to the end of the default arguments
- **`force_x86`** – Force x86 (32bit) installation on 64 bit systems. Defaults to false.
- **`package_args`** – A list of arguments you want to pass to the package

Returns Results of the `chocolatey` command

Return type str

CLI Example:

```
salt "*" chocolatey.upgrade all
salt "*" chocolatey.upgrade <package name> pre_versions=True
```

`salt.modules.chocolatey.`**`version`**`(`*name, check_remote=False, source=None, pre_versions=False*`)`

Instructs Chocolatey to check an installed package version, and optionally compare it to one available from a remote feed.

Parameters

- **name** (*str*) – The name of the package to check. Required.
- **check_remote** (*bool*) – Get the version number of the latest package from the remote feed. Default is False.
- **source** (*str*) – Chocolatey repository (directory, share or remote URL feed) the package comes from. Defaults to the official Chocolatey feed. Default is None.
- **pre_versions** (*bool*) – Include pre-release packages in comparison. Default is False.

Returns A dictionary of currently installed software and versions

Return type dict

CLI Example:

```
salt "*" chocolatey.version <package name>
salt "*" chocolatey.version <package name> check_remote=True
```

21.9.77 salt.modules.chronos module

Module providing a simple management interface to a chronos cluster.

Currently this only works when run through a proxy minion.

New in version 2015.8.2.

`salt.modules.chronos.`**`has_job`**`(`*name*`)`

Return whether the given job is currently configured.

CLI Example:

```
salt chronos-minion-id chronos.has_job my-job
```

`salt.modules.chronos.`**`job`**`(`*name*`)`

Return the current server configuration for the specified job.

CLI Example:

```
salt chronos-minion-id chronos.job my-job
```

`salt.modules.chronos.`**`jobs`**`()`

Return a list of the currently installed job names.

CLI Example:

```
salt chronos-minion-id chronos.jobs
```

`salt.modules.chronos.`**`rm_job`**`(`*name*`)`

Remove the specified job from the server.

CLI Example:

```
salt chronos-minion-id chronos.rm_job my-job
```

`salt.modules.chronos.`**`update_job`**`(`*name, config*`)`

Update the specified job with the given configuration.

CLI Example:

```
salt chronos-minion-id chronos.update_job my-job '<config yaml>'
```

21.9.78 salt.modules.cimc module

Module to provide Cisco UCS compatibility to Salt

codeauthor `Spencer Ervin <spencer_ervin@hotmail.com>`

maturity new

depends none

platform unix

Configuration

This module accepts connection configuration details either as parameters, or as configuration settings in pillar as a Salt proxy. Options passed into opts will be ignored if options are passed into pillar.

See also:

`Cisco UCS Proxy Module`

About

This execution module was designed to handle connections to a Cisco UCS server. This module adds support to send connections directly to the device through the rest API.

`salt.modules.cimc.`**`activate_backup_image`**`(`*reset=False*`)`

Activates the firmware backup image.

CLI Example:

Parameters **`reset`** (*`bool`*) – Reset the CIMC device on activate.

```
salt '*' cimc.activate_backup_image
salt '*' cimc.activate_backup_image reset=True
```

`salt.modules.cimc.`**`create_user`**`(`*uid=None, username=None, password=None, priv=None*`)`

Create a CIMC user with username and password.

Parameters

- **`uid`** (*`int`*) – The user ID slot to create the user account in.
- **`username`** (*`str`*) – The name of the user.
- **`password`** (*`str`*) – The clear text password of the user.
- **`priv`** (*`str`*) – The privilege level of the user.

CLI Example:

```
salt '*' cimc.create_user 11 username=admin password=foobar priv=admin
```

salt.modules.cimc.**get_bios_defaults**()

Get the default values of BIOS tokens.

CLI Example:

```
salt '*' cimc.get_bios_defaults
```

salt.modules.cimc.**get_bios_settings**()

Get the C240 server BIOS token values.

CLI Example:

```
salt '*' cimc.get_bios_settings
```

salt.modules.cimc.**get_boot_order**()

Retrieves the configured boot order table.

CLI Example:

```
salt '*' cimc.get_boot_order
```

salt.modules.cimc.**get_cpu_details**()

Get the CPU product ID details.

CLI Example:

```
salt '*' cimc.get_cpu_details
```

salt.modules.cimc.**get_disks**()

Get the HDD product ID details.

CLI Example:

```
salt '*' cimc.get_disks
```

salt.modules.cimc.**get_ethernet_interfaces**()

Get the adapter Ethernet interface details.

CLI Example:

```
salt '*' cimc.get_ethernet_interfaces
```

salt.modules.cimc.**get_fibre_channel_interfaces**()

Get the adapter fibre channel interface details.

CLI Example:

```
salt '*' cimc.get_fibre_channel_interfaces
```

salt.modules.cimc.**get_firmware**()

Retrieves the current running firmware versions of server components.

CLI Example:

```
salt '*' cimc.get_firmware
```

salt.modules.cimc.**get_hostname**()

Retrieves the hostname from the device.

New in version 2019.2.0.

CLI Example:

```
salt '*' cimc.get_hostname
```

`salt.modules.cimc.`**`get_ldap`**`()`

Retrieves LDAP server details.

CLI Example:

```
salt '*' cimc.get_ldap
```

`salt.modules.cimc.`**`get_management_interface`**`()`

Retrieve the management interface details.

CLI Example:

```
salt '*' cimc.get_management_interface
```

`salt.modules.cimc.`**`get_memory_token`**`()`

Get the memory RAS BIOS token.

CLI Example:

```
salt '*' cimc.get_memory_token
```

`salt.modules.cimc.`**`get_memory_unit`**`()`

Get the IMM/Memory unit product ID details.

CLI Example:

```
salt '*' cimc.get_memory_unit
```

`salt.modules.cimc.`**`get_network_adapters`**`()`

Get the list of network adapaters and configuration details.

CLI Example:

```
salt '*' cimc.get_network_adapters
```

`salt.modules.cimc.`**`get_ntp`**`()`

Retrieves the current running NTP configuration.

CLI Example:

```
salt '*' cimc.get_ntp
```

`salt.modules.cimc.`**`get_pci_adapters`**`()`

Get the PCI adapter product ID details.

CLI Example:

```
salt '*' cimc.get_disks
```

`salt.modules.cimc.`**`get_power_configuration`**`()`

Get the configuration of the power settings from the device. This is only available on some C-Series servers.

New in version 2019.2.0.

CLI Example:

```
salt '*' cimc.get_power_configuration
```

salt.modules.cimc.**get_power_supplies**()

Retrieves the power supply unit details.

CLI Example:

```
salt '*' cimc.get_power_supplies
```

salt.modules.cimc.**get_snmp_config**()

Get the snmp configuration details.

CLI Example:

```
salt '*' cimc.get_snmp_config
```

salt.modules.cimc.**get_syslog**()

Get the Syslog client-server details.

CLI Example:

```
salt '*' cimc.get_syslog
```

salt.modules.cimc.**get_syslog_settings**()

Get the Syslog configuration settings from the system.

New in version 2019.2.0.

CLI Example:

```
salt '*' cimc.get_syslog_settings
```

salt.modules.cimc.**get_system_info**()

Get the system information.

CLI Example:

```
salt '*' cimc.get_system_info
```

salt.modules.cimc.**get_users**()

Get the CIMC users.

CLI Example:

```
salt '*' cimc.get_users
```

salt.modules.cimc.**get_vic_adapters**()

Get the VIC adapter general profile details.

CLI Example:

```
salt '*' cimc.get_vic_adapters
```

salt.modules.cimc.**get_vic_uplinks**()

Get the VIC adapter uplink port details.

CLI Example:

```
salt '*' cimc.get_vic_uplinks
```

`salt.modules.cimc.`**`mount_share`**(*name=None, remote_share=None, remote_file=None, mount_type='nfs', username=None, password=None*)

Mounts a remote file through a remote share. Currently, this feature is supported in version 1.5 or greater. The remote share can be either NFS, CIFS, or WWW.

Some of the advantages of CIMC Mounted vMedia include: Communication between mounted media and target stays local (inside datacenter) Media mounts can be scripted/automated No vKVM requirements for media connection Multiple share types supported Connections supported through all CIMC interfaces

Note: CIMC Mounted vMedia is enabled through BIOS configuration.

Parameters

- **name** (*str*) – The name of the volume on the CIMC device.
- **remote_share** (*str*) – The file share link that will be used to mount the share. This can be NFS, CIFS, or WWW. This
- **be the directory path and not the full path to the remote file.** (*must*) –
- **remote_file** (*str*) – The name of the remote file to mount. It must reside within remote_share.
- **mount_type** (*str*) – The type of share to mount. Valid options are nfs, cifs, and www.
- **username** (*str*) – An optional requirement to pass credentials to the remote share. If not provided, an
- **connection attempt will be made.** (*unauthenticated*) –
- **password** (*str*) – An optional requirement to pass a password to the remote share. If not provided, an
- **connection attempt will be made.** –

CLI Example:

```
salt '*' cimc.mount_share name=WIN7 remote_share=10.xxx.27.xxx:/nfs remote_
↪file=sl1huu.iso

salt '*' cimc.mount_share name=WIN7 remote_share=10.xxx.27.xxx:/nfs remote_
↪file=sl1huu.iso username=bob password=badpassword
```

`salt.modules.cimc.`**`reboot`**()

Power cycling the server.

CLI Example:

```
salt '*' cimc.reboot
```

`salt.modules.cimc.`**`set_hostname`**(*hostname=None*)

Sets the hostname on the server.

New in version 2019.2.0.

Parameters **hostname** (*str*) – The new hostname to set.

CLI Example:

```
salt '*' cimc.set_hostname foobar
```

salt.modules.cimc.**set_logging_levels**(*remote=None*, *local=None*)

Sets the logging levels of the CIMC devices. The logging levels must match the following options: emergency, alert, critical, error, warning, notice, informational, debug.

New in version 2019.2.0.

Parameters

- **remote** (*str*) – The logging level for SYSLOG logs.
- **local** (*str*) – The logging level for the local device.

CLI Example:

```
salt '*' cimc.set_logging_levels remote=error local=notice
```

salt.modules.cimc.**set_ntp_server**(*server1=”*, *server2=”*, *server3=”*, *server4=”*)

Sets the NTP servers configuration. This will also enable the client NTP service.

Parameters

- **server1** (*str*) – The first IP address or FQDN of the NTP servers.
- **server2** (*str*) – The second IP address or FQDN of the NTP servers.
- **server3** (*str*) – The third IP address or FQDN of the NTP servers.
- **server4** (*str*) – The fourth IP address or FQDN of the NTP servers.

CLI Example:

```
salt '*' cimc.set_ntp_server 10.10.10.1

salt '*' cimc.set_ntp_server 10.10.10.1 foo.bar.com
```

salt.modules.cimc.**set_power_configuration**(*policy=None*, *delayType=None*, *delayValue=None*)

Sets the power configuration on the device. This is only available for some C-Series servers.

New in version 2019.2.0.

Parameters

- **policy** (*str*) – The action to be taken when chassis power is restored after
- **unexpected power loss. This can be one of the following** (*an*) – reset: The server is allowed to boot up normally when power is restored. The server can restart immediately or, optionally, after a fixed or random delay.

  stay-off: The server remains off until it is manually restarted.

  last-state: The server restarts and the system attempts to restore any processes that were running before power was lost.
- **delayType** (*str*) – If the selected policy is reset, the restart can be
- **with this option. This can be one of the following** (*delayed*) – fixed: The server restarts after a fixed delay.

  random: The server restarts after a random delay.
- **delayValue** (*int*) – If a fixed delay is selected, once chassis power is
- **and the Cisco IMC has finished rebooting, the system waits for** (*restored*) –

- **specified number of seconds before restarting the server. Enter an** (*the*) –
- **between 0 and 240.** (*integer*) –

CLI Example:

```
salt '*' cimc.set_power_configuration stay-off

salt '*' cimc.set_power_configuration reset fixed 0
```

`salt.modules.cimc.`**`set_syslog_server`**(*server=None, type='primary'*)

Set the SYSLOG server on the host.

Parameters

- **server** (*str*) – The hostname or IP address of the SYSLOG server.
- **type** (*str*) – Specifies the type of SYSLOG server. This can either be primary (default) or secondary.

CLI Example:

```
salt '*' cimc.set_syslog_server foo.bar.com

salt '*' cimc.set_syslog_server foo.bar.com primary

salt '*' cimc.set_syslog_server foo.bar.com secondary
```

`salt.modules.cimc.`**`set_user`**(*uid=None, username=None, password=None, priv=None, status=None*)

Sets a CIMC user with specified configurations.

New in version 2019.2.0.

Parameters

- **uid** (*int*) – The user ID slot to create the user account in.
- **username** (*str*) – The name of the user.
- **password** (*str*) – The clear text password of the user.
- **priv** (*str*) – The privilege level of the user.
- **status** (*str*) – The account status of the user.

CLI Example:

```
salt '*' cimc.set_user 11 username=admin password=foobar priv=admin active
```

`salt.modules.cimc.`**`tftp_update_bios`**(*server=None, path=None*)

Update the BIOS firmware through TFTP.

Parameters

- **server** (*str*) – The IP address or hostname of the TFTP server.
- **path** (*str*) – The TFTP path and filename for the BIOS image.

CLI Example:

```
salt '*' cimc.tftp_update_bios foo.bar.com HP-SL2.cap
```

`salt.modules.cimc.`**`tftp_update_cimc`**(*server=None, path=None*)

Update the CIMC firmware through TFTP.

Parameters

- **server** (*str*) – The IP address or hostname of the TFTP server.
- **path** (*str*) – The TFTP path and filename for the CIMC image.

CLI Example:

```
salt '*' cimc.tftp_update_cimc foo.bar.com HP-SL2.bin
```

21.9.79 salt.modules.ciscoconfparse_mod module

Execution module for ciscoconfparse

New in version 2019.2.0.

This module can be used for basic configuration parsing, audit or validation for a variety of network platforms having Cisco IOS style configuration (one space indentation), including: Cisco IOS, Cisco Nexus, Cisco IOS-XR, Cisco IOS-XR, Cisco ASA, Arista EOS, Brocade, HP Switches, Dell PowerConnect Switches, or Extreme Networks devices. In newer versions, `ciscoconfparse` provides support for brace-delimited configuration style as well, for platforms such as: Juniper Junos, Palo Alto, or F5 Networks.

See http://www.pennington.net/py/ciscoconfparse/index.html for further details.

depends ciscoconfparse

This module depends on the Python library with the same name, `ciscoconfparse` - to install execute: `pip install ciscoconfparse`.

`salt.modules.ciscoconfparse_mod.`**`filter_lines`**(*config=None, config_path=None, parent_regex=None, child_regex=None, saltenv='base'*)

Return a list of detailed matches, for the configuration blocks (parent-child relationship) whose parent respects the regular expressions configured via the `parent_regex` argument, and the child matches the `child_regex` regular expression. The result is a list of dictionaries with the following keys:

- `match`: a boolean value that tells whether `child_regex` matched any children lines.
- `parent`: the parent line (as text).
- `child`: the child line (as text). If no child line matched, this field will be `None`.

Note that the return list contains the elements that matched the parent condition, the `parent_regex` regular expression. Therefore, the `parent` field will always have a valid value, while `match` and `child` may default to `False` and `None` respectively when there is not child match.

CLI Example:

```
salt '*' ciscoconfparse.filter_lines config_path=https://bit.ly/2mAdq7z parent_
↪regex='Gigabit' child_regex='shutdown'
```

Example output (for the example above):

```
[
    {
        'parent': 'interface GigabitEthernet1',
        'match': False,
        'child': None
    },
    {
```

(continues on next page)

(continued from previous page)

```
        'parent': 'interface GigabitEthernet2',
        'match': True,
        'child': ' shutdown'
    },
    {
        'parent': 'interface GigabitEthernet3',
        'match': True,
        'child': ' shutdown'
    }
]
```

`salt.modules.ciscoconfparse_mod.`**`find_lines`**`(`*config=None, config_path=None, regex=None, saltenv='base'*`)`

Return all the lines (as text) that match the expression in the `regex` argument.

config The configuration sent as text.

> **Note:** This argument is ignored when `config_path` is specified.

config_path The absolute or remote path to the file with the configuration to be parsed. This argument supports the usual Salt filesystem URIs, e.g., `salt://`, `https://`, `ftp://`, `s3://`, etc.

regex The regular expression to match the lines against.

saltenv: **`base`** Salt fileserver environment from which to retrieve the file. This argument is ignored when `config_path` is not a `salt://` URL.

CLI Example:

```
salt '*' ciscoconfparse.find_lines config_path=https://bit.ly/2mAdq7z regex='ip
↪address'
```

Output example:

```
cisco-ios-router:
    -  ip address dhcp
    -  ip address 172.20.0.1 255.255.255.0
    -  no ip address
```

`salt.modules.ciscoconfparse_mod.`**`find_lines_w_child`**`(`*config=None, config_path=None, parent_regex=None, child_regex=None, ignore_ws=False, saltenv='base'*`)`

Return a list of parent lines (as text) matching the regular expression `parent_regex` that have children lines matching `child_regex`.

config The configuration sent as text.

> **Note:** This argument is ignored when `config_path` is specified.

config_path The absolute or remote path to the file with the configuration to be parsed. This argument supports the usual Salt filesystem URIs, e.g., `salt://`, `https://`, `ftp://`, `s3://`, etc.

parent_regex The regular expression to match the parent lines against.

child_regex The regular expression to match the child lines against.

ignore_ws: **False** Whether to ignore the white spaces.

saltenv: **base** Salt fileserver environment from which to retrieve the file. This argument is ignored when `config_path` is not a `salt://` URL.

CLI Example:

```
salt '*' ciscoconfparse.find_lines_w_child config_path=https://bit.ly/2mAdq7z
↪parent_line='line con' child_line='stopbits'
salt '*' ciscoconfparse.find_lines_w_child config_path=https://bit.ly/2uIRxau
↪parent_regex='ge-(.*)' child_regex='unit \d+'
```

`salt.modules.ciscoconfparse_mod.`**`find_lines_wo_child`**(*config=None, config_path=None, parent_regex=None, child_regex=None, ignore_ws=False, saltenv='base'*)

Return a list of parent `ciscoconfparse.IOSCfgLine` lines as text, which matched the `parent_regex` and whose children did *not* match `child_regex`. Only the parent `ciscoconfparse.IOSCfgLine` text lines will be returned. For simplicity, this method only finds oldest ancestors without immediate children that match.

config The configuration sent as text.

Note: This argument is ignored when `config_path` is specified.

config_path The absolute or remote path to the file with the configuration to be parsed. This argument supports the usual Salt filesystem URIs, e.g., `salt://`, `https://`, `ftp://`, `s3://`, etc.

parent_regex The regular expression to match the parent lines against.

child_regex The regular expression to match the child lines against.

ignore_ws: **False** Whether to ignore the white spaces.

saltenv: **base** Salt fileserver environment from which to retrieve the file. This argument is ignored when `config_path` is not a `salt://` URL.

CLI Example:

```
salt '*' ciscoconfparse.find_lines_wo_child config_path=https://bit.ly/2mAdq7z
↪parent_line='line con' child_line='stopbits'
```

`salt.modules.ciscoconfparse_mod.`**`find_objects`**(*config=None, config_path=None, regex=None, saltenv='base'*)

Return all the line objects that match the expression in the `regex` argument.

Warning: This function is mostly valuable when invoked from other Salt components (i.e., execution modules, states, templates etc.). For CLI usage, please consider using `ciscoconfparse.find_lines`

config The configuration sent as text.

Note: This argument is ignored when `config_path` is specified.

config_path The absolute or remote path to the file with the configuration to be parsed. This argument supports the usual Salt filesystem URIs, e.g., `salt://`, `https://`, `ftp://`, `s3://`, etc.

regex The regular expression to match the lines against.

saltenv: **`base`** Salt fileserver environment from which to retrieve the file. This argument is ignored when `config_path` is not a `salt://` URL.

Usage example:

```
objects = __salt__['ciscoconfparse.find_objects'](config_path='salt://path/to/
↪config.txt',
                                                  regex='Gigabit')
for obj in objects:
    print(obj.text)
```

`salt.modules.ciscoconfparse_mod.`**`find_objects_w_child`**(*config=None, config_path=None, parent_regex=None, child_regex=None, ignore_ws=False, saltenv='base'*)

Parse through the children of all parent lines matching `parent_regex`, and return a list of child objects, which matched the `child_regex`.

Warning: This function is mostly valuable when invoked from other Salt components (i.e., execution modules, states, templates etc.). For CLI usage, please consider using `ciscoconfparse.find_lines_w_child`

config The configuration sent as text.

Note: This argument is ignored when `config_path` is specified.

config_path The absolute or remote path to the file with the configuration to be parsed. This argument supports the usual Salt filesystem URIs, e.g., `salt://`, `https://`, `ftp://`, `s3://`, etc.

parent_regex The regular expression to match the parent lines against.

child_regex The regular expression to match the child lines against.

ignore_ws: **`False`** Whether to ignore the white spaces.

saltenv: **`base`** Salt fileserver environment from which to retrieve the file. This argument is ignored when `config_path` is not a `salt://` URL.

Usage example:

```
objects = __salt__['ciscoconfparse.find_objects_w_child'](config_path='https://
↪bit.ly/2mAdq7z',
                                                          parent_regex='line con',
                                                          child_regex='stopbits')
for obj in objects:
    print(obj.text)
```

`salt.modules.ciscoconfparse_mod.`**`find_objects_wo_child`**(*config=None, config_path=None, parent_regex=None, child_regex=None, ignore_ws=False, saltenv='base'*)

Return a list of parent `ciscoconfparse.IOSCfgLine` objects, which matched the `parent_regex` and whose children did *not* match `child_regex`. Only the parent `ciscoconfparse.IOSCfgLine` objects will be returned. For simplicity, this method only finds oldest ancestors without immediate children that match.

Warning: This function is mostly valuable when invoked from other Salt components (i.e., execution modules, states, templates etc.). For CLI usage, please consider using `ciscoconfparse.find_lines_wo_child`

config The configuration sent as text.

Note: This argument is ignored when `config_path` is specified.

config_path The absolute or remote path to the file with the configuration to be parsed. This argument supports the usual Salt filesystem URIs, e.g., `salt://`, `https://`, `ftp://`, `s3://`, etc.

parent_regex The regular expression to match the parent lines against.

child_regex The regular expression to match the child lines against.

ignore_ws: `False` Whether to ignore the white spaces.

saltenv: `base` Salt fileserver environment from which to retrieve the file. This argument is ignored when `config_path` is not a `salt://` URL.

Usage example:

```
objects = __salt__['ciscoconfparse.find_objects_wo_child'](config_path='https://
↪bit.ly/2mAdq7z',
                                                            parent_regex='line con
↪',
                                                            child_regex='stopbits')
for obj in objects:
    print(obj.text)
```

21.9.80 salt.modules.cisconso

Execution module for Cisco Network Services Orchestrator Proxy minions

For documentation on setting up the cisconso proxy minion look in the documentation for *`salt.proxy.cisconso`*.

`salt.modules.cisconso.`**`apply_rollback`**(*datastore*, *name*)

Apply a system rollback

Parameters

- **`datastore`** (`DatastoreType` (`str` enum).) – The datastore, e.g. running, operational. One of the NETCONF store IETF types
- **`name`** (`str`) – an ID of the rollback to restore

```
salt cisco-nso cisconso.apply_rollback 52
```

`salt.modules.cisconso.`**`get_data`**(*datastore*, *path*)

Get the configuration of the device tree at the given path

Parameters

- **datastore** (`DatastoreType` (`str` enum).) – The datastore, e.g. running, operational. One of the NETCONF store IETF types
- **path** (`list`, `str` OR `tuple`) – The device path to set the value at, a list of element names in order, / separated

Returns The network configuration at that tree

Return type `dict`

```
salt cisco-nso cisconso.get_data running 'devices/ex0'
```

`salt.modules.cisconso.`**`get_rollback`**`(`*name*`)`

Get the backup of stored a configuration rollback

Parameters **name** (`str`) – Typically an ID of the backup

Return type `str`

Returns the contents of the rollback snapshot

```
salt cisco-nso cisconso.get_rollback 52
```

`salt.modules.cisconso.`**`get_rollbacks`**`()`

Get a list of stored configuration rollbacks

```
salt cisco-nso cisconso.get_rollbacks
```

`salt.modules.cisconso.`**`info`**`()`

Return system information for grains of the NSO proxy minion

```
salt '*' cisconso.info
```

`salt.modules.cisconso.`**`set_data_value`**`(`*datastore, path, data*`)`

Set a data entry in a datastore

Parameters

- **datastore** (`DatastoreType` (`str` enum).) – The datastore, e.g. running, operational. One of the NETCONF store IETF types
- **path** (`list`, `str` OR `tuple`) – The device path to set the value at, a list of element names in order, / separated
- **data** (`dict`) – The new value at the given path

Return type `bool`

Returns `True` if successful, otherwise error.

```
salt cisco-nso cisconso.set_data_value running 'devices/ex0/routes' 10.0.0.20/24
```

21.9.81 salt.modules.cloud

Salt-specific interface for calling Salt Cloud directly

`salt.modules.cloud.`**`action`**`(`*fun=None, cloudmap=None, names=None, provider=None, instance=None, **kwargs*`)`

Execute a single action on the given provider/instance

CLI Example:

```
salt minionname cloud.action start instance=myinstance
salt minionname cloud.action stop instance=myinstance
salt minionname cloud.action show_image provider=my-ec2-config image=ami-1624987f
```

`salt.modules.cloud.`**`create`**(*provider, names, opts=None, **kwargs*)

Create an instance using Salt Cloud

CLI Example:

```
salt minionname cloud.create my-ec2-config myinstance image=ami-1624987f size='t1.
↪micro' ssh_username=ec2-user securitygroup=default delvol_on_destroy=True
```

`salt.modules.cloud.`**`destroy`**(*names*)

Destroy the named VM(s)

CLI Example:

```
salt minionname cloud.destroy myinstance
```

`salt.modules.cloud.`**`full_query`**(*query_type='list_nodes_full'*)

List all available cloud provider data

CLI Example:

```
salt minionname cloud.full_query
```

`salt.modules.cloud.`**`get_instance`**(*name, provider=None*)

Return details on an instance.

Similar to the cloud action show_instance but returns only the instance details.

CLI Example:

```
salt minionname cloud.get_instance myinstance
```

SLS Example:

```
{{ salt['cloud.get_instance']('myinstance')['mac_address'] }}
```

`salt.modules.cloud.`**`has_instance`**(*name, provider=None*)

Return true if the instance is found on a provider

CLI Example:

```
salt minionname cloud.has_instance myinstance
```

`salt.modules.cloud.`**`list_images`**(*provider='all'*)

List cloud provider images for the given providers

CLI Example:

```
salt minionname cloud.list_images my-gce-config
```

`salt.modules.cloud.`**`list_locations`**(*provider='all'*)

List cloud provider locations for the given providers

CLI Example:

```
salt minionname cloud.list_locations my-gce-config
```

`salt.modules.cloud.`**`list_sizes`**(*provider='all'*)

List cloud provider sizes for the given providers

CLI Example:

```
salt minionname cloud.list_sizes my-gce-config
```

`salt.modules.cloud.`**`map_run`**(*path=None*, ***kwargs*)

Execute a salt cloud map file

Cloud Map data can be retrieved from several sources:

- a local file (provide the path to the file to the 'path' argument)
- a JSON-formatted map directly (provide the appropriately formatted to using the 'map_data' argument)
- the Salt Pillar (provide the map name of under 'pillar:cloud:maps' to the 'map_pillar' argument)

Note: Only one of these sources can be read at a time. The options are listed in their order of precedence.

CLI Examples:

```
salt minionname cloud.map_run /path/to/cloud.map
salt minionname cloud.map_run path=/path/to/cloud.map
salt minionname cloud.map_run map_pillar='<map_pillar>'
  .. versionchanged:: 2018.3.1
salt minionname cloud.map_run map_data='<actual map data>'
```

`salt.modules.cloud.`**`network_create`**(*provider*, *names*, ***kwargs*)

Create private network

CLI Example:

```
salt minionname cloud.network_create my-nova names=['salt'] cidr='192.168.100.0/24
↪'
```

`salt.modules.cloud.`**`network_list`**(*provider*)

List private networks

CLI Example:

```
salt minionname cloud.network_list my-nova
```

`salt.modules.cloud.`**`profile_`**(*profile*, *names*, *vm_overrides=None*, *opts=None*, ***kwargs*)

Spin up an instance using Salt Cloud

CLI Example:

```
salt minionname cloud.profile my-gce-config myinstance
```

`salt.modules.cloud.`**`query`**(*query_type='list_nodes'*)

List cloud provider data for all providers

CLI Examples:

```
salt minionname cloud.query
salt minionname cloud.query list_nodes_full
salt minionname cloud.query list_nodes_select
```

`salt.modules.cloud.`**`select_query`**(*query_type='list_nodes_select'*)

List selected nodes

CLI Example:

```
salt minionname cloud.select_query
```

`salt.modules.cloud.`**`virtual_interface_create`**(*provider, names, **kwargs*)

Attach private interfaces to a server

CLI Example:

```
salt minionname cloud.virtual_interface_create my-nova names=['salt-master'] net_
↪name='salt'
```

`salt.modules.cloud.`**`virtual_interface_list`**(*provider, names, **kwargs*)

List virtual interfaces on a server

CLI Example:

```
salt minionname cloud.virtual_interface_list my-nova names=['salt-master']
```

`salt.modules.cloud.`**`volume_attach`**(*provider, names, **kwargs*)

Attach volume to a server

CLI Example:

```
salt minionname cloud.volume_attach my-nova myblock server_name=myserver device='/
↪dev/xvdf'
```

`salt.modules.cloud.`**`volume_create`**(*provider, names, **kwargs*)

Create volume

CLI Example:

```
salt minionname cloud.volume_create my-nova myblock size=100 voltype=SSD
```

`salt.modules.cloud.`**`volume_delete`**(*provider, names, **kwargs*)

Delete volume

CLI Example:

```
salt minionname cloud.volume_delete my-nova myblock
```

`salt.modules.cloud.`**`volume_detach`**(*provider, names, **kwargs*)

Detach volume from a server

CLI Example:

```
salt minionname cloud.volume_detach my-nova myblock server_name=myserver
```

`salt.modules.cloud.`**`volume_list`**(*provider*)

List block storage volumes

CLI Example:

```
salt minionname cloud.volume_list my-nova
```

21.9.82 salt.modules.cmdmod

A module for shelling out.

Keep in mind that this module is insecure, in that it can give whomever has access to the master root execution access to all salt minions.

`salt.modules.cmdmod.`**`exec_code`**(*lang, code, cwd=None, args=None, **kwargs*)

Pass in two strings, the first naming the executable language, aka - python2, python3, ruby, perl, lua, etc. the second string containing the code you wish to execute. The stdout will be returned.

All parameters from `cmd.run_all` except python_shell can be used.

CLI Example:

```
salt '*' cmd.exec_code ruby 'puts "cheese"'
salt '*' cmd.exec_code ruby 'puts "cheese"' args='["arg1", "arg2"]' env='{"FOO":
↪"bar"}'
```

`salt.modules.cmdmod.`**`exec_code_all`**(*lang, code, cwd=None, args=None, **kwargs*)

Pass in two strings, the first naming the executable language, aka - python2, python3, ruby, perl, lua, etc. the second string containing the code you wish to execute. All cmd artifacts (stdout, stderr, retcode, pid) will be returned.

All parameters from `cmd.run_all` except python_shell can be used.

CLI Example:

```
salt '*' cmd.exec_code_all ruby 'puts "cheese"'
salt '*' cmd.exec_code_all ruby 'puts "cheese"' args='["arg1", "arg2"]' env='{"FOO
↪": "bar"}'
```

`salt.modules.cmdmod.`**`has_exec`**(*cmd*)

Returns true if the executable is available on the minion, false otherwise

CLI Example:

```
salt '*' cmd.has_exec cat
```

`salt.modules.cmdmod.`**`powershell`**(*cmd, cwd=None, stdin=None, runas=None, shell='/bin/sh', env=None, clean_env=False, template=None, rstrip=True, umask=None, output_encoding=None, output_loglevel='debug', hide_output=False, timeout=None, reset_system_locale=True, ignore_retcode=False, saltenv='base', use_vt=False, password=None, depth=None, encode_cmd=False, success_retcodes=None, **kwargs*)

Execute the passed PowerShell command and return the output as a dictionary.

Other `cmd.*` functions (besides `cmd.powershell_all`) return the raw text output of the command. This function appends `| ConvertTo-JSON` to the command and then parses the JSON into a Python dictionary. If you want the raw textual result of your PowerShell command you should use `cmd.run` with the `shell=powershell` option.

For example:

```
salt '*' cmd.run '$PSVersionTable.CLRVersion' shell=powershell
salt '*' cmd.run 'Get-NetTCPConnection' shell=powershell
```

New in version 2016.3.0.

> **Warning:** This passes the cmd argument directly to PowerShell without any further processing! Be absolutely sure that you have properly sanitized the command passed to this function and do not use untrusted inputs.

In addition to the normal `cmd.run` parameters, this command offers the `depth` parameter to change the Windows default depth for the `ConvertTo-JSON` powershell command. The Windows default is 2. If you need more depth, set that here.

> **Note:** For some commands, setting the depth to a value greater than 4 greatly increases the time it takes for the command to return and in many cases returns useless data.

Parameters

- **cmd** (*str*) – The powershell command to run.
- **cwd** (*str*) – The directory from which to execute the command. Defaults to the home directory of the user specified by `runas` (or the user under which Salt is running if `runas` is not specified).
- **stdin** (*str*) – A string of standard input can be specified for the command to be run using the `stdin` parameter. This can be useful in cases where sensitive information must be read from standard input.
- **runas** (*str*) – Specify an alternate user to run the command. The default behavior is to run as the user under which Salt is running. If running on a Windows minion you must also use the `password` argument, and the target user account must be in the Administrators group.
- **password** (*str*) – Windows only. Required when specifying `runas`. This parameter will be ignored on non-Windows platforms.

  New in version 2016.3.0.
- **shell** (*str*) – Specify an alternate shell. Defaults to the system's default shell.
- **python_shell** (*bool*) – If False, let python handle the positional arguments. Set to True to use shell features, such as pipes or redirection.
- **env** (*dict*) – Environment variables to be set prior to execution.

  > **Note:** When passing environment variables on the CLI, they should be passed as the string representation of a dictionary.
  >
  > ```
  > salt myminion cmd.powershell 'some command' env='{"FOO": "bar"}'
  > ```
- **clean_env** (*bool*) – Attempt to clean out all other shell environment variables and set only those provided in the 'env' argument to this function.

- **template** (*str*) – If this setting is applied then the named templating engine will be used to render the downloaded file. Currently jinja, mako, and wempy are supported.
- **rstrip** (*bool*) – Strip all whitespace off the end of output before it is returned.
- **umask** (*str*) – The umask (in octal) to use when running the command.
- **output_encoding** (*str*) – Control the encoding used to decode the command's output.

  Note: This should not need to be used in most cases. By default, Salt will try to use the encoding detected from the system locale, and will fall back to UTF-8 if this fails. This should only need to be used in cases where the output of the command is encoded in something other than the system locale or UTF-8.

  To see the encoding Salt has detected from the system locale, check the *locale* line in the output of *test.versions_report*.

  New in version 2018.3.0.

- **output_loglevel** (*str*) – Control the loglevel at which the output from the command is logged to the minion log.

  Note: The command being run will still be logged at the `debug` loglevel regardless, unless `quiet` is used for this value.

- **ignore_retcode** (*bool*) – If the exit code of the command is nonzero, this is treated as an error condition, and the output from the command will be logged to the minion log. However, there are some cases where programs use the return code for signaling and a nonzero exit code doesn't necessarily mean failure. Pass this argument as `True` to skip logging the output if the command has a nonzero exit code.
- **hide_output** (*bool*) – If `True`, suppress stdout and stderr in the return data.

  Note: This is separate from `output_loglevel`, which only handles how Salt logs to the minion log.

  New in version 2018.3.0.

- **timeout** (*int*) – A timeout in seconds for the executed process to return.
- **use_vt** (*bool*) – Use VT utils (saltstack) to stream the command output more interactively to the console and the logs. This is experimental.
- **reset_system_locale** (*bool*) – Resets the system locale
- **saltenv** (*str*) – The salt environment to use. Default is 'base'
- **depth** (*int*) – The number of levels of contained objects to be included. Default is 2. Values greater than 4 seem to greatly increase the time it takes for the command to complete for some commands. eg: `dir`

  New in version 2016.3.4.

- **encode_cmd** (*bool*) – Encode the command before executing. Use in cases where characters may be dropped or incorrectly converted when executed. Default is False.
- **success_retcodes** (*list*) –

This parameter will be allow a list of non-zero return codes that should be considered a success. If the return code returned from the run matches any in the provided list, the return code will be overridden with zero.

New in version 2019.2.0.

- **`stdin_raw_newlines`** (*bool*) –

False If `True`, Salt will not automatically convert the characters `\n` present in the `stdin` value to newlines.

New in version 2019.2.0.

Returns

dict A dictionary of data returned by the powershell command.

CLI Example:

```
salt '*' cmd.powershell "$PSVersionTable.CLRVersion"
```

`salt.modules.cmdmod.`**`powershell_all`**(*cmd, cwd=None, stdin=None, runas=None, shell='/bin/sh', env=None, clean_env=False, template=None, rstrip=True, umask=None, output_encoding=None, output_loglevel='debug', quiet=False, timeout=None, reset_system_locale=True, ignore_retcode=False, saltenv='base', use_vt=False, password=None, depth=None, encode_cmd=False, force_list=False, success_retcodes=None, **kwargs*)

Execute the passed PowerShell command and return a dictionary with a result field representing the output of the command, as well as other fields showing us what the PowerShell invocation wrote to `stderr`, the process id, and the exit code of the invocation.

This function appends `| ConvertTo-JSON` to the command before actually invoking powershell.

An unquoted empty string is not valid JSON, but it's very normal for the Powershell output to be exactly that. Therefore, we do not attempt to parse empty Powershell output (which would result in an exception). Instead we treat this as a special case and one of two things will happen:

- If the value of the `force_list` parameter is `True`, then the `result` field of the return dictionary will be an empty list.
- If the value of the `force_list` parameter is `False`, then the return dictionary **will not have a result key added to it**. We aren't setting `result` to `None` in this case, because `None` is the Python representation of "null" in JSON. (We likewise can't use `False` for the equivalent reason.)

If Powershell's output is not an empty string and Python cannot parse its content, then a `CommandExecutionError` exception will be raised.

If Powershell's output is not an empty string, Python is able to parse its content, and the type of the resulting Python object is other than `list` then one of two things will happen:

- If the value of the `force_list` parameter is `True`, then the `result` field will be a singleton list with the Python object as its sole member.
- If the value of the `force_list` parameter is `False`, then the value of `result` will be the unmodified Python object.

If Powershell's output is not an empty string, Python is able to parse its content, and the type of the resulting Python object is `list`, then the value of `result` will be the unmodified Python object. The `force_list` parameter has no effect in this case.

Note: An example of why the `force_list` parameter is useful is as follows: The Powershell command `dir x | Convert-ToJson` results in

- no output when x is an empty directory.
- a dictionary object when x contains just one item.
- a list of dictionary objects when x contains multiple items.

By setting `force_list` to `True` we will always end up with a list of dictionary items, representing files, no matter how many files x contains. Conversely, if `force_list` is `False`, we will end up with no `result` key in our return dictionary when x is an empty directory, and a dictionary object when x contains just one file.

If you want a similar function but with a raw textual result instead of a Python dictionary, you should use `cmd.run_all` in combination with `shell=powershell`.

The remaining fields in the return dictionary are described in more detail in the `Returns` section.

Example:

```
salt '*' cmd.run_all '$PSVersionTable.CLRVersion' shell=powershell
salt '*' cmd.run_all 'Get-NetTCPConnection' shell=powershell
```

New in version 2018.3.0.

Warning: This passes the cmd argument directly to PowerShell without any further processing! Be absolutely sure that you have properly sanitized the command passed to this function and do not use untrusted inputs.

In addition to the normal `cmd.run` parameters, this command offers the `depth` parameter to change the Windows default depth for the `ConvertTo-JSON` powershell command. The Windows default is 2. If you need more depth, set that here.

Note: For some commands, setting the depth to a value greater than 4 greatly increases the time it takes for the command to return and in many cases returns useless data.

Parameters

- **cmd** (*str*) – The powershell command to run.
- **cwd** (*str*) – The directory from which to execute the command. Defaults to the home directory of the user specified by `runas` (or the user under which Salt is running if `runas` is not specified).
- **stdin** (*str*) – A string of standard input can be specified for the command to be run using the `stdin` parameter. This can be useful in cases where sensitive information must be read from standard input.
- **runas** (*str*) – Specify an alternate user to run the command. The default behavior is to run as the user under which Salt is running. If running on a Windows minion you must also use the `password` argument, and the target user account must be in the Administrators group.

- **password** (*str*) – Windows only. Required when specifying `runas`. This parameter will be ignored on non-Windows platforms.
- **shell** (*str*) – Specify an alternate shell. Defaults to the system's default shell.
- **python_shell** (*bool*) – If False, let python handle the positional arguments. Set to True to use shell features, such as pipes or redirection.
- **env** (*dict*) – Environment variables to be set prior to execution.

Note: When passing environment variables on the CLI, they should be passed as the string representation of a dictionary.

```
salt myminion cmd.powershell_all 'some command' env='{"FOO": "bar"}
↪'
```

- **clean_env** (*bool*) – Attempt to clean out all other shell environment variables and set only those provided in the 'env' argument to this function.
- **template** (*str*) – If this setting is applied then the named templating engine will be used to render the downloaded file. Currently jinja, mako, and wempy are supported.
- **rstrip** (*bool*) – Strip all whitespace off the end of output before it is returned.
- **umask** (*str*) – The umask (in octal) to use when running the command.
- **output_encoding** (*str*) – Control the encoding used to decode the command's output.

Note: This should not need to be used in most cases. By default, Salt will try to use the encoding detected from the system locale, and will fall back to UTF-8 if this fails. This should only need to be used in cases where the output of the command is encoded in something other than the system locale or UTF-8.

To see the encoding Salt has detected from the system locale, check the *locale* line in the output of *test.versions_report*.

New in version 2018.3.0.

- **output_loglevel** (*str*) – Control the loglevel at which the output from the command is logged to the minion log.

Note: The command being run will still be logged at the `debug` loglevel regardless, unless `quiet` is used for this value.

- **ignore_retcode** (*bool*) – If the exit code of the command is nonzero, this is treated as an error condition, and the output from the command will be logged to the minion log. However, there are some cases where programs use the return code for signaling and a nonzero exit code doesn't necessarily mean failure. Pass this argument as `True` to skip logging the output if the command has a nonzero exit code.
- **timeout** (*int*) – A timeout in seconds for the executed process to return.
- **use_vt** (*bool*) – Use VT utils (saltstack) to stream the command output more interactively to the console and the logs. This is experimental.
- **reset_system_locale** (*bool*) – Resets the system locale

- **`ignore_retcode`** – If the exit code of the command is nonzero, this is treated as an error condition, and the output from the command will be logged to the minion log. However, there are some cases where programs use the return code for signaling and a nonzero exit code doesn't necessarily mean failure. Pass this argument as `True` to skip logging the output if the command has a nonzero exit code.
- **`saltenv`** (*str*) – The salt environment to use. Default is 'base'
- **`depth`** (*int*) – The number of levels of contained objects to be included. Default is 2. Values greater than 4 seem to greatly increase the time it takes for the command to complete for some commands. eg: `dir`
- **`encode_cmd`** (*bool*) – Encode the command before executing. Use in cases where characters may be dropped or incorrectly converted when executed. Default is False.
- **`force_list`** (*bool*) – The purpose of this parameter is described in the preamble of this function's documentation. Default value is False.
- **`success_retcodes`** (*list*) –

  This parameter will be allow a list of non-zero return codes that should be considered a success. If the return code returned from the run matches any in the provided list, the return code will be overridden with zero.

  New in version 2019.2.0.
- **`stdin_raw_newlines`** (*bool*) –

  False If `True`, Salt will not automatically convert the characters `\n` present in the `stdin` value to newlines.

  New in version 2019.2.0.

Returns

A dictionary with the following entries:

result For a complete description of this field, please refer to this function's preamble. **This key will not be added to the dictionary when force_list is False and Powershell's output is the empty string.**

stderr What the PowerShell invocation wrote to `stderr`.

pid The process id of the PowerShell invocation

retcode This is the exit code of the invocation of PowerShell. If the final execution status (in PowerShell) of our command (with `| ConvertTo-JSON` appended) is `False` this should be non-0. Likewise if PowerShell exited with `$LASTEXITCODE` set to some non-0 value, then `retcode` will end up with this value.

Return type dict

CLI Example:

```
salt '*' cmd.powershell_all "$PSVersionTable.CLRVersion"
```

CLI Example:

```
salt '*' cmd.powershell_all "dir mydirectory" force_list=True
```

`salt.modules.cmdmod.`**`retcode`**(*cmd, cwd=None, stdin=None, runas=None, group=None, shell='/bin/sh', python_shell=None, env=None, clean_env=False, template=None, umask=None, output_encoding=None, output_loglevel='debug', log_callback=None, timeout=None, reset_system_locale=True, ignore_retcode=False, saltenv='base', use_vt=False, password=None, success_retcodes=None, **kwargs*)

Execute a shell command and return the command's return code.

Parameters

- **`cmd`** (*`str`*) – The command to run. ex: `ls -lart /home`
- **`cwd`** (*`str`*) – The directory from which to execute the command. Defaults to the home directory of the user specified by `runas` (or the user under which Salt is running if `runas` is not specified).
- **`stdin`** (*`str`*) – A string of standard input can be specified for the command to be run using the `stdin` parameter. This can be useful in cases where sensitive information must be read from standard input.
- **`runas`** (*`str`*) – Specify an alternate user to run the command. The default behavior is to run as the user under which Salt is running. If running on a Windows minion you must also use the `password` argument, and the target user account must be in the Administrators group.

  Warning: For versions 2018.3.3 and above on macosx while using runas, to pass special characters to the command you need to escape the characters on the shell.

  Example:

  ```
  cmd.retcode 'echo '\''h=\"baz\"'\''' runas=macuser
  ```

- **`password`** (*`str`*) – Windows only. Required when specifying `runas`. This parameter will be ignored on non-Windows platforms.

  New in version 2016.3.0.

- **`group`** (*`str`*) – Group to run command as. Not currently supported on Windows.
- **`shell`** (*`str`*) – Specify an alternate shell. Defaults to the system's default shell.
- **`python_shell`** (*`bool`*) – If False, let python handle the positional arguments. Set to True to use shell features, such as pipes or redirection.
- **`env`** (*`dict`*) – Environment variables to be set prior to execution.

  Note: When passing environment variables on the CLI, they should be passed as the string representation of a dictionary.

  ```
  salt myminion cmd.retcode 'some command' env='{"FOO": "bar"}'
  ```

- **`clean_env`** (*`bool`*) – Attempt to clean out all other shell environment variables and set only those provided in the 'env' argument to this function.
- **`template`** (*`str`*) – If this setting is applied then the named templating engine will be used to render the downloaded file. Currently jinja, mako, and wempy are supported.
- **`rstrip`** (*`bool`*) – Strip all whitespace off the end of output before it is returned.

- **`umask`** (*`str`*) – The umask (in octal) to use when running the command.
- **`output_encoding`** (*`str`*) – Control the encoding used to decode the command's output.

  Note: This should not need to be used in most cases. By default, Salt will try to use the encoding detected from the system locale, and will fall back to UTF-8 if this fails. This should only need to be used in cases where the output of the command is encoded in something other than the system locale or UTF-8.

  To see the encoding Salt has detected from the system locale, check the *locale* line in the output of *`test.versions_report`*.

  New in version 2018.3.0.

- **`output_loglevel`** (*`str`*) – Control the loglevel at which the output from the command is logged to the minion log.

  Note: The command being run will still be logged at the `debug` loglevel regardless, unless `quiet` is used for this value.

- **`ignore_retcode`** (*`bool`*) – If the exit code of the command is nonzero, this is treated as an error condition, and the output from the command will be logged to the minion log. However, there are some cases where programs use the return code for signaling and a nonzero exit code doesn't necessarily mean failure. Pass this argument as `True` to skip logging the output if the command has a nonzero exit code.
- **`timeout`** (*`int`*) – A timeout in seconds for the executed process to return.
- **`use_vt`** (*`bool`*) – Use VT utils (saltstack) to stream the command output more interactively to the console and the logs. This is experimental.
- **`success_retcodes`** (*`list`*) –

  This parameter will be allow a list of non-zero return codes that should be considered a success. If the return code returned from the run matches any in the provided list, the return code will be overridden with zero.

  New in version 2019.2.0.

- **`stdin_raw_newlines`** (*`bool`*) –

  False If `True`, Salt will not automatically convert the characters `\n` present in the `stdin` value to newlines.

  New in version 2019.2.0.

Return type int

Return type None

Returns Return Code as an int or None if there was an exception.

CLI Example:

```
salt '*' cmd.retcode "file /bin/bash"
```

The template arg can be set to 'jinja' or another supported template engine to render the command arguments before execution. For example:

```
salt '*' cmd.retcode template=jinja "file {{grains.pythonpath[0]}}/python"
```

A string of standard input can be specified for the command to be run using the `stdin` parameter. This can be useful in cases where sensitive information must be read from standard input.

```
salt '*' cmd.retcode "grep f" stdin='one\ntwo\nthree\nfour\nfive\n'
```

`salt.modules.cmdmod.`**`run`**(*cmd*, *cwd=None*, *stdin=None*, *runas=None*, *group=None*, *shell='/bin/sh'*, *python_shell=None*, *env=None*, *clean_env=False*, *template=None*, *rstrip=True*, *umask=None*, *output_encoding=None*, *output_loglevel='debug'*, *log_callback=None*, *hide_output=False*, *timeout=None*, *reset_system_locale=True*, *ignore_retcode=False*, *saltenv='base'*, *use_vt=False*, *bg=False*, *password=None*, *encoded_cmd=False*, *raise_err=False*, *prepend_path=None*, *success_retcodes=None*, ***kwargs*)

Execute the passed command and return the output as a string

Parameters

- **`cmd`** (*`str`*) – The command to run. ex: `ls -lart /home`
- **`cwd`** (*`str`*) – The directory from which to execute the command. Defaults to the home directory of the user specified by `runas` (or the user under which Salt is running if `runas` is not specified).
- **`stdin`** (*`str`*) – A string of standard input can be specified for the command to be run using the `stdin` parameter. This can be useful in cases where sensitive information must be read from standard input.
- **`runas`** (*`str`*) – Specify an alternate user to run the command. The default behavior is to run as the user under which Salt is running.

  > **Warning:** For versions 2018.3.3 and above on macosx while using runas, on linux while using run, to pass special characters to the command you need to escape the characters on the shell.
  >
  > Example:
  >
  > ```
  > cmd.run 'echo '\''h=\"baz\"'\''' runas=macuser
  > ```

- **`group`** (*`str`*) – Group to run command as. Not currently supported on Windows.
- **`password`** (*`str`*) – Windows only. Required when specifying `runas`. This parameter will be ignored on non-Windows platforms.

  New in version 2016.3.0.
- **`shell`** (*`str`*) – Specify an alternate shell. Defaults to the system's default shell.
- **`python_shell`** (*`bool`*) – If `False`, let python handle the positional arguments. Set to `True` to use shell features, such as pipes or redirection.
- **`bg`** (*`bool`*) – If `True`, run command in background and do not await or deliver it's results

  New in version 2016.3.0.
- **`env`** (*`dict`*) – Environment variables to be set prior to execution.

  > **Note:** When passing environment variables on the CLI, they should be passed as the string representation of a dictionary.

```
salt myminion cmd.run 'some command' env='{"FOO": "bar"}'
```

- **clean_env** (*bool*) – Attempt to clean out all other shell environment variables and set only those provided in the 'env' argument to this function.
- **prepend_path** (*str*) – $PATH segment to prepend (trailing ':' not necessary) to $PATH

  New in version 2018.3.0.
- **template** (*str*) – If this setting is applied then the named templating engine will be used to render the downloaded file. Currently jinja, mako, and wempy are supported.
- **rstrip** (*bool*) – Strip all whitespace off the end of output before it is returned.
- **umask** (*str*) – The umask (in octal) to use when running the command.
- **output_encoding** (*str*) – Control the encoding used to decode the command's output.

  Note: This should not need to be used in most cases. By default, Salt will try to use the encoding detected from the system locale, and will fall back to UTF-8 if this fails. This should only need to be used in cases where the output of the command is encoded in something other than the system locale or UTF-8.

  To see the encoding Salt has detected from the system locale, check the *locale* line in the output of `test.versions_report`.

  New in version 2018.3.0.
- **output_loglevel** (*str*) – Control the loglevel at which the output from the command is logged to the minion log.

  Note: The command being run will still be logged at the `debug` loglevel regardless, unless `quiet` is used for this value.
- **ignore_retcode** (*bool*) – If the exit code of the command is nonzero, this is treated as an error condition, and the output from the command will be logged to the minion log. However, there are some cases where programs use the return code for signaling and a nonzero exit code doesn't necessarily mean failure. Pass this argument as `True` to skip logging the output if the command has a nonzero exit code.
- **hide_output** (*bool*) – If `True`, suppress stdout and stderr in the return data.

  Note: This is separate from `output_loglevel`, which only handles how Salt logs to the minion log.

  New in version 2018.3.0.
- **timeout** (*int*) – A timeout in seconds for the executed process to return.
- **use_vt** (*bool*) – Use VT utils (saltstack) to stream the command output more interactively to the console and the logs. This is experimental.
- **encoded_cmd** (*bool*) – Specify if the supplied command is encoded. Only applies to shell 'powershell'.

- **raise_err** (*bool*) – If `True` and the command has a nonzero exit code, a CommandExecutionError exception will be raised.

> **Warning:** This function does not process commands through a shell unless the python_shell flag is set to True. This means that any shell-specific functionality such as 'echo' or the use of pipes, redirection or &&, should either be migrated to cmd.shell or have the python_shell=True flag set here.
>
> The use of python_shell=True means that the shell will accept _any_ input including potentially malicious commands such as 'good_command;rm -rf /'. Be absolutely certain that you have sanitized your input prior to using python_shell=True

Parameters

- **success_retcodes** (*list*) –

  This parameter will be allow a list of non-zero return codes that should be considered a success. If the return code returned from the run matches any in the provided list, the return code will be overridden with zero.

  New in version 2019.2.0.

- **stdin_raw_newlines** (*bool*) –

  False If `True`, Salt will not automatically convert the characters `\\n` present in the `stdin` value to newlines.

  New in version 2019.2.0.

CLI Example:

```
salt '*' cmd.run "ls -l | awk '/foo/{print \\$2}'"
```

The template arg can be set to 'jinja' or another supported template engine to render the command arguments before execution. For example:

```
salt '*' cmd.run template=jinja "ls -l /tmp/{{grains.id}} | awk '/foo/{print \\$2}
↪'"
```

Specify an alternate shell with the shell parameter:

```
salt '*' cmd.run "Get-ChildItem C:\\ " shell='powershell'
```

A string of standard input can be specified for the command to be run using the `stdin` parameter. This can be useful in cases where sensitive information must be read from standard input.

```
salt '*' cmd.run "grep f" stdin='one\\ntwo\\nthree\\nfour\\nfive\\n'
```

If an equal sign (=) appears in an argument to a Salt command it is interpreted as a keyword argument in the format `key=val`. That processing can be bypassed in order to pass an equal sign through to the remote shell command by manually specifying the kwarg:

```
salt '*' cmd.run cmd='sed -e s/=/:/g'
```

`salt.modules.cmdmod.`**`run_all`**`(`*cmd, cwd=None, stdin=None, runas=None, group=None, shell='/bin/sh', python_shell=None, env=None, clean_env=False, template=None, rstrip=True, umask=None, output_encoding=None, output_loglevel='debug', log_callback=None, hide_output=False, timeout=None, reset_system_locale=True, ignore_retcode=False, saltenv='base', use_vt=False, redirect_stderr=False, password=None, encoded_cmd=False, prepend_path=None, success_retcodes=None, **kwargs*`)`

Execute the passed command and return a dict of return data

Parameters

- **cmd** (*str*) – The command to run. ex: `ls -lart /home`
- **cwd** (*str*) – The directory from which to execute the command. Defaults to the home directory of the user specified by `runas` (or the user under which Salt is running if `runas` is not specified).
- **stdin** (*str*) – A string of standard input can be specified for the command to be run using the `stdin` parameter. This can be useful in cases where sensitive information must be read from standard input.
- **runas** (*str*) – Specify an alternate user to run the command. The default behavior is to run as the user under which Salt is running. If running on a Windows minion you must also use the `password` argument, and the target user account must be in the Administrators group.

  Warning: For versions 2018.3.3 and above on macosx while using runas, to pass special characters to the command you need to escape the characters on the shell.

  Example:

  ```
  cmd.run_all 'echo '\''h=\"baz\"'\''' runas=macuser
  ```

- **password** (*str*) – Windows only. Required when specifying `runas`. This parameter will be ignored on non-Windows platforms.

  New in version 2016.3.0.

- **group** (*str*) – Group to run command as. Not currently supported on Windows.
- **shell** (*str*) – Specify an alternate shell. Defaults to the system's default shell.
- **python_shell** (*bool*) – If False, let python handle the positional arguments. Set to True to use shell features, such as pipes or redirection.
- **env** (*dict*) – Environment variables to be set prior to execution.

  Note: When passing environment variables on the CLI, they should be passed as the string representation of a dictionary.

  ```
  salt myminion cmd.run_all 'some command' env='{"FOO": "bar"}'
  ```

- **clean_env** (*bool*) – Attempt to clean out all other shell environment variables and set only those provided in the 'env' argument to this function.
- **prepend_path** (*str*) – $PATH segment to prepend (trailing ':' not necessary) to $PATH

New in version 2018.3.0.

- **template** (*str*) – If this setting is applied then the named templating engine will be used to render the downloaded file. Currently jinja, mako, and wempy are supported.
- **rstrip** (*bool*) – Strip all whitespace off the end of output before it is returned.
- **umask** (*str*) – The umask (in octal) to use when running the command.
- **output_encoding** (*str*) – Control the encoding used to decode the command's output.

Note: This should not need to be used in most cases. By default, Salt will try to use the encoding detected from the system locale, and will fall back to UTF-8 if this fails. This should only need to be used in cases where the output of the command is encoded in something other than the system locale or UTF-8.

To see the encoding Salt has detected from the system locale, check the *locale* line in the output of `test.versions_report`.

New in version 2018.3.0.

- **output_loglevel** (*str*) – Control the loglevel at which the output from the command is logged to the minion log.

Note: The command being run will still be logged at the `debug` loglevel regardless, unless `quiet` is used for this value.

- **ignore_retcode** (*bool*) – If the exit code of the command is nonzero, this is treated as an error condition, and the output from the command will be logged to the minion log. However, there are some cases where programs use the return code for signaling and a nonzero exit code doesn't necessarily mean failure. Pass this argument as `True` to skip logging the output if the command has a nonzero exit code.
- **hide_output** (*bool*) – If `True`, suppress stdout and stderr in the return data.

Note: This is separate from `output_loglevel`, which only handles how Salt logs to the minion log.

New in version 2018.3.0.

- **timeout** (*int*) – A timeout in seconds for the executed process to return.
- **use_vt** (*bool*) – Use VT utils (saltstack) to stream the command output more interactively to the console and the logs. This is experimental.
- **encoded_cmd** (*bool*) – Specify if the supplied command is encoded. Only applies to shell 'powershell'.

  New in version 2018.3.0.

- **redirect_stderr** (*bool*) – If set to `True`, then stderr will be redirected to stdout. This is helpful for cases where obtaining both the retcode and output is desired, but it is not desired to have the output separated into both stdout and stderr.

  New in version 2015.8.2.

- **password** – Windows only. Required when specifying `runas`. This parameter will be ignored on non-Windows platforms.

New in version 2016.3.0.

- **bg** (*bool*) – If `True`, run command in background and do not await or deliver its results

  New in version 2016.3.6.

- **success_retcodes** (*list*) –

  This parameter will be allow a list of non-zero return codes that should be considered a success. If the return code returned from the run matches any in the provided list, the return code will be overridden with zero.

  New in version 2019.2.0.

- **stdin_raw_newlines** (*bool*) –

  False If `True`, Salt will not automatically convert the characters `\n` present in the `stdin` value to newlines.

  New in version 2019.2.0.

CLI Example:

```
salt '*' cmd.run_all "ls -l | awk '/foo/{print \$2}'"
```

The template arg can be set to 'jinja' or another supported template engine to render the command arguments before execution. For example:

```
salt '*' cmd.run_all template=jinja "ls -l /tmp/{{grains.id}} | awk '/foo/{print \
↪$2}'"
```

A string of standard input can be specified for the command to be run using the `stdin` parameter. This can be useful in cases where sensitive information must be read from standard input.

```
salt '*' cmd.run_all "grep f" stdin='one\ntwo\nthree\nfour\nfive\n'
```

`salt.modules.cmdmod.`**`run_bg`**(*cmd, cwd=None, runas=None, group=None, shell='/bin/sh', python_shell=None, env=None, clean_env=False, template=None, umask=None, timeout=None, output_encoding=None, output_loglevel='debug', log_callback=None, reset_system_locale=True, ignore_retcode=False, saltenv='base', password=None, prepend_path=None, success_retcodes=None, **kwargs*)

Execute the passed command in the background and return it's PID

Note: If the init system is systemd and the backgrounded task should run even if the salt-minion process is restarted, prepend `systemd-run --scope` to the command. This will reparent the process in its own scope separate from salt-minion, and will not be affected by restarting the minion service.

Parameters

- **cmd** (*str*) – The command to run. ex: `ls -lart /home`
- **cwd** (*str*) – The directory from which to execute the command. Defaults to the home directory of the user specified by `runas` (or the user under which Salt is running if `runas` is not specified).
- **group** (*str*) – Group to run command as. Not currently supported on Windows.
- **shell** (*str*) – Shell to execute under. Defaults to the system default shell.

- **`output_encoding`** (*`str`*) – Control the encoding used to decode the command's output.

  Note: This should not need to be used in most cases. By default, Salt will try to use the encoding detected from the system locale, and will fall back to UTF-8 if this fails. This should only need to be used in cases where the output of the command is encoded in something other than the system locale or UTF-8.

  To see the encoding Salt has detected from the system locale, check the *locale* line in the output of *`test.versions_report`*.

  New in version 2018.3.0.

- **`output_loglevel`** (*`str`*) – Control the loglevel at which the output from the command is logged to the minion log.

  Note: The command being run will still be logged at the `debug` loglevel regardless, unless `quiet` is used for this value.

- **`ignore_retcode`** (*`bool`*) – If the exit code of the command is nonzero, this is treated as an error condition, and the output from the command will be logged to the minion log. However, there are some cases where programs use the return code for signaling and a nonzero exit code doesn't necessarily mean failure. Pass this argument as `True` to skip logging the output if the command has a nonzero exit code.

- **`runas`** (*`str`*) – Specify an alternate user to run the command. The default behavior is to run as the user under which Salt is running. If running on a Windows minion you must also use the `password` argument, and the target user account must be in the Administrators group.

  Warning: For versions 2018.3.3 and above on macosx while using runas, to pass special characters to the command you need to escape the characters on the shell.

  Example:

  ```
  cmd.run_bg 'echo '\''h=\"baz\"'\''' runas=macuser
  ```

- **`password`** (*`str`*) – Windows only. Required when specifying `runas`. This parameter will be ignored on non-Windows platforms.

  New in version 2016.3.0.

- **`shell`** – Specify an alternate shell. Defaults to the system's default shell.

- **`python_shell`** (*`bool`*) – If False, let python handle the positional arguments. Set to True to use shell features, such as pipes or redirection.

- **`env`** (*`dict`*) – Environment variables to be set prior to execution.

  Note: When passing environment variables on the CLI, they should be passed as the string representation of a dictionary.

  ```
  salt myminion cmd.run_bg 'some command' env='{"FOO": "bar"}'
  ```

- **clean_env** (*bool*) – Attempt to clean out all other shell environment variables and set only those provided in the 'env' argument to this function.
- **prepend_path** (*str*) – $PATH segment to prepend (trailing ':' not necessary) to $PATH

  New in version 2018.3.0.
- **template** (*str*) – If this setting is applied then the named templating engine will be used to render the downloaded file. Currently jinja, mako, and wempy are supported.
- **umask** (*str*) – The umask (in octal) to use when running the command.
- **timeout** (*int*) – A timeout in seconds for the executed process to return.

> **Warning:** This function does not process commands through a shell unless the `python_shell` argument is set to `True`. This means that any shell-specific functionality such as 'echo' or the use of pipes, redirection or &&, should either be migrated to cmd.shell or have the python_shell=True flag set here.
>
> The use of `python_shell=True` means that the shell will accept _any_ input including potentially malicious commands such as 'good_command;rm -rf /'. Be absolutely certain that you have sanitized your input prior to using `python_shell=True`.

Parameters

- **success_retcodes** (*list*) –

  This parameter will be allow a list of non-zero return codes that should be considered a success. If the return code returned from the run matches any in the provided list, the return code will be overridden with zero.

  New in version 2019.2.0.
- **stdin_raw_newlines** (*bool*) –

  False If `True`, Salt will not automatically convert the characters `\\n` present in the `stdin` value to newlines.

  New in version 2019.2.0.

CLI Example:

```
salt '*' cmd.run_bg "fstrim-all"
```

The template arg can be set to 'jinja' or another supported template engine to render the command arguments before execution. For example:

```
salt '*' cmd.run_bg template=jinja "ls -l /tmp/{{grains.id}} | awk '/foo/{print \\
↪$2}'"
```

Specify an alternate shell with the shell parameter:

```
salt '*' cmd.run_bg "Get-ChildItem C:\\ " shell='powershell'
```

If an equal sign (=) appears in an argument to a Salt command it is interpreted as a keyword argument in the format `key=val`. That processing can be bypassed in order to pass an equal sign through to the remote shell command by manually specifying the kwarg:

```
salt '*' cmd.run_bg cmd='ls -lR / | sed -e s/=/:/g > /tmp/dontwait'
```

`salt.modules.cmdmod.`**`run_chroot`**(*root, cmd, cwd=None, stdin=None, runas=None, group=None, shell='/bin/sh', python_shell=True, binds=None, env=None, clean_env=False, template=None, rstrip=True, umask=None, output_encoding=None, output_loglevel='quiet', log_callback=None, hide_output=False, timeout=None, reset_system_locale=True, ignore_retcode=False, saltenv='base', use_vt=False, bg=False, success_retcodes=None, **kwargs*)

New in version 2014.7.0.

This function runs *`cmd.run_all`* wrapped within a chroot, with dev and proc mounted in the chroot

Parameters

- **`root`** (*`str`*) – Path to the root of the jail to use.
- **`stdin`** (*`str`*) – A string of standard input can be specified for the command to be run using the `stdin` parameter. This can be useful in cases where sensitive information must be read from standard input.:
- **`runas`** (*`str`*) – User to run script as.
- **`group`** (*`str`*) – Group to run script as.
- **`shell`** (*`str`*) – Shell to execute under. Defaults to the system default shell.
- **`cmd`** (*`str`*) – The command to run. ex: `ls -lart /home`
- **`cwd`** (*`str`*) – The directory from which to execute the command. Defaults to the home directory of the user specified by `runas` (or the user under which Salt is running if `runas` is not specified).
- **`runas`** – Specify an alternate user to run the command. The default behavior is to run as the user under which Salt is running. If running on a Windows minion you must also use the `password` argument, and the target user account must be in the Administrators group.
- **`shell`** – Specify an alternate shell. Defaults to the system's default shell.
- **`python_shell`** (*`bool`*) – If False, let python handle the positional arguments. Set to True to use shell features, such as pipes or redirection.
- **`binds`** (*`list`*) – List of directories that will be exported inside the chroot with the bind option.

  New in version 3000.
- **`env`** (*`dict`*) – Environment variables to be set prior to execution.

  Note: When passing environment variables on the CLI, they should be passed as the string representation of a dictionary.

  ```
  salt myminion cmd.run_chroot 'some command' env='{"FOO": "bar"}'
  ```
- **`clean_env`** (*`dict`*) – Attempt to clean out all other shell environment variables and set only those provided in the 'env' argument to this function.
- **`template`** (*`str`*) – If this setting is applied then the named templating engine will be used to render the downloaded file. Currently jinja, mako, and wempy are supported.
- **`rstrip`** (*`bool`*) – Strip all whitespace off the end of output before it is returned.
- **`umask`** (*`str`*) – The umask (in octal) to use when running the command.

- **output_encoding** (*str*) – Control the encoding used to decode the command's output.

  Note: This should not need to be used in most cases. By default, Salt will try to use the encoding detected from the system locale, and will fall back to UTF-8 if this fails. This should only need to be used in cases where the output of the command is encoded in something other than the system locale or UTF-8.

  To see the encoding Salt has detected from the system locale, check the *locale* line in the output of *test.versions_report*.

  New in version 2018.3.0.

- **output_loglevel** (*str*) – Control the loglevel at which the output from the command is logged to the minion log.

  Note: The command being run will still be logged at the `debug` loglevel regardless, unless `quiet` is used for this value.

- **ignore_retcode** (*bool*) – If the exit code of the command is nonzero, this is treated as an error condition, and the output from the command will be logged to the minion log. However, there are some cases where programs use the return code for signaling and a nonzero exit code doesn't necessarily mean failure. Pass this argument as `True` to skip logging the output if the command has a nonzero exit code.
- **hide_output** (*bool*) – If `True`, suppress stdout and stderr in the return data.

  Note: This is separate from `output_loglevel`, which only handles how Salt logs to the minion log.

  New in version 2018.3.0.

- **timeout** (*int*) – A timeout in seconds for the executed process to return.
- **use_vt** (*bool*) – Use VT utils (saltstack) to stream the command output more interactively to the console and the logs. This is experimental.

Parar str stdin A string of standard input can be specified for the command to be run using the `stdin` parameter. This can be useful in cases where sensitive information must be read from standard input.

success_retcodes: This parameter will be allow a list of

non-zero return codes that should be considered a success. If the return code returned from the run matches any in the provided list, the return code will be overridden with zero.

New in version 2019.2.0.

CLI Example:

```
salt '*' cmd.run_chroot /var/lib/lxc/container_name/rootfs 'sh /tmp/bootstrap.sh'
```

`salt.modules.cmdmod.`**`run_stderr`**(*cmd, cwd=None, stdin=None, runas=None, group=None, shell='/bin/sh', python_shell=None, env=None, clean_env=False, template=None, rstrip=True, umask=None, output_encoding=None, output_loglevel='debug', log_callback=None, hide_output=False, timeout=None, reset_system_locale=True, ignore_retcode=False, saltenv='base', use_vt=False, password=None, prepend_path=None, success_retcodes=None, **kwargs*)

Execute a command and only return the standard error

Parameters

- **`cmd`** (*`str`*) – The command to run. ex: `ls -lart /home`
- **`cwd`** (*`str`*) – The directory from which to execute the command. Defaults to the home directory of the user specified by `runas` (or the user under which Salt is running if `runas` is not specified).
- **`stdin`** (*`str`*) – A string of standard input can be specified for the command to be run using the `stdin` parameter. This can be useful in cases where sensitive information must be read from standard input.
- **`runas`** (*`str`*) – Specify an alternate user to run the command. The default behavior is to run as the user under which Salt is running. If running on a Windows minion you must also use the `password` argument, and the target user account must be in the Administrators group.

  Warning: For versions 2018.3.3 and above on macosx while using runas, to pass special characters to the command you need to escape the characters on the shell.

  Example:

  ```
  cmd.run_stderr 'echo '\''h=\"baz\"'\''' runas=macuser
  ```

- **`password`** (*`str`*) – Windows only. Required when specifying `runas`. This parameter will be ignored on non-Windows platforms.

  New in version 2016.3.0.

- **`group`** (*`str`*) – Group to run command as. Not currently supported on Windows.
- **`shell`** (*`str`*) – Specify an alternate shell. Defaults to the system's default shell.
- **`python_shell`** (*`bool`*) – If False, let python handle the positional arguments. Set to True to use shell features, such as pipes or redirection.
- **`env`** (*`dict`*) – Environment variables to be set prior to execution.

  Note: When passing environment variables on the CLI, they should be passed as the string representation of a dictionary.

  ```
  salt myminion cmd.run_stderr 'some command' env='{"FOO": "bar"}'
  ```

- **`clean_env`** (*`bool`*) – Attempt to clean out all other shell environment variables and set only those provided in the 'env' argument to this function.
- **`prepend_path`** (*`str`*) – $PATH segment to prepend (trailing ':' not necessary) to $PATH

  New in version 2018.3.0.

- **template** (*str*) – If this setting is applied then the named templating engine will be used to render the downloaded file. Currently jinja, mako, and wempy are supported.
- **rstrip** (*bool*) – Strip all whitespace off the end of output before it is returned.
- **umask** (*str*) – The umask (in octal) to use when running the command.
- **output_encoding** (*str*) – Control the encoding used to decode the command's output.

  Note: This should not need to be used in most cases. By default, Salt will try to use the encoding detected from the system locale, and will fall back to UTF-8 if this fails. This should only need to be used in cases where the output of the command is encoded in something other than the system locale or UTF-8.

  To see the encoding Salt has detected from the system locale, check the *locale* line in the output of `test.versions_report`.

  New in version 2018.3.0.

- **output_loglevel** (*str*) – Control the loglevel at which the output from the command is logged to the minion log.

  Note: The command being run will still be logged at the `debug` loglevel regardless, unless `quiet` is used for this value.

- **ignore_retcode** (*bool*) – If the exit code of the command is nonzero, this is treated as an error condition, and the output from the command will be logged to the minion log. However, there are some cases where programs use the return code for signaling and a nonzero exit code doesn't necessarily mean failure. Pass this argument as `True` to skip logging the output if the command has a nonzero exit code.
- **hide_output** (*bool*) – If `True`, suppress stdout and stderr in the return data.

  Note: This is separate from `output_loglevel`, which only handles how Salt logs to the minion log.

  New in version 2018.3.0.

- **timeout** (*int*) – A timeout in seconds for the executed process to return.
- **use_vt** (*bool*) – Use VT utils (saltstack) to stream the command output more interactively to the console and the logs. This is experimental.
- **success_retcodes** (*list*) –

  This parameter will be allow a list of non-zero return codes that should be considered a success. If the return code returned from the run matches any in the provided list, the return code will be overridden with zero.

  New in version 2019.2.0.

- **stdin_raw_newlines** (*bool*) –

  False If `True`, Salt will not automatically convert the characters `\n` present in the `stdin` value to newlines.

  New in version 2019.2.0.

CLI Example:

```
salt '*' cmd.run_stderr "ls -l | awk '/foo/{print \$2}'"
```

The template arg can be set to 'jinja' or another supported template engine to render the command arguments before execution. For example:

```
salt '*' cmd.run_stderr template=jinja "ls -l /tmp/{{grains.id}} | awk '/foo/
↪{print \$2}'"
```

A string of standard input can be specified for the command to be run using the `stdin` parameter. This can be useful in cases where sensitive information must be read from standard input.

```
salt '*' cmd.run_stderr "grep f" stdin='one\ntwo\nthree\nfour\nfive\n'
```

`salt.modules.cmdmod.`**`run_stdout`**(*cmd, cwd=None, stdin=None, runas=None, group=None, shell='/bin/sh', python_shell=None, env=None, clean_env=False, template=None, rstrip=True, umask=None, output_encoding=None, output_loglevel='debug', log_callback=None, hide_output=False, timeout=None, reset_system_locale=True, ignore_retcode=False, saltenv='base', use_vt=False, password=None, prepend_path=None, success_retcodes=None, **kwargs*)

Execute a command, and only return the standard out

Parameters

- **cmd** (*str*) – The command to run. ex: `ls -lart /home`
- **cwd** (*str*) – The directory from which to execute the command. Defaults to the home directory of the user specified by `runas` (or the user under which Salt is running if `runas` is not specified).
- **stdin** (*str*) – A string of standard input can be specified for the command to be run using the `stdin` parameter. This can be useful in cases where sensitive information must be read from standard input.
- **runas** (*str*) – Specify an alternate user to run the command. The default behavior is to run as the user under which Salt is running. If running on a Windows minion you must also use the `password` argument, and the target user account must be in the Administrators group.

  > **Warning:** For versions 2018.3.3 and above on macosx while using runas, to pass special characters to the command you need to escape the characters on the shell.
  >
  > Example:
  >
  > ```
  > cmd.run_stdout 'echo '\''h=\"baz\"'\''' runas=macuser
  > ```

- **password** (*str*) – Windows only. Required when specifying `runas`. This parameter will be ignored on non-Windows platforms.

  New in version 2016.3.0.

- **group** (*str*) – Group to run command as. Not currently supported on Windows.
- **shell** (*str*) – Specify an alternate shell. Defaults to the system's default shell.
- **python_shell** (*bool*) – If False, let python handle the positional arguments. Set to True to use shell features, such as pipes or redirection.

- **`env`** (*`dict`*) – Environment variables to be set prior to execution.

  Note: When passing environment variables on the CLI, they should be passed as the string representation of a dictionary.

  ```
  salt myminion cmd.run_stdout 'some command' env='{"FOO": "bar"}'
  ```

- **`clean_env`** (*`bool`*) – Attempt to clean out all other shell environment variables and set only those provided in the 'env' argument to this function.
- **`prepend_path`** (*`str`*) – $PATH segment to prepend (trailing ':' not necessary) to $PATH

  New in version 2018.3.0.

- **`template`** (*`str`*) – If this setting is applied then the named templating engine will be used to render the downloaded file. Currently jinja, mako, and wempy are supported.
- **`rstrip`** (*`bool`*) – Strip all whitespace off the end of output before it is returned.
- **`umask`** (*`str`*) – The umask (in octal) to use when running the command.
- **`output_encoding`** (*`str`*) – Control the encoding used to decode the command's output.

  Note: This should not need to be used in most cases. By default, Salt will try to use the encoding detected from the system locale, and will fall back to UTF-8 if this fails. This should only need to be used in cases where the output of the command is encoded in something other than the system locale or UTF-8.

  To see the encoding Salt has detected from the system locale, check the *locale* line in the output of *`test.versions_report`*.

  New in version 2018.3.0.

- **`output_loglevel`** (*`str`*) – Control the loglevel at which the output from the command is logged to the minion log.

  Note: The command being run will still be logged at the `debug` loglevel regardless, unless `quiet` is used for this value.

- **`ignore_retcode`** (*`bool`*) – If the exit code of the command is nonzero, this is treated as an error condition, and the output from the command will be logged to the minion log. However, there are some cases where programs use the return code for signaling and a nonzero exit code doesn't necessarily mean failure. Pass this argument as `True` to skip logging the output if the command has a nonzero exit code.
- **`hide_output`** (*`bool`*) – If `True`, suppress stdout and stderr in the return data.

  Note: This is separate from `output_loglevel`, which only handles how Salt logs to the minion log.

  New in version 2018.3.0.

- **`timeout`** (*`int`*) – A timeout in seconds for the executed process to return.

- **use_vt** (*bool*) – Use VT utils (saltstack) to stream the command output more interactively to the console and the logs. This is experimental.
- **success_retcodes** (*list*) –

  This parameter will be allow a list of non-zero return codes that should be considered a success. If the return code returned from the run matches any in the provided list, the return code will be overridden with zero.

  New in version 2019.2.0.
- **stdin_raw_newlines** (*bool*) –

  False If `True`, Salt will not automatically convert the characters `\n` present in the `stdin` value to newlines.

  New in version 2019.2.0.

CLI Example:

```
salt '*' cmd.run_stdout "ls -l | awk '/foo/{print \$2}'"
```

The template arg can be set to 'jinja' or another supported template engine to render the command arguments before execution. For example:

```
salt '*' cmd.run_stdout template=jinja "ls -l /tmp/{{grains.id}} | awk '/foo/
↪{print \$2}'"
```

A string of standard input can be specified for the command to be run using the `stdin` parameter. This can be useful in cases where sensitive information must be read from standard input.

```
salt '*' cmd.run_stdout "grep f" stdin='one\ntwo\nthree\nfour\nfive\n'
```

`salt.modules.cmdmod.`**`script`**(*source, args=None, cwd=None, stdin=None, runas=None, group=None, shell='/bin/sh', python_shell=None, env=None, template=None, umask=None, output_encoding=None, output_loglevel='debug', log_callback=None, hide_output=False, timeout=None, reset_system_locale=True, saltenv='base', use_vt=False, bg=False, password=None, success_retcodes=None, **kwargs*)

Download a script from a remote location and execute the script locally. The script can be located on the salt master file server or on an HTTP/FTP server.

The script will be executed directly, so it can be written in any available programming language.

Parameters

- **source** (*str*) – The location of the script to download. If the file is located on the master in the directory named spam, and is called eggs, the source string is salt://spam/eggs
- **args** (*str*) – String of command line args to pass to the script. Only used if no args are specified as part of the *name* argument. To pass a string containing spaces in YAML, you will need to doubly-quote it:

  ```
  salt myminion cmd.script salt://foo.sh "arg1 'arg two' arg3"
  ```
- **cwd** (*str*) – The directory from which to execute the command. Defaults to the home directory of the user specified by `runas` (or the user under which Salt is running if `runas` is not specified).

- **stdin** (*str*) – A string of standard input can be specified for the command to be run using the `stdin` parameter. This can be useful in cases where sensitive information must be read from standard input.
- **runas** (*str*) – Specify an alternate user to run the command. The default behavior is to run as the user under which Salt is running. If running on a Windows minion you must also use the `password` argument, and the target user account must be in the Administrators group.

Note: For Window's users, specifically Server users, it may be necessary to specify your runas user using the User Logon Name instead of the legacy logon name. Traditionally, logons would be in the following format.

```
Domain/user
```

In the event this causes issues when executing scripts, use the UPN format which looks like the following.

```
user@domain.local
```

More information <https://github.com/saltstack/salt/issues/55080>

- **password** (*str*) – Windows only. Required when specifying `runas`. This parameter will be ignored on non-Windows platforms.

  New in version 2016.3.0.
- **group** (*str*) – Group to run script as. Not currently supported on Windows.
- **shell** (*str*) – Specify an alternate shell. Defaults to the system's default shell.
- **python_shell** (*bool*) – If False, let python handle the positional arguments. Set to True to use shell features, such as pipes or redirection.
- **bg** (*bool*) – If True, run script in background and do not await or deliver it's results
- **env** (*dict*) – Environment variables to be set prior to execution.

Note: When passing environment variables on the CLI, they should be passed as the string representation of a dictionary.

```
salt myminion cmd.script 'some command' env='{"FOO": "bar"}'
```

- **template** (*str*) – If this setting is applied then the named templating engine will be used to render the downloaded file. Currently jinja, mako, and wempy are supported.
- **umask** (*str*) – The umask (in octal) to use when running the command.
- **output_encoding** (*str*) – Control the encoding used to decode the command's output.

Note: This should not need to be used in most cases. By default, Salt will try to use the encoding detected from the system locale, and will fall back to UTF-8 if this fails. This should only need to be used in cases where the output of the command is encoded in something other than the system locale or UTF-8.

To see the encoding Salt has detected from the system locale, check the *locale* line in the output of `test.versions_report`.

New in version 2018.3.0.

- **`output_loglevel`** (*str*) – Control the loglevel at which the output from the command is logged to the minion log.

  Note: The command being run will still be logged at the `debug` loglevel regardless, unless `quiet` is used for this value.

- **`ignore_retcode`** (*bool*) – If the exit code of the command is nonzero, this is treated as an error condition, and the output from the command will be logged to the minion log. However, there are some cases where programs use the return code for signaling and a nonzero exit code doesn't necessarily mean failure. Pass this argument as `True` to skip logging the output if the command has a nonzero exit code.
- **`hide_output`** (*bool*) – If `True`, suppress stdout and stderr in the return data.

  Note: This is separate from `output_loglevel`, which only handles how Salt logs to the minion log.

  New in version 2018.3.0.

- **`timeout`** (*int*) – If the command has not terminated after timeout seconds, send the subprocess sigterm, and if sigterm is ignored, follow up with sigkill
- **`use_vt`** (*bool*) – Use VT utils (saltstack) to stream the command output more interactively to the console and the logs. This is experimental.
- **`success_retcodes`** (*list*) –

  This parameter will be allow a list of non-zero return codes that should be considered a success. If the return code returned from the run matches any in the provided list, the return code will be overridden with zero.

  New in version 2019.2.0.

- **`stdin_raw_newlines`** (*bool*) –

  False If `True`, Salt will not automatically convert the characters `\n` present in the `stdin` value to newlines.

  New in version 2019.2.0.

CLI Example:

```
salt '*' cmd.script salt://scripts/runme.sh
salt '*' cmd.script salt://scripts/runme.sh 'arg1 arg2 "arg 3"'
salt '*' cmd.script salt://scripts/windows_task.ps1 args=' -Input c:\tmp\infile.
↪txt' shell='powershell'
```

```
salt '*' cmd.script salt://scripts/runme.sh stdin='one\ntwo\nthree\nfour\nfive\n'
```

`salt.modules.cmdmod.`**`script_retcode`**(*source, args=None, cwd=None, stdin=None, runas=None, group=None, shell='/bin/sh', python_shell=None, env=None, template='jinja', umask=None, timeout=None, reset_system_locale=True, saltenv='base', output_encoding=None, output_loglevel='debug', log_callback=None, use_vt=False, password=None, success_retcodes=None, **kwargs*)

Download a script from a remote location and execute the script locally. The script can be located on the salt master file server or on an HTTP/FTP server.

The script will be executed directly, so it can be written in any available programming language.

The script can also be formatted as a template, the default is jinja.

Only evaluate the script return code and do not block for terminal output

Parameters

- **source** (*str*) – The location of the script to download. If the file is located on the master in the directory named spam, and is called eggs, the source string is salt://spam/eggs
- **args** (*str*) – String of command line args to pass to the script. Only used if no args are specified as part of the *name* argument. To pass a string containing spaces in YAML, you will need to doubly-quote it: "arg1 'arg two' arg3"
- **cwd** (*str*) – The directory from which to execute the command. Defaults to the home directory of the user specified by `runas` (or the user under which Salt is running if `runas` is not specified).
- **stdin** (*str*) – A string of standard input can be specified for the command to be run using the `stdin` parameter. This can be useful in cases where sensitive information must be read from standard input.
- **runas** (*str*) – Specify an alternate user to run the command. The default behavior is to run as the user under which Salt is running. If running on a Windows minion you must also use the `password` argument, and the target user account must be in the Administrators group.
- **password** (*str*) – Windows only. Required when specifying `runas`. This parameter will be ignored on non-Windows platforms.

  New in version 2016.3.0.
- **group** (*str*) – Group to run script as. Not currently supported on Windows.
- **shell** (*str*) – Specify an alternate shell. Defaults to the system's default shell.
- **python_shell** (*bool*) – If False, let python handle the positional arguments. Set to True to use shell features, such as pipes or redirection.
- **env** (*dict*) – Environment variables to be set prior to execution.

  Note: When passing environment variables on the CLI, they should be passed as the string representation of a dictionary.

  ```
  salt myminion cmd.script_retcode 'some command' env='{"FOO": "bar"}
  ↪'
  ```
- **template** (*str*) – If this setting is applied then the named templating engine will be used to render the downloaded file. Currently jinja, mako, and wempy are supported.
- **umask** (*str*) – The umask (in octal) to use when running the command.
- **output_encoding** (*str*) – Control the encoding used to decode the command's output.

  Note: This should not need to be used in most cases. By default, Salt will try to use the encoding detected from the system locale, and will fall back to UTF-8 if this fails. This should only need to be used in cases where the output of the command is encoded in something other than the system locale or UTF-8.

To see the encoding Salt has detected from the system locale, check the *locale* line in the output of `test.versions_report`.

New in version 2018.3.0.

- **`output_loglevel`** (*str*) – Control the loglevel at which the output from the command is logged to the minion log.

Note: The command being run will still be logged at the `debug` loglevel regardless, unless `quiet` is used for this value.

- **`ignore_retcode`** (*bool*) – If the exit code of the command is nonzero, this is treated as an error condition, and the output from the command will be logged to the minion log. However, there are some cases where programs use the return code for signaling and a nonzero exit code doesn't necessarily mean failure. Pass this argument as `True` to skip logging the output if the command has a nonzero exit code.
- **`timeout`** (*int*) – If the command has not terminated after timeout seconds, send the subprocess sigterm, and if sigterm is ignored, follow up with sigkill
- **`use_vt`** (*bool*) – Use VT utils (saltstack) to stream the command output more interactively to the console and the logs. This is experimental.
- **`success_retcodes`** (*list*) –

This parameter will be allow a list of non-zero return codes that should be considered a success. If the return code returned from the run matches any in the provided list, the return code will be overridden with zero.

New in version 2019.2.0.

- **`stdin_raw_newlines`** (*bool*) –

False If `True`, Salt will not automatically convert the characters `\n` present in the `stdin` value to newlines.

New in version 2019.2.0.

CLI Example:

```
salt '*' cmd.script_retcode salt://scripts/runme.sh
salt '*' cmd.script_retcode salt://scripts/runme.sh 'arg1 arg2 "arg 3"'
salt '*' cmd.script_retcode salt://scripts/windows_task.ps1 args=' -Input
↪c:\tmp\infile.txt' shell='powershell'
```

A string of standard input can be specified for the command to be run using the `stdin` parameter. This can be useful in cases where sensitive information must be read from standard input.

```
salt '*' cmd.script_retcode salt://scripts/runme.sh stdin=
↪'one\ntwo\nthree\nfour\nfive\n'
```

`salt.modules.cmdmod.`**`shell`**(*cmd, cwd=None, stdin=None, runas=None, group=None, shell='/bin/sh', env=None, clean_env=False, template=None, rstrip=True, umask=None, output_encoding=None, output_loglevel='debug', log_callback=None, hide_output=False, timeout=None, reset_system_locale=True, ignore_retcode=False, saltenv='base', use_vt=False, bg=False, password=None, prepend_path=None, success_retcodes=None, **kwargs*)

Execute the passed command and return the output as a string.

New in version 2015.5.0.

Parameters

- **cmd** (*str*) – The command to run. ex: `ls -lart /home`
- **cwd** (*str*) – The directory from which to execute the command. Defaults to the home directory of the user specified by `runas` (or the user under which Salt is running if `runas` is not specified).
- **stdin** (*str*) – A string of standard input can be specified for the command to be run using the `stdin` parameter. This can be useful in cases where sensitive information must be read from standard input.
- **runas** (*str*) – Specify an alternate user to run the command. The default behavior is to run as the user under which Salt is running. If running on a Windows minion you must also use the `password` argument, and the target user account must be in the Administrators group.

  Warning: For versions 2018.3.3 and above on macosx while using runas, to pass special characters to the command you need to escape the characters on the shell.

  Example:

  ```
  cmd.shell 'echo '\''h=\"baz\"'\''' runas=macuser
  ```

- **group** (*str*) – Group to run command as. Not currently supported on Windows.
- **password** (*str*) – Windows only. Required when specifying `runas`. This parameter will be ignored on non-Windows platforms.

  New in version 2016.3.0.

- **shell** (*int*) – Shell to execute under. Defaults to the system default shell.
- **bg** (*bool*) – If True, run command in background and do not await or deliver its results
- **env** (*dict*) – Environment variables to be set prior to execution.

  Note: When passing environment variables on the CLI, they should be passed as the string representation of a dictionary.

  ```
  salt myminion cmd.shell 'some command' env='{"FOO": "bar"}'
  ```

- **clean_env** (*bool*) – Attempt to clean out all other shell environment variables and set only those provided in the 'env' argument to this function.
- **prepend_path** (*str*) – $PATH segment to prepend (trailing ':' not necessary) to $PATH

  New in version 2018.3.0.

- **template** (*str*) – If this setting is applied then the named templating engine will be used to render the downloaded file. Currently jinja, mako, and wempy are supported.
- **rstrip** (*bool*) – Strip all whitespace off the end of output before it is returned.
- **umask** (*str*) – The umask (in octal) to use when running the command.
- **output_encoding** (*str*) – Control the encoding used to decode the command's output.

  Note: This should not need to be used in most cases. By default, Salt will try to use the encoding detected from the system locale, and will fall back to UTF-8 if this fails. This should only need to be used in cases where the output of the command is encoded in something other than the system locale or UTF-8.

  To see the encoding Salt has detected from the system locale, check the *locale* line in the output of `test.versions_report`.

  New in version 2018.3.0.

- **output_loglevel** (*str*) – Control the loglevel at which the output from the command is logged to the minion log.

  Note: The command being run will still be logged at the `debug` loglevel regardless, unless `quiet` is used for this value.

- **ignore_retcode** (*bool*) – If the exit code of the command is nonzero, this is treated as an error condition, and the output from the command will be logged to the minion log. However, there are some cases where programs use the return code for signaling and a nonzero exit code doesn't necessarily mean failure. Pass this argument as `True` to skip logging the output if the command has a nonzero exit code.
- **hide_output** (*bool*) – If `True`, suppress stdout and stderr in the return data.

  Note: This is separate from `output_loglevel`, which only handles how Salt logs to the minion log.

  New in version 2018.3.0.

- **timeout** (*int*) – A timeout in seconds for the executed process to return.
- **use_vt** (*bool*) – Use VT utils (saltstack) to stream the command output more interactively to the console and the logs. This is experimental.

Warning: This passes the cmd argument directly to the shell without any further processing! Be absolutely sure that you have properly sanitized the command passed to this function and do not use untrusted inputs.

Parameters

- **success_retcodes** (*list*) –

  This parameter will be allow a list of non-zero return codes that should be considered a success. If the return code returned from the run matches any in the provided list, the return code will be overridden with zero.

  New in version 2019.2.0.

- **`stdin_raw_newlines`** (*`bool`*) –

  False If `True`, Salt will not automatically convert the characters `\n` present in the `stdin` value to newlines.

  New in version 2019.2.0.

CLI Example:

```
salt '*' cmd.shell "ls -l | awk '/foo/{print \$2}'"
```

The template arg can be set to 'jinja' or another supported template engine to render the command arguments before execution. For example:

```
salt '*' cmd.shell template=jinja "ls -l /tmp/{{grains.id}} | awk '/foo/{print \
↪$2}'"
```

Specify an alternate shell with the shell parameter:

```
salt '*' cmd.shell "Get-ChildItem C:\ " shell='powershell'
```

A string of standard input can be specified for the command to be run using the `stdin` parameter. This can be useful in cases where sensitive information must be read from standard input.

```
salt '*' cmd.shell "grep f" stdin='one\ntwo\nthree\nfour\nfive\n'
```

If an equal sign (=) appears in an argument to a Salt command it is interpreted as a keyword argument in the format `key=val`. That processing can be bypassed in order to pass an equal sign through to the remote shell command by manually specifying the kwarg:

```
salt '*' cmd.shell cmd='sed -e s/=/:/g'
```

`salt.modules.cmdmod.`**`shell_info`**(*shell*, *list_modules=False*)

New in version 2016.11.0.

Provides information about a shell or script languages which often use `#!`. The values returned are dependent on the shell or scripting languages all return the `installed`, `path`, `version`, `version_raw`

Parameters

- **`shell`** (*`str`*) – Name of the shell. Support shells/script languages include
- **`cmd, perl, php, powershell, python, ruby and zsh`** (*`bash,`*) –
- **`list_modules`** (*`bool`*) – True to list modules available to the shell.
- **`only lists powershell modules.`** (*`Currently`*) –

Returns A dictionary of information about the shell

Return type dict

```
{'version': '<2 or 3 numeric components dot-separated>',
 'version_raw': '<full version string>',
 'path': '<full path to binary>',
 'installed': <True, False or None>,
 '<attribute>': '<attribute value>'}
```

Note:

- `installed` is always returned, if `None` or `False` also returns error and may also return `stdout` for diagnostics.
- `version` is for use in determine if a shell/script language has a particular feature set, not for package management.
- The shell must be within the executable search path.

CLI Example:

```
salt '*' cmd.shell_info bash
salt '*' cmd.shell_info powershell
```

Codeauthor Damon Atkins <https://github.com/damon-atkins>

`salt.modules.cmdmod.`**`shells`**`()`

Lists the valid shells on this system via the /etc/shells file

New in version 2015.5.0.

CLI Example:

```
salt '*' cmd.shells
```

`salt.modules.cmdmod.`**`tty`**`(`*device*, *echo=''*`)`

Echo a string to a specific tty

CLI Example:

```
salt '*' cmd.tty tty0 'This is a test'
salt '*' cmd.tty pts3 'This is a test'
```

`salt.modules.cmdmod.`**`which`**`(`*cmd*`)`

Returns the path of an executable available on the minion, None otherwise

CLI Example:

```
salt '*' cmd.which cat
```

`salt.modules.cmdmod.`**`which_bin`**`(`*cmds*`)`

Returns the first command found in a list of commands

CLI Example:

```
salt '*' cmd.which_bin '[pip2, pip, pip-python]'
```

21.9.83 salt.modules.composer

Use composer to install PHP dependencies for a directory

`salt.modules.composer.`**`did_composer_install`**`(`*dir*`)`

Test to see if the vendor directory exists in this directory

dir Directory location of the composer.json file

CLI Example:

```
salt '*' composer.did_composer_install /var/www/application
```

`salt.modules.composer.`**`install`**(*directory*, *composer=None*, *php=None*, *runas=None*, *prefer_source=None*, *prefer_dist=None*, *no_scripts=None*, *no_plugins=None*, *optimize=None*, *no_dev=None*, *quiet=False*, *composer_home='/root'*, *env=None*)

Install composer dependencies for a directory.

If composer has not been installed globally making it available in the system PATH & making it executable, the `composer` and `php` parameters will need to be set to the location of the executables.

directory Directory location of the composer.json file.

composer Location of the composer.phar file. If not set composer will just execute "composer" as if it is installed globally. (i.e. /path/to/composer.phar)

php Location of the php executable to use with composer. (i.e. /usr/bin/php)

runas Which system user to run composer as.

prefer_source –prefer-source option of composer.

prefer_dist –prefer-dist option of composer.

no_scripts –no-scripts option of composer.

no_plugins –no-plugins option of composer.

optimize –optimize-autoloader option of composer. Recommended for production.

no_dev –no-dev option for composer. Recommended for production.

quiet –quiet option for composer. Whether or not to return output from composer.

composer_home $COMPOSER_HOME environment variable

env A list of environment variables to be set prior to execution.

CLI Example:

```
salt '*' composer.install /var/www/application

salt '*' composer.install /var/www/application              no_dev=True
↪optimize=True
```

`salt.modules.composer.`**`selfupdate`**(*composer=None*, *php=None*, *runas=None*, *quiet=False*, *composer_home='/root'*)

Update composer itself.

If composer has not been installed globally making it available in the system PATH & making it executable, the `composer` and `php` parameters will need to be set to the location of the executables.

composer Location of the composer.phar file. If not set composer will just execute "composer" as if it is installed globally. (i.e. /path/to/composer.phar)

php Location of the php executable to use with composer. (i.e. /usr/bin/php)

runas Which system user to run composer as.

quiet –quiet option for composer. Whether or not to return output from composer.

composer_home $COMPOSER_HOME environment variable

CLI Example:

```
salt '*' composer.selfupdate
```

`salt.modules.composer.`**`update`**(*directory*, *composer=None*, *php=None*, *runas=None*, *prefer_source=None*, *prefer_dist=None*, *no_scripts=None*, *no_plugins=None*, *optimize=None*, *no_dev=None*, *quiet=False*, *composer_home='/root'*, *env=None*)

Update composer dependencies for a directory.

If *composer install* has not yet been run, this runs *composer install* instead.

If composer has not been installed globally making it available in the system PATH & making it executable, the `composer` and `php` parameters will need to be set to the location of the executables.

directory Directory location of the composer.json file.

composer Location of the composer.phar file. If not set composer will just execute "composer" as if it is installed globally. (i.e. /path/to/composer.phar)

php Location of the php executable to use with composer. (i.e. /usr/bin/php)

runas Which system user to run composer as.

prefer_source –prefer-source option of composer.

prefer_dist –prefer-dist option of composer.

no_scripts –no-scripts option of composer.

no_plugins –no-plugins option of composer.

optimize –optimize-autoloader option of composer. Recommended for production.

no_dev –no-dev option for composer. Recommended for production.

quiet –quiet option for composer. Whether or not to return output from composer.

composer_home $COMPOSER_HOME environment variable

env A list of environment variables to be set prior to execution.

CLI Example:

```
salt '*' composer.update /var/www/application

salt '*' composer.update /var/www/application              no_dev=True
→optimize=True
```

21.9.84 salt.modules.config

Return config information

`salt.modules.config.`**`backup_mode`**(*backup=''*)

Return the backup mode

CLI Example:

```
salt '*' config.backup_mode
```

`salt.modules.config.`**`dot_vals`**(*value*)

Pass in a configuration value that should be preceded by the module name and a dot, this will return a list of all read key/value pairs

CLI Example:

```
salt '*' config.dot_vals host
```

`salt.modules.config.`**`gather_bootstrap_script`**`(`*bootstrap=None*`)`

Download the salt-bootstrap script, and return its location

bootstrap URL of alternate bootstrap script

CLI Example:

```
salt '*' config.gather_bootstrap_script
```

`salt.modules.config.`**`get`**`(`*key, default='', delimiter=':', merge=None, omit_opts=False, omit_pillar=False, omit_master=False, omit_grains=False*`)`

Attempt to retrieve the named value from the minion config file, pillar, grains or the master config. If the named value is not available, return the value specified by the `default` argument. If this argument is not specified, `default` falls back to an empty string.

Values can also be retrieved from nested dictionaries. Assume the below data structure:

```
{'pkg': {'apache': 'httpd'}}
```

To retrieve the value associated with the `apache` key, in the sub-dictionary corresponding to the `pkg` key, the following command can be used:

```
salt myminion config.get pkg:apache
```

The `:` (colon) is used to represent a nested dictionary level.

Changed in version 2015.5.0: The `delimiter` argument was added, to allow delimiters other than `:` to be used.

This function traverses these data stores in this order, returning the first match found:

- Minion configuration
- Minion's grains
- Minion's pillar data
- Master configuration (requires `pillar_opts` to be set to `True` in Minion config file in order to work)

This means that if there is a value that is going to be the same for the majority of minions, it can be configured in the Master config file, and then overridden using the grains, pillar, or Minion config file.

Adding config options to the Master or Minion configuration file is easy:

```
my-config-option: value
cafe-menu:
  - egg and bacon
  - egg sausage and bacon
  - egg and spam
  - egg bacon and spam
  - egg bacon sausage and spam
  - spam bacon sausage and spam
  - spam egg spam spam bacon and spam
  - spam sausage spam spam bacon spam tomato and spam
```

Note: Minion configuration options built into Salt (like those defined *here*) will *always* be defined in the Minion configuration and thus *cannot be overridden by grains or pillar data*. However, additional (user-defined) configuration options (as in the above example) will not be in the Minion configuration by default and thus can be overridden using grains/pillar data by leaving the option out of the minion config file.

Arguments

delimiter New in version 2015.5.0.

Override the delimiter used to separate nested levels of a data structure.

merge New in version 2015.5.0.

If passed, this parameter will change the behavior of the function so that, instead of traversing each data store above in order and returning the first match, the data stores are first merged together and then searched. The pillar data is merged into the master config data, then the grains are merged, followed by the Minion config data. The resulting data structure is then searched for a match. This allows for configurations to be more flexible.

Note: The merging described above does not mean that grain data will end up in the Minion's pillar data, or pillar data will end up in the master config data, etc. The data is just combined for the purposes of searching an amalgam of the different data stores.

The supported merge strategies are as follows:

- **recurse** - If a key exists in both dictionaries, and the new value is not a dictionary, it is replaced. Otherwise, the sub-dictionaries are merged together into a single dictionary, recursively on down, following the same criteria. For example:

```
>>> dict1 = {'foo': {'bar': 1, 'qux': True},
             'hosts': ['a', 'b', 'c'],
             'only_x': None}
>>> dict2 = {'foo': {'baz': 2, 'qux': False},
             'hosts': ['d', 'e', 'f'],
             'only_y': None}
>>> merged
{'foo': {'bar': 1, 'baz': 2, 'qux': False},
 'hosts': ['d', 'e', 'f'],
 'only_dict1': None,
 'only_dict2': None}
```

- **overwrite** - If a key exists in the top level of both dictionaries, the new value completely overwrites the old. For example:

```
>>> dict1 = {'foo': {'bar': 1, 'qux': True},
             'hosts': ['a', 'b', 'c'],
             'only_x': None}
>>> dict2 = {'foo': {'baz': 2, 'qux': False},
             'hosts': ['d', 'e', 'f'],
             'only_y': None}
>>> merged
{'foo': {'baz': 2, 'qux': False},
 'hosts': ['d', 'e', 'f'],
 'only_dict1': None,
 'only_dict2': None}
```

CLI Example:

```
salt '*' config.get pkg:apache
salt '*' config.get lxc.container_profile:centos merge=recurse
```

`salt.modules.config.`**`items`**`()`

Return the complete config from the currently running minion process. This includes defaults for values not set in the config file.

CLI Example:

```
salt '*' config.items
```

`salt.modules.config.`**`manage_mode`**`(`*mode*`)`

Return a mode value, normalized to a string

CLI Example:

```
salt '*' config.manage_mode
```

`salt.modules.config.`**`merge`**`(`*value, default=", omit_opts=False, omit_master=False, omit_pillar=False*`)`

Retrieves an option based on key, merging all matches.

Same as `option()` except that it merges all matches, rather than taking the first match.

CLI Example:

```
salt '*' config.merge schedule
```

`salt.modules.config.`**`option`**`(`*value, default=None, omit_opts=False, omit_grains=False, omit_pillar=False, omit_master=False, omit_all=False, wildcard=False*`)`

Returns the setting for the specified config value. The priority for matches is the same as in `config.get`, only this function does not recurse into nested data structures. Another difference between this function and `config.get` is that it comes with a set of "sane defaults". To view these, you can run the following command:

```
salt '*' config.option '*' omit_all=True wildcard=True
```

default The default value if no match is found. If not specified, then the fallback default will be an empty string, unless `wildcard=True`, in which case the return will be an empty dictionary.

omit_opts [False] Pass as `True` to exclude matches from the minion configuration file

omit_grains [False] Pass as `True` to exclude matches from the grains

omit_pillar [False] Pass as `True` to exclude matches from the pillar data

omit_master [False] Pass as `True` to exclude matches from the master configuration file

omit_all [True] Shorthand to omit all of the above and return matches only from the "sane defaults".

New in version 3000.

wildcard [False] If used, this will perform pattern matching on keys. Note that this will also significantly change the return data. Instead of only a value being returned, a dictionary mapping the matched keys to their values is returned. For example, using `wildcard=True` with a `key` of `'foo.ba*` could return a dictionary like so:

```
{'foo.bar': True, 'foo.baz': False}
```

New in version 3000.

CLI Example:

```
salt '*' config.option redis.host
```

`salt.modules.config.`**`valid_fileproto`**`(`*uri*`)`

Returns a boolean value based on whether or not the URI passed has a valid remote file protocol designation

CLI Example:

```
salt '*' config.valid_fileproto salt://path/to/file
```

21.9.85 salt.modules.consul

Interact with Consul

https://www.consul.io

`salt.modules.consul.`**`acl_clone`**`(`*consul_url=None, token=None, **kwargs*`)`

Information about an ACL token.

Parameters

- **`consul_url`** – The Consul server URL.
- **`id`** – Unique identifier for the ACL to update.

Returns Boolean, message of success or failure, and new ID of cloned ACL.

CLI Example:

```
salt '*' consul.acl_info id='c1c4d223-91cb-3d1f-1ee8-f2af9e7b6716'
```

`salt.modules.consul.`**`acl_create`**`(`*consul_url=None, token=None, **kwargs*`)`

Create a new ACL token.

Parameters

- **`consul_url`** – The Consul server URL.
- **`name`** – Meaningful indicator of the ACL's purpose.
- **`type`** – Type is either client or management. A management token is comparable to a root user and has the ability to perform any action including creating, modifying, and deleting ACLs.
- **`rules`** – The Consul server URL.

Returns Boolean & message of success or failure.

CLI Example:

```
salt '*' consul.acl_create
```

`salt.modules.consul.`**`acl_delete`**`(`*consul_url=None, token=None, **kwargs*`)`

Delete an ACL token.

Parameters

- **`consul_url`** – The Consul server URL.
- **`id`** – Unique identifier for the ACL to update.

Returns Boolean & message of success or failure.

CLI Example:

```
salt '*' consul.acl_delete id='c1c4d223-91cb-3d1f-1ee8-f2af9e7b6716'
```

`salt.modules.consul.`**`acl_info`**(*consul_url=None*, ***kwargs*)

Information about an ACL token.

Parameters

- **`consul_url`** – The Consul server URL.
- **`id`** – Unique identifier for the ACL to update.

Returns Information about the ACL requested.

CLI Example:

```
salt '*' consul.acl_info id='c1c4d223-91cb-3d1f-1ee8-f2af9e7b6716'
```

`salt.modules.consul.`**`acl_list`**(*consul_url=None*, *token=None*, ***kwargs*)

List the ACL tokens.

Parameters **`consul_url`** – The Consul server URL.

Returns List of ACLs

CLI Example:

```
salt '*' consul.acl_list
```

`salt.modules.consul.`**`acl_update`**(*consul_url=None*, *token=None*, ***kwargs*)

Update an ACL token.

Parameters

- **`consul_url`** – The Consul server URL.
- **`name`** – Meaningful indicator of the ACL's purpose.
- **`id`** – Unique identifier for the ACL to update.
- **`type`** – Type is either client or management. A management token is comparable to a root user and has the ability to perform any action including creating, modifying, and deleting ACLs.
- **`rules`** – The Consul server URL.

Returns Boolean & message of success or failure.

CLI Example:

```
salt '*' consul.acl_update
```

`salt.modules.consul.`**`agent_check_deregister`**(*consul_url=None*, *token=None*, *checkid=None*)

The agent will take care of deregistering the check from the Catalog.

Parameters

- **`consul_url`** – The Consul server URL.
- **`checkid`** – The ID of the check to deregister from Consul.

Returns Boolean and message indicating success or failure.

CLI Example:

```
salt '*' consul.agent_check_deregister checkid='Memory Utilization'
```

`salt.modules.consul.`**`agent_check_fail`**(*consul_url=None*, *token=None*, *checkid=None*, ***kwargs*)

This endpoint is used with a check that is of the TTL type. When this is called, the status of the check is set to critical and the TTL clock is reset.

Parameters

- **`consul_url`** – The Consul server URL.
- **`checkid`** – The ID of the check to deregister from Consul.
- **`note`** – A human-readable message with the status of the check.

Returns Boolean and message indicating success or failure.

CLI Example:

```
salt '*' consul.agent_check_fail checkid='redis_check1' note='Forcing check into
↪critical state.'
```

`salt.modules.consul.`**`agent_check_pass`**(*consul_url=None*, *token=None*, *checkid=None*, ***kwargs*)

This endpoint is used with a check that is of the TTL type. When this is called, the status of the check is set to passing and the TTL clock is reset.

Parameters

- **`consul_url`** – The Consul server URL.
- **`checkid`** – The ID of the check to mark as passing.
- **`note`** – A human-readable message with the status of the check.

Returns Boolean and message indicating success or failure.

CLI Example:

```
salt '*' consul.agent_check_pass checkid='redis_check1' note='Forcing check into
↪passing state.'
```

`salt.modules.consul.`**`agent_check_register`**(*consul_url=None*, *token=None*, ***kwargs*)

The register endpoint is used to add a new check to the local agent.

Parameters

- **`consul_url`** – The Consul server URL.
- **`name`** – The description of what the check is for.
- **`id`** – The unique name to use for the check, if not provided 'name' is used.
- **`notes`** – Human readable description of the check.
- **`script`** – If script is provided, the check type is a script, and Consul will evaluate that script based on the interval parameter.
- **`http`** – Check will perform an HTTP GET request against the value of HTTP (expected to be a URL) based on the interval parameter.
- **`ttl`** – If a TTL type is used, then the TTL update endpoint must be used periodically to update the state of the check.
- **`interval`** – Interval at which the check should run.

Returns Boolean and message indicating success or failure.

CLI Example:

```
salt '*' consul.agent_check_register name='Memory Utilization' script='/usr/local/
↪bin/check_mem.py' interval='15s'
```

`salt.modules.consul.`**`agent_check_warn`**(*consul_url=None*, *token=None*, *checkid=None*, ***kwargs*)

This endpoint is used with a check that is of the TTL type. When this is called, the status of the check is set to warning and the TTL clock is reset.

Parameters

- **`consul_url`** – The Consul server URL.
- **`checkid`** – The ID of the check to deregister from Consul.
- **`note`** – A human-readable message with the status of the check.

Returns Boolean and message indicating success or failure.

CLI Example:

```
salt '*' consul.agent_check_warn checkid='redis_check1' note='Forcing check into
↪warning state.'
```

`salt.modules.consul.`**`agent_checks`**(*consul_url=None*, *token=None*)

Returns the checks the local agent is managing

Parameters **`consul_url`** – The Consul server URL.

Returns Returns the checks the local agent is managing

CLI Example:

```
salt '*' consul.agent_checks
```

`salt.modules.consul.`**`agent_join`**(*consul_url=None*, *token=None*, *address=None*, ***kwargs*)

Triggers the local agent to join a node

Parameters

- **`consul_url`** – The Consul server URL.
- **`address`** – The address for the agent to connect to.
- **`wan`** – Causes the agent to attempt to join using the WAN pool.

Returns Boolean and message indicating success or failure.

CLI Example:

```
salt '*' consul.agent_join address='192.168.1.1'
```

`salt.modules.consul.`**`agent_leave`**(*consul_url=None*, *token=None*, *node=None*)

Used to instruct the agent to force a node into the left state.

Parameters

- **`consul_url`** – The Consul server URL.
- **`node`** – The node the agent will force into left state

Returns Boolean and message indicating success or failure.

CLI Example:

```
salt '*' consul.agent_leave node='web1.example.com'
```

`salt.modules.consul.`**`agent_maintenance`**(*consul_url=None*, *token=None*, ***kwargs*)

Manages node maintenance mode

Parameters

- **`consul_url`** – The Consul server URL.
- **`enable`** – The enable flag is required. Acceptable values are either true (to enter maintenance mode) or false (to resume normal operation).
- **`reason`** – If provided, its value should be a text string explaining the reason for placing the node into maintenance mode.

Returns Boolean and message indicating success or failure.

CLI Example:

```
salt '*' consul.agent_maintenance enable='False' reason='Upgrade in progress'
```

`salt.modules.consul.`**`agent_members`**(*consul_url=None*, *token=None*, ***kwargs*)

Returns the members as seen by the local serf agent

Parameters **`consul_url`** – The Consul server URL.

Returns Returns the members as seen by the local serf agent

CLI Example:

```
salt '*' consul.agent_members
```

`salt.modules.consul.`**`agent_self`**(*consul_url=None*, *token=None*)

Returns the local node configuration

Parameters **`consul_url`** – The Consul server URL.

Returns Returns the local node configuration

CLI Example:

```
salt '*' consul.agent_self
```

`salt.modules.consul.`**`agent_service_deregister`**(*consul_url=None*, *token=None*, *serviceid=None*)

Used to remove a service.

Parameters

- **`consul_url`** – The Consul server URL.
- **`serviceid`** – A serviceid describing the service.

Returns Boolean and message indicating success or failure.

CLI Example:

```
salt '*' consul.agent_service_deregister serviceid='redis'
```

`salt.modules.consul.`**`agent_service_maintenance`**(*consul_url=None*, *token=None*, *serviceid=None*, ***kwargs*)

Used to place a service into maintenance mode.

Parameters

- **consul_url** – The Consul server URL.
- **serviceid** – A name of the service.
- **enable** – Whether the service should be enabled or disabled.
- **reason** – A human readable message of why the service was enabled or disabled.

Returns Boolean and message indicating success or failure.

CLI Example:

```
salt '*' consul.agent_service_deregister serviceid='redis' enable='True' reason=
↪'Down for upgrade'
```

`salt.modules.consul.`**`agent_service_register`**(*consul_url=None, token=None, **kwargs*)

The used to add a new service, with an optional health check, to the local agent.

Parameters

- **consul_url** – The Consul server URL.
- **name** – A name describing the service.
- **address** – The address used by the service, defaults to the address of the agent.
- **port** – The port used by the service.
- **id** – Unique ID to identify the service, if not provided the value of the name parameter is used.
- **tags** – Identifying tags for service, string or list.
- **script** – If script is provided, the check type is a script, and Consul will evaluate that script based on the interval parameter.
- **http** – Check will perform an HTTP GET request against the value of HTTP (expected to be a URL) based on the interval parameter.
- **check_ttl** – If a TTL type is used, then the TTL update endpoint must be used periodically to update the state of the check.
- **check_interval** – Interval at which the check should run.

Returns Boolean and message indicating success or failure.

CLI Example:

```
salt '*' consul.agent_service_register name='redis' tags='["master", "v1"]'
↪address="127.0.0.1" port="8080" check_script="/usr/local/bin/check_redis.py"
↪interval="10s"
```

`salt.modules.consul.`**`agent_services`**(*consul_url=None, token=None*)

Returns the services the local agent is managing

Parameters **consul_url** – The Consul server URL.

Returns Returns the services the local agent is managing

CLI Example:

```
salt '*' consul.agent_services
```

`salt.modules.consul.`**`catalog_datacenters`**(*consul_url=None*, *token=None*)

Return list of available datacenters from catalog.

Parameters **`consul_url`** – The Consul server URL.

Returns The list of available datacenters.

CLI Example:

```
salt '*' consul.catalog_datacenters
```

`salt.modules.consul.`**`catalog_deregister`**(*consul_url=None*, *token=None*, ***kwargs*)

Deregisters a node, service, or check

Parameters

- **`consul_url`** – The Consul server URL.
- **`node`** – The node to deregister.
- **`datacenter`** – By default, the datacenter of the agent is queried; however, the dc can be provided using the "dc" parameter.
- **`checkid`** – The ID of the health check to deregister.
- **`serviceid`** – The ID of the service to deregister.

Returns Boolean & message of success or failure.

CLI Example:

```
salt '*' consul.catalog_register node='node1' serviceid='redis_server1' checkid=
↪'redis_check1'
```

`salt.modules.consul.`**`catalog_node`**(*consul_url=None*, *token=None*, *node=None*, ***kwargs*)

Information about the registered node.

Parameters

- **`consul_url`** – The Consul server URL.
- **`node`** – The node to request information about.
- **`dc`** – By default, the datacenter of the agent is queried; however, the dc can be provided using the "dc" parameter.

Returns Information about the requested node.

CLI Example:

```
salt '*' consul.catalog_service service='redis'
```

`salt.modules.consul.`**`catalog_nodes`**(*consul_url=None*, *token=None*, ***kwargs*)

Return list of available nodes from catalog.

Parameters

- **`consul_url`** – The Consul server URL.
- **`dc`** – By default, the datacenter of the agent is queried; however, the dc can be provided using the "dc" parameter.

Returns The list of available nodes.

CLI Example:

```
salt '*' consul.catalog_nodes
```

`salt.modules.consul.`**`catalog_register`**(*consul_url=None*, *token=None*, ***kwargs*)
Registers a new node, service, or check

Parameters

- **`consul_url`** – The Consul server URL.
- **`dc`** – By default, the datacenter of the agent is queried; however, the dc can be provided using the "dc" parameter.
- **`node`** – The node to register.
- **`address`** – The address of the node.
- **`service`** – The service that will be registered.
- **`service_address`** – The address that the service listens on.
- **`service_port`** – The port for the service.
- **`service_id`** – A unique identifier for the service, if this is not provided "name" will be used.
- **`service_tags`** – Any tags associated with the service.
- **`check`** – The name of the health check to register
- **`check_status`** – The initial status of the check, must be one of unknown, passing, warning, or critical.
- **`check_service`** – The service that the check is performed against.
- **`check_id`** – Unique identifier for the service.
- **`check_notes`** – An opaque field that is meant to hold human-readable text.

Returns Boolean & message of success or failure.

CLI Example:

```
salt '*' consul.catalog_register node='node1' address='192.168.1.1' service='redis
↪' service_address='127.0.0.1' service_port='8080' service_id='redis_server1'
```

`salt.modules.consul.`**`catalog_service`**(*consul_url=None*, *token=None*, *service=None*, ***kwargs*)
Information about the registered service.

Parameters

- **`consul_url`** – The Consul server URL.
- **`dc`** – By default, the datacenter of the agent is queried; however, the dc can be provided using the "dc" parameter.
- **`tag`** – Filter returned services with tag parameter.

Returns Information about the requested service.

CLI Example:

```
salt '*' consul.catalog_service service='redis'
```

`salt.modules.consul.`**`catalog_services`**(*consul_url=None*, *token=None*, ***kwargs*)
Return list of available services rom catalog.

Parameters

- **consul_url** – The Consul server URL.
- **dc** – By default, the datacenter of the agent is queried; however, the dc can be provided using the "dc" parameter.

Returns The list of available services.

CLI Example:

```
salt '*' consul.catalog_services
```

salt.modules.consul.**delete**(*consul_url=None*, *token=None*, *key=None*, ***kwargs*)

Delete values from Consul

Parameters

- **consul_url** – The Consul server URL.
- **key** – The key to use as the starting point for the list.
- **recurse** – Delete values recursively beginning at the value of key.
- **cas** – This flag is used to turn the DELETE into a Check-And-Set operation.

Returns Boolean & message of success or failure.

CLI Example:

```
salt '*' consul.delete key='web'
salt '*' consul.delete key='web' recurse='True'
```

salt.modules.consul.**event_fire**(*consul_url=None*, *token=None*, *name=None*, ***kwargs*)

List the ACL tokens.

Parameters

- **consul_url** – The Consul server URL.
- **name** – The name of the event to fire.
- **dc** – By default, the datacenter of the agent is queried; however, the dc can be provided using the "dc" parameter.
- **node** – Filter by node name.
- **service** – Filter by service name.
- **tag** – Filter by tag name.

Returns List of ACLs

CLI Example:

```
salt '*' consul.event_fire name='deploy'
```

salt.modules.consul.**event_list**(*consul_url=None*, *token=None*, ***kwargs*)

List the recent events.

Parameters

- **consul_url** – The Consul server URL.
- **name** – The name of the event to fire.

Returns List of ACLs

CLI Example:

```
salt '*' consul.event_list
```

`salt.modules.consul.`**`get`**(*consul_url=None*, *key=None*, *token=None*, *recurse=False*, *decode=False*, *raw=False*)

Get key from Consul

Parameters

- **`consul_url`** – The Consul server URL.
- **`key`** – The key to use as the starting point for the list.
- **`recurse`** – Return values recursively beginning at the value of key.
- **`decode`** – By default values are stored as Base64 encoded values, decode will return the whole key with the value decoded.
- **`raw`** – Simply return the decoded value of the key.

Returns The keys in Consul.

CLI Example:

```
salt '*' consul.get key='web/key1'
salt '*' consul.get key='web' recurse=True
salt '*' consul.get key='web' recurse=True decode=True
```

By default values stored in Consul are base64 encoded, passing the decode option will show them as the decoded values.

```
salt '*' consul.get key='web' recurse=True decode=True raw=True
```

By default Consul will return other information about the key, the raw option will return only the raw value.

`salt.modules.consul.`**`health_checks`**(*consul_url=None*, *token=None*, *service=None*, ***kwargs*)

Health information about the registered service.

Parameters

- **`consul_url`** – The Consul server URL.
- **`service`** – The service to request health information about.
- **`dc`** – By default, the datacenter of the agent is queried; however, the dc can be provided using the "dc" parameter.

Returns Health information about the requested node.

CLI Example:

```
salt '*' consul.health_checks service='redis1'
```

`salt.modules.consul.`**`health_node`**(*consul_url=None*, *token=None*, *node=None*, ***kwargs*)

Health information about the registered node.

Parameters

- **`consul_url`** – The Consul server URL.
- **`node`** – The node to request health information about.
- **`dc`** – By default, the datacenter of the agent is queried; however, the dc can be provided using the "dc" parameter.

Returns Health information about the requested node.

CLI Example:

```
salt '*' consul.health_node node='node1'
```

`salt.modules.consul.`**`health_service`**(*consul_url=None*, *token=None*, *service=None*, ***kwargs*)

Health information about the registered service.

Parameters

- **`consul_url`** – The Consul server URL.
- **`service`** – The service to request health information about.
- **`dc`** – By default, the datacenter of the agent is queried; however, the dc can be provided using the "dc" parameter.
- **`tag`** – Filter returned services with tag parameter.
- **`passing`** – Filter results to only nodes with all checks in the passing state.

Returns Health information about the requested node.

CLI Example:

```
salt '*' consul.health_service service='redis1'

salt '*' consul.health_service service='redis1' passing='True'
```

`salt.modules.consul.`**`health_state`**(*consul_url=None*, *token=None*, *state=None*, ***kwargs*)

Returns the checks in the state provided on the path.

Parameters

- **`consul_url`** – The Consul server URL.
- **`state`** – The state to show checks for. The supported states are any, unknown, passing, warning, or critical. The any state is a wildcard that can be used to return all checks.
- **`dc`** – By default, the datacenter of the agent is queried; however, the dc can be provided using the "dc" parameter.

Returns The checks in the provided state.

CLI Example:

```
salt '*' consul.health_state state='redis1'

salt '*' consul.health_state service='redis1' passing='True'
```

`salt.modules.consul.`**`list_`**(*consul_url=None*, *token=None*, *key=None*, ***kwargs*)

List keys in Consul

Parameters

- **`consul_url`** – The Consul server URL.
- **`key`** – The key to use as the starting point for the list.

Returns The list of keys.

CLI Example:

```
salt '*' consul.list
salt '*' consul.list key='web'
```

salt.modules.consul.**put**(*consul_url=None*, *token=None*, *key=None*, *value=None*, ***kwargs*)

Put values into Consul

Parameters

- **consul_url** – The Consul server URL.
- **key** – The key to use as the starting point for the list.
- **value** – The value to set the key to.
- **flags** – This can be used to specify an unsigned value between 0 and 2^64-1. Clients can choose to use this however makes sense for their application.
- **cas** – This flag is used to turn the PUT into a Check-And-Set operation.
- **acquire** – This flag is used to turn the PUT into a lock acquisition operation.
- **release** – This flag is used to turn the PUT into a lock release operation.

Returns Boolean & message of success or failure.

CLI Example:

```
salt '*' consul.put key='web/key1' value="Hello there"

salt '*' consul.put key='web/key1' value="Hello there" acquire='d5d371f4-c380-
↪5280-12fd-8810be175592'

salt '*' consul.put key='web/key1' value="Hello there" release='d5d371f4-c380-
↪5280-12fd-8810be175592'
```

salt.modules.consul.**session_create**(*consul_url=None*, *token=None*, ***kwargs*)

Used to create a session.

Parameters

- **consul_url** – The Consul server URL.
- **lockdelay** – Duration string using a "s" suffix for seconds. The default is 15s.
- **node** – Must refer to a node that is already registered, if specified. By default, the agent's own node name is used.
- **name** – A human-readable name for the session
- **checks** – A list of associated health checks. It is highly recommended that, if you override this list, you include the default "serfHealth".
- **behavior** – Can be set to either release or delete. This controls the behavior when a session is invalidated. By default, this is release, causing any locks that are held to be released. Changing this to delete causes any locks that are held to be deleted. delete is useful for creating ephemeral key/value entries.
- **ttl** – Session is invalidated if it is not renewed before the TTL expires

Returns Boolean and message indicating success or failure.

CLI Example:

```
salt '*' consul.session_create node='node1' name='my-session' behavior='delete'
↪ttl='3600s'
```

salt.modules.consul.**session_destroy**(*consul_url=None*, *token=None*, *session=None*, ***kwargs*)
Destroy session

Parameters

- **consul_url** – The Consul server URL.
- **session** – The ID of the session to destroy.
- **dc** – By default, the datacenter of the agent is queried; however, the dc can be provided using the "dc" parameter.

Returns Boolean & message of success or failure.

CLI Example:

```
salt '*' consul.session_destroy session='c1c4d223-91cb-3d1f-1ee8-f2af9e7b6716'
```

salt.modules.consul.**session_info**(*consul_url=None*, *token=None*, *session=None*, ***kwargs*)
Information about a session

Parameters

- **consul_url** – The Consul server URL.
- **session** – The ID of the session to return information about.
- **dc** – By default, the datacenter of the agent is queried; however, the dc can be provided using the "dc" parameter.

Returns Boolean & message of success or failure.

CLI Example:

```
salt '*' consul.session_info session='c1c4d223-91cb-3d1f-1ee8-f2af9e7b6716'
```

salt.modules.consul.**session_list**(*consul_url=None*, *token=None*, *return_list=False*, ***kwargs*)
Used to list sessions.

Parameters

- **consul_url** – The Consul server URL.
- **dc** – By default, the datacenter of the agent is queried; however, the dc can be provided using the "dc" parameter.
- **return_list** – By default, all information about the sessions is returned, using the return_list parameter will return a list of session IDs.

Returns A list of all available sessions.

CLI Example:

```
salt '*' consul.session_list
```

salt.modules.consul.**status_leader**(*consul_url=None*, *token=None*)
Returns the current Raft leader

Parameters **consul_url** – The Consul server URL.

Returns The address of the Raft leader.

CLI Example:

```
salt '*' consul.status_leader
```

`salt.modules.consul.`**`status_peers`**`(`*consul_url, token=None*`)`

Returns the current Raft peer set

Parameters **`consul_url`** – The Consul server URL.

Returns Retrieves the Raft peers for the datacenter in which the agent is running.

CLI Example:

```
salt '*' consul.status_peers
```

21.9.86 salt.modules.container_resource

Common resources for LXC and systemd-nspawn containers

New in version 2015.8.0.

These functions are not designed to be called directly, but instead from the *`lxc`*, *`nspawn`*, and `docker` execution modules. They provide for common logic to be re-used for common actions.

`salt.modules.container_resource.`**`cache_file`**`(`*source*`)`

Wrapper for cp.cache_file which raises an error if the file was unable to be cached.

CLI Example:

```
salt myminion container_resource.cache_file salt://foo/bar/baz.txt
```

`salt.modules.container_resource.`**`copy_to`**`(`*name, source, dest, container_type=None, path=None, exec_driver=None, overwrite=False, makedirs=False*`)`

Common logic for copying files to containers

path path to the container parent (for LXC only) default: /var/lib/lxc (system default)

CLI Example:

```
salt mymininion container_resource.copy_to mycontainer /local/file/path /container/
↪file/path container_type=docker exec_driver=nsenter
```

`salt.modules.container_resource.`**`run`**`(`*name, cmd, container_type=None, exec_driver=None, output=None, no_start=False, stdin=None, python_shell=True, output_loglevel='debug', ignore_retcode=False, path=None, use_vt=False, keep_env=None*`)`

Common logic for running shell commands in containers

path path to the container parent (for LXC only) default: /var/lib/lxc (system default)

CLI Example:

```
salt myminion container_resource.run mycontainer 'ps aux' container_type=docker
↪exec_driver=nsenter output=stdout
```

21.9.87 salt.modules.cp

Minion side functions for salt-cp

`salt.modules.cp.`**`cache_dir`**(*path*, *saltenv='base'*, *include_empty=False*, *include_pat=None*, *exclude_pat=None*)

Download and cache everything under a directory from the master

include_pat [None] Glob or regex to narrow down the files cached from the given path. If matching with a regex, the regex must be prefixed with `E@`, otherwise the expression will be interpreted as a glob.

New in version 2014.7.0.

exclude_pat [None] Glob or regex to exclude certain files from being cached from the given path. If matching with a regex, the regex must be prefixed with `E@`, otherwise the expression will be interpreted as a glob.

Note: If used with `include_pat`, files matching this pattern will be excluded from the subset of files defined by `include_pat`.

New in version 2014.7.0.

CLI Examples:

```
salt '*' cp.cache_dir salt://path/to/dir
salt '*' cp.cache_dir salt://path/to/dir include_pat='E@*.py$'
```

`salt.modules.cp.`**`cache_file`**(*path*, *saltenv='base'*, *source_hash=None*)

Used to cache a single file on the Minion

Returns the location of the new cached file on the Minion

source_hash If `name` is an http(s) or ftp URL and the file exists in the minion's file cache, this option can be passed to keep the minion from re-downloading the file if the cached copy matches the specified hash.

New in version 2018.3.0.

CLI Example:

```
salt '*' cp.cache_file salt://path/to/file
```

There are two ways of defining the fileserver environment (a.k.a. `saltenv`) from which to cache the file. One is to use the `saltenv` parameter, and the other is to use a querystring syntax in the `salt://` URL. The below two examples are equivalent:

```
salt '*' cp.cache_file salt://foo/bar.conf saltenv=config
salt '*' cp.cache_file salt://foo/bar.conf?saltenv=config
```

If the path being cached is a `salt://` URI, and the path does not exist, then `False` will be returned.

Note: It may be necessary to quote the URL when using the querystring method, depending on the shell being used to run the command.

`salt.modules.cp.`**`cache_files`**(*paths*, *saltenv='base'*)

Used to gather many files from the Master, the gathered files will be saved in the minion cachedir reflective to the paths retrieved from the Master

CLI Example:

```
salt '*' cp.cache_files salt://pathto/file1,salt://pathto/file1
```

There are two ways of defining the fileserver environment (a.k.a. `saltenv`) from which to cache the files. One is to use the `saltenv` parameter, and the other is to use a querystring syntax in the `salt://` URL. The below two examples are equivalent:

```
salt '*' cp.cache_files salt://foo/bar.conf,salt://foo/baz.conf saltenv=config
salt '*' cp.cache_files salt://foo/bar.conf?saltenv=config,salt://foo/baz.conf?
↪saltenv=config
```

The querystring method is less useful when all files are being cached from the same environment, but is a good way of caching files from multiple different environments in the same command. For example, the below command will cache the first file from the `config1` environment, and the second one from the `config2` environment.

```
salt '*' cp.cache_files salt://foo/bar.conf?saltenv=config1,salt://foo/bar.conf?
↪saltenv=config2
```

Note: It may be necessary to quote the URL when using the querystring method, depending on the shell being used to run the command.

`salt.modules.cp.`**`cache_local_file`**(*path*)

Cache a local file on the minion in the localfiles cache

CLI Example:

```
salt '*' cp.cache_local_file /etc/hosts
```

`salt.modules.cp.`**`cache_master`**(*saltenv='base'*)

Retrieve all of the files on the master and cache them locally

CLI Example:

```
salt '*' cp.cache_master
```

`salt.modules.cp.`**`envs`**()

List available environments for fileserver

CLI Example

```
salt '*' cp.envs
```

`salt.modules.cp.`**`get_dir`**(*path, dest, saltenv='base', template=None, gzip=None, **kwargs*)

Used to recursively copy a directory from the salt master

CLI Example:

```
salt '*' cp.get_dir salt://path/to/dir/ /minion/dest
```

get_dir supports the same template and gzip arguments as get_file.

`salt.modules.cp.`**`get_file`**(*path, dest, saltenv='base', makedirs=False, template=None, gzip=None, **kwargs*)

Changed in version 2018.3.0: `dest` can now be a directory

Used to get a single file from the salt master

CLI Example:

```
salt '*' cp.get_file salt://path/to/file /minion/dest
```

Template rendering can be enabled on both the source and destination file names like so:

```
salt '*' cp.get_file "salt://{{grains.os}}/vimrc" /etc/vimrc template=jinja
```

This example would instruct all Salt minions to download the vimrc from a directory with the same name as their os grain and copy it to /etc/vimrc

For larger files, the cp.get_file module also supports gzip compression. Because gzip is CPU-intensive, this should only be used in scenarios where the compression ratio is very high (e.g. pretty-printed JSON or YAML files).

Use the *gzip* named argument to enable it. Valid values are 1..9, where 1 is the lightest compression and 9 the heaviest. 1 uses the least CPU on the master (and minion), 9 uses the most.

There are two ways of defining the fileserver environment (a.k.a. `saltenv`) from which to retrieve the file. One is to use the `saltenv` parameter, and the other is to use a querystring syntax in the `salt://` URL. The below two examples are equivalent:

```
salt '*' cp.get_file salt://foo/bar.conf /etc/foo/bar.conf saltenv=config
salt '*' cp.get_file salt://foo/bar.conf?saltenv=config /etc/foo/bar.conf
```

Note: It may be necessary to quote the URL when using the querystring method, depending on the shell being used to run the command.

`salt.modules.cp.`**`get_file_str`**(*path*, *saltenv='base'*)

Download a file from a URL to the Minion cache directory and return the contents of that file

Returns `False` if Salt was unable to cache a file from a URL.

CLI Example:

```
salt '*' cp.get_file_str salt://my/file
```

`salt.modules.cp.`**`get_template`**(*path*, *dest*, *template='jinja'*, *saltenv='base'*, *makedirs=False*, ***kwargs*)

Render a file as a template before setting it down. Warning, order is not the same as in fileclient.cp for non breaking old API.

CLI Example:

```
salt '*' cp.get_template salt://path/to/template /minion/dest
```

`salt.modules.cp.`**`get_url`**(*path*, *dest=''*, *saltenv='base'*, *makedirs=False*, *source_hash=None*)

Changed in version 2018.3.0: `dest` can now be a directory

Used to get a single file from a URL.

path A URL to download a file from. Supported URL schemes are: `salt://`, `http://`, `https://`, `ftp://`, `s3://`, `swift://` and `file://` (local filesystem). If no scheme was specified, this is equivalent of using `file://`. If a `file://` URL is given, the function just returns absolute path to that file on a local filesystem. The function returns `False` if Salt was unable to fetch a file from a `salt://` URL.

dest The default behaviour is to write the fetched file to the given destination path. If this parameter is omitted or set as empty string (`''`), the function places the remote file on the local filesystem inside the Minion cache directory and returns the path to that file.

Note: To simply return the file contents instead, set destination to `None`. This works with `salt://`, `http://`, `https://` and `file://` URLs. The files fetched by `http://` and `https://` will not be cached.

---

saltenv [base] Salt fileserver envrionment from which to retrieve the file. Ignored if `path` is not a `salt://` URL.

source_hash If `path` is an http(s) or ftp URL and the file exists in the minion's file cache, this option can be passed to keep the minion from re-downloading the file if the cached copy matches the specified hash.

New in version 2018.3.0.

CLI Example:

```
salt '*' cp.get_url salt://my/file /tmp/this_file_is_mine
salt '*' cp.get_url http://www.slashdot.org /tmp/index.html
```

`salt.modules.cp.`**`hash_file`**(*path*, *saltenv='base'*)

Return the hash of a file, to get the hash of a file on the salt master file server prepend the path with salt://<file on server> otherwise, prepend the file with / for a local file.

CLI Example:

```
salt '*' cp.hash_file salt://path/to/file
```

`salt.modules.cp.`**`is_cached`**(*path*, *saltenv='base'*)

Returns the full path to a file if it is cached locally on the minion otherwise returns a blank string

CLI Example:

```
salt '*' cp.is_cached salt://path/to/file
```

`salt.modules.cp.`**`list_master`**(*saltenv='base'*, *prefix="*)

List all of the files stored on the master

CLI Example:

```
salt '*' cp.list_master
```

`salt.modules.cp.`**`list_master_dirs`**(*saltenv='base'*, *prefix="*)

List all of the directories stored on the master

CLI Example:

```
salt '*' cp.list_master_dirs
```

`salt.modules.cp.`**`list_master_symlinks`**(*saltenv='base'*, *prefix="*)

List all of the symlinks stored on the master

CLI Example:

```
salt '*' cp.list_master_symlinks
```

`salt.modules.cp.`**`list_minion`**(*saltenv='base'*)

List all of the files cached on the minion

CLI Example:

```
salt '*' cp.list_minion
```

`salt.modules.cp.`**`list_states`**(*saltenv='base'*)

List all of the available state modules in an environment

CLI Example:

```
salt '*' cp.list_states
```

`salt.modules.cp.`**`push`**(*path*, *keep_symlinks=False*, *upload_path=None*, *remove_source=False*)

WARNING Files pushed to the master will have global read permissions..

Push a file from the minion up to the master, the file will be saved to the salt master in the master's minion files cachedir (defaults to `/var/cache/salt/master/minions/minion-id/files`)

Since this feature allows a minion to push a file up to the master server it is disabled by default for security purposes. To enable, set `file_recv` to `True` in the master configuration file, and restart the master.

keep_symlinks Keep the path value without resolving its canonical form

upload_path Provide a different path inside the master's minion files cachedir

remove_source Remove the source file on the minion

New in version 2016.3.0.

CLI Example:

```
salt '*' cp.push /etc/fstab
salt '*' cp.push /etc/system-release keep_symlinks=True
salt '*' cp.push /etc/fstab upload_path='/new/path/fstab'
salt '*' cp.push /tmp/filename remove_source=True
```

`salt.modules.cp.`**`push_dir`**(*path*, *glob=None*, *upload_path=None*)

Push a directory from the minion up to the master, the files will be saved to the salt master in the master's minion files cachedir (defaults to `/var/cache/salt/master/minions/minion-id/files`). It also has a glob for matching specific files using globbing.

New in version 2014.7.0.

Since this feature allows a minion to push files up to the master server it is disabled by default for security purposes. To enable, set `file_recv` to `True` in the master configuration file, and restart the master.

upload_path Provide a different path and directory name inside the master's minion files cachedir

CLI Example:

```
salt '*' cp.push /usr/lib/mysql
salt '*' cp.push /usr/lib/mysql upload_path='/newmysql/path'
salt '*' cp.push_dir /etc/modprobe.d/ glob='*.conf'
```

`salt.modules.cp.`**`recv`**(*files*, *dest*)

Used with salt-cp, pass the files dict, and the destination.

This function receives small fast copy files from the master via salt-cp. It does not work via the CLI.

`salt.modules.cp.`**`recv_chunked`**(*dest*, *chunk*, *append=False*, *compressed=True*, *mode=None*)

This function receives files copied to the minion using `salt-cp` and is not intended to be used directly on the CLI.

`salt.modules.cp.`**`stat_file`**(*path*, *saltenv='base'*, *octal=True*)

Return the permissions of a file, to get the permissions of a file on the salt master file server prepend the path with salt://<file on server> otherwise, prepend the file with / for a local file.

CLI Example:

```
salt '*' cp.stat_file salt://path/to/file
```

21.9.88 salt.modules.cpan

Manage Perl modules using CPAN

New in version 2015.5.0.

`salt.modules.cpan.`**`install`**(*module*)

Install a Perl module from CPAN

CLI Example:

```
salt '*' cpan.install Template::Alloy
```

`salt.modules.cpan.`**`list_`**()

List installed Perl modules, and the version installed

CLI Example:

```
salt '*' cpan.list
```

`salt.modules.cpan.`**`remove`**(*module*, *details=False*)

Attempt to remove a Perl module that was installed from CPAN. Because the `cpan` command doesn't actually support "uninstall"-like functionality, this function will attempt to do what it can, with what it has from CPAN.

Until this function is declared stable, USE AT YOUR OWN RISK!

CLI Example:

```
salt '*' cpan.remove Old::Package
```

`salt.modules.cpan.`**`show`**(*module*)

Show information about a specific Perl module

CLI Example:

```
salt '*' cpan.show Template::Alloy
```

`salt.modules.cpan.`**`show_config`**()

Return a dict of CPAN configuration values

CLI Example:

```
salt '*' cpan.show_config
```

21.9.89 salt.modules.cron

Work with cron

Note: Salt does not escape cron metacharacters automatically. You should backslash-escape percent characters and any other metacharacters that might be interpreted incorrectly by the shell.

`salt.modules.cron.`**`list_tab`**(*user*)

Return the contents of the specified user's crontab

CLI Example:

```
salt '*' cron.list_tab root
```

`salt.modules.cron.`**`ls`**(*user*)

This function is an alias of `list_tab`.

> Return the contents of the specified user's crontab
>
> CLI Example:
>
> ```
> salt '*' cron.list_tab root
> ```

`salt.modules.cron.`**`raw_cron`**(*user*)

Return the contents of the user's crontab

CLI Example:

```
salt '*' cron.raw_cron root
```

`salt.modules.cron.`**`rm`**(*user, cmd, minute=None, hour=None, daymonth=None, month=None, dayweek=None, identifier=None*)

This function is an alias of `rm_job`.

> Remove a cron job for a specified user. If any of the day/time params are specified, the job will only be removed if the specified params match.
>
> CLI Example:
>
> ```
> salt '*' cron.rm_job root /usr/local/weekly
> salt '*' cron.rm_job root /usr/bin/foo dayweek=1
> ```

`salt.modules.cron.`**`rm_env`**(*user, name*)

Remove cron environment variable for a specified user.

CLI Example:

```
salt '*' cron.rm_env root MAILTO
```

`salt.modules.cron.`**`rm_job`**(*user, cmd, minute=None, hour=None, daymonth=None, month=None, dayweek=None, identifier=None*)

Remove a cron job for a specified user. If any of the day/time params are specified, the job will only be removed if the specified params match.

CLI Example:

```
salt '*' cron.rm_job root /usr/local/weekly
salt '*' cron.rm_job root /usr/bin/foo dayweek=1
```

`salt.modules.cron.`**`rm_special`**(*user, cmd, special=None, identifier=None*)

Remove a special cron job for a specified user.

CLI Example:

```
salt '*' cron.rm_special root /usr/bin/foo
```

`salt.modules.cron.`**`set_env`**(*user*, *name*, *value=None*)

Set up an environment variable in the crontab.

CLI Example:

```
salt '*' cron.set_env root MAILTO user@example.com
```

`salt.modules.cron.`**`set_job`**(*user*, *minute*, *hour*, *daymonth*, *month*, *dayweek*, *cmd*, *commented=False*, *comment=None*, *identifier=None*)

Sets a cron job up for a specified user.

CLI Example:

```
salt '*' cron.set_job root '*' '*' '*' '*' 1 /usr/local/weekly
```

`salt.modules.cron.`**`set_special`**(*user*, *special*, *cmd*, *commented=False*, *comment=None*, *identifier=None*)

Set up a special command in the crontab.

CLI Example:

```
salt '*' cron.set_special root @hourly 'echo foobar'
```

`salt.modules.cron.`**`write_cron_file`**(*user*, *path*)

Writes the contents of a file to a user's crontab

CLI Example:

```
salt '*' cron.write_cron_file root /tmp/new_cron
```

Changed in version 2015.8.9.

Note: Some OS' do not support specifying user via the *crontab* command i.e. (Solaris, AIX)

`salt.modules.cron.`**`write_cron_file_verbose`**(*user*, *path*)

Writes the contents of a file to a user's crontab and return error message on error

CLI Example:

```
salt '*' cron.write_cron_file_verbose root /tmp/new_cron
```

Changed in version 2015.8.9.

Note: Some OS' do not support specifying user via the *crontab* command i.e. (Solaris, AIX)

21.9.90 salt.modules.csf

Support for Config Server Firewall (CSF)

maintainer Mostafa Hussein <mostafa.hussein91@gmail.com>

maturity new

platform Linux

`salt.modules.csf.`**`allow`**`(`*ip, port=None, proto='tcp', direction='in', port_origin='d', ip_origin='s', ttl=None, comment=''*`)`

Add an rule to csf allowed hosts See `_access_rule()`. 1- Add an IP: CLI Example:

```
salt '*' csf.allow 127.0.0.1
salt '*' csf.allow 127.0.0.1 comment="Allow localhost"
```

`salt.modules.csf.`**`allow_port`**`(`*port, proto='tcp', direction='both'*`)`

Like allow_ports, but it will append to the existing entry instead of replacing it. Takes a single port instead of a list of ports.

CLI Example:

```
salt '*' csf.allow_port 22 proto='tcp' direction='in'
```

`salt.modules.csf.`**`allow_ports`**`(`*ports, proto='tcp', direction='in'*`)`

Fully replace the incoming or outgoing ports line in the csf.conf file - e.g. TCP_IN, TCP_OUT, UDP_IN, UDP_OUT, etc.

CLI Example:

```
salt '*' csf.allow_ports ports="[22,80,443,4505,4506]" proto='tcp' direction='in'
```

`salt.modules.csf.`**`build_directions`**`(`*direction*`)`

`salt.modules.csf.`**`deny`**`(`*ip, port=None, proto='tcp', direction='in', port_origin='d', ip_origin='d', ttl=None, comment=''*`)`

Add an rule to csf denied hosts See `_access_rule()`. 1- Deny an IP: CLI Example:

```
salt '*' csf.deny 127.0.0.1
salt '*' csf.deny 127.0.0.1 comment="Too localhosty"
```

`salt.modules.csf.`**`disable`**`()`

Disable csf permanently CLI Example:

```
salt '*' csf.disable
```

`salt.modules.csf.`**`disable_testing_mode`**`()`

`salt.modules.csf.`**`enable`**`()`

Activate csf if not running CLI Example:

```
salt '*' csf.enable
```

`salt.modules.csf.`**`enable_testing_mode`**`()`

`salt.modules.csf.`**`exists`**`(`*method, ip, port=None, proto='tcp', direction='in', port_origin='d', ip_origin='d', ttl=None, comment=''*`)`

Returns true a rule for the ip already exists based on the method supplied. Returns false if not found. CLI Example:

```
salt '*' csf.exists allow 1.2.3.4
salt '*' csf.exists tempdeny 1.2.3.4
```

`salt.modules.csf.`**`get_option`**`(`*option*`)`

`salt.modules.csf.`**`get_ports`**`(`*proto='tcp', direction='in'*`)`

Lists ports from csf.conf based on direction and protocol. e.g. - TCP_IN, TCP_OUT, UDP_IN, UDP_OUT, etc..

CLI Example:

```
salt '*' csf.allow_port 22 proto='tcp' direction='in'
```

salt.modules.csf.**get_skipped_nics**(*ipv6=False*)

salt.modules.csf.**get_testing_status**()

salt.modules.csf.**reload**()
Restart csf CLI Example:

```
salt '*' csf.reload
```

salt.modules.csf.**remove_rule**(*method, ip, port=None, proto='tcp', direction='in', port_origin='d', ip_origin='s', ttl=None, comment="*)

salt.modules.csf.**remove_temp_rule**(*ip*)

salt.modules.csf.**running**()
Check csf status CLI Example:

```
salt '*' csf.running
```

salt.modules.csf.**set_option**(*option, value*)

salt.modules.csf.**skip_nic**(*nic, ipv6=False*)

salt.modules.csf.**skip_nics**(*nics, ipv6=False*)

salt.modules.csf.**split_option**(*option*)

salt.modules.csf.**tempallow**(*ip=None, ttl=None, port=None, direction=None, comment="*)
Add an rule to the temporary ip allow list. See `_access_rule()`. 1- Add an IP: CLI Example:

```
salt '*' csf.tempallow 127.0.0.1 3600 port=22 direction='in' comment='# Temp dev
↪ssh access'
```

salt.modules.csf.**tempdeny**(*ip=None, ttl=None, port=None, direction=None, comment="*)
Add a rule to the temporary ip deny list. See `_access_rule()`. 1- Add an IP: CLI Example:

```
salt '*' csf.tempdeny 127.0.0.1 300 port=22 direction='in' comment='# Brute force
↪attempt'
```

salt.modules.csf.**unallow**(*ip*)
Remove a rule from the csf denied hosts See `_access_rule()`. 1- Deny an IP: CLI Example:

```
salt '*' csf.unallow 127.0.0.1
```

salt.modules.csf.**undeny**(*ip*)
Remove a rule from the csf denied hosts See `_access_rule()`. 1- Deny an IP: CLI Example:

```
salt '*' csf.undeny 127.0.0.1
```

21.9.91 salt.modules.cyg

Manage cygwin packages.

Module file to accompany the cyg state.

`salt.modules.cyg.`**`check_valid_package`**(*package*, *cyg_arch='x86_64'*, *mirrors=None*)

Check if the package is valid on the given mirrors.

Parameters

- **`package`** – The name of the package
- **`cyg_arch`** – The cygwin architecture
- **`mirrors`** – any mirrors to check

Returns (bool): True if Valid, otherwise False

CLI Example:

```
salt '*' cyg.check_valid_package <package name>
```

`salt.modules.cyg.`**`install`**(*packages=None*, *cyg_arch='x86_64'*, *mirrors=None*)

Install one or several packages.

packages [None] The packages to install

cyg_arch [x86_64] Specify the architecture to install the package under Current options are x86 and x86_64

CLI Example:

```
salt '*' cyg.install dos2unix
salt '*' cyg.install dos2unix mirrors="[{'http://mirror': 'http://url/to/public/
↪key}]'
```

`salt.modules.cyg.`**`list_`**(*package=''*, *cyg_arch='x86_64'*)

List locally installed packages.

package [''] package name to check. else all

cyg_arch : Cygwin architecture to use Options are x86 and x86_64

CLI Example:

```
salt '*' cyg.list
```

`salt.modules.cyg.`**`uninstall`**(*packages*, *cyg_arch='x86_64'*, *mirrors=None*)

Uninstall one or several packages.

packages The packages to uninstall.

cyg_arch [x86_64] Specify the architecture to remove the package from Current options are x86 and x86_64

CLI Example:

```
salt '*' cyg.uninstall dos2unix
salt '*' cyg.uninstall dos2unix mirrors="[{'http://mirror': 'http://url/to/public/
↪key}]"
```

`salt.modules.cyg.`**`update`**(*cyg_arch='x86_64'*, *mirrors=None*)

Update all packages.

cyg_arch [x86_64] Specify the cygwin architecture update Current options are x86 and x86_64

CLI Example:

```
salt '*' cyg.update
salt '*' cyg.update dos2unix mirrors="[{'http://mirror': 'http://url/to/public/
↪key}]"
```

21.9.92 salt.modules.daemontools

daemontools service module. This module will create daemontools type service watcher.

This module is compatible with the `service` states, so it can be used to maintain services using the `provider` argument:

```
myservice:
  service.running:
    - provider: daemontools
```

`salt.modules.daemontools.`**`available`**`(`*name*`)`

Returns `True` if the specified service is available, otherwise returns `False`.

CLI Example:

```
salt '*' daemontools.available foo
```

`salt.modules.daemontools.`**`disabled`**`(`*name*`)`

Return True if the named service is enabled, false otherwise

New in version 2015.5.6.

CLI Example:

```
salt '*' daemontools.disabled <service name>
```

`salt.modules.daemontools.`**`enabled`**`(`*name*, ***kwargs*`)`

Return True if the named service is enabled, false otherwise A service is considered enabled if in your service directory: - an executable ./run file exist - a file named "down" does not exist

New in version 2015.5.7.

name Service name

CLI Example:

```
salt '*' daemontools.enabled <service name>
```

`salt.modules.daemontools.`**`full_restart`**`(`*name*`)`

Calls daemontools.restart() function

CLI Example:

```
salt '*' daemontools.full_restart <service name>
```

`salt.modules.daemontools.`**`get_all`**`()`

Return a list of all available services

CLI Example:

```
salt '*' daemontools.get_all
```

`salt.modules.daemontools.`**`missing`**`(`*name*`)`

The inverse of daemontools.available. Returns `True` if the specified service is not available, otherwise returns `False`.

CLI Example:

```
salt '*' daemontools.missing foo
```

`salt.modules.daemontools.`**`reload_`**`(`*name*`)`

Wrapper for term()

CLI Example:

```
salt '*' daemontools.reload <service name>
```

`salt.modules.daemontools.`**`restart`**`(`*name*`)`

Restart service via daemontools. This will stop/start service

CLI Example:

```
salt '*' daemontools.restart <service name>
```

`salt.modules.daemontools.`**`start`**`(`*name*`)`

Starts service via daemontools

CLI Example:

```
salt '*' daemontools.start <service name>
```

`salt.modules.daemontools.`**`status`**`(`*name, sig=None*`)`

Return the status for a service via daemontools, return pid if running

CLI Example:

```
salt '*' daemontools.status <service name>
```

`salt.modules.daemontools.`**`stop`**`(`*name*`)`

Stops service via daemontools

CLI Example:

```
salt '*' daemontools.stop <service name>
```

`salt.modules.daemontools.`**`term`**`(`*name*`)`

Send a TERM to service via daemontools

CLI Example:

```
salt '*' daemontools.term <service name>
```

21.9.93 salt.modules.data

Manage a local persistent data structure that can hold any arbitrary data specific to the minion

`salt.modules.data.`**`cas`**`(`*key, value, old_value*`)`

Check and set a value in the minion datastore

CLI Example:

```
salt '*' data.cas <key> <value> <old_value>
```

`salt.modules.data.`**`clear`**`()`

Clear out all of the data in the minion datastore, this function is destructive!

CLI Example:

```
salt '*' data.clear
```

`salt.modules.data.`**`dump`**`(`*new_data*`)`

Replace the entire datastore with a passed data structure

CLI Example:

```
salt '*' data.dump '{'eggs': 'spam'}'
```

`salt.modules.data.`**`get`**`(`*key, default=None*`)`

Get a (list of) value(s) from the minion datastore

New in version 2015.8.0.

CLI Example:

```
salt '*' data.get key
salt '*' data.get '["key1", "key2"]'
```

`salt.modules.data.`**`has_key`**`(`*key*`)`

Check if key is in the minion datastore

New in version 2015.8.0.

CLI Example:

```
salt '*' data.has_key <mykey>
```

`salt.modules.data.`**`items`**`()`

Get items from the minion datastore

New in version 2015.8.0.

CLI Example:

```
salt '*' data.items
```

`salt.modules.data.`**`keys`**`()`

Get all keys from the minion datastore

New in version 2015.8.0.

CLI Example:

```
salt '*' data.keys
```

`salt.modules.data.`**`load`**`()`

Return all of the data in the minion datastore

CLI Example:

```
salt '*' data.load
```

`salt.modules.data.`**`pop`**`(`*key, default=None*`)`

Pop (return & delete) a value from the minion datastore

New in version 2015.5.2.

CLI Example:

```
salt '*' data.pop <key> "there was no val"
```

`salt.modules.data.`**`update`**(*key*, *value*)

Update a key with a value in the minion datastore

CLI Example:

```
salt '*' data.update <key> <value>
```

`salt.modules.data.`**`values`**()

Get values from the minion datastore

New in version 2015.8.0.

CLI Example:

```
salt '*' data.values
```

21.9.94 salt.modules.datadog_api

An execution module that interacts with the Datadog API

The following parameters are required for all functions.

api_key The datadog API key

app_key The datadog application key

Full argument reference is available on the Datadog API reference page https://docs.datadoghq.com/api/

`salt.modules.datadog_api.`**`cancel_downtime`**(*api_key=None*, *app_key=None*, *scope=None*, *id=None*)

Cancel a downtime by id or by scope.

CLI Example:

```
salt-call datadog.cancel_downtime scope='host:app01' \
                                  api_key='0123456789' \
                                  app_key='9876543210'`
```

Arguments - Either scope or id is required.

Parameters

- **`id`** – The downtime ID
- **`scope`** – The downtime scope

`salt.modules.datadog_api.`**`post_event`**(*api_key=None*, *app_key=None*, *title=None*, *text=None*, *date_happened=None*, *priority=None*, *host=None*, *tags=None*, *alert_type=None*, *aggregation_key=None*, *source_type_name=None*)

Post an event to the Datadog stream.

CLI Example

```
salt-call datadog.post_event api_key='0123456789' \
                             app_key='9876543210' \
                             title='Salt Highstate' \
```

(continues on next page)

(continued from previous page)

```
                              text="Salt highstate was run on $(salt-call grains.
↪get id)" \
                              tags='["service:salt", "event:highstate"]'
```

Required arguments

Parameters

- **title** – The event title. Limited to 100 characters.
- **text** – The body of the event. Limited to 4000 characters. The text supports markdown.

Optional arguments

Parameters

- **date_happened** – POSIX timestamp of the event.
- **priority** – The priority of the event ('normal' or 'low').
- **host** – Host name to associate with the event.
- **tags** – A list of tags to apply to the event.
- **alert_type** – "error", "warning", "info" or "success".
- **aggregation_key** – An arbitrary string to use for aggregation, max length of 100 characters.
- **source_type_name** – The type of event being posted.

salt.modules.datadog_api.**schedule_downtime**(*scope, api_key=None, app_key=None, monitor_id=None, start=None, end=None, message=None, recurrence=None, timezone=None, test=False*)

Schedule downtime for a scope of monitors.

CLI Example:

```
salt-call datadog.schedule_downtime 'host:app2' \
                                    stop=$(date --date='30 minutes' +%s) \
                                    app_key='0123456789' \
                                    api_key='9876543210'
```

Optional arguments

Parameters

- **monitor_id** – The ID of the monitor
- **start** – Start time in seconds since the epoch
- **end** – End time in seconds since the epoch
- **message** – A message to send in a notification for this downtime
- **recurrence** – Repeat this downtime periodically
- **timezone** – Specify the timezone

21.9.95 salt.modules.ddns

Support for RFC 2136 dynamic DNS updates.

depends

- dnspython Python module

configuration If you want to use TSIG authentication for the server, there are a couple of optional configuration parameters made available to support this (the keyname is only needed if the keyring contains more than one key):

```
keyfile: keyring file (default=None)
keyname: key name in file (default=None)
keyalgorithm: algorithm used to create the key
              (default='HMAC-MD5.SIG-ALG.REG.INT').
    Other possible values: hmac-sha1, hmac-sha224, hmac-sha256,
        hmac-sha384, hmac-sha512
```

The keyring file needs to be in json format and the key name needs to end with an extra period in the file, similar to this:

```
{"keyname.": "keycontent"}
```

`salt.modules.ddns.`**`add_host`**(*zone, name, ttl, ip, nameserver='127.0.0.1', replace=True, timeout=5, port=53, **kwargs*)

Add, replace, or update the A and PTR (reverse) records for a host.

CLI Example:

```
salt ns1 ddns.add_host example.com host1 60 10.1.1.1
```

`salt.modules.ddns.`**`delete`**(*zone, name, rdtype=None, data=None, nameserver='127.0.0.1', timeout=5, port=53, **kwargs*)

Delete a DNS record.

CLI Example:

```
salt ns1 ddns.delete example.com host1 A
```

`salt.modules.ddns.`**`delete_host`**(*zone, name, nameserver='127.0.0.1', timeout=5, port=53, **kwargs*)

Delete the forward and reverse records for a host.

Returns true if any records are deleted.

CLI Example:

```
salt ns1 ddns.delete_host example.com host1
```

`salt.modules.ddns.`**`update`**(*zone, name, ttl, rdtype, data, nameserver='127.0.0.1', timeout=5, replace=False, port=53, **kwargs*)

Add, replace, or update a DNS record. nameserver must be an IP address and the minion running this module must have update privileges on that server. If replace is true, first deletes all records for this name and type.

CLI Example:

```
salt ns1 ddns.update example.com host1 60 A 10.0.0.1
```

21.9.96 salt.modules.deb_apache

Support for Apache

Please note: The functions in here are Debian-specific. Placing them in this separate file will allow them to load only on Debian-based systems, while still loading under the `apache` namespace.

`salt.modules.deb_apache.`**`a2disconf`**`(`*conf*`)`

New in version 2016.3.0.

Runs a2disconf for the given conf.

This will only be functional on Debian-based operating systems (Ubuntu, Mint, etc).

CLI Examples:

```
salt '*' apache.a2disconf security
```

`salt.modules.deb_apache.`**`a2dismod`**`(`*mod*`)`

Runs a2dismod for the given mod.

This will only be functional on Debian-based operating systems (Ubuntu, Mint, etc).

CLI Examples:

```
salt '*' apache.a2dismod vhost_alias
```

`salt.modules.deb_apache.`**`a2dissite`**`(`*site*`)`

Runs a2dissite for the given site.

This will only be functional on Debian-based operating systems (Ubuntu, Mint, etc).

CLI Examples:

```
salt '*' apache.a2dissite example.com
```

`salt.modules.deb_apache.`**`a2enconf`**`(`*conf*`)`

New in version 2016.3.0.

Runs a2enconf for the given conf.

This will only be functional on Debian-based operating systems (Ubuntu, Mint, etc).

CLI Examples:

```
salt '*' apache.a2enconf security
```

`salt.modules.deb_apache.`**`a2enmod`**`(`*mod*`)`

Runs a2enmod for the given mod.

This will only be functional on Debian-based operating systems (Ubuntu, Mint, etc).

CLI Examples:

```
salt '*' apache.a2enmod vhost_alias
```

`salt.modules.deb_apache.`**`a2ensite`**`(`*site*`)`

Runs a2ensite for the given site.

This will only be functional on Debian-based operating systems (Ubuntu, Mint, etc).

CLI Examples:

```
salt '*' apache.a2ensite example.com
```

`salt.modules.deb_apache.`**`check_conf_enabled`**`(`*conf*`)`

New in version 2016.3.0.

Checks to see if the specific conf symlink is in /etc/apache2/conf-enabled.

This will only be functional on Debian-based operating systems (Ubuntu, Mint, etc).

CLI Examples:

```
salt '*' apache.check_conf_enabled security
salt '*' apache.check_conf_enabled security.conf
```

`salt.modules.deb_apache.`**`check_mod_enabled`**`(`*mod*`)`

Checks to see if the specific mod symlink is in /etc/apache2/mods-enabled.

This will only be functional on Debian-based operating systems (Ubuntu, Mint, etc).

CLI Examples:

```
salt '*' apache.check_mod_enabled status
salt '*' apache.check_mod_enabled status.load
salt '*' apache.check_mod_enabled status.conf
```

`salt.modules.deb_apache.`**`check_site_enabled`**`(`*site*`)`

Checks to see if the specific site symlink is in /etc/apache2/sites-enabled.

This will only be functional on Debian-based operating systems (Ubuntu, Mint, etc).

CLI Examples:

```
salt '*' apache.check_site_enabled example.com
salt '*' apache.check_site_enabled example.com.conf
```

21.9.97 salt.modules.deb_postgres

Module to provide Postgres compatibility to salt for debian family specific tools.

`salt.modules.deb_postgres.`**`cluster_create`**`(`*version, name='main', port=None, locale=None, encoding=None, datadir=None*`)`

Adds a cluster to the Postgres server.

CLI Example:

```
salt '*' postgres.cluster_create '9.3'

salt '*' postgres.cluster_create '9.3' 'main'

salt '*' postgres.cluster_create '9.3' locale='fr_FR'
```

`salt.modules.deb_postgres.`**`cluster_exists`**`(`*version, name='main'*`)`

Checks if a given version and name of a cluster exists.

CLI Example:

```
salt '*' postgres.cluster_exists '9.3'

salt '*' postgres.cluster_exists '9.3' 'main'
```

`salt.modules.deb_postgres.`**`cluster_list`**`(`*verbose=False*`)`

Return a list of cluster of Postgres server (tuples of version and name).

CLI Example:

```
salt '*' postgres.cluster_list

salt '*' postgres.cluster_list verbose=True
```

`salt.modules.deb_postgres.`**`cluster_remove`**(*version*, *name='main'*, *stop=False*)

Remove a cluster on a Postgres server. By default it doesn't try to stop the cluster.

CLI Example:

```
salt '*' postgres.cluster_remove '9.3'

salt '*' postgres.cluster_remove '9.3' 'main'

salt '*' postgres.cluster_remove '9.3' 'main' stop=True
```

21.9.98 salt.modules.debuild_pkgbuild

Debian Package builder system

New in version 2015.8.0.

This system allows for all of the components to build debs safely in chrooted environments. This also provides a function to generate debian repositories

This module implements the pkgbuild interface

`salt.modules.debuild_pkgbuild.`**`build`**(*runas*, *tgt*, *dest_dir*, *spec*, *sources*, *deps*, *env*, *template*, *saltenv='base'*, *log_dir='/var/log/salt/pkgbuild'*)

Given the package destination directory, the tarball containing debian files (e.g. control) and package sources, use pbuilder to safely build the platform package

CLI Example:

Debian

```
salt '*' pkgbuild.make_src_pkg deb-8-x86_64 /var/www/html
        https://raw.githubusercontent.com/saltstack/libnacl/master/pkg/deb/python-
↪libnacl.control
        https://pypi.python.org/packages/source/l/libnacl/libnacl-1.3.5.tar.gz
```

This example command should build the libnacl package for Debian using pbuilder and place it in /var/www/html/ on the minion

`salt.modules.debuild_pkgbuild.`**`make_repo`**(*repodir*, *keyid=None*, *env=None*, *use_passphrase=False*, *gnupghome='/etc/salt/gpgkeys'*, *runas='root'*, *timeout=15.0*)

Make a package repository and optionally sign it and packages present

Given the repodir (directory to create repository in), create a Debian repository and optionally sign it and packages present. This state is best used with onchanges linked to your package building states.

repodir The directory to find packages that will be in the repository.

keyid Changed in version 2016.3.0.

Optional Key ID to use in signing packages and repository. This consists of the last 8 hex digits of the GPG key ID.

Utilizes Public and Private keys associated with keyid which have been loaded into the minion's Pillar data. Leverages gpg-agent and gpg-preset-passphrase for caching keys, etc. These pillar values are assumed to be filenames which are present in `gnupghome`. The pillar keys shown below have to match exactly.

For example, contents from a Pillar data file with named Public and Private keys as follows:

```
gpg_pkg_priv_keyname: gpg_pkg_key.pem
gpg_pkg_pub_keyname: gpg_pkg_key.pub
```

env Changed in version 2016.3.0.

A dictionary of environment variables to be utilized in creating the repository.

use_passphrase [False] New in version 2016.3.0.

Use a passphrase with the signing key presented in `keyid`. Passphrase is received from Pillar data which could be passed on the command line with `pillar` parameter. For example:

```
pillar='{ "gpg_passphrase" : "my_passphrase" }'
```

gnupghome [/etc/salt/gpgkeys] New in version 2016.3.0.

Location where GPG related files are stored, used with `keyid`.

runas [root] New in version 2016.3.0.

User to create the repository as, and optionally sign packages.

Note: Ensure the user has correct permissions to any files and directories which are to be utilized.

timeout [15.0] New in version 2016.3.4.

Timeout in seconds to wait for the prompt for inputting the passphrase.

CLI Example:

```
salt '*' pkgbuild.make_repo /var/www/html
```

`salt.modules.debuild_pkgbuild.`**`make_src_pkg`**(*dest_dir, spec, sources, env=None, saltenv='base', runas='root'*)

Create a platform specific source package from the given platform spec/control file and sources

CLI Example:

Debian

```
salt '*' pkgbuild.make_src_pkg /var/www/html/
        https://raw.githubusercontent.com/saltstack/libnacl/master/pkg/deb/python-
↪libnacl.control.tar.xz
        https://pypi.python.org/packages/source/l/libnacl/libnacl-1.3.5.tar.gz
```

This example command should build the libnacl SOURCE package and place it in /var/www/html/ on the minion

dest_dir Absolute path for directory to write source package

spec Absolute path to spec file or equivalent

sources Absolute path to source files to build source package from

env [None] A list or dictionary of environment variables to be set prior to execution. Example:

```
- env:
    - DEB_BUILD_OPTIONS: 'nocheck'
```

Warning: The above illustrates a common PyYAML pitfall, that **yes**, **no**, **on**, **off**, **true**, and **false** are all loaded as boolean `True` and `False` values, and must be enclosed in quotes to be used as strings. More info on this (and other) PyYAML idiosyncrasies can be found *here*.

saltenv: base

Salt environment variables

runas [root] New in version 2019.2.1.

User to create the files and directories

Note: Ensure the user has correct permissions to any files and directories which are to be utilized.

21.9.99 salt.modules.debconfmod

Support for Debconf

`salt.modules.debconfmod.`**`get_selections`**(*fetchempty=True*)

Answers to debconf questions for all packages in the following format:

```
{'package': [['question', 'type', 'value'], ...]}
```

CLI Example:

```
salt '*' debconf.get_selections
```

`salt.modules.debconfmod.`**`set_`**(*package*, *question*, *type*, *value*, **extra*)

Set answers to debconf questions for a package.

CLI Example:

```
salt '*' debconf.set <package> <question> <type> <value> [<value> ...]
```

`salt.modules.debconfmod.`**`set_file`**(*path*, *saltenv='base'*, ***kwargs*)

Set answers to debconf questions from a file.

CLI Example:

```
salt '*' debconf.set_file salt://pathto/pkg.selections
```

`salt.modules.debconfmod.`**`set_template`**(*path*, *template*, *context*, *defaults*, *saltenv='base'*, ***kwargs*)

Set answers to debconf questions from a template.

path location of the file containing the package selections

template template format

context variables to add to the template environment

default default values for the template environment

CLI Example:

```
salt '*' debconf.set_template salt://pathto/pkg.selections.jinja jinja None None
```

`salt.modules.debconfmod.`**`show`**`(`*name*`)`

Answers to debconf questions for a package in the following format:

```
[['question', 'type', 'value'], ...]
```

If debconf doesn't know about a package, we return None.

CLI Example:

```
salt '*' debconf.show <package name>
```

21.9.100 salt.modules.debian_ip

The networking module for Debian-based distros

References:

- http://www.debian.org/doc/manuals/debian-reference/ch05.en.html

`salt.modules.debian_ip.`**`apply_network_settings`**`(`***settings*`)`

Apply global network configuration.

CLI Example:

```
salt '*' ip.apply_network_settings
```

`salt.modules.debian_ip.`**`build_bond`**`(`*iface, **settings*`)`

Create a bond script in /etc/modprobe.d with the passed settings and load the bonding kernel module.

CLI Example:

```
salt '*' ip.build_bond bond0 mode=balance-alb
```

`salt.modules.debian_ip.`**`build_interface`**`(`*iface, iface_type, enabled, **settings*`)`

Build an interface script for a network interface.

CLI Example:

```
salt '*' ip.build_interface eth0 eth <settings>
```

`salt.modules.debian_ip.`**`build_network_settings`**`(`***settings*`)`

Build the global network script.

CLI Example:

```
salt '*' ip.build_network_settings <settings>
```

`salt.modules.debian_ip.`**`build_routes`**`(`*iface, **settings*`)`

Add route scripts for a network interface using up commands.

CLI Example:

```
salt '*' ip.build_routes eth0 <settings>
```

`salt.modules.debian_ip.`**`down`**`(`*iface*, *iface_type*`)`
Shutdown a network interface

CLI Example:

```
salt '*' ip.down eth0 eth
```

`salt.modules.debian_ip.`**`get_bond`**`(`*iface*`)`
Return the content of a bond script

CLI Example:

```
salt '*' ip.get_bond bond0
```

`salt.modules.debian_ip.`**`get_interface`**`(`*iface*`)`
Return the contents of an interface script

CLI Example:

```
salt '*' ip.get_interface eth0
```

`salt.modules.debian_ip.`**`get_network_settings`**`()`
Return the contents of the global network script.

CLI Example:

```
salt '*' ip.get_network_settings
```

`salt.modules.debian_ip.`**`get_routes`**`(`*iface*`)`
Return the routes for the interface

CLI Example:

```
salt '*' ip.get_routes eth0
```

`salt.modules.debian_ip.`**`up`**`(`*iface*, *iface_type*`)`
Start up a network interface

CLI Example:

```
salt '*' ip.up eth0 eth
```

21.9.101 salt.modules.debian_service

Service support for Debian systems (uses update-rc.d and /sbin/service)

Important: If you feel that Salt should be using this module to manage services on a minion, and it is using a different module (or gives an error similar to *'service.start' is not available*), see here.

`salt.modules.debian_service.`**`available`**`(`*name*`)`
Returns `True` if the specified service is available, otherwise returns `False`.

CLI Example:

```
salt '*' service.available sshd
```

`salt.modules.debian_service.`**`disable`**`(`*name*, ***kwargs*`)`

Disable the named service to start at boot

CLI Example:

```
salt '*' service.disable <service name>
```

`salt.modules.debian_service.`**`disabled`**`(`*name*`)`

Return True if the named service is disabled, false otherwise

CLI Example:

```
salt '*' service.disabled <service name>
```

`salt.modules.debian_service.`**`enable`**`(`*name*, ***kwargs*`)`

Enable the named service to start at boot

CLI Example:

```
salt '*' service.enable <service name>
```

`salt.modules.debian_service.`**`enabled`**`(`*name*, ***kwargs*`)`

Return True if the named service is enabled, false otherwise

CLI Example:

```
salt '*' service.enabled <service name>
```

`salt.modules.debian_service.`**`force_reload`**`(`*name*`)`

Force-reload the named service

CLI Example:

```
salt '*' service.force_reload <service name>
```

`salt.modules.debian_service.`**`get_all`**`()`

Return all available boot services

CLI Example:

```
salt '*' service.get_all
```

`salt.modules.debian_service.`**`get_disabled`**`()`

Return a set of services that are installed but disabled

CLI Example:

```
salt '*' service.get_disabled
```

`salt.modules.debian_service.`**`get_enabled`**`()`

Return a list of service that are enabled on boot

CLI Example:

```
salt '*' service.get_enabled
```

`salt.modules.debian_service.`**`missing`**`(`*name*`)`

The inverse of service.available. Returns `True` if the specified service is not available, otherwise returns `False`.

CLI Example:

```
salt '*' service.missing sshd
```

`salt.modules.debian_service.`**`reload_`**`(`*name*`)`

Reload the named service

CLI Example:

```
salt '*' service.reload <service name>
```

`salt.modules.debian_service.`**`restart`**`(`*name*`)`

Restart the named service

CLI Example:

```
salt '*' service.restart <service name>
```

`salt.modules.debian_service.`**`start`**`(`*name*`)`

Start the specified service

CLI Example:

```
salt '*' service.start <service name>
```

`salt.modules.debian_service.`**`status`**`(`*name, sig=None*`)`

Return the status for a service. If the name contains globbing, a dict mapping service name to True/False values is returned.

Changed in version 2018.3.0: The service name can now be a glob (e.g. `salt*`)

Parameters

- **`name`** (*`str`*) – The name of the service to check
- **`sig`** (*`str`*) – Signature to use to find the service via ps

Returns True if running, False otherwise dict: Maps service name to True if running, False otherwise

Return type bool

CLI Example:

```
salt '*' service.status <service name> [service signature]
```

`salt.modules.debian_service.`**`stop`**`(`*name*`)`

Stop the specified service

CLI Example:

```
salt '*' service.stop <service name>
```

21.9.102 salt.modules.defaults

Module to work with salt formula defaults files

`salt.modules.defaults.`**`deepcopy`**`(`*source*`)`

Allows deep copy of objects in formulas.

> By default, Python does not copy objects, it creates bindings between a target and an object.

It is more typical to use this in a templating language in formulas, instead of directly on the command-line.

`salt.modules.defaults.`**`get`**(*key*, *default=''*)

defaults.get is used much like pillar.get except that it will read a default value for a pillar from defaults.json or defaults.yaml files that are stored in the root of a salt formula.

CLI Example:

```
salt '*' defaults.get core:users:root
```

The defaults is computed from pillar key. The first entry is considered as the formula namespace.

For example, querying `core:users:root` will try to load `salt://core/defaults.yaml` and `salt://core/defaults.json`.

`salt.modules.defaults.`**`merge`**(*dest*, *src*, *merge_lists=False*, *in_place=True*)

Allows deep merging of dicts in formulas.

merge_lists [False] If True, it will also merge lists instead of replace their items.

in_place [True] If True, it will merge into dest dict, if not it will make a new copy from that dict and return it.

> CLI Example: .. code-block:: bash
>
> salt '*' default.merge a=b d=e

It is more typical to use this in a templating language in formulas, instead of directly on the command-line.

`salt.modules.defaults.`**`update`**(*dest*, *defaults*, *merge_lists=True*, *in_place=True*)

Allows to set defaults for group of data set e.g. group for nodes.

> This function is a combination of defaults.merge and defaults.deepcopy to avoid redundant in jinja.
>
> Example:

```
group01:
  defaults:
    enabled: True
    extra:
      - test
      - stage
  nodes:
    host01:
      index: foo
      upstream: bar
    host02:
      index: foo2
      upstream: bar2
```

```
{% do salt['defaults.update'](group01.nodes, group01.defaults) %}
```

> Each node will look like the following:

```
host01:
  enabled: True
  index: foo
  upstream: bar
  extra:
    - test
    - stage
```

merge_lists [True] If True, it will also merge lists instead of replace their items.

in_place [True] If True, it will merge into dest dict. if not it will make a new copy from that dict and return it.

It is more typical to use this in a templating language in formulas, instead of directly on the command-line.

21.9.103 salt.modules.devmap

Device-Mapper module

`salt.modules.devmap.`**`multipath_flush`**`(`*device*`)`

Device-Mapper Multipath flush

CLI Example:

```
salt '*' devmap.multipath_flush mpath1
```

`salt.modules.devmap.`**`multipath_list`**`()`

Device-Mapper Multipath list

CLI Example:

```
salt '*' devmap.multipath_list
```

21.9.104 salt.modules.dig

Compendium of generic DNS utilities. The 'dig' command line tool must be installed in order to use this module.

`salt.modules.dig.`**`A`**`(`*host, nameserver=None*`)`

Return the A record for `host`.

Always returns a list.

CLI Example:

```
salt ns1 dig.A www.google.com
```

`salt.modules.dig.`**`AAAA`**`(`*host, nameserver=None*`)`

Return the AAAA record for `host`.

Always returns a list.

CLI Example:

```
salt ns1 dig.AAAA www.google.com
```

`salt.modules.dig.`**`MX`**`(`*domain, resolve=False, nameserver=None*`)`

Return a list of lists for the MX of `domain`.

If the `resolve` argument is True, resolve IPs for the servers.

It's limited to one IP, because although in practice it's very rarely a round robin, it is an acceptable configuration and pulling just one IP lets the data be similar to the non-resolved version. If you think an MX has multiple IPs, don't use the resolver here, resolve them in a separate step.

CLI Example:

```
salt ns1 dig.MX google.com
```

`salt.modules.dig.`**`NS`**`(`*domain, resolve=True, nameserver=None*`)`

Return a list of IPs of the nameservers for `domain`

If `resolve` is False, don't resolve names.

CLI Example:

```
salt ns1 dig.NS google.com
```

`salt.modules.dig.`**`SPF`**`(`*domain, record='SPF', nameserver=None*`)`

Return the allowed IPv4 ranges in the SPF record for `domain`.

If record is `SPF` and the SPF record is empty, the TXT record will be searched automatically. If you know the domain uses TXT and not SPF, specifying that will save a lookup.

CLI Example:

```
salt ns1 dig.SPF google.com
```

`salt.modules.dig.`**`TXT`**`(`*host, nameserver=None*`)`

Return the TXT record for `host`.

Always returns a list.

CLI Example:

```
salt ns1 dig.TXT google.com
```

`salt.modules.dig.`**`check_ip`**`(`*addr*`)`

Check if address is a valid IP. returns True if valid, otherwise False.

CLI Example:

```
salt ns1 dig.check_ip 127.0.0.1
salt ns1 dig.check_ip 1111:2222:3333:4444:5555:6666:7777:8888
```

21.9.105 salt.modules.disk

Module for managing disks and blockdevices

`salt.modules.disk.`**`blkid`**`(`*device=None, token=None*`)`

Return block device attributes: UUID, LABEL, etc. This function only works on systems where blkid is available.

device Device name from the system

token Any valid token used for the search

CLI Example:

```
salt '*' disk.blkid
salt '*' disk.blkid /dev/sda
salt '*' disk.blkid token='UUID=6a38ee5-7235-44e7-8b22-816a403bad5d'
salt '*' disk.blkid token='TYPE=ext4'
```

`salt.modules.disk.`**`dump`**`(`*device, args=None*`)`

Return all contents of dumpe2fs for a specified device

CLI Example: .. code-block:: bash

salt '*' disk.dump /dev/sda1

`salt.modules.disk.`**`format_`**(*device*, *fs_type='ext4'*, *inode_size=None*, *lazy_itable_init=None*, *fat=None*, *force=False*)

Format a filesystem onto a device

New in version 2016.11.0.

device The device in which to create the new filesystem

fs_type The type of filesystem to create

inode_size Size of the inodes

This option is only enabled for ext and xfs filesystems

lazy_itable_init If enabled and the uninit_bg feature is enabled, the inode table will not be fully initialized by mke2fs. This speeds up filesystem initialization noticeably, but it requires the kernel to finish initializing the filesystem in the background when the filesystem is first mounted. If the option value is omitted, it defaults to 1 to enable lazy inode table zeroing.

This option is only enabled for ext filesystems

fat FAT size option. Can be 12, 16 or 32, and can only be used on fat or vfat filesystems.

force Force mke2fs to create a filesystem, even if the specified device is not a partition on a block special device. This option is only enabled for ext and xfs filesystems

This option is dangerous, use it with caution.

CLI Example:

```
salt '*' disk.format /dev/sdX1
```

`salt.modules.disk.`**`fstype`**(*device*)

Return the filesystem name of the specified device

New in version 2016.11.0.

device The name of the device

CLI Example:

```
salt '*' disk.fstype /dev/sdX1
```

`salt.modules.disk.`**`hdparms`**(*disks*, *args=None*)

Retrieve all info's for all disks parse 'em into a nice dict (which, considering hdparms output, is quite a hassle)

New in version 2016.3.0.

CLI Example: .. code-block:: bash

salt '*' disk.hdparms /dev/sda

`salt.modules.disk.`**`hpa`**(*disks*, *size=None*)

Get/set Host Protected Area settings

T13 INCITS 346-2001 (1367D) defines the BEER (Boot Engineering Extension Record) and PARTIES (Protected Area Run Time Interface Extension Services), allowing for a Host Protected Area on a disk.

It's often used by OEMS to hide parts of a disk, and for overprovisioning SSD's

> **Warning:** Setting the HPA might clobber your data, be very careful with this on active disks!

New in version 2016.3.0.

CLI Example:

```
salt '*' disk.hpa /dev/sda
salt '*' disk.hpa /dev/sda 5%
salt '*' disk.hpa /dev/sda 10543256
```

`salt.modules.disk.`**`inodeusage`**`(`*args=None*`)`
Return inode usage information for volumes mounted on this minion

CLI Example:

```
salt '*' disk.inodeusage
```

`salt.modules.disk.`**`iostat`**`(`*interval=1, count=5, disks=None*`)`
Gather and return (averaged) IO stats.

New in version 2016.3.0.

Changed in version 2016.11.4: Added support for AIX

CLI Example:

```
salt '*' disk.iostat 1 5 disks=sda
```

`salt.modules.disk.`**`percent`**`(`*args=None*`)`
Return partition information for volumes mounted on this minion

CLI Example:

```
salt '*' disk.percent /var
```

`salt.modules.disk.`**`resize2fs`**`(`*device*`)`
Resizes the filesystem.

CLI Example: .. code-block:: bash

salt '*' disk.resize2fs /dev/sda1

`salt.modules.disk.`**`smart_attributes`**`(`*dev, attributes=None, values=None*`)`
Fetch SMART attributes Providing attributes will deliver only requested attributes Providing values will deliver only requested values for attributes

Default is the Backblaze recommended set (https://www.backblaze.com/blog/hard-drive-smart-stats/): (5,187,188,197,198)

New in version 2016.3.0.

CLI Example:

```
salt '*' disk.smart_attributes /dev/sda
salt '*' disk.smart_attributes /dev/sda attributes=(5,187,188,197,198)
```

`salt.modules.disk.`**`tune`**`(`*device, **kwargs*`)`
Set attributes for the specified device

CLI Example:

```
salt '*' disk.tune /dev/sda1 read-ahead=1024 read-write=True
```

Valid options are: `read-ahead`, `filesystem-read-ahead`, `read-only`, `read-write`.

See the `blockdev(8)` manpage for a more complete description of these options.

`salt.modules.disk.`**`usage`**`(`*args=None*`)`

Return usage information for volumes mounted on this minion

Changed in version 2019.2.0: Default for SunOS changed to 1 kilobyte blocks

CLI Example:

```
salt '*' disk.usage
```

`salt.modules.disk.`**`wipe`**`(`*device*`)`

Remove the filesystem information

CLI Example:

```
salt '*' disk.wipe /dev/sda1
```

21.9.106 salt.modules.djangomod

Manage Django sites

`salt.modules.djangomod.`**`collectstatic`**`(`*settings_module, bin_env=None, no_post_process=False, ignore=None, dry_run=False, clear=False, link=False, no_default_ignore=False, pythonpath=None, env=None, runas=None*`)`

Collect static files from each of your applications into a single location that can easily be served in production.

CLI Example:

```
salt '*' django.collectstatic <settings_module>
```

`salt.modules.djangomod.`**`command`**`(`*settings_module, command, bin_env=None, pythonpath=None, env=None, runas=None, *args, **kwargs*`)`

Run arbitrary django management command

CLI Example:

```
salt '*' django.command <settings_module> <command>
```

`salt.modules.djangomod.`**`createsuperuser`**`(`*settings_module, username, email, bin_env=None, database=None, pythonpath=None, env=None, runas=None*`)`

Create a super user for the database. This function defaults to use the `--noinput` flag which prevents the creation of a password for the superuser.

CLI Example:

```
salt '*' django.createsuperuser <settings_module> user user@example.com
```

`salt.modules.djangomod.`**`loaddata`**`(`*settings_module, fixtures, bin_env=None, database=None, pythonpath=None, env=None*`)`

Load fixture data

Fixtures: comma separated list of fixtures to load

CLI Example:

```
salt '*' django.loaddata <settings_module> <comma delimited list of fixtures>
```

`salt.modules.djangomod.`**`migrate`**(*settings_module*, *app_label=None*, *migration_name=None*, *bin_env=None*, *database=None*, *pythonpath=None*, *env=None*, *noinput=True*, *runas=None*)

Run migrate

Execute the Django-Admin migrate command (requires Django 1.7 or higher).

New in version 3000.

settings_module Specifies the settings module to use. The settings module should be in Python package syntax, e.g. mysite.settings. If this isn't provided, django-admin will use the DJANGO_SETTINGS_MODULE environment variable.

app_label Specific app to run migrations for, instead of all apps. This may involve running other apps' migrations too, due to dependencies.

migration_name Named migration to be applied to a specific app. Brings the database schema to a state where the named migration is applied, but no later migrations in the same app are applied. This may involve unapplying migrations if you have previously migrated past the named migration. Use the name zero to unapply all migrations for an app.

bin_env Path to pip (or to a virtualenv). This can be used to specify the path to the pip to use when more than one Python release is installed (e.g. `/usr/bin/pip-2.7` or `/usr/bin/pip-2.6`. If a directory path is specified, it is assumed to be a virtualenv.

database Database to migrate. Defaults to 'default'.

pythonpath Adds the given filesystem path to the Python import search path. If this isn't provided, django-admin will use the PYTHONPATH environment variable.

env A list of environment variables to be set prior to execution.

Example:

```
module.run:
  - name: django.migrate
  - settings_module: my_django_app.settings
  - env:
    - DATABASE_USER: 'mydbuser'
```

noinput Suppresses all user prompts. Defaults to True.

runas The user name to run the command as.

CLI Example:

```
salt '*' django.migrate <settings_module>
salt '*' django.migrate <settings_module> <app_label>
salt '*' django.migrate <settings_module> <app_label> <migration_name>
```

`salt.modules.djangomod.`**`syncdb`**(*settings_module*, *bin_env=None*, *migrate=False*, *database=None*, *pythonpath=None*, *env=None*, *noinput=True*, *runas=None*)

Run syncdb

Execute the Django-Admin syncdb command, if South is available on the minion the `migrate` option can be passed as `True` calling the migrations to run after the syncdb completes

NOTE: The syncdb command was deprecated in Django 1.7 and removed in Django 1.9. For Django versions 1.9 or higher use the *migrate* command instead.

CLI Example:

```
salt '*' django.syncdb <settings_module>
```

21.9.107 salt.modules.dnsmasq

Module for managing dnsmasq

`salt.modules.dnsmasq.`**`fullversion`**`()`

Shows installed version of dnsmasq and compile options.

CLI Example:

```
salt '*' dnsmasq.fullversion
```

`salt.modules.dnsmasq.`**`get_config`**`(`*config_file='/etc/dnsmasq.conf'*`)`

Dumps all options from the config file.

config_file The location of the config file from which to obtain contents. Defaults to `/etc/dnsmasq.conf`.

CLI Examples:

```
salt '*' dnsmasq.get_config
salt '*' dnsmasq.get_config config_file=/etc/dnsmasq.conf
```

`salt.modules.dnsmasq.`**`set_config`**`(`*config_file='/etc/dnsmasq.conf', follow=True, **kwargs*`)`

Sets a value or a set of values in the specified file. By default, if conf-dir is configured in this file, salt will attempt to set the option in any file inside the conf-dir where it has already been enabled. If it does not find it inside any files, it will append it to the main config file. Setting follow to False will turn off this behavior.

If a config option currently appears multiple times (such as dhcp-host, which is specified at least once per host), the new option will be added to the end of the main config file (and not to any includes). If you need an option added to a specific include file, specify it as the config_file.

Parameters

- **`config_file`** (*string*) – config file where settings should be updated / added.
- **`follow`** (*bool*) – attempt to set the config option inside any file within the `conf-dir` where it has already been enabled.
- **`kwargs`** – key value pairs that contain the configuration settings that you want set.

CLI Examples:

```
salt '*' dnsmasq.set_config domain=mydomain.com
salt '*' dnsmasq.set_config follow=False domain=mydomain.com
salt '*' dnsmasq.set_config config_file=/etc/dnsmasq.conf domain=mydomain.com
```

`salt.modules.dnsmasq.`**`version`**`()`

Shows installed version of dnsmasq.

CLI Example:

```
salt '*' dnsmasq.version
```

21.9.108 salt.modules.dnsutil

Compendium of generic DNS utilities.

Note: Some functions in the `dnsutil` execution module depend on `dig`.

`salt.modules.dnsutil.`**`A`**`(`*host*, *nameserver=None*`)`

Return the A record(s) for `host`.

Always returns a list.

CLI Example:

```
salt ns1 dnsutil.A www.google.com
```

`salt.modules.dnsutil.`**`AAAA`**`(`*host*, *nameserver=None*`)`

Return the AAAA record(s) for `host`.

Always returns a list.

New in version 2014.7.5.

CLI Example:

```
salt ns1 dnsutil.AAAA www.google.com
```

`salt.modules.dnsutil.`**`MX`**`(`*domain*, *resolve=False*, *nameserver=None*`)`

Return a list of lists for the MX of `domain`.

If the 'resolve' argument is True, resolve IPs for the servers.

It's limited to one IP, because although in practice it's very rarely a round robin, it is an acceptable configuration and pulling just one IP lets the data be similar to the non-resolved version. If you think an MX has multiple IPs, don't use the resolver here, resolve them in a separate step.

CLI Example:

```
salt ns1 dnsutil.MX google.com
```

`salt.modules.dnsutil.`**`NS`**`(`*domain*, *resolve=True*, *nameserver=None*`)`

Return a list of IPs of the nameservers for `domain`

If 'resolve' is False, don't resolve names.

CLI Example:

```
salt ns1 dnsutil.NS google.com
```

`salt.modules.dnsutil.`**`SPF`**`(`*domain*, *record='SPF'*, *nameserver=None*`)`

Return the allowed IPv4 ranges in the SPF record for `domain`.

If record is `SPF` and the SPF record is empty, the TXT record will be searched automatically. If you know the domain uses TXT and not SPF, specifying that will save a lookup.

CLI Example:

```
salt ns1 dnsutil.SPF google.com
```

`salt.modules.dnsutil.`**`check_ip`**`(`*ip_addr*`)`

Check that string ip_addr is a valid IP

CLI Example:

```
salt ns1 dnsutil.check_ip 127.0.0.1
```

`salt.modules.dnsutil.`**`hosts_append`**(*hostsfile='/etc/hosts', ip_addr=None, entries=None*)

Append a single line to the /etc/hosts file.

CLI Example:

```
salt '*' dnsutil.hosts_append /etc/hosts 127.0.0.1 ad1.yuk.co,ad2.yuk.co
```

`salt.modules.dnsutil.`**`hosts_remove`**(*hostsfile='/etc/hosts', entries=None*)

Remove a host from the /etc/hosts file. If doing so will leave a line containing only an IP address, then the line will be deleted. This function will leave comments and blank lines intact.

CLI Examples:

```
salt '*' dnsutil.hosts_remove /etc/hosts ad1.yuk.co
salt '*' dnsutil.hosts_remove /etc/hosts ad2.yuk.co,ad1.yuk.co
```

`salt.modules.dnsutil.`**`parse_hosts`**(*hostsfile='/etc/hosts', hosts=None*)

Parse /etc/hosts file.

CLI Example:

```
salt '*' dnsutil.parse_hosts
```

`salt.modules.dnsutil.`**`parse_zone`**(*zonefile=None, zone=None*)

Parses a zone file. Can be passed raw zone data on the API level.

CLI Example:

```
salt ns1 dnsutil.parse_zone /var/lib/named/example.com.zone
```

`salt.modules.dnsutil.`**`serial`**(*zone='', update=False*)

Return, store and update a dns serial for your zone files.

zone: a keyword for a specific zone

update: store an updated version of the serial in a grain

If `update` is False, the function will retrieve an existing serial or return the current date if no serial is stored. Nothing will be stored

If `update` is True, the function will set the serial to the current date if none exist or if the existing serial is for a previous date. If a serial for greater than the current date is already stored, the function will increment it.

This module stores the serial in a grain, you can explicitly set the stored value as a grain named `dnsserial_<zone_name>`.

CLI Example:

```
salt ns1 dnsutil.serial example.com
```

21.9.109 salt.modules.dockercompose module

Module to import docker-compose via saltstack

New in version 2016.3.0.

maintainer Jean Praloran <jeanpralo@gmail.com>

maturity new

depends docker-compose>=1.5

platform all

Introduction

This module allows one to deal with docker-compose file in a directory.

This is a first version only, the following commands are missing at the moment but will be built later on if the community is interested in this module:

- run
- logs
- port
- scale

Installation Prerequisites

This execution module requires at least version 1.4.0 of both docker-compose and Docker. docker-compose can easily be installed using `pip.install`:

```
salt myminion pip.install docker-compose>=1.5.0
```

How to use this module?

In order to use the module if you have no docker-compose file on the server you can issue the command create, it takes two arguments the path where the docker-compose.yml will be stored and the content of this latter:

```
# salt-call -l debug dockercompose.create /tmp/toto '
database:
image: mongo:3.0
command: mongod --smallfiles --quiet --logpath=/dev/null
'
```

Then you can execute a list of method defined at the bottom with at least one argument (the path where the docker-compose.yml will be read) and an optional python list which corresponds to the services names:

```
# salt-call -l debug dockercompose.up /tmp/toto
# salt-call -l debug dockercompose.restart /tmp/toto '[database]'
# salt-call -l debug dockercompose.stop /tmp/toto
# salt-call -l debug dockercompose.rm /tmp/toto
```

Docker-compose method supported

- up
- restart
- stop

- start
- pause
- unpause
- kill
- rm
- ps
- pull
- build

Functions

- **docker-compose.yml management**
    - *dockercompose.create*
    - *dockercompose.get*
- **Manage containers**
    - *dockercompose.restart*
    - *dockercompose.stop*
    - *dockercompose.pause*
    - *dockercompose.unpause*
    - *dockercompose.start*
    - *dockercompose.kill*
    - *dockercompose.rm*
    - *dockercompose.up*
- **Manage containers image:**
    - *dockercompose.pull*
    - *dockercompose.build*
- **Gather information about containers:**
    - *dockercompose.ps*
- **Manage service definitions:**
    - *dockercompose.service_create*
    - *dockercompose.service_upsert*
    - *dockercompose.service_remove*
    - *dockercompose.service_set_tag*

Detailed Function Documentation

`salt.modules.dockercompose.`**`build`**`(`*path*, *service_names=None*`)`

Build image for containers in the docker-compose file, service_names is a python list, if omitted build images for all containers. Please note that at the moment the module does not allow you to upload your Dockerfile, nor any other file you could need with your docker-compose.yml, you will have to make sure the files you need are actually in the directory specified in the *build* keyword

path Path where the docker-compose file is stored on the server

service_names If specified will pull only the image for the specified services

CLI Example:

```
salt myminion dockercompose.build /path/where/docker-compose/stored
salt myminion dockercompose.build /path/where/docker-compose/stored '[janus]'
```

`salt.modules.dockercompose.`**`create`**`(`*path*, *docker_compose*`)`

Create and validate a docker-compose file into a directory

path Path where the docker-compose file will be stored on the server

docker_compose docker_compose file

CLI Example:

```
salt mymininion dockercompose.create /path/where/docker-compose/stored content
```

`salt.modules.dockercompose.`**`get`**`(`*path*`)`

Get the content of the docker-compose file into a directory

path Path where the docker-compose file is stored on the server

CLI Example:

```
salt myminion dockercompose.get /path/where/docker-compose/stored
```

`salt.modules.dockercompose.`**`kill`**`(`*path*, *service_names=None*`)`

Kill containers in the docker-compose file, service_names is a python list, if omitted kill all containers

path Path where the docker-compose file is stored on the server

service_names If specified will kill only the specified services

CLI Example:

```
salt myminion dockercompose.kill /path/where/docker-compose/stored
salt myminion dockercompose.kill /path/where/docker-compose/stored '[janus]'
```

`salt.modules.dockercompose.`**`pause`**`(`*path*, *service_names=None*`)`

Pause running containers in the docker-compose file, service_names is a python list, if omitted pause all containers

path Path where the docker-compose file is stored on the server

service_names If specified will pause only the specified services

CLI Example:

```
salt myminion dockercompose.pause /path/where/docker-compose/stored
salt myminion dockercompose.pause /path/where/docker-compose/stored '[janus]'
```

`salt.modules.dockercompose.`**`ps`**`(`*path*`)`

List all running containers and report some information about them

path Path where the docker-compose file is stored on the server

CLI Example:

```
salt myminion dockercompose.ps /path/where/docker-compose/stored
```

`salt.modules.dockercompose.`**`pull`**`(`*path, service_names=None*`)`

Pull image for containers in the docker-compose file, service_names is a python list, if omitted pull all images

path Path where the docker-compose file is stored on the server

service_names If specified will pull only the image for the specified services

CLI Example:

```
salt myminion dockercompose.pull /path/where/docker-compose/stored
salt myminion dockercompose.pull /path/where/docker-compose/stored '[janus]'
```

`salt.modules.dockercompose.`**`restart`**`(`*path, service_names=None*`)`

Restart container(s) in the docker-compose file, service_names is a python list, if omitted restart all containers

path Path where the docker-compose file is stored on the server

service_names If specified will restart only the specified services

CLI Example:

```
salt myminion dockercompose.restart /path/where/docker-compose/stored
salt myminion dockercompose.restart /path/where/docker-compose/stored '[janus]'
```

`salt.modules.dockercompose.`**`rm`**`(`*path, service_names=None*`)`

Remove stopped containers in the docker-compose file, service_names is a python list, if omitted remove all stopped containers

path Path where the docker-compose file is stored on the server

service_names If specified will remove only the specified stopped services

CLI Example:

```
salt myminion dockercompose.rm /path/where/docker-compose/stored
salt myminion dockercompose.rm /path/where/docker-compose/stored '[janus]'
```

`salt.modules.dockercompose.`**`service_create`**`(`*path, service_name, definition*`)`

Create the definition of a docker-compose service This fails when the service already exists This does not pull or up the service This wil re-write your yaml file. Comments will be lost. Indentation is set to 2 spaces

path Path where the docker-compose file is stored on the server

service_name Name of the service to create

definition Service definition as yaml or json string

CLI Example:

```
salt myminion dockercompose.service_create /path/where/docker-compose/stored
↪service_name definition
```

`salt.modules.dockercompose.`**`service_remove`**`(`*path*, *service_name*`)`

Remove the definition of a docker-compose service This does not rm the container This wil re-write your yaml file. Comments will be lost. Indentation is set to 2 spaces

path Path where the docker-compose file is stored on the server

service_name Name of the service to remove

CLI Example:

```
salt myminion dockercompose.service_remove /path/where/docker-compose/stored␣
↪service_name
```

`salt.modules.dockercompose.`**`service_set_tag`**`(`*path*, *service_name*, *tag*`)`

Change the tag of a docker-compose service This does not pull or up the service This wil re-write your yaml file. Comments will be lost. Indentation is set to 2 spaces

path Path where the docker-compose file is stored on the server

service_name Name of the service to remove

tag Name of the tag (often used as version) that the service image should have

CLI Example:

```
salt myminion dockercompose.service_create /path/where/docker-compose/stored␣
↪service_name tag
```

`salt.modules.dockercompose.`**`service_upsert`**`(`*path*, *service_name*, *definition*`)`

Create or update the definition of a docker-compose service This does not pull or up the service This wil re-write your yaml file. Comments will be lost. Indentation is set to 2 spaces

path Path where the docker-compose file is stored on the server

service_name Name of the service to create

definition Service definition as yaml or json string

CLI Example:

```
salt myminion dockercompose.service_upsert /path/where/docker-compose/stored␣
↪service_name definition
```

`salt.modules.dockercompose.`**`start`**`(`*path*, *service_names=None*`)`

Start containers in the docker-compose file, service_names is a python list, if omitted start all containers

path Path where the docker-compose file is stored on the server

service_names If specified will start only the specified services

CLI Example:

```
salt myminion dockercompose.start /path/where/docker-compose/stored
salt myminion dockercompose.start /path/where/docker-compose/stored '[janus]'
```

`salt.modules.dockercompose.`**`stop`**`(`*path*, *service_names=None*`)`

Stop running containers in the docker-compose file, service_names is a python list, if omitted stop all containers

path Path where the docker-compose file is stored on the server

service_names If specified will stop only the specified services

CLI Example:

```
salt myminion dockercompose.stop /path/where/docker-compose/stored
salt myminion dockercompose.stop  /path/where/docker-compose/stored '[janus]'
```

`salt.modules.dockercompose.`**`unpause`**`(`*path*, *service_names=None*`)`

Un-Pause containers in the docker-compose file, service_names is a python list, if omitted unpause all containers

path Path where the docker-compose file is stored on the server

service_names If specified will un-pause only the specified services

CLI Example:

```
salt myminion dockercompose.pause /path/where/docker-compose/stored
salt myminion dockercompose.pause /path/where/docker-compose/stored '[janus]'
```

`salt.modules.dockercompose.`**`up`**`(`*path*, *service_names=None*`)`

Create and start containers defined in the docker-compose.yml file located in path, service_names is a python list, if omitted create and start all containers

path Path where the docker-compose file is stored on the server

service_names If specified will create and start only the specified services

CLI Example:

```
salt myminion dockercompose.up /path/where/docker-compose/stored
salt myminion dockercompose.up /path/where/docker-compose/stored '[janus]'
```

21.9.110 salt.modules.dockermod

Management of Docker Containers

New in version 2015.8.0.

Changed in version 2017.7.0: This module has replaced the legacy docker execution module.

depends docker Python module

Note: Older releases of the Python bindings for Docker were called docker-py in PyPI. All releases of docker, and releases of docker-py >= 1.6.0 are supported. These python bindings can easily be installed using *`pip.install`*:

```
salt myminion pip.install docker
```

To upgrade from docker-py to docker, you must first uninstall docker-py, and then install docker:

```
salt myminion pip.uninstall docker-py
salt myminion pip.install docker
```

Authentication

If you have previously performed a `docker login` from the minion, then the credentials saved in `~/.docker/config.json` will be used for any actions which require authentication. If not, then credentials can be configured in any of the following locations:

- Minion config file

- Grains
- Pillar data
- Master config file (requires `pillar_opts` to be set to `True` in Minion config file in order to work)

Important: Versions prior to 3000 require that Docker credentials are configured in Pillar data. Be advised that Pillar data is still recommended though, because this keeps the configuration from being stored on the Minion.

Also, keep in mind that if one gets your `~/.docker/config.json`, the password can be decoded from its contents.

The configuration schema is as follows:

```
docker-registries:
  <registry_url>:
    username: <username>
    password: <password>
```

For example:

```
docker-registries:
  hub:
    username: foo
    password: s3cr3t
```

Note: As of the 2016.3.7, 2016.11.4, and 2017.7.0 releases of Salt, credentials for the Docker Hub can be configured simply by specifying `hub` in place of the registry URL. In earlier releases, it is necessary to specify the actual registry URL for the Docker Hub (i.e. `https://index.docker.io/v1/`).

More than one registry can be configured. Salt will look for Docker credentials in the `docker-registries` Pillar key, as well as any key ending in `-docker-registries`. For example:

```
docker-registries:
  'https://mydomain.tld/registry:5000':
    username: foo
    password: s3cr3t

foo-docker-registries:
  https://index.foo.io/v1/:
    username: foo
    password: s3cr3t

bar-docker-registries:
  https://index.bar.io/v1/:
    username: foo
    password: s3cr3t
```

To login to the configured registries, use the *`docker.login`* function. This only needs to be done once for a given registry, and it will store/update the credentials in `~/.docker/config.json`.

Note: For Salt releases before 2016.3.7 and 2016.11.4, *`docker.login`* is not available. Instead, Salt will try to authenticate using each of your configured registries for each push/pull, behavior which is not correct and has been resolved in newer releases.

Configuration Options

The following configuration options can be set to fine-tune how Salt uses Docker:

- `docker.url`: URL to the docker service (default: local socket).
- `docker.version`: API version to use (should not need to be set manually in the vast majority of cases)
- `docker.exec_driver`: Execution driver to use, one of `nsenter`, `lxc-attach`, or `docker-exec`. See the *Executing Commands Within a Running Container* section for more details on how this config parameter is used.

These configuration options are retrieved using *`config.get`* (click the link for further information).

Executing Commands Within a Running Container

Note: With the release of Docker 1.13.1, the Execution Driver has been removed. Starting in versions 2016.3.6, 2016.11.4, and 2017.7.0, Salt defaults to using `docker exec` to run commands in containers, however for older Salt releases it will be necessary to set the `docker.exec_driver` config option to either `docker-exec` or `nsenter` for Docker versions 1.13.1 and newer.

Multiple methods exist for executing commands within Docker containers:

- lxc-attach: Default for older versions of docker
- nsenter: Enters container namespace to run command
- docker-exec: Native support for executing commands in Docker containers (added in Docker 1.3)

Adding a configuration option (see *`config.get`*) called `docker.exec_driver` will tell Salt which execution driver to use:

```
docker.exec_driver: docker-exec
```

If this configuration option is not found, Salt will use the appropriate interface (either nsenter or lxc-attach) based on the `Execution Driver` value returned from `docker info`. docker-exec will not be used by default, as it is presently (as of version 1.6.2) only able to execute commands as the effective user of the container. Thus, if a `USER` directive was used to run as a non-privileged user, docker-exec would be unable to perform the action as root. Salt can still use docker-exec as an execution driver, but must be explicitly configured (as in the example above) to do so at this time.

If possible, try to manually specify the execution driver, as it will save Salt a little work.

This execution module provides functions that shadow those from the *`cmd`* module. They are as follows:

- *`docker.retcode`*
- *`docker.run`*
- *`docker.run_all`*
- *`docker.run_stderr`*
- *`docker.run_stdout`*
- *`docker.script`*
- *`docker.script_retcode`*

Detailed Function Documentation

class `salt.modules.dockermod.`**`DockerJSONDecoder`**(**, object_hook=None, parse_float=None, parse_int=None, parse_constant=None, strict=True, object_pairs_hook=None*)

`decode`(*s, _w=None*)
Return the Python representation of `s` (a `str` instance containing a JSON document).

`salt.modules.dockermod.`**`apply_`**(*name, mods=None, **kwargs*)
New in version 2019.2.0.

Apply states! This function will call highstate or state.sls based on the arguments passed in, `apply` is intended to be the main gateway for all state executions.

CLI Example:

```
salt 'docker' docker.apply web01
salt 'docker' docker.apply web01 test
salt 'docker' docker.apply web01 test,pkgs
```

`salt.modules.dockermod.`**`build`**(*path=None, repository=None, tag=None, cache=True, rm=True, api_response=False, fileobj=None, dockerfile=None, buildargs=None*)
Changed in version 2018.3.0: If the built image should be tagged, then the repository and tag must now be passed separately using the `repository` and `tag` arguments, rather than together in the (now deprecated) `image` argument.

Builds a docker image from a Dockerfile or a URL

path Path to directory on the Minion containing a Dockerfile

repository Optional repository name for the image being built

New in version 2018.3.0.

tag [latest] Tag name for the image (required if `repository` is passed)

New in version 2018.3.0.

image Deprecated since version 2018.3.0: Use both `repository` and `tag` instead

cache [True] Set to `False` to force the build process not to use the Docker image cache, and pull all required intermediate image layers

rm [True] Remove intermediate containers created during build

api_response [False] If `True`: an `API_Response` key will be present in the return data, containing the raw output from the Docker API.

fileobj Allows for a file-like object containing the contents of the Dockerfile to be passed in place of a file `path` argument. This argument should not be used from the CLI, only from other Salt code.

dockerfile Allows for an alternative Dockerfile to be specified. Path to alternative Dockefile is relative to the build path for the Docker container.

New in version 2016.11.0.

buildargs A dictionary of build arguments provided to the docker build process.

RETURN DATA

A dictionary containing one or more of the following keys:

- `Id` - ID of the newly-built image

- `Time_Elapsed` - Time in seconds taken to perform the build
- `Intermediate_Containers` - IDs of containers created during the course of the build process

  (Only present if rm=False)
- **`Images` - A dictionary containing one or more of the following keys:**
  - `Already_Pulled` - Layers that that were already present on the Minion
  - `Pulled` - Layers that that were pulled

  (Only present if the image specified by the "repository" and "tag" arguments was not present on the Minion, or if cache=False)
- `Status` - A string containing a summary of the pull action (usually a message saying that an image was downloaded, or that it was up to date).

  (Only present if the image specified by the "repository" and "tag" arguments was not present on the Minion, or if cache=False)

CLI Example:

```
salt myminion docker.build /path/to/docker/build/dir
salt myminion docker.build https://github.com/myuser/myrepo.git␣
↪repository=myimage tag=latest
salt myminion docker.build /path/to/docker/build/dir dockerfile=Dockefile.
↪different repository=myimage tag=dev
```

`salt.modules.dockermod.`**`call`**(*name*, *function*, **args*, ***kwargs*)

Executes a Salt function inside a running container

New in version 2016.11.0.

The container does not need to have Salt installed, but Python is required.

name Container name or ID

function Salt execution module function

CLI Example:

```
salt myminion docker.call test.ping
salt myminion test.arg arg1 arg2 key1=val1
salt myminion dockerng.call compassionate_mirzakhani test.arg arg1 arg2 key1=val1
```

`salt.modules.dockermod.`**`commit`**(*name*, *repository*, *tag='latest'*, *message=None*, *author=None*)

Changed in version 2018.3.0: The repository and tag must now be passed separately using the `repository` and `tag` arguments, rather than together in the (now deprecated) `image` argument.

Commits a container, thereby promoting it to an image. Equivalent to running the `docker commit` Docker CLI command.

name Container name or ID to commit

repository Repository name for the image being committed

New in version 2018.3.0.

tag [latest] Tag name for the image

New in version 2018.3.0.

image Deprecated since version 2018.3.0: Use both `repository` and `tag` instead

message Commit message (Optional)

author Author name (Optional)

RETURN DATA

A dictionary containing the following keys:

- `Id` - ID of the newly-created image
- `Image` - Name of the newly-created image
- `Time_Elapsed` - Time in seconds taken to perform the commit

CLI Example:

```
salt myminion docker.commit mycontainer myuser/myimage mytag
```

`salt.modules.dockermod.`**`compare_container`**(*first*, *second*, *ignore=None*)

This function is an alias of `compare_containers`.

New in version 2017.7.0.

Changed in version 2018.3.0: Renamed from `docker.compare_container` to `docker.compare_containers` (old function name remains as an alias)

Compare two containers' Config and and HostConfig and return any differences between the two.

first Name or ID of first container

second Name or ID of second container

ignore A comma-separated list (or Python list) of keys to ignore when comparing. This is useful when comparing two otherwise identical containers which have different hostnames.

CLI Examples:

```
salt myminion docker.compare_containers foo bar
salt myminion docker.compare_containers foo bar ignore=Hostname
```

`salt.modules.dockermod.`**`compare_container_networks`**(*first*, *second*)

New in version 2018.3.0.

Returns the differences between two containers' networks. When a network is only present one of the two containers, that network's diff will simply be represented with `True` for the side of the diff in which the network is present) and `False` for the side of the diff in which the network is absent.

This function works by comparing the contents of both containers' `Networks` keys (under `NetworkSettings`) in the return data from *`docker.inspect_container`*. Because each network contains some items that either A) only set at runtime, B) naturally varying from container to container, or both, by default the following keys in each network are examined:

- **Aliases**
- **Links**
- **IPAMConfig**

The exception to this is if `IPAMConfig` is unset (i.e. null) in one container but not the other. This happens when no static IP configuration is set, and automatic IP configuration is in effect. So, in order to report on changes between automatic IP configuration in one container and static IP configuration in another container (as we need to do for the *`docker_container.running`* state), automatic IP configuration will also be checked in these cases.

This function uses the *`docker.compare_container_networks`* minion config option to determine which keys to examine. This provides flexibility in the event that features added in a future Docker release necessitate changes to how Salt compares networks. In these cases, rather than waiting for a new Salt release one can just set *docker.compare_container_networks*.

Changed in version 3000: This config option can now also be set in pillar data and grains. Additionally, it can be set in the master config file, provided that `pillar_opts` is enabled on the minion.

Note: The checks for automatic IP configuration described above only apply if `IPAMConfig` is among the keys set for static IP checks in *docker.compare_container_networks*.

---

first Name or ID of first container (old)

second Name or ID of second container (new)

CLI Example:

```
salt myminion docker.compare_container_networks foo bar
```

`salt.modules.dockermod.`**`compare_containers`**(*first*, *second*, *ignore=None*)

New in version 2017.7.0.

Changed in version 2018.3.0: Renamed from `docker.compare_container` to `docker.compare_containers` (old function name remains as an alias)

Compare two containers' Config and and HostConfig and return any differences between the two.

first Name or ID of first container

second Name or ID of second container

ignore A comma-separated list (or Python list) of keys to ignore when comparing. This is useful when comparing two otherwise identical containers which have different hostnames.

CLI Examples:

```
salt myminion docker.compare_containers foo bar
salt myminion docker.compare_containers foo bar ignore=Hostname
```

`salt.modules.dockermod.`**`compare_networks`**(*first*, *second*, *ignore='Name, Id, Created, Containers'*)

New in version 2018.3.0.

Compare two networks and return any differences between the two

first Name or ID of first container

second Name or ID of second container

ignore [Name,Id,Created,Containers] A comma-separated list (or Python list) of keys to ignore when comparing.

CLI Example:

```
salt myminion docker.compare_network foo bar
```

`salt.modules.dockermod.`**`connect_container_to_network`**(*container*, *net_id*, ***kwargs*)

New in version 2015.8.3.

Changed in version 2017.7.0: Support for `ipv4_address` argument added

Changed in version 2018.3.0: All arguments are now passed through to connect_container_to_network(), allowing for any new arguments added to this function to be supported automagically.

Connect container to network. See the connect_container_to_network() docs for information on supported arguments.

container Container name or ID

net_id Network name or ID

CLI Examples:

```
salt myminion docker.connect_container_to_network web-1 mynet
salt myminion docker.connect_container_to_network web-1 mynet ipv4_address=10.20.
↪0.10
salt myminion docker.connect_container_to_network web-1␣
↪1f9d2454d0872b68dd9e8744c6e7a4c66b86f10abaccc21e14f7f014f729b2bc
```

`salt.modules.dockermod.`**`connected`**`(`*name*, *verbose=False*`)`

New in version 2018.3.0.

Return a list of running containers attached to the specified network

name Network name

verbose [False] If `True`, return extended info about each container (IP configuration, etc.)

CLI Example:

```
salt myminion docker.connected net_name
```

`salt.modules.dockermod.`**`copy_from`**`(`*name*, *source*, *dest*, *overwrite=False*, *makedirs=False*`)`

Copy a file from inside a container to the Minion

name Container name

source Path of the file on the container's filesystem

dest Destination on the Minion. Must be an absolute path. If the destination is a directory, the file will be copied into that directory.

overwrite [False] Unless this option is set to `True`, then if a file exists at the location specified by the `dest` argument, an error will be raised.

makedirs [False] Create the parent directory on the container if it does not already exist.

RETURN DATA

A boolean (`True` if successful, otherwise `False`)

CLI Example:

```
salt myminion docker.copy_from mycontainer /var/log/nginx/access.log /home/myuser
```

`salt.modules.dockermod.`**`copy_to`**`(`*name*, *source*, *dest*, *exec_driver=None*, *overwrite=False*, *makedirs=False*`)`

Copy a file from the host into a container

name Container name

source File to be copied to the container. Can be a local path on the Minion or a remote file from the Salt fileserver.

dest Destination on the container. Must be an absolute path. If the destination is a directory, the file will be copied into that directory.

exec_driver [None] If not passed, the execution driver will be detected as described *above*.

overwrite [False] Unless this option is set to `True`, then if a file exists at the location specified by the `dest` argument, an error will be raised.

makedirs [False] Create the parent directory on the container if it does not already exist.

RETURN DATA

A boolean (`True` if successful, otherwise `False`)

CLI Example:

```
salt myminion docker.copy_to mycontainer /tmp/foo /root/foo
```

`salt.modules.dockermod.`**`create`**(*image, name=None, start=False, skip_translate=None, ignore_collisions=False, validate_ip_addrs=True, client_timeout=60, **kwargs*)

Create a new container

image Image from which to create the container

name Name for the new container. If not provided, Docker will randomly generate one for you (it will be included in the return data).

start [False] If `True`, start container after creating it

New in version 2018.3.0.

skip_translate This function translates Salt CLI or SLS input into the format which docker-py expects. However, in the event that Salt's translation logic fails (due to potential changes in the Docker Remote API, or to bugs in the translation code), this argument can be used to exert granular control over which arguments are translated and which are not.

Pass this argument as a comma-separated list (or Python list) of arguments, and translation for each passed argument name will be skipped. Alternatively, pass `True` and *all* translation will be skipped.

Skipping tranlsation allows for arguments to be formatted directly in the format which docker-py expects. This allows for API changes and other issues to be more easily worked around. An example of using this option to skip translation would be:

```
salt myminion docker.create image=centos:7.3.1611 skip_translate=environment␣
↪environment="{'FOO': 'bar'}"
```

See the following links for more information:

- docker-py Low-level API
- Docker Engine API

ignore_collisions [False] Since many of docker-py's arguments differ in name from their CLI counterparts (with which most Docker users are more familiar), Salt detects usage of these and aliases them to the docker-py version of that argument. However, if both the alias and the docker-py version of the same argument (e.g. `env` and `environment`) are used, an error will be raised. Set this argument to `True` to suppress these errors and keep the docker-py version of the argument.

validate_ip_addrs [True] For parameters which accept IP addresses as input, IP address validation will be performed. To disable, set this to `False`

client_timeout [60] Timeout in seconds for the Docker client. This is not a timeout for this function, but for receiving a response from the API.

Note: This is only used if Salt needs to pull the requested image.

CONTAINER CONFIGURATION ARGUMENTS

auto_remove (or *rm*) [False] Enable auto-removal of the container on daemon side when the container's process exits (analogous to running a docker container with `--rm` on the CLI).

Examples:

- `auto_remove=True`
- `rm=True`

binds Files/directories to bind mount. Each bind mount should be passed in one of the following formats:

- `<host_path>:<container_path> - host_path` is mounted within the container as `container_path` with read-write access.
- `<host_path>:<container_path>:<selinux_context> - host_path` is mounted within the container as `container_path` with read-write access. Additionally, the specified selinux context will be set within the container.
- `<host_path>:<container_path>:<read_only>` - `host_path` is mounted within the container as `container_path`, with the read-only or read-write setting explicitly defined.
- `<host_path>:<container_path>:<read_only>,<selinux_context>` - `host_path` is mounted within the container as `container_path`, with the read-only or read-write setting explicitly defined. Additionally, the specified selinux context will be set within the container.

`<read_only>` can be either `ro` for read-write access, or `ro` for read-only access. When omitted, it is assumed to be read-write.

`<selinux_context>` can be `z` if the volume is shared between multiple containers, or `Z` if the volume should be private.

Note: When both `<read_only>` and `<selinux_context>` are specified, there must be a comma before `<selinux_context>`.

Binds can be expressed as a comma-separated list or a Python list, however in cases where both ro/rw and an selinux context are specified, the binds *must* be specified as a Python list.

Examples:

- `binds=/srv/www:/var/www:ro`
- `binds=/srv/www:/var/www:rw`
- `binds=/srv/www:/var/www`
- `binds="['/srv/www:/var/www:ro,Z']"`
- `binds="['/srv/www:/var/www:rw,Z']"`
- `binds=/srv/www:/var/www:Z`

Note: The second and third examples above are equivalent to each other, as are the last two examples.

blkio_weight Block IO weight (relative weight), accepts a weight value between 10 and 1000.

Example: `blkio_weight=100`

blkio_weight_device Block IO weight (relative device weight), specified as a list of expressions in the format `PATH:WEIGHT`

Example: `blkio_weight_device=/dev/sda:100`

cap_add List of capabilities to add within the container. Can be passed as a comma-separated list or a Python list. Requires Docker 1.2.0 or newer.

Examples:

- `cap_add=SYS_ADMIN,MKNOD`
- `cap_add="[SYS_ADMIN, MKNOD]"`

cap_drop List of capabilities to drop within the container. Can be passed as a comma-separated string or a Python list. Requires Docker 1.2.0 or newer.

Examples:

- `cap_drop=SYS_ADMIN,MKNOD`,
- `cap_drop="[SYS_ADMIN, MKNOD]"`

command (or *cmd*) Command to run in the container

Example: `command=bash` or `cmd=bash`

Changed in version 2015.8.1: `cmd` is now also accepted

cpuset_cpus (or *cpuset*) CPUs on which which to allow execution, specified as a string containing a range (e.g. `0-3`) or a comma-separated list of CPUs (e.g. `0,1`).

Examples:

- `cpuset_cpus="0-3"`
- `cpuset="0,1"`

cpuset_mems Memory nodes on which which to allow execution, specified as a string containing a range (e.g. `0-3`) or a comma-separated list of MEMs (e.g. `0,1`). Only effective on NUMA systems.

Examples:

- `cpuset_mems="0-3"`
- `cpuset_mems="0,1"`

cpu_group The length of a CPU period in microseconds

Example: `cpu_group=100000`

cpu_period Microseconds of CPU time that the container can get in a CPU period

Example: `cpu_period=50000`

cpu_shares CPU shares (relative weight), specified as an integer between 2 and 1024.

Example: `cpu_shares=512`

detach [False] If `True`, run the container's command in the background (daemon mode)

Example: `detach=True`

devices List of host devices to expose within the container

Examples:

- `devices="/dev/net/tun,/dev/xvda1:/dev/xvda1,/dev/xvdb1:/dev/xvdb1:r"`

- `devices="['/dev/net/tun', '/dev/xvda1:/dev/xvda1', '/dev/xvdb1:/dev/xvdb1:r']"`

device_read_bps Limit read rate (bytes per second) from a device, specified as a list of expressions in the format `PATH:RATE`, where `RATE` is either an integer number of bytes, or a string ending in `kb`, `mb`, or `gb`.

Examples:

- `device_read_bps="/dev/sda:1mb,/dev/sdb:5mb"`
- `device_read_bps="['/dev/sda:100mb', '/dev/sdb:5mb']"`

device_read_iops Limit read rate (I/O per second) from a device, specified as a list of expressions in the format `PATH:RATE`, where `RATE` is a number of I/O operations.

Examples:

- `device_read_iops="/dev/sda:1000,/dev/sdb:500"`
- `device_read_iops="['/dev/sda:1000', '/dev/sdb:500']"`

device_write_bps Limit write rate (bytes per second) from a device, specified as a list of expressions in the format `PATH:RATE`, where `RATE` is either an integer number of bytes, or a string ending in `kb`, `mb` or `gb`.

Examples:

- `device_write_bps="/dev/sda:100mb,/dev/sdb:50mb"`
- `device_write_bps="['/dev/sda:100mb', '/dev/sdb:50mb']"`

device_read_iops Limit write rate (I/O per second) from a device, specified as a list of expressions in the format `PATH:RATE`, where `RATE` is a number of I/O operations.

Examples:

- `device_read_iops="/dev/sda:1000,/dev/sdb:500"`
- `device_read_iops="['/dev/sda:1000', '/dev/sdb:500']"`

dns List of DNS nameservers. Can be passed as a comma-separated list or a Python list.

Examples:

- `dns=8.8.8.8,8.8.4.4`
- `dns="['8.8.8.8', '8.8.4.4']"`

Note: To skip IP address validation, use `validate_ip_addrs=False`

dns_opt Additional options to be added to the container's `resolv.conf` file

Example: `dns_opt=ndots:9`

dns_search List of DNS search domains. Can be passed as a comma-separated list or a Python list.

Examples:

- `dns_search=foo1.domain.tld,foo2.domain.tld`
- `dns_search="[foo1.domain.tld, foo2.domain.tld]"`

domainname The domain name to use for the container

Example: `domainname=domain.tld`

entrypoint Entrypoint for the container. Either a string (e.g. `"mycmd --arg1 --arg2"`) or a Python list (e.g. `"['mycmd', '--arg1', '--arg2']"`)

Examples:

- `entrypoint="cat access.log"`
- `entrypoint="['cat', 'access.log']"`

environment (or *env*) Either a dictionary of environment variable names and their values, or a Python list of strings in the format `VARNAME=value`.

Examples:

- `environment='VAR1=value,VAR2=value'`
- `environment="['VAR1=value', 'VAR2=value']"`
- `environment="{'VAR1': 'value', 'VAR2': 'value'}"`

extra_hosts Additional hosts to add to the container's /etc/hosts file. Can be passed as a comma-separated list or a Python list. Requires Docker 1.3.0 or newer.

Examples:

- `extra_hosts=web1:10.9.8.7,web2:10.9.8.8`
- `extra_hosts="['web1:10.9.8.7', 'web2:10.9.8.8']"`
- `extra_hosts="{'web1': '10.9.8.7', 'web2': '10.9.8.8'}"`

Note: To skip IP address validation, use `validate_ip_addrs=False`

group_add List of additional group names and/or IDs that the container process will run as

Examples:

- `group_add=web,network`
- `group_add="['web', 'network']"`

hostname Hostname of the container. If not provided, and if a `name` has been provided, the `hostname` will default to the `name` that was passed.

Example: `hostname=web1`

Warning: If the container is started with `network_mode=host`, the hostname will be overridden by the hostname of the Minion.

interactive (or *stdin_open*): False Leave stdin open, even if not attached

Examples:

- `interactive=True`
- `stdin_open=True`

ipc_mode (or *ipc*) Set the IPC mode for the container. The default behavior is to create a private IPC namespace for the container, but this option can be used to change that behavior:

- `container:<container_name_or_id>` reuses another container shared memory, semaphores and message queues

- `host`: use the host's shared memory, semaphores and message queues

Examples:

- `ipc_mode=container:foo`
- `ipc=host`

Warning: Using `host` gives the container full access to local shared memory and is therefore considered insecure.

isolation Specifies the type of isolation technology used by containers

Example: `isolation=hyperv`

Note: The default value on Windows server is `process`, while the default value on Windows client is `hyperv`. On Linux, only `default` is supported.

---

labels (or *label*) Add metadata to the container. Labels can be set both with and without values:

Examples:

- `labels=foo,bar=baz`
- `labels="['foo', 'bar=baz']"`

Changed in version 2018.3.0: Labels both with and without values can now be mixed. Earlier releases only permitted one method or the other.

links Link this container to another. Links should be specified in the format `<container_name_or_id>:<link_alias>`. Multiple links can be passed, ether as a comma separated list or a Python list.

Examples:

- `links=web1:link1,web2:link2`,
- `links="['web1:link1', 'web2:link2']"`
- `links="{'web1': 'link1', 'web2': 'link2'}"`

log_driver Set container's logging driver. Requires Docker 1.6 or newer.

Example:

- `log_driver=syslog`

Note: The logging driver feature was improved in Docker 1.13 introducing option name changes. Please see Docker's Configure logging drivers documentation for more information.

---

log_opt Config options for the `log_driver` config option. Requires Docker 1.6 or newer.

Example:

- `log_opt="syslog-address=tcp://192.168.0.42,syslog-facility=daemon"`
- `log_opt="['syslog-address=tcp://192.168.0.42', 'syslog-facility=daemon']"`
- `log_opt="{'syslog-address': 'tcp://192.168.0.42', 'syslog-facility': 'daemon'}"`

lxc_conf Additional LXC configuration parameters to set before starting the container.

Examples:

- `lxc_conf="lxc.utsname=docker,lxc.arch=x86_64"`
- `lxc_conf="['lxc.utsname=docker', 'lxc.arch=x86_64']"`
- `lxc_conf="{'lxc.utsname': 'docker', 'lxc.arch': 'x86_64'}"`

Note: These LXC configuration parameters will only have the desired effect if the container is using the LXC execution driver, which has been deprecated for some time.

---

mac_address MAC address to use for the container. If not specified, a random MAC address will be used.

Example: `mac_address=01:23:45:67:89:0a`

mem_limit (or *memory*) [0] Memory limit. Can be specified in bytes or using single-letter units (i.e. `512M`, `2G`, etc.). A value of `0` (the default) means no memory limit.

Examples:

- `mem_limit=512M`
- `memory=1073741824`

mem_swappiness Tune a container's memory swappiness behavior. Accepts an integer between 0 and 100.

Example: `mem_swappiness=60`

memswap_limit (or *memory_swap*) [-1] Total memory limit (memory plus swap). Set to `-1` to disable swap. A value of `0` means no swap limit.

Examples:

- `memswap_limit=1G`
- `memory_swap=2147483648`

network_disabled [False] If `True`, networking will be disabled within the container

Example: `network_disabled=True`

network_mode [bridge] One of the following:

- `bridge` - Creates a new network stack for the container on the docker bridge
- `none` - No networking (equivalent of the Docker CLI argument `--net=none`). Not to be confused with Python's `None`.
- `container:<name_or_id>` - Reuses another container's network stack
- `host` - Use the host's network stack inside the container

Warning: Using `host` mode gives the container full access to the hosts system's services (such as D-Bus), and is therefore considered insecure.

Examples:

- `network_mode=null`
- `network_mode=container:web1`

oom_kill_disable Whether to disable OOM killer

Example: `oom_kill_disable=False`

oom_score_adj An integer value containing the score given to the container in order to tune OOM killer preferences

Example: `oom_score_adj=500`

pid_mode Set to `host` to use the host container's PID namespace within the container. Requires Docker 1.5.0 or newer.

Example: `pid_mode=host`

pids_limit Set the container's PID limit. Set to `-1` for unlimited.

Example: `pids_limit=2000`

port_bindings (or *publish*) Bind exposed ports which were exposed using the `ports` argument to *`docker.create`*. These should be passed in the same way as the `--publish` argument to the `docker run` CLI command:

- `ip:hostPort:containerPort` - Bind a specific IP and port on the host to a specific port within the container.
- `ip::containerPort` - Bind a specific IP and an ephemeral port to a specific port within the container.
- `hostPort:containerPort` - Bind a specific port on all of the host's interfaces to a specific port within the container.
- `containerPort` - Bind an ephemeral port on all of the host's interfaces to a specific port within the container.

Multiple bindings can be separated by commas, or passed as a Python list. The below two examples are equivalent:

- `port_bindings="5000:5000,2123:2123/udp,8080"`
- `port_bindings="['5000:5000', '2123:2123/udp', 8080]"`

Port bindings can also include ranges:

- `port_bindings="14505-14506:4505-4506"`

Note: When specifying a protocol, it must be passed in the `containerPort` value, as seen in the examples above.

ports A list of ports to expose on the container. Can be passed as comma-separated list or a Python list. If the protocol is omitted, the port will be assumed to be a TCP port.

Examples:

- `ports=1111,2222/udp`
- `ports="[1111, '2222/udp']"`

privileged [False] If `True`, runs the exec process with extended privileges

Example: `privileged=True`

publish_all_ports (or *publish_all*): False Publish all ports to the host

Example: `publish_all_ports=True`

read_only [False] If `True`, mount the container's root filesystem as read only

Example: `read_only=True`

restart_policy (or *restart*) Set a restart policy for the container. Must be passed as a string in the format `policy[:retry_count]` where `policy` is one of `always`, `unless-stopped`, or `on-failure`, and `retry_count` is an optional limit to the number of retries. The retry count is ignored when using the `always` or `unless-stopped` restart policy.

Examples:

- `restart_policy=on-failure:5`
- `restart_policy=always`

security_opt Security configuration for MLS systems such as SELinux and AppArmor. Can be passed as a comma-separated list or a Python list.

Examples:

- `security_opt=apparmor:unconfined,param2:value2`
- `security_opt='["apparmor:unconfined", "param2:value2"]'`

Important: Some security options can contain commas. In these cases, this argument *must* be passed as a Python list, as splitting by comma will result in an invalid configuration.

Note: See the documentation for security_opt at https://docs.docker.com/engine/reference/run/#security-configuration

shm_size Size of /dev/shm

Example: `shm_size=128M`

stop_signal The signal used to stop the container. The default is `SIGTERM`.

Example: `stop_signal=SIGRTMIN+3`

stop_timeout Timeout to stop the container, in seconds

Example: `stop_timeout=5`

storage_opt Storage driver options for the container

Examples:

- `storage_opt='dm.basesize=40G'`
- `storage_opt="['dm.basesize=40G']"`
- `storage_opt="{'dm.basesize': '40G'}"`

sysctls (or *sysctl*) Set sysctl options for the container

Examples:

- `sysctl='fs.nr_open=1048576,kernel.pid_max=32768'`
- `sysctls="['fs.nr_open=1048576', 'kernel.pid_max=32768']"`
- `sysctls="{'fs.nr_open': '1048576', 'kernel.pid_max': '32768'}"`

tmpfs A map of container directories which should be replaced by tmpfs mounts, and their corresponding mount options. Can be passed as Python list of PATH:VALUE mappings, or a Python dictionary. However, since commas usually appear in the values, this option *cannot* be passed as a comma-separated list.

Examples:

- `tmpfs="['/run:rw,noexec,nosuid,size=65536k', '/var/lib/mysql:rw, noexec,nosuid,size=600m']"`
- `tmpfs="{'/run': 'rw,noexec,nosuid,size=65536k', '/var/lib/mysql': 'rw,noexec,nosuid,size=600m'}"`

tty [False] Attach TTYs

Example: `tty=True`

ulimits (or *ulimit*) List of ulimits. These limits should be passed in the format `<ulimit_name>:<soft_limit>:<hard_limit>`, with the hard limit being optional. Can be passed as a comma-separated list or a Python list.

Examples:

- `ulimits="nofile=1024:1024,nproc=60"`
- `ulimits="['nofile=1024:1024', 'nproc=60']"`

user User under which to run exec process

Example: `user=foo`

userns_mode (or *user_ns_mode*) Sets the user namsepace mode, when the user namespace remapping option is enabled.

Example: `userns_mode=host`

volumes (or *volume*) List of directories to expose as volumes. Can be passed as a comma-separated list or a Python list.

Examples:

- `volumes=/mnt/vol1,/mnt/vol2`
- `volume="['/mnt/vol1', '/mnt/vol2']"`

volumes_from Container names or IDs from which the container will get volumes. Can be passed as a comma-separated list or a Python list.

Example: `volumes_from=foo`, `volumes_from=foo,bar`, `volumes_from="[foo, bar]"`

volume_driver Sets the container's volume driver

Example: `volume_driver=foobar`

working_dir (or *workdir*) Working directory inside the container

Examples:

- `working_dir=/var/log/nginx`
- `workdir=/var/www/myapp`

RETURN DATA

A dictionary containing the following keys:

- `Id` - ID of the newly-created container

- `Name` - Name of the newly-created container

CLI Example:

```
# Create a data-only container
salt myminion docker.create myuser/mycontainer volumes="/mnt/vol1,/mnt/vol2"
# Create a CentOS 7 container that will stay running once started
salt myminion docker.create centos:7 name=mycent7 interactive=True tty=True
↪command=bash
```

`salt.modules.dockermod.`**`create_network`**(*name*, *skip_translate=None*, *ignore_collisions=False*, *validate_ip_addrs=True*, *client_timeout=60*, ***kwargs*)

Changed in version 2018.3.0: Support added for network configuration options other than `driver` and `driver_opts`, as well as IPAM configuration.

Create a new network

Note: This function supports all arguments for network and IPAM pool configuration which are available for the release of docker-py installed on the minion. For that reason, the arguments described below in the *NETWORK CONFIGURATION ARGUMENTS* and *IP ADDRESS MANAGEMENT (IPAM)* sections may not accurately reflect what is available on the minion. The `docker.get_client_args` function can be used to check the available arguments for the installed version of docker-py (they are found in the `network_config` and `ipam_config` sections of the return data), but Salt will not prevent a user from attempting to use an argument which is unsupported in the release of Docker which is installed. In those cases, network creation be attempted but will fail.

---

name Network name

skip_translate This function translates Salt CLI or SLS input into the format which docker-py expects. However, in the event that Salt's translation logic fails (due to potential changes in the Docker Remote API, or to bugs in the translation code), this argument can be used to exert granular control over which arguments are translated and which are not.

Pass this argument as a comma-separated list (or Python list) of arguments, and translation for each passed argument name will be skipped. Alternatively, pass `True` and *all* translation will be skipped.

Skipping tranlsation allows for arguments to be formatted directly in the format which docker-py expects. This allows for API changes and other issues to be more easily worked around. See the following links for more information:

- docker-py Low-level API
- Docker Engine API

New in version 2018.3.0.

ignore_collisions [False] Since many of docker-py's arguments differ in name from their CLI counterparts (with which most Docker users are more familiar), Salt detects usage of these and aliases them to the docker-py version of that argument. However, if both the alias and the docker-py version of the same argument (e.g. `options` and `driver_opts`) are used, an error will be raised. Set this argument to `True` to suppress these errors and keep the docker-py version of the argument.

New in version 2018.3.0.

validate_ip_addrs [True] For parameters which accept IP addresses as input, IP address validation will be performed. To disable, set this to `False`

Note: When validating subnets, whether or not the IP portion of the subnet is a valid subnet boundary will not be checked. The IP will portion will be validated, and the subnet size will be checked to confirm it is a valid number (1-32 for IPv4, 1-128 for IPv6).

New in version 2018.3.0.

NETWORK CONFIGURATION ARGUMENTS

driver Network driver

Example: `driver=macvlan`

driver_opts (or *driver_opt*, or *options*) Options for the network driver. Either a dictionary of option names and values or a Python list of strings in the format `varname=value`.

Examples:

- `driver_opts='macvlan_mode=bridge,parent=eth0'`
- `driver_opts="['macvlan_mode=bridge', 'parent=eth0']"`
- `driver_opts="{'macvlan_mode': 'bridge', 'parent': 'eth0'}"`

check_duplicate [True] If `True`, checks for networks with duplicate names. Since networks are primarily keyed based on a random ID and not on the name, and network name is strictly a user-friendly alias to the network which is uniquely identified using ID, there is no guaranteed way to check for duplicates. This option providess a best effort, checking for any networks which have the same name, but it is not guaranteed to catch all name collisions.

Example: `check_duplicate=False`

internal [False] If `True`, restricts external access to the network

Example: `internal=True`

labels Add metadata to the network. Labels can be set both with and without values:

Examples (*with* values):

- `labels="label1=value1,label2=value2"`
- `labels="['label1=value1', 'label2=value2']"`
- `labels="{'label1': 'value1', 'label2': 'value2'}"`

Examples (*without* values):

- `labels=label1,label2`
- `labels="['label1', 'label2']"`

enable_ipv6 (or *ipv6*) [False] Enable IPv6 on the network

Example: `enable_ipv6=True`

Note: While it should go without saying, this argument must be set to `True` to *configure an IPv6 subnet*. Also, if this option is turned on without an IPv6 subnet explicitly configured, you will get an error unless you have set up a fixed IPv6 subnet. Consult the Docker IPv6 docs for information on how to do this.

---

attachable [False] If `True`, and the network is in the global scope, non-service containers on worker nodes will be able to connect to the network.

Example: `attachable=True`

Note: While support for this option was added in API version 1.24, its value was not added to the inpsect results until API version 1.26. The version of Docker which is available for CentOS 7 runs API version 1.24, meaning that while Salt can pass this argument to the API, it has no way of knowing the value of this config option in an existing Docker network.

---

scope Specify the network's scope (`local`, `global` or `swarm`)

Example: `scope=local`

ingress [False] If `True`, create an ingress network which provides the routing-mesh in swarm mode

Example: `ingress=True`

IP ADDRESS MANAGEMENT (IPAM)

This function supports networks with either IPv4, or both IPv4 and IPv6. If configuring IPv4, then you can pass the IPAM arguments as shown below, as individual arguments on the Salt CLI. However, if configuring IPv4 and IPv6, the arguments must be passed as a list of dictionaries, in the `ipam_pools` argument. See the **CLI Examples** below. These docs also have more information on these arguments.

IPAM ARGUMENTS

ipam_driver IPAM driver to use, if different from the default one

Example: `ipam_driver=foo`

ipam_opts Options for the IPAM driver. Either a dictionary of option names and values or a Python list of strings in the format `varname=value`.

Examples:

- `ipam_opts='foo=bar,baz=qux'`
- `ipam_opts="['foo=bar', 'baz=quz']"`
- `ipam_opts="{'foo': 'bar', 'baz': 'qux'}"`

IPAM POOL ARGUMENTS

subnet Subnet in CIDR format that represents a network segment

Example: `subnet=192.168.50.0/25`

iprange (or *ip_range*) Allocate container IP from a sub-range within the subnet

Subnet in CIDR format that represents a network segment

Example: `iprange=192.168.50.64/26`

gateway IPv4 gateway for the master subnet

Example: `gateway=192.168.50.1`

aux_addresses (or *aux_address*) A dictionary of mapping container names to IP addresses which should be allocated for them should they connect to the network. Either a dictionary of option names and values or a Python list of strings in the format `host=ipaddr`.

Examples:

- `aux_addresses='foo.bar.tld=192.168.50.10,hello.world.tld=192.168.50.11'`
- `aux_addresses="['foo.bar.tld=192.168.50.10', 'hello.world.tld=192.168.50.11']"`

  - `aux_addresses="{'foo.bar.tld': '192.168.50.10', 'hello.world.tld': '192.168.50.11'}"`

CLI Examples:

```
salt myminion docker.create_network web_network driver=bridge
# IPv4
salt myminion docker.create_network macvlan_network driver=macvlan driver_opts="{
↪'parent':'eth0'}" gateway=172.20.0.1 subnet=172.20.0.0/24
# IPv4 and IPv6
salt myminion docker.create_network mynet ipam_pools='[{"subnet": "10.0.0.0/24",
↪"gateway": "10.0.0.1"}, {"subnet": "fe3f:2180:26:1::60/123", "gateway":
↪"fe3f:2180:26:1::61"}]'
```

`salt.modules.dockermod.`**`create_volume`**`(`*name*, *driver=None*, *driver_opts=None*`)`

Create a new volume

New in version 2015.8.4.

name name of volume

driver Driver of the volume

driver_opts Options for the driver volume

CLI Example:

```
salt myminion docker.create_volume my_volume driver=local
```

`salt.modules.dockermod.`**`dangling`**`(`*prune=False*, *force=False*`)`

Return top-level images (those on which no other images depend) which do not have a tag assigned to them. These include:

- Images which were once tagged but were later untagged, such as those which were superseded by committing a new copy of an existing tagged image.
- Images which were loaded using *`docker.load`* (or the `docker load` Docker CLI command), but not tagged.

prune [False] Remove these images

force [False] If `True`, and if `prune=True`, then forcibly remove these images.

RETURN DATA

If `prune=False`, the return data will be a list of dangling image IDs.

If `prune=True`, the return data will be a dictionary with each key being the ID of the dangling image, and the following information for each image:

- `Comment` - Any error encountered when trying to prune a dangling image

  (Only present if prune failed)
- `Removed` - A boolean (`True` if prune was successful, `False` if not)

CLI Example:

```
salt myminion docker.dangling
salt myminion docker.dangling prune=True
```

`salt.modules.dockermod.`**`depends`**`(`*name*`)`

Returns the containers and images, if any, which depend on the given image

name Name or ID of image

RETURN DATA

A dictionary containing the following keys:

- `Containers` - A list of containers which depend on the specified image
- `Images` - A list of IDs of images which depend on the specified image

CLI Example:

```
salt myminion docker.depends myimage
salt myminion docker.depends 0123456789ab
```

`salt.modules.dockermod.`**`diff`**`(`*name*`)`

Get information on changes made to container's filesystem since it was created. Equivalent to running the `docker diff` Docker CLI command.

name Container name or ID

RETURN DATA

A dictionary containing any of the following keys:

- `Added` - A list of paths that were added.
- `Changed` - A list of paths that were changed.
- `Deleted` - A list of paths that were deleted.

These keys will only be present if there were changes, so if the container has no differences the return dict will be empty.

CLI Example:

```
salt myminion docker.diff mycontainer
```

`salt.modules.dockermod.`**`disconnect_all_containers_from_network`**`(`*network_id*`)`

New in version 2018.3.0.

Runs *`docker.disconnect_container_from_network`* on all containers connected to the specified network, and returns the names of all containers that were disconnected.

network_id Network name or ID

CLI Examples:

```
salt myminion docker.disconnect_all_containers_from_network mynet
salt myminion docker.disconnect_all_containers_from_network⏎
↪1f9d2454d0872b68dd9e8744c6e7a4c66b86f10abaccc21e14f7f014f729b2bc
```

`salt.modules.dockermod.`**`disconnect_container_from_network`**`(`*container, network_id*`)`

New in version 2015.8.3.

Disconnect container from network

container Container name or ID

network_id Network name or ID

CLI Examples:

```
salt myminion docker.disconnect_container_from_network web-1 mynet
salt myminion docker.disconnect_container_from_network web-1⏎
↪1f9d2454d0872b68dd9e8744c6e7a4c66b86f10abaccc21e14f7f014f729b2bc
```

(continues on next page)

(continued from previous page)

```
```

`salt.modules.dockermod.`**`exists`**`(`*name*`)`

Check if a given container exists

name Container name or ID

RETURN DATA

A boolean (`True` if the container exists, otherwise `False`)

CLI Example:

```
salt myminion docker.exists mycontainer
```

`salt.modules.dockermod.`**`export`**`(`*name, path, overwrite=False, makedirs=False, compression=None, **kwargs*`)`

Exports a container to a tar archive. It can also optionally compress that tar archive, and push it up to the Master.

name Container name or ID

path Absolute path on the Minion where the container will be exported

overwrite [False] Unless this option is set to `True`, then if a file exists at the location specified by the `path` argument, an error will be raised.

makedirs [False] If `True`, then if the parent directory of the file specified by the `path` argument does not exist, Salt will attempt to create it.

compression [None] Can be set to any of the following:

- `gzip` or `gz` for gzip compression
- `bzip2` or `bz2` for bzip2 compression
- `xz` or `lzma` for XZ compression (requires xz-utils, as well as the `lzma` module from Python 3.3, available in Python 2 and Python 3.0-3.2 as backports.lzma)

This parameter can be omitted and Salt will attempt to determine the compression type by examining the filename passed in the `path` parameter.

push [False] If `True`, the container will be pushed to the master using *`cp.push`*.

Note: This requires *`file_recv`* to be set to `True` on the Master.

RETURN DATA

A dictionary will containing the following keys:

- `Path` - Path of the file that was exported
- `Push` - Reports whether or not the file was successfully pushed to the Master

  (Only present if push=True)
- `Size` - Size of the file, in bytes
- `Size_Human` - Size of the file, in human-readable units
- `Time_Elapsed` - Time in seconds taken to perform the export

CLI Examples:

```
salt myminion docker.export mycontainer /tmp/mycontainer.tar
salt myminion docker.export mycontainer /tmp/mycontainer.tar.xz push=True
```

`salt.modules.dockermod.`**`get_client_args`**`(`*limit=None*`)`

New in version 2016.3.6,2016.11.4,2017.7.0.

Changed in version 2017.7.0: Replaced the container config args with the ones from the API's `create_container` function.

Changed in version 2018.3.0: Added ability to limit the input to specific client functions

Many functions in Salt have been written to support the full list of arguments for a given function in the docker-py Low-level API. However, depending on the version of docker-py installed on the minion, the available arguments may differ. This function will get the arguments for various functions in the installed version of docker-py, to be used as a reference.

limit An optional list of categories for which to limit the return. This is useful if only a specific set of arguments is desired, and also keeps other function's argspecs from needlessly being examined.

AVAILABLE LIMITS

- `create_container` - arguments accepted by create_container() (used by *`docker.create`*)
- `host_config` - arguments accepted by create_host_config() (used to build the host config for *`docker.create`*)
- `connect_container_to_network` - arguments used by connect_container_to_network() to construct an endpoint config when connecting to a network (used by *`docker.connect_container_to_network`*)
- `create_network` - arguments accepted by create_network() (used by *`docker.create_network`*)
- `ipam_config` - arguments used to create an IPAM pool (used by *`docker.create_network`* in the process of constructing an IPAM config dictionary)

CLI Example:

```
salt myminion docker.get_client_args
salt myminion docker.get_client_args logs
salt myminion docker.get_client_args create_container,connect_container_to_network
```

`salt.modules.dockermod.`**`highstate`**`(`*name, saltenv='base', **kwargs*`)`

Apply a highstate to the running container

New in version 2019.2.0.

The container does not need to have Salt installed, but Python is required.

name Container name or ID

saltenv [base] Specify the environment from which to retrieve the SLS indicated by the *mods* parameter.

CLI Example:

```
salt myminion docker.highstate compassionate_mirzakhani
```

`salt.modules.dockermod.`**`history`**`(`*name, quiet=False*`)`

Return the history for an image. Equivalent to running the `docker history` Docker CLI command.

name Container name or ID

quiet [False] If `True`, the return data will simply be a list of the commands run to build the container.

```
$ salt myminion docker.history nginx:latest quiet=True
myminion:
    - FROM scratch
    - ADD↩
↪file:ef063ed0ae9579362871b9f23d2bc0781ef7cd4de6ac822052cf6c9c5a12b1e2 in /
    - CMD [/bin/bash]
    - MAINTAINER NGINX Docker Maintainers "docker-maint@nginx.com"
    - apt-key adv --keyserver pgp.mit.edu --recv-keys↩
↪573BFD6B3D8FBC641079A6ABABF5BD827BD9BF62
    - echo "deb http://nginx.org/packages/mainline/debian/ wheezy nginx" >> /
↪etc/apt/sources.list
    - ENV NGINX_VERSION=1.7.10-1~wheezy
    - apt-get update &&     apt-get install -y ca-certificates nginx=${NGINX_
↪VERSION} &&     rm -rf /var/lib/apt/lists/*
    - ln -sf /dev/stdout /var/log/nginx/access.log
    - ln -sf /dev/stderr /var/log/nginx/error.log
    - VOLUME [/var/cache/nginx]
    - EXPOSE map[80/tcp:{} 443/tcp:{}]
    - CMD [nginx -g daemon off;]
            https://github.com/saltstack/salt/pull/22421
```

RETURN DATA

If `quiet=False`, the return value will be a list of dictionaries containing information about each step taken to build the image. The keys in each step include the following:

- `Command` - The command executed in this build step
- `Id` - Layer ID
- `Size` - Cumulative image size, in bytes
- `Size_Human` - Cumulative image size, in human-readable units
- `Tags` - Tag(s) assigned to this layer
- `Time_Created_Epoch` - Time this build step was completed (Epoch time)
- `Time_Created_Local` - Time this build step was completed (Minion's local timezone)

CLI Example:

```
salt myminion docker.exists mycontainer
```

`salt.modules.dockermod.`**`images`**(*verbose=False*, ***kwargs*)

Returns information about the Docker images on the Minion. Equivalent to running the `docker images` Docker CLI command.

all [False] If `True`, untagged images will also be returned

verbose [False] If `True`, a `docker inspect` will be run on each image returned.

RETURN DATA

A dictionary with each key being an image ID, and each value some general info about that image (time created, size, tags associated with the image, etc.)

CLI Example:

```
salt myminion docker.images
salt myminion docker.images all=True
```

`salt.modules.dockermod.`**`import_`**`(`*source*, *repository*, *tag='latest'*, *api_response=False*`)`

Changed in version 2018.3.0: The repository and tag must now be passed separately using the `repository` and `tag` arguments, rather than together in the (now deprecated) `image` argument.

Imports content from a local tarball or a URL as a new docker image

source Content to import (URL or absolute path to a tarball). URL can be a file on the Salt fileserver (i.e. `salt://path/to/rootfs/tarball.tar.xz`. To import a file from a saltenv other than `base` (e.g. `dev`), pass it at the end of the URL (ex. `salt://path/to/rootfs/tarball.tar.xz?saltenv=dev`).

repository Repository name for the image being imported

New in version 2018.3.0.

tag [latest] Tag name for the image

New in version 2018.3.0.

image Deprecated since version 2018.3.0: Use both `repository` and `tag` instead

api_response [False] If `True` an `api_response` key will be present in the return data, containing the raw output from the Docker API.

RETURN DATA

A dictionary containing the following keys:

- `Id` - ID of the newly-created image
- `Image` - Name of the newly-created image
- `Time_Elapsed` - Time in seconds taken to perform the commit

CLI Example:

```
salt myminion docker.import /tmp/cent7-minimal.tar.xz myuser/centos
salt myminion docker.import /tmp/cent7-minimal.tar.xz myuser/centos:7
salt myminion docker.import salt://dockerimages/cent7-minimal.tar.xz myuser/
↪centos:7
```

`salt.modules.dockermod.`**`info`**`()`

Returns a dictionary of system-wide information. Equivalent to running the `docker info` Docker CLI command.

CLI Example:

```
salt myminion docker.info
```

`salt.modules.dockermod.`**`inspect`**`(`*name*`)`

Changed in version 2017.7.0: Volumes and networks are now checked, in addition to containers and images.

This is a generic container/image/volume/network inspecton function. It will run the following functions in order:

- *`docker.inspect_container`*
- *`docker.inspect_image`*
- *`docker.inspect_volume`*
- *`docker.inspect_network`*

The first of these to find a match will be returned.

name Container/image/volume/network name or ID

RETURN DATA

A dictionary of container/image/volume/network information

CLI Example:

```
salt myminion docker.inspect mycontainer
salt myminion docker.inspect busybox
```

`salt.modules.dockermod.`**`inspect_container`**`(`*name*`)`

Retrieves container information. Equivalent to running the `docker inspect` Docker CLI command, but will only look for container information.

name Container name or ID

RETURN DATA

A dictionary of container information

CLI Example:

```
salt myminion docker.inspect_container mycontainer
salt myminion docker.inspect_container 0123456789ab
```

`salt.modules.dockermod.`**`inspect_image`**`(`*name*`)`

Retrieves image information. Equivalent to running the `docker inspect` Docker CLI command, but will only look for image information.

Note: To inspect an image, it must have been pulled from a registry or built locally. Images on a Docker registry which have not been pulled cannot be inspected.

name Image name or ID

RETURN DATA

A dictionary of image information

CLI Examples:

```
salt myminion docker.inspect_image busybox
salt myminion docker.inspect_image centos:6
salt myminion docker.inspect_image 0123456789ab
```

`salt.modules.dockermod.`**`inspect_network`**`(`*network_id*`)`

Inspect Network

network_id ID of network

CLI Example:

```
salt myminion docker.inspect_network
↪1f9d2454d0872b68dd9e8744c6e7a4c66b86f10abaccc21e14f7f014f729b2bc
```

`salt.modules.dockermod.`**`inspect_volume`**`(`*name*`)`

Inspect Volume

New in version 2015.8.4.

name Name of volume

CLI Example:

```
salt myminion docker.inspect_volume my_volume
```

`salt.modules.dockermod.`**`kill`**`(`*name*`)`

Kill all processes in a running container instead of performing a graceful shutdown

name Container name or ID

RETURN DATA

A dictionary will be returned, containing the following keys:

- `status` - A dictionary showing the prior state of the container as well as the new state
- `result` - A boolean noting whether or not the action was successful
- `comment` - Only present if the container cannot be killed

CLI Example:

```
salt myminion docker.kill mycontainer
```

`salt.modules.dockermod.`**`layers`**`(`*name*`)`

Returns a list of the IDs of layers belonging to the specified image, with the top-most layer (the one correspnding to the passed name) appearing last.

name Image name or ID

CLI Example:

```
salt myminion docker.layers centos:7
```

`salt.modules.dockermod.`**`list_containers`**`(`***kwargs*`)`

Returns a list of containers by name. This is different from `docker.ps` in that `docker.ps` returns its results organized by container ID.

all [False] If `True`, stopped containers will be included in return data

CLI Example:

```
salt myminion docker.inspect_image <image>
```

`salt.modules.dockermod.`**`list_tags`**`()`

Returns a list of tagged images

CLI Example:

```
salt myminion docker.list_tags
```

`salt.modules.dockermod.`**`load`**`(`*path*, *repository=None*, *tag=None*`)`

Changed in version 2018.3.0: If the loaded image should be tagged, then the repository and tag must now be passed separately using the `repository` and `tag` arguments, rather than together in the (now deprecated) `image` argument.

Load a tar archive that was created using `docker.save` (or via the Docker CLI using `docker save`).

path Path to docker tar archive. Path can be a file on the Minion, or the URL of a file on the Salt fileserver (i.e. `salt://path/to/docker/saved/image.tar`). To load a file from a saltenv other than `base` (e.g. `dev`), pass it at the end of the URL (ex. `salt://path/to/rootfs/tarball.tar.xz?saltenv=dev`).

repository If specified, the topmost layer of the newly-loaded image will be tagged with the specified repo using `docker.tag`. If a repository name is provided, then the `tag` argument is also required.

New in version 2018.3.0.

tag Tag name to go along with the repository name, if the loaded image is to be tagged.

New in version 2018.3.0.

image Deprecated since version 2018.3.0: Use both `repository` and `tag` instead

RETURN DATA

A dictionary will be returned, containing the following keys:

- `Path` - Path of the file that was saved
- `Layers` - A list containing the IDs of the layers which were loaded. Any layers in the file that was loaded, which were already present on the Minion, will not be included.
- `Image` - Name of tag applied to topmost layer

  (Only present if tag was specified and tagging was successful)
- `Time_Elapsed` - Time in seconds taken to load the file
- `Warning` - Message describing any problems encountered in attemp to tag the topmost layer

  (Only present if tag was specified and tagging failed)

CLI Example:

```
salt myminion docker.load /path/to/image.tar
salt myminion docker.load salt://path/to/docker/saved/image.tar repository=myuser/
↪myimage tag=mytag
```

`salt.modules.dockermod.`**`login`**`(`**registries*`)`

New in version 2016.3.7,2016.11.4,2017.7.0.

Performs a `docker login` to authenticate to one or more configured repositories. See the documentation at the top of this page to configure authentication credentials.

Multiple registry URLs (matching those configured in Pillar) can be passed, and Salt will attempt to login to *just* those registries. If no registry URLs are provided, Salt will attempt to login to *all* configured registries.

RETURN DATA

A dictionary containing the following keys:

- `Results` - A dictionary mapping registry URLs to the authentication result. `True` means a successful login, `False` means a failed login.
- `Errors` - A list of errors encountered during the course of this function.

CLI Example:

```
salt myminion docker.login
salt myminion docker.login hub
salt myminion docker.login hub https://mydomain.tld/registry/
```

`salt.modules.dockermod.`**`logs`**`(`*name*, ***kwargs*`)`

Changed in version 2018.3.0: Support for all of docker-py's logs() function's arguments, with the exception of `stream`.

Returns the logs for the container. An interface to docker-py's logs() function.

name Container name or ID

stdout [True] Return stdout lines

stderr [True] Return stdout lines

timestamps [False] Show timestamps

tail [all] Output specified number of lines at the end of logs. Either an integer number of lines or the string `all`.

since Show logs since the specified time, passed as a UNIX epoch timestamp. Optionally, if timelib is installed on the minion the timestamp can be passed as a string which will be resolved to a date using `timelib.strtodatetime()`.

follow [False] If `True`, this function will block until the container exits and return the logs when it does. The default behavior is to return what is in the log at the time this function is executed.

CLI Examples:

```
# All logs
salt myminion docker.logs mycontainer
# Last 100 lines of log
salt myminion docker.logs mycontainer tail=100
# Just stderr
salt myminion docker.logs mycontainer stdout=False
# Logs since a specific UNIX timestamp
salt myminion docker.logs mycontainer since=1511688459
# Flexible format for "since" argument (requires timelib)
salt myminion docker.logs mycontainer since='1 hour ago'
salt myminion docker.logs mycontainer since='1 week ago'
salt myminion docker.logs mycontainer since='1 fortnight ago'
```

`salt.modules.dockermod.`**`networks`**`(`*names=None, ids=None*`)`

Changed in version 2017.7.0: The `names` and `ids` can be passed as a comma-separated list now, as well as a Python list.

Changed in version 2018.3.0: The `Containers` key for each network is no longer always empty.

List existing networks

names Filter by name

ids Filter by id

CLI Example:

```
salt myminion docker.networks names=network-web
salt myminion docker.networks⏎
↪ids=1f9d2454d0872b68dd9e8744c6e7a4c66b86f10abaccc21e14f7f014f729b2bc
```

`salt.modules.dockermod.`**`pause`**`(`*name*`)`

Pauses a container

name Container name or ID

RETURN DATA

A dictionary will be returned, containing the following keys:

- `status` - A dictionary showing the prior state of the container as well as the new state
- `result` - A boolean noting whether or not the action was successful
- `comment` - Only present if the container cannot be paused

CLI Example:

```
salt myminion docker.pause mycontainer
```

`salt.modules.dockermod.`**`pid`**(*name*)

Returns the PID of a container

name Container name or ID

CLI Example:

```
salt myminion docker.pid mycontainer
salt myminion docker.pid 0123456789ab
```

`salt.modules.dockermod.`**`port`**(*name*, *private_port=None*)

Returns port mapping information for a given container. Equivalent to running the `docker port` Docker CLI command.

name Container name or ID

> Changed in version 2019.2.0: This value can now be a pattern expression (using the pattern-matching characters defined in fnmatch). If a pattern expression is used, this function will return a dictionary mapping container names which match the pattern to the mappings for those containers. When no pattern expression is used, a dictionary of the mappings for the specified container name will be returned.

private_port [None] If specified, get information for that specific port. Can be specified either as a port number (i.e. `5000`), or as a port number plus the protocol (i.e. `5000/udp`).

> If this argument is omitted, all port mappings will be returned.

RETURN DATA

A dictionary of port mappings, with the keys being the port and the values being the mapping(s) for that port.

CLI Examples:

```
salt myminion docker.port mycontainer
salt myminion docker.port mycontainer 5000
salt myminion docker.port mycontainer 5000/udp
```

`salt.modules.dockermod.`**`prune`**(*containers=False*, *networks=False*, *images=False*, *build=False*, *volumes=False*, *system=None*, ***filters*)

New in version 2019.2.0.

Prune Docker's various subsystems

Note: This requires docker-py version 2.1.0 or later.

containers [False] If `True`, prunes stopped containers (documentation)

images [False] If `True`, prunes unused images (documentation)

networks [False] If `False`, prunes unreferenced networks (documentation)

build [False] If `True`, clears the builder cache

> **Note:** Only supported in Docker 17.07.x and newer. Additionally, filters do not apply to this argument.

volumes [False] If `True`, prunes unreferenced volumes (documentation)

system If `True`, prunes containers, images, networks, and builder cache. Assumed to be `True` if none of `containers`, `images`, `networks`, or `build` are set to `True`.

Note: `volumes=True` must still be used to prune volumes

---

filters

- `dangling=True` (images only) - remove only dangling images
- `until=<timestamp>` - only remove objects created before given timestamp. Not applicable to volumes. See the documentation links above for examples of valid time expressions.
- `label` - only remove objects matching the label expression. Valid expressions include `labelname` or `labelname=value`.

CLI Examples:

```
salt myminion docker.prune system=True
salt myminion docker.prune system=True until=12h
salt myminion docker.prune images=True dangling=True
salt myminion docker.prune images=True label=foo,bar=baz
```

`salt.modules.dockermod.`**`ps_`**(*filters=None*, ***kwargs*)

Returns information about the Docker containers on the Minion. Equivalent to running the `docker ps` Docker CLI command.

all [False] If `True`, stopped containers will also be returned

host: False If `True`, local host's network topology will be included

verbose [False] If `True`, a `docker inspect` will be run on each container returned.

filters: None A dictionary of filters to be processed on the container list. Available filters:

- exited (int): Only containers with specified exit code
- status (str): One of restarting, running, paused, exited
- label (str): format either "key" or "key=value"

RETURN DATA

A dictionary with each key being an container ID, and each value some general info about that container (time created, name, command, etc.)

CLI Example:

```
salt myminion docker.ps
salt myminion docker.ps all=True
salt myminion docker.ps filters="{'label': 'role=web'}"
```

`salt.modules.dockermod.`**`pull`**(*image*, *insecure_registry=False*, *api_response=False*, *client_timeout=60*)

Changed in version 2018.3.0: If no tag is specified in the `image` argument, all tags for the image will be pulled. For this reason is it recommended to pass `image` using the `repo:tag` notation.

Pulls an image from a Docker registry

image Image to be pulled

insecure_registry [False] If `True`, the Docker client will permit the use of insecure (non-HTTPS) registries.

api_response [False] If `True`, an `API_Response` key will be present in the return data, containing the raw output from the Docker API.

Note: This may result in a **lot** of additional return data, especially for larger images.

---

client_timeout Timeout in seconds for the Docker client. This is not a timeout for this function, but for receiving a response from the API.

RETURN DATA

A dictionary will be returned, containing the following keys:

- **`Layers`** - **A dictionary containing one or more of the following keys**:
  - `Already_Pulled` - Layers that that were already present on the Minion
  - `Pulled` - Layers that that were pulled
- `Status` - A string containing a summary of the pull action (usually a message saying that an image was downloaded, or that it was up to date).
- `Time_Elapsed` - Time in seconds taken to perform the pull

CLI Example:

```
salt myminion docker.pull centos
salt myminion docker.pull centos:6
```

`salt.modules.dockermod.`**`push`**`(`*image*, *insecure_registry=False*, *api_response=False*, *client_timeout=60*`)`

Changed in version 2015.8.4: The `Id` and `Image` keys are no longer present in the return data. This is due to changes in the Docker Remote API.

Pushes an image to a Docker registry. See the documentation at top of this page to configure authentication credentials.

image Image to be pushed. If just the repository name is passed, then all tagged images for the specified repo will be pushed. If the image name is passed in `repo:tag` notation, only the specified image will be pushed.

insecure_registry [False] If `True`, the Docker client will permit the use of insecure (non-HTTPS) registries.

api_response [False] If `True`, an `API_Response` key will be present in the return data, containing the raw output from the Docker API.

client_timeout Timeout in seconds for the Docker client. This is not a timeout for this function, but for receiving a response from the API.

RETURN DATA

A dictionary will be returned, containing the following keys:

- **`Layers`** - **A dictionary containing one or more of the following keys**:
  - `Already_Pushed` - Layers that that were already present on the Minion
  - `Pushed` - Layers that that were pushed
- `Time_Elapsed` - Time in seconds taken to perform the push

CLI Example:

```
salt myminion docker.push myuser/mycontainer
salt myminion docker.push myuser/mycontainer:mytag
```

`salt.modules.dockermod.`**`remove_network`**`(`*network_id*`)`

Remove a network

network_id Network name or ID

CLI Examples:

```
salt myminion docker.remove_network mynet
salt myminion docker.remove_network␣
↪1f9d2454d0872b68dd9e8744c6e7a4c66b86f10abaccc21e14f7f014f729b2bc
```

`salt.modules.dockermod.`**`remove_volume`**`(`*name*`)`

Remove a volume

New in version 2015.8.4.

name Name of volume

CLI Example:

```
salt myminion docker.remove_volume my_volume
```

`salt.modules.dockermod.`**`rename`**`(`*name, new_name*`)`

New in version 2017.7.0.

Renames a container. Returns `True` if successful, and raises an error if the API returns one. If unsuccessful and the API returns no error (should not happen), then `False` will be returned.

name Name or ID of existing container

new_name New name to assign to container

CLI Example:

```
salt myminion docker.rename foo bar
```

`salt.modules.dockermod.`**`resolve_image_id`**`(`*name*`)`

New in version 2018.3.0.

Given an image name (or partial image ID), return the full image ID. If no match is found among the locally-pulled images, then `False` will be returned.

CLI Examples:

```
salt myminion docker.resolve_image_id foo
salt myminion docker.resolve_image_id foo:bar
salt myminion docker.resolve_image_id 36540f359ca3
```

`salt.modules.dockermod.`**`resolve_tag`**`(`*name, **kwargs*`)`

New in version 2017.7.2.

Changed in version 2018.3.0: Instead of matching against pulled tags using *`docker.list_tags`*, this function now simply inspects the passed image name using *`docker.inspect_image`* and returns the first matching tag. If no matching tags are found, it is assumed that the passed image is an untagged image ID, and the full ID is returned.

Inspects the specified image name and returns the first matching tag in the inspect results. If the specified image is not pulled locally, this function will return `False`.

name Image name to resolve. If the image is found but there are no tags, this means that the image name passed was an untagged image. In this case the image ID will be returned.

all [False] If `True`, a list of all matching tags will be returned. If the image is found but there are no tags, then a list will still be returned, but it will simply contain the image ID.

New in version 2018.3.0.

tags Deprecated since version 2018.3.0.

CLI Examples:

```
salt myminion docker.resolve_tag busybox
salt myminion docker.resolve_tag centos:7 all=True
salt myminion docker.resolve_tag c9f378ac27d9
```

`salt.modules.dockermod.`**`restart`**`(`*name, timeout=10*`)`

Restarts a container

name Container name or ID

timeout [10] Timeout in seconds after which the container will be killed (if it has not yet gracefully shut down)

RETURN DATA

A dictionary will be returned, containing the following keys:

- `status` - A dictionary showing the prior state of the container as well as the new state
- `result` - A boolean noting whether or not the action was successful
- `restarted` - If restart was successful, this key will be present and will be set to `True`.

CLI Examples:

```
salt myminion docker.restart mycontainer
salt myminion docker.restart mycontainer timeout=20
```

`salt.modules.dockermod.`**`retcode`**`(`*name, cmd, exec_driver=None, stdin=None, python_shell=True, output_loglevel='debug', use_vt=False, ignore_retcode=False, keep_env=None*`)`

Run *`cmd.retcode`* within a container

name Container name or ID in which to run the command

cmd Command to run

exec_driver [None] If not passed, the execution driver will be detected as described *above*.

stdin [None] Standard input to be used for the command

output_loglevel [debug] Level at which to log the output from the command. Set to `quiet` to suppress logging.

use_vt [False] Use SaltStack's utils.vt to stream output to console.

keep_env [None] If not passed, only a sane default PATH environment variable will be set. If `True`, all environment variables from the container's host will be kept. Otherwise, a comma-separated list (or Python list) of environment variable names can be passed, and those environment variables will be kept.

CLI Example:

```
salt myminion docker.retcode mycontainer 'ls -l /etc'
```

`salt.modules.dockermod.`**`rm_`**`(`*name*, *force=False*, *volumes=False*, ***kwargs*`)`

Removes a container

name Container name or ID

force [False] If `True`, the container will be killed first before removal, as the Docker API will not permit a running container to be removed. This option is set to `False` by default to prevent accidental removal of a running container.

stop [False] If `True`, the container will be stopped first before removal, as the Docker API will not permit a running container to be removed. This option is set to `False` by default to prevent accidental removal of a running container.

New in version 2017.7.0.

timeout Optional timeout to be passed to *`docker.stop`* if stopping the container.

New in version 2018.3.0.

volumes [False] Also remove volumes associated with container

RETURN DATA

A list of the IDs of containers which were removed

CLI Example:

```
salt myminion docker.rm mycontainer
salt myminion docker.rm mycontainer force=True
```

`salt.modules.dockermod.`**`rmi`**`(`**names*, ***kwargs*`)`

Removes an image

name Name (in `repo:tag` notation) or ID of image.

force [False] If `True`, the image will be removed even if the Minion has containers created from that image

prune [True] If `True`, untagged parent image layers will be removed as well, set this to `False` to keep them.

RETURN DATA

A dictionary will be returned, containing the following two keys:

- `Layers` - A list of the IDs of image layers that were removed
- `Tags` - A list of the tags that were removed
- `Errors` - A list of any errors that were encountered

CLI Examples:

```
salt myminion docker.rmi busybox
salt myminion docker.rmi busybox force=True
salt myminion docker.rmi foo bar baz
```

`salt.modules.dockermod.`**`run`**`(`*name*, *cmd*, *exec_driver=None*, *stdin=None*, *python_shell=True*, *output_loglevel='debug'*, *use_vt=False*, *ignore_retcode=False*, *keep_env=None*`)`

Run *`cmd.run`* within a container

name Container name or ID in which to run the command

cmd Command to run

exec_driver [None] If not passed, the execution driver will be detected as described *above*.

stdin [None] Standard input to be used for the command

output_loglevel [debug] Level at which to log the output from the command. Set to `quiet` to suppress logging.

use_vt [False] Use SaltStack's utils.vt to stream output to console.

keep_env [None] If not passed, only a sane default PATH environment variable will be set. If `True`, all environment variables from the container's host will be kept. Otherwise, a comma-separated list (or Python list) of environment variable names can be passed, and those environment variables will be kept.

CLI Example:

```
salt myminion docker.run mycontainer 'ls -l /etc'
```

`salt.modules.dockermod.`**`run_all`**`(`*name, cmd, exec_driver=None, stdin=None, python_shell=True, output_loglevel='debug', use_vt=False, ignore_retcode=False, keep_env=None*`)`

Run `cmd.run_all` within a container

Note: While the command is run within the container, it is initiated from the host. Therefore, the PID in the return dict is from the host, not from the container.

name Container name or ID in which to run the command

cmd Command to run

exec_driver [None] If not passed, the execution driver will be detected as described *above*.

stdin [None] Standard input to be used for the command

output_loglevel [debug] Level at which to log the output from the command. Set to `quiet` to suppress logging.

use_vt [False] Use SaltStack's utils.vt to stream output to console.

keep_env [None] If not passed, only a sane default PATH environment variable will be set. If `True`, all environment variables from the container's host will be kept. Otherwise, a comma-separated list (or Python list) of environment variable names can be passed, and those environment variables will be kept.

CLI Example:

```
salt myminion docker.run_all mycontainer 'ls -l /etc'
```

`salt.modules.dockermod.`**`run_container`**`(`*image, name=None, skip_translate=None, ignore_collisions=False, validate_ip_addrs=True, client_timeout=60, bg=False, replace=False, force=False, networks=None, **kwargs*`)`

New in version 2018.3.0.

Equivalent to `docker run` on the Docker CLI. Runs the container, waits for it to exit, and returns the container's logs when complete.

Note: Not to be confused with `docker.run`, which provides a `cmd.run`-like interface for executing commands in a running container.

This function accepts the same arguments as `docker.create`, with the exception of `start`. In addition, it accepts the arguments from `docker.logs`, with the exception of `follow`, to control how logs are returned.

Finally, the `bg` argument described below can be used to optionally run the container in the background (the default behavior is to block until the container exits).

bg [False] If `True`, this function will not wait for the container to exit and will not return its logs. It will however return the container's name and ID, allowing for `docker.logs` to be used to view the logs.

> **Note:** The logs will be inaccessible once the container exits if `auto_remove` is set to `True`, so keep this in mind.

replace [False] If `True`, and if the named container already exists, this will remove the existing container. The default behavior is to return a `False` result when the container already exists.

force [False] If `True`, and the named container already exists, *and* `replace` is also set to `True`, then the container will be forcibly removed. Otherwise, the state will not proceed and will return a `False` result.

networks Networks to which the container should be connected. If automatic IP configuration is being used, the networks can be a simple list of network names. If custom IP configuration is being used, then this argument must be passed as a dictionary.

CLI Examples:

```
salt myminion docker.run_container myuser/myimage command=/usr/local/bin/myscript.
↪sh
# Run container in the background
salt myminion docker.run_container myuser/myimage command=/usr/local/bin/myscript.
↪sh bg=True
# Connecting to two networks using automatic IP configuration
salt myminion docker.run_container myuser/myimage command='perl /scripts/sync.py'
↪networks=net1,net2
# net1 using automatic IP, net2 using static IPv4 address
salt myminion docker.run_container myuser/myimage command='perl /scripts/sync.py'
↪networks='{"net1": {}, "net2": {"ipv4_address": "192.168.27.12"}}'
```

`salt.modules.dockermod.`**`run_stderr`**(*name*, *cmd*, *exec_driver=None*, *stdin=None*, *python_shell=True*, *output_loglevel='debug'*, *use_vt=False*, *ignore_retcode=False*, *keep_env=None*)

Run `cmd.run_stderr` within a container

name Container name or ID in which to run the command

cmd Command to run

exec_driver [None] If not passed, the execution driver will be detected as described *above*.

stdin [None] Standard input to be used for the command

output_loglevel [debug] Level at which to log the output from the command. Set to `quiet` to suppress logging.

use_vt [False] Use SaltStack's utils.vt to stream output to console.

keep_env [None] If not passed, only a sane default PATH environment variable will be set. If `True`, all environment variables from the container's host will be kept. Otherwise, a comma-separated list (or Python list) of environment variable names can be passed, and those environment variables will be kept.

CLI Example:

```
salt myminion docker.run_stderr mycontainer 'ls -l /etc'
```

`salt.modules.dockermod.`**`run_stdout`**`(`*name, cmd, exec_driver=None, stdin=None, python_shell=True, output_loglevel='debug', use_vt=False, ignore_retcode=False, keep_env=None*`)`

Run *`cmd.run_stdout`* within a container

name Container name or ID in which to run the command

cmd Command to run

exec_driver [None] If not passed, the execution driver will be detected as described *above*.

stdin [None] Standard input to be used for the command

output_loglevel [debug] Level at which to log the output from the command. Set to `quiet` to suppress logging.

use_vt [False] Use SaltStack's utils.vt to stream output to console.

keep_env [None] If not passed, only a sane default PATH environment variable will be set. If `True`, all environment variables from the container's host will be kept. Otherwise, a comma-separated list (or Python list) of environment variable names can be passed, and those environment variables will be kept.

CLI Example:

```
salt myminion docker.run_stdout mycontainer 'ls -l /etc'
```

`salt.modules.dockermod.`**`save`**`(`*name, path, overwrite=False, makedirs=False, compression=None, **kwargs*`)`

Saves an image and to a file on the minion. Equivalent to running the `docker save` Docker CLI command, but unlike `docker save` this will also work on named images instead of just images IDs.

name Name or ID of image. Specify a specific tag by using the `repo:tag` notation.

path Absolute path on the Minion where the image will be exported

overwrite [False] Unless this option is set to `True`, then if the destination file exists an error will be raised.

makedirs [False] If `True`, then if the parent directory of the file specified by the `path` argument does not exist, Salt will attempt to create it.

compression [None] Can be set to any of the following:

- `gzip` or `gz` for gzip compression
- `bzip2` or `bz2` for bzip2 compression
- `xz` or `lzma` for XZ compression (requires xz-utils, as well as the `lzma` module from Python 3.3, available in Python 2 and Python 3.0-3.2 as backports.lzma)

This parameter can be omitted and Salt will attempt to determine the compression type by examining the filename passed in the `path` parameter.

Note: Since the Docker API does not support `docker save`, compression will be a bit slower with this function than with *`docker.export`* since the image(s) will first be saved and then the compression done afterwards.

push [False] If `True`, the container will be pushed to the master using *`cp.push`*.

Note: This requires *`file_recv`* to be set to `True` on the Master.

RETURN DATA

A dictionary will be returned, containing the following keys:

- `Path` - Path of the file that was saved
- `Push` - Reports whether or not the file was successfully pushed to the Master

  (Only present if push=True)
- `Size` - Size of the file, in bytes
- `Size_Human` - Size of the file, in human-readable units
- `Time_Elapsed` - Time in seconds taken to perform the save

CLI Examples:

```
salt myminion docker.save centos:7 /tmp/cent7.tar
salt myminion docker.save 0123456789ab cdef01234567 /tmp/saved.tar
```

`salt.modules.dockermod.`**`script`**(*name, source, saltenv='base', args=None, template=None, exec_driver=None, stdin=None, python_shell=True, output_loglevel='debug', ignore_retcode=False, use_vt=False, keep_env=None*)

Run *`cmd.script`* within a container

Note: While the command is run within the container, it is initiated from the host. Therefore, the PID in the return dict is from the host, not from the container.

---

name Container name or ID

source Path to the script. Can be a local path on the Minion or a remote file from the Salt fileserver.

args A string containing additional command-line options to pass to the script.

template [None] Templating engine to use on the script before running.

exec_driver [None] If not passed, the execution driver will be detected as described *above*.

stdin [None] Standard input to be used for the script

output_loglevel [debug] Level at which to log the output from the script. Set to `quiet` to suppress logging.

use_vt [False] Use SaltStack's utils.vt to stream output to console.

keep_env [None] If not passed, only a sane default PATH environment variable will be set. If `True`, all environment variables from the container's host will be kept. Otherwise, a comma-separated list (or Python list) of environment variable names can be passed, and those environment variables will be kept.

CLI Example:

```
salt myminion docker.script mycontainer salt://docker_script.py
salt myminion docker.script mycontainer salt://scripts/runme.sh 'arg1 arg2 "arg 3"
↪'
salt myminion docker.script mycontainer salt://scripts/runme.sh stdin=
↪'one\ntwo\nthree\nfour\nfive\n' output_loglevel=quiet
```

`salt.modules.dockermod.`**`script_retcode`**(*name, source, saltenv='base', args=None, template=None, exec_driver=None, stdin=None, python_shell=True, output_loglevel='debug', ignore_retcode=False, use_vt=False, keep_env=None*)

Run *`cmd.script_retcode`* within a container

name Container name or ID

source Path to the script. Can be a local path on the Minion or a remote file from the Salt fileserver.

args A string containing additional command-line options to pass to the script.

template [None] Templating engine to use on the script before running.

exec_driver [None] If not passed, the execution driver will be detected as described *above*.

stdin [None] Standard input to be used for the script

output_loglevel [debug] Level at which to log the output from the script. Set to `quiet` to suppress logging.

use_vt [False] Use SaltStack's utils.vt to stream output to console.

keep_env [None] If not passed, only a sane default PATH environment variable will be set. If `True`, all environment variables from the container's host will be kept. Otherwise, a comma-separated list (or Python list) of environment variable names can be passed, and those environment variables will be kept.

CLI Example:

```
salt myminion docker.script_retcode mycontainer salt://docker_script.py
salt myminion docker.script_retcode mycontainer salt://scripts/runme.sh 'arg1
↪arg2 "arg 3"'
salt myminion docker.script_retcode mycontainer salt://scripts/runme.sh stdin=
↪'one\ntwo\nthree\nfour\nfive\n' output_loglevel=quiet
```

`salt.modules.dockermod.`**`search`**(*name, official=False, trusted=False*)

Searches the registry for an image

name Search keyword

official [False] Limit results to official builds

trusted [False] Limit results to trusted builds

RETURN DATA

A dictionary with each key being the name of an image, and the following information for each image:

- `Description` - Image description
- `Official` - A boolean (`True` if an official build, `False` if not)
- `Stars` - Number of stars the image has on the registry
- `Trusted` - A boolean (`True` if a trusted build, `False` if not)

CLI Example:

```
salt myminion docker.search centos
salt myminion docker.search centos official=True
```

`salt.modules.dockermod.`**`signal_`**(*name, signal*)

Send a signal to a container. Signals can be either strings or numbers, and are defined in the **Standard Signals** section of the `signal(7)` manpage. Run `man 7 signal` on a Linux host to browse this manpage.

name Container name or ID

signal Signal to send to container

RETURN DATA

If the signal was successfully sent, `True` will be returned. Otherwise, an error will be raised.

CLI Example:

```
salt myminion docker.signal mycontainer SIGHUP
```

`salt.modules.dockermod.`**`sls`**(*name*, *mods=None*, ***kwargs*)

Apply the states defined by the specified SLS modules to the running container

New in version 2016.11.0.

The container does not need to have Salt installed, but Python is required.

name Container name or ID

mods [None] A string containing comma-separated list of SLS with defined states to apply to the container.

saltenv [base] Specify the environment from which to retrieve the SLS indicated by the *mods* parameter.

pillarenv Specify a Pillar environment to be used when applying states. This can also be set in the minion config file using the `pillarenv` option. When neither the `pillarenv` minion config option nor this CLI argument is used, all Pillar environments will be merged together.

New in version 2018.3.0.

pillar Custom Pillar values, passed as a dictionary of key-value pairs

Note: Values passed this way will override Pillar values set via `pillar_roots` or an external Pillar source.

New in version 2018.3.0.

CLI Example:

```
salt myminion docker.sls compassionate_mirzakhani mods=rails,web
```

`salt.modules.dockermod.`**`sls_build`**(*repository*, *tag='latest'*, *base='opensuse/python'*, *mods=None*, *dryrun=False*, ***kwargs*)

Changed in version 2018.3.0: The repository and tag must now be passed separately using the `repository` and `tag` arguments, rather than together in the (now deprecated) `image` argument.

Build a Docker image using the specified SLS modules on top of base image

New in version 2016.11.0.

The base image does not need to have Salt installed, but Python is required.

repository Repository name for the image to be built

New in version 2018.3.0.

tag [latest] Tag name for the image to be built

New in version 2018.3.0.

name Deprecated since version 2018.3.0: Use both `repository` and `tag` instead

base [opensuse/python] Name or ID of the base image

mods A string containing comma-separated list of SLS with defined states to apply to the base image.

saltenv [base] Specify the environment from which to retrieve the SLS indicated by the *mods* parameter.

pillarenv Specify a Pillar environment to be used when applying states. This can also be set in the minion config file using the `pillarenv` option. When neither the `pillarenv` minion config option nor this CLI argument is used, all Pillar environments will be merged together.

New in version 2018.3.0.

pillar Custom Pillar values, passed as a dictionary of key-value pairs

Note: Values passed this way will override Pillar values set via `pillar_roots` or an external Pillar source.

New in version 2018.3.0.

dryrun: False when set to True the container will not be committed at the end of the build. The dryrun succeed also when the state contains errors.

RETURN DATA

A dictionary with the ID of the new container. In case of a dryrun, the state result is returned and the container gets removed.

CLI Example:

```
salt myminion docker.sls_build imgname base=mybase mods=rails,web
```

`salt.modules.dockermod.`**`start_`**`(`*name*`)`

Start a container

name Container name or ID

RETURN DATA

A dictionary will be returned, containing the following keys:

- `status` - A dictionary showing the prior state of the container as well as the new state
- `result` - A boolean noting whether or not the action was successful
- `comment` - Only present if the container cannot be started

CLI Example:

```
salt myminion docker.start mycontainer
```

`salt.modules.dockermod.`**`state`**`(`*name*`)`

Returns the state of the container

name Container name or ID

RETURN DATA

A string representing the current state of the container (either `running`, `paused`, or `stopped`)

CLI Example:

```
salt myminion docker.state mycontainer
```

`salt.modules.dockermod.`**`stop`**`(`*name, timeout=None, **kwargs*`)`

Stops a running container

name Container name or ID

unpause [False] If `True` and the container is paused, it will be unpaused before attempting to stop the container.

timeout Timeout in seconds after which the container will be killed (if it has not yet gracefully shut down)

Changed in version 2017.7.0: If this argument is not passed, then the container's configuration will be checked. If the container was created using the `stop_timeout` argument, then the configured timeout will be used, otherwise the timeout will be 10 seconds.

RETURN DATA

A dictionary will be returned, containing the following keys:

- `status` - A dictionary showing the prior state of the container as well as the new state
- `result` - A boolean noting whether or not the action was successful
- `comment` - Only present if the container can not be stopped

CLI Examples:

```
salt myminion docker.stop mycontainer
salt myminion docker.stop mycontainer unpause=True
salt myminion docker.stop mycontainer timeout=20
```

`salt.modules.dockermod.`**`tag_`**`(`*name*, *repository*, *tag='latest'*, *force=False*`)`

Changed in version 2018.3.0: The repository and tag must now be passed separately using the `repository` and `tag` arguments, rather than together in the (now deprecated) `image` argument.

Tag an image into a repository and return `True`. If the tag was unsuccessful, an error will be raised.

name ID of image

repository Repository name for the image to be built

New in version 2018.3.0.

tag [latest] Tag name for the image to be built

New in version 2018.3.0.

image Deprecated since version 2018.3.0: Use both `repository` and `tag` instead

force [False] Force apply tag

CLI Example:

```
salt myminion docker.tag 0123456789ab myrepo/mycontainer mytag
```

`salt.modules.dockermod.`**`top`**`(`*name*`)`

Runs the *docker top* command on a specific container

name Container name or ID

CLI Example:

RETURN DATA

A list of dictionaries containing information about each process

```
salt myminion docker.top mycontainer
salt myminion docker.top 0123456789ab
```

`salt.modules.dockermod.`**`unpause`**`(`*name*`)`

Unpauses a container

name Container name or ID

RETURN DATA

A dictionary will be returned, containing the following keys:

- `status` - A dictionary showing the prior state of the container as well as the new state
- `result` - A boolean noting whether or not the action was successful
- `comment` - Only present if the container can not be unpaused

CLI Example:

```
salt myminion docker.pause mycontainer
```

`salt.modules.dockermod.`**`version`**`()`

Returns a dictionary of Docker version information. Equivalent to running the `docker version` Docker CLI command.

CLI Example:

```
salt myminion docker.version
```

`salt.modules.dockermod.`**`volumes`**`(`*filters=None*`)`

List existing volumes

New in version 2015.8.4.

filters There is one available filter: dangling=true

CLI Example:

```
salt myminion docker.volumes filters="{'dangling': True}"
```

`salt.modules.dockermod.`**`wait`**`(`*name, ignore_already_stopped=False, fail_on_exit_status=False*`)`

Wait for the container to exit gracefully, and return its exit code

Note: This function will block until the container is stopped.

name Container name or ID

ignore_already_stopped Boolean flag that prevents execution to fail, if a container is already stopped.

fail_on_exit_status Boolean flag to report execution as failure if `exit_status` is different than 0.

RETURN DATA

A dictionary will be returned, containing the following keys:

- `status` - A dictionary showing the prior state of the container as well as the new state
- `result` - A boolean noting whether or not the action was successful
- `exit_status` - Exit status for the container
- `comment` - Only present if the container is already stopped

CLI Example:

```
salt myminion docker.wait mycontainer
```

21.9.111 salt.modules.dpkg_lowpkg

Support for DEB packages

`salt.modules.dpkg_lowpkg.`**`bin_pkg_info`**`(`*path, saltenv='base'*`)`

New in version 2015.8.0.

Parses RPM metadata and returns a dictionary of information about the package (name, version, etc.).

path Path to the file. Can either be an absolute path to a file on the minion, or a salt fileserver URL (e.g. `salt://path/to/file.rpm`). If a salt fileserver URL is passed, the file will be cached to the minion so that it can be examined.

saltenv [base] Salt fileserver envrionment from which to retrieve the package. Ignored if `path` is a local file path on the minion.

CLI Example:

```
salt '*' lowpkg.bin_pkg_info /root/foo-1.2.3-1ubuntu1_all.deb
salt '*' lowpkg.bin_pkg_info salt://foo-1.2.3-1ubuntu1_all.deb
```

`salt.modules.dpkg_lowpkg.`**`file_dict`**`(`**packages*`)`

List the files that belong to a package, grouped by package. Not specifying any packages will return a list of _every_ file on the system's package database (not generally recommended).

CLI Examples:

```
salt '*' lowpkg.file_list httpd
salt '*' lowpkg.file_list httpd postfix
salt '*' lowpkg.file_list
```

`salt.modules.dpkg_lowpkg.`**`file_list`**`(`**packages*`)`

List the files that belong to a package. Not specifying any packages will return a list of _every_ file on the system's package database (not generally recommended).

CLI Examples:

```
salt '*' lowpkg.file_list httpd
salt '*' lowpkg.file_list httpd postfix
salt '*' lowpkg.file_list
```

`salt.modules.dpkg_lowpkg.`**`info`**`(`**packages, **kwargs*`)`

Returns a detailed summary of package information for provided package names. If no packages are specified, all packages will be returned.

New in version 2015.8.1.

packages The names of the packages for which to return information.

failhard Whether to throw an exception if none of the packages are installed. Defaults to True.

New in version 2016.11.3.

CLI example:

```
salt '*' lowpkg.info
salt '*' lowpkg.info apache2 bash
salt '*' lowpkg.info 'php5*' failhard=false
```

`salt.modules.dpkg_lowpkg.`**`list_pkgs`**`(`**packages*`)`

List the packages currently installed in a dict:

```
{'<package_name>': '<version>'}
```

External dependencies:

```
Virtual package resolution requires aptitude. Because this function
uses dpkg, virtual packages will be reported as not installed.
```

CLI Example:

```
salt '*' lowpkg.list_pkgs
salt '*' lowpkg.list_pkgs httpd
```

`salt.modules.dpkg_lowpkg.`**`unpurge`**(**packages*)

Change package selection for each package specified to 'install'

CLI Example:

```
salt '*' lowpkg.unpurge curl
```

21.9.112 salt.modules.drac

Manage Dell DRAC

`salt.modules.drac.`**`change_password`**(*username, password, uid=None*)

Change users password

CLI Example:

```
salt dell drac.change_password [USERNAME] [PASSWORD] [UID - optional]
salt dell drac.change_password diana secret
```

`salt.modules.drac.`**`create_user`**(*username, password, permissions, users=None*)

Create user accounts

CLI Example:

```
salt dell drac.create_user [USERNAME] [PASSWORD] [PRIVILEGES]
salt dell drac.create_user diana secret login,test_alerts,clear_logs
```

DRAC Privileges

- login : Login to iDRAC
- drac : Configure iDRAC
- user_management : Configure Users
- clear_logs : Clear Logs
- server_control_commands : Execute Server Control Commands
- console_redirection : Access Console Redirection
- virtual_media : Access Virtual Media
- test_alerts : Test Alerts
- debug_commands : Execute Debug Commands

`salt.modules.drac.`**`delete_user`**`(`*username*, *uid=None*`)`

Delete a user

CLI Example:

```
salt dell drac.delete_user [USERNAME] [UID - optional]
salt dell drac.delete_user diana 4
```

`salt.modules.drac.`**`email_alerts`**`(`*action*`)`

Enable/Disable email alerts

CLI Example:

```
salt dell drac.email_alerts True
salt dell drac.email_alerts False
```

`salt.modules.drac.`**`list_users`**`()`

List all DRAC users

CLI Example:

```
salt dell drac.list_users
```

`salt.modules.drac.`**`nameservers`**`(`**ns*`)`

Configure the nameservers on the DRAC

CLI Example:

```
salt dell drac.nameservers [NAMESERVERS]
salt dell drac.nameservers ns1.example.com ns2.example.com
```

`salt.modules.drac.`**`network_info`**`()`

Return Network Configuration

CLI Example:

```
salt dell drac.network_info
```

`salt.modules.drac.`**`server_hardreset`**`()`

Performs a reset (reboot) operation on the managed server.

CLI Example:

```
salt dell drac.server_hardreset
```

`salt.modules.drac.`**`server_poweroff`**`()`

Powers down the managed server.

CLI Example:

```
salt dell drac.server_poweroff
```

`salt.modules.drac.`**`server_poweron`**`()`

Powers up the managed server.

CLI Example:

```
salt dell drac.server_poweron
```

`salt.modules.drac.`**`server_pxe`**`()`

Configure server to PXE perform a one off PXE boot

CLI Example:

```
salt dell drac.server_pxe
```

`salt.modules.drac.`**`server_reboot`**`()`

Issues a power-cycle operation on the managed server. This action is similar to pressing the power button on the system's front panel to power down and then power up the system.

CLI Example:

```
salt dell drac.server_reboot
```

`salt.modules.drac.`**`set_network`**`(`*ip*, *netmask*, *gateway*`)`

Configure Network

CLI Example:

```
salt dell drac.set_network [DRAC IP] [NETMASK] [GATEWAY]
salt dell drac.set_network 192.168.0.2 255.255.255.0 192.168.0.1
```

`salt.modules.drac.`**`set_permissions`**`(`*username*, *permissions*, *uid=None*`)`

Configure users permissions

CLI Example:

```
salt dell drac.set_permissions [USERNAME] [PRIVILEGES] [USER INDEX - optional]
salt dell drac.set_permissions diana login,test_alerts,clear_logs 4
```

DRAC Privileges

- login : Login to iDRAC
- drac : Configure iDRAC
- user_management : Configure Users
- clear_logs : Clear Logs
- server_control_commands : Execute Server Control Commands
- console_redirection : Access Console Redirection
- virtual_media : Access Virtual Media
- test_alerts : Test Alerts
- debug_commands : Execute Debug Commands

`salt.modules.drac.`**`set_snmp`**`(`*community*`)`

Configure SNMP community string

CLI Example:

```
salt dell drac.set_snmp [COMMUNITY]
salt dell drac.set_snmp public
```

`salt.modules.drac.`**`syslog`**`(`*server*, *enable=True*`)`

Configure syslog remote logging, by default syslog will automatically be enabled if a server is specified. However, if you want to disable syslog you will need to specify a server followed by False

CLI Example:

```
salt dell drac.syslog [SYSLOG IP] [ENABLE/DISABLE]
salt dell drac.syslog 0.0.0.0 False
```

`salt.modules.drac.`**`system_info`**`()`

Return System information

CLI Example:

```
salt dell drac.system_info
```

21.9.113 salt.modules.dracr

Manage Dell DRAC.

New in version 2015.8.2.

`salt.modules.dracr.`**`bare_rac_cmd`**`(`*cmd, host=None, admin_username=None, admin_password=None*`)`

`salt.modules.dracr.`**`change_password`**`(`*username, password, uid=None, host=None, admin_username=None, admin_password=None, module=None*`)`

Change user's password

CLI Example:

```
salt dell dracr.change_password [USERNAME] [PASSWORD] uid=[OPTIONAL]
    host=<remote DRAC> admin_username=<DRAC user>
    admin_password=<DRAC PW>
salt dell dracr.change_password diana secret
```

Note that if only a username is specified then this module will look up details for all 16 possible DRAC users. This is time consuming, but might be necessary if one is not sure which user slot contains the one you want. Many late-model Dell chassis have 'root' as UID 1, so if you can depend on that then setting the password is much quicker. Raises an error if the supplied password is greater than 20 chars.

`salt.modules.dracr.`**`create_user`**`(`*username, password, permissions, users=None, host=None, admin_username=None, admin_password=None*`)`

Create user accounts

CLI Example:

```
salt dell dracr.create_user [USERNAME] [PASSWORD] [PRIVILEGES]
salt dell dracr.create_user diana secret login,test_alerts,clear_logs
```

DRAC Privileges

- login : Login to iDRAC
- drac : Configure iDRAC
- user_management : Configure Users
- clear_logs : Clear Logs
- server_control_commands : Execute Server Control Commands
- console_redirection : Access Console Redirection

- virtual_media : Access Virtual Media
- test_alerts : Test Alerts
- debug_commands : Execute Debug Commands

`salt.modules.dracr.`**`delete_user`**`(`*username, uid=None, host=None, admin_username=None, admin_password=None*`)`

Delete a user

CLI Example:

```
salt dell dracr.delete_user [USERNAME] [UID - optional]
salt dell dracr.delete_user diana 4
```

`salt.modules.dracr.`**`deploy_password`**`(`*username, password, host=None, admin_username=None, admin_password=None, module=None*`)`

Change the QuickDeploy password, used for switches as well

CLI Example:

```
salt dell dracr.deploy_password [USERNAME] [PASSWORD]
    host=<remote DRAC> admin_username=<DRAC user>
    admin_password=<DRAC PW>
salt dell dracr.change_password diana secret
```

Note that if only a username is specified then this module will look up details for all 16 possible DRAC users. This is time consuming, but might be necessary if one is not sure which user slot contains the one you want. Many late-model Dell chassis have 'root' as UID 1, so if you can depend on that then setting the password is much quicker.

`salt.modules.dracr.`**`deploy_snmp`**`(`*snmp, host=None, admin_username=None, admin_password=None, module=None*`)`

Change the QuickDeploy SNMP community string, used for switches as well

CLI Example:

```
salt dell dracr.deploy_snmp SNMP_STRING
    host=<remote DRAC or CMC> admin_username=<DRAC user>
    admin_password=<DRAC PW>
salt dell dracr.deploy_password diana secret
```

`salt.modules.dracr.`**`email_alerts`**`(`*action, host=None, admin_username=None, admin_password=None*`)`

Enable/Disable email alerts

CLI Example:

```
salt dell dracr.email_alerts True
salt dell dracr.email_alerts False
```

`salt.modules.dracr.`**`get_chassis_datacenter`**`(`*host=None, admin_username=None, admin_password=None*`)`

Get the datacenter of the chassis.

host The chassis host.

admin_username The username used to access the chassis.

admin_password The password used to access the chassis.

CLI Example:

```
salt '*' dracr.set_chassis_location host=111.222.333.444
   admin_username=root admin_password=secret
```

`salt.modules.dracr.`**`get_chassis_location`**`(`*host=None, admin_username=None, admin_password=None*`)`

Get the location of the chassis.

host The chassis host.

admin_username The username used to access the chassis.

admin_password The password used to access the chassis.

CLI Example:

```
salt '*' dracr.set_chassis_location host=111.222.333.444
   admin_username=root admin_password=secret
```

`salt.modules.dracr.`**`get_chassis_name`**`(`*host=None, admin_username=None, admin_password=None*`)`

Get the name of a chassis.

host The chassis host.

admin_username The username used to access the chassis.

admin_password The password used to access the chassis.

CLI Example:

```
salt '*' dracr.get_chassis_name host=111.222.333.444
    admin_username=root admin_password=secret
```

`salt.modules.dracr.`**`get_dns_dracname`**`(`*host=None, admin_username=None, admin_password=None*`)`

`salt.modules.dracr.`**`get_general`**`(`*cfg_sec, cfg_var, host=None, admin_username=None, admin_password=None*`)`

`salt.modules.dracr.`**`get_slotname`**`(`*slot, host=None, admin_username=None, admin_password=None*`)`

Get the name of a slot number in the chassis.

slot The number of the slot for which to obtain the name.

host The chassis host.

admin_username The username used to access the chassis.

admin_password The password used to access the chassis.

CLI Example:

```
salt-call --local dracr.get_slotname 0 host=111.222.333.444
   admin_username=root admin_password=secret
```

`salt.modules.dracr.`**`idrac_general`**`(`*blade_name, command, idrac_password=None, host=None, admin_username=None, admin_password=None*`)`

Run a generic racadm command against a particular blade in a chassis. Blades are usually named things like 'server-1', 'server-2', etc. If the iDRAC has a different password than the CMC, then you can pass it with the idrac_password kwarg.

Parameters

- **blade_name** – Name of the blade to run the command on
- **command** – Command like to pass to racadm
- **idrac_password** – Password for the iDRAC if different from the CMC
- **host** – Chassis hostname
- **admin_username** – CMC username
- **admin_password** – CMC password

Returns stdout if the retcode is 0, otherwise a standard cmd.run_all dictionary

CLI Example:

```
salt fx2 chassis.cmd idrac_general server-1 'get BIOS.SysProfileSettings'
```

`salt.modules.dracr.`**`inventory`**(*host=None, admin_username=None, admin_password=None*)

`salt.modules.dracr.`**`list_slotnames`**(*host=None, admin_username=None, admin_password=None*)

List the names of all slots in the chassis.

host The chassis host.

admin_username The username used to access the chassis.

admin_password The password used to access the chassis.

CLI Example:

```
salt-call --local dracr.list_slotnames host=111.222.333.444
    admin_username=root admin_password=secret
```

`salt.modules.dracr.`**`list_users`**(*host=None, admin_username=None, admin_password=None, module=None*)

List all DRAC users

CLI Example:

```
salt dell dracr.list_users
```

`salt.modules.dracr.`**`nameservers`**(*ns, host=None, admin_username=None, admin_password=None, module=None*)

Configure the nameservers on the DRAC

CLI Example:

```
salt dell dracr.nameservers [NAMESERVERS]
salt dell dracr.nameservers ns1.example.com ns2.example.com
    admin_username=root admin_password=calvin module=server-1
    host=192.168.1.1
```

`salt.modules.dracr.`**`network_info`**(*host=None, admin_username=None, admin_password=None, module=None*)

Return Network Configuration

CLI Example:

```
salt dell dracr.network_info
```

`salt.modules.dracr.`**`server_hardreset`**`(`*host=None, admin_username=None, admin_password=None, module=None*`)`

Performs a reset (reboot) operation on the managed server.

host The chassis host.

admin_username The username used to access the chassis.

admin_password The password used to access the chassis.

module The element to hard reset on the chassis such as a blade. If not provided, the chassis will be reset.

CLI Example:

```
salt dell dracr.server_hardreset
salt dell dracr.server_hardreset module=server-1
```

`salt.modules.dracr.`**`server_power`**`(`*status, host=None, admin_username=None, admin_password=None, module=None*`)`

status One of 'powerup', 'powerdown', 'powercycle', 'hardreset', 'graceshutdown'

host The chassis host.

admin_username The username used to access the chassis.

admin_password The password used to access the chassis.

module The element to reboot on the chassis such as a blade. If not provided, the chassis will be rebooted.

CLI Example:

```
salt dell dracr.server_reboot
salt dell dracr.server_reboot module=server-1
```

`salt.modules.dracr.`**`server_poweroff`**`(`*host=None, admin_username=None, admin_password=None, module=None*`)`

Powers down the managed server.

host The chassis host.

admin_username The username used to access the chassis.

admin_password The password used to access the chassis.

module The element to power off on the chassis such as a blade. If not provided, the chassis will be powered off.

CLI Example:

```
salt dell dracr.server_poweroff
salt dell dracr.server_poweroff module=server-1
```

`salt.modules.dracr.`**`server_poweron`**`(`*host=None, admin_username=None, admin_password=None, module=None*`)`

Powers up the managed server.

host The chassis host.

admin_username The username used to access the chassis.

admin_password The password used to access the chassis.

module The element to power on located on the chassis such as a blade. If not provided, the chassis will be powered on.

CLI Example:

```
salt dell dracr.server_poweron
salt dell dracr.server_poweron module=server-1
```

`salt.modules.dracr.`**`server_powerstatus`**(*host=None, admin_username=None, admin_password=None, module=None*)

return the power status for the passed module

CLI Example:

```
salt dell drac.server_powerstatus
```

`salt.modules.dracr.`**`server_pxe`**(*host=None, admin_username=None, admin_password=None*)

Configure server to PXE perform a one off PXE boot

CLI Example:

```
salt dell dracr.server_pxe
```

`salt.modules.dracr.`**`server_reboot`**(*host=None, admin_username=None, admin_password=None, module=None*)

Issues a power-cycle operation on the managed server. This action is similar to pressing the power button on the system's front panel to power down and then power up the system.

host The chassis host.

admin_username The username used to access the chassis.

admin_password The password used to access the chassis.

module The element to reboot on the chassis such as a blade. If not provided, the chassis will be rebooted.

CLI Example:

```
salt dell dracr.server_reboot
salt dell dracr.server_reboot module=server-1
```

`salt.modules.dracr.`**`set_chassis_datacenter`**(*location, host=None, admin_username=None, admin_password=None*)

Set the location of the chassis.

location The name of the datacenter to be set on the chassis.

host The chassis host.

admin_username The username used to access the chassis.

admin_password The password used to access the chassis.

CLI Example:

```
salt '*' dracr.set_chassis_datacenter datacenter-name host=111.222.333.444
    admin_username=root admin_password=secret
```

`salt.modules.dracr.`**`set_chassis_location`**(*location, host=None, admin_username=None, admin_password=None*)

Set the location of the chassis.

location The name of the location to be set on the chassis.

host The chassis host.

admin_username The username used to access the chassis.

admin_password The password used to access the chassis.

CLI Example:

```
salt '*' dracr.set_chassis_location location-name host=111.222.333.444
    admin_username=root admin_password=secret
```

salt.modules.dracr.**set_chassis_name**(*name, host=None, admin_username=None, admin_password=None*)

Set the name of the chassis.

name The name to be set on the chassis.

host The chassis host.

admin_username The username used to access the chassis.

admin_password The password used to access the chassis.

CLI Example:

```
salt '*' dracr.set_chassis_name my-chassis host=111.222.333.444
    admin_username=root admin_password=secret
```

salt.modules.dracr.**set_dns_dracname**(*name, host=None, admin_username=None, admin_password=None*)

salt.modules.dracr.**set_general**(*cfg_sec, cfg_var, val, host=None, admin_username=None, admin_password=None*)

salt.modules.dracr.**set_network**(*ip, netmask, gateway, host=None, admin_username=None, admin_password=None*)

Configure Network on the CMC or individual iDRAC. Use `set_niccfg` for blade and switch addresses.

CLI Example:

```
salt dell dracr.set_network [DRAC IP] [NETMASK] [GATEWAY]
salt dell dracr.set_network 192.168.0.2 255.255.255.0 192.168.0.1
    admin_username=root admin_password=calvin host=192.168.1.1
```

salt.modules.dracr.**set_niccfg**(*ip=None, netmask=None, gateway=None, dhcp=False, host=None, admin_username=None, admin_password=None, module=None*)

salt.modules.dracr.**set_nicvlan**(*vlan=None, host=None, admin_username=None, admin_password=None, module=None*)

salt.modules.dracr.**set_permissions**(*username, permissions, uid=None, host=None, admin_username=None, admin_password=None*)

Configure users permissions

CLI Example:

```
salt dell dracr.set_permissions [USERNAME] [PRIVILEGES]
     [USER INDEX - optional]
salt dell dracr.set_permissions diana login,test_alerts,clear_logs 4
```

DRAC Privileges

- login : Login to iDRAC
- drac : Configure iDRAC
- user_management : Configure Users

- clear_logs : Clear Logs
- server_control_commands : Execute Server Control Commands
- console_redirection : Access Console Redirection
- virtual_media : Access Virtual Media
- test_alerts : Test Alerts
- debug_commands : Execute Debug Commands

`salt.modules.dracr.`**`set_slotname`**`(`*slot, name, host=None, admin_username=None, admin_password=None*`)`

Set the name of a slot in a chassis.

slot The slot number to change.

name The name to set. Can only be 15 characters long.

host The chassis host.

admin_username The username used to access the chassis.

admin_password The password used to access the chassis.

CLI Example:

```
salt '*' dracr.set_slotname 2 my-slotname host=111.222.333.444
    admin_username=root admin_password=secret
```

`salt.modules.dracr.`**`set_snmp`**`(`*community, host=None, admin_username=None, admin_password=None*`)`

Configure CMC or individual iDRAC SNMP community string. Use `deploy_snmp` for configuring chassis switch SNMP.

CLI Example:

```
salt dell dracr.set_snmp [COMMUNITY]
salt dell dracr.set_snmp public
```

`salt.modules.dracr.`**`syslog`**`(`*server, enable=True, host=None, admin_username=None, admin_password=None, module=None*`)`

Configure syslog remote logging, by default syslog will automatically be enabled if a server is specified. However, if you want to disable syslog you will need to specify a server followed by False

CLI Example:

```
salt dell dracr.syslog [SYSLOG IP] [ENABLE/DISABLE]
salt dell dracr.syslog 0.0.0.0 False
```

`salt.modules.dracr.`**`system_info`**`(`*host=None, admin_username=None, admin_password=None, module=None*`)`

Return System information

CLI Example:

```
salt dell dracr.system_info
```

`salt.modules.dracr.`**`update_firmware`**`(`*filename, host=None, admin_username=None, admin_password=None*`)`

Updates firmware using local firmware file

```
salt dell dracr.update_firmware firmware.exe
```

This executes the following command on your FX2 (using username and password stored in the pillar data)

```
racadm update -f firmware.exe -u user -p pass
```

`salt.modules.dracr.`**`update_firmware_nfs_or_cifs`**(*filename, share, host=None, admin_username=None, admin_password=None*)

Executes the following for CIFS (using username and password stored in the pillar data)

```
racadm update -f <updatefile> -u user -p pass -l //IP-Address/share
```

Or for NFS (using username and password stored in the pillar data)

```
racadm update -f <updatefile> -u user -p pass -l IP-address:/share
```

Salt command for CIFS:

```
salt dell dracr.update_firmware_nfs_or_cifs          firmware.exe //IP-Address/
↪share
```

Salt command for NFS:

```
salt dell dracr.update_firmware_nfs_or_cifs          firmware.exe IP-address:/
↪share
```

21.9.114 salt.modules.drbd

DRBD administration module

`salt.modules.drbd.`**`overview`**()

Show status of the DRBD devices, support two nodes only.

CLI Example:

```
salt '*' drbd.overview
```

21.9.115 salt.modules.dummyproxy_pkg module

Package support for the dummy proxy used by the test suite

`salt.modules.dummyproxy_pkg.`**`install`**(*name=None, refresh=False, fromrepo=None, pkgs=None, sources=None, **kwargs*)

`salt.modules.dummyproxy_pkg.`**`installed`**(*name, version=None, refresh=False, fromrepo=None, skip_verify=False, pkgs=None, sources=None, **kwargs*)

`salt.modules.dummyproxy_pkg.`**`list_pkgs`**(*versions_as_list=False, **kwargs*)

`salt.modules.dummyproxy_pkg.`**`remove`**(*name=None, pkgs=None, **kwargs*)

`salt.modules.dummyproxy_pkg.`**`upgrade`**(*name=None, pkgs=None, refresh=True, skip_verify=True, normalize=True, **kwargs*) `salt.modules.dummyproxy_pkg.`**`version`**`(`**names*, ***kwargs*`)`

Returns a string representing the package version or an empty string if not installed. If more than one package name is specified, a dict of name/version pairs is returned.

CLI Example:

```
salt '*' pkg.version <package name>
salt '*' pkg.version <package1> <package2> <package3> ...
```

21.9.116 salt.modules.dummyproxy_service module

Provide the service module for the dummy proxy used in integration tests

`salt.modules.dummyproxy_service.`**`enabled`**`(`*name*, *sig=None*`)`

Only the 'redbull' service is 'enabled' in the test

New in version 2016.11.3.

`salt.modules.dummyproxy_service.`**`get_all`**`()`

Return a list of all available services

New in version 2016.11.3.

CLI Example:

```
salt '*' service.get_all
```

`salt.modules.dummyproxy_service.`**`list_`**`()`

Return a list of all available services.

New in version 2016.11.3.

CLI Example:

```
salt '*' service.list
```

`salt.modules.dummyproxy_service.`**`restart`**`(`*name*, *sig=None*`)`

Restart the specified service with dummy.

New in version 2016.11.3.

CLI Example:

```
salt '*' service.restart <service name>
```

`salt.modules.dummyproxy_service.`**`running`**`(`*name*, *sig=None*`)`

Return whether this service is running.

New in version 2016.11.3.

`salt.modules.dummyproxy_service.`**`start`**`(`*name*, *sig=None*`)`

Start the specified service on the dummy

New in version 2016.11.3.

CLI Example:

```
salt '*' service.start <service name>
```

`salt.modules.dummyproxy_service.`**`status`**`(`*name*, *sig=None*`)`

Return the status for a service via dummy, returns a bool whether the service is running.

New in version 2016.11.3.

CLI Example:

```
salt '*' service.status <service name>
```

`salt.modules.dummyproxy_service.`**`stop`**`(`*name*, *sig=None*`)`

Stop the specified service on the dummy

New in version 2016.11.3.

CLI Example:

```
salt '*' service.stop <service name>
```

21.9.117 salt.modules.ebuildpkg

Support for Portage

Important: If you feel that Salt should be using this module to manage packages on a minion, and it is using a different module (or gives an error similar to *'pkg.install' is not available*), see *here*.

---

optdepends

- portage Python adapter

For now all package names *MUST* include the package category, i.e. `'vim'` will not work, `'app-editors/vim'` will.

`salt.modules.ebuildpkg.`**`check_db`**`(`**names*, ***kwargs*`)`

New in version 0.17.0.

Returns a dict containing the following information for each specified package:

1. A key `found`, which will be a boolean value denoting if a match was found in the package database.
2. If `found` is `False`, then a second key called `suggestions` will be present, which will contain a list of possible matches. This list will be empty if the package name was specified in `category/pkgname` format, since the suggestions are only intended to disambiguate ambiguous package names (ones submitted without a category).

CLI Examples:

```
salt '*' pkg.check_db <package1> <package2> <package3>
```

`salt.modules.ebuildpkg.`**`check_extra_requirements`**`(`*pkgname*, *pkgver*`)`

Check if the installed package already has the given requirements.

CLI Example:

```
salt '*' pkg.check_extra_requirements 'sys-devel/gcc' '~>4.1.2:4.1::gentoo[nls,
↪fortran]'
```

`salt.modules.ebuildpkg.`**`depclean`**`(`*name=None*, *slot=None*, *fromrepo=None*, *pkgs=None*`)`

Portage has a function to remove unused dependencies. If a package is provided, it will only removed the package if no other package depends on it.

name The name of the package to be cleaned.

slot Restrict the remove to a specific slot. Ignored if `name` is None.

fromrepo Restrict the remove to a specific slot. Ignored if `name` is None.

pkgs Clean multiple packages. `slot` and `fromrepo` arguments are ignored if this argument is present. Must be passed as a python list.

Return a list containing the removed packages:

CLI Example:

```
salt '*' pkg.depclean <package name>
```

`salt.modules.ebuildpkg.`**`ex_mod_init`**(*low*)

If the config option `ebuild.enforce_nice_config` is set to True, this module will enforce a nice tree structure for /etc/portage/package.* configuration files.

New in version 0.17.0: Initial automatic enforcement added when pkg is used on a Gentoo system.

Changed in version 2014.1.0-Hydrogen: Configure option added to make this behaviour optional, defaulting to off.

See also:

`ebuild.ex_mod_init` is called automatically when a state invokes a pkg state on a Gentoo system. `salt.states.pkg.mod_init()`

`ebuild.ex_mod_init` uses `portage_config.enforce_nice_config` to do the lifting. *`salt.modules.portage_config.enforce_nice_config()`*

CLI Example:

```
salt '*' pkg.ex_mod_init
```

`salt.modules.ebuildpkg.`**`install`**(*name=None*, *refresh=False*, *pkgs=None*, *sources=None*, *slot=None*, *fromrepo=None*, *uses=None*, *binhost=None*, ***kwargs*)

Changed in version 2015.8.12,2016.3.3,2016.11.0: On minions running systemd>=205, systemd-run(1) is now used to isolate commands which modify installed packages from the `salt-minion` daemon's control group. This is done to keep systemd from killing any emerge commands spawned by Salt when the `salt-minion` service is restarted. (see `KillMode` in the systemd.kill(5) manpage for more information). If desired, usage of systemd-run(1) can be suppressed by setting a *`config option`* called `systemd.scope`, with a value of `False` (no quotes).

Install the passed package(s), add refresh=True to sync the portage tree before package is installed.

name The name of the package to be installed. Note that this parameter is ignored if either "pkgs" or "sources" is passed. Additionally, please note that this option can only be used to emerge a package from the portage tree. To install a tbz2 package manually, use the "sources" option described below.

CLI Example:

```
salt '*' pkg.install <package name>
```

refresh Whether or not to sync the portage tree before installing.

version Install a specific version of the package, e.g. 1.0.9-r1. Ignored if "pkgs" or "sources" is passed.

slot Similar to version, but specifies a valid slot to be installed. It will install the latest available version in the specified slot. Ignored if "pkgs" or "sources" or "version" is passed.

CLI Example:

```
salt '*' pkg.install sys-devel/gcc slot='4.4'
```

fromrepo Similar to slot, but specifies the repository from the package will be installed. It will install the latest available version in the specified repository. Ignored if "pkgs" or "sources" or "version" is passed.

CLI Example:

```
salt '*' pkg.install salt fromrepo='gentoo'
```

uses Similar to slot, but specifies a list of use flag. Ignored if "pkgs" or "sources" or "version" is passed.

CLI Example:

```
salt '*' pkg.install sys-devel/gcc uses='["nptl","-nossp"]'
```

Multiple Package Installation Options:

pkgs A list of packages to install from the portage tree. Must be passed as a python list.

CLI Example:

```
salt '*' pkg.install pkgs='["foo","bar","~category/
↪package:slot::repository[use]"]'
```

sources A list of tbz2 packages to install. Must be passed as a list of dicts, with the keys being package names, and the values being the source URI or local path to the package.

CLI Example:

```
salt '*' pkg.install sources='[{"foo": "salt://foo.tbz2"},{"bar": "salt://bar.
↪tbz2"}]'
```

binhost has two options try and force. try - tells emerge to try and install the package from a configured binhost. force - forces emerge to install the package from a binhost otherwise it fails out.

Returns a dict containing the new package names and versions:

```
{'<package>': {'old': '<old-version>',
               'new': '<new-version>'}}
```

`salt.modules.ebuildpkg.`**`latest_version`**(**names, **kwargs*)

Return the latest version of the named package available for upgrade or installation. If more than one package name is specified, a dict of name/version pairs is returned.

CLI Example:

```
salt '*' pkg.latest_version <package name>
salt '*' pkg.latest_version <package1> <package2> <package3> ...
```

`salt.modules.ebuildpkg.`**`list_pkgs`**(*versions_as_list=False, **kwargs*)

List the packages currently installed in a dict:

```
{'<package_name>': '<version>'}
```

CLI Example:

```
salt '*' pkg.list_pkgs
```

`salt.modules.ebuildpkg.`**`list_upgrades`**(*refresh=True, backtrack=3, **kwargs*)

List all available package upgrades.

refresh Whether or not to sync the portage tree before checking for upgrades.

backtrack Specifies an integer number of times to backtrack if dependency calculation fails due to a conflict or an unsatisfied dependency (default: ´3´).

CLI Example:

```
salt '*' pkg.list_upgrades
```

`salt.modules.ebuildpkg.`**`porttree_matches`**`(`*name*`)`

Returns a list containing the matches for a given package name from the portage tree. Note that the specific version of the package will not be provided for packages that have several versions in the portage tree, but rather the name of the package (i.e. "dev-python/paramiko").

`salt.modules.ebuildpkg.`**`purge`**`(`*name=None, slot=None, fromrepo=None, pkgs=None, **kwargs*`)`

Changed in version 2015.8.12,2016.3.3,2016.11.0: On minions running systemd>=205, systemd-run(1) is now used to isolate commands which modify installed packages from the `salt-minion` daemon's control group. This is done to keep systemd from killing any emerge commands spawned by Salt when the `salt-minion` service is restarted. (see `KillMode` in the systemd.kill(5) manpage for more information). If desired, usage of systemd-run(1) can be suppressed by setting a *`config option`* called `systemd.scope`, with a value of `False` (no quotes).

Portage does not have a purge, this function calls remove followed by depclean to emulate a purge process

name The name of the package to be deleted.

slot Restrict the remove to a specific slot. Ignored if name is None.

fromrepo Restrict the remove to a specific slot. Ignored if `name` is None.

Multiple Package Options:

pkgs Uninstall multiple packages. `slot` and `fromrepo` arguments are ignored if this argument is present. Must be passed as a python list.

New in version 0.16.0.

Returns a dict containing the changes.

CLI Example:

```
salt '*' pkg.purge <package name>
salt '*' pkg.purge <package name> slot=4.4
salt '*' pkg.purge <package1>,<package2>,<package3>
salt '*' pkg.purge pkgs='["foo", "bar"]'
```

`salt.modules.ebuildpkg.`**`refresh_db`**`()`

Update the portage tree using the first available method from the following list:

- emaint sync
- eix-sync
- emerge-webrsync
- emerge –sync

To prevent the portage tree from being synced within one day of the previous sync, add the following pillar data for this minion:

```
portage:
  sync_wait_one_day: True
```

CLI Example:

```
salt '*' pkg.refresh_db
```

`salt.modules.ebuildpkg.`**`remove`**`(`*name=None, slot=None, fromrepo=None, pkgs=None, **kwargs*`)`

Changed in version 2015.8.12,2016.3.3,2016.11.0: On minions running systemd>=205, systemd-run(1) is now used to isolate commands which modify installed packages from the `salt-minion` daemon's control group. This is done to keep systemd from killing any emerge commands spawned by Salt when the `salt-minion` service is restarted. (see `KillMode` in the systemd.kill(5) manpage for more information). If desired, usage of systemd-run(1) can be suppressed by setting a *`config option`* called `systemd.scope`, with a value of `False` (no quotes).

Remove packages via emerge –unmerge.

name The name of the package to be deleted.

slot Restrict the remove to a specific slot. Ignored if `name` is None.

fromrepo Restrict the remove to a specific slot. Ignored if `name` is None.

Multiple Package Options:

pkgs Uninstall multiple packages. `slot` and `fromrepo` arguments are ignored if this argument is present. Must be passed as a python list.

New in version 0.16.0.

Returns a dict containing the changes.

CLI Example:

```
salt '*' pkg.remove <package name>
salt '*' pkg.remove <package name> slot=4.4 fromrepo=gentoo
salt '*' pkg.remove <package1>,<package2>,<package3>
salt '*' pkg.remove pkgs='["foo", "bar"]'
```

`salt.modules.ebuildpkg.`**`update`**`(`*pkg, slot=None, fromrepo=None, refresh=False, binhost=None*`)`

Changed in version 2015.8.12,2016.3.3,2016.11.0: On minions running systemd>=205, systemd-run(1) is now used to isolate commands which modify installed packages from the `salt-minion` daemon's control group. This is done to keep systemd from killing any emerge commands spawned by Salt when the `salt-minion` service is restarted. (see `KillMode` in the systemd.kill(5) manpage for more information). If desired, usage of systemd-run(1) can be suppressed by setting a *`config option`* called `systemd.scope`, with a value of `False` (no quotes).

Updates the passed package (emerge –update package)

slot Restrict the update to a particular slot. It will update to the latest version within the slot.

fromrepo Restrict the update to a particular repository. It will update to the latest version within the repository.

binhost has two options try and force. try - tells emerge to try and install the package from a configured binhost. force - forces emerge to install the package from a binhost otherwise it fails out.

Return a dict containing the new package names and versions:

```
{'<package>': {'old': '<old-version>',
               'new': '<new-version>'}}
```

CLI Example:

```
salt '*' pkg.update <package name>
```

`salt.modules.ebuildpkg.`**`upgrade`**(*refresh=True*, *binhost=None*, *backtrack=3*)

Changed in version 2015.8.12,2016.3.3,2016.11.0: On minions running systemd>=205, systemd-run(1) is now used to isolate commands which modify installed packages from the `salt-minion` daemon's control group. This is done to keep systemd from killing any emerge commands spawned by Salt when the `salt-minion` service is restarted. (see `KillMode` in the systemd.kill(5) manpage for more information). If desired, usage of systemd-run(1) can be suppressed by setting a *`config option`* called `systemd.scope`, with a value of `False` (no quotes).

Run a full system upgrade (emerge -uDN @world)

binhost has two options try and force. try - tells emerge to try and install the package from a configured binhost. force - forces emerge to install the package from a binhost otherwise it fails out.

backtrack Specifies an integer number of times to backtrack if dependency calculation fails due to a conflict or an unsatisfied dependency (default: ´3´).

Returns a dictionary containing the changes:

```
{'<package>':  {'old': '<old-version>',
                'new': '<new-version>'}}
```

CLI Example:

```
salt '*' pkg.upgrade
```

`salt.modules.ebuildpkg.`**`upgrade_available`**(*name*)

Check whether or not an upgrade is available for a given package

CLI Example:

```
salt '*' pkg.upgrade_available <package name>
```

`salt.modules.ebuildpkg.`**`version`**(**names*, ***kwargs*)

Returns a string representing the package version or an empty string if not installed. If more than one package name is specified, a dict of name/version pairs is returned.

CLI Example:

```
salt '*' pkg.version <package name>
salt '*' pkg.version <package1> <package2> <package3> ...
```

`salt.modules.ebuildpkg.`**`version_clean`**(*version*)

Clean the version string removing extra data.

CLI Example:

```
salt '*' pkg.version_clean <version_string>
```

`salt.modules.ebuildpkg.`**`version_cmp`**(*pkg1*, *pkg2*, ***kwargs*)

Do a cmp-style comparison on two packages. Return -1 if pkg1 < pkg2, 0 if pkg1 == pkg2, and 1 if pkg1 > pkg2. Return None if there was a problem making the comparison.

CLI Example:

```
salt '*' pkg.version_cmp '0.2.4-0' '0.2.4.1-0'
```

21.9.118 salt.modules.eix

Support for Eix

salt.modules.eix.**sync**()

Sync portage/overlay trees and update the eix database

CLI Example:

```
salt '*' eix.sync
```

salt.modules.eix.**update**()

Update the eix database

CLI Example:

```
salt '*' eix.update
```

21.9.119 salt.modules.elasticsearch

Elasticsearch - A distributed RESTful search and analytics server

Module to provide Elasticsearch compatibility to Salt (compatible with Elasticsearch version 1.5.2+)

New in version 2015.8.0.

depends elasticsearch-py

configuration This module accepts connection configuration details either as parameters or as configuration settings in /etc/salt/minion on the relevant minions:

```
elasticsearch:
  host: '10.10.10.100:9200'

elasticsearch-cluster:
  hosts:
    - '10.10.10.100:9200'
    - '10.10.10.101:9200'
    - '10.10.10.102:9200'

elasticsearch-extra:
  hosts:
    - '10.10.10.100:9200'
  use_ssl: True
  verify_certs: True
  ca_certs: /path/to/custom_ca_bundle.pem
  number_of_shards: 1
  number_of_replicas: 0
  functions_blacklist:
    - 'saltutil.find_job'
    - 'pillar.items'
    - 'grains.items'
  proxies:
    - http: http://proxy:3128
    - https: http://proxy:1080
```

When specifying proxies the requests backend will be used and the 'proxies' data structure is passed as-is to that module.

This data can also be passed into pillar. Options passed into opts will overwrite options passed into pillar.

Some functionality might be limited by elasticsearch-py and Elasticsearch server versions.

`salt.modules.elasticsearch.`**`alias_create`**(*indices*, *alias*, *hosts=None*, *body=None*, *profile=None*, *source=None*)

Create an alias for a specific index/indices

indices Single or multiple indices separated by comma, use _all to perform the operation on all indices.

alias Alias name

body Optional definition such as routing or filter as defined in https://www.elastic.co/guide/en/elasticsearch/reference/current/indices-aliases.html

source URL of file specifying optional definition such as routing or filter. Cannot be used in combination with `body`.

CLI example:

```
salt myminion elasticsearch.alias_create testindex_v1 testindex
```

`salt.modules.elasticsearch.`**`alias_delete`**(*indices*, *aliases*, *hosts=None*, *body=None*, *profile=None*, *source=None*)

Delete an alias of an index

indices Single or multiple indices separated by comma, use _all to perform the operation on all indices.

aliases Alias names separated by comma

CLI example:

```
salt myminion elasticsearch.alias_delete testindex_v1 testindex
```

`salt.modules.elasticsearch.`**`alias_exists`**(*aliases*, *indices=None*, *hosts=None*, *profile=None*)

Return a boolean indicating whether given alias exists

indices Single or multiple indices separated by comma, use _all to perform the operation on all indices.

aliases Alias names separated by comma

CLI example:

```
salt myminion elasticsearch.alias_exists None testindex
```

`salt.modules.elasticsearch.`**`alias_get`**(*indices=None*, *aliases=None*, *hosts=None*, *profile=None*)

Check for the existence of an alias and if it exists, return it

indices Single or multiple indices separated by comma, use _all to perform the operation on all indices.

aliases Alias names separated by comma

CLI example:

```
salt myminion elasticsearch.alias_get testindex
```

`salt.modules.elasticsearch.`**`cluster_get_settings`**(*flat_settings=False*, *include_defaults=False*, *hosts=None*, *profile=None*)

New in version 3000.

Return Elasticsearch cluster settings.

flat_settings Return settings in flat format.

include_defaults Whether to return all default clusters setting.

CLI example:

```
salt myminion elasticsearch.cluster_get_settings
```

`salt.modules.elasticsearch.`**`cluster_health`**(*index=None, level='cluster', local=False, hosts=None, profile=None*)

New in version 2017.7.0.

Return Elasticsearch cluster health.

index Limit the information returned to a specific index

level Specify the level of detail for returned information, default 'cluster', valid choices are: 'cluster', 'indices', 'shards'

local Return local information, do not retrieve the state from master node

CLI example:

```
salt myminion elasticsearch.cluster_health
```

`salt.modules.elasticsearch.`**`cluster_put_settings`**(*body=None, flat_settings=False, hosts=None, profile=None*)

New in version 3000.

Set Elasticsearch cluster settings.

body The settings to be updated. Can be either 'transient' or 'persistent' (survives cluster restart) http://www.elastic.co/guide/en/elasticsearch/reference/current/cluster-update-settings.html

flat_settings Return settings in flat format.

CLI example:

```
salt myminion elasticsearch.cluster_put_settings '{"persistent": {"indices.
↪recovery.max_bytes_per_sec": "50mb"}}'
salt myminion elasticsearch.cluster_put_settings '{"transient": {"indices.
↪recovery.max_bytes_per_sec": "50mb"}}'
```

`salt.modules.elasticsearch.`**`cluster_stats`**(*nodes=None, hosts=None, profile=None*)

New in version 2017.7.0.

Return Elasticsearch cluster stats.

nodes List of cluster nodes (id or name) to display stats for. Use _local for connected node, empty for all

CLI example:

```
salt myminion elasticsearch.cluster_stats
```

`salt.modules.elasticsearch.`**`document_create`**(*index, doc_type, body=None, id=None, hosts=None, profile=None, source=None*)

Create a document in a specified index

index Index name where the document should reside

doc_type Type of the document

body Document to store

source URL of file specifying document to store. Cannot be used in combination with `body`.

id Optional unique document identifier for specified doc_type (empty for random)

CLI example:

```
salt myminion elasticsearch.document_create testindex doctype1 '{}'
```

salt.modules.elasticsearch.**document_delete**(*index*, *doc_type*, *id*, *hosts=None*, *profile=None*)

Delete a document from an index

index Index name where the document resides

doc_type Type of the document

id Document identifier

CLI example:

```
salt myminion elasticsearch.document_delete testindex doctype1 AUx-384m0Bug_
↪8U80wQZ
```

salt.modules.elasticsearch.**document_exists**(*index*, *id*, *doc_type='_all'*, *hosts=None*, *profile=None*)

Return a boolean indicating whether given document exists

index Index name where the document resides

id Document identifier

doc_type Type of the document, use _all to fetch the first document matching the ID across all types

CLI example:

```
salt myminion elasticsearch.document_exists testindex AUx-384m0Bug_8U80wQZ
```

salt.modules.elasticsearch.**document_get**(*index*, *id*, *doc_type='_all'*, *hosts=None*, *profile=None*)

Check for the existence of a document and if it exists, return it

index Index name where the document resides

id Document identifier

doc_type Type of the document, use _all to fetch the first document matching the ID across all types

CLI example:

```
salt myminion elasticsearch.document_get testindex AUx-384m0Bug_8U80wQZ
```

salt.modules.elasticsearch.**flush_synced**(*hosts=None*, *profile=None*, ***kwargs*)

New in version 3000.

Perform a normal flush, then add a generated unique marker (sync_id) to all shards. http://www.elastic.co/guide/en/elasticsearch/reference/current/indices-synced-flush.html

index (Optional, string) A comma-separated list of index names; use _all or empty string for all indices. Defaults to '_all'.

ignore_unavailable (Optional, boolean) If true, missing or closed indices are not included in the response. Defaults to false.

allow_no_indices (Optional, boolean) If true, the request does not return an error if a wildcard expression or _all value retrieves only missing or closed indices. This parameter also applies to index aliases that point to a missing or closed index.

expand_wildcards (Optional, string) Controls what kind of indices that wildcard expressions can expand to.

Valid values are:

```
all - Expand to open and closed indices.
open - Expand only to open indices.
closed - Expand only to closed indices.
none - Wildcard expressions are not accepted.
```

The defaults settings for the above parameters depend on the API being used.

CLI example:

```
salt myminion elasticsearch.flush_synced index='index1,index2' ignore_
↪unavailable=True allow_no_indices=True expand_wildcards='all'
```

`salt.modules.elasticsearch.`**`index_close`**(*index*, *allow_no_indices=True*, *expand_wildcards='open'*, *ignore_unavailable=True*, *hosts=None*, *profile=None*)

New in version 2017.7.0.

Close specified index.

index Index to be closed

allow_no_indices Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes _all string or when no indices have been specified)

expand_wildcards Whether to expand wildcard expression to concrete indices that are open, closed or both., default 'open', valid choices are: 'open', 'closed', 'none', 'all'

ignore_unavailable Whether specified concrete indices should be ignored when unavailable (missing or closed)

CLI example:

```
salt myminion elasticsearch.index_close testindex
```

`salt.modules.elasticsearch.`**`index_create`**(*index*, *body=None*, *hosts=None*, *profile=None*, *source=None*)

Create an index

index Index name

body Index definition, such as settings and mappings as defined in https://www.elastic.co/guide/en/elasticsearch/reference/current/indices-create-index.html

source URL to file specifying index definition. Cannot be used in combination with `body`.

CLI example:

```
salt myminion elasticsearch.index_create testindex
salt myminion elasticsearch.index_create testindex2 '{"settings" : {"index" : {
↪"number_of_shards" : 3, "number_of_replicas" : 2}}}'
```

`salt.modules.elasticsearch.`**`index_delete`**(*index*, *hosts=None*, *profile=None*)

Delete an index

index Index name

CLI example:

```
salt myminion elasticsearch.index_delete testindex
```

`salt.modules.elasticsearch.`**`index_exists`**(*index*, *hosts=None*, *profile=None*)

Return a boolean indicating whether given index exists

index Index name

CLI example:

```
salt myminion elasticsearch.index_exists testindex
```

`salt.modules.elasticsearch.`**`index_get`**(*index*, *hosts=None*, *profile=None*)

Check for the existence of an index and if it exists, return it

index Index name

CLI example:

```
salt myminion elasticsearch.index_get testindex
```

`salt.modules.elasticsearch.`**`index_get_settings`**(*hosts=None*, *profile=None*, ***kwargs*)

New in version 3000.

Check for the existence of an index and if it exists, return its settings http://www.elastic.co/guide/en/elasticsearch/reference/current/indices-get-settings.html

index (Optional, string) A comma-separated list of index names; use _all or empty string for all indices. Defaults to '_all'.

name (Optional, string) The name of the settings that should be included

allow_no_indices (Optional, boolean) Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes _all string or when no indices have been specified)

expand_wildcards (Optional, string) Whether to expand wildcard expression to concrete indices that are open, closed or both. Valid choices are: 'open', 'closed', 'none', 'all'

flat_settings (Optional, boolean) Return settings in flat format

ignore_unavailable (Optional, boolean) Whether specified concrete indices should be ignored when unavailable (missing or closed)

include_defaults (Optional, boolean) Whether to return all default setting for each of the indices.

local (Optional, boolean) Return local information, do not retrieve the state from master node

The defaults settings for the above parameters depend on the API version being used.

CLI example:

```
salt myminion elasticsearch.index_get_settings index=testindex
```

`salt.modules.elasticsearch.`**`index_open`**(*index*, *allow_no_indices=True*, *expand_wildcards='closed'*, *ignore_unavailable=True*, *hosts=None*, *profile=None*)

New in version 2017.7.0.

Open specified index.

index Index to be opened

allow_no_indices Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes _all string or when no indices have been specified)

expand_wildcards Whether to expand wildcard expression to concrete indices that are open, closed or both., default 'closed', valid choices are: 'open', 'closed', 'none', 'all'

ignore_unavailable Whether specified concrete indices should be ignored when unavailable (missing or closed)

CLI example:

```
salt myminion elasticsearch.index_open testindex
```

`salt.modules.elasticsearch.`**`index_put_settings`**(*body=None*, *hosts=None*, *profile=None*, *source=None*, ***kwargs*)

New in version 3000.

Update existing index settings https://www.elastic.co/guide/en/elasticsearch/reference/current/indices-update-settings.html

body The index settings to be updated.

source URL to file specifying index definition. Cannot be used in combination with `body`.

index (Optional, string) A comma-separated list of index names; use _all or empty string to perform the operation on all indices

allow_no_indices (Optional, boolean) Whether to ignore if a wildcard indices expression resolves into no concrete indices. (This includes _all string or when no indices have been specified)

expand_wildcards (Optional, string) Whether to expand wildcard expression to concrete indices that are open, closed or both. Valid choices are: 'open', 'closed', 'none', 'all'

flat_settings (Optional, boolean) Return settings in flat format (default: false)

ignore_unavailable (Optional, boolean) Whether specified concrete indices should be ignored when unavailable (missing or closed)

master_timeout (Optional, time units) Explicit operation timeout for connection to master node

preserve_existing (Optional, boolean) Whether to update existing settings. If set to true existing settings on an index remain unchanged, the default is false

The defaults settings for the above parameters depend on the API version being used.

..**note**:: Elasticsearch time units can be found here: https://www.elastic.co/guide/en/elasticsearch/reference/current/common-options.html#time-units

CLI example:

```
salt myminion elasticsearch.index_put_settings index=testindex body='{"settings"
↪: {"index" : {"number_of_replicas" : 2}}}'
```

`salt.modules.elasticsearch.`**`index_template_create`**(*name*, *body=None*, *hosts=None*, *profile=None*, *source=None*)

Create an index template

name Index template name

body Template definition as specified in http://www.elastic.co/guide/en/elasticsearch/reference/current/indices-templates.html

source URL to file specifying template definition. Cannot be used in combination with `body`.

CLI example:

```
salt myminion elasticsearch.index_template_create testindex_templ '{ "template":
↪"logstash-*", "order": 1, "settings": { "number_of_shards": 1 } }'
```

`salt.modules.elasticsearch.`**`index_template_delete`**(*name*, *hosts=None*, *profile=None*)

Delete an index template (type) along with its data

name Index template name

CLI example:

```
salt myminion elasticsearch.index_template_delete testindex_templ user
```

`salt.modules.elasticsearch.`**`index_template_exists`**(*name*, *hosts=None*, *profile=None*)

Return a boolean indicating whether given index template exists

name Index template name

CLI example:

```
salt myminion elasticsearch.index_template_exists testindex_templ
```

`salt.modules.elasticsearch.`**`index_template_get`**(*name*, *hosts=None*, *profile=None*)

Retrieve template definition of index or index/type

name Index template name

CLI example:

```
salt myminion elasticsearch.index_template_get testindex_templ
```

`salt.modules.elasticsearch.`**`info`**(*hosts=None*, *profile=None*)

New in version 2017.7.0.

Return Elasticsearch information.

CLI example:

```
salt myminion elasticsearch.info
salt myminion elasticsearch.info profile=elasticsearch-extra
```

`salt.modules.elasticsearch.`**`mapping_create`**(*index*, *doc_type*, *body=None*, *hosts=None*, *profile=None*, *source=None*)

Create a mapping in a given index

index Index for the mapping

doc_type Name of the document type

body Mapping definition as specified in https://www.elastic.co/guide/en/elasticsearch/reference/current/indices-put-mapping.html

source URL to file specifying mapping definition. Cannot be used in combination with `body`.

CLI example:

```
salt myminion elasticsearch.mapping_create testindex user '{ "user" : {
↪"properties" : { "message" : {"type" : "string", "store" : true } } } }'
```

`salt.modules.elasticsearch.`**`mapping_delete`**(*index*, *doc_type*, *hosts=None*, *profile=None*)

Delete a mapping (type) along with its data. As of Elasticsearch 5.0 this is no longer available.

index Index for the mapping

doc_type Name of the document type

CLI example:

```
salt myminion elasticsearch.mapping_delete testindex user
```

`salt.modules.elasticsearch.`**`mapping_get`**(*index*, *doc_type*, *hosts=None*, *profile=None*)

Retrieve mapping definition of index or index/type

index Index for the mapping

doc_type Name of the document type

CLI example:

```
salt myminion elasticsearch.mapping_get testindex user
```

`salt.modules.elasticsearch.`**`node_info`**(*nodes=None, flat_settings=False, hosts=None, profile=None*)

New in version 2017.7.0.

Return Elasticsearch node information.

nodes List of cluster nodes (id or name) to display stats for. Use _local for connected node, empty for all

flat_settings Flatten settings keys

CLI example:

```
salt myminion elasticsearch.node_info flat_settings=True
```

`salt.modules.elasticsearch.`**`ping`**(*allow_failure=False, hosts=None, profile=None*)

New in version 2017.7.0.

Test connection to Elasticsearch instance. This method does not fail if not explicitly specified.

allow_failure Throw exception if ping fails

CLI example:

```
salt myminion elasticsearch.ping allow_failure=True
salt myminion elasticsearch.ping profile=elasticsearch-extra
```

`salt.modules.elasticsearch.`**`pipeline_create`**(*id, body, hosts=None, profile=None*)

New in version 2017.7.0.

Create Ingest pipeline by supplied definition. Available since Elasticsearch 5.0.

id Pipeline id

body Pipeline definition as specified in https://www.elastic.co/guide/en/elasticsearch/reference/master/pipeline.html

CLI example:

```
salt myminion elasticsearch.pipeline_create mypipeline '{"description": "my␣
↪custom pipeline", "processors": [{"set" : {"field": "collector_timestamp_millis
↪", "value": "{{_ingest.timestamp}}"}}]}'
```

`salt.modules.elasticsearch.`**`pipeline_delete`**(*id, hosts=None, profile=None*)

New in version 2017.7.0.

Delete Ingest pipeline. Available since Elasticsearch 5.0.

id Pipeline id

CLI example:

```
salt myminion elasticsearch.pipeline_delete mypipeline
```

`salt.modules.elasticsearch.`**`pipeline_get`**(*id, hosts=None, profile=None*)

New in version 2017.7.0.

Retrieve Ingest pipeline definition. Available since Elasticsearch 5.0.

id Pipeline id

CLI example:

```
salt myminion elasticsearch.pipeline_get mypipeline
```

`salt.modules.elasticsearch.`**`pipeline_simulate`**(*id, body, verbose=False, hosts=None, profile=None*)

New in version 2017.7.0.

Simulate existing Ingest pipeline on provided data. Available since Elasticsearch 5.0.

id Pipeline id

body Pipeline definition as specified in https://www.elastic.co/guide/en/elasticsearch/reference/master/pipeline.html

verbose Specify if the output should be more verbose

CLI example:

```
salt myminion elasticsearch.pipeline_simulate mypipeline '{"docs":[{"_index":
↪"index","_type":"type","_id":"id","_source":{"foo":"bar"}},{"_index":"index","_
↪type":"type","_id":"id","_source":{"foo":"rab"}}]}' verbose=True
```

`salt.modules.elasticsearch.`**`repository_create`**(*name, body, hosts=None, profile=None*)

New in version 2017.7.0.

Create repository for storing snapshots. Note that shared repository paths have to be specified in path.repo Elasticsearch configuration option.

name Repository name

body Repository definition as in https://www.elastic.co/guide/en/elasticsearch/reference/current/modules-snapshots.html

CLI example:

```
salt myminion elasticsearch.repository_create testrepo '{"type":"fs","settings":{
↪"location":"/tmp/test","compress":true}}'
```

`salt.modules.elasticsearch.`**`repository_delete`**(*name, hosts=None, profile=None*)

New in version 2017.7.0.

Delete existing repository.

name Repository name

CLI example:

```
salt myminion elasticsearch.repository_delete testrepo
```

`salt.modules.elasticsearch.`**`repository_get`**(*name, local=False, hosts=None, profile=None*)

New in version 2017.7.0.

Get existing repository details.

name Repository name

local Retrieve only local information, default is false

CLI example:

```
salt myminion elasticsearch.repository_get testrepo
```

`salt.modules.elasticsearch.`**`repository_verify`**(*name*, *hosts=None*, *profile=None*)

New in version 2017.7.0.

Obtain list of cluster nodes which successfully verified this repository.

name Repository name

CLI example:

```
salt myminion elasticsearch.repository_verify testrepo
```

`salt.modules.elasticsearch.`**`search_template_create`**(*id*, *body*, *hosts=None*, *profile=None*)

New in version 2017.7.0.

Create search template by supplied definition

id Template ID

body Search template definition

CLI example:

```
salt myminion elasticsearch.search_template_create mytemplate '{"template":{"query
↪":{"match":{"title":"{{query_string}}"}}}}'
```

`salt.modules.elasticsearch.`**`search_template_delete`**(*id*, *hosts=None*, *profile=None*)

New in version 2017.7.0.

Delete existing search template definition.

id Template ID

CLI example:

```
salt myminion elasticsearch.search_template_delete mytemplate
```

`salt.modules.elasticsearch.`**`search_template_get`**(*id*, *hosts=None*, *profile=None*)

New in version 2017.7.0.

Obtain existing search template definition.

id Template ID

CLI example:

```
salt myminion elasticsearch.search_template_get mytemplate
```

`salt.modules.elasticsearch.`**`snapshot_create`**(*repository*, *snapshot*, *body=None*, *hosts=None*, *profile=None*)

New in version 2017.7.0.

Create snapshot in specified repository by supplied definition.

repository Repository name

snapshot Snapshot name

body Snapshot definition as in https://www.elastic.co/guide/en/elasticsearch/reference/current/modules-snapshots.html

CLI example:

```
salt myminion elasticsearch.snapshot_create testrepo testsnapshot '{"indices":
↪"index_1,index_2","ignore_unavailable":true,"include_global_state":false}'
```

`salt.modules.elasticsearch.`**`snapshot_delete`**`(`*repository, snapshot, hosts=None, profile=None*`)`

New in version 2017.7.0.

Delete snapshot from specified repository.

repository Repository name

snapshot Snapshot name

CLI example:

```
salt myminion elasticsearch.snapshot_delete testrepo testsnapshot
```

`salt.modules.elasticsearch.`**`snapshot_get`**`(`*repository, snapshot, ignore_unavailable=False, hosts=None, profile=None*`)`

New in version 2017.7.0.

Obtain snapshot residing in specified repository.

repository Repository name

snapshot Snapshot name, use _all to obtain all snapshots in specified repository

ignore_unavailable Ignore unavailable snapshots

CLI example:

```
salt myminion elasticsearch.snapshot_get testrepo testsnapshot
```

`salt.modules.elasticsearch.`**`snapshot_restore`**`(`*repository, snapshot, body=None, hosts=None, profile=None*`)`

New in version 2017.7.0.

Restore existing snapshot in specified repository by supplied definition.

repository Repository name

snapshot Snapshot name

body Restore definition as in https://www.elastic.co/guide/en/elasticsearch/reference/current/modules-snapshots.html

CLI example:

```
salt myminion elasticsearch.snapshot_restore testrepo testsnapshot '{"indices":
↪"index_1,index_2","ignore_unavailable":true,"include_global_state":true}'
```

`salt.modules.elasticsearch.`**`snapshot_status`**`(`*repository=None, snapshot=None, ignore_unavailable=False, hosts=None, profile=None*`)`

New in version 2017.7.0.

Obtain status of all currently running snapshots.

repository Particular repository to look for snapshots

snapshot Snapshot name

ignore_unavailable Ignore unavailable snapshots

CLI example:

```
salt myminion elasticsearch.snapshot_status ignore_unavailable=True
```

21.9.120 salt.modules.environ

Support for getting and setting the environment variables of the current salt process.

`salt.modules.environ.`**`get`**`(`*key, default=''*`)`

Get a single salt process environment variable.

key String used as the key for environment lookup.

default If the key is not found in the environment, return this value. Default: ''

CLI Example:

```
salt '*' environ.get foo
salt '*' environ.get baz default=False
```

`salt.modules.environ.`**`has_value`**`(`*key, value=None*`)`

Determine whether the key exists in the current salt process environment dictionary. Optionally compare the current value of the environment against the supplied value string.

key Must be a string. Used as key for environment lookup.

value: Optional. If key exists in the environment, compare the current value with this value. Return True if they are equal.

CLI Example:

```
salt '*' environ.has_value foo
```

`salt.modules.environ.`**`item`**`(`*keys, default=''*`)`

Get one or more salt process environment variables. Returns a dict.

keys Either a string or a list of strings that will be used as the keys for environment lookup.

default If the key is not found in the environment, return this value. Default: ''

CLI Example:

```
salt '*' environ.item foo
salt '*' environ.item '[foo, baz]' default=None
```

`salt.modules.environ.`**`items`**`()`

Return a dict of the entire environment set for the salt process

CLI Example:

```
salt '*' environ.items
```

`salt.modules.environ.`**`setenv`**`(`*environ, false_unsets=False, clear_all=False, update_minion=False, permanent=False*`)`

Set multiple salt process environment variables from a dict. Returns a dict.

environ Must be a dict. The top-level keys of the dict are the names of the environment variables to set. Each key's value must be a string or False. Refer to the 'false_unsets' parameter for behavior when a value set to False.

false_unsets If a key's value is False and false_unsets is True, then the key will be removed from the salt processes environment dict entirely. If a key's value is False and false_unsets is not True, then the key's value will be set to an empty string. Default: False

clear_all USE WITH CAUTION! This option can unset environment variables needed for salt to function properly. If clear_all is True, then any environment variables not defined in the environ dict will be deleted. Default: False

update_minion If True, apply these environ changes to the main salt-minion process. If False, the environ changes will only affect the current salt subprocess. Default: False

permanent On Windows minions this will set the environment variable in the registry so that it is always added as an environment variable when applications open. If you want to set the variable to HKLM instead of HKCU just pass in "HKLM" for this parameter. On all other minion types this will be ignored. Note: This will only take affect on applications opened after this has been set.

CLI Example:

```
salt '*' environ.setenv '{"foo": "bar", "baz": "quux"}'
salt '*' environ.setenv '{"a": "b", "c": False}' false_unsets=True
```

`salt.modules.environ.`**`setval`**(*key*, *val*, *false_unsets=False*, *permanent=False*)

Set a single salt process environment variable. Returns True on success.

key The environment key to set. Must be a string.

val The value to set. Must be a string or False. Refer to the 'false_unsets' parameter for behavior when set to False.

false_unsets If val is False and false_unsets is True, then the key will be removed from the salt processes environment dict entirely. If val is False and false_unsets is not True, then the key's value will be set to an empty string. Default: False.

permanent On Windows minions this will set the environment variable in the registry so that it is always added as an environment variable when applications open. If you want to set the variable to HKLM instead of HKCU just pass in "HKLM" for this parameter. On all other minion types this will be ignored. Note: This will only take affect on applications opened after this has been set.

CLI Example:

```
salt '*' environ.setval foo bar
salt '*' environ.setval baz val=False false_unsets=True
salt '*' environ.setval baz bar permanent=True
salt '*' environ.setval baz bar permanent=HKLM
```

21.9.121 salt.modules.eselect

Support for eselect, Gentoo's configuration and management tool.

`salt.modules.eselect.`**`exec_action`**(*module*, *action*, *module_parameter=None*, *action_parameter=None*, *state_only=False*)

Execute an arbitrary action on a module.

module name of the module to be executed

action name of the module's action to be run

module_parameter additional params passed to the defined module

action_parameter additional params passed to the defined action

state_only don't return any output but only the success/failure of the operation

CLI Example (updating the `php` implementation used for `apache2`):

```
salt '*' eselect.exec_action php update action_parameter='apache2'
```

`salt.modules.eselect.`**`get_current_target`**(*module*, *module_parameter=None*, *action_parameter=None*)

Get the currently selected target for the given module.

module name of the module to be queried for its current target

module_parameter additional params passed to the defined module

action_parameter additional params passed to the 'show' action

CLI Example (current target of system-wide `java-vm`):

```
salt '*' eselect.get_current_target java-vm action_parameter='system'
```

CLI Example (current target of `kernel` symlink):

```
salt '*' eselect.get_current_target kernel
```

`salt.modules.eselect.`**`get_modules`**()

List available `eselect` modules.

CLI Example:

```
salt '*' eselect.get_modules
```

`salt.modules.eselect.`**`get_target_list`**(*module*, *action_parameter=None*)

List available targets for the given module.

module name of the module to be queried for its targets

action_parameter additional params passed to the defined action

New in version 2016.11.0.

CLI Example:

```
salt '*' eselect.get_target_list kernel
```

`salt.modules.eselect.`**`set_target`**(*module*, *target*, *module_parameter=None*, *action_parameter=None*)

Set the target for the given module. Target can be specified by index or name.

module name of the module for which a target should be set

target name of the target to be set for this module

module_parameter additional params passed to the defined module

action_parameter additional params passed to the defined action

CLI Example (setting target of system-wide `java-vm`):

```
salt '*' eselect.set_target java-vm icedtea-bin-7 action_parameter='system'
```

CLI Example (setting target of `kernel` symlink):

```
salt '*' eselect.set_target kernel linux-3.17.5-gentoo
```

21.9.122 salt.modules.esxi

Glues the VMware vSphere Execution Module to the VMware ESXi Proxy Minions to the *esxi proxymodule*.

New in version 2015.8.4.

Depends: *vSphere Remote Execution Module (salt.modules.vsphere)*

For documentation on commands that you can direct to an ESXi host via proxy, look in the documentation for *salt.modules.vsphere*.

This execution module calls through to a function in the ESXi proxy module called `ch_config`, which looks up the function passed in the `command` parameter in *salt.modules.vsphere* and calls it.

To execute commands with an ESXi Proxy Minion using the vSphere Execution Module, use the `esxi.cmd <vsphere-function-name>` syntax. Both args and kwargs needed for various vsphere execution module functions must be passed through in a kwarg- type manor.

```
salt 'esxi-proxy' esxi.cmd system_info
salt 'exsi-proxy' esxi.cmd get_service_policy service_name='ssh'
```

`salt.modules.esxi.`**`cmd`**`(`*command*, **args*, ***kwargs*`)`

`salt.modules.esxi.`**`get_details`**`()`

21.9.123 salt.modules.etcd_mod

Execution module to work with etcd

depends

- python-etcd

Configuration

To work with an etcd server you must configure an etcd profile. The etcd config can be set in either the Salt Minion configuration file or in pillar:

```
my_etd_config:
  etcd.host: 127.0.0.1
  etcd.port: 4001
```

It is technically possible to configure etcd without using a profile, but this is not considered to be a best practice, especially when multiple etcd servers or clusters are available.

```
etcd.host: 127.0.0.1
etcd.port: 4001
```

Note: The etcd configuration can also be set in the Salt Master config file, but in order to use any etcd configurations defined in the Salt Master config, the *pillar_opts* must be set to `True`.

Be aware that setting `pillar_opts` to `True` has security implications as this makes all master configuration settings available in all minion's pillars.

`salt.modules.etcd_mod.`**`get_`**(*key*, *recurse=False*, *profile=None*, ***kwargs*)

New in version 2014.7.0.

Get a value from etcd, by direct path. Returns None on failure.

CLI Examples:

```
salt myminion etcd.get /path/to/key
salt myminion etcd.get /path/to/key profile=my_etcd_config
salt myminion etcd.get /path/to/key recurse=True profile=my_etcd_config
salt myminion etcd.get /path/to/key host=127.0.0.1 port=2379
```

`salt.modules.etcd_mod.`**`ls_`**(*path='/'*, *profile=None*, ***kwargs*)

New in version 2014.7.0.

Return all keys and dirs inside a specific path. Returns an empty dict on failure.

CLI Example:

```
salt myminion etcd.ls /path/to/dir/
salt myminion etcd.ls /path/to/dir/ profile=my_etcd_config
salt myminion etcd.ls /path/to/dir/ host=127.0.0.1 port=2379
```

`salt.modules.etcd_mod.`**`rm_`**(*key*, *recurse=False*, *profile=None*, ***kwargs*)

New in version 2014.7.0.

Delete a key from etcd. Returns True if the key was deleted, False if it was not and None if there was a failure.

CLI Example:

```
salt myminion etcd.rm /path/to/key
salt myminion etcd.rm /path/to/key profile=my_etcd_config
salt myminion etcd.rm /path/to/key host=127.0.0.1 port=2379
salt myminion etcd.rm /path/to/dir recurse=True profile=my_etcd_config
```

`salt.modules.etcd_mod.`**`set_`**(*key*, *value*, *profile=None*, *ttl=None*, *directory=False*, ***kwargs*)

New in version 2014.7.0.

Set a key in etcd by direct path. Optionally, create a directory or set a TTL on the key. Returns None on failure.

CLI Example:

```
salt myminion etcd.set /path/to/key value
salt myminion etcd.set /path/to/key value profile=my_etcd_config
salt myminion etcd.set /path/to/key value host=127.0.0.1 port=2379
salt myminion etcd.set /path/to/dir '' directory=True
salt myminion etcd.set /path/to/key value ttl=5
```

`salt.modules.etcd_mod.`**`tree`**(*path='/'*, *profile=None*, ***kwargs*)

New in version 2014.7.0.

Recurse through etcd and return all values. Returns None on failure.

CLI Example:

```
salt myminion etcd.tree
salt myminion etcd.tree profile=my_etcd_config
salt myminion etcd.tree host=127.0.0.1 port=2379
salt myminion etcd.tree /path/to/keys profile=my_etcd_config
```

`salt.modules.etcd_mod.`**`update`**(*fields*, *path=''*, *profile=None*, ***kwargs*)

New in version 2016.3.0.

Sets a dictionary of values in one call. Useful for large updates in syndic environments. The dictionary can contain a mix of formats such as:

```
{
  '/some/example/key': 'bar',
  '/another/example/key': 'baz'
}
```

Or it may be a straight dictionary, which will be flattened to look like the above format:

```
{
    'some': {
        'example': {
            'key': 'bar'
        }
    },
    'another': {
        'example': {
            'key': 'baz'
        }
    }
}
```

You can even mix the two formats and it will be flattened to the first format. Leading and trailing '/' will be removed.

Empty directories can be created by setting the value of the key to an empty dictionary.

The 'path' parameter will optionally set the root of the path to use.

CLI Example:

```
salt myminion etcd.update "{'/path/to/key': 'baz', '/another/key': 'bar'}"
salt myminion etcd.update "{'/path/to/key': 'baz', '/another/key': 'bar'}"
↪profile=my_etcd_config
salt myminion etcd.update "{'/path/to/key': 'baz', '/another/key': 'bar'}"
↪host=127.0.0.1 port=2379
salt myminion etcd.update "{'/path/to/key': 'baz', '/another/key': 'bar'}" path='/
↪some/root'
```

`salt.modules.etcd_mod.`**`watch`**(*key*, *recurse=False*, *profile=None*, *timeout=0*, *index=None*, ***kwargs*)

New in version 2016.3.0.

Makes a best effort to watch for a key or tree change in etcd. Returns a dict containing the new key value (or None if the key was deleted), the modifiedIndex of the key, whether the key changed or not, the path to the key that changed and whether it is a directory or not.

If something catastrophic happens, returns {}

CLI Example:

```
salt myminion etcd.watch /path/to/key
salt myminion etcd.watch /path/to/key timeout=10
salt myminion etcd.watch /patch/to/key profile=my_etcd_config index=10
salt myminion etcd.watch /patch/to/key host=127.0.0.1 port=2379
```

21.9.124 salt.modules.ethtool module

Module for running ethtool command

New in version 2016.3.0.

codeauthor Krzysztof Pawlowski <msciciel@msciciel.eu>

maturity new

depends python-ethtool

platform linux

`salt.modules.ethtool.`**`set_coalesce`**(*devname*, ***kwargs*)

Changes the coalescing settings of the specified network device

CLI Example:

```
salt '*' ethtool.set_coalesce <devname> [adaptive_rx=on|off] [adaptive_tx=on|off]↩
↪[rx_usecs=N] [rx_frames=N]
    [rx_usecs_irq=N] [rx_frames_irq=N] [tx_usecs=N] [tx_frames=N] [tx_usecs_
↪irq=N] [tx_frames_irq=N]
    [stats_block_usecs=N] [pkt_rate_low=N] [rx_usecs_low=N] [rx_frames_low=N] [tx_
↪usecs_low=N] [tx_frames_low=N]
    [pkt_rate_high=N] [rx_usecs_high=N] [rx_frames_high=N] [tx_usecs_high=N] [tx_
↪frames_high=N]
    [sample_interval=N]
```

`salt.modules.ethtool.`**`set_offload`**(*devname*, ***kwargs*)

Changes the offload parameters and other features of the specified network device

CLI Example:

```
salt '*' ethtool.set_offload <devname> tcp_segmentation_offload=on
```

`salt.modules.ethtool.`**`set_ring`**(*devname*, ***kwargs*)

Changes the rx/tx ring parameters of the specified network device

CLI Example:

```
salt '*' ethtool.set_ring <devname> [rx=N] [rx_mini=N] [rx_jumbo=N] [tx=N]
```

`salt.modules.ethtool.`**`show_coalesce`**(*devname*)

Queries the specified network device for coalescing information

CLI Example:

```
salt '*' ethtool.show_coalesce <devname>
```

`salt.modules.ethtool.`**`show_driver`**(*devname*)

Queries the specified network device for associated driver information

CLI Example:

```
salt '*' ethtool.show_driver <devname>
```

`salt.modules.ethtool.`**`show_offload`**(*devname*)

Queries the specified network device for the state of protocol offload and other features

CLI Example:

```
salt '*' ethtool.show_offload <devname>
```

`salt.modules.ethtool.`**`show_ring`**(*devname*)

Queries the specified network device for rx/tx ring parameter information

CLI Example:

```
salt '*' ethtool.show_ring <devname>
```

21.9.125 salt.modules.event

Use the *Salt Event System* to fire events from the master to the minion and vice-versa.

`salt.modules.event.`**`fire`**`(`*data, tag*`)`

Fire an event on the local minion event bus. Data must be formed as a dict.

CLI Example:

```
salt '*' event.fire '{"data":"my event data"}' 'tag'
```

`salt.modules.event.`**`fire_master`**`(`*data, tag, preload=None*`)`

Fire an event off up to the master server

CLI Example:

```
salt '*' event.fire_master '{"data":"my event data"}' 'tag'
```

`salt.modules.event.`**`send`**`(`*tag, data=None, preload=None, with_env=False, with_grains=False, with_pillar=False, with_env_opts=False, **kwargs*`)`

Send an event to the Salt Master

New in version 2014.7.0.

Parameters

- **`tag`** – A tag to give the event. Use slashes to create a namespace for related events. E.g., `myco/build/buildserver1/start`, `myco/build/buildserver1/success`, `myco/build/buildserver1/failure`.
- **`data`** – A dictionary of data to send in the event. This is free-form. Send any data points that are needed for whoever is consuming the event. Arguments on the CLI are interpreted as YAML so complex data structures are possible.
- **`with_env`** (Specify `True` to include all environment variables, or specify a list of strings of variable names to include.) – Include environment variables from the current shell environment in the event data as `environ`.. This is a short-hand for working with systems that seed the environment with relevant data such as Jenkins.
- **`with_grains`** (Specify `True` to include all grains, or specify a list of strings of grain names to include.) – Include grains from the current minion in the event data as `grains`.
- **`with_pillar`** (Specify `True` to include all Pillar values, or specify a list of strings of Pillar keys to include. It is a best-practice to only specify a relevant subset of Pillar data.) – Include Pillar values from the current minion in the event data as `pillar`. Remember Pillar data is often sensitive data so be careful. This is useful for passing ephemeral Pillar values through an event. Such as passing the `pillar={}` kwarg in *`state.sls`* from the Master, through an event on the Minion, then back to the Master.
- **`with_env_opts`** (Specify `True` to include `saltenv` and `pillarenv` values or `False` to omit them.) – Include `saltenv` and `pillarenv` set on minion at the moment when event is send into event data.
- **`kwargs`** – Any additional keyword arguments passed to this function will be interpreted as key-value pairs and included in the event data. This provides a convenient alternative to YAML for simple values.

CLI Example:

```
salt-call event.send myco/mytag foo=Foo bar=Bar
salt-call event.send 'myco/mytag' '{foo: Foo, bar: Bar}'
```

A convenient way to allow Jenkins to execute `salt-call` is via sudo. The following rule in sudoers will allow the `jenkins` user to run only the following command.

`/etc/sudoers` (allow preserving the environment):

```
jenkins ALL=(ALL) NOPASSWD:SETENV: /usr/bin/salt-call event.send*
```

Call Jenkins via sudo (preserve the environment):

```
sudo -E salt-call event.send myco/jenkins/build/success with_env=[BUILD_ID, BUILD_
↪URL, GIT_BRANCH, GIT_COMMIT]
```

21.9.126 salt.modules.extfs

Module for managing ext2/3/4 file systems

`salt.modules.extfs.`**`attributes`**`(`*device, args=None*`)`

Return attributes from dumpe2fs for a specified device

CLI Example:

```
salt '*' extfs.attributes /dev/sda1
```

`salt.modules.extfs.`**`blocks`**`(`*device, args=None*`)`

Return block and inode info from dumpe2fs for a specified device

CLI Example:

```
salt '*' extfs.blocks /dev/sda1
```

`salt.modules.extfs.`**`dump`**`(`*device, args=None*`)`

Return all contents of dumpe2fs for a specified device

CLI Example:

```
salt '*' extfs.dump /dev/sda1
```

`salt.modules.extfs.`**`mkfs`**`(`*device, fs_type, **kwargs*`)`

Create a file system on the specified device

CLI Example:

```
salt '*' extfs.mkfs /dev/sda1 fs_type=ext4 opts='acl,noexec'
```

Valid options are:

- **block_size**: 1024, 2048 or 4096
- **check**: check for bad blocks
- **direct**: use direct IO
- **ext_opts**: extended file system options (comma-separated)
- **fragment_size**: size of fragments

- **force**: setting force to True will cause mke2fs to specify the -F option twice (it is already set once); this is truly dangerous
- **blocks_per_group**: number of blocks in a block group
- **number_of_groups**: ext4 option for a virtual block group
- **bytes_per_inode**: set the bytes/inode ratio
- **inode_size**: size of the inode
- **journal**: set to True to create a journal (default on ext3/4)
- **journal_opts**: options for the fs journal (comma separated)
- **blocks_file**: read bad blocks from file
- **label**: label to apply to the file system
- **reserved**: percentage of blocks reserved for super-user
- **last_dir**: last mounted directory
- **test**: set to True to not actually create the file system (mke2fs -n)
- **number_of_inodes**: override default number of inodes
- **creator_os**: override "creator operating system" field
- **opts**: mount options (comma separated)
- **revision**: set the filesystem revision (default 1)
- **super**: write superblock and group descriptors only
- **fs_type**: set the filesystem type (REQUIRED)
- **usage_type**: how the filesystem is going to be used
- **uuid**: set the UUID for the file system

See the `mke2fs(8)` manpage for a more complete description of these options.

`salt.modules.extfs.`**`tune`**`(`*device*, ***kwargs*`)`

Set attributes for the specified device (using tune2fs)

CLI Example:

```
salt '*' extfs.tune /dev/sda1 force=True label=wildstallyns opts='acl,noexec'
```

Valid options are:

- **max**: max mount count
- **count**: mount count
- **error**: error behavior
- **extended_opts**: extended options (comma separated)
- **force**: force, even if there are errors (set to True)
- **group**: group name or gid that can use the reserved blocks
- **interval**: interval between checks
- **journal**: set to True to create a journal (default on ext3/4)
- **journal_opts**: options for the fs journal (comma separated)

- **label**: label to apply to the file system
- **reserved**: percentage of blocks reserved for super-user
- **last_dir**: last mounted directory
- **opts**: mount options (comma separated)
- **feature**: set or clear a feature (comma separated)
- **mmp_check**: mmp check interval
- **reserved**: reserved blocks count
- **quota_opts**: quota options (comma separated)
- **time**: time last checked
- **user**: user or uid who can use the reserved blocks
- **uuid**: set the UUID for the file system

See the `mke2fs(8)` manpage for a more complete description of these options.

21.9.127 salt.modules.file

Manage information about regular files, directories, and special files on the minion, set/read user, group, mode, and data

class `salt.modules.file.`**`AttrChanges`**`(`*added*, *removed*`)`

`added`
Alias for field number 0

`removed`
Alias for field number 1

`salt.modules.file.`**`access`**`(`*path*, *mode*`)`
New in version 2014.1.0.

Test whether the Salt process has the specified access to the file. One of the following modes must be specified:

CLI Example:

```
salt '*' file.access /path/to/file f
salt '*' file.access /path/to/file x
```

`salt.modules.file.`**`append`**`(`*path*, **args*, ***kwargs*`)`
New in version 0.9.5.

Append text to the end of a file

path path to file

***args** strings to append to file

CLI Example:

```
salt '*' file.append /etc/motd \
        "With all thine offerings thou shalt offer salt." \
        "Salt is what makes things taste bad when it isn't in them."
```

Attention

If you need to pass a string to append and that string contains an equal sign, you **must** include the argument name, args. For example:

```
salt '*' file.append /etc/motd args='cheese=spam'

salt '*' file.append /etc/motd args="['cheese=spam','spam=cheese']"
```

salt.modules.file.**apply_template_on_contents**(*contents, template, context, defaults, saltenv*)

Return the contents after applying the templating engine

contents template string

template template format

context Overrides default context variables passed to the template.

defaults Default context passed to the template.

CLI Example:

```
salt '*' file.apply_template_on_contents \
    contents='This is a {{ template }} string.' \
    template=jinja \
    "context={}" "defaults={'template': 'cool'}" \
    saltenv=base
```

salt.modules.file.**basename**(*path*)

Returns the final component of a pathname

New in version 2015.5.0.

This can be useful at the CLI but is frequently useful when scripting.

```
{%- set filename = salt['file.basename'](source_file) %}
```

CLI Example:

```
salt '*' file.basename 'test/test.config'
```

salt.modules.file.**blockreplace**(*path, marker_start='#– start managed zone –', marker_end='#– end managed zone –', content='', append_if_not_found=False, prepend_if_not_found=False, backup='.bak', dry_run=False, show_changes=True, append_newline=False*)

New in version 2014.1.0.

Replace content of a text block in a file, delimited by line markers

A block of content delimited by comments can help you manage several lines entries without worrying about old entries removal.

Note: This function will store two copies of the file in-memory (the original version and the edited version) in order to detect changes and only edit the targeted file if necessary.

path Filesystem path to the file to be edited

marker_start The line content identifying a line as the start of the content block. Note that the whole line containing this marker will be considered, so whitespace or extra content before or after the marker is included in final output

marker_end The line content identifying the end of the content block. As of versions 2017.7.5 and 2018.3.1, everything up to the text matching the marker will be replaced, so it's important to ensure that your marker includes the beginning of the text you wish to replace.

content The content to be used between the two lines identified by marker_start and marker_stop.

append_if_not_found: False If markers are not found and set to `True` then, the markers and content will be appended to the file.

prepend_if_not_found: False If markers are not found and set to `True` then, the markers and content will be prepended to the file.

backup The file extension to use for a backup of the file if any edit is made. Set to `False` to skip making a backup.

dry_run: False If `True`, do not make any edits to the file and simply return the changes that *would* be made.

show_changes: True Controls how changes are presented. If `True`, this function will return a unified diff of the changes made. If False, then it will return a boolean (`True` if any changes were made, otherwise `False`).

append_newline: False Controls whether or not a newline is appended to the content block. If the value of this argument is `True` then a newline will be added to the content block. If it is `False`, then a newline will *not* be added to the content block. If it is `None` then a newline will only be added to the content block if it does not already end in a newline.

New in version 2016.3.4.

Changed in version 2017.7.5,2018.3.1: New behavior added when value is `None`.

Changed in version 2019.2.0: The default value of this argument will change to `None` to match the behavior of the *`file.blockreplace state`*

CLI Example:

```
salt '*' file.blockreplace /etc/hosts '#-- start managed zone foobar : DO NOT
↪EDIT --' \
'#-- end managed zone foobar --' $'10.0.1.1 foo.foobar\n10.0.1.2 bar.foobar' True
```

`salt.modules.file.`**`chattr`**(**files*, ***kwargs*)

New in version 2018.3.0.

Change the attributes of files. This function accepts one or more files and the following options:

operator Can be wither `add` or `remove`. Determines whether attributes should be added or removed from files

attributes One or more of the following characters: `aAcCdDeijPsStTu`, representing attributes to add to/remove from files

version a version number to assign to the file(s)

flags One or more of the following characters: `RVf`, representing flags to assign to chattr (recurse, verbose, suppress most errors)

CLI Example:

```
salt '*' file.chattr foo1.txt foo2.txt operator=add attributes=ai
salt '*' file.chattr foo3.txt operator=remove attributes=i version=2
```

`salt.modules.file.`**`check_file_meta`**(*name, sfn, source, source_sum, user, group, mode, attrs, saltenv, contents=None*)

Check for the changes in the file metadata.

CLI Example:

```
salt '*' file.check_file_meta /etc/httpd/conf.d/httpd.conf salt://http/httpd.conf
↪'{hash_type: 'md5', 'hsum': <md5sum>}' root, root, '755' base
```

Note: Supported hash types include sha512, sha384, sha256, sha224, sha1, and md5.

---

name Path to file destination

sfn Template-processed source file contents

source URL to file source

source_sum File checksum information as a dictionary

```
{hash_type: md5, hsum: <md5sum>}
```

user Destination file user owner

group Destination file group owner

mode Destination file permissions mode

attrs Destination file attributes

New in version 2018.3.0.

saltenv Salt environment used to resolve source files

contents File contents

`salt.modules.file.`**`check_hash`**(*path, file_hash*)

Check if a file matches the given hash string

Returns `True` if the hash matches, otherwise `False`.

path Path to a file local to the minion.

hash The hash to check against the file specified in the `path` argument.

Changed in version 2016.11.4.

For this and newer versions the hash can be specified without an accompanying hash type (e.g. `e138491e9d5b97023cea823fe17bac22`), but for earlier releases it is necessary to also specify the hash type in the format `<hash_type>=<hash_value>` (e.g. `md5=e138491e9d5b97023cea823fe17bac22`).

CLI Example:

```
salt '*' file.check_hash /etc/fstab e138491e9d5b97023cea823fe17bac22
salt '*' file.check_hash /etc/fstab md5=e138491e9d5b97023cea823fe17bac22
```

`salt.modules.file.`**`check_managed`**(*name, source, source_hash, source_hash_name, user, group, mode, attrs, template, context, defaults, saltenv, contents=None, skip_verify=False, **kwargs*)

Check to see what changes need to be made for a file

CLI Example:

```
salt '*' file.check_managed /etc/httpd/conf.d/httpd.conf salt://http/httpd.conf '
↪{hash_type: 'md5', 'hsum': <md5sum>}' root, root, '755' jinja True None None
↪base
```

salt.modules.file.**check_managed_changes**(*name, source, source_hash, source_hash_name, user, group, mode, attrs, template, context, defaults, saltenv, contents=None, skip_verify=False, keep_mode=False, **kwargs*)

Return a dictionary of what changes need to be made for a file

CLI Example:

```
salt '*' file.check_managed_changes /etc/httpd/conf.d/httpd.conf salt://http/
↪httpd.conf '{hash_type: 'md5', 'hsum': <md5sum>}' root, root, '755' jinja True
↪None None base
```

salt.modules.file.**check_perms**(*name, ret, user, group, mode, attrs=None, follow_symlinks=False*)

Check the permissions on files, modify attributes and chown if needed. File attributes are only verified if lsattr(1) is installed.

CLI Example:

```
salt '*' file.check_perms /etc/sudoers '{}' root root 400 ai
```

Changed in version 2014.1.3: `follow_symlinks` option added

salt.modules.file.**chgrp**(*path, group*)

Change the group of a file

path path to the file or directory

group group owner

CLI Example:

```
salt '*' file.chgrp /etc/passwd root
```

salt.modules.file.**chown**(*path, user, group*)

Chown a file, pass the file the desired user and group

path path to the file or directory

user user owner

group group owner

CLI Example:

```
salt '*' file.chown /etc/passwd root root
```

salt.modules.file.**comment**(*path, regex, char='#', backup='.bak'*)

Deprecated since version 0.17.0: Use `replace()` instead.

Comment out specified lines in a file

path The full path to the file to be edited

regex A regular expression used to find the lines that are to be commented; this pattern will be wrapped in parenthesis and will move any preceding/trailing `^` or `$` characters outside the parenthesis (e.g., the pattern `^foo$` will be rewritten as `^(foo)$`)

char: # The character to be inserted at the beginning of a line in order to comment it out

backup: **`.bak`** The file will be backed up before edit with this file extension

> **Warning:** This backup will be overwritten each time `sed` / `comment` / `uncomment` is called. Meaning the backup will only be useful after the first invocation.

CLI Example:

```
salt '*' file.comment /etc/modules pcspkr
```

`salt.modules.file.`**`comment_line`**(*path*, *regex*, *char='#'*, *cmnt=True*, *backup='.bak'*)

Comment or Uncomment a line in a text file.

Parameters

- **`path`** – string The full path to the text file.
- **`regex`** – string A regex expression that begins with ^ that will find the line you wish to comment. Can be as simple as `^color =`
- **`char`** – string The character used to comment a line in the type of file you're referencing. Default is #
- **`cmnt`** – boolean True to comment the line. False to uncomment the line. Default is True.
- **`backup`** – string The file extension to give the backup file. Default is `.bak` Set to False/None to not keep a backup.

Returns boolean Returns True if successful, False if not

CLI Example:

The following example will comment out the `pcspkr` line in the `/etc/modules` file using the default `#` character and create a backup file named `modules.bak`

```
salt '*' file.comment_line '/etc/modules' '^pcspkr'
```

CLI Example:

The following example will uncomment the `log_level` setting in `minion` config file if it is set to either `warning`, `info`, or `debug` using the `#` character and create a backup file named `minion.bk`

```
salt '*' file.comment_line 'C:\salt\conf\minion' '^log_level: (warning|info|debug)
↪' '#' False '.bk'
```

`salt.modules.file.`**`contains`**(*path*, *text*)

Deprecated since version 0.17.0: Use `search()` instead.

Return `True` if the file at `path` contains `text`

CLI Example:

```
salt '*' file.contains /etc/crontab 'mymaintenance.sh'
```

`salt.modules.file.`**`contains_glob`**(*path*, *glob_expr*)

Deprecated since version 0.17.0: Use `search()` instead.

Return `True` if the given glob matches a string in the named file

CLI Example:

```
salt '*' file.contains_glob /etc/foobar '*cheese*'
```

`salt.modules.file.`**`contains_regex`**(*path*, *regex*, *lchar=''*)

Deprecated since version 0.17.0: Use *`search()`* instead.

Return True if the given regular expression matches on any line in the text of a given file.

If the lchar argument (leading char) is specified, it will strip *lchar* from the left side of each line before trying to match

CLI Example:

```
salt '*' file.contains_regex /etc/crontab
```

`salt.modules.file.`**`copy`**(*src*, *dst*, *recurse=False*, *remove_existing=False*)

Copy a file or directory from source to dst

In order to copy a directory, the recurse flag is required, and will by default overwrite files in the destination with the same path, and retain all other existing files. (similar to cp -r on unix)

remove_existing will remove all files in the target directory, and then copy files from the source.

Note: The copy function accepts paths that are local to the Salt minion. This function does not support salt://, http://, or the other additional file paths that are supported by *`states.file.managed`* and *`states.file.recurse`*.

CLI Example:

```
salt '*' file.copy /path/to/src /path/to/dst
salt '*' file.copy /path/to/src_dir /path/to/dst_dir recurse=True
salt '*' file.copy /path/to/src_dir /path/to/dst_dir recurse=True remove_
↪existing=True
```

`salt.modules.file.`**`delete_backup`**(*path*, *backup_id*)

New in version 0.17.0.

Delete a previous version of a file that was backed up using Salt's *file state backup* system.

path The path on the minion to check for backups

backup_id The numeric id for the backup you wish to delete, as found using *`file.list_backups`*

CLI Example:

```
salt '*' file.delete_backup /var/cache/salt/minion/file_backup/home/foo/bar/baz.
↪txt 0
```

`salt.modules.file.`**`directory_exists`**(*path*)

Tests to see if path is a valid directory. Returns True/False.

CLI Example:

```
salt '*' file.directory_exists /etc
```

`salt.modules.file.`**`dirname`**(*path*)

Returns the directory component of a pathname

New in version 2015.5.0.

This can be useful at the CLI but is frequently useful when scripting.

```
{%- from salt['file.dirname'](tpldir) + '/vars.jinja' import parent_vars %}
```

CLI Example:

```
salt '*' file.dirname 'test/path/filename.config'
```

`salt.modules.file.`**`diskusage`**(*path*)

Recursively calculate disk usage of path and return it in bytes

CLI Example:

```
salt '*' file.diskusage /path/to/check
```

`salt.modules.file.`**`extract_hash`**(*hash_fn*, *hash_type='sha256'*, *file_name=''*, *source=''*, *source_hash_name=None*)

Changed in version 2016.3.5: Prior to this version, only the `file_name` argument was considered for filename matches in the hash file. This would be problematic for cases in which the user was relying on a remote checksum file that they do not control, and they wished to use a different name for that file on the minion from the filename on the remote server (and in the checksum file). For example, managing `/tmp/myfile.tar.gz` when the remote file was at `https://mydomain.tld/different_name.tar.gz`. The *`file.managed`* state now also passes this function the source URI as well as the `source_hash_name` (if specified). In cases where `source_hash_name` is specified, it takes precedence over both the `file_name` and `source`. When it is not specified, `file_name` takes precedence over `source`. This allows for better capability for matching hashes.

Changed in version 2016.11.0: File name and source URI matches are no longer disregarded when `source_hash_name` is specified. They will be used as fallback matches if there is no match to the `source_hash_name` value.

This routine is called from the *`file.managed`* state to pull a hash from a remote file. Regular expressions are used line by line on the `source_hash` file, to find a potential candidate of the indicated hash type. This avoids many problems of arbitrary file layout rules. It specifically permits pulling hash codes from debian `*.dsc` files.

If no exact match of a hash and filename are found, then the first hash found (if any) will be returned. If no hashes at all are found, then `None` will be returned.

For example:

```
openerp_7.0-latest-1.tar.gz:
  file.managed:
    - name: /tmp/openerp_7.0-20121227-075624-1_all.deb
    - source: http://nightly.openerp.com/7.0/nightly/deb/openerp_7.0-20121227-
↪075624-1.tar.gz
    - source_hash: http://nightly.openerp.com/7.0/nightly/deb/openerp_7.0-
↪20121227-075624-1.dsc
```

CLI Example:

```
salt '*' file.extract_hash /path/to/hash/file sha512 /etc/foo
```

`salt.modules.file.`**`file_exists`**(*path*)

Tests to see if path is a valid file. Returns True/False.

CLI Example:

```
salt '*' file.file_exists /etc/passwd
```

`salt.modules.file.`**`find`**`(`*path*, **args*, ***kwargs*`)`

Approximate the Unix `find(1)` command and return a list of paths that meet the specified criteria.

The options include match criteria:

```
name    = path-glob                 # case sensitive
iname   = path-glob                 # case insensitive
regex   = path-regex                # case sensitive
iregex  = path-regex                # case insensitive
type    = file-types                # match any listed type
user    = users                     # match any listed user
group   = groups                    # match any listed group
size    = [+-]number[size-unit]     # default unit = byte
mtime   = interval                  # modified since date
grep    = regex                     # search file contents
```

and/or actions:

```
delete [= file-types]               # default type = 'f'
exec    = command [arg ...]         # where {} is replaced by pathname
print  [= print-opts]
```

and/or depth criteria:

```
maxdepth = maximum depth to transverse in path
mindepth = minimum depth to transverse before checking files or directories
```

The default action is `print=path`

`path-glob`:

```
*                 = match zero or more chars
?                 = match any char
[abc]             = match a, b, or c
[!abc] or [^abc] = match anything except a, b, and c
[x-y]             = match chars x through y
[!x-y] or [^x-y] = match anything except chars x through y
{a,b,c}           = match a or b or c
```

`path-regex`: a Python Regex (regular expression) pattern to match pathnames

`file-types`: a string of one or more of the following:

```
a: all file types
b: block device
c: character device
d: directory
p: FIFO (named pipe)
f: plain file
l: symlink
s: socket
```

`users`: a space and/or comma separated list of user names and/or uids

`groups`: a space and/or comma separated list of group names and/or gids

`size-unit`:

```
b: bytes
k: kilobytes
m: megabytes
g: gigabytes
t: terabytes
```

interval:

```
[<num>w] [<num>d] [<num>h] [<num>m] [<num>s]

where:
    w: week
    d: day
    h: hour
    m: minute
    s: second
```

print-opts: a comma and/or space separated list of one or more of the following:

```
group: group name
md5:   MD5 digest of file contents
mode:  file permissions (as integer)
mtime: last modification time (as time_t)
name:  file basename
path:  file absolute path
size:  file size in bytes
type:  file type
user:  user name
```

CLI Examples:

```
salt '*' file.find / type=f name=\*.bak size=+10m
salt '*' file.find /var mtime=+30d size=+10m print=path,size,mtime
salt '*' file.find /var/log name=\*.[0-9] mtime=+30d size=+10m delete
```

`salt.modules.file.`**`get_devmm`**(*name*)

Get major/minor info from a device

CLI Example:

```
salt '*' file.get_devmm /dev/chr
```

`salt.modules.file.`**`get_diff`**(*file1, file2, saltenv='base', show_filenames=True, show_changes=True, template=False, source_hash_file1=None, source_hash_file2=None*)

Return unified diff of two files

file1 The first file to feed into the diff utility

Changed in version 2018.3.0: Can now be either a local or remote file. In earlier releases, thuis had to be a file local to the minion.

file2 The second file to feed into the diff utility

Changed in version 2018.3.0: Can now be either a local or remote file. In earlier releases, this had to be a file on the salt fileserver (i.e. `salt://somefile.txt`)

show_filenames: True Set to `False` to hide the filenames in the top two lines of the diff.

show_changes: True If set to `False`, and there are differences, then instead of a diff a simple message stating that show_changes is set to `False` will be returned.

template: False Set to `True` if two templates are being compared. This is not useful except for within states, with the `obfuscate_templates` option set to `True`.

New in version 2018.3.0.

source_hash_file1 If `file1` is an http(s)/ftp URL and the file exists in the minion's file cache, this option can be passed to keep the minion from re-downloading the archive if the cached copy matches the specified hash.

New in version 2018.3.0.

source_hash_file2 If `file2` is an http(s)/ftp URL and the file exists in the minion's file cache, this option can be passed to keep the minion from re-downloading the archive if the cached copy matches the specified hash.

New in version 2018.3.0.

CLI Examples:

```
salt '*' file.get_diff /home/fred/.vimrc salt://users/fred/.vimrc
salt '*' file.get_diff /tmp/foo.txt /tmp/bar.txt
```

`salt.modules.file.`**`get_gid`**(*path*, *follow_symlinks=True*)

Return the id of the group that owns a given file

path file or directory of which to get the gid

follow_symlinks indicated if symlinks should be followed

CLI Example:

```
salt '*' file.get_gid /etc/passwd
```

Changed in version 0.16.4: `follow_symlinks` option added

`salt.modules.file.`**`get_group`**(*path*, *follow_symlinks=True*)

Return the group that owns a given file

path file or directory of which to get the group

follow_symlinks indicated if symlinks should be followed

CLI Example:

```
salt '*' file.get_group /etc/passwd
```

Changed in version 0.16.4: `follow_symlinks` option added

`salt.modules.file.`**`get_hash`**(*path*, *form='sha256'*, *chunk_size=65536*)

Get the hash sum of a file

This is better than `get_sum` for the following reasons:

- It does not read the entire file into memory.
- **It does not return a string on error. The returned value of** `get_sum` cannot really be trusted since it is vulnerable to collisions: `get_sum(..., 'xyz') == 'Hash xyz not supported'`

path path to the file or directory

form desired sum format

chunk_size amount to sum at once

CLI Example:

```
salt '*' file.get_hash /etc/shadow
```

`salt.modules.file.`**`get_managed`**`(`*name, template, source, source_hash, source_hash_name, user, group, mode, attrs, saltenv, context, defaults, skip_verify=False, **kwargs*`)`

Return the managed file data for file.managed

name location where the file lives on the server

template template format

source managed source file

source_hash hash of the source file

source_hash_name When `source_hash` refers to a remote file, this specifies the filename to look for in that file.

New in version 2016.3.5.

user Owner of file

group Group owner of file

mode Permissions of file

attrs Attributes of file

New in version 2018.3.0.

context Variables to add to the template context

defaults Default values of for context_dict

skip_verify If `True`, hash verification of remote file sources (`http://`, `https://`, `ftp://`) will be skipped, and the `source_hash` argument will be ignored.

New in version 2016.3.0.

CLI Example:

```
salt '*' file.get_managed /etc/httpd/conf.d/httpd.conf jinja salt://http/httpd.
↪conf '{hash_type: 'md5', 'hsum': <md5sum>}' None root root '755' base None None
```

`salt.modules.file.`**`get_mode`**`(`*path, follow_symlinks=True*`)`

Return the mode of a file

path file or directory of which to get the mode

follow_symlinks indicated if symlinks should be followed

CLI Example:

```
salt '*' file.get_mode /etc/passwd
```

Changed in version 2014.1.0: `follow_symlinks` option added

`salt.modules.file.`**`get_selinux_context`**`(`*path*`)`

Get an SELinux context from a given path

CLI Example:

```
salt '*' file.get_selinux_context /etc/hosts
```

`salt.modules.file.`**`get_source_sum`**(*file_name='', source='', source_hash=None, source_hash_name=None, saltenv='base'*)

New in version 2016.11.0.

Used by *`file.get_managed`* to obtain the hash and hash type from the parameters specified below.

file_name Optional file name being managed, for matching with *`file.extract_hash`*.

source Source file, as used in *`file`* and other states. If `source_hash` refers to a file containing hashes, then this filename will be used to match a filename in that file. If the `source_hash` is a hash expression, then this argument will be ignored.

source_hash Hash file/expression, as used in *`file`* and other states. If this value refers to a remote URL or absolute path to a local file, it will be cached and *`file.extract_hash`* will be used to obtain a hash from it.

source_hash_name Specific file name to look for when `source_hash` refers to a remote file, used to disambiguate ambiguous matches.

saltenv: base Salt fileserver environment from which to retrieve the source_hash. This value will only be used when `source_hash` refers to a file on the Salt fileserver (i.e. one beginning with `salt://`).

CLI Example:

```
salt '*' file.get_source_sum /tmp/foo.tar.gz source=http://mydomain.tld/foo.tar.
↪gz source_hash=499ae16dcae71eeb7c3a30c75ea7a1a6
salt '*' file.get_source_sum /tmp/foo.tar.gz source=http://mydomain.tld/foo.tar.
↪gz source_hash=https://mydomain.tld/hashes.md5
salt '*' file.get_source_sum /tmp/foo.tar.gz source=http://mydomain.tld/foo.tar.
↪gz source_hash=https://mydomain.tld/hashes.md5 source_hash_name=./dir2/foo.tar.
↪gz
```

`salt.modules.file.`**`get_sum`**(*path, form='sha256'*)

Return the checksum for the given file. The following checksum algorithms are supported:

- md5
- sha1
- sha224
- sha256 **(default)**
- sha384
- sha512

path path to the file or directory

form desired sum format

CLI Example:

```
salt '*' file.get_sum /etc/passwd sha512
```

`salt.modules.file.`**`get_uid`**(*path, follow_symlinks=True*)

Return the id of the user that owns a given file

path file or directory of which to get the uid

follow_symlinks indicated if symlinks should be followed

CLI Example:

```
salt '*' file.get_uid /etc/passwd
```

Changed in version 0.16.4: `follow_symlinks` option added

`salt.modules.file.`**`get_user`**`(`*path*, *follow_symlinks=True*`)`
Return the user that owns a given file

path file or directory of which to get the user

follow_symlinks indicated if symlinks should be followed

CLI Example:

```
salt '*' file.get_user /etc/passwd
```

Changed in version 0.16.4: `follow_symlinks` option added

`salt.modules.file.`**`gid_to_group`**`(`*gid*`)`
Convert the group id to the group name on this system

gid gid to convert to a group name

CLI Example:

```
salt '*' file.gid_to_group 0
```

`salt.modules.file.`**`grep`**`(`*path*, *pattern*, **opts*`)`
Grep for a string in the specified file

Note: This function's return value is slated for refinement in future versions of Salt

path Path to the file to be searched

> **Note:** Globbing is supported (i.e. `/var/log/foo/*.log`, but if globbing is being used then the path should be quoted to keep the shell from attempting to expand the glob expression.

pattern Pattern to match. For example: `test`, or `a[0-5]`

opts Additional command-line flags to pass to the grep command. For example: `-v`, or `-i -B2`

> **Note:** The options should come after a double-dash (as shown in the examples below) to keep Salt's own argument parser from interpreting them.

CLI Example:

```
salt '*' file.grep /etc/passwd nobody
salt '*' file.grep /etc/sysconfig/network-scripts/ifcfg-eth0 ipaddr -- -i
salt '*' file.grep /etc/sysconfig/network-scripts/ifcfg-eth0 ipaddr -- -i -B2
salt '*' file.grep "/etc/sysconfig/network-scripts/*" ipaddr -- -i -l
```

`salt.modules.file.`**`group_to_gid`**`(`*group*`)`
Convert the group to the gid on this system

group group to convert to its gid

CLI Example:

```
salt '*' file.group_to_gid root
```

`salt.modules.file.`**`is_blkdev`**`(`*name*`)`

Check if a file exists and is a block device.

CLI Example:

```
salt '*' file.is_blkdev /dev/blk
```

`salt.modules.file.`**`is_chrdev`**`(`*name*`)`

Check if a file exists and is a character device.

CLI Example:

```
salt '*' file.is_chrdev /dev/chr
```

`salt.modules.file.`**`is_fifo`**`(`*name*`)`

Check if a file exists and is a FIFO.

CLI Example:

```
salt '*' file.is_fifo /dev/fifo
```

`salt.modules.file.`**`is_hardlink`**`(`*path*`)`

Check if the path is a hard link by verifying that the number of links is larger than 1

CLI Example:

```
salt '*' file.is_hardlink /path/to/link
```

`salt.modules.file.`**`is_link`**`(`*path*`)`

Check if the path is a symbolic link

CLI Example:

```
salt '*' file.is_link /path/to/link
```

`salt.modules.file.`**`join`**`(`**args*`)`

Return a normalized file system path for the underlying OS

New in version 2014.7.0.

This can be useful at the CLI but is frequently useful when scripting combining path variables:

```
{% set www_root = '/var' %}
{% set app_dir = 'myapp' %}

myapp_config:
  file:
    - managed
    - name: {{ salt['file.join'](www_root, app_dir, 'config.yaml') }}
```

CLI Example:

```
salt '*' file.join '/' 'usr' 'local' 'bin'
```

`salt.modules.file.`**`lchown`**`(`*path, user, group*`)`

Chown a file, pass the file the desired user and group without following symlinks.

path path to the file or directory

user user owner

group group owner

CLI Example:

```
salt '*' file.chown /etc/passwd root root
```

`salt.modules.file.`**`line`**(*path*, *content=None*, *match=None*, *mode=None*, *location=None*, *before=None*, *after=None*, *show_changes=True*, *backup=False*, *quiet=False*, *indent=True*)

New in version 2015.8.0.

Edit a line in the configuration file. The `path` and `content` arguments are required, as well as passing in one of the `mode` options.

path Filesystem path to the file to be edited.

content Content of the line. Allowed to be empty if mode=delete.

match Match the target line for an action by a fragment of a string or regular expression.

If neither `before` nor `after` are provided, and `match` is also `None`, match becomes the `content` value.

mode Defines how to edit a line. One of the following options is required:

- **ensure** If line does not exist, it will be added. This is based on the `content` argument.
- **replace** If line already exists, it will be replaced.
- **delete** Delete the line, once found.
- **insert** Insert a line.

Note: If `mode=insert` is used, at least one of the following options must also be defined: `location`, `before`, or `after`. If `location` is used, it takes precedence over the other two options.

location Defines where to place content in the line. Note this option is only used when `mode=insert` is specified. If a location is passed in, it takes precedence over both the `before` and `after` kwargs. Valid locations are:

- **start** Place the content at the beginning of the file.
- **end** Place the content at the end of the file.

before Regular expression or an exact case-sensitive fragment of the string. This option is only used when either the `ensure` or `insert` mode is defined.

after Regular expression or an exact case-sensitive fragment of the string. This option is only used when either the `ensure` or `insert` mode is defined.

show_changes Output a unified diff of the old file and the new file. If `False` return a boolean if any changes were made. Default is `True`

Note: Using this option will store two copies of the file in-memory (the original version and the edited version) in order to generate the diff.

backup Create a backup of the original file with the extension: "Year-Month-Day-Hour-Minutes-Seconds".

quiet Do not raise any exceptions. E.g. ignore the fact that the file that is tried to be edited does not exist and nothing really happened.

indent Keep indentation with the previous line. This option is not considered when the `delete` mode is specified.

CLI Example:

```
salt '*' file.line /etc/nsswitch.conf "networks:        files dns" after="hosts:.
↪*?" mode='ensure'
```

Note: If an equal sign (=) appears in an argument to a Salt command, it is interpreted as a keyword argument in the format of `key=val`. That processing can be bypassed in order to pass an equal sign through to the remote shell command by manually specifying the kwarg:

```
salt '*' file.line /path/to/file content="CREATEMAIL_SPOOL=no" match="CREATE_MAIL_
↪SPOOL=yes" mode="replace"
```

`salt.modules.file.`**`link`**(*src, path*)

New in version 2014.1.0.

Create a hard link to a file

CLI Example:

```
salt '*' file.link /path/to/file /path/to/link
```

`salt.modules.file.`**`list_backups`**(*path, limit=None*)

New in version 0.17.0.

Lists the previous versions of a file backed up using Salt's *file state backup* system.

path The path on the minion to check for backups

limit Limit the number of results to the most recent N backups

CLI Example:

```
salt '*' file.list_backups /foo/bar/baz.txt
```

`salt.modules.file.`**`list_backups_dir`**(*path, limit=None*)

Lists the previous versions of a directory backed up using Salt's *file state backup* system.

path The directory on the minion to check for backups

limit Limit the number of results to the most recent N backups

CLI Example:

```
salt '*' file.list_backups_dir /foo/bar/baz/
```

`salt.modules.file.`**`lsattr`**(*path*)

New in version 2018.3.0.

Changed in version 2018.3.1: If `lsattr` is not installed on the system, `None` is returned.

Changed in version 2018.3.4: If on `AIX`, `None` is returned even if in filesystem as lsattr on `AIX` is not the same thing as the linux version.

Obtain the modifiable attributes of the given file. If path is to a directory, an empty list is returned.

path path to file to obtain attributes of. File/directory must exist.

CLI Example:

```
salt '*' file.lsattr foo1.txt
```

`salt.modules.file.`**`lstat`**`(`*path*`)`

New in version 2014.1.0.

Returns the lstat attributes for the given file or dir. Does not support symbolic links.

CLI Example:

```
salt '*' file.lstat /path/to/file
```

`salt.modules.file.`**`makedirs_`**`(`*path*, *user=None*, *group=None*, *mode=None*`)`

Ensure that the directory containing this path is available.

Note: The path must end with a trailing slash otherwise the directory/directories will be created up to the parent directory. For example if path is `/opt/code`, then it would be treated as `/opt/` but if the path ends with a trailing slash like `/opt/code/`, then it would be treated as `/opt/code/`.

CLI Example:

```
salt '*' file.makedirs /opt/code/
```

`salt.modules.file.`**`makedirs_perms`**`(`*name*, *user=None*, *group=None*, *mode='0755'*`)`

Taken and modified from os.makedirs to set user, group and mode for each directory created.

CLI Example:

```
salt '*' file.makedirs_perms /opt/code
```

`salt.modules.file.`**`manage_file`**`(`*name*, *sfn*, *ret*, *source*, *source_sum*, *user*, *group*, *mode*, *attrs*, *saltenv*, *backup*, *makedirs=False*, *template=None*, *show_changes=True*, *contents=None*, *dir_mode=None*, *follow_symlinks=True*, *skip_verify=False*, *keep_mode=False*, *encoding=None*, *encoding_errors='strict'*, ***kwargs*`)`

Checks the destination against what was retrieved with get_managed and makes the appropriate modifications (if necessary).

name location to place the file

sfn location of cached file on the minion

> This is the path to the file stored on the minion. This file is placed on the minion using cp.cache_file. If the hash sum of that file matches the source_sum, we do not transfer the file to the minion again.
>
> This file is then grabbed and if it has template set, it renders the file to be placed into the correct place on the system using salt.files.utils.copyfile()

ret The initial state return data structure. Pass in `None` to use the default structure.

source file reference on the master

source_sum sum hash for source

user user owner

group group owner

backup backup_mode

attrs attributes to be set on file: " means remove all of them

New in version 2018.3.0.

makedirs make directories if they do not exist

template format of templating

show_changes Include diff in state return

contents: contents to be placed in the file

dir_mode mode for directories created with makedirs

skip_verify: False If `True`, hash verification of remote file sources (`http://`, `https://`, `ftp://`) will be skipped, and the `source_hash` argument will be ignored.

New in version 2016.3.0.

keep_mode: False If `True`, and the `source` is a file from the Salt fileserver (or a local file on the minion), the mode of the destination file will be set to the mode of the source file.

Note: keep_mode does not work with salt-ssh.

As a consequence of how the files are transferred to the minion, and the inability to connect back to the master with salt-ssh, salt is unable to stat the file as it exists on the fileserver and thus cannot mirror the mode on the salt-ssh minion

---

encoding If specified, then the specified encoding will be used. Otherwise, the file will be encoded using the system locale (usually UTF-8). See https://docs.python.org/3/library/codecs.html#standard-encodings for the list of available encodings.

New in version 2017.7.0.

encoding_errors: 'strict' Default is `'strict'`. See https://docs.python.org/2/library/codecs.html#codec-base-classes for the error handling schemes.

New in version 2017.7.0.

CLI Example:

```
salt '*' file.manage_file /etc/httpd/conf.d/httpd.conf '' '{}' salt://http/httpd.
↪conf '{hash_type: 'md5', 'hsum': <md5sum>}' root root '755' '' base ''
```

Changed in version 2014.7.0: `follow_symlinks` option added

`salt.modules.file.`**`mkdir`**(*dir_path, user=None, group=None, mode=None*)

Ensure that a directory is available.

CLI Example:

```
salt '*' file.mkdir /opt/jetty/context
```

`salt.modules.file.`**`mknod`**(*name, ntype, major=0, minor=0, user=None, group=None, mode='0600'*)

New in version 0.17.0.

Create a block device, character device, or fifo pipe. Identical to the gnu mknod.

CLI Examples:

```
salt '*' file.mknod /dev/chr c 180 31
salt '*' file.mknod /dev/blk b 8 999
salt '*' file.nknod /dev/fifo p
```

`salt.modules.file.`**`mknod_blkdev`**(*name, major, minor, user=None, group=None, mode='0660'*)

New in version 0.17.0.

Create a block device.

CLI Example:

```
salt '*' file.mknod_blkdev /dev/blk 8 999
```

`salt.modules.file.`**`mknod_chrdev`**(*name, major, minor, user=None, group=None, mode='0660'*)

New in version 0.17.0.

Create a character device.

CLI Example:

```
salt '*' file.mknod_chrdev /dev/chr 180 31
```

`salt.modules.file.`**`mknod_fifo`**(*name, user=None, group=None, mode='0660'*)

New in version 0.17.0.

Create a FIFO pipe.

CLI Example:

```
salt '*' file.mknod_fifo /dev/fifo
```

`salt.modules.file.`**`move`**(*src, dst*)

Move a file or directory

CLI Example:

```
salt '*' file.move /path/to/src /path/to/dst
```

`salt.modules.file.`**`normpath`**(*path*)

Returns Normalize path, eliminating double slashes, etc.

New in version 2015.5.0.

This can be useful at the CLI but is frequently useful when scripting.

```
{%- from salt['file.normpath'](tpldir + '/../vars.jinja') import parent_vars %}
```

CLI Example:

```
salt '*' file.normpath 'a/b/c/..'
```

`salt.modules.file.`**`open_files`**(*by_pid=False*)

Return a list of all physical open files on the system.

CLI Examples:

```
salt '*' file.open_files
salt '*' file.open_files by_pid=True
```

`salt.modules.file.`**`pardir`**()

Return the relative parent directory path symbol for underlying OS

New in version 2014.7.0.

This can be useful when constructing Salt Formulas.

```
{% set pardir = salt['file.pardir']() %}
{% set final_path = salt['file.join']('subdir', pardir, 'confdir') %}
```

CLI Example:

```
salt '*' file.pardir
```

`salt.modules.file.`**`patch`**(*originalfile*, *patchfile*, *options=''*, *dry_run=False*)

New in version 0.10.4.

Apply a patch to a file or directory.

Equivalent to:

```
patch <options> -i <patchfile> <originalfile>
```

Or, when a directory is patched:

```
patch <options> -i <patchfile> -d <originalfile> -p0
```

originalfile The full path to the file or directory to be patched

patchfile A patch file to apply to `originalfile`

options Options to pass to patch.

CLI Example:

```
salt '*' file.patch /opt/file.txt /tmp/file.txt.patch
```

`salt.modules.file.`**`path_exists_glob`**(*path*)

Tests to see if path after expansion is a valid path (file or directory). Expansion allows usage of ? * and character ranges []. Tilde expansion is not supported. Returns True/False.

New in version 2014.7.0.

CLI Example:

```
salt '*' file.path_exists_glob /etc/pam*/pass*
```

`salt.modules.file.`**`prepend`**(*path*, **args*, ***kwargs*)

New in version 2014.7.0.

Prepend text to the beginning of a file

path path to file

***args** strings to prepend to the file

CLI Example:

```
salt '*' file.prepend /etc/motd \
        "With all thine offerings thou shalt offer salt." \
        "Salt is what makes things taste bad when it isn't in them."
```

Attention

If you need to pass a string to append and that string contains an equal sign, you **must** include the argument name, args. For example:

```
salt '*' file.prepend /etc/motd args='cheese=spam'

salt '*' file.prepend /etc/motd args="['cheese=spam','spam=cheese']"
```

`salt.modules.file.`**`psed`**(*path*, *before*, *after*, *limit=''*, *backup='.bak'*, *flags='gMS'*, *escape_all=False*, *multi=False*)

Deprecated since version 0.17.0: Use `replace()` instead.

Make a simple edit to a file (pure Python version)

Equivalent to:

```
sed <backup> <options> "/<limit>/ s/<before>/<after>/<flags> <file>"
```

path The full path to the file to be edited

before A pattern to find in order to replace with `after`

after Text that will replace `before`

limit: **`''`** An initial pattern to search for before searching for `before`

backup: **`.bak`** The file will be backed up before edit with this file extension; **WARNING**: each time `sed`/`comment`/`uncomment` is called will overwrite this backup

flags: **`gMS`**

Flags to modify the search. Valid values are:

- `g`: Replace all occurrences of the pattern, not just the first.
- `I`: Ignore case.
- `L`: Make `\w`, `\W`, `\b`, `\B`, `\s` and `\S` dependent on the locale.
- `M`: Treat multiple lines as a single line.
- `S`: Make . match all characters, including newlines.
- `U`: Make `\w`, `\W`, `\b`, `\B`, `\d`, `\D`, `\s` and `\S` dependent on Unicode.
- `X`: Verbose (whitespace is ignored).

multi: **`False`** If True, treat the entire file as a single line

Forward slashes and single quotes will be escaped automatically in the `before` and `after` patterns.

CLI Example:

```
salt '*' file.sed /etc/httpd/httpd.conf 'LogLevel warn' 'LogLevel info'
```

`salt.modules.file.`**`read`**(*path*, *binary=False*)

New in version 2017.7.0.

Return the content of the file.

CLI Example:

```
salt '*' file.read /path/to/file
```

`salt.modules.file.`**`readdir`**`(`*path*`)`

New in version 2014.1.0.

Return a list containing the contents of a directory

CLI Example:

```
salt '*' file.readdir /path/to/dir/
```

`salt.modules.file.`**`readlink`**`(`*path*, *canonicalize=False*`)`

New in version 2014.1.0.

Return the path that a symlink points to If canonicalize is set to True, then it return the final target

CLI Example:

```
salt '*' file.readlink /path/to/link
```

`salt.modules.file.`**`remove`**`(`*path*`)`

Remove the named file. If a directory is supplied, it will be recursively deleted.

CLI Example:

```
salt '*' file.remove /tmp/foo
```

Changed in version 3000: The method now works on all types of file system entries, not just files, directories and symlinks.

`salt.modules.file.`**`rename`**`(`*src*, *dst*`)`

Rename a file or directory

CLI Example:

```
salt '*' file.rename /path/to/src /path/to/dst
```

`salt.modules.file.`**`replace`**`(`*path*, *pattern*, *repl*, *count=0*, *flags=8*, *bufsize=1*, *append_if_not_found=False*, *prepend_if_not_found=False*, *not_found_content=None*, *backup='.bak'*, *dry_run=False*, *search_only=False*, *show_changes=True*, *ignore_if_missing=False*, *preserve_inode=True*, *backslash_literal=False*`)`

New in version 0.17.0.

Replace occurrences of a pattern in a file. If `show_changes` is `True`, then a diff of what changed will be returned, otherwise a `True` will be returned when changes are made, and `False` when no changes are made.

This is a pure Python implementation that wraps Python's `sub()`.

path Filesystem path to the file to be edited. If a symlink is specified, it will be resolved to its target.

pattern A regular expression, to be matched using Python's `search()`.

repl The replacement text

count: 0 Maximum number of pattern occurrences to be replaced. If count is a positive integer `n`, only `n` occurrences will be replaced, otherwise all occurrences will be replaced.

flags (list or int) A list of flags defined in the `re` module documentation from the Python standard library. Each list item should be a string that will correlate to the human-friendly flag name. E.g., `['IGNORECASE', 'MULTILINE']`. Optionally, `flags` may be an int, with a value corresponding to the XOR (|) of all the desired flags. Defaults to 8 (which supports 'MULTILINE').

bufsize (int or str) How much of the file to buffer into memory at once. The default value `1` processes one line at a time. The special value `file` may be specified which will read the entire file into memory before processing.

append_if_not_found: False New in version 2014.7.0.

If set to `True`, and pattern is not found, then the content will be appended to the file.

prepend_if_not_found: False New in version 2014.7.0.

If set to `True` and pattern is not found, then the content will be prepended to the file.

not_found_content New in version 2014.7.0.

Content to use for append/prepend if not found. If None (default), uses `repl`. Useful when `repl` uses references to group in pattern.

backup: .bak The file extension to use for a backup of the file before editing. Set to `False` to skip making a backup.

dry_run: False If set to `True`, no changes will be made to the file, the function will just return the changes that would have been made (or a `True`/`False` value if `show_changes` is set to `False`).

search_only: False If set to true, this no changes will be performed on the file, and this function will simply return `True` if the pattern was matched, and `False` if not.

show_changes: True If `True`, return a diff of changes made. Otherwise, return `True` if changes were made, and `False` if not.

Note: Using this option will store two copies of the file in memory (the original version and the edited version) in order to generate the diff. This may not normally be a concern, but could impact performance if used with large files.

ignore_if_missing: False New in version 2015.8.0.

If set to `True`, this function will simply return `False` if the file doesn't exist. Otherwise, an error will be thrown.

preserve_inode: True New in version 2015.8.0.

Preserve the inode of the file, so that any hard links continue to share the inode with the original filename. This works by *copying* the file, reading from the copy, and writing to the file at the original inode. If `False`, the file will be *moved* rather than copied, and a new file will be written to a new inode, but using the original filename. Hard links will then share an inode with the backup, instead (if using `backup` to create a backup copy).

backslash_literal: False New in version 2016.11.7.

Interpret backslashes as literal backslashes for the repl and not escape characters. This will help when using append/prepend so that the backslashes are not interpreted for the repl on the second run of the state.

If an equal sign (`=`) appears in an argument to a Salt command it is interpreted as a keyword argument in the format `key=val`. That processing can be bypassed in order to pass an equal sign through to the remote shell command by manually specifying the kwarg:

```
salt '*' file.replace /path/to/file pattern='=' repl=':'
salt '*' file.replace /path/to/file pattern="bind-address\s*=" repl='bind-address:
↪'
```

CLI Examples:

```
salt '*' file.replace /etc/httpd/httpd.conf pattern='LogLevel warn' repl=
↪'LogLevel info'
salt '*' file.replace /some/file pattern='before' repl='after' flags='[MULTILINE,⤶
↪IGNORECASE]'
```

`salt.modules.file.`**`restore_backup`**(*path*, *backup_id*)

New in version 0.17.0.

Restore a previous version of a file that was backed up using Salt's *file state backup* system.

path The path on the minion to check for backups

backup_id The numeric id for the backup you wish to restore, as found using `file.list_backups`

CLI Example:

```
salt '*' file.restore_backup /foo/bar/baz.txt 0
```

`salt.modules.file.`**`restorecon`**(*path*, *recursive=False*)

Reset the SELinux context on a given path

CLI Example:

```
salt '*' file.restorecon /home/user/.ssh/authorized_keys
```

`salt.modules.file.`**`rmdir`**(*path*)

New in version 2014.1.0.

Remove the specified directory. Fails if a directory is not empty.

CLI Example:

```
salt '*' file.rmdir /tmp/foo/
```

`salt.modules.file.`**`search`**(*path*, *pattern*, *flags=8*, *bufsize=1*, *ignore_if_missing=False*, *multiline=False*)

New in version 0.17.0.

Search for occurrences of a pattern in a file

Except for multiline, params are identical to `replace()`.

multiline If true, inserts 'MULTILINE' into `flags` and sets `bufsize` to 'file'.

New in version 2015.8.0.

CLI Example:

```
salt '*' file.search /etc/crontab 'mymaintenance.sh'
```

`salt.modules.file.`**`sed`**(*path*, *before*, *after*, *limit=''*, *backup='.bak'*, *options='-r -e'*, *flags='g'*, *escape_all=False*, *negate_match=False*)

Deprecated since version 0.17.0: Use `replace()` instead.

Make a simple edit to a file

Equivalent to:

```
sed <backup> <options> "/<limit>/ s/<before>/<after>/<flags> <file>"
```

path The full path to the file to be edited

before A pattern to find in order to replace with `after`

after Text that will replace `before`

limit: `''` An initial pattern to search for before searching for `before`

backup: `.bak` The file will be backed up before edit with this file extension; **WARNING:** each time `sed`/`comment`/`uncomment` is called will overwrite this backup

options: `-r -e` Options to pass to sed

flags: `g` Flags to modify the sed search; e.g., `i` for case-insensitive pattern matching

negate_match: False Negate the search command (`!`)

New in version 0.17.0.

Forward slashes and single quotes will be escaped automatically in the `before` and `after` patterns.

CLI Example:

```
salt '*' file.sed /etc/httpd/httpd.conf 'LogLevel warn' 'LogLevel info'
```

`salt.modules.file.`**`sed_contains`**(*path*, *text*, *limit=''*, *flags='g'*)

Deprecated since version 0.17.0: Use `search()` instead.

Return True if the file at `path` contains `text`. Utilizes sed to perform the search (line-wise search).

Note: the `p` flag will be added to any flags you pass in.

CLI Example:

```
salt '*' file.contains /etc/crontab 'mymaintenance.sh'
```

`salt.modules.file.`**`seek_read`**(*path*, *size*, *offset*)

New in version 2014.1.0.

Seek to a position on a file and read it

path path to file

seek amount to read at once

offset offset to start into the file

CLI Example:

```
salt '*' file.seek_read /path/to/file 4096 0
```

`salt.modules.file.`**`seek_write`**(*path*, *data*, *offset*)

New in version 2014.1.0.

Seek to a position on a file and write to it

path path to file

data data to write to file

offset position in file to start writing

CLI Example:

```
salt '*' file.seek_write /path/to/file 'some data' 4096
```

`salt.modules.file.`**`set_mode`**(*path*, *mode*)

Set the mode of a file

path file or directory of which to set the mode

mode mode to set the path to

CLI Example:

```
salt '*' file.set_mode /etc/passwd 0644
```

salt.modules.file.**set_selinux_context**(*path, user=None, role=None, type=None, range=None*)

Set a specific SELinux label on a given path

CLI Example:

```
salt '*' file.set_selinux_context path <user> <role> <type> <range>
salt '*' file.set_selinux_context /etc/yum.repos.d/epel.repo system_u object_r
↪system_conf_t s0
```

salt.modules.file.**source_list**(*source, source_hash, saltenv*)

Check the source list and return the source to use

CLI Example:

```
salt '*' file.source_list salt://http/httpd.conf '{hash_type: 'md5', 'hsum':
↪<md5sum>}' base
```

salt.modules.file.**stats**(*path, hash_type=None, follow_symlinks=True*)

Return a dict containing the stats for a given file

CLI Example:

```
salt '*' file.stats /etc/passwd
```

salt.modules.file.**statvfs**(*path*)

New in version 2014.1.0.

Perform a statvfs call against the filesystem that the file resides on

CLI Example:

```
salt '*' file.statvfs /path/to/file
```

salt.modules.file.**symlink**(*src, path*)

Create a symbolic link (symlink, soft link) to a file

CLI Example:

```
salt '*' file.symlink /path/to/file /path/to/link
```

salt.modules.file.**touch**(*name, atime=None, mtime=None*)

New in version 0.9.5.

Just like the `touch` command, create a file if it doesn't exist or simply update the atime and mtime if it already does.

atime: Access time in Unix epoch time

mtime: Last modification in Unix epoch time

CLI Example:

```
salt '*' file.touch /var/log/emptyfile
```

`salt.modules.file.`**`truncate`**`(`*path*, *length*`)`

New in version 2014.1.0.

Seek to a position on a file and delete everything after that point

path path to file

length offset into file to truncate

CLI Example:

```
salt '*' file.truncate /path/to/file 512
```

`salt.modules.file.`**`uid_to_user`**`(`*uid*`)`

Convert a uid to a user name

uid uid to convert to a username

CLI Example:

```
salt '*' file.uid_to_user 0
```

`salt.modules.file.`**`uncomment`**`(`*path*, *regex*, *char='#'*, *backup='.bak'*`)`

Deprecated since version 0.17.0: Use `replace()` instead.

Uncomment specified commented lines in a file

path The full path to the file to be edited

regex A regular expression used to find the lines that are to be uncommented. This regex should not include the comment character. A leading ^ character will be stripped for convenience (for easily switching between comment() and uncomment()).

char: **#** The character to remove in order to uncomment a line

backup: **`.bak`** The file will be backed up before edit with this file extension; **WARNING**: each time `sed`/`comment`/`uncomment` is called will overwrite this backup

CLI Example:

```
salt '*' file.uncomment /etc/hosts.deny 'ALL: PARANOID'
```

`salt.modules.file.`**`user_to_uid`**`(`*user*`)`

Convert user name to a uid

user user name to convert to its uid

CLI Example:

```
salt '*' file.user_to_uid root
```

`salt.modules.file.`**`write`**`(`*path*, **args*, ***kwargs*`)`

New in version 2014.7.0.

Write text to a file, overwriting any existing contents.

path path to file

args strings to write to the file

CLI Example:

```
salt '*' file.write /etc/motd \
        "With all thine offerings thou shalt offer salt."
```

Attention

If you need to pass a string to append and that string contains an equal sign, you **must** include the argument name, args. For example:

```
salt '*' file.write /etc/motd args='cheese=spam'

salt '*' file.write /etc/motd args="['cheese=spam','spam=cheese']"
```

21.9.128 salt.modules.firewalld

Support for firewalld.

New in version 2015.2.0.

`salt.modules.firewalld.`**`add_interface`**(*zone*, *interface*, *permanent=True*)

Bind an interface to a zone

New in version 2016.3.0.

CLI Example:

```
salt '*' firewalld.add_interface zone eth0
```

`salt.modules.firewalld.`**`add_masquerade`**(*zone=None*, *permanent=True*)

Enable masquerade on a zone. If zone is omitted, default zone will be used.

New in version 2015.8.0.

CLI Example:

```
salt '*' firewalld.add_masquerade
```

To enable masquerade on a specific zone

```
salt '*' firewalld.add_masquerade dmz
```

`salt.modules.firewalld.`**`add_port`**(*zone*, *port*, *permanent=True*, *force_masquerade=False*)

Allow specific ports in a zone.

New in version 2015.8.0.

CLI Example:

```
salt '*' firewalld.add_port internal 443/tcp
```

force_masquerade when a zone is created ensure masquerade is also enabled on that zone.

`salt.modules.firewalld.`**`add_port_fwd`**(*zone*, *src*, *dest*, *proto='tcp'*, *dstaddr=''*, *permanent=True*, *force_masquerade=False*)

Add port forwarding.

New in version 2015.8.0.

CLI Example:

```
salt '*' firewalld.add_port_fwd public 80 443 tcp
```

force_masquerade when a zone is created ensure masquerade is also enabled on that zone.

`salt.modules.firewalld.`**`add_rich_rule`**(*zone*, *rule*, *permanent=True*)

Add a rich rule to a zone

New in version 2016.11.0.

CLI Example:

```
salt '*' firewalld.add_rich_rule zone 'rule'
```

`salt.modules.firewalld.`**`add_service`**(*service*, *zone=None*, *permanent=True*)

Add a service for zone. If zone is omitted, default zone will be used.

CLI Example:

```
salt '*' firewalld.add_service ssh
```

To assign a service to a specific zone:

```
salt '*' firewalld.add_service ssh my_zone
```

`salt.modules.firewalld.`**`add_service_port`**(*service*, *port*)

Add a new port to the specified service.

New in version 2016.11.0.

CLI Example:

```
salt '*' firewalld.add_service_port zone 80
```

`salt.modules.firewalld.`**`add_service_protocol`**(*service*, *protocol*)

Add a new protocol to the specified service.

New in version 2016.11.0.

CLI Example:

```
salt '*' firewalld.add_service_protocol zone ssh
```

`salt.modules.firewalld.`**`add_source`**(*zone*, *source*, *permanent=True*)

Bind a source to a zone

New in version 2016.3.0.

CLI Example:

```
salt '*' firewalld.add_source zone 192.168.1.0/24
```

`salt.modules.firewalld.`**`allow_icmp`**(*zone*, *icmp*, *permanent=True*)

Allow a specific ICMP type on a zone

New in version 2015.8.0.

CLI Example:

```
salt '*' firewalld.allow_icmp zone echo-reply
```

`salt.modules.firewalld.`**`block_icmp`**`(`*zone, icmp, permanent=True*`)`

Block a specific ICMP type on a zone

New in version 2015.8.0.

CLI Example:

```
salt '*' firewalld.block_icmp zone echo-reply
```

`salt.modules.firewalld.`**`default_zone`**`()`

Print default zone for connections and interfaces

CLI Example:

```
salt '*' firewalld.default_zone
```

`salt.modules.firewalld.`**`delete_service`**`(`*name, restart=True*`)`

Delete an existing service

CLI Example:

```
salt '*' firewalld.delete_service my_service
```

By default firewalld will be reloaded. However, to avoid reloading you need to specify the restart as False

```
salt '*' firewalld.delete_service my_service False
```

`salt.modules.firewalld.`**`delete_zone`**`(`*zone, restart=True*`)`

Delete an existing zone

CLI Example:

```
salt '*' firewalld.delete_zone my_zone
```

By default firewalld will be reloaded. However, to avoid reloading you need to specify the restart as False

```
salt '*' firewalld.delete_zone my_zone False
```

`salt.modules.firewalld.`**`get_icmp_types`**`(`*permanent=True*`)`

Print predefined icmptypes

CLI Example:

```
salt '*' firewalld.get_icmp_types
```

`salt.modules.firewalld.`**`get_interfaces`**`(`*zone, permanent=True*`)`

List interfaces bound to a zone

New in version 2016.3.0.

CLI Example:

```
salt '*' firewalld.get_interfaces zone
```

`salt.modules.firewalld.`**`get_masquerade`**`(`*zone=None, permanent=True*`)`

Show if masquerading is enabled on a zone. If zone is omitted, default zone will be used.

CLI Example:

```
salt '*' firewalld.get_masquerade zone
```

`salt.modules.firewalld.`**`get_rich_rules`**(*zone*, *permanent=True*)

List rich rules bound to a zone

New in version 2016.11.0.

CLI Example:

```
salt '*' firewalld.get_rich_rules zone
```

`salt.modules.firewalld.`**`get_service_ports`**(*service*)

List ports of a service.

New in version 2016.11.0.

CLI Example:

```
salt '*' firewalld.get_service_ports zone
```

`salt.modules.firewalld.`**`get_service_protocols`**(*service*)

List protocols of a service.

New in version 2016.11.0.

CLI Example:

```
salt '*' firewalld.get_service_protocols zone
```

`salt.modules.firewalld.`**`get_services`**(*permanent=True*)

Print predefined services

CLI Example:

```
salt '*' firewalld.get_services
```

`salt.modules.firewalld.`**`get_sources`**(*zone*, *permanent=True*)

List sources bound to a zone

New in version 2016.3.0.

CLI Example:

```
salt '*' firewalld.get_sources zone
```

`salt.modules.firewalld.`**`get_zones`**(*permanent=True*)

Print predefined zones

CLI Example:

```
salt '*' firewalld.get_zones
```

`salt.modules.firewalld.`**`list_all`**(*zone=None*, *permanent=True*)

List everything added for or enabled in a zone

CLI Example:

```
salt '*' firewalld.list_all
```

List a specific zone

```
salt '*' firewalld.list_all my_zone
```

`salt.modules.firewalld.`**`list_icmp_block`**`(`*zone*, *permanent=True*`)`

List ICMP blocks on a zone

New in version 2015.8.0.

CLI Example:

```
salt '*' firewlld.list_icmp_block zone
```

`salt.modules.firewalld.`**`list_port_fwd`**`(`*zone*, *permanent=True*`)`

List port forwarding

New in version 2015.8.0.

CLI Example:

```
salt '*' firewalld.list_port_fwd public
```

`salt.modules.firewalld.`**`list_ports`**`(`*zone*, *permanent=True*`)`

List all ports in a zone.

New in version 2015.8.0.

CLI Example:

```
salt '*' firewalld.list_ports
```

`salt.modules.firewalld.`**`list_services`**`(`*zone=None*, *permanent=True*`)`

List services added for zone as a space separated list. If zone is omitted, default zone will be used.

CLI Example:

```
salt '*' firewalld.list_services
```

List a specific zone

```
salt '*' firewalld.list_services my_zone
```

`salt.modules.firewalld.`**`list_zones`**`(`*permanent=True*`)`

List everything added for or enabled in all zones

CLI Example:

```
salt '*' firewalld.list_zones
```

`salt.modules.firewalld.`**`make_permanent`**`()`

Make current runtime configuration permanent.

New in version 2016.3.0.

CLI Example:

```
salt '*' firewalld.make_permanent
```

`salt.modules.firewalld.`**`new_service`**`(`*name*, *restart=True*`)`

Add a new service

CLI Example:

```
salt '*' firewalld.new_service my_service
```

By default firewalld will be reloaded. However, to avoid reloading you need to specify the restart as False

```
salt '*' firewalld.new_service my_service False
```

salt.modules.firewalld.new_zone(*zone*, *restart=True*)

Add a new zone

CLI Example:

```
salt '*' firewalld.new_zone my_zone
```

By default firewalld will be reloaded. However, to avoid reloading you need to specify the restart as False

```
salt '*' firewalld.new_zone my_zone False
```

salt.modules.firewalld.reload_rules()

Reload the firewall rules, which makes the permanent configuration the new runtime configuration without losing state information.

New in version 2016.11.0.

CLI Example:

```
salt '*' firewalld.reload
```

salt.modules.firewalld.remove_interface(*zone*, *interface*, *permanent=True*)

Remove an interface bound to a zone

New in version 2016.3.0.

CLI Example:

```
salt '*' firewalld.remove_interface zone eth0
```

salt.modules.firewalld.remove_masquerade(*zone=None*, *permanent=True*)

Remove masquerade on a zone. If zone is omitted, default zone will be used.

New in version 2015.8.0.

CLI Example:

```
salt '*' firewalld.remove_masquerade
```

To remove masquerade on a specific zone

```
salt '*' firewalld.remove_masquerade dmz
```

salt.modules.firewalld.remove_port(*zone*, *port*, *permanent=True*)

Remove a specific port from a zone.

New in version 2015.8.0.

CLI Example:

```
salt '*' firewalld.remove_port internal 443/tcp
```

salt.modules.firewalld.remove_port_fwd(*zone*, *src*, *dest*, *proto='tcp'*, *dstaddr=''*, *permanent=True*)

Remove Port Forwarding.

New in version 2015.8.0.

CLI Example:

```
salt '*' firewalld.remove_port_fwd public 80 443 tcp
```

salt.modules.firewalld.**remove_rich_rule**(*zone, rule, permanent=True*)

Add a rich rule to a zone

New in version 2016.11.0.

CLI Example:

```
salt '*' firewalld.remove_rich_rule zone 'rule'
```

salt.modules.firewalld.**remove_service**(*service, zone=None, permanent=True*)

Remove a service from zone. This option can be specified multiple times. If zone is omitted, default zone will be used.

CLI Example:

```
salt '*' firewalld.remove_service ssh
```

To remove a service from a specific zone

```
salt '*' firewalld.remove_service ssh dmz
```

salt.modules.firewalld.**remove_service_port**(*service, port*)

Remove a port from the specified service.

New in version 2016.11.0.

CLI Example:

```
salt '*' firewalld.remove_service_port zone 80
```

salt.modules.firewalld.**remove_service_protocol**(*service, protocol*)

Remove a protocol from the specified service.

New in version 2016.11.0.

CLI Example:

```
salt '*' firewalld.remove_service_protocol zone ssh
```

salt.modules.firewalld.**remove_source**(*zone, source, permanent=True*)

Remove a source bound to a zone

New in version 2016.3.0.

CLI Example:

```
salt '*' firewalld.remove_source zone 192.168.1.0/24
```

salt.modules.firewalld.**set_default_zone**(*zone*)

Set default zone

CLI Example:

```
salt '*' firewalld.set_default_zone damian
```

salt.modules.firewalld.**version**()

Return version from firewall-cmd

CLI Example:

```
salt '*' firewalld.version
```

21.9.129 salt.modules.freebsd_sysctl

Module for viewing and modifying sysctl parameters

`salt.modules.freebsd_sysctl.`**`assign`**`(`*name, value*`)`

Assign a single sysctl parameter for this minion

CLI Example:

```
salt '*' sysctl.assign net.inet.icmp.icmplim 50
```

`salt.modules.freebsd_sysctl.`**`get`**`(`*name*`)`

Return a single sysctl parameter for this minion

CLI Example:

```
salt '*' sysctl.get hw.physmem
```

`salt.modules.freebsd_sysctl.`**`persist`**`(`*name, value, config='/etc/sysctl.conf'*`)`

Assign and persist a simple sysctl parameter for this minion

CLI Example:

```
salt '*' sysctl.persist net.inet.icmp.icmplim 50
salt '*' sysctl.persist coretemp_load NO config=/boot/loader.conf
```

`salt.modules.freebsd_sysctl.`**`show`**`(`*config_file=False*`)`

Return a list of sysctl parameters for this minion

CLI Example:

```
salt '*' sysctl.show
```

21.9.130 salt.modules.freebsd_update module

Support for freebsd-update utility on FreeBSD.

New in version 2017.7.0.

maintainer George Mamalakis <mamalos@gmail.com>

maturity new

platform FreeBSD

`salt.modules.freebsd_update.`**`fetch`**`(`***kwargs*`)`

New in version 2016.3.4.

freebsd-update fetch wrapper. Based on the currently installed world and the configuration options set, fetch all available binary updates.

kwargs: Parameters of freebsd-update command.

`salt.modules.freebsd_update.`**`ids`**`(`***kwargs*`)`

New in version 2016.3.4.

freebsd-update IDS wrapper function. Compares the system against a "known good" index of the installed release.

kwargs: Parameters of freebsd-update command.

`salt.modules.freebsd_update.`**`install`**`(`*`**kwargs`*`)`

New in version 2016.3.4.

freebsd-update install wrapper. Install the most recently fetched updates or upgrade.

kwargs: Parameters of freebsd-update command.

`salt.modules.freebsd_update.`**`rollback`**`(`*`**kwargs`*`)`

New in version 2016.3.4.

freebsd-update rollback wrapper. Uninstalls the most recently installed updates.

kwargs: Parameters of freebsd-update command.

`salt.modules.freebsd_update.`**`update`**`(`*`**kwargs`*`)`

New in version 2016.3.4.

Command that simplifies freebsd-update by running freebsd-update fetch first and then freebsd-update install.

kwargs: Parameters of freebsd-update command.

`salt.modules.freebsd_update.`**`upgrade`**`(`*`**kwargs`*`)`

New in version 2016.3.4.

Dummy function used only to print a message that upgrade is not available. The reason is that upgrade needs manual intervention and reboot, so even if used with:

> yes | freebsd-upgrade -r VERSION

the additional freebsd-update install that needs to run after the reboot cannot be implemented easily.

kwargs: Parameters of freebsd-update command.

21.9.131 salt.modules.freebsdjail

The jail module for FreeBSD

`salt.modules.freebsdjail.`**`fstab`**`(`*`jail`*`)`

Display contents of a fstab(5) file defined in specified jail's configuration. If no file is defined, return False.

CLI Example:

```
salt '*' jail.fstab <jail name>
```

`salt.modules.freebsdjail.`**`get_enabled`**`()`

Return which jails are set to be run

CLI Example:

```
salt '*' jail.get_enabled
```

`salt.modules.freebsdjail.`**`is_enabled`**`()`

See if jail service is actually enabled on boot

CLI Example:

```
salt '*' jail.is_enabled <jail name>
```

`salt.modules.freebsdjail.`**`restart`**`(`*jail=''*`)`
Restart the specified jail or all, if none specified

CLI Example:

```
salt '*' jail.restart [<jail name>]
```

`salt.modules.freebsdjail.`**`show_config`**`(`*jail*`)`
Display specified jail's configuration

CLI Example:

```
salt '*' jail.show_config <jail name>
```

`salt.modules.freebsdjail.`**`start`**`(`*jail=''*`)`
Start the specified jail or all, if none specified

CLI Example:

```
salt '*' jail.start [<jail name>]
```

`salt.modules.freebsdjail.`**`status`**`(`*jail*`)`
See if specified jail is currently running

CLI Example:

```
salt '*' jail.status <jail name>
```

`salt.modules.freebsdjail.`**`stop`**`(`*jail=''*`)`
Stop the specified jail or all, if none specified

CLI Example:

```
salt '*' jail.stop [<jail name>]
```

`salt.modules.freebsdjail.`**`sysctl`**`()`
Dump all jail related kernel states (sysctl)

CLI Example:

```
salt '*' jail.sysctl
```

21.9.132 salt.modules.freebsdkmod

Module to manage FreeBSD kernel modules

`salt.modules.freebsdkmod.`**`available`**`()`
Return a list of all available kernel modules

CLI Example:

```
salt '*' kmod.available
```

`salt.modules.freebsdkmod.`**`check_available`**`(`*mod*`)`
Check to see if the specified kernel module is available

CLI Example:

```
salt '*' kmod.check_available vmm
```

`salt.modules.freebsdkmod.`**`is_loaded`**`(`*mod*`)`

Check to see if the specified kernel module is loaded

CLI Example:

```
salt '*' kmod.is_loaded vmm
```

`salt.modules.freebsdkmod.`**`load`**`(`*mod, persist=False*`)`

Load the specified kernel module

mod Name of the module to add

persist Write the module to sysrc kld_modules to make it load on system reboot

CLI Example:

```
salt '*' kmod.load bhyve
```

`salt.modules.freebsdkmod.`**`lsmod`**`()`

Return a dict containing information about currently loaded modules

CLI Example:

```
salt '*' kmod.lsmod
```

`salt.modules.freebsdkmod.`**`mod_list`**`(`*only_persist=False*`)`

Return a list of the loaded module names

CLI Example:

```
salt '*' kmod.mod_list
```

`salt.modules.freebsdkmod.`**`remove`**`(`*mod, persist=False, comment=True*`)`

Remove the specified kernel module

mod Name of module to remove

persist Also remove module from /boot/loader.conf

comment If persist is set don't remove line from /boot/loader.conf but only comment it

CLI Example:

```
salt '*' kmod.remove vmm
```

21.9.133 salt.modules.freebsdpkg

Remote package support using `pkg_add(1)`

Important: If you feel that Salt should be using this module to manage packages on a minion, and it is using a different module (or gives an error similar to *'pkg.install' is not available*), see *here*.

Warning: This module has been completely rewritten. Up to and including version 0.17.0, it supported `pkg_add(1)`, but checked for the existence of a pkgng local database and, if found, would provide some of pkgng's functionality. The rewrite of this module has removed all pkgng support, and moved it to the *pkgng* execution module. For versions <= 0.17.0, the documentation here should not be considered accurate. If your Minion is running one of these versions, then the documentation for this module can be viewed using the `sys.doc` function:

```
salt bsdminion sys.doc pkg
```

This module acts as the default package provider for FreeBSD 9 and older. If you need to use pkgng on a FreeBSD 9 system, you will need to override the `pkg` provider by setting the *`providers`* parameter in your Minion config file, in order to use pkgng.

```
providers:
  pkg: pkgng
```

More information on pkgng support can be found in the documentation for the *pkgng* module.

This module will respect the `PACKAGEROOT` and `PACKAGESITE` environment variables, if set, but these values can also be overridden in several ways:

1. **Salt configuration parameters.** The configuration parameters `freebsdpkg.PACKAGEROOT` and `freebsdpkg.PACKAGESITE` are recognized. These config parameters are looked up using *`config.get`* and can thus be specified in the Master config file, Grains, Pillar, or in the Minion config file. Example:

```
freebsdpkg.PACKAGEROOT: ftp://ftp.freebsd.org/
freebsdpkg.PACKAGESITE: ftp://ftp.freebsd.org/pub/FreeBSD/ports/ia64/packages-9-
↪stable/Latest/
```

2. **CLI arguments.** Both the `packageroot` (used interchangeably with `fromrepo` for API compatibility) and `packagesite` CLI arguments are recognized, and override their config counterparts from section 1 above.

```
    salt -G 'os:FreeBSD' pkg.install zsh fromrepo=ftp://ftp2.freebsd.org/
    salt -G 'os:FreeBSD' pkg.install zsh packageroot=ftp://ftp2.freebsd.org/
    salt -G 'os:FreeBSD' pkg.install zsh packagesite=ftp://ftp2.freebsd.org/pub/
↪FreeBSD/ports/ia64/packages-9-stable/Latest/

.. note::

    These arguments can also be passed through in states:

    .. code-block:: yaml

        zsh:
          pkg.installed:
            - fromrepo: ftp://ftp2.freebsd.org/
```

`salt.modules.freebsdpkg.`**`file_dict`**`(`**packages*`)`

List the files that belong to a package, grouped by package. Not specifying any packages will return a list of _every_ file on the system's package database (not generally recommended).

CLI Examples:

```
salt '*' pkg.file_list httpd
salt '*' pkg.file_list httpd postfix
salt '*' pkg.file_list
```

`salt.modules.freebsdpkg.`**`file_list`**`(`**packages*`)`

List the files that belong to a package. Not specifying any packages will return a list of _every_ file on the system's package database (not generally recommended).

CLI Examples:

```
salt '*' pkg.file_list httpd
salt '*' pkg.file_list httpd postfix
salt '*' pkg.file_list
```

`salt.modules.freebsdpkg.`**`install`**`(`*name=None, refresh=False, fromrepo=None, pkgs=None, sources=None, **kwargs*`)`

Install package(s) using `pkg_add(1)`

name The name of the package to be installed.

refresh Whether or not to refresh the package database before installing.

fromrepo or packageroot Specify a package repository from which to install. Overrides the system default, as well as the PACKAGEROOT environment variable.

packagesite Specify the exact directory from which to install the remote package. Overrides the PACKAGESITE environment variable, if present.

Multiple Package Installation Options:

pkgs A list of packages to install from a software repository. Must be passed as a python list.

CLI Example:

```
salt '*' pkg.install pkgs='["foo", "bar"]'
```

sources A list of packages to install. Must be passed as a list of dicts, with the keys being package names, and the values being the source URI or local path to the package.

CLI Example:

```
salt '*' pkg.install sources='[{"foo": "salt://foo.deb"}, {"bar": "salt://bar.
↪deb"}]'
```

Return a dict containing the new package names and versions:

```
{'<package>': {'old': '<old-version>',
               'new': '<new-version>'}}
```

CLI Example:

```
salt '*' pkg.install <package name>
```

`salt.modules.freebsdpkg.`**`latest_version`**`(`**names, **kwargs*`)`

`pkg_add(1)` is not capable of querying for remote packages, so this function will always return results as if there is no package available for install or upgrade.

CLI Example:

```
salt '*' pkg.latest_version <package name>
salt '*' pkg.latest_version <package1> <package2> <package3> ...
```

`salt.modules.freebsdpkg.`**`list_pkgs`**`(`*versions_as_list=False, with_origin=False, **kwargs*`)`

List the packages currently installed as a dict:

```
{'<package_name>': '<version>'}
```

with_origin [False] Return a nested dictionary containing both the origin name and version for each installed package.

New in version 2014.1.0.

CLI Example:

```
salt '*' pkg.list_pkgs
```

`salt.modules.freebsdpkg.`**`refresh_db`**`()`

`pkg_add(1)` does not use a local database of available packages, so this function simply returns `True`. it exists merely for API compatibility.

CLI Example:

```
salt '*' pkg.refresh_db
```

`salt.modules.freebsdpkg.`**`remove`**`(`*name=None*, *pkgs=None*, ***kwargs*`)`

Remove packages using `pkg_delete(1)`

name The name of the package to be deleted.

Multiple Package Options:

pkgs A list of packages to delete. Must be passed as a python list. The `name` parameter will be ignored if this option is passed.

New in version 0.16.0.

Returns a dict containing the changes.

CLI Example:

```
salt '*' pkg.remove <package name>
salt '*' pkg.remove <package1>,<package2>,<package3>
salt '*' pkg.remove pkgs='["foo", "bar"]'
```

`salt.modules.freebsdpkg.`**`version`**`(`**names*, ***kwargs*`)`

Returns a string representing the package version or an empty string if not installed. If more than one package name is specified, a dict of name/version pairs is returned.

with_origin [False] Return a nested dictionary containing both the origin name and version for each specified package.

New in version 2014.1.0.

CLI Example:

```
salt '*' pkg.version <package name>
salt '*' pkg.version <package1> <package2> <package3> ...
```

21.9.134 salt.modules.freebsdports

Install software from the FreeBSD `ports(7)` system

New in version 2014.1.0.

This module allows you to install ports using `BATCH=yes` to bypass configuration prompts. It is recommended to use the `ports state` to install ports, but it is also possible to use this module exclusively from the command line.

```
salt minion-id ports.config security/nmap IPV6=off
salt minion-id ports.install security/nmap
```

`salt.modules.freebsdports.`**`config`**`(`*name*, *reset=False*, ***kwargs*`)`

Modify configuration options for a given port. Multiple options can be specified. To see the available options for a port, use *`ports.showconfig`*.

name The port name, in `category/name` format

reset [False] If `True`, runs a `make rmconfig` for the port, clearing its configuration before setting the desired options

CLI Examples:

```
salt '*' ports.config security/nmap IPV6=off
```

`salt.modules.freebsdports.`**`deinstall`**`(`*name*`)`

De-install a port.

CLI Example:

```
salt '*' ports.deinstall security/nmap
```

`salt.modules.freebsdports.`**`install`**`(`*name*, *clean=True*`)`

Install a port from the ports tree. Installs using `BATCH=yes` for non-interactive building. To set config options for a given port, use *`ports.config`*.

clean [True] If `True`, cleans after installation. Equivalent to running `make install clean BATCH=yes`.

Note: It may be helpful to run this function using the `-t` option to set a higher timeout, since compiling a port may cause the Salt command to exceed the default timeout.

CLI Example:

```
salt -t 1200 '*' ports.install security/nmap
```

`salt.modules.freebsdports.`**`list_all`**`()`

Lists all ports available.

CLI Example:

```
salt '*' ports.list_all
```

Warning: Takes a while to run, and returns a **LOT** of output

`salt.modules.freebsdports.`**`rmconfig`**`(`*name*`)`

Clear the cached options for the specified port; run a `make rmconfig`

name The name of the port to clear

CLI Example:

```
salt '*' ports.rmconfig security/nmap
```

`salt.modules.freebsdports.`**`search`**`(`*name*`)`

Search for matches in the ports tree. Globs are supported, and the category is optional

CLI Examples:

```
salt '*' ports.search 'security/*'
salt '*' ports.search 'security/n*'
salt '*' ports.search nmap
```

Warning: Takes a while to run

`salt.modules.freebsdports.`**`showconfig`**`(`*name, default=False, dict_return=False*`)`

Show the configuration options for a given port.

default [False] Show the default options for a port (not necessarily the same as the current configuration)

dict_return [False] Instead of returning the output of `make showconfig`, return the data in an dictionary

CLI Example:

```
salt '*' ports.showconfig security/nmap
salt '*' ports.showconfig security/nmap default=True
```

`salt.modules.freebsdports.`**`update`**`(`*extract=False*`)`

Update the ports tree

extract [False] If `True`, runs a `portsnap extract` after fetching, should be used for first-time installation of the ports tree.

CLI Example:

```
salt '*' ports.update
```

21.9.135 salt.modules.freebsdservice

The service module for FreeBSD

Important: If you feel that Salt should be using this module to manage services on a minion, and it is using a different module (or gives an error similar to *'service.start' is not available*), see *here*.

`salt.modules.freebsdservice.`**`available`**`(`*name, jail=None*`)`

Check that the given service is available.

Changed in version 2016.3.4.

jail: optional jid or jail name

CLI Example:

```
salt '*' service.available sshd
```

`salt.modules.freebsdservice.`**`disable`**`(`*name, **kwargs*`)`

Disable the named service to start at boot

Arguments the same as for enable()

Changed in version 2016.3.4.

jail (optional keyword argument) the jail's id or name

chroot (optional keyword argument) the jail's chroot, if the jail's /etc is not mounted read-write

CLI Example:

```
salt '*' service.disable <service name>
```

`salt.modules.freebsdservice.`**`disabled`**`(`*name*, ***kwargs*`)`

Return True if the named service is enabled, false otherwise

CLI Example:

```
salt '*' service.disabled <service name>
```

`salt.modules.freebsdservice.`**`enable`**`(`*name*, ***kwargs*`)`

Enable the named service to start at boot

name service name

config [/etc/rc.conf] Config file for managing service. If config value is empty string, then /etc/rc.conf.d/<service> used. See man rc.conf(5) for details.

Also service.config variable can be used to change default.

Changed in version 2016.3.4.

jail (optional keyword argument) the jail's id or name

chroot (optional keyword argument) the jail's chroot, if the jail's /etc is not mounted read-write

CLI Example:

```
salt '*' service.enable <service name>
```

`salt.modules.freebsdservice.`**`enabled`**`(`*name*, ***kwargs*`)`

Return True if the named service is enabled, false otherwise

name Service name

Changed in version 2016.3.4.

Support for jail (representing jid or jail name) keyword argument in kwargs

CLI Example:

```
salt '*' service.enabled <service name>
```

`salt.modules.freebsdservice.`**`get_all`**`(`*jail=None*`)`

Return a list of all available services

Changed in version 2016.3.4.

jail: optional jid or jail name

CLI Example:

```
salt '*' service.get_all
```

`salt.modules.freebsdservice.`**`get_disabled`**`(`*jail=None*`)`

Return what services are available but not enabled to start at boot

Changed in version 2016.3.4.

Support for jail (representing jid or jail name) keyword argument in kwargs

CLI Example:

```
salt '*' service.get_disabled
```

`salt.modules.freebsdservice.`**`get_enabled`**`(`*jail=None*`)`

Return what services are set to run on boot

Changed in version 2016.3.4.

Support for jail (representing jid or jail name) keyword argument in kwargs

CLI Example:

```
salt '*' service.get_enabled
```

`salt.modules.freebsdservice.`**`missing`**`(`*name*, *jail=None*`)`

The inverse of service.available. Returns `True` if the specified service is not available, otherwise returns `False`.

Changed in version 2016.3.4.

jail: optional jid or jail name

CLI Example:

```
salt '*' service.missing sshd
```

`salt.modules.freebsdservice.`**`reload_`**`(`*name*, *jail=None*`)`

Restart the named service

Changed in version 2016.3.4.

jail: optional jid or jail name

CLI Example:

```
salt '*' service.reload <service name>
```

`salt.modules.freebsdservice.`**`restart`**`(`*name*, *jail=None*`)`

Restart the named service

Changed in version 2016.3.4.

jail: optional jid or jail name

CLI Example:

```
salt '*' service.restart <service name>
```

`salt.modules.freebsdservice.`**`start`**`(`*name*, *jail=None*`)`

Start the specified service

Changed in version 2016.3.4.

jail: optional jid or jail name

CLI Example:

```
salt '*' service.start <service name>
```

`salt.modules.freebsdservice.`**`status`**(*name*, *sig=None*, *jail=None*)

Return the status for a service. If the name contains globbing, a dict mapping service name to True/False values is returned.

Changed in version 2016.3.4.

Changed in version 2018.3.0: The service name can now be a glob (e.g. `salt*`)

Parameters

- **`name`** (*`str`*) – The name of the service to check
- **`sig`** (*`str`*) – Signature to use to find the service via ps

Returns True if running, False otherwise dict: Maps service name to True if running, False otherwise

Return type bool

CLI Example:

```
salt '*' service.status <service name> [service signature]
```

`salt.modules.freebsdservice.`**`stop`**(*name*, *jail=None*)

Stop the specified service

Changed in version 2016.3.4.

jail: optional jid or jail name

CLI Example:

```
salt '*' service.stop <service name>
```

21.9.136 salt.modules.gem

Manage ruby gems.

`salt.modules.gem.`**`install`**(*gems*, *ruby=None*, *gem_bin=None*, *runas=None*, *version=None*, *rdoc=False*, *ri=False*, *pre_releases=False*, *proxy=None*, *source=None*)

Installs one or several gems.

Parameters

- **`gems`** – string The gems to install
- **`gem_bin`** – string : None Full path to `gem` binary to use.
- **`ruby`** – string : None If RVM or rbenv are installed, the ruby version and gemset to use. Ignored if `gem_bin` is specified.
- **`runas`** – string : None The user to run gem as.
- **`version`** – string : None Specify the version to install for the gem. Doesn't play nice with multiple gems at once
- **`rdoc`** – boolean : False Generate RDoc documentation for the gem(s). For rubygems > 3 this is interpreted as the –no-document arg and the ri option will then be ignored
- **`ri`** – boolean : False Generate RI documentation for the gem(s). For rubygems > 3 this is interpreted as the –no-document arg and the rdoc option will then be ignored

- **`pre_releases`** – boolean : False Include pre-releases in the available versions
- **`proxy`** – string : None Use the specified HTTP proxy server for all outgoing traffic. Format: http://hostname{[}:port]

source [None] Use the specified HTTP gem source server to download gem. Format: http://hostname{[}:port]

CLI Example:

```
salt '*' gem.install vagrant

salt '*' gem.install redphone gem_bin=/opt/sensu/embedded/bin/gem
```

`salt.modules.gem.`**`list_`**(*prefix=", ruby=None, runas=None, gem_bin=None*)

List locally installed gems.

Parameters

- **`prefix`** – string : Only list gems when the name matches this prefix.
- **`gem_bin`** – string : None Full path to `gem` binary to use.
- **`ruby`** – string : None If RVM or rbenv are installed, the ruby version and gemset to use. Ignored if `gem_bin` is specified.
- **`runas`** – string : None The user to run gem as.

CLI Example:

```
salt '*' gem.list
```

`salt.modules.gem.`**`list_upgrades`**(*ruby=None, runas=None, gem_bin=None*)

New in version 2015.8.0.

Check if an upgrade is available for installed gems

gem_bin [None] Full path to `gem` binary to use.

ruby [None] If RVM or rbenv are installed, the ruby version and gemset to use. Ignored if `gem_bin` is specified.

runas [None] The user to run gem as.

CLI Example:

```
salt '*' gem.list_upgrades
```

`salt.modules.gem.`**`sources_add`**(*source_uri, ruby=None, runas=None, gem_bin=None*)

Add a gem source.

Parameters

- **`source_uri`** – string The source URI to add.
- **`gem_bin`** – string : None Full path to `gem` binary to use.
- **`ruby`** – string : None If RVM or rbenv are installed, the ruby version and gemset to use. Ignored if `gem_bin` is specified.
- **`runas`** – string : None The user to run gem as.

CLI Example:

```
salt '*' gem.sources_add http://rubygems.org/
```

`salt.modules.gem.`**`sources_list`**`(`*ruby=None*, *runas=None*, *gem_bin=None*`)`
List the configured gem sources.

Parameters

- **`gem_bin`** – string : None Full path to `gem` binary to use.
- **`ruby`** – string : None If RVM or rbenv are installed, the ruby version and gemset to use. Ignored if `gem_bin` is specified.
- **`runas`** – string : None The user to run gem as.

CLI Example:

```
salt '*' gem.sources_list
```

`salt.modules.gem.`**`sources_remove`**`(`*source_uri*, *ruby=None*, *runas=None*, *gem_bin=None*`)`
Remove a gem source.

Parameters

- **`source_uri`** – string The source URI to remove.
- **`gem_bin`** – string : None Full path to `gem` binary to use.
- **`ruby`** – string : None If RVM or rbenv are installed, the ruby version and gemset to use. Ignored if `gem_bin` is specified.
- **`runas`** – string : None The user to run gem as.

CLI Example:

```
salt '*' gem.sources_remove http://rubygems.org/
```

`salt.modules.gem.`**`uninstall`**`(`*gems*, *ruby=None*, *runas=None*, *gem_bin=None*`)`
Uninstall one or several gems.

Parameters

- **`gems`** – string The gems to uninstall.
- **`gem_bin`** – string : None Full path to `gem` binary to use.
- **`ruby`** – string : None If RVM or rbenv are installed, the ruby version and gemset to use. Ignored if `gem_bin` is specified.
- **`runas`** – string : None The user to run gem as.

CLI Example:

```
salt '*' gem.uninstall vagrant
```

`salt.modules.gem.`**`update`**`(`*gems*, *ruby=None*, *runas=None*, *gem_bin=None*`)`
Update one or several gems.

Parameters

- **`gems`** – string The gems to update.
- **`gem_bin`** – string : None Full path to `gem` binary to use.
- **`ruby`** – string : None If RVM or rbenv are installed, the ruby version and gemset to use. Ignored if `gem_bin` is specified.

- **runas** – string : None The user to run gem as.

CLI Example:

```
salt '*' gem.update vagrant
```

`salt.modules.gem.`**`update_system`**(*version=”*, *ruby=None*, *runas=None*, *gem_bin=None*)

Update rubygems.

Parameters

- **version** – string : (newest) The version of rubygems to install.
- **gem_bin** – string : None Full path to `gem` binary to use.
- **ruby** – string : None If RVM or rbenv are installed, the ruby version and gemset to use. Ignored if `gem_bin` is specified.
- **runas** – string : None The user to run gem as.

CLI Example:

```
salt '*' gem.update_system
```

`salt.modules.gem.`**`version`**(*ruby=None*, *runas=None*, *gem_bin=None*)

Print out the version of gem

Parameters

- **gem_bin** – string : None Full path to `gem` binary to use.
- **ruby** – string : None If RVM or rbenv are installed, the ruby version and gemset to use. Ignored if `gem_bin` is specified.
- **runas** – string : None The user to run gem as.

CLI Example:

```
salt '*' gem.version
```

21.9.137 salt.modules.genesis

Module for managing container and VM images

New in version 2014.7.0.

`salt.modules.genesis.`**`avail_platforms`**()

Return which platforms are available

CLI Example:

```
salt myminion genesis.avail_platforms
```

`salt.modules.genesis.`**`bootstrap`**(*platform*, *root*, *img_format='dir'*, *fs_format='ext2'*, *fs_opts=None*, *arch=None*, *flavor=None*, *repo_url=None*, *static_qemu=None*, *img_size=None*, *mount_dir=None*, *pkg_cache=None*, *pkgs=None*, *exclude_pkgs=None*, *epel_url='http://download.fedoraproject.org/pub/epel/6/i386/epel-release-6-8.noarch.rpm'*)

Create an image for a specific platform.

Please note that this function *MUST* be run as root, as images that are created make files belonging to root.

platform Which platform to use to create the image. Currently supported platforms are rpm, deb and pacman.

root Local path to create the root of the image filesystem.

img_format Which format to create the image in. By default, just copies files into a directory on the local filesystem (`dir`). Future support will exist for `sparse`.

fs_format When using a non-`dir` `img_format`, which filesystem to format the image to. By default, `ext2`.

fs_opts When using a non-`dir` `img_format`, a dict of opts may be specified.

arch Architecture to install packages for, if supported by the underlying bootstrap tool. Currently only used for deb.

flavor Which flavor of operating system to install. This correlates to a specific directory on the distribution repositories. For instance, `wheezy` on Debian.

repo_url Mainly important for Debian-based repos. Base URL for the mirror to install from. (e.x.: http://ftp.debian.org/debian/)

static_qemu Local path to the static qemu binary required for this arch. (e.x.: /usr/bin/qemu-amd64-static)

pkg_confs The location of the conf files to copy into the image, to point the installer to the right repos and configuration.

img_size If img_format is not `dir`, then the size of the image must be specified.

mount_dir If img_format is not `dir`, then the image must be mounted somewhere. If the `mount_dir` is not specified, then it will be created at `/opt/salt-genesis.<random_uuid>`. This directory will be unmounted and removed when the process is finished.

pkg_cache This points to a directory containing a cache of package files to be copied to the image. It does not need to be specified.

pkgs A list of packages to be installed on this image. For RedHat, this will include `yum`, `centos-release` and `iputils` by default.

exclude_pkgs A list of packages to be excluded. If you do not want to install the defaults, you need to include them in this list.

epel_url The URL to download the EPEL release package from.

CLI Examples:

```
salt myminion genesis.bootstrap pacman /root/arch
salt myminion genesis.bootstrap rpm /root/redhat
salt myminion genesis.bootstrap deb /root/wheezy arch=amd64 
↪flavor=wheezy static_qemu=/usr/bin/qemu-x86_64-static
```

`salt.modules.genesis.`**`ldd_deps`**(*filename*, *ret=None*)

Recurse through a set of dependencies reported by `ldd`, to find associated dependencies.

Please note that this does not necessarily resolve all (non-package) dependencies for a file; but it does help.

CLI Example:

salt myminion genesis.ldd_deps bash salt myminion genesis.ldd_deps /bin/bash

`salt.modules.genesis.`**`mksls`**(*fmt*, *src*, *dst=None*)

Convert an installation file/script to an SLS file. Currently supports `kickstart`, `preseed`, and `autoyast`.

CLI Examples:

salt <minion> genesis.mksls kickstart /path/to/kickstart.cfg salt <minion> genesis.mksls kickstart /path/to/kickstart.cfg /path/to/dest.sls

New in version Beryllium.

`salt.modules.genesis.`**`pack`**`(`*name*, *root*, *path=None*, *pack_format='tar'*, *compress='bzip2'*`)`

Pack up a directory structure, into a specific format

CLI Examples:

```
salt myminion genesis.pack centos /root/centos
salt myminion genesis.pack centos /root/centos pack_format='tar'
```

`salt.modules.genesis.`**`unpack`**`(`*name*, *dest=None*, *path=None*, *pack_format='tar'*, *compress='bz2'*`)`

Unpack an image into a directory structure

CLI Example:

```
salt myminion genesis.unpack centos /root/centos
```

21.9.138 salt.modules.gentoo_service

Top level package command wrapper, used to translate the os detected by grains to the correct service manager

---

Important: If you feel that Salt should be using this module to manage services on a minion, and it is using a different module (or gives an error similar to *'service.start' is not available*), see *here*.

`salt.modules.gentoo_service.`**`available`**`(`*name*`)`

Returns `True` if the specified service is available, otherwise returns `False`.

CLI Example:

```
salt '*' service.available sshd
```

`salt.modules.gentoo_service.`**`disable`**`(`*name*, ***kwargs*`)`

Disable the named service to start at boot

CLI Example:

```
salt '*' service.disable <service name> <runlevels=single-runlevel>
salt '*' service.disable <service name> <runlevels=[runlevel1,runlevel2]>
```

`salt.modules.gentoo_service.`**`disabled`**`(`*name*`)`

Return True if the named service is enabled, false otherwise

CLI Example:

```
salt '*' service.disabled <service name> <runlevels=[runlevel]>
```

`salt.modules.gentoo_service.`**`enable`**`(`*name*, ***kwargs*`)`

Enable the named service to start at boot

CLI Example:

```
salt '*' service.enable <service name> <runlevels=single-runlevel>
salt '*' service.enable <service name> <runlevels=[runlevel1,runlevel2]>
```

`salt.modules.gentoo_service.`**`enabled`**`(`*name*, ***kwargs*`)`

Return True if the named service is enabled, false otherwise

CLI Example:

```
salt '*' service.enabled <service name> <runlevels=single-runlevel>
salt '*' service.enabled <service name> <runlevels=[runlevel1,runlevel2]>
```

`salt.modules.gentoo_service.`**`get_all`**`()`

Return all available boot services

CLI Example:

```
salt '*' service.get_all
```

`salt.modules.gentoo_service.`**`get_disabled`**`()`

Return a set of services that are installed but disabled

CLI Example:

```
salt '*' service.get_disabled
```

`salt.modules.gentoo_service.`**`get_enabled`**`()`

Return a list of service that are enabled on boot

CLI Example:

```
salt '*' service.get_enabled
```

`salt.modules.gentoo_service.`**`missing`**`(`*name*`)`

The inverse of service.available. Returns `True` if the specified service is not available, otherwise returns `False`.

CLI Example:

```
salt '*' service.missing sshd
```

`salt.modules.gentoo_service.`**`reload_`**`(`*name*`)`

Reload the named service

CLI Example:

```
salt '*' service.reload <service name>
```

`salt.modules.gentoo_service.`**`restart`**`(`*name*`)`

Restart the named service

CLI Example:

```
salt '*' service.restart <service name>
```

`salt.modules.gentoo_service.`**`start`**`(`*name*`)`

Start the specified service

CLI Example:

```
salt '*' service.start <service name>
```

`salt.modules.gentoo_service.`**`status`**`(`*name, sig=None*`)`

Return the status for a service. If the name contains globbing, a dict mapping service name to True/False values is returned.

Changed in version 2018.3.0: The service name can now be a glob (e.g. `salt*`)

Parameters

- **name** (*str*) – The name of the service to check
- **sig** (*str*) – Signature to use to find the service via ps

Returns True if running, False otherwise dict: Maps service name to True if running, False otherwise

Return type bool

CLI Example:

```
salt '*' service.status <service name> [service signature]
```

`salt.modules.gentoo_service.`**`stop`**(*name*)

Stop the specified service

CLI Example:

```
salt '*' service.stop <service name>
```

`salt.modules.gentoo_service.`**`zap`**(*name*)

Resets service state

CLI Example:

```
salt '*' service.zap <service name>
```

21.9.139 salt.modules.gentoolkitmod

Support for Gentoolkit

`salt.modules.gentoolkitmod.`**`eclean_dist`**(*destructive=False, package_names=False, size_limit=0, time_limit=0, fetch_restricted=False, exclude_file='/etc/eclean/distfiles.exclude'*)

Clean obsolete portage sources

destructive Only keep minimum for reinstallation

package_names Protect all versions of installed packages. Only meaningful if used with destructive=True

size_limit <size> Don't delete distfiles bigger than <size>. <size> is a size specification: "10M" is "ten megabytes", "200K" is "two hundreds kilobytes", etc. Units are: G, M, K and B.

time_limit <time> Don't delete distfiles files modified since <time> <time> is an amount of time: "1y" is "one year", "2w" is "two weeks", etc. Units are: y (years), m (months), w (weeks), d (days) and h (hours).

fetch_restricted Protect fetch-restricted files. Only meaningful if used with destructive=True

exclude_file Path to exclusion file. Default is /etc/eclean/distfiles.exclude This is the same default eclean-dist uses. Use None if this file exists and you want to ignore.

Returns a dict containing the cleaned, saved, and deprecated dists:

```
{'cleaned': {<dist file>: <size>},
 'deprecated': {<package>: <dist file>},
 'saved': {<package>: <dist file>},
 'total_cleaned': <size>}
```

CLI Example:

```
salt '*' gentoolkit.eclean_dist destructive=True
```

`salt.modules.gentoolkitmod.`**`eclean_pkg`**(*destructive=False, package_names=False, time_limit=0, exclude_file='/etc/eclean/packages.exclude'*)

Clean obsolete binary packages

destructive Only keep minimum for reinstallation

package_names Protect all versions of installed packages. Only meaningful if used with destructive=True

time_limit <time> Don't delete distfiles files modified since <time> <time> is an amount of time: "1y" is "one year", "2w" is "two weeks", etc. Units are: y (years), m (months), w (weeks), d (days) and h (hours).

exclude_file Path to exclusion file. Default is /etc/eclean/packages.exclude This is the same default eclean-pkg uses. Use None if this file exists and you want to ignore.

Returns a dict containing the cleaned binary packages:

```
{'cleaned': {<dist file>: <size>},
 'total_cleaned': <size>}
```

CLI Example:

```
salt '*' gentoolkit.eclean_pkg destructive=True
```

`salt.modules.gentoolkitmod.`**`glsa_check_list`**(*glsa_list*)

List the status of Gentoo Linux Security Advisories

glsa_list can contain an arbitrary number of GLSA ids, filenames containing GLSAs or the special identifiers 'all' and 'affected'

Returns a dict containing glsa ids with a description, status, and CVEs:

```
{<glsa_id>: {'description': <glsa_description>,
 'status': <glsa status>,
 'CVEs': [<list of CVEs>]}}
```

CLI Example:

```
salt '*' gentoolkit.glsa_check_list 'affected'
```

`salt.modules.gentoolkitmod.`**`revdep_rebuild`**(*lib=None*)

Fix up broken reverse dependencies

lib Search for reverse dependencies for a particular library rather than every library on the system. It can be a full path to a library or basic regular expression.

CLI Example:

```
salt '*' gentoolkit.revdep_rebuild
```

21.9.140 salt.modules.git

Support for the Git SCM

`salt.modules.git.`**`add`**(*cwd, filename, opts='', git_opts='', user=None, password=None, ignore_retcode=False, output_encoding=None*)

Changed in version 2015.8.0: The `--verbose` command line argument is now implied

Interface to git-add(1)

cwd The path to the git checkout

filename The location of the file/directory to add, relative to `cwd`

opts Any additional options to add to the command line, in a single string

---

Note: On the Salt CLI, if the opts are preceded with a dash, it is necessary to precede them with `opts=` (as in the CLI examples below) to avoid causing errors with Salt's own argument parsing.

---

git_opts Any additional options to add to git command itself (not the `add` subcommand), in a single string. This is useful for passing `-c` to run git with temporary changes to the git configuration.

New in version 2017.7.0.

Note: This is only supported in git 1.7.2 and newer.

---

user User under which to run the git command. By default, the command is run by the user under which the minion is running.

password

Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

New in version 2016.3.4.

ignore_retcode [False] If `True`, do not log an error to the minion log if the git command returns a nonzero exit status.

New in version 2015.8.0.

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

Note: This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

New in version 2018.3.1.

CLI Examples:

```
salt myminion git.add /path/to/repo foo/bar.py
salt myminion git.add /path/to/repo foo/bar.py opts='--dry-run'
```

`salt.modules.git.`**`archive`**(*cwd, output, rev='HEAD', prefix=None, git_opts='', user=None, password=None, ignore_retcode=False, output_encoding=None, **kwargs*)

Changed in version 2015.8.0: Returns `True` if successful, raises an error if not.

Interface to git-archive(1), exports a tarball/zip file of the repository

cwd The path to be archived

Note: `git archive` permits a partial archive to be created. Thus, this path does not need to be the root of the git repository. Only the files within the directory specified by `cwd` (and its subdirectories) will be in the resulting archive. For example, if there is a git checkout at `/tmp/foo`, then passing `/tmp/foo/bar` as the `cwd` will result in just the files underneath `/tmp/foo/bar` to be exported as an archive.

output The path of the archive to be created

overwrite [False] Unless set to `True`, Salt will over overwrite an existing archive at the path specified by the `output` argument.

New in version 2015.8.0.

rev [HEAD] The revision from which to create the archive

format Manually specify the file format of the resulting archive. This argument can be omitted, and `git archive` will attempt to guess the archive type (and compression) from the filename. `zip`, `tar`, `tar.gz`, and `tgz` are extensions that are recognized automatically, and git can be configured to support other archive types with the addition of git configuration keys.

See the git-archive(1) manpage explanation of the `--format` argument (as well as the `CONFIGURATION` section of the manpage) for further information.

New in version 2015.8.0.

prefix Prepend `<prefix>` to every filename in the archive. If unspecified, the name of the directory at the top level of the repository will be used as the prefix (e.g. if `cwd` is set to `/foo/bar/baz`, the prefix will be `baz`, and the resulting archive will contain a top-level directory by that name).

Note: The default behavior if the `--prefix` option for `git archive` is not specified is to not prepend a prefix, so Salt's behavior differs slightly from `git archive` in this respect. Use `prefix=''` to create an archive with no prefix.

Changed in version 2015.8.0: The behavior of this argument has been changed slightly. As of this version, it is necessary to include the trailing slash when specifying a prefix, if the prefix is intended to create a top-level directory.

git_opts Any additional options to add to git command itself (not the `archive` subcommand), in a single string. This is useful for passing `-c` to run git with temporary changes to the git configuration.

New in version 2017.7.0.

Note: This is only supported in git 1.7.2 and newer.

user User under which to run the git command. By default, the command is run by the user under which the minion is running.

password

Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

New in version 2016.3.4.

ignore_retcode [False] If `True`, do not log an error to the minion log if the git command returns a nonzero exit status.

New in version 2015.8.0.

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

Note: This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

New in version 2018.3.1.

CLI Example:

```
salt myminion git.archive /path/to/repo /path/to/archive.tar
```

`salt.modules.git.`**`branch`**(*cwd*, *name=None*, *opts=''*, *git_opts=''*, *user=None*, *password=None*, *ignore_retcode=False*, *output_encoding=None*)

Interface to git-branch(1)

cwd The path to the git checkout

name Name of the branch on which to operate. If not specified, the current branch will be assumed.

opts Any additional options to add to the command line, in a single string

---

Note: To create a branch based on something other than HEAD, pass the name of the revision as `opts`. If the revision is in the format `remotename/branch`, then this will also set the remote tracking branch.

Additionally, on the Salt CLI, if the opts are preceded with a dash, it is necessary to precede them with `opts=` (as in the CLI examples below) to avoid causing errors with Salt's own argument parsing.

---

git_opts Any additional options to add to git command itself (not the `branch` subcommand), in a single string. This is useful for passing `-c` to run git with temporary changes to the git configuration.

New in version 2017.7.0.

Note: This is only supported in git 1.7.2 and newer.

---

user User under which to run the git command. By default, the command is run by the user under which the minion is running.

password

Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

New in version 2016.3.4.

ignore_retcode [False] If `True`, do not log an error to the minion log if the git command returns a nonzero exit status.

New in version 2015.8.0.

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

Note: This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

New in version 2018.3.1.

CLI Examples:

```
# Set remote tracking branch
salt myminion git.branch /path/to/repo mybranch opts='--set-upstream-to origin/
↪mybranch'
# Create new branch
salt myminion git.branch /path/to/repo mybranch upstream/somebranch
# Delete branch
salt myminion git.branch /path/to/repo mybranch opts='-d'
# Rename branch (2015.8.0 and later)
salt myminion git.branch /path/to/repo newbranch opts='-m oldbranch'
```

`salt.modules.git.`**`checkout`**(*cwd*, *rev=None*, *force=False*, *opts="*, *git_opts="*, *user=None*, *password=None*, *ignore_retcode=False*, *output_encoding=None*)

Interface to git-checkout(1)

cwd The path to the git checkout

opts Any additional options to add to the command line, in a single string

---

Note: On the Salt CLI, if the opts are preceded with a dash, it is necessary to precede them with `opts=` (as in the CLI examples below) to avoid causing errors with Salt's own argument parsing.

---

git_opts Any additional options to add to git command itself (not the `checkout` subcommand), in a single string. This is useful for passing `-c` to run git with temporary changes to the git configuration.

New in version 2017.7.0.

Note: This is only supported in git 1.7.2 and newer.

---

rev The remote branch or revision to checkout.

Changed in version 2015.8.0: Optional when using `-b` or `-B` in `opts`.

force [False] Force a checkout even if there might be overwritten changes

user User under which to run the git command. By default, the command is run by the user under which the minion is running.

password

Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

New in version 2016.3.4.

ignore_retcode [False] If `True`, do not log an error to the minion log if the git command returns a nonzero exit status.

New in version 2015.8.0.

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

Note: This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

New in version 2018.3.1.

CLI Examples:

```
# Checking out local local revisions
salt myminion git.checkout /path/to/repo somebranch user=jeff
salt myminion git.checkout /path/to/repo opts='testbranch -- conf/file1 file2'
salt myminion git.checkout /path/to/repo rev=origin/mybranch opts='--track'
# Checking out remote revision into new branch
salt myminion git.checkout /path/to/repo upstream/master opts='-b newbranch'
# Checking out current revision into new branch (2015.8.0 and later)
salt myminion git.checkout /path/to/repo opts='-b newbranch'
```

`salt.modules.git.`**`clone`**(*cwd*, *url=None*, *name=None*, *opts=''*, *git_opts=''*, *user=None*, *password=None*, *identity=None*, *https_user=None*, *https_pass=None*, *ignore_retcode=False*, *saltenv='base'*, *output_encoding=None*)

Interface to git-clone(1)

cwd Location of git clone

Changed in version 2015.8.0: If `name` is passed, then the clone will be made *within* this directory.

url The URL of the repository to be cloned

Changed in version 2015.8.0: Argument renamed from `repository` to `url`

name Optional alternate name for the top-level directory to be created by the clone

New in version 2015.8.0.

opts Any additional options to add to the command line, in a single string

git_opts Any additional options to add to git command itself (not the `clone` subcommand), in a single string. This is useful for passing `-c` to run git with temporary changes to the git configuration.

New in version 2017.7.0.

Note: This is only supported in git 1.7.2 and newer.

user User under which to run the git command. By default, the command is run by the user under which the minion is running.

password

Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

New in version 2016.3.4.

identity Path to a private key to use for ssh URLs

Warning: Unless Salt is invoked from the minion using `salt-call`, the key(s) must be passphraseless. For greater security with passphraseless private keys, see the sshd(8) manpage for information on securing the keypair from the remote side in the `authorized_keys` file.

Changed in version 2015.8.7: Salt will no longer attempt to use passphrase-protected keys unless invoked from the minion using `salt-call`, to prevent blocking waiting for user input.

Key can also be specified as a SaltStack file server URL, eg. salt://location/identity_file

Changed in version 2016.3.0.

https_user Set HTTP Basic Auth username. Only accepted for HTTPS URLs.

New in version 20515.5.0.

https_pass Set HTTP Basic Auth password. Only accepted for HTTPS URLs.

New in version 2015.5.0.

ignore_retcode [False] If `True`, do not log an error to the minion log if the git command returns a nonzero exit status.

New in version 2015.8.0.

saltenv The default salt environment to pull sls files from

New in version 2016.3.1.

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

> **Note:** This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

New in version 2018.3.1.

CLI Example:

```
salt myminion git.clone /path/to/repo_parent_dir git://github.com/saltstack/salt.
↪git
```

`salt.modules.git.`**`commit`**(*cwd, message, opts='', git_opts='', user=None, password=None, filename=None, ignore_retcode=False, output_encoding=None*)

Interface to git-commit(1)

cwd The path to the git checkout

message Commit message

opts Any additional options to add to the command line, in a single string. These opts will be added to the end of the git command being run.

> **Note:** On the Salt CLI, if the opts are preceded with a dash, it is necessary to precede them with `opts=` (as in the CLI examples below) to avoid causing errors with Salt's own argument parsing.
>
> The `-m` option should not be passed here, as the commit message will be defined by the `message` argument.

git_opts Any additional options to add to git command itself (not the `commit` subcommand), in a single string. This is useful for passing `-c` to run git with temporary changes to the git configuration.

New in version 2017.7.0.

> **Note:** This is only supported in git 1.7.2 and newer.

user User under which to run the git command. By default, the command is run by the user under which the minion is running.

password

Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

New in version 2016.3.4.

filename The location of the file/directory to commit, relative to `cwd`. This argument is optional, and can be used to commit a file without first staging it.

Note: This argument only works on files which are already tracked by the git repository.

New in version 2015.8.0.

ignore_retcode [False] If `True`, do not log an error to the minion log if the git command returns a nonzero exit status.

New in version 2015.8.0.

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

Note: This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

New in version 2018.3.1.

CLI Examples:

```
salt myminion git.commit /path/to/repo 'The commit message'
salt myminion git.commit /path/to/repo 'The commit message' filename=foo/bar.py
```

`salt.modules.git.`**`config_get`**(*key*, *cwd=None*, *user=None*, *password=None*, *ignore_retcode=False*, *output_encoding=None*, ***kwargs*)

Get the value of a key in the git configuration file

key The name of the configuration key to get

Changed in version 2015.8.0: Argument renamed from `setting_name` to `key`

cwd The path to the git checkout

Changed in version 2015.8.0: Now optional if `global` is set to `True`

global [False] If `True`, query the global git configuration. Otherwise, only the local git configuration will be queried.

New in version 2015.8.0.

all [False] If `True`, return a list of all values set for `key`. If the key does not exist, `None` will be returned.

New in version 2015.8.0.

user User under which to run the git command. By default, the command is run by the user under which the minion is running.

password

Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

New in version 2016.3.4.

ignore_retcode [False] If `True`, do not log an error to the minion log if the git command returns a nonzero exit status.

New in version 2015.8.0.

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

Note: This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

New in version 2018.3.1.

CLI Examples:

```
salt myminion git.config_get user.name cwd=/path/to/repo
salt myminion git.config_get user.email global=True
salt myminion git.config_get core.gitproxy cwd=/path/to/repo all=True
```

`salt.modules.git.`**`config_get_regexp`**(*key*, *value_regex=None*, *cwd=None*, *user=None*, *password=None*, *ignore_retcode=False*, *output_encoding=None*, ***kwargs*)

New in version 2015.8.0.

Get the value of a key or keys in the git configuration file using regexes for more flexible matching. The return data is a dictionary mapping keys to lists of values matching the `value_regex`. If no values match, an empty dictionary will be returned.

key Regex on which key names will be matched

value_regex If specified, return all values matching this regex. The return data will be a dictionary mapping keys to lists of values matching the regex.

Important: Only values matching the `value_regex` will be part of the return data. So, if `key` matches a multivar, then it is possible that not all of the values will be returned. To get all values set for a multivar, simply omit the `value_regex` argument.

---

cwd The path to the git checkout

global [False] If `True`, query the global git configuration. Otherwise, only the local git configuration will be queried.

user User under which to run the git command. By default, the command is run by the user under which the minion is running.

password

> Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

New in version 2016.3.4.

ignore_retcode [False] If `True`, do not log an error to the minion log if the git command returns a nonzero exit status.

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

Note: This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

New in version 2018.3.1.

CLI Examples:

```
# Matches any values for key 'foo.bar'
salt myminion git.config_get_regexp /path/to/repo foo.bar
# Matches any value starting with 'baz' set for key 'foo.bar'
salt myminion git.config_get_regexp /path/to/repo foo.bar 'baz.*'
# Matches any key starting with 'user.'
salt myminion git.config_get_regexp '^user\.' global=True
```

`salt.modules.git.`**`config_set`**(*key, value=None, multivar=None, cwd=None, user=None, password=None, ignore_retcode=False, output_encoding=None, **kwargs*)

Changed in version 2015.8.0: Return the value(s) of the key being set

Set a key in the git configuration file

cwd The path to the git checkout. Must be an absolute path, or the word `global` to indicate that a global key should be set.

Changed in version 2014.7.0: Made `cwd` argument optional if `is_global=True`

key The name of the configuration key to set

Changed in version 2015.8.0: Argument renamed from `setting_name` to `key`

value The value to set for the specified key. Incompatible with the `multivar` argument.

Changed in version 2015.8.0: Argument renamed from `setting_value` to `value`

add [False] Add a value to a key, creating/updating a multivar

New in version 2015.8.0.

multivar Set a multivar all at once. Values can be comma-separated or passed as a Python list. Incompatible with the `value` argument.

New in version 2015.8.0.

user User under which to run the git command. By default, the command is run by the user under which the minion is running.

password

Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

New in version 2016.3.4.

ignore_retcode [False] If `True`, do not log an error to the minion log if the git command returns a nonzero exit status.

New in version 2015.8.0.

global [False] If `True`, set a global variable

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

Note: This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

New in version 2018.3.1.

CLI Examples:

```
salt myminion git.config_set user.email me@example.com cwd=/path/to/repo
salt myminion git.config_set user.email foo@bar.com global=True
```

`salt.modules.git.`**`config_unset`**(*key*, *value_regex=None*, *cwd=None*, *user=None*, *password=None*, *ignore_retcode=False*, *output_encoding=None*, ***kwargs*)

New in version 2015.8.0.

Unset a key in the git configuration file

cwd The path to the git checkout. Must be an absolute path, or the word `global` to indicate that a global key should be unset.

key The name of the configuration key to unset

value_regex Regular expression that matches exactly one key, used to delete a single value from a multivar. Ignored if `all` is set to `True`.

all [False] If `True` unset all values for a multivar. If `False`, and `key` is a multivar, an error will be raised.

global [False] If `True`, unset set a global variable. Otherwise, a local variable will be unset.

user User under which to run the git command. By default, the command is run by the user under which the minion is running.

password

Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

New in version 2016.3.4.

ignore_retcode [False] If `True`, do not log an error to the minion log if the git command returns a nonzero exit status.

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

Note: This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

New in version 2018.3.1.

CLI Example:

```
salt myminion git.config_unset /path/to/repo foo.bar
salt myminion git.config_unset /path/to/repo foo.bar all=True
```

`salt.modules.git.`**`current_branch`**(*cwd*, *user=None*, *password=None*, *ignore_retcode=False*, *output_encoding=None*)

Returns the current branch name of a local checkout. If HEAD is detached, return the SHA1 of the revision which is currently checked out.

cwd The path to the git checkout

user User under which to run the git command. By default, the command is run by the user under which the minion is running.

password

Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

New in version 2016.3.4.

ignore_retcode [False] If `True`, do not log an error to the minion log if the git command returns a nonzero exit status.

New in version 2015.8.0.

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

Note: This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

New in version 2018.3.1.

CLI Example:

```
salt myminion git.current_branch /path/to/repo
```

`salt.modules.git.`**`describe`**(*cwd*, *rev='HEAD'*, *user=None*, *password=None*, *ignore_retcode=False*, *output_encoding=None*)

Returns the git-describe(1) string (or the SHA1 hash if there are no tags) for the given revision.

cwd The path to the git checkout

rev [HEAD] The revision to describe

user User under which to run the git command. By default, the command is run by the user under which the minion is running.

password

Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

New in version 2016.3.4.

ignore_retcode [False] If `True`, do not log an error to the minion log if the git command returns a nonzero exit status.

New in version 2015.8.0.

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

Note: This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

New in version 2018.3.1.

CLI Examples:

```
salt myminion git.describe /path/to/repo
salt myminion git.describe /path/to/repo develop
```

`salt.modules.git.`**`diff`**(*cwd*, *item1=None*, *item2=None*, *opts=''*, *git_opts=''*, *user=None*, *password=None*, *no_index=False*, *cached=False*, *paths=None*, *output_encoding=None*)

New in version 2015.8.12,2016.3.3,2016.11.0.

Interface to git-diff(1)

cwd The path to the git checkout

item1 and item2 Revision(s) to pass to the `git diff` command. One or both of these arguments may be ignored if some of the options below are set to `True`. When `cached` is `False`, and no revisions are passed to this function, then the current working tree will be compared against the index (i.e. unstaged changes). When two revisions are passed, they will be compared to each other.

opts Any additional options to add to the command line, in a single string

Note: On the Salt CLI, if the opts are preceded with a dash, it is necessary to precede them with `opts=` (as in the CLI examples below) to avoid causing errors with Salt's own argument parsing.

git_opts Any additional options to add to git command itself (not the `diff` subcommand), in a single string. This is useful for passing `-c` to run git with temporary changes to the git configuration.

New in version 2017.7.0.

Note: This is only supported in git 1.7.2 and newer.

user User under which to run the git command. By default, the command is run by the user under which the minion is running.

password

Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

New in version 2016.3.4.

no_index [False] When it is necessary to diff two files in the same repo against each other, and not diff two different revisions, set this option to `True`. If this is left `False` in these instances, then a normal `git diff` will be performed against the index (i.e. unstaged changes), and files in the `paths` option will be used to narrow down the diff output.

Note: Requires Git 1.5.1 or newer. Additionally, when set to `True`, `item1` and `item2` will be ignored.

cached [False] If `True`, compare staged changes to `item1` (if specified), otherwise compare them to the most recent commit.

Note: `item2` is ignored if this option is is set to `True`.

paths File paths to pass to the `git diff` command. Can be passed as a comma-separated list or a Python list.

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

Note: This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

New in version 2018.3.1.

CLI Example:

```
# Perform diff against the index (staging area for next commit)
salt myminion git.diff /path/to/repo
# Compare staged changes to the most recent commit
salt myminion git.diff /path/to/repo cached=True
# Compare staged changes to a specific revision
salt myminion git.diff /path/to/repo mybranch cached=True
# Perform diff against the most recent commit (includes staged changes)
salt myminion git.diff /path/to/repo HEAD
# Diff two commits
salt myminion git.diff /path/to/repo abcdef1 aabbccd
# Diff two commits, only showing differences in the specified paths
salt myminion git.diff /path/to/repo abcdef1 aabbccd paths=path/to/file1,path/to/
↪file2
# Diff two files with one being outside the working tree
salt myminion git.diff /path/to/repo no_index=True paths=path/to/file1,/absolute/
↪path/to/file2
```

`salt.modules.git.`**`discard_local_changes`**(*cwd, path='.', user=None, password=None, ignore_retcode=False, output_encoding=None*)

New in version 2019.2.0.

Runs a `git checkout -- <path>` from the directory specified by `cwd`.

cwd The path to the git checkout

path path relative to cwd (defaults to `.`)

user User under which to run the git command. By default, the command is run by the user under which the minion is running.

password Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

ignore_retcode [False] If `True`, do not log an error to the minion log if the git command returns a nonzero exit status.

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

Note: This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

CLI Example:

```
salt myminion git.discard_local_changes /path/to/repo
salt myminion git.discard_local_changes /path/to/repo path=foo
```

`salt.modules.git.`**`fetch`**(*cwd, remote=None, force=False, refspecs=None, opts='', git_opts='', user=None, password=None, identity=None, ignore_retcode=False, saltenv='base', output_encoding=None*)

Changed in version 2015.8.2: Return data is now a dictionary containing information on branches and tags that were added/updated

Interface to git-fetch(1)

cwd The path to the git checkout

remote Optional remote name to fetch. If not passed, then git will use its default behavior (as detailed in git-fetch(1)).

New in version 2015.8.0.

force Force the fetch even when it is not a fast-forward.

New in version 2015.8.0.

refspecs Override the refspec(s) configured for the remote with this argument. Multiple refspecs can be passed, comma-separated.

New in version 2015.8.0.

opts Any additional options to add to the command line, in a single string

Note: On the Salt CLI, if the opts are preceded with a dash, it is necessary to precede them with `opts=` (as in the CLI examples below) to avoid causing errors with Salt's own argument parsing.

git_opts Any additional options to add to git command itself (not the `fetch` subcommand), in a single string. This is useful for passing `-c` to run git with temporary changes to the git configuration.

New in version 2017.7.0.

Note: This is only supported in git 1.7.2 and newer.

user User under which to run the git command. By default, the command is run by the user under which the minion is running.

password

Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

New in version 2016.3.4.

identity Path to a private key to use for ssh URLs

Warning: Unless Salt is invoked from the minion using `salt-call`, the key(s) must be passphraseless. For greater security with passphraseless private keys, see the sshd(8) manpage for information on securing the keypair from the remote side in the `authorized_keys` file.

Changed in version 2015.8.7: Salt will no longer attempt to use passphrase-protected keys unless invoked from the minion using `salt-call`, to prevent blocking waiting for user input.

Key can also be specified as a SaltStack file server URL, eg. salt://location/identity_file

Changed in version 2016.3.0.

ignore_retcode [False] If `True`, do not log an error to the minion log if the git command returns a nonzero exit status.

New in version 2015.8.0.

saltenv The default salt environment to pull sls files from

New in version 2016.3.1.

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

Note: This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

New in version 2018.3.1.

CLI Example:

```
salt myminion git.fetch /path/to/repo upstream
salt myminion git.fetch /path/to/repo identity=/root/.ssh/id_rsa
```

`salt.modules.git.init`(*cwd*, *bare=False*, *template=None*, *separate_git_dir=None*, *shared=None*, *opts=''*, *git_opts=''*, *user=None*, *password=None*, *ignore_retcode=False*, *output_encoding=None*)

Interface to git-init(1)

cwd The path to the directory to be initialized

bare [False] If `True`, init a bare repository

New in version 2015.8.0.

template Set this argument to specify an alternate template directory

New in version 2015.8.0.

separate_git_dir Set this argument to specify an alternate `$GIT_DIR`

New in version 2015.8.0.

shared Set sharing permissions on git repo. See git-init(1) for more details.

New in version 2015.8.0.

opts Any additional options to add to the command line, in a single string

---

Note: On the Salt CLI, if the opts are preceded with a dash, it is necessary to precede them with `opts=` (as in the CLI examples below) to avoid causing errors with Salt's own argument parsing.

---

git_opts Any additional options to add to git command itself (not the `init` subcommand), in a single string. This is useful for passing `-c` to run git with temporary changes to the git configuration.

New in version 2017.7.0.

Note: This is only supported in git 1.7.2 and newer.

---

user User under which to run the git command. By default, the command is run by the user under which the minion is running.

password

Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

New in version 2016.3.4.

ignore_retcode [False] If `True`, do not log an error to the minion log if the git command returns a nonzero exit status.

New in version 2015.8.0.

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

Note: This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

New in version 2018.3.1.

CLI Examples:

```
salt myminion git.init /path/to/repo
# Init a bare repo (before 2015.8.0)
salt myminion git.init /path/to/bare/repo.git opts='--bare'
# Init a bare repo (2015.8.0 and later)
salt myminion git.init /path/to/bare/repo.git bare=True
```

`salt.modules.git.`**`is_worktree`**(*cwd*, *user=None*, *password=None*, *output_encoding=None*)

New in version 2015.8.0.

This function will attempt to determine if `cwd` is part of a worktree by checking its `.git` to see if it is a file containing a reference to another gitdir.

cwd path to the worktree to be removed

user User under which to run the git command. By default, the command is run by the user under which the minion is running.

password

Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

New in version 2016.3.4.

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

Note: This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

New in version 2018.3.1.

CLI Example:

```
salt myminion git.is_worktree /path/to/repo
```

`salt.modules.git.`**`list_branches`**(*cwd*, *remote=False*, *user=None*, *password=None*, *ignore_retcode=False*, *output_encoding=None*)

New in version 2015.8.0.

Return a list of branches

cwd The path to the git checkout

remote [False] If `True`, list remote branches. Otherwise, local branches will be listed.

> **Warning:** This option will only return remote branches of which the local checkout is aware, use `git.fetch` to update remotes.

user User under which to run the git command. By default, the command is run by the user under which the minion is running.

password

Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

New in version 2016.3.4.

ignore_retcode [False] If `True`, do not log an error to the minion log if the git command returns a nonzero exit status.

New in version 2015.8.0.

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

> **Note:** This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

New in version 2018.3.1.

CLI Examples:

```
salt myminion git.list_branches /path/to/repo
salt myminion git.list_branches /path/to/repo remote=True
```

`salt.modules.git.`**`list_tags`**(*cwd*, *user=None*, *password=None*, *ignore_retcode=False*, *output_encoding=None*)

New in version 2015.8.0.

Return a list of tags

cwd The path to the git checkout

user User under which to run the git command. By default, the command is run by the user under which the minion is running.

password

Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

New in version 2016.3.4.

ignore_retcode [False] If `True`, do not log an error to the minion log if the git command returns a nonzero exit status.

New in version 2015.8.0.

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

Note: This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

New in version 2018.3.1.

CLI Examples:

```
salt myminion git.list_tags /path/to/repo
```

`salt.modules.git.`**`list_worktrees`**`(`*cwd*, *stale=False*, *user=None*, *password=None*, *output_encoding=None*, ***kwargs*`)`

New in version 2015.8.0.

Returns information on worktrees

Changed in version 2015.8.4: Version 2.7.0 added the `list` subcommand to git-worktree(1) which provides a lot of additional information. The return data has been changed to include this information, even for pre-2.7.0 versions of git. In addition, if a worktree has a detached head, then any tags which point to the worktree's HEAD will be included in the return data.

Note: By default, only worktrees for which the worktree directory is still present are returned, but this can be changed using the `all` and `stale` arguments (described below).

cwd The path to the git checkout

user User under which to run the git command. By default, the command is run by the user under which the minion is running.

password

Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

New in version 2016.3.4.

all [False] If `True`, then return all worktrees tracked under $GIT_DIR/worktrees, including ones for which the gitdir is no longer present.

stale [False] If `True`, return *only* worktrees whose gitdir is no longer present.

Note: Only one of `all` and `stale` can be set to `True`.

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

Note: This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

New in version 2018.3.1.

CLI Examples:

```
salt myminion git.list_worktrees /path/to/repo
salt myminion git.list_worktrees /path/to/repo all=True
salt myminion git.list_worktrees /path/to/repo stale=True
```

`salt.modules.git.`**`ls_remote`**(*cwd=None*, *remote='origin'*, *ref=None*, *opts="*, *git_opts="*, *user=None*, *password=None*, *identity=None*, *https_user=None*, *https_pass=None*, *ignore_retcode=False*, *output_encoding=None*, *saltenv='base'*)

Interface to git-ls-remote(1). Returns the upstream hash for a remote reference.

cwd The path to the git checkout. Optional (and ignored if present) when `remote` is set to a URL instead of a remote name.

remote [origin] The name of the remote to query. Can be the name of a git remote (which exists in the git checkout defined by the `cwd` parameter), or the URL of a remote repository.

Changed in version 2015.8.0: Argument renamed from `repository` to `remote`

ref The name of the ref to query. Optional, if not specified, all refs are returned. Can be a branch or tag name, or the full name of the reference (for example, to get the hash for a Github pull request number 1234, `ref` can be set to `refs/pull/1234/head`

Changed in version 2015.8.0: Argument renamed from `branch` to `ref`

Changed in version 2015.8.4: Defaults to returning all refs instead of master.

opts Any additional options to add to the command line, in a single string

New in version 2015.8.0.

git_opts Any additional options to add to git command itself (not the `ls-remote` subcommand), in a single string. This is useful for passing `-c` to run git with temporary changes to the git configuration.

New in version 2017.7.0.

Note: This is only supported in git 1.7.2 and newer.

---

user User under which to run the git command. By default, the command is run by the user under which the minion is running.

password

Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

New in version 2016.3.4.

identity Path to a private key to use for ssh URLs

Warning: Unless Salt is invoked from the minion using `salt-call`, the key(s) must be passphraseless. For greater security with passphraseless private keys, see the sshd(8) manpage for information on securing the keypair from the remote side in the `authorized_keys` file.

Changed in version 2015.8.7: Salt will no longer attempt to use passphrase-protected keys unless invoked from the minion using `salt-call`, to prevent blocking waiting for user input.

Key can also be specified as a SaltStack file server URL, eg. salt://location/identity_file

Changed in version 2016.3.0.

https_user Set HTTP Basic Auth username. Only accepted for HTTPS URLs.

New in version 2015.5.0.

https_pass Set HTTP Basic Auth password. Only accepted for HTTPS URLs.

New in version 2015.5.0.

ignore_retcode [False] If `True`, do not log an error to the minion log if the git command returns a nonzero exit status.

New in version 2015.8.0.

saltenv The default salt environment to pull sls files from

New in version 2016.3.1.

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

Note: This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

New in version 2018.3.1.

CLI Example:

```
salt myminion git.ls_remote /path/to/repo origin master
salt myminion git.ls_remote remote=https://mydomain.tld/repo.git ref=mytag opts='-
↪-tags'
```

`salt.modules.git.merge`(*cwd*, *rev=None*, *opts=''*, *git_opts=''*, *user=None*, *password=None*, *identity=None*, *ignore_retcode=False*, *output_encoding=None*, ***kwargs*)

Interface to git-merge(1)

cwd The path to the git checkout

rev Revision to merge into the current branch. If not specified, the remote tracking branch will be merged.

New in version 2015.8.0.

opts Any additional options to add to the command line, in a single string

Note: On the Salt CLI, if the opts are preceded with a dash, it is necessary to precede them with `opts=` (as in the CLI examples below) to avoid causing errors with Salt's own argument parsing.

git_opts Any additional options to add to git command itself (not the `merge` subcommand), in a single string. This is useful for passing `-c` to run git with temporary changes to the git configuration.

New in version 2017.7.0.

Note: This is only supported in git 1.7.2 and newer.

user User under which to run the git command. By default, the command is run by the user under which the minion is running.

password

Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

New in version 2016.3.4.

identity Path to a private key to use for ssh URLs. Salt will not attempt to use passphrase-protected keys unless invoked from the minion using `salt-call`, to prevent blocking waiting for user input. Key can also be specified as a SaltStack file server URL, eg. `salt://location/identity_file`.

Note: For greater security with passphraseless private keys, see the sshd(8) manpage for information on securing the keypair from the remote side in the `authorized_keys` file.

New in version 2018.3.5,2019.2.1,3000.

ignore_retcode [False] If `True`, do not log an error to the minion log if the git command returns a nonzero exit status.

New in version 2015.8.0.

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

Note: This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

New in version 2018.3.1.

CLI Example:

```
# Fetch first...
salt myminion git.fetch /path/to/repo
# ... then merge the remote tracking branch
salt myminion git.merge /path/to/repo
# .. or merge another rev
salt myminion git.merge /path/to/repo rev=upstream/foo
```

`salt.modules.git.`**`merge_base`**`(`*cwd, refs=None, octopus=False, is_ancestor=False, independent=False, fork_point=None, opts='', git_opts='', user=None, password=None, ignore_retcode=False, output_encoding=None, **kwargs*`)`

New in version 2015.8.0.

Interface to git-merge-base(1).

cwd The path to the git checkout

refs Any refs/commits to check for a merge base. Can be passed as a comma-separated list or a Python list.

all [False] Return a list of all matching merge bases. Not compatible with any of the below options except for `octopus`.

octopus [False] If `True`, then this function will determine the best common ancestors of all specified commits, in preparation for an n-way merge. See here for a description of how these bases are determined.

Set `all` to `True` with this option to return all computed merge bases, otherwise only the "best" will be returned.

is_ancestor [False] If `True`, then instead of returning the merge base, return a boolean telling whether or not the first commit is an ancestor of the second commit.

Note: This option requires two commits to be passed.

Changed in version 2015.8.2: Works properly in git versions older than 1.8.0, where the `--is-ancestor` CLI option is not present.

independent [False] If `True`, this function will return the IDs of the refs/commits passed which cannot be reached by another commit.

fork_point If passed, then this function will return the commit where the commit diverged from the ref specified by `fork_point`. If no fork point is found, `None` is returned.

Note: At most one commit is permitted to be passed if a `fork_point` is specified. If no commits are passed, then `HEAD` is assumed.

---

opts Any additional options to add to the command line, in a single string

---

Note: On the Salt CLI, if the opts are preceded with a dash, it is necessary to precede them with `opts=` (as in the CLI examples below) to avoid causing errors with Salt's own argument parsing.

This option should not be necessary unless new CLI arguments are added to git-merge-base(1) and are not yet supported in Salt.

---

git_opts Any additional options to add to git command itself (not the `merge-base` subcommand), in a single string. This is useful for passing `-c` to run git with temporary changes to the git configuration.

New in version 2017.7.0.

Note: This is only supported in git 1.7.2 and newer.

---

user User under which to run the git command. By default, the command is run by the user under which the minion is running.

password

> Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

New in version 2016.3.4.

ignore_retcode [False] if `True`, do not log an error to the minion log if the git command returns a nonzero exit status.

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

Note: This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

New in version 2018.3.1.

CLI Examples:

```
salt myminion git.merge_base /path/to/repo HEAD upstream/mybranch
salt myminion git.merge_base /path/to/repo 8f2e542,4ad8cab,cdc9886 octopus=True
salt myminion git.merge_base /path/to/repo refs=8f2e542,4ad8cab,cdc9886␣
↪independent=True
salt myminion git.merge_base /path/to/repo refs=8f2e542,4ad8cab is_ancestor=True
salt myminion git.merge_base /path/to/repo fork_point=upstream/master
salt myminion git.merge_base /path/to/repo refs=mybranch fork_point=upstream/
↪master
```

`salt.modules.git.`**`merge_tree`**(*cwd*, *ref1*, *ref2*, *base=None*, *user=None*, *password=None*, *ignore_retcode=False*, *output_encoding=None*)

New in version 2015.8.0.

Interface to git-merge-tree(1), shows the merge results and conflicts from a 3-way merge without touching the index.

cwd The path to the git checkout

ref1 First ref/commit to compare

ref2 Second ref/commit to compare

base The base tree to use for the 3-way-merge. If not provided, then *`git.merge_base`* will be invoked on `ref1` and `ref2` to determine the merge base to use.

user User under which to run the git command. By default, the command is run by the user under which the minion is running.

password

> Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

New in version 2016.3.4.

ignore_retcode [False] if `True`, do not log an error to the minion log if the git command returns a nonzero exit status.

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

Note: This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

New in version 2018.3.1.

CLI Examples:

```
salt myminion git.merge_tree /path/to/repo HEAD upstream/dev
salt myminion git.merge_tree /path/to/repo HEAD upstream/dev base=aaf3c3d
```

`salt.modules.git.`**`pull`**(*cwd*, *opts=''*, *git_opts=''*, *user=None*, *password=None*, *identity=None*, *ignore_retcode=False*, *saltenv='base'*, *output_encoding=None*)

Interface to git-pull(1)

cwd The path to the git checkout

opts Any additional options to add to the command line, in a single string

---

Note: On the Salt CLI, if the opts are preceded with a dash, it is necessary to precede them with `opts=` (as in the CLI examples below) to avoid causing errors with Salt's own argument parsing.

---

git_opts Any additional options to add to git command itself (not the `pull` subcommand), in a single string. This is useful for passing `-c` to run git with temporary changes to the git configuration.

New in version 2017.7.0.

Note: This is only supported in git 1.7.2 and newer.

---

user User under which to run the git command. By default, the command is run by the user under which the minion is running.

password

> Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

New in version 2016.3.4.

identity Path to a private key to use for ssh URLs

> **Warning:** Unless Salt is invoked from the minion using `salt-call`, the key(s) must be passphraseless. For greater security with passphraseless private keys, see the sshd(8) manpage for information on securing the keypair from the remote side in the `authorized_keys` file.

Changed in version 2015.8.7: Salt will no longer attempt to use passphrase-protected keys unless invoked from the minion using `salt-call`, to prevent blocking waiting for user input.

Key can also be specified as a SaltStack file server URL, eg. salt://location/identity_file

Changed in version 2016.3.0.

ignore_retcode [False] If `True`, do not log an error to the minion log if the git command returns a nonzero exit status.

New in version 2015.8.0.

saltenv The default salt environment to pull sls files from

New in version 2016.3.1.

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

Note: This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

New in version 2018.3.1.

CLI Example:

```
salt myminion git.pull /path/to/repo opts='--rebase origin master'
```

`salt.modules.git.`**`push`**(*cwd*, *remote=None*, *ref=None*, *opts="*, *git_opts="*, *user=None*, *password=None*, *identity=None*, *ignore_retcode=False*, *saltenv='base'*, *output_encoding=None*, ***kwargs*)

Interface to git-push(1)

cwd The path to the git checkout

remote Name of the remote to which the ref should being pushed

New in version 2015.8.0.

ref [master] Name of the ref to push

Note: Being a refspec, this argument can include a colon to define local and remote ref names.

opts Any additional options to add to the command line, in a single string

Note: On the Salt CLI, if the opts are preceded with a dash, it is necessary to precede them with `opts=` (as in the CLI examples below) to avoid causing errors with Salt's own argument parsing.

git_opts Any additional options to add to git command itself (not the `push` subcommand), in a single string. This is useful for passing `-c` to run git with temporary changes to the git configuration.

New in version 2017.7.0.

Note: This is only supported in git 1.7.2 and newer.

user User under which to run the git command. By default, the command is run by the user under which the minion is running.

password

Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

New in version 2016.3.4.

identity Path to a private key to use for ssh URLs

Warning: Unless Salt is invoked from the minion using `salt-call`, the key(s) must be passphraseless. For greater security with passphraseless private keys, see the sshd(8) manpage for information on securing the keypair from the remote side in the `authorized_keys` file.

Changed in version 2015.8.7: Salt will no longer attempt to use passphrase-protected keys unless invoked from the minion using `salt-call`, to prevent blocking waiting for user input.

Key can also be specified as a SaltStack file server URL, eg. salt://location/identity_file

Changed in version 2016.3.0.

ignore_retcode [False] If `True`, do not log an error to the minion log if the git command returns a nonzero exit status.

New in version 2015.8.0.

saltenv The default salt environment to pull sls files from

New in version 2016.3.1.

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

Note: This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

New in version 2018.3.1.

CLI Example:

```
# Push master as origin/master
salt myminion git.push /path/to/repo origin master
# Push issue21 as upstream/develop
salt myminion git.push /path/to/repo upstream issue21:develop
# Delete remote branch 'upstream/temp'
salt myminion git.push /path/to/repo upstream :temp
```

`salt.modules.git.`**`rebase`**`(`*cwd, rev='master', opts='', git_opts='', user=None, password=None, ignore_retcode=False, output_encoding=None*`)`

Interface to git-rebase(1)

cwd The path to the git checkout

rev [master] The revision to rebase onto the current branch

opts Any additional options to add to the command line, in a single string

---

Note: On the Salt CLI, if the opts are preceded with a dash, it is necessary to precede them with `opts=` (as in the CLI examples below) to avoid causing errors with Salt's own argument parsing.

---

git_opts Any additional options to add to git command itself (not the `rebase` subcommand), in a single string. This is useful for passing `-c` to run git with temporary changes to the git configuration.

New in version 2017.7.0.

Note: This is only supported in git 1.7.2 and newer.

---

user User under which to run the git command. By default, the command is run by the user under which the minion is running.

password

Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

New in version 2016.3.4.

ignore_retcode [False] If `True`, do not log an error to the minion log if the git command returns a nonzero exit status.

New in version 2015.8.0.

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

Note: This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

New in version 2018.3.1.

CLI Example:

```
salt myminion git.rebase /path/to/repo master
salt myminion git.rebase /path/to/repo 'origin master'
salt myminion git.rebase /path/to/repo origin/master opts='--onto newbranch'
```

`salt.modules.git.`**`remote_get`**(*cwd, remote='origin', user=None, password=None, redact_auth=True, ignore_retcode=False, output_encoding=None*)

Get the fetch and push URL for a specific remote

cwd The path to the git checkout

remote [origin] Name of the remote to query

user User under which to run the git command. By default, the command is run by the user under which the minion is running.

password

Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

New in version 2016.3.4.

redact_auth [True] Set to `False` to include the username/password if the remote uses HTTPS Basic Auth. Otherwise, this information will be redacted.

Warning: Setting this to `False` will not only reveal any HTTPS Basic Auth that is configured, but the return data will also be written to the job cache. When possible, it is recommended to use SSH for authentication.

New in version 2015.5.6.

ignore_retcode [False] If `True`, do not log an error to the minion log if the git command returns a nonzero exit status.

New in version 2015.8.0.

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

Note: This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

New in version 2018.3.1.

CLI Examples:

```
salt myminion git.remote_get /path/to/repo
salt myminion git.remote_get /path/to/repo upstream
```

`salt.modules.git.`**`remote_refs`**(*url*, *heads=False*, *tags=False*, *user=None*, *password=None*, *identity=None*, *https_user=None*, *https_pass=None*, *ignore_retcode=False*, *output_encoding=None*, *saltenv='base'*, ***kwargs*)

New in version 2015.8.0.

Return the remote refs for the specified URL by running `git ls-remote`.

url URL of the remote repository

filter Optionally provide a ref name to `git ls-remote`. This can be useful to make this function run faster on repositories with many branches/tags.

New in version 2019.2.0.

heads [False] Restrict output to heads. Can be combined with `tags`.

tags [False] Restrict output to tags. Can be combined with `heads`.

user User under which to run the git command. By default, the command is run by the user under which the minion is running.

password

> Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

New in version 2016.3.4.

identity Path to a private key to use for ssh URLs

> **Warning:** Unless Salt is invoked from the minion using `salt-call`, the key(s) must be passphraseless. For greater security with passphraseless private keys, see the sshd(8) manpage for information on securing the keypair from the remote side in the `authorized_keys` file.

Changed in version 2015.8.7: Salt will no longer attempt to use passphrase-protected keys unless invoked from the minion using `salt-call`, to prevent blocking waiting for user input.

Key can also be specified as a SaltStack file server URL, eg. salt://location/identity_file

Changed in version 2016.3.0.

https_user Set HTTP Basic Auth username. Only accepted for HTTPS URLs.

https_pass Set HTTP Basic Auth password. Only accepted for HTTPS URLs.

ignore_retcode [False] If `True`, do not log an error to the minion log if the git command returns a nonzero exit status.

saltenv The default salt environment to pull sls files from

New in version 2016.3.1.

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

> **Note:** This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

New in version 2018.3.1.

CLI Example:

```
salt myminion git.remote_refs https://github.com/saltstack/salt.git
salt myminion git.remote_refs https://github.com/saltstack/salt.git filter=develop
```

`salt.modules.git.`**`remote_set`**(*cwd*, *url*, *remote='origin'*, *user=None*, *password=None*, *https_user=None*, *https_pass=None*, *push_url=None*, *push_https_user=None*, *push_https_pass=None*, *ignore_retcode=False*, *output_encoding=None*)

cwd The path to the git checkout

url Remote URL to set

remote [origin] Name of the remote to set

push_url If unset, the push URL will be identical to the fetch URL.

New in version 2015.8.0.

user User under which to run the git command. By default, the command is run by the user under which the minion is running.

password

Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

New in version 2016.3.4.

https_user Set HTTP Basic Auth username. Only accepted for HTTPS URLs.

New in version 2015.5.0.

https_pass Set HTTP Basic Auth password. Only accepted for HTTPS URLs.

New in version 2015.5.0.

push_https_user Set HTTP Basic Auth user for `push_url`. Ignored if `push_url` is unset. Only accepted for HTTPS URLs.

New in version 2015.8.0.

push_https_pass Set HTTP Basic Auth password for `push_url`. Ignored if `push_url` is unset. Only accepted for HTTPS URLs.

New in version 2015.8.0.

ignore_retcode [False] If `True`, do not log an error to the minion log if the git command returns a nonzero exit status.

New in version 2015.8.0.

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

Note: This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

New in version 2018.3.1.

CLI Examples:

```
salt myminion git.remote_set /path/to/repo git@github.com:user/repo.git
salt myminion git.remote_set /path/to/repo git@github.com:user/repo.git␣
↪remote=upstream
salt myminion git.remote_set /path/to/repo https://github.com/user/repo.git␣
↪remote=upstream push_url=git@github.com:user/repo.git
```

`salt.modules.git.`**`remotes`**(*cwd*, *user=None*, *password=None*, *redact_auth=True*, *ignore_retcode=False*, *output_encoding=None*)

Get fetch and push URLs for each remote in a git checkout

cwd The path to the git checkout

user User under which to run the git command. By default, the command is run by the user under which the minion is running.

password

> Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

New in version 2016.3.4.

redact_auth [True] Set to `False` to include the username/password for authenticated remotes in the return data. Otherwise, this information will be redacted.

> **Warning:** Setting this to `False` will not only reveal any HTTPS Basic Auth that is configured, but the return data will also be written to the job cache. When possible, it is recommended to use SSH for authentication.

New in version 2015.5.6.

ignore_retcode [False] If `True`, do not log an error to the minion log if the git command returns a nonzero exit status.

New in version 2015.8.0.

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

> **Note:** This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

New in version 2018.3.1.

CLI Example:

```
salt myminion git.remotes /path/to/repo
```

`salt.modules.git.`**`reset`**(*cwd*, *opts=''*, *git_opts=''*, *user=None*, *password=None*, *identity=None*, *ignore_retcode=False*, *output_encoding=None*)

Interface to git-reset(1), returns the stdout from the git command

cwd The path to the git checkout

opts Any additional options to add to the command line, in a single string

Note: On the Salt CLI, if the opts are preceded with a dash, it is necessary to precede them with `opts=` (as in the CLI examples below) to avoid causing errors with Salt's own argument parsing.

git_opts Any additional options to add to git command itself (not the `reset` subcommand), in a single string. This is useful for passing `-c` to run git with temporary changes to the git configuration.

New in version 2017.7.0.

Note: This is only supported in git 1.7.2 and newer.

user User under which to run the git command. By default, the command is run by the user under which the minion is running.

password

Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

New in version 2016.3.4.

identity Path to a private key to use for ssh URLs. Salt will not attempt to use passphrase-protected keys unless invoked from the minion using `salt-call`, to prevent blocking waiting for user input. Key can also be specified as a SaltStack file server URL, eg. `salt://location/identity_file`.

Note: For greater security with passphraseless private keys, see the sshd(8) manpage for information on securing the keypair from the remote side in the `authorized_keys` file.

New in version 2018.3.5,2019.2.1,3000.

ignore_retcode [False] If `True`, do not log an error to the minion log if the git command returns a nonzero exit status.

New in version 2015.8.0.

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

Note: This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

New in version 2018.3.1.

CLI Examples:

```
# Soft reset to a specific commit ID
salt myminion git.reset /path/to/repo ac3ee5c
# Hard reset
salt myminion git.reset /path/to/repo opts='--hard origin/master'
```

`salt.modules.git.`**`rev_parse`**(*cwd, rev=None, opts='', git_opts='', user=None, password=None, ignore_retcode=False, output_encoding=None*)

New in version 2015.8.0.

Interface to git-rev-parse(1)

cwd The path to the git checkout

rev Revision to parse. See the SPECIFYING REVISIONS section of the git-rev-parse(1) manpage for details on how to format this argument.

This argument is optional when using the options in the *Options for Files* section of the git-rev-parse(1) manpage.

opts Any additional options to add to the command line, in a single string

git_opts Any additional options to add to git command itself (not the `rev-parse` subcommand), in a single string. This is useful for passing `-c` to run git with temporary changes to the git configuration.

New in version 2017.7.0.

Note: This is only supported in git 1.7.2 and newer.

user User under which to run the git command. By default, the command is run by the user under which the minion is running.

password

> Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

New in version 2016.3.4.

ignore_retcode [False] If `True`, do not log an error to the minion log if the git command returns a nonzero exit status.

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

Note: This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

New in version 2018.3.1.

CLI Examples:

```
# Get the full SHA1 for HEAD
salt myminion git.rev_parse /path/to/repo HEAD
# Get the short SHA1 for HEAD
salt myminion git.rev_parse /path/to/repo HEAD opts='--short'
# Get the develop branch's upstream tracking branch
salt myminion git.rev_parse /path/to/repo 'develop@{upstream}' opts='--abbrev-ref'
# Get the SHA1 for the commit corresponding to tag v1.2.3
salt myminion git.rev_parse /path/to/repo 'v1.2.3^{commit}'
# Find out whether or not the repo at /path/to/repo is a bare repository
salt myminion git.rev_parse /path/to/repo opts='--is-bare-repository'
```

`salt.modules.git.`**`revision`**(*cwd*, *rev='HEAD'*, *short=False*, *user=None*, *password=None*, *ignore_retcode=False*, *output_encoding=None*)

Returns the SHA1 hash of a given identifier (hash, branch, tag, HEAD, etc.)

cwd The path to the git checkout

rev [HEAD] The revision

short [False] If `True`, return an abbreviated SHA1 git hash

user User under which to run the git command. By default, the command is run by the user under which the minion is running.

password

Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

New in version 2016.3.4.

ignore_retcode [False] If `True`, do not log an error to the minion log if the git command returns a nonzero exit status.

New in version 2015.8.0.

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

Note: This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

New in version 2018.3.1.

CLI Example:

```
salt myminion git.revision /path/to/repo mybranch
```

`salt.modules.git.`**`rm_`**(*cwd*, *filename*, *opts=''*, *git_opts=''*, *user=None*, *password=None*, *ignore_retcode=False*, *output_encoding=None*)

Interface to git-rm(1)

cwd The path to the git checkout

filename The location of the file/directory to remove, relative to `cwd`

Note: To remove a directory, `-r` must be part of the `opts` parameter.

---

opts Any additional options to add to the command line, in a single string

---

Note: On the Salt CLI, if the opts are preceded with a dash, it is necessary to precede them with `opts=` (as in the CLI examples below) to avoid causing errors with Salt's own argument parsing.

---

git_opts Any additional options to add to git command itself (not the `rm` subcommand), in a single string. This is useful for passing `-c` to run git with temporary changes to the git configuration.

New in version 2017.7.0.

Note: This is only supported in git 1.7.2 and newer.

---

user User under which to run the git command. By default, the command is run by the user under which the minion is running.

password

Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

New in version 2016.3.4.

ignore_retcode [False] If `True`, do not log an error to the minion log if the git command returns a nonzero exit status.

New in version 2015.8.0.

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

Note: This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

New in version 2018.3.1.

CLI Examples:

```
salt myminion git.rm /path/to/repo foo/bar.py
salt myminion git.rm /path/to/repo foo/bar.py opts='--dry-run'
salt myminion git.rm /path/to/repo foo/baz opts='-r'
```

`salt.modules.git.`**`stash`**(*cwd*, *action='save'*, *opts=''*, *git_opts=''*, *user=None*, *password=None*, *ignore_retcode=False*, *output_encoding=None*)

Interface to git-stash(1), returns the stdout from the git command

cwd The path to the git checkout

opts Any additional options to add to the command line, in a single string. Use this to complete the `git stash` command by adding the remaining arguments (i.e. `'save <stash comment>'`, `'apply stash@{2}'`, `'show'`, etc.). Omitting this argument will simply run `git stash`.

git_opts Any additional options to add to git command itself (not the `stash` subcommand), in a single string. This is useful for passing `-c` to run git with temporary changes to the git configuration.

New in version 2017.7.0.

Note: This is only supported in git 1.7.2 and newer.

user User under which to run the git command. By default, the command is run by the user under which the minion is running.

password

Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

New in version 2016.3.4.

ignore_retcode [False] If `True`, do not log an error to the minion log if the git command returns a nonzero exit status.

New in version 2015.8.0.

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

Note: This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

New in version 2018.3.1.

CLI Examples:

```
salt myminion git.stash /path/to/repo save opts='work in progress'
salt myminion git.stash /path/to/repo apply opts='stash@{1}'
salt myminion git.stash /path/to/repo drop opts='stash@{1}'
salt myminion git.stash /path/to/repo list
```

`salt.modules.git.`**`status`**(*cwd*, *user=None*, *password=None*, *ignore_retcode=False*, *output_encoding=None*)

Changed in version 2015.8.0: Return data has changed from a list of lists to a dictionary

Returns the changes to the repository

cwd The path to the git checkout

user User under which to run the git command. By default, the command is run by the user under which the minion is running.

password

Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

New in version 2016.3.4.

ignore_retcode [False] If `True`, do not log an error to the minion log if the git command returns a nonzero exit status.

New in version 2015.8.0.

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

Note: This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

New in version 2018.3.1.

CLI Example:

```
salt myminion git.status /path/to/repo
```

`salt.modules.git.`**`submodule`**(*cwd*, *command*, *opts=''*, *git_opts=''*, *user=None*, *password=None*, *identity=None*, *ignore_retcode=False*, *saltenv='base'*, *output_encoding=None*, ***kwargs*)

Changed in version 2015.8.0: Added the `command` argument to allow for operations other than `update` to be run on submodules, and deprecated the `init` argument. To do a submodule update with `init=True` moving forward, use `command=update opts='--init'`

Interface to git-submodule(1)

cwd The path to the submodule

command Submodule command to run, see *git-submodule(1) <git submodule>* for more information. Any additional arguments after the command (such as the URL when adding a submodule) must be passed in the `opts` parameter.

New in version 2015.8.0.

opts Any additional options to add to the command line, in a single string

---

Note: On the Salt CLI, if the opts are preceded with a dash, it is necessary to precede them with `opts=` (as in the CLI examples below) to avoid causing errors with Salt's own argument parsing.

---

git_opts Any additional options to add to git command itself (not the `submodule` subcommand), in a single string. This is useful for passing `-c` to run git with temporary changes to the git configuration.

New in version 2017.7.0.

Note: This is only supported in git 1.7.2 and newer.

init [False] If `True`, ensures that new submodules are initialized

Deprecated since version 2015.8.0: Pass `init` as the `command` parameter, or include `--init` in the `opts` param with `command` set to update.

user User under which to run the git command. By default, the command is run by the user under which the minion is running.

password

> Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

New in version 2016.3.4.

identity Path to a private key to use for ssh URLs

> **Warning:** Unless Salt is invoked from the minion using `salt-call`, the key(s) must be passphraseless. For greater security with passphraseless private keys, see the sshd(8) manpage for information on securing the keypair from the remote side in the `authorized_keys` file.

Changed in version 2015.8.7: Salt will no longer attempt to use passphrase-protected keys unless invoked from the minion using `salt-call`, to prevent blocking waiting for user input.

Key can also be specified as a SaltStack file server URL, eg. salt://location/identity_file

Changed in version 2016.3.0.

ignore_retcode [False] If `True`, do not log an error to the minion log if the git command returns a nonzero exit status.

New in version 2015.8.0.

saltenv The default salt environment to pull sls files from

New in version 2016.3.1.

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

Note: This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

New in version 2018.3.1.

CLI Example:

```
# Update submodule and ensure it is initialized (before 2015.8.0)
salt myminion git.submodule /path/to/repo/sub/repo init=True
# Update submodule and ensure it is initialized (2015.8.0 and later)
salt myminion git.submodule /path/to/repo/sub/repo update opts='--init'

# Rebase submodule (2015.8.0 and later)
salt myminion git.submodule /path/to/repo/sub/repo update opts='--rebase'

# Add submodule (2015.8.0 and later)
salt myminion git.submodule /path/to/repo/sub/repo add opts='https://mydomain.tld/
↪repo.git'

# Unregister submodule (2015.8.0 and later)
salt myminion git.submodule /path/to/repo/sub/repo deinit
```

`salt.modules.git.`**`symbolic_ref`**(*cwd, ref, value=None, opts='', git_opts='', user=None, password=None, ignore_retcode=False, output_encoding=None*)

New in version 2015.8.0.

Interface to git-symbolic-ref(1)

cwd The path to the git checkout

ref Symbolic ref to read/modify

value If passed, then the symbolic ref will be set to this value and an empty string will be returned.

If not passed, then the ref to which `ref` points will be returned, unless `--delete` is included in `opts` (in which case the symbolic ref will be deleted).

opts Any additional options to add to the command line, in a single string

git_opts Any additional options to add to git command itself (not the `symbolic-refs` subcommand), in a single string. This is useful for passing `-c` to run git with temporary changes to the git configuration.

New in version 2017.7.0.

Note: This is only supported in git 1.7.2 and newer.

user User under which to run the git command. By default, the command is run by the user under which the minion is running.

password

Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

New in version 2016.3.4.

ignore_retcode [False] If `True`, do not log an error to the minion log if the git command returns a nonzero exit status.

New in version 2015.8.0.

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

> **Note:** This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

New in version 2018.3.1.

CLI Examples:

```
# Get ref to which HEAD is pointing
salt myminion git.symbolic_ref /path/to/repo HEAD
# Set/overwrite symbolic ref 'FOO' to local branch 'foo'
salt myminion git.symbolic_ref /path/to/repo FOO refs/heads/foo
# Delete symbolic ref 'FOO'
salt myminion git.symbolic_ref /path/to/repo FOO opts='--delete'
```

`salt.modules.git.`**`tag`**`(`*cwd, name, ref='HEAD', message=None, opts='', git_opts='', user=None, password=None, ignore_retcode=False, output_encoding=None*`)`

New in version 2018.3.4.

Interface to git-tag(1), adds and removes tags.

cwd The path to the main git checkout or a linked worktree

name Name of the tag

ref [HEAD] Which ref to tag (defaults to local clone's HEAD)

> **Note:** This argument is ignored when either `-d` or `--delete` is present in the `opts` passed to this function.

message Optional message to include with the tag. If provided, an annotated tag will be created.

opts Any additional options to add to the command line, in a single string

> **Note:** Additionally, on the Salt CLI, if the opts are preceded with a dash, it is necessary to precede them with `opts=` (as in the CLI examples below) to avoid causing errors with Salt's own argument parsing.

git_opts Any additional options to add to git command itself (not the `worktree` subcommand), in a single string. This is useful for passing `-c` to run git with temporary changes to the git configuration.

> **Note:** This is only supported in git 1.7.2 and newer.

user User under which to run the git command. By default, the command is run by the user under which the minion is running.

password Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

ignore_retcode [False] If `True`, do not log an error to the minion log if the git command returns a nonzero exit status.

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

> **Note:** This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

CLI Example:

```
# Create an non-annotated tag
salt myminion git.tag /path/to/repo v1.2
# Create an annotated tag
salt myminion git.tag /path/to/repo v1.2 message='Version 1.2'
# Delete the tag
salt myminion git.tag /path/to/repo v1.2 opts='-d'
```

`salt.modules.git.`**`version`**(*versioninfo=False*)

New in version 2015.8.0.

Returns the version of Git installed on the minion

versioninfo [False] If `True`, return the version in a versioninfo list (e.g. `[2, 5, 0]`)

CLI Example:

```
salt myminion git.version
```

`salt.modules.git.`**`worktree_add`**(*cwd, worktree_path, ref=None, reset_branch=None, force=None, detach=False, opts='', git_opts='', user=None, password=None, ignore_retcode=False, output_encoding=None, **kwargs*)

New in version 2015.8.0.

Interface to git-worktree(1), adds a worktree

cwd The path to the git checkout

worktree_path Path to the new worktree. Can be either absolute, or relative to `cwd`.

branch Name of new branch to create. If omitted, will be set to the basename of the `worktree_path`. For example, if the `worktree_path` is `/foo/bar/baz`, then `branch` will be `baz`.

ref Name of the ref on which to base the new worktree. If omitted, then `HEAD` is use, and a new branch will be created, named for the basename of the `worktree_path`. For example, if the `worktree_path` is `/foo/bar/baz` then a new branch `baz` will be created, and pointed at `HEAD`.

reset_branch [False] If `False`, then git-worktree(1) will fail to create the worktree if the targeted branch already exists. Set this argument to `True` to reset the targeted branch to point at `ref`, and checkout the newly-reset branch into the new worktree.

force [False] By default, git-worktree(1) will not permit the same branch to be checked out in more than one worktree. Set this argument to `True` to override this.

opts Any additional options to add to the command line, in a single string

> **Note:** On the Salt CLI, if the opts are preceded with a dash, it is necessary to precede them with `opts=` to avoid causing errors with Salt's own argument parsing.
>
> All CLI options for adding worktrees as of Git 2.5.0 are already supported by this function as of Salt 2015.8.0, so using this argument is unnecessary unless new CLI arguments are added to git-worktree(1) and are not yet supported in Salt.

git_opts Any additional options to add to git command itself (not the `worktree` subcommand), in a single string. This is useful for passing `-c` to run git with temporary changes to the git configuration.

New in version 2017.7.0.

Note: This is only supported in git 1.7.2 and newer.

---

user User under which to run the git command. By default, the command is run by the user under which the minion is running.

password

Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

New in version 2016.3.4.

ignore_retcode [False] If `True`, do not log an error to the minion log if the git command returns a nonzero exit status.

New in version 2015.8.0.

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

Note: This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

New in version 2018.3.1.

CLI Examples:

```
salt myminion git.worktree_add /path/to/repo/main ../hotfix ref=origin/master
salt myminion git.worktree_add /path/to/repo/main ../hotfix branch=hotfix21␣
↪ref=v2.1.9.3
```

`salt.modules.git.`**`worktree_prune`**`(`*cwd, dry_run=False, verbose=True, expire=None, opts='', git_opts='', user=None, password=None, ignore_retcode=False, output_encoding=None*`)`

New in version 2015.8.0.

Interface to git-worktree(1), prunes stale worktree administrative data from the gitdir

cwd The path to the main git checkout or a linked worktree

dry_run [False] If `True`, then this function will report what would have been pruned, but no changes will be made.

verbose [True] Report all changes made. Set to `False` to suppress this output.

expire Only prune unused worktree data older than a specific period of time. The date format for this parameter is described in the documentation for the `gc.pruneWorktreesExpire` config param in the git-config(1) manpage.

opts Any additional options to add to the command line, in a single string

---

Note: On the Salt CLI, if the opts are preceded with a dash, it is necessary to precede them with `opts=` to avoid causing errors with Salt's own argument parsing.

All CLI options for pruning worktrees as of Git 2.5.0 are already supported by this function as of Salt 2015.8.0, so using this argument is unnecessary unless new CLI arguments are added to git-worktree(1) and are not yet supported in Salt.

git_opts Any additional options to add to git command itself (not the `worktree` subcommand), in a single string. This is useful for passing `-c` to run git with temporary changes to the git configuration.

New in version 2017.7.0.

Note: This is only supported in git 1.7.2 and newer.

user User under which to run the git command. By default, the command is run by the user under which the minion is running.

password

Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

New in version 2016.3.4.

ignore_retcode [False] If `True`, do not log an error to the minion log if the git command returns a nonzero exit status.

New in version 2015.8.0.

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

Note: This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

New in version 2018.3.1.

CLI Examples:

```
salt myminion git.worktree_prune /path/to/repo
salt myminion git.worktree_prune /path/to/repo dry_run=True
salt myminion git.worktree_prune /path/to/repo expire=1.day.ago
```

`salt.modules.git.`**`worktree_rm`**(*cwd*, *user=None*, *output_encoding=None*)

New in version 2015.8.0.

Recursively removes the worktree located at `cwd`, returning `True` if successful. This function will attempt to determine if `cwd` is actually a worktree by invoking *`git.is_worktree`*. If the path does not correspond to a worktree, then an error will be raised and no action will be taken.

Warning: There is no undoing this action. Be **VERY** careful before running this function.

cwd Path to the worktree to be removed

user Used for path expansion when `cwd` is not an absolute path. By default, when `cwd` is not absolute, the path will be assumed to be relative to the home directory of the user under which the minion is running. Setting this option will change the home directory from which path expansion is performed.

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

Note: This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

New in version 2018.3.1.

CLI Examples:

```
salt myminion git.worktree_rm /path/to/worktree
```

21.9.141 salt.modules.github module

Module for interacting with the GitHub v3 API.

New in version 2016.3.0.

depends PyGithub python module

Configuration

Configure this module by specifying the name of a configuration profile in the minion config, minion pillar, or master config. The module will use the 'github' key by default, if defined.

For example:

```
github:
  token: abc1234
  org_name: my_organization

  # optional: some functions require a repo_name, which
  # can be set in the config file, or passed in at the CLI.
  repo_name: my_repo

  # optional: it can be dangerous to change the privacy of a repository
  # in an automated way. set this to True to allow privacy modifications
  allow_repo_privacy_changes: False
```

`salt.modules.github.`**`add_repo`**(*name, description=None, homepage=None, private=None, has_issues=None, has_wiki=None, has_downloads=None, auto_init=None, gitignore_template=None, license_template=None, profile='github'*)

Create a new github repository.

name The name of the team to be created.

description The description of the repository.

homepage The URL with more information about the repository.

private The visiblity of the repository. Note that private repositories require a paid GitHub account.

has_issues Whether to enable issues for this repository.

has_wiki Whether to enable the wiki for this repository.

has_downloads Whether to enable downloads for this repository.

auto_init Whether to create an initial commit with an empty README.

gitignore_template The desired language or platform for a .gitignore, e.g "Haskell".

license_template The desired LICENSE template to apply, e.g "mit" or "mozilla".

profile The name of the profile configuration to use. Defaults to `github`.

CLI Example:

```
salt myminion github.add_repo 'repo_name'
```

New in version 2016.11.0.

`salt.modules.github.`**`add_team`**(*name*, *description=None*, *repo_names=None*, *privacy=None*, *permission=None*, *profile='github'*)

Create a new Github team within an organization.

name The name of the team to be created.

description The description of the team.

repo_names The names of repositories to add the team to.

privacy The level of privacy for the team, can be 'secret' or 'closed'.

permission The default permission for new repositories added to the team, can be 'pull', 'push' or 'admin'.

profile The name of the profile configuration to use. Defaults to `github`.

CLI Example:

```
salt myminion github.add_team 'team_name'
```

New in version 2016.11.0.

`salt.modules.github.`**`add_team_member`**(*name*, *team_name*, *profile='github'*)

Adds a team member to a team with team_name.

name The name of the team member to add.

team_name The name of the team of which to add the user.

profile The name of the profile configuration to use. Defaults to `github`.

CLI Example:

```
salt myminion github.add_team_member 'user_name' 'team_name'
```

New in version 2016.11.0.

`salt.modules.github.`**`add_team_repo`**(*repo_name*, *team_name*, *profile='github'*, *permission=None*)

Adds a repository to a team with team_name.

repo_name The name of the repository to add.

team_name The name of the team of which to add the repository.

profile The name of the profile configuration to use. Defaults to `github`.

permission The permission for team members within the repository, can be 'pull', 'push' or 'admin'. If not specified, the default permission specified on the team will be used.

New in version 2017.7.0.

CLI Example:

```
salt myminion github.add_team_repo 'my_repo' 'team_name'
```

New in version 2016.11.0.

`salt.modules.github.`**`add_user`**`(`*name*, *profile='github'*`)`

Add a GitHub user.

name The user for which to obtain information.

profile The name of the profile configuration to use. Defaults to `github`.

CLI Example:

```
salt myminion github.add_user github-handle
```

`salt.modules.github.`**`edit_repo`**`(`*name*, *description=None*, *homepage=None*, *private=None*, *has_issues=None*, *has_wiki=None*, *has_downloads=None*, *profile='github'*`)`

Updates an existing Github repository.

name The name of the team to be created.

description The description of the repository.

homepage The URL with more information about the repository.

private The visiblity of the repository. Note that private repositories require a paid GitHub account.

has_issues Whether to enable issues for this repository.

has_wiki Whether to enable the wiki for this repository.

has_downloads Whether to enable downloads for this repository.

profile The name of the profile configuration to use. Defaults to `github`.

CLI Example:

```
salt myminion github.add_repo 'repo_name'
```

New in version 2016.11.0.

`salt.modules.github.`**`edit_team`**`(`*name*, *description=None*, *privacy=None*, *permission=None*, *profile='github'*`)`

Updates an existing Github team.

name The name of the team to be edited.

description The description of the team.

privacy The level of privacy for the team, can be 'secret' or 'closed'.

permission The default permission for new repositories added to the team, can be 'pull', 'push' or 'admin'.

profile The name of the profile configuration to use. Defaults to `github`.

CLI Example:

```
salt myminion github.edit_team 'team_name' description='Team description'
```

New in version 2016.11.0.

`salt.modules.github.`**`get_issue`**`(`*issue_number*, *repo_name=None*, *profile='github'*, *output='min'*`)`

Return information about a single issue in a named repository.

New in version 2016.11.0.

issue_number The number of the issue to retrieve.

repo_name The name of the repository from which to get the issue. This argument is required, either passed via the CLI, or defined in the configured profile. A `repo_name` passed as a CLI argument will override the repo_name defined in the configured profile, if provided.

profile The name of the profile configuration to use. Defaults to `github`.

output The amount of data returned by each issue. Defaults to `min`. Change to `full` to see all issue output.

CLI Example:

```
salt myminion github.get_issue 514
salt myminion github.get_issue 514 repo_name=salt
```

`salt.modules.github.`**`get_issue_comments`**`(`*issue_number*, *repo_name=None*, *profile='github'*, *since=None*, *output='min'*`)`

Return information about the comments for a given issue in a named repository.

New in version 2016.11.0.

issue_number The number of the issue for which to retrieve comments.

repo_name The name of the repository to which the issue belongs. This argument is required, either passed via the CLI, or defined in the configured profile. A `repo_name` passed as a CLI argument will override the repo_name defined in the configured profile, if provided.

profile The name of the profile configuration to use. Defaults to `github`.

since Only comments updated at or after this time are returned. This is a timestamp in ISO 8601 format: `YYYY-MM-DDTHH:MM:SSZ`.

output The amount of data returned by each issue. Defaults to `min`. Change to `full` to see all issue output.

CLI Example:

```
salt myminion github.get_issue_comments 514
salt myminion github.get_issue 514 repo_name=salt
```

`salt.modules.github.`**`get_issues`**`(`*repo_name=None*, *profile='github'*, *milestone=None*, *state='open'*, *assignee=None*, *creator=None*, *mentioned=None*, *labels=None*, *sort='created'*, *direction='desc'*, *since=None*, *output='min'*, *per_page=None*`)`

Returns information for all issues in a given repository, based on the search options.

New in version 2016.11.0.

repo_name The name of the repository for which to list issues. This argument is required, either passed via the CLI, or defined in the configured profile. A `repo_name` passed as a CLI argument will override the repo_name defined in the configured profile, if provided.

profile The name of the profile configuration to use. Defaults to `github`.

milestone The number of a GitHub milestone, or a string of either `*` or `none`.

If a number is passed, it should refer to a milestone by its number field. Use the `github.get_milestone` function to obtain a milestone's number.

If the string `*` is passed, issues with any milestone are accepted. If the string `none` is passed, issues without milestones are returned.

state Indicates the state of the issues to return. Can be either `open`, `closed`, or `all`. Default is `open`.

assignee Can be the name of a user. Pass in `none` (as a string) for issues with no assigned user or `*` for issues assigned to any user.

creator The user that created the issue.

mentioned A user that's mentioned in the issue.

labels A string of comma separated label names. For example, `bug,ui,@high`.

sort What to sort results by. Can be either `created`, `updated`, or `comments`. Default is `created`.

direction The direction of the sort. Can be either `asc` or `desc`. Default is `desc`.

since Only issues updated at or after this time are returned. This is a timestamp in ISO 8601 format: `YYYY-MM-DDTHH:MM:SSZ`.

output The amount of data returned by each issue. Defaults to `min`. Change to `full` to see all issue output.

per_page GitHub paginates data in their API calls. Use this value to increase or decrease the number of issues gathered from GitHub, per page. If not set, GitHub defaults are used. Maximum is 100.

CLI Example:

```
salt myminion github.get_issues my-github-repo
```

`salt.modules.github.`**`get_milestone`**(*number=None, name=None, repo_name=None, profile='github', output='min'*)

Return information about a single milestone in a named repository.

New in version 2016.11.0.

number The number of the milestone to retrieve. If provided, this option will be favored over `name`.

name The name of the milestone to retrieve.

repo_name The name of the repository for which to list issues. This argument is required, either passed via the CLI, or defined in the configured profile. A `repo_name` passed as a CLI argument will override the repo_name defined in the configured profile, if provided.

profile The name of the profile configuration to use. Defaults to `github`.

output The amount of data returned by each issue. Defaults to `min`. Change to `full` to see all issue output.

CLI Example:

```
salt myminion github.get_milestone 72
salt myminion github.get_milestone name=my_milestone
```

`salt.modules.github.`**`get_milestones`**(*repo_name=None, profile='github', state='open', sort='due_on', direction='asc', output='min', per_page=None*)

Return information about milestones for a given repository.

New in version 2016.11.0.

repo_name The name of the repository for which to list issues. This argument is required, either passed via the CLI, or defined in the configured profile. A `repo_name` passed as a CLI argument will override the repo_name defined in the configured profile, if provided.

profile The name of the profile configuration to use. Defaults to `github`.

state The state of the milestone. Either `open`, `closed`, or `all`. Default is `open`.

sort What to sort results by. Either `due_on` or `completeness`. Default is `due_on`.

direction The direction of the sort. Either `asc` or `desc`. Default is `asc`.

output The amount of data returned by each issue. Defaults to `min`. Change to `full` to see all issue output.

per_page GitHub paginates data in their API calls. Use this value to increase or decrease the number of issues gathered from GitHub, per page. If not set, GitHub defaults are used.

CLI Example:

```
salt myminion github.get_milestones
```

`salt.modules.github.`**`get_prs`**(*repo_name=None, profile='github', state='open', head=None, base=None, sort='created', direction='desc', output='min', per_page=None*)

Returns information for all pull requests in a given repository, based on the search options provided.

New in version 2017.7.0.

repo_name The name of the repository for which to list pull requests. This argument is required, either passed via the CLI, or defined in the configured profile. A `repo_name` passed as a CLI argument will override the `repo_name` defined in the configured profile, if provided.

profile The name of the profile configuration to use. Defaults to `github`.

state Indicates the state of the pull requests to return. Can be either `open`, `closed`, or `all`. Default is `open`.

head Filter pull requests by head user and branch name in the format of `user:ref-name`. Example: `'github:new-script-format'`. Default is `None`.

base Filter pulls by base branch name. Example: `gh-pages`. Default is `None`.

sort What to sort results by. Can be either `created`, `updated`, `popularity` (comment count), or `long-running` (age, filtering by pull requests updated within the last month). Default is `created`.

direction The direction of the sort. Can be either `asc` or `desc`. Default is `desc`.

output The amount of data returned by each pull request. Defaults to `min`. Change to `full` to see all pull request output.

per_page GitHub paginates data in their API calls. Use this value to increase or decrease the number of pull requests gathered from GitHub, per page. If not set, GitHub defaults are used. Maximum is 100.

CLI Example:

```
salt myminion github.get_prs
salt myminion github.get_prs base=2016.11
```

`salt.modules.github.`**`get_repo_info`**(*repo_name, profile='github', ignore_cache=False*)

Return information for a given repo.

New in version 2016.11.0.

repo_name The name of the repository.

profile The name of the profile configuration to use. Defaults to `github`.

CLI Example:

```
salt myminion github.get_repo_info salt
salt myminion github.get_repo_info salt profile='my-github-profile'
```

`salt.modules.github.`**`get_repo_teams`**(*repo_name, profile='github'*)

Return teams belonging to a repository.

New in version 2017.7.0.

repo_name The name of the repository from which to retrieve teams.

profile The name of the profile configuration to use. Defaults to `github`.

CLI Example:

```
salt myminion github.get_repo_teams salt
salt myminion github.get_repo_teams salt profile='my-github-profile'
```

`salt.modules.github.`**`get_team`**(*name*, *profile='github'*)

Returns the team details if a team with the given name exists, or None otherwise.

name The team name for which to obtain information.

profile The name of the profile configuration to use. Defaults to `github`.

CLI Example:

```
salt myminion github.get_team 'team_name'
```

`salt.modules.github.`**`get_user`**(*name*, *profile='github'*, *user_details=False*)

Get a GitHub user by name.

name The user for which to obtain information.

profile The name of the profile configuration to use. Defaults to `github`.

user_details Prints user information details. Defaults to `False`. If the user is already in the organization and user_details is set to False, the get_user function returns `True`. If the user is not already present in the organization, user details will be printed by default.

CLI Example:

```
salt myminion github.get_user github-handle
salt myminion github.get_user github-handle user_details=true
```

`salt.modules.github.`**`is_team_member`**(*name*, *team_name*, *profile='github'*)

Returns True if the github user is in the team with team_name, or False otherwise.

name The name of the user whose membership to check.

team_name The name of the team to check membership in.

profile The name of the profile configuration to use. Defaults to `github`.

CLI Example:

```
salt myminion github.is_team_member 'user_name' 'team_name'
```

New in version 2016.11.0.

`salt.modules.github.`**`list_members_without_mfa`**(*profile='github'*, *ignore_cache=False*)

List all members (in lower case) without MFA turned on.

profile The name of the profile configuration to use. Defaults to `github`.

ignore_cache Bypasses the use of cached team repos.

CLI Example:

```
salt myminion github.list_members_without_mfa
```

New in version 2016.11.0.

`salt.modules.github.`**`list_private_repos`**(*profile='github'*)

List private repositories within the organization. Dependent upon the access rights of the profile token.

New in version 2016.11.0.

profile The name of the profile configuration to use. Defaults to `github`.

CLI Example:

```
salt myminion github.list_private_repos
salt myminion github.list_private_repos profile='my-github-profile'
```

`salt.modules.github.`**`list_public_repos`**(*profile='github'*)

List public repositories within the organization.

New in version 2016.11.0.

profile The name of the profile configuration to use. Defaults to `github`.

CLI Example:

```
salt myminion github.list_public_repos
salt myminion github.list_public_repos profile='my-github-profile'
```

`salt.modules.github.`**`list_repos`**(*profile='github'*)

List all repositories within the organization. Includes public and private repositories within the organization Dependent upon the access rights of the profile token.

New in version 2016.11.0.

profile The name of the profile configuration to use. Defaults to `github`.

CLI Example:

```
salt myminion github.list_repos
salt myminion github.list_repos profile='my-github-profile'
```

`salt.modules.github.`**`list_team_members`**(*team_name*, *profile='github'*, *ignore_cache=False*)

Gets the names of team members in lower case.

team_name The name of the team from which to list members.

profile The name of the profile configuration to use. Defaults to `github`.

ignore_cache Bypasses the use of cached team members.

CLI Example:

```
salt myminion github.list_team_members 'team_name'
```

New in version 2016.11.0.

`salt.modules.github.`**`list_team_repos`**(*team_name*, *profile='github'*, *ignore_cache=False*)

Gets the repo details for a given team as a dict from repo_name to repo details. Note that repo names are always in lower case.

team_name The name of the team from which to list repos.

profile The name of the profile configuration to use. Defaults to `github`.

ignore_cache Bypasses the use of cached team repos.

CLI Example:

```
salt myminion github.list_team_repos 'team_name'
```

New in version 2016.11.0.

`salt.modules.github.`**`list_teams`**(*profile='github', ignore_cache=False*)

Lists all teams with the organization.

profile The name of the profile configuration to use. Defaults to `github`.

ignore_cache Bypasses the use of cached teams.

CLI Example:

```
salt myminion github.list_teams
```

New in version 2016.11.0.

`salt.modules.github.`**`list_users`**(*profile='github', ignore_cache=False*)

List all users within the organization.

profile The name of the profile configuration to use. Defaults to `github`.

ignore_cache Bypasses the use of cached users.

New in version 2016.11.0.

CLI Example:

```
salt myminion github.list_users
salt myminion github.list_users profile='my-github-profile'
```

`salt.modules.github.`**`remove_repo`**(*name, profile='github'*)

Remove a Github repository.

name The name of the repository to be removed.

profile The name of the profile configuration to use. Defaults to `github`.

CLI Example:

```
salt myminion github.remove_repo 'my-repo'
```

New in version 2016.11.0.

`salt.modules.github.`**`remove_team`**(*name, profile='github'*)

Remove a github team.

name The name of the team to be removed.

profile The name of the profile configuration to use. Defaults to `github`.

CLI Example:

```
salt myminion github.remove_team 'team_name'
```

New in version 2016.11.0.

`salt.modules.github.`**`remove_team_member`**(*name, team_name, profile='github'*)

Removes a team member from a team with team_name.

name The name of the team member to remove.

team_name The name of the team from which to remove the user.

profile The name of the profile configuration to use. Defaults to `github`.

CLI Example:

```
salt myminion github.remove_team_member 'user_name' 'team_name'
```

New in version 2016.11.0.

salt.modules.github.**remove_team_repo**(*repo_name*, *team_name*, *profile='github'*)

Removes a repository from a team with team_name.

repo_name The name of the repository to remove.

team_name The name of the team of which to remove the repository.

profile The name of the profile configuration to use. Defaults to `github`.

CLI Example:

```
salt myminion github.remove_team_repo 'my_repo' 'team_name'
```

New in version 2016.11.0.

salt.modules.github.**remove_user**(*name*, *profile='github'*)

Remove a Github user by name.

name The user for which to obtain information.

profile The name of the profile configuration to use. Defaults to `github`.

CLI Example:

```
salt myminion github.remove_user github-handle
```

21.9.142 salt.modules.glance

Module for handling openstack glance calls.

optdepends

- glanceclient Python adapter

configuration This module is not usable until the following are specified either in a pillar or in the minion's config file:

```
keystone.user: admin
keystone.password: verybadpass
keystone.tenant: admin
keystone.insecure: False   #(optional)
keystone.auth_url: 'http://127.0.0.1:5000/v2.0/'
```

If configuration for multiple openstack accounts is required, they can be set up as different configuration profiles: For example:

```
openstack1:
  keystone.user: admin
  keystone.password: verybadpass
  keystone.tenant: admin
  keystone.auth_url: 'http://127.0.0.1:5000/v2.0/'

openstack2:
  keystone.user: admin
  keystone.password: verybadpass
```

(continues on next page)

(continued from previous page)

```
  keystone.tenant: admin
  keystone.auth_url: 'http://127.0.0.2:5000/v2.0/'
```

With this configuration in place, any of the glance functions can make use of a configuration profile by declaring it explicitly. For example:

```
salt '*' glance.image_list profile=openstack1
```

`salt.modules.glance.`**`image_create`**(*name, location=None, profile=None, visibility=None, container_format='bare', disk_format='raw', protected=None*)

Create an image (glance image-create)

CLI Example, old format:

```
salt '*' glance.image_create name=f16-jeos \
         disk_format=qcow2 container_format=ovf
```

CLI Example, new format resembling Glance API v2:

```
salt '*' glance.image_create name=f16-jeos visibility=public \
         disk_format=qcow2 container_format=ovf
```

The parameter 'visibility' defaults to 'public' if not specified.

`salt.modules.glance.`**`image_delete`**(*id=None, name=None, profile=None*)

Delete an image (glance image-delete)

CLI Examples:

```
salt '*' glance.image_delete c2eb2eb0-53e1-4a80-b990-8ec887eae7df
salt '*' glance.image_delete id=c2eb2eb0-53e1-4a80-b990-8ec887eae7df
salt '*' glance.image_delete name=f16-jeos
```

`salt.modules.glance.`**`image_list`**(*id=None, profile=None, name=None*)

Return a list of available images (glance image-list)

CLI Example:

```
salt '*' glance.image_list
```

`salt.modules.glance.`**`image_schema`**(*profile=None*)

Returns names and descriptions of the schema "image"'s properties for this profile's instance of glance

CLI Example:

```
salt '*' glance.image_schema
```

`salt.modules.glance.`**`image_show`**(*id=None, name=None, profile=None*)

Return details about a specific image (glance image-show)

CLI Example:

```
salt '*' glance.image_show
```

`salt.modules.glance.`**`image_update`**(*id=None, name=None, profile=None, **kwargs*)

Update properties of given image. Known to work for: - min_ram (in MB) - protected (bool) - visibility ('public' or 'private')

CLI Example:

```
salt '*' glance.image_update id=c2eb2eb0-53e1-4a80-b990-8ec887eae7df
salt '*' glance.image_update name=f16-jeos
```

salt.modules.glance.**schema_get**(*name*, *profile=None*)

Known valid names of schemas are:

- image
- images
- member
- members

CLI Example:

```
salt '*' glance.schema_get name=f16-jeos
```

21.9.143 salt.modules.glanceng

Glance module for interacting with OpenStack Glance

New in version 2018.3.0.

:depends:shade

Example configuration

```
glance:
  cloud: default
```

```
glance:
  auth:
    username: admin
    password: password123
    user_domain_name: mydomain
    project_name: myproject
    project_domain_name: myproject
    auth_url: https://example.org:5000/v3
  identity_api_version: 3
```

salt.modules.glanceng.**compare_changes**(*obj*, ***kwargs*)
Compare two dicts returning only keys that exist in the first dict and are different in the second one

salt.modules.glanceng.**get_openstack_cloud**(*auth=None*)
Return an openstack_cloud

salt.modules.glanceng.**get_operator_cloud**(*auth=None*)
Return an operator_cloud

salt.modules.glanceng.**image_create**(*auth=None*, ***kwargs*)
Create an image

CLI Example:

```
salt '*' glanceng.image_create name=cirros file=cirros.raw disk_format=raw
salt '*' glanceng.image_create name=cirros file=cirros.raw disk_format=raw hw_
→scsi_model=virtio-scsi hw_disk_bus=scsi
```

`salt.modules.glanceng.`**`image_delete`**`(auth=None, **kwargs)`

Delete an image

CLI Example:

```
salt '*' glanceng.image_delete name=image1
salt '*' glanceng.image_delete name=0e4febc2a5ab4f2c8f374b054162506d
```

`salt.modules.glanceng.`**`image_get`**`(auth=None, **kwargs)`

Get a single image

CLI Example:

```
salt '*' glanceng.image_get name=image1
salt '*' glanceng.image_get name=0e4febc2a5ab4f2c8f374b054162506d
```

`salt.modules.glanceng.`**`image_list`**`(auth=None, **kwargs)`

List images

CLI Example:

```
salt '*' glanceng.image_list
salt '*' glanceng.image_list
```

`salt.modules.glanceng.`**`image_search`**`(auth=None, **kwargs)`

Search for images

CLI Example:

```
salt '*' glanceng.image_search name=image1
salt '*' glanceng.image_search
```

`salt.modules.glanceng.`**`setup_clouds`**`(auth=None)`

Call functions to create Shade cloud objects in __context__ to take advantage of Shade's in-memory caching across several states

`salt.modules.glanceng.`**`update_image_properties`**`(auth=None, **kwargs)`

Update properties for an image

CLI Example:

```
salt '*' glanceng.update_image_properties name=image1 hw_scsi_model=virtio-scsi⊠
↪hw_disk_bus=scsi
salt '*' glanceng.update_image_properties name=0e4febc2a5ab4f2c8f374b054162506d⊠
↪min_ram=1024
```

21.9.144 salt.modules.glusterfs

Manage a glusterfs pool

`salt.modules.glusterfs.`**`add_volume_bricks`**`(name, bricks)`

Add brick(s) to an existing volume

name Volume name

bricks List of bricks to add to the volume

CLI Example:

```
salt '*' glusterfs.add_volume_bricks <volume> <bricks>
```

salt.modules.glusterfs.**create_volume**(*name, bricks, stripe=False, replica=False, device_vg=False, transport='tcp', start=False, force=False, arbiter=False*)

Create a glusterfs volume

name Name of the gluster volume

bricks Bricks to create volume from, in <peer>:<brick path> format. For multiple bricks use list format: '["<peer1>:<brick1>", "<peer2>:<brick2>"]'

stripe Stripe count, the number of bricks should be a multiple of the stripe count for a distributed striped volume

replica Replica count, the number of bricks should be a multiple of the replica count for a distributed replicated volume

arbiter If true, specifies volume should use arbiter brick(s). Valid configuration limited to "replica 3 arbiter 1" per Gluster documentation. Every third brick in the brick list is used as an arbiter brick.

New in version 2019.2.0.

device_vg If true, specifies volume should use block backend instead of regular posix backend. Block device backend volume does not support multiple bricks

transport Transport protocol to use, can be 'tcp', 'rdma' or 'tcp,rdma'

start Start the volume after creation

force Force volume creation, this works even if creating in root FS

CLI Examples:

```
salt host1 glusterfs.create newvolume host1:/brick

salt gluster1 glusterfs.create vol2 '["gluster1:/export/vol2/brick",
↪"gluster2:/export/vol2/brick"]' replica=2 start=True
```

salt.modules.glusterfs.**delete_volume**(*target, stop=True*)

Deletes a gluster volume

target Volume to delete

stop [True] If `True`, stop volume before delete

CLI Example:

```
salt '*' glusterfs.delete_volume <volume>
```

salt.modules.glusterfs.**disable_quota_volume**(*name*)

Disable quota on a glusterfs volume.

name Name of the gluster volume

CLI Example:

```
salt '*' glusterfs.disable_quota_volume <volume>
```

salt.modules.glusterfs.**enable_quota_volume**(*name*)

Enable quota on a glusterfs volume.

name Name of the gluster volume

CLI Example:

```
salt '*' glusterfs.enable_quota_volume <volume>
```

`salt.modules.glusterfs.`**`get_max_op_version`**`()`

New in version 2019.2.0.

Returns the glusterfs volume's max op-version value Requires Glusterfs version > 3.9

CLI Example: .. code-block:: bash

salt '*' glusterfs.get_max_op_version

`salt.modules.glusterfs.`**`get_op_version`**`(`*name*`)`

New in version 2019.2.0.

Returns the glusterfs volume op-version

name Name of the glusterfs volume

CLI Example:

```
salt '*' glusterfs.get_op_version <volume>
```

`salt.modules.glusterfs.`**`get_version`**`()`

New in version 2019.2.0.

Returns the version of glusterfs. CLI Example: .. code-block:: bash

salt '*' glusterfs.get_version

`salt.modules.glusterfs.`**`info`**`(`*name=None*`)`

New in version 2015.8.4.

Return gluster volume info.

name Optional name to retrieve only information of one volume

CLI Example:

```
salt '*' glusterfs.info
```

`salt.modules.glusterfs.`**`list_quota_volume`**`(`*name*`)`

List quotas of glusterfs volume

name Name of the gluster volume

CLI Example:

```
salt '*' glusterfs.list_quota_volume <volume>
```

`salt.modules.glusterfs.`**`list_volumes`**`()`

List configured volumes

CLI Example:

```
salt '*' glusterfs.list_volumes
```

`salt.modules.glusterfs.`**`peer`**`(`*name*`)`

Add another node into the peer list.

name The remote host to probe.

CLI Example:

```
salt 'one.gluster.*' glusterfs.peer two
```

GLUSTER direct CLI example (to show what salt is sending to gluster):

$ gluster peer probe ftp2

GLUSTER CLI 3.4.4 return example (so we know what we are parsing): #if the "peer" is the local host: peer probe: success: on localhost not needed

#if the peer was just added: peer probe: success

#if the peer was already part of the cluster: peer probe: success: host ftp2 port 24007 already in peer list

`salt.modules.glusterfs.`**`peer_status`**`()`

Return peer status information

The return value is a dictionary with peer UUIDs as keys and dicts of peer information as values. Hostnames are listed in one list. GlusterFS separates one of the hostnames but the only reason for this seems to be which hostname happens to be used first in peering.

CLI Example:

```
salt '*' glusterfs.peer_status
```

GLUSTER direct CLI example (to show what salt is sending to gluster):

$ gluster peer status

GLUSTER CLI 3.4.4 return example (so we know what we are parsing):

Number of Peers: 2

Hostname: ftp2 Port: 24007 Uuid: cbcb256b-e66e-4ec7-a718-21082d396c24 State: Peer in Cluster (Connected)

Hostname: ftp3 Uuid: 5ea10457-6cb2-427b-a770-7897509625e9 State: Peer in Cluster (Connected)

`salt.modules.glusterfs.`**`set_op_version`**`(`*version*`)`

New in version 2019.2.0.

Set the glusterfs volume op-version

version Version to set the glusterfs volume op-version

CLI Example:

```
salt '*' glusterfs.set_op_version <volume>
```

`salt.modules.glusterfs.`**`set_quota_volume`**`(`*name, path, size, enable_quota=False*`)`

Set quota to glusterfs volume.

name Name of the gluster volume

path Folder path for restriction in volume ("/")

size Hard-limit size of the volume (MB/GB)

enable_quota Enable quota before set up restriction

CLI Example:

```
salt '*' glusterfs.set_quota_volume <volume> <path> <size> enable_quota=True
```

`salt.modules.glusterfs.`**`start_volume`**(*name*, *force=False*)

Start a gluster volume

name Volume name

force Force the volume start even if the volume is started .. versionadded:: 2015.8.4

CLI Example:

```
salt '*' glusterfs.start mycluster
```

`salt.modules.glusterfs.`**`status`**(*name*)

Check the status of a gluster volume.

name Volume name

CLI Example:

```
salt '*' glusterfs.status myvolume
```

`salt.modules.glusterfs.`**`stop_volume`**(*name*, *force=False*)

Stop a gluster volume

name Volume name

force Force stop the volume

New in version 2015.8.4.

CLI Example:

```
salt '*' glusterfs.stop_volume mycluster
```

`salt.modules.glusterfs.`**`unset_quota_volume`**(*name*, *path*)

Unset quota on glusterfs volume

name Name of the gluster volume

path Folder path for restriction in volume

CLI Example:

```
salt '*' glusterfs.unset_quota_volume <volume> <path>
```

21.9.145 salt.modules.gnomedesktop

GNOME implementations

`salt.modules.gnomedesktop.`**`get`**(*schema=None*, *key=None*, *user=None*, ***kwargs*)

Get key in a particular GNOME schema

CLI Example:

```
salt '*' gnome.get user=<username> schema=org.gnome.desktop.screensaver key=idle-
↪activation-enabled
```

`salt.modules.gnomedesktop.`**`getClockFormat`**(***kwargs*)

Return the current clock format, either 12h or 24h format.

CLI Example:

```
salt '*' gnome.getClockFormat user=<username>
```

`salt.modules.gnomedesktop.`**`getClockShowDate`**`(`*`**kwargs`*`)`

Return the current setting, if the date is shown in the clock

CLI Example:

```
salt '*' gnome.getClockShowDate user=<username>
```

`salt.modules.gnomedesktop.`**`getIdleActivation`**`(`*`**kwargs`*`)`

Get whether the idle activation is enabled

CLI Example:

```
salt '*' gnome.getIdleActivation user=<username>
```

`salt.modules.gnomedesktop.`**`getIdleDelay`**`(`*`**kwargs`*`)`

Return the current idle delay setting in seconds

CLI Example:

```
salt '*' gnome.getIdleDelay user=<username>
```

`salt.modules.gnomedesktop.`**`ping`**`(`*`**kwargs`*`)`

A test to ensure the GNOME module is loaded

CLI Example:

```
salt '*' gnome.ping user=<username>
```

`salt.modules.gnomedesktop.`**`setClockFormat`**`(`*`clockFormat, **kwargs`*`)`

Set the clock format, either 12h or 24h format.

CLI Example:

```
salt '*' gnome.setClockFormat <12h|24h> user=<username>
```

`salt.modules.gnomedesktop.`**`setClockShowDate`**`(`*`kvalue, **kwargs`*`)`

Set whether the date is visible in the clock

CLI Example:

```
salt '*' gnome.setClockShowDate <True|False> user=<username>
```

`salt.modules.gnomedesktop.`**`setIdleActivation`**`(`*`kvalue, **kwargs`*`)`

Set whether the idle activation is enabled

CLI Example:

```
salt '*' gnome.setIdleActivation <True|False> user=<username>
```

`salt.modules.gnomedesktop.`**`setIdleDelay`**`(`*`delaySeconds, **kwargs`*`)`

Set the current idle delay setting in seconds

CLI Example:

```
salt '*' gnome.setIdleDelay <seconds> user=<username>
```

`salt.modules.gnomedesktop.`**`set_`**`(`*schema=None, key=None, user=None, value=None, **kwargs*`)`

Set key in a particular GNOME schema

CLI Example:

```
salt '*' gnome.set user=<username> schema=org.gnome.desktop.screensaver key=idle-
↪activation-enabled value=False
```

21.9.146 salt.modules.google_chat

Module for sending messages to google chat.

New in version 2019.2.0.

To use this module you need to configure a webhook in the google chat room where you would like the message to be sent, see:

> https://developers.google.com/hangouts/chat/how-tos/webhooks

`salt.modules.google_chat.`**`send_message`**`(`*url, message*`)`

Send a message to the google chat room specified in the webhook url.

```
salt '*' google_chat.send_message "https://chat.googleapis.com/v1/spaces/example_
↪space/messages?key=example_key" "This is a test message"
```

21.9.147 salt.modules.gpg

Manage a GPG keychains, add keys, create keys, retrieve keys from keyservers. Sign, encrypt and sign plus encrypt text and files.

New in version 2015.5.0.

Note: The `python-gnupg` library and `gpg` binary are required to be installed.

`salt.modules.gpg.`**`create_key`**`(`*key_type='RSA', key_length=1024, name_real='Autogenerated Key', name_comment='Generated by SaltStack', name_email=None, subkey_type=None, subkey_length=None, expire_date=None, use_passphrase=False, user=None, gnupghome=None*`)`

Create a key in the GPG keychain

Note: GPG key generation requires *a lot* of entropy and randomness. Difficult to do over a remote connection, consider having another process available which is generating randomness for the machine. Also especially difficult on virtual machines, consider the rng-tools package.

The create_key process takes awhile so increasing the timeout may be necessary, e.g. -t 15.

key_type The type of the primary key to generate. It must be capable of signing. 'RSA' or 'DSA'.

key_length The length of the primary key in bits.

name_real The real name of the user identity which is represented by the key.

name_comment A comment to attach to the user id.

name_email An email address for the user.

subkey_type The type of the secondary key to generate.

subkey_length The length of the secondary key in bits.

expire_date The expiration date for the primary and any secondary key. You can specify an ISO date, A number of days/weeks/months/years, an epoch value, or 0 for a non-expiring key.

use_passphrase Whether to use a passphrase with the signing key. Passphrase is received from Pillar.

user Which user's keychain to access, defaults to user Salt is running as. Passing the user as `salt` will set the GnuPG home directory to the `/etc/salt/gpgkeys`.

gnupghome Specify the location where GPG keyring and related files are stored.

CLI Example:

```
salt -t 15 '*' gpg.create_key
```

`salt.modules.gpg.`**`decrypt`**(*user=None*, *text=None*, *filename=None*, *output=None*, *use_passphrase=False*, *gnupghome=None*, *bare=False*)

Decrypt a message or file

user Which user's keychain to access, defaults to user Salt is running as. Passing the user as `salt` will set the GnuPG home directory to the `/etc/salt/gpgkeys`.

text The encrypted text to decrypt.

filename The encrypted filename to decrypt.

output The filename where the decrypted data will be written, default is standard out.

use_passphrase Whether to use a passphrase with the signing key. Passphrase is received from Pillar.

gnupghome Specify the location where GPG keyring and related files are stored.

bare If `True`, return the (armored) decrypted block as a string without the standard comment/res dict.

CLI Example:

```
salt '*' gpg.decrypt filename='/path/to/important.file.gpg'

salt '*' gpg.decrypt filename='/path/to/important.file.gpg' use_passphrase=True
```

`salt.modules.gpg.`**`delete_key`**(*keyid=None*, *fingerprint=None*, *delete_secret=False*, *user=None*, *gnupghome=None*)

Get a key from the GPG keychain

keyid The keyid of the key to be deleted.

fingerprint The fingerprint of the key to be deleted.

delete_secret Whether to delete a corresponding secret key prior to deleting the public key. Secret keys must be deleted before deleting any corresponding public keys.

user Which user's keychain to access, defaults to user Salt is running as. Passing the user as `salt` will set the GnuPG home directory to the `/etc/salt/gpgkeys`.

gnupghome Specify the location where GPG keyring and related files are stored.

CLI Example:

```
salt '*' gpg.delete_key keyid=3FAD9F1E

salt '*' gpg.delete_key fingerprint=53C96788253E58416D20BCD352952C84C3252192
```

(continues on next page)

(continued from previous page)

```
salt '*' gpg.delete_key keyid=3FAD9F1E user=username

salt '*' gpg.delete_key keyid=3FAD9F1E user=username delete_secret=True
```

`salt.modules.gpg.`**`encrypt`**(*user=None, recipients=None, text=None, filename=None, output=None, sign=None, use_passphrase=False, gnupghome=None, bare=False*)

Encrypt a message or file

user Which user's keychain to access, defaults to user Salt is running as. Passing the user as `salt` will set the GnuPG home directory to the `/etc/salt/gpgkeys`.

recipients The fingerprints for those recipient whom the data is being encrypted for.

text The text to encrypt.

filename The filename to encrypt.

output The filename where the signed file will be written, default is standard out.

sign Whether to sign, in addition to encrypt, the data. `True` to use default key or fingerprint to specify a different key to sign with.

use_passphrase Whether to use a passphrase with the signing key. Passphrase is received from Pillar.

gnupghome Specify the location where GPG keyring and related files are stored.

bare If `True`, return the (armored) encrypted block as a string without the standard comment/res dict.

CLI Example:

```
salt '*' gpg.encrypt text='Hello there.  How are you?'

salt '*' gpg.encrypt filename='/path/to/important.file'

salt '*' gpg.encrypt filename='/path/to/important.file' use_passphrase=True
```

`salt.modules.gpg.`**`export_key`**(*keyids=None, secret=False, user=None, gnupghome=None*)

Export a key from the GPG keychain

keyids The key ID(s) of the key(s) to be exported. Can be specified as a comma separated string or a list. Anything which GnuPG itself accepts to identify a key - for example, the key ID or the fingerprint could be used.

secret Export the secret key identified by the `keyids` information passed.

user Which user's keychain to access, defaults to user Salt is running as. Passing the user as `salt` will set the GnuPG home directory to the `/etc/salt/gpgkeys`.

gnupghome Specify the location where GPG keyring and related files are stored.

CLI Example:

```
salt '*' gpg.export_key keyids=3FAD9F1E

salt '*' gpg.export_key keyids=3FAD9F1E secret=True

salt '*' gpg.export_key keyids="['3FAD9F1E','3FBD8F1E']" user=username
```

`salt.modules.gpg.`**`get_key`**(*keyid=None, fingerprint=None, user=None, gnupghome=None*)

Get a key from the GPG keychain

keyid The key ID (short or long) of the key to be retrieved.

fingerprint The fingerprint of the key to be retrieved.

user Which user's keychain to access, defaults to user Salt is running as. Passing the user as `salt` will set the GnuPG home directory to the `/etc/salt/gpgkeys`.

gnupghome Specify the location where GPG keyring and related files are stored.

CLI Example:

```
salt '*' gpg.get_key keyid=3FAD9F1E

salt '*' gpg.get_key fingerprint=53C96788253E58416D20BCD352952C84C3252192

salt '*' gpg.get_key keyid=3FAD9F1E user=username
```

`salt.modules.gpg.`**`get_secret_key`**(*keyid=None, fingerprint=None, user=None, gnupghome=None*)
Get a key from the GPG keychain

keyid The key ID (short or long) of the key to be retrieved.

fingerprint The fingerprint of the key to be retrieved.

user Which user's keychain to access, defaults to user Salt is running as. Passing the user as `salt` will set the GnuPG home directory to the `/etc/salt/gpgkeys`.

gnupghome Specify the location where GPG keyring and related files are stored.

CLI Example:

```
salt '*' gpg.get_secret_key keyid=3FAD9F1E

salt '*' gpg.get_secret_key fingerprint=53C96788253E58416D20BCD352952C84C3252192

salt '*' gpg.get_secret_key keyid=3FAD9F1E user=username
```

`salt.modules.gpg.`**`import_key`**(*text=None, filename=None, user=None, gnupghome=None*)
Import a key from text or file

text The text containing to import.

filename The filename containing the key to import.

user Which user's keychain to access, defaults to user Salt is running as. Passing the user as `salt` will set the GnuPG home directory to the `/etc/salt/gpgkeys`.

gnupghome Specify the location where GPG keyring and related files are stored.

CLI Example:

```
salt '*' gpg.import_key text='-----BEGIN PGP PUBLIC KEY BLOCK-----\n ... -----END
↪PGP PUBLIC KEY BLOCK-----'
salt '*' gpg.import_key filename='/path/to/public-key-file'
```

`salt.modules.gpg.`**`list_keys`**(*user=None, gnupghome=None*)
List keys in GPG keychain

user Which user's keychain to access, defaults to user Salt is running as. Passing the user as `salt` will set the GnuPG home directory to the `/etc/salt/gpgkeys`.

gnupghome Specify the location where GPG keyring and related files are stored.

CLI Example:

```
salt '*' gpg.list_keys
```

`salt.modules.gpg.`**`list_secret_keys`**`(`*user=None, gnupghome=None*`)`

List secret keys in GPG keychain

user Which user's keychain to access, defaults to user Salt is running as. Passing the user as `salt` will set the GnuPG home directory to the `/etc/salt/gpgkeys`.

gnupghome Specify the location where GPG keyring and related files are stored.

CLI Example:

```
salt '*' gpg.list_secret_keys
```

`salt.modules.gpg.`**`receive_keys`**`(`*keyserver=None, keys=None, user=None, gnupghome=None*`)`

Receive key(s) from keyserver and add them to keychain

keyserver Keyserver to use for searching for GPG keys, defaults to pgp.mit.edu

keys The keyID(s) to retrieve from the keyserver. Can be specified as a comma separated string or a list.

user Which user's keychain to access, defaults to user Salt is running as. Passing the user as `salt` will set the GnuPG home directory to the `/etc/salt/gpgkeys`.

gnupghome Specify the location where GPG keyring and related files are stored.

CLI Example:

```
salt '*' gpg.receive_keys keys='3FAD9F1E'

salt '*' gpg.receive_keys keys="['3FAD9F1E','3FBD9F2E']"

salt '*' gpg.receive_keys keys=3FAD9F1E user=username
```

`salt.modules.gpg.`**`search_keys`**`(`*text, keyserver=None, user=None*`)`

Search keys from keyserver

text Text to search the keyserver for, e.g. email address, keyID or fingerprint.

keyserver Keyserver to use for searching for GPG keys, defaults to pgp.mit.edu.

user Which user's keychain to access, defaults to user Salt is running as. Passing the user as `salt` will set the GnuPG home directory to the `/etc/salt/gpgkeys`.

CLI Example:

```
salt '*' gpg.search_keys user@example.com

salt '*' gpg.search_keys user@example.com keyserver=keyserver.ubuntu.com

salt '*' gpg.search_keys user@example.com keyserver=keyserver.ubuntu.com
↪user=username
```

`salt.modules.gpg.`**`sign`**`(`*user=None, keyid=None, text=None, filename=None, output=None, use_passphrase=False, gnupghome=None*`)`

Sign message or file

user Which user's keychain to access, defaults to user Salt is running as. Passing the user as `salt` will set the GnuPG home directory to the `/etc/salt/gpgkeys`.

keyid The keyid of the key to set the trust level for, defaults to first key in the secret keyring.

text The text to sign.

filename The filename to sign.

output The filename where the signed file will be written, default is standard out.

use_passphrase Whether to use a passphrase with the signing key. Passphrase is received from Pillar.

gnupghome Specify the location where GPG keyring and related files are stored.

CLI Example:

```
salt '*' gpg.sign text='Hello there.  How are you?'

salt '*' gpg.sign filename='/path/to/important.file'

salt '*' gpg.sign filename='/path/to/important.file' use_passphrase=True
```

`salt.modules.gpg.`**`trust_key`**(*keyid=None*, *fingerprint=None*, *trust_level=None*, *user=None*)

Set the trust level for a key in GPG keychain

keyid The keyid of the key to set the trust level for.

fingerprint The fingerprint of the key to set the trust level for.

trust_level The trust level to set for the specified key, must be one of the following: expired, unknown, not_trusted, marginally, fully, ultimately

user Which user's keychain to access, defaults to user Salt is running as. Passing the user as `salt` will set the GnuPG home directory to the `/etc/salt/gpgkeys`.

CLI Example:

```
salt '*' gpg.trust_key keyid='3FAD9F1E' trust_level='marginally'
salt '*' gpg.trust_key fingerprint='53C96788253E58416D20BCD352952C84C3252192'
↪trust_level='not_trusted'
salt '*' gpg.trust_key keys=3FAD9F1E trust_level='ultimately' user='username'
```

`salt.modules.gpg.`**`verify`**(*text=None*, *user=None*, *filename=None*, *gnupghome=None*, *signature=None*, *trustmodel=None*)

Verify a message or file

text The text to verify.

filename The filename to verify.

user Which user's keychain to access, defaults to user Salt is running as. Passing the user as `salt` will set the GnuPG home directory to the `/etc/salt/gpgkeys`.

gnupghome Specify the location where GPG keyring and related files are stored.

signature Specify the filename of a detached signature.

New in version 2018.3.0.

trustmodel

Explicitly define the used trust model. One of:

- pgp
- classic
- tofu
- tofu+pgp
- direct

- always
- auto

New in version 2019.2.0.

CLI Example:

```
salt '*' gpg.verify text='Hello there.  How are you?'
salt '*' gpg.verify filename='/path/to/important.file'
salt '*' gpg.verify filename='/path/to/important.file' use_passphrase=True
salt '*' gpg.verify filename='/path/to/important.file' trustmodel=direct
```

21.9.148 salt.modules.grafana4 module

Module for working with the Grafana v4 API

New in version 2017.7.0.

depends requests

configuration This module requires a configuration profile to be configured in the minion config, minion pillar, or master config. The module will use the 'grafana' key by default, if defined.

For example:

```
grafana:
    grafana_url: http://grafana.localhost
    grafana_user: admin
    grafana_password: admin
    grafana_timeout: 3
```

`salt.modules.grafana4.`**`create_datasource`**`(`*orgname=None, profile='grafana', **kwargs*`)`

Create a new datasource in an organisation.

name Name of the data source.

type Type of the datasource ('graphite', 'influxdb' etc.).

access Use proxy or direct.

url The URL to the data source API.

user Optional - user to authenticate with the data source.

password Optional - password to authenticate with the data source.

database Optional - database to use with the data source.

basicAuth Optional - set to True to use HTTP basic auth to authenticate with the data source.

basicAuthUser Optional - HTTP basic auth username.

basicAuthPassword Optional - HTTP basic auth password.

jsonData Optional - additional json data to post (eg. "timeInterval").

isDefault Optional - set data source as default.

withCredentials Optional - Whether credentials such as cookies or auth headers should be sent with cross-site requests.

typeLogoUrl Optional - Logo to use for this datasource.

orgname Name of the organization in which the data source should be created.

profile Configuration profile used to connect to the Grafana instance. Default is 'grafana'.

CLI Example:

```
salt '*' grafana4.create_datasource
```

salt.modules.grafana4.**create_org**(*profile='grafana'*, ***kwargs*)

Create a new organization.

name Name of the organization.

profile Configuration profile used to connect to the Grafana instance. Default is 'grafana'.

CLI Example:

```
salt '*' grafana4.create_org <name>
```

salt.modules.grafana4.**create_org_user**(*orgname=None*, *profile='grafana'*, ***kwargs*)

Add user to the organization.

loginOrEmail Login or email of the user.

role

Role of the user for this organization. Should be one of:

- Admin
- Editor
- Read Only Editor
- Viewer

orgname Name of the organization in which users are added.

profile Configuration profile used to connect to the Grafana instance. Default is 'grafana'.

CLI Example:

```
salt '*' grafana4.create_org_user <orgname> loginOrEmail=<loginOrEmail> role=
↪<role>
```

salt.modules.grafana4.**create_update_dashboard**(*orgname=None*, *profile='grafana'*, ***kwargs*)

Create or update a dashboard.

dashboard A dict that defines the dashboard to create/update.

overwrite Whether the dashboard should be overwritten if already existing.

orgname Name of the organization.

profile Configuration profile used to connect to the Grafana instance. Default is 'grafana'.

CLI Example:

```
salt '*' grafana4.create_update_dashboard dashboard=<dashboard> overwrite=True
↪orgname=<orgname>
```

salt.modules.grafana4.**create_user**(*profile='grafana'*, ***kwargs*)

Create a new user.

login Login of the new user.

password Password of the new user.

email Email of the new user.

name Optional - Full name of the new user.

profile Configuration profile used to connect to the Grafana instance. Default is 'grafana'.

CLI Example:

```
salt '*' grafana4.create_user login=<login> password=<password> email=<email>
```

`salt.modules.grafana4.`**`delete_dashboard`**(*slug, orgname=None, profile='grafana'*)

Delete a dashboard.

slug Slug (name) of the dashboard.

orgname Name of the organization.

profile Configuration profile used to connect to the Grafana instance. Default is 'grafana'.

CLI Example:

```
salt '*' grafana4.delete_dashboard <slug>
```

`salt.modules.grafana4.`**`delete_datasource`**(*datasourceid, orgname=None, profile='grafana'*)

Delete a datasource.

datasourceid Id of the datasource.

profile Configuration profile used to connect to the Grafana instance. Default is 'grafana'.

CLI Example:

```
salt '*' grafana4.delete_datasource <datasource_id>
```

`salt.modules.grafana4.`**`delete_org`**(*orgid, profile='grafana'*)

Delete an organization.

orgid Id of the organization.

profile Configuration profile used to connect to the Grafana instance. Default is 'grafana'.

CLI Example:

```
salt '*' grafana4.delete_org <org_id>
```

`salt.modules.grafana4.`**`delete_org_user`**(*userid, orgname=None, profile='grafana'*)

Remove user from the organization.

userid Id of the user.

orgname Name of the organization in which users are updated.

profile Configuration profile used to connect to the Grafana instance. Default is 'grafana'.

CLI Example:

```
salt '*' grafana4.delete_org_user <user_id> <orgname>
```

`salt.modules.grafana4.`**`delete_user`**(*userid, profile='grafana'*)

Delete a user.

userid Id of the user.

profile Configuration profile used to connect to the Grafana instance. Default is 'grafana'.

CLI Example:

```
salt '*' grafana4.delete_user <user_id>
```

`salt.modules.grafana4.`**`delete_user_org`**(*userid*, *orgid*, *profile='grafana'*)

Remove a user from an organization.

userid Id of the user.

orgid Id of the organization.

profile Configuration profile used to connect to the Grafana instance. Default is 'grafana'.

CLI Example:

```
salt '*' grafana4.delete_user_org <user_id> <org_id>
```

`salt.modules.grafana4.`**`get_dashboard`**(*slug*, *orgname=None*, *profile='grafana'*)

Get a dashboard.

slug Slug (name) of the dashboard.

orgname Name of the organization.

profile Configuration profile used to connect to the Grafana instance. Default is 'grafana'.

CLI Example:

```
salt '*' grafana4.get_dashboard <slug>
```

`salt.modules.grafana4.`**`get_datasource`**(*name*, *orgname=None*, *profile='grafana'*)

Show a single datasource in an organisation.

name Name of the datasource.

orgname Name of the organization.

profile Configuration profile used to connect to the Grafana instance. Default is 'grafana'.

CLI Example:

```
salt '*' grafana4.get_datasource <name> <orgname>
```

`salt.modules.grafana4.`**`get_datasources`**(*orgname=None*, *profile='grafana'*)

List all datasources in an organisation.

orgname Name of the organization.

profile Configuration profile used to connect to the Grafana instance. Default is 'grafana'.

CLI Example:

```
salt '*' grafana4.get_datasources <orgname>
```

`salt.modules.grafana4.`**`get_org`**(*name*, *profile='grafana'*)

Show a single organization.

name Name of the organization.

profile Configuration profile used to connect to the Grafana instance. Default is 'grafana'.

CLI Example:

```
salt '*' grafana4.get_org <name>
```

`salt.modules.grafana4.`**`get_org_address`**(*orgname=None, profile='grafana'*)

Get the organization address.

orgname Name of the organization in which users are updated.

profile Configuration profile used to connect to the Grafana instance. Default is 'grafana'.

CLI Example:

```
salt '*' grafana4.get_org_address <orgname>
```

`salt.modules.grafana4.`**`get_org_prefs`**(*orgname=None, profile='grafana'*)

Get the organization preferences.

orgname Name of the organization in which users are updated.

profile Configuration profile used to connect to the Grafana instance. Default is 'grafana'.

CLI Example:

```
salt '*' grafana4.get_org_prefs <orgname>
```

`salt.modules.grafana4.`**`get_org_users`**(*orgname=None, profile='grafana'*)

Get the list of users that belong to the organization.

orgname Name of the organization.

profile Configuration profile used to connect to the Grafana instance. Default is 'grafana'.

CLI Example:

```
salt '*' grafana4.get_org_users <orgname>
```

`salt.modules.grafana4.`**`get_orgs`**(*profile='grafana'*)

List all organizations.

profile Configuration profile used to connect to the Grafana instance. Default is 'grafana'.

CLI Example:

```
salt '*' grafana4.get_orgs
```

`salt.modules.grafana4.`**`get_user`**(*login, profile='grafana'*)

Show a single user.

login Login of the user.

profile Configuration profile used to connect to the Grafana instance. Default is 'grafana'.

CLI Example:

```
salt '*' grafana4.get_user <login>
```

`salt.modules.grafana4.`**`get_user_data`**(*userid, profile='grafana'*)

Get user data.

userid Id of the user.

profile Configuration profile used to connect to the Grafana instance. Default is 'grafana'.

CLI Example:

```
salt '*' grafana4.get_user_data <user_id>
```

`salt.modules.grafana4.`**`get_user_orgs`**(*userid*, *profile='grafana'*)

Get the list of organisations a user belong to.

userid Id of the user.

profile Configuration profile used to connect to the Grafana instance. Default is 'grafana'.

CLI Example:

```
salt '*' grafana4.get_user_orgs <user_id>
```

`salt.modules.grafana4.`**`get_users`**(*profile='grafana'*)

List all users.

profile Configuration profile used to connect to the Grafana instance. Default is 'grafana'.

CLI Example:

```
salt '*' grafana4.get_users
```

`salt.modules.grafana4.`**`switch_org`**(*orgname*, *profile='grafana'*)

Switch the current organization.

name Name of the organization to switch to.

profile Configuration profile used to connect to the Grafana instance. Default is 'grafana'.

CLI Example:

```
salt '*' grafana4.switch_org <name>
```

`salt.modules.grafana4.`**`update_datasource`**(*datasourceid*, *orgname=None*, *profile='grafana'*, ***kwargs*)

Update a datasource.

datasourceid Id of the datasource.

name Name of the data source.

type Type of the datasource ('graphite', 'influxdb' etc.).

access Use proxy or direct.

url The URL to the data source API.

user Optional - user to authenticate with the data source.

password Optional - password to authenticate with the data source.

database Optional - database to use with the data source.

basicAuth Optional - set to True to use HTTP basic auth to authenticate with the data source.

basicAuthUser Optional - HTTP basic auth username.

basicAuthPassword Optional - HTTP basic auth password.

jsonData Optional - additional json data to post (eg. "timeInterval").

isDefault Optional - set data source as default.

withCredentials Optional - Whether credentials such as cookies or auth headers should be sent with cross-site requests.

typeLogoUrl Optional - Logo to use for this datasource.

profile Configuration profile used to connect to the Grafana instance. Default is 'grafana'.

CLI Example:

```
salt '*' grafana4.update_datasource <datasourceid>
```

`salt.modules.grafana4.`**`update_org`**(*orgid*, *profile='grafana'*, ***kwargs*)
Update an existing organization.

orgid Id of the organization.

name New name of the organization.

profile Configuration profile used to connect to the Grafana instance. Default is 'grafana'.

CLI Example:

```
salt '*' grafana4.update_org <org_id> name=<name>
```

`salt.modules.grafana4.`**`update_org_address`**(*orgname=None*, *profile='grafana'*, ***kwargs*)
Update the organization address.

orgname Name of the organization in which users are updated.

address1 Optional - address1 of the org.

address2 Optional - address2 of the org.

city Optional - city of the org.

zip_code Optional - zip_code of the org.

state Optional - state of the org.

country Optional - country of the org.

profile Configuration profile used to connect to the Grafana instance. Default is 'grafana'.

CLI Example:

```
salt '*' grafana4.update_org_address <orgname> country=<country>
```

`salt.modules.grafana4.`**`update_org_prefs`**(*orgname=None*, *profile='grafana'*, ***kwargs*)
Update the organization preferences.

orgname Name of the organization in which users are updated.

theme Selected theme for the org.

homeDashboardId Home dashboard for the org.

timezone Timezone for the org (one of: "browser", "utc", or "").

profile Configuration profile used to connect to the Grafana instance. Default is 'grafana'.

CLI Example:

```
salt '*' grafana4.update_org_prefs <orgname> theme=<theme> timezone=<timezone>
```

`salt.modules.grafana4.`**`update_org_user`**(*userid*, *orgname=None*, *profile='grafana'*, ***kwargs*)
Update user role in the organization.

userid Id of the user.

loginOrEmail Login or email of the user.

role

Role of the user for this organization. Should be one of:

- Admin
- Editor
- Read Only Editor
- Viewer

orgname Name of the organization in which users are updated.

profile Configuration profile used to connect to the Grafana instance. Default is 'grafana'.

CLI Example:

```
salt '*' grafana4.update_org_user <user_id> <orgname> loginOrEmail=<loginOrEmail>␣
↪role=<role>
```

`salt.modules.grafana4.`**`update_user`**(*userid*, *profile='grafana'*, ***kwargs*)

Update an existing user.

userid Id of the user.

login Optional - Login of the user.

email Optional - Email of the user.

name Optional - Full name of the user.

profile Configuration profile used to connect to the Grafana instance. Default is 'grafana'.

CLI Example:

```
salt '*' grafana4.update_user <user_id> login=<login> email=<email>
```

`salt.modules.grafana4.`**`update_user_password`**(*userid*, *profile='grafana'*, ***kwargs*)

Update a user password.

userid Id of the user.

password New password of the user.

profile Configuration profile used to connect to the Grafana instance. Default is 'grafana'.

CLI Example:

```
salt '*' grafana4.update_user_password <user_id> password=<password>
```

`salt.modules.grafana4.`**`update_user_permissions`**(*userid*, *profile='grafana'*, ***kwargs*)

Update a user password.

userid Id of the user.

isGrafanaAdmin Whether user is a Grafana admin.

profile Configuration profile used to connect to the Grafana instance. Default is 'grafana'.

CLI Example:

```
salt '*' grafana4.update_user_permissions <user_id> isGrafanaAdmin=<true|false>
```

21.9.149 salt.modules.grains

Return/control aspects of the grains data

Grains set or altered with this module are stored in the 'grains' file on the minions. By default, this file is located at: `/etc/salt/grains`

Note: This does **NOT** override any grains set in the minion config file.

`salt.modules.grains.`**`append`**`(`*key*, *val*, *convert=False*, *delimiter=':'*`)`

New in version 0.17.0.

Append a value to a list in the grains config file. If the grain doesn't exist, the grain key is added and the value is appended to the new grain as a list item.

key The grain key to be appended to

val The value to append to the grain key

convert If convert is True, convert non-list contents into a list. If convert is False and the grain contains non-list contents, an error is given. Defaults to False.

delimiter The key can be a nested dict key. Use this parameter to specify the delimiter you use, instead of the default `:`. You can now append values to a list in nested dictionary grains. If the list doesn't exist at this level, it will be created.

New in version 2014.7.6.

CLI Example:

```
salt '*' grains.append key val
```

`salt.modules.grains.`**`delkey`**`(`*key*`)`

New in version 2017.7.0.

Remove a grain completely from the grain system, this will remove the grain key and value

key The grain key from which to delete the value.

CLI Example:

```
salt '*' grains.delkey key
```

`salt.modules.grains.`**`delval`**`(`*key*, *destructive=False*`)`

New in version 0.17.0.

Delete a grain value from the grains config file. This will just set the grain value to `None`. To completely remove the grain, run `grains.delkey` or pass `destructive=True` to `grains.delval`.

key The grain key from which to delete the value.

destructive Delete the key, too. Defaults to False.

CLI Example:

```
salt '*' grains.delval key
```

`salt.modules.grains.`**`equals`**`(`*key*, *value*`)`

Used to make sure the minion's grain key/value matches.

Returns `True` if matches otherwise `False`.

New in version 2017.7.0.

CLI Example:

```
salt '*' grains.equals fqdn <expected_fqdn>
salt '*' grains.equals systemd:version 219
```

`salt.modules.grains.`**`fetch`**`(`*key, default='', delimiter=':', ordered=True*`)`

Attempt to retrieve the named value from grains, if the named value is not available return the passed default. The default return is an empty string.

The value can also represent a value in a nested dict using a ":" delimiter for the dict. This means that if a dict in grains looks like this:

```
{'pkg': {'apache': 'httpd'}}
```

To retrieve the value associated with the apache key in the pkg dict this key can be passed:

```
pkg:apache
```

Parameters

- **`delimiter`** – Specify an alternate delimiter to use when traversing a nested dict. This is useful for when the desired key contains a colon. See CLI example below for usage.

  New in version 2014.7.0.

- **`ordered`** – Outputs an ordered dict if applicable (default: True)

  New in version 2016.11.0.

CLI Example:

```
salt '*' grains.get pkg:apache
salt '*' grains.get abc::def|ghi delimiter='|'
```

`salt.modules.grains.`**`filter_by`**`(`*lookup_dict, grain='os_family', merge=None, default='default', base=None*`)`

New in version 0.17.0.

Look up the given grain in a given dictionary for the current OS and return the result

Although this may occasionally be useful at the CLI, the primary intent of this function is for use in Jinja to make short work of creating lookup tables for OS-specific data. For example:

```
{% set apache = salt['grains.filter_by']({
    'Debian': {'pkg': 'apache2', 'srv': 'apache2'},
    'RedHat': {'pkg': 'httpd', 'srv': 'httpd'},
}, default='Debian') %}

myapache:
  pkg.installed:
    - name: {{ apache.pkg }}
  service.running:
    - name: {{ apache.srv }}
```

Values in the lookup table may be overridden by values in Pillar. An example Pillar to override values in the example above could be as follows:

```
apache:
  lookup:
    pkg: apache_13
    srv: apache
```

The call to `filter_by()` would be modified as follows to reference those Pillar values:

```
{% set apache = salt['grains.filter_by']({
    ...
}, merge=salt['pillar.get']('apache:lookup')) %}
```

Parameters

- **`lookup_dict`** – A dictionary, keyed by a grain, containing a value or values relevant to systems matching that grain. For example, a key could be the grain for an OS and the value could the name of a package on that particular OS.

  Changed in version 2016.11.0: The dictionary key could be a globbing pattern. The function will return the corresponding `lookup_dict` value where grain value matches the pattern. For example:

  ```
  # this will render 'got some salt' if Minion ID begins from 'salt'
  salt '*' grains.filter_by '{salt*: got some salt, default: salt is⊠
  ↪not here}' id
  ```

- **`grain`** – The name of a grain to match with the current system's grains. For example, the value of the "os_family" grain for the current system could be used to pull values from the `lookup_dict` dictionary.

  Changed in version 2016.11.0: The grain value could be a list. The function will return the `lookup_dict` value for a first found item in the list matching one of the `lookup_dict` keys.

- **`merge`** – A dictionary to merge with the results of the grain selection from `lookup_dict`. This allows Pillar to override the values in the `lookup_dict`. This could be useful, for example, to override the values for non-standard package names such as when using a different Python version from the default Python version provided by the OS (e.g., `python26-mysql` instead of `python-mysql`).

- **`default`** – default lookup_dict's key used if the grain does not exists or if the grain value has no match on lookup_dict. If unspecified the value is "default".

  New in version 2014.1.0.

- **`base`** – A lookup_dict key to use for a base dictionary. The grain-selected `lookup_dict` is merged over this and then finally the `merge` dictionary is merged. This allows common values for each case to be collected in the base and overridden by the grain selection dictionary and the merge dictionary. Default is unset.

  New in version 2015.5.0.

CLI Example:

```
salt '*' grains.filter_by '{Debian: Debheads rule, RedHat: I love my hat}'
# this one will render {D: {E: I, G: H}, J: K}
salt '*' grains.filter_by '{A: B, C: {D: {E: F, G: H}}}' 'xxx' '{D: {E: I}, J: K}
↪' 'C'
# next one renders {A: {B: G}, D: J}
```

(continues on next page)

(continued from previous page)

```
salt '*' grains.filter_by '{default: {A: {B: C}, D: E}, F: {A: {B: G}}, H: {D: I}}
↪' 'xxx' '{D: J}' 'F' 'default'
# next same as above when default='H' instead of 'F' renders {A: {B: C}, D: J}
```

`salt.modules.grains.`**`get`**`(`*key, default=”, delimiter=’:’, ordered=True*`)`

Attempt to retrieve the named value from grains, if the named value is not available return the passed default. The default return is an empty string.

The value can also represent a value in a nested dict using a ”:” delimiter for the dict. This means that if a dict in grains looks like this:

```
{'pkg': {'apache': 'httpd'}}
```

To retrieve the value associated with the apache key in the pkg dict this key can be passed:

```
pkg:apache
```

Parameters

- **`delimiter`** – Specify an alternate delimiter to use when traversing a nested dict. This is useful for when the desired key contains a colon. See CLI example below for usage.

  New in version 2014.7.0.

- **`ordered`** – Outputs an ordered dict if applicable (default: True)

  New in version 2016.11.0.

CLI Example:

```
salt '*' grains.get pkg:apache
salt '*' grains.get abc::def|ghi delimiter='|'
```

`salt.modules.grains.`**`get_or_set_hash`**`(`*`name, length=8, chars=’abcdefghijklmnopqrstuvwxyz0123456789!@#$%^&*(-_=+)’`*`)`

Perform a one-time generation of a hash and write it to the local grains. If that grain has already been set return the value instead.

This is useful for generating passwords or keys that are specific to a single minion that don’t need to be stored somewhere centrally.

State Example:

```
some_mysql_user:
  mysql_user:
    - present
    - host: localhost
    - password: {{ salt['grains.get_or_set_hash']('mysql:some_mysql_user') }}
```

CLI Example:

```
salt '*' grains.get_or_set_hash 'django:SECRET_KEY' 50
```

> **Warning:** This function could return strings which may contain characters which are reserved as directives by the YAML parser, such as strings beginning with %. To avoid issues when using the output of this function in an SLS file containing YAML+Jinja, surround the call with single quotes.

`salt.modules.grains.`**`has_value`**`(`*key*`)`

Determine whether a key exists in the grains dictionary.

Given a grains dictionary that contains the following structure:

```
{'pkg': {'apache': 'httpd'}}
```

One would determine if the apache key in the pkg dict exists by:

```
pkg:apache
```

CLI Example:

```
salt '*' grains.has_value pkg:apache
```

`salt.modules.grains.`**`item`**`(`**args, **kwargs*`)`

Return one or more grains

CLI Example:

```
salt '*' grains.item os
salt '*' grains.item os osrelease oscodename
```

Sanitized CLI Example:

```
salt '*' grains.item host sanitize=True
```

`salt.modules.grains.`**`items`**`(`*sanitize=False*`)`

Return all of the minion's grains

CLI Example:

```
salt '*' grains.items
```

Sanitized CLI Example:

```
salt '*' grains.items sanitize=True
```

`salt.modules.grains.`**`ls`**`()`

Return a list of all available grains

CLI Example:

```
salt '*' grains.ls
```

`salt.modules.grains.`**`remove`**`(`*key, val, delimiter=':'*`)`

New in version 0.17.0.

Remove a value from a list in the grains config file

key The grain key to remove.

val The value to remove.

delimiter The key can be a nested dict key. Use this parameter to specify the delimiter you use, instead of the default `:`. You can now append values to a list in nested dictionary grains. If the list doesn't exist at this level, it will be created.

New in version 2015.8.2.

CLI Example:

```
salt '*' grains.remove key val
```

`salt.modules.grains.`**`set`**(*key*, *val=''*, *force=False*, *destructive=False*, *delimiter=':'*)

Set a key to an arbitrary value. It is used like setval but works with nested keys.

This function is conservative. It will only overwrite an entry if its value and the given one are not a list or a dict. The `force` parameter is used to allow overwriting in all cases.

New in version 2015.8.0.

Parameters

- **`force`** – Force writing over existing entry if given or existing values are list or dict. Defaults to False.
- **`destructive`** – If an operation results in a key being removed, delete the key, too. Defaults to False.
- **`delimiter`** – Specify an alternate delimiter to use when traversing a nested dict, the default being **:**

CLI Example:

```
salt '*' grains.set 'apps:myApp:port' 2209
salt '*' grains.set 'apps:myApp' '{port: 2209}'
```

`salt.modules.grains.`**`setval`**(*key*, *val*, *destructive=False*)

Set a grains value in the grains config file

key The grain key to be set.

val The value to set the grain key to.

destructive If an operation results in a key being removed, delete the key, too. Defaults to False.

CLI Example:

```
salt '*' grains.setval key val
salt '*' grains.setval key "{'sub-key': 'val', 'sub-key2': 'val2'}"
```

`salt.modules.grains.`**`setvals`**(*grains*, *destructive=False*)

Set new grains values in the grains config file

destructive If an operation results in a key being removed, delete the key, too. Defaults to False.

CLI Example:

```
salt '*' grains.setvals "{'key1': 'val1', 'key2': 'val2'}"
```

21.9.150 salt.modules.groupadd

Manage groups on Linux, OpenBSD and NetBSD

Important: If you feel that Salt should be using this module to manage groups on a minion, and it is using a different module (or gives an error similar to *'group.info' is not available*), see *here*.

`salt.modules.groupadd.`**`add`**(*name*, *gid=None*, *system=False*, *root=None*)

Add the specified group

name Name of the new group

gid Use GID for the new group

system Create a system account

root Directory to chroot into

CLI Example:

```
salt '*' group.add foo 3456
```

`salt.modules.groupadd.`**`adduser`**(*name*, *username*, *root=None*)

Add a user in the group.

name Name of the group to modify

username Username to add to the group

root Directory to chroot into

CLI Example:

```
salt '*' group.adduser foo bar
```

Verifies if a valid username 'bar' as a member of an existing group 'foo', if not then adds it.

`salt.modules.groupadd.`**`chgid`**(*name*, *gid*, *root=None*)

Change the gid for a named group

name Name of the group to modify

gid Change the group ID to GID

root Directory to chroot into

CLI Example:

```
salt '*' group.chgid foo 4376
```

`salt.modules.groupadd.`**`delete`**(*name*, *root=None*)

Remove the named group

name Name group to delete

root Directory to chroot into

CLI Example:

```
salt '*' group.delete foo
```

`salt.modules.groupadd.`**`deluser`**(*name*, *username*, *root=None*)

Remove a user from the group.

name Name of the group to modify

username Username to delete from the group

root Directory to chroot into

CLI Example:

```
salt '*' group.deluser foo bar
```

Removes a member user 'bar' from a group 'foo'. If group is not present then returns True.

`salt.modules.groupadd.`**`getent`**`(`*refresh=False*, *root=None*`)`

Return info on all groups

refresh Force a refresh of group information

root Directory to chroot into

CLI Example:

```
salt '*' group.getent
```

`salt.modules.groupadd.`**`info`**`(`*name*, *root=None*`)`

Return information about a group

name Name of the group

root Directory to chroot into

CLI Example:

```
salt '*' group.info foo
```

`salt.modules.groupadd.`**`members`**`(`*name*, *members_list*, *root=None*`)`

Replaces members of the group with a provided list.

name Name of the group to modify

members_list Username list to set into the group

root Directory to chroot into

CLI Example:

> salt '*' group.members foo 'user1,user2,user3,…'

Replaces a membership list for a local group 'foo'. foo:x:1234:user1,user2,user3,…

21.9.151 salt.modules.grub_legacy

Support for GRUB Legacy

`salt.modules.grub_legacy.`**`conf`**`()`

Parse GRUB conf file

CLI Example:

```
salt '*' grub.conf
```

`salt.modules.grub_legacy.`**`version`**`()`

Return server version from grub –version

CLI Example:

```
salt '*' grub.version
```

21.9.152 salt.modules.guestfs

Interact with virtual machine images via libguestfs

depends

- libguestfs

`salt.modules.guestfs.`**`mount`**`(`*location*, *access='rw'*, *root=None*`)`

Mount an image

CLI Example:

```
salt '*' guest.mount /srv/images/fedora.qcow
```

21.9.153 salt.modules.hadoop

Support for hadoop

maintainer Yann Jouanin <yann.jouanin@intelunix.fr>

maturity new

depends

platform linux

`salt.modules.hadoop.`**`dfs`**`(`*command=None*, **args*`)`

Execute a command on DFS

CLI Example:

```
salt '*' hadoop.dfs ls /
```

`salt.modules.hadoop.`**`dfs_absent`**`(`*path*`)`

Check if a file or directory is absent on the distributed FS.

CLI Example:

```
salt '*' hadoop.dfs_absent /some_random_file
```

Returns True if the file is absent

`salt.modules.hadoop.`**`dfs_present`**`(`*path*`)`

Check if a file or directory is present on the distributed FS.

CLI Example:

```
salt '*' hadoop.dfs_present /some_random_file
```

Returns True if the file is present

`salt.modules.hadoop.`**`dfsadmin_report`**`(`*arg=None*`)`

New in version 2019.2.0.

Reports basic filesystem information and statistics. Optional flags may be used to filter the list of displayed DataNodes.

arg [live] [dead] [decommissioning]

CLI Example:

```
salt '*' hadoop.dfsadmin -report
```

`salt.modules.hadoop.`**`namenode_format`**`(`*force=None*`)`

Format a name node

```
salt '*' hadoop.namenode_format force=True
```

`salt.modules.hadoop.`**`version`**`()`

Return version from hadoop version

CLI Example:

```
salt '*' hadoop.version
```

21.9.154 salt.modules.haproxyconn

Support for haproxy

New in version 2014.7.0.

`salt.modules.haproxyconn.`**`disable_server`**`(`*name, backend, socket='/var/run/haproxy.sock'*`)`

Disable server in haproxy.

name Server to disable

backend haproxy backend, or all backends if "*" is supplied

socket haproxy stats socket, default `/var/run/haproxy.sock`

CLI Example:

```
salt '*' haproxy.disable_server db1.example.com mysql
```

`salt.modules.haproxyconn.`**`enable_server`**`(`*name, backend, socket='/var/run/haproxy.sock'*`)`

Enable Server in haproxy

name Server to enable

backend haproxy backend, or all backends if "*" is supplied

socket haproxy stats socket, default `/var/run/haproxy.sock`

CLI Example:

```
salt '*' haproxy.enable_server web1.example.com www
```

`salt.modules.haproxyconn.`**`get_backend`**`(`*backend, socket='/var/run/haproxy.sock'*`)`

Receive information about a specific backend.

backend haproxy backend

socket haproxy stats socket, default `/var/run/haproxy.sock`

CLI Example:

```
salt '*' haproxy.get_backend mysql
```

`salt.modules.haproxyconn.`**`get_sessions`**`(`*name, backend, socket='/var/run/haproxy.sock'*`)`

New in version 2016.11.0.

Get number of current sessions on server in backend (scur)

name Server name

backend haproxy backend

socket haproxy stats socket, default `/var/run/haproxy.sock`

CLI Example:

```
salt '*' haproxy.get_sessions web1.example.com www
```

`salt.modules.haproxyconn.`**`get_weight`**(*name*, *backend*, *socket='/var/run/haproxy.sock'*)

Get server weight

name Server name

backend haproxy backend

socket haproxy stats socket, default `/var/run/haproxy.sock`

CLI Example:

```
salt '*' haproxy.get_weight web1.example.com www
```

`salt.modules.haproxyconn.`**`list_backends`**(*servers=True*, *socket='/var/run/haproxy.sock'*)

List HaProxy Backends

socket haproxy stats socket, default `/var/run/haproxy.sock`

servers list backends with servers

CLI Example:

```
salt '*' haproxy.list_backends
```

`salt.modules.haproxyconn.`**`list_frontends`**(*socket='/var/run/haproxy.sock'*)

List HaProxy frontends

socket haproxy stats socket, default `/var/run/haproxy.sock`

CLI Example:

```
salt '*' haproxy.list_frontends
```

`salt.modules.haproxyconn.`**`list_servers`**(*backend*, *socket='/var/run/haproxy.sock'*, *objectify=False*)

List servers in haproxy backend.

backend haproxy backend

socket haproxy stats socket, default `/var/run/haproxy.sock`

CLI Example:

```
salt '*' haproxy.list_servers mysql
```

`salt.modules.haproxyconn.`**`set_state`**(*name*, *backend*, *state*, *socket='/var/run/haproxy.sock'*)

Force a server's administrative state to a new state. This can be useful to disable load balancing and/or any traffic to a server. Setting the state to "ready" puts the server in normal mode, and the command is the equivalent of the "enable server" command. Setting the state to "maint" disables any traffic to the server as well as any health checks. This is the equivalent of the "disable server" command. Setting the mode to "drain" only removes the server from load balancing but still allows it to be checked and to accept new persistent connections. Changes are propagated to tracking servers if any.

name Server name

backend haproxy backend

state A string of the state to set. Must be 'ready', 'drain', or 'maint'

socket haproxy stats socket, default `/var/run/haproxy.sock`

CLI Example:

```
salt '*' haproxy.set_state my_proxy_server my_backend ready
```

`salt.modules.haproxyconn.`**`set_weight`**(*name*, *backend*, *weight=0*, *socket='/var/run/haproxy.sock'*)

Set server weight

name Server name

backend haproxy backend

weight Server Weight

socket haproxy stats socket, default `/var/run/haproxy.sock`

CLI Example:

```
salt '*' haproxy.set_weight web1.example.com www 13
```

`salt.modules.haproxyconn.`**`show_backends`**(*socket='/var/run/haproxy.sock'*)

Show HaProxy Backends

socket haproxy stats socket, default `/var/run/haproxy.sock`

CLI Example:

```
salt '*' haproxy.show_backends
```

`salt.modules.haproxyconn.`**`show_frontends`**(*socket='/var/run/haproxy.sock'*)

Show HaProxy frontends

socket haproxy stats socket, default `/var/run/haproxy.sock`

CLI Example:

```
salt '*' haproxy.show_frontends
```

`salt.modules.haproxyconn.`**`wait_state`**(*backend*, *server*, *value='up'*, *timeout=300*, *socket='/var/run/haproxy.sock'*)

Wait for a specific server state

backend haproxy backend

server targeted server

value state value

timeout timeout before giving up state value, default 5 min

socket haproxy stats socket, default `/var/run/haproxy.sock`

CLI Example:

```
salt '*' haproxy.wait_state mysql server01 up 60
```

21.9.155 salt.modules.hashutil

A collection of hashing and encoding functions `salt.modules.hashutil.`**`base64_b64decode`**`(`*instr*`)`

Decode a base64-encoded string using the "modern" Python interface

New in version 2016.3.0.

CLI Example:

```
salt '*' hashutil.base64_b64decode 'Z2V0IHNhbHRlZA=='
```

`salt.modules.hashutil.`**`base64_b64encode`**`(`*instr*`)`

Encode a string as base64 using the "modern" Python interface.

Among other possible differences, the "modern" encoder does not include newline ('n') characters in the encoded output.

New in version 2016.3.0.

CLI Example:

```
salt '*' hashutil.base64_b64encode 'get salted'
```

`salt.modules.hashutil.`**`base64_decodefile`**`(`*instr, outfile*`)`

Decode a base64-encoded string and write the result to a file

New in version 2016.3.0.

CLI Example:

```
salt '*' hashutil.base64_decodefile instr='Z2V0IHNhbHRlZAo=' outfile='/path/to/
↪binary_file'
```

`salt.modules.hashutil.`**`base64_decodestring`**`(`*instr*`)`

Decode a base64-encoded byte-like object using the "modern" Python interface

New in version 3000.

CLI Example:

```
salt '*' hashutil.base64_decodestring instr='Z2V0IHNhbHRlZAo='
```

`salt.modules.hashutil.`**`base64_encodefile`**`(`*fname*`)`

Read a file from the file system and return as a base64 encoded string

New in version 2016.3.0.

Pillar example:

```
path:
  to:
    data: |
      {{ salt.hashutil.base64_encodefile('/path/to/binary_file') | indent(6) }}
```

The *`file.decode`* state function can be used to decode this data and write it to disk.

CLI Example:

```
salt '*' hashutil.base64_encodefile /path/to/binary_file
```

`salt.modules.hashutil.`**`base64_encodestring`**`(`*instr*`)`

Encode a byte-like object as base64 using the "modern" Python interface.

Among other possible differences, the "modern" encoder includes a newline ('n') character after every 76 characters and always at the end of the encoded byte-like object.

New in version 3000.

CLI Example:

```
salt '*' hashutil.base64_encodestring 'get salted'
```

`salt.modules.hashutil.`**`digest`**`(`*instr*, *checksum='md5'*`)`

Return a checksum digest for a string

instr A string

checksum [`md5`] The hashing algorithm to use to generate checksums. Valid options: md5, sha256, sha512.

CLI Example:

```
salt '*' hashutil.digest 'get salted'
```

`salt.modules.hashutil.`**`digest_file`**`(`*infile*, *checksum='md5'*`)`

Return a checksum digest for a file

infile A file path

checksum [`md5`] The hashing algorithm to use to generate checksums. Wraps the *`hashutil.digest`* execution function.

CLI Example:

```
salt '*' hashutil.digest_file /path/to/file
```

`salt.modules.hashutil.`**`github_signature`**`(`*string*, *shared_secret*, *challenge_hmac*`)`

Verify a challenging hmac signature against a string / shared-secret for github webhooks.

New in version 2017.7.0.

Returns a boolean if the verification succeeded or failed.

CLI Example:

```
salt '*' hashutil.github_signature '{"ref":....} ' 'shared secret'
↪'sha1=bc6550fc290acf5b42283fa8deaf55cea0f8c206'
```

`salt.modules.hashutil.`**`hmac_compute`**`(`*string*, *shared_secret*`)`

New in version 3000.

Compute a HMAC SHA256 digest using a string and secret.

CLI Example:

```
salt '*' hashutil.hmac_compute 'get salted' 'shared secret'
```

`salt.modules.hashutil.`**`hmac_signature`**`(`*string*, *shared_secret*, *challenge_hmac*`)`

Verify a challenging hmac signature against a string / shared-secret

New in version 2014.7.0.

Returns a boolean if the verification succeeded or failed.

CLI Example:

```
salt '*' hashutil.hmac_signature 'get salted' 'shared secret'
↪'eBWf9bstXg+NiP5AOwppB5HMvZiYMPzEM9W5YMm/AmQ='
```

`salt.modules.hashutil.`**`md5_digest`**`(`*instr*`)`

Generate an md5 hash of a given string

New in version 2014.7.0.

CLI Example:

```
salt '*' hashutil.md5_digest 'get salted'
```

`salt.modules.hashutil.`**`sha256_digest`**`(`*instr*`)`

Generate an sha256 hash of a given string

New in version 2014.7.0.

CLI Example:

```
salt '*' hashutil.sha256_digest 'get salted'
```

`salt.modules.hashutil.`**`sha512_digest`**`(`*instr*`)`

Generate an sha512 hash of a given string

New in version 2014.7.0.

CLI Example:

```
salt '*' hashutil.sha512_digest 'get salted'
```

21.9.156 salt.modules.heat module

Module for handling OpenStack Heat calls

New in version 2017.7.0.

depends

- heatclient Python module

configuration This module is not usable until the user, password, tenant, and auth URL are specified either in a pillar or in the minion's config file. For example:

```
keystone.user: admin
keystone.password: verybadpass
keystone.tenant: admin
keystone.insecure: False   #(optional)
keystone.auth_url: 'http://127.0.0.1:5000/v2.0/'
# Optional
keystone.region_name: 'RegionOne'
```

If configuration for multiple OpenStack accounts is required, they can be set up as different configuration profiles: For example:

```
openstack1:
  keystone.user: admin
  keystone.password: verybadpass
  keystone.tenant: admin
  keystone.auth_url: 'http://127.0.0.1:5000/v2.0/'

openstack2:
  keystone.user: admin
```

(continues on next page)

(continued from previous page)

```
keystone.password: verybadpass
keystone.tenant: admin
keystone.auth_url: 'http://127.0.0.2:5000/v2.0/'
```

With this configuration in place, any of the heat functions can make use of a configuration profile by declaring it explicitly. For example:

```
salt '*' heat.flavor_list profile=openstack1
```

`salt.modules.heat.`**`create_stack`**(*name=None, template_file=None, environment=None, parameters=None, poll=0, rollback=False, timeout=60, profile=None*)

Create a stack (heat stack-create)

name Name of the new stack

template_file File of template

environment File of environment

parameters Parameter dict used to create the stack

poll Poll and report events until stack complete

rollback Enable rollback on create failure

timeout Stack creation timeout in minutes

profile Profile to build on

CLI Example:

```
salt '*' heat.create_stack name=mystack \
         template_file=salt://template.yaml \
         environment=salt://environment.yaml \
         parameters="{"image": "Debian 8", "flavor": "m1.small"}" \
         poll=5 rollback=False timeout=60 profile=openstack1
```

New in version 2017.7.5,2018.3.1: The spelling mistake in parameter *enviroment* was corrected to *environment*. The *enviroment* spelling mistake has been removed in Salt 3000.

`salt.modules.heat.`**`delete_stack`**(*name=None, poll=0, timeout=60, profile=None*)

Delete a stack (heat stack-delete)

name Name of the stack

poll Poll and report events until stack complete

timeout Stack creation timeout in minute

profile Profile to use

CLI Examples:

```
salt '*' heat.delete_stack name=mystack poll=5 \
         profile=openstack1
```

`salt.modules.heat.`**`list_stack`**(*profile=None*)

Return a list of available stack (heat stack-list)

profile Profile to use

CLI Example:

```
salt '*' heat.list_stack profile=openstack1
```

`salt.modules.heat.`**`show_stack`**(*name=None*, *profile=None*)
Return details about a specific stack (heat stack-show)

name Name of the stack

profile Profile to use

CLI Example:

```
salt '*' heat.show_stack name=mystack profile=openstack1
```

`salt.modules.heat.`**`template_stack`**(*name=None*, *profile=None*)
Return template a specific stack (heat stack-template)

name Name of the stack

profile Profile to use

CLI Example:

```
salt '*' heat.template_stack name=mystack profile=openstack1
```

`salt.modules.heat.`**`update_stack`**(*name=None*, *template_file=None*, *environment=None*, *parameters=None*, *poll=0*, *rollback=False*, *timeout=60*, *profile=None*)
Update a stack (heat stack-template)

name Name of the stack

template_file File of template

environment File of environment

parameters Parameter dict used to update the stack

poll Poll and report events until stack complete

rollback Enable rollback on update failure

timeout Stack creation timeout in minutes

profile Profile to build on

CLI Example:

```
salt '*' heat.update_stack name=mystack \
         template_file=salt://template.yaml \
         environment=salt://environment.yaml \
         parameters="{"image": "Debian 8", "flavor": "m1.small"}" \
         poll=5 rollback=False timeout=60 profile=openstack1
```

New in version 2017.7.5,2018.3.1: The spelling mistake in parameter *enviroment* was corrected to *environment*. The *enviroment* spelling mistake has been removed in Salt 3000.

21.9.157 salt.modules.hg

Support for the Mercurial SCM

`salt.modules.hg.`**`archive`**(*cwd*, *output*, *rev='tip'*, *fmt=None*, *prefix=None*, *user=None*)
Export a tarball from the repository

cwd The path to the Mercurial repository

output The path to the archive tarball

rev: tip The revision to create an archive from

fmt: None Format of the resulting archive. Mercurial supports: tar, tbz2, tgz, zip, uzip, and files formats.

prefix [None] Prepend <prefix>/ to every filename in the archive

user [None] Run hg as a user other than what the minion runs as

If `prefix` is not specified it defaults to the basename of the repo directory.

CLI Example:

```
salt '*' hg.archive /path/to/repo output=/tmp/archive.tgz fmt=tgz
```

`salt.modules.hg.`**`clone`**(*cwd*, *repository*, *opts=None*, *user=None*, *identity=None*)

Clone a new repository

cwd The path to the Mercurial repository

repository The hg URI of the repository

opts [None] Any additional options to add to the command line

user [None] Run hg as a user other than what the minion runs as

identity [None] Private SSH key on the minion server for authentication (ssh://)

New in version 2015.5.0.

CLI Example:

```
salt '*' hg.clone /path/to/repo https://bitbucket.org/birkenfeld/sphinx
```

`salt.modules.hg.`**`describe`**(*cwd*, *rev='tip'*, *user=None*)

Mimic git describe and return an identifier for the given revision

cwd The path to the Mercurial repository

rev: tip The path to the archive tarball

user [None] Run hg as a user other than what the minion runs as

CLI Example:

```
salt '*' hg.describe /path/to/repo
```

`salt.modules.hg.`**`pull`**(*cwd*, *opts=None*, *user=None*, *identity=None*, *repository=None*)

Perform a pull on the given repository

cwd The path to the Mercurial repository

repository [None] Perform pull from the repository different from .hg/hgrc:[paths]:default

opts [None] Any additional options to add to the command line

user [None] Run hg as a user other than what the minion runs as

identity [None] Private SSH key on the minion server for authentication (ssh://)

New in version 2015.5.0.

CLI Example:

```
salt '*' hg.pull /path/to/repo opts=-u
```

`salt.modules.hg.`**`revision`**(*cwd*, *rev='tip'*, *short=False*, *user=None*)

Returns the long hash of a given identifier (hash, branch, tag, HEAD, etc)

cwd The path to the Mercurial repository

rev: **tip** The revision

short: **False** Return an abbreviated commit hash

user [None] Run hg as a user other than what the minion runs as

CLI Example:

```
salt '*' hg.revision /path/to/repo mybranch
```

`salt.modules.hg.`**`status`**(*cwd*, *opts=None*, *user=None*)

Show changed files of the given repository

cwd The path to the Mercurial repository

opts [None] Any additional options to add to the command line

user [None] Run hg as a user other than what the minion runs as

CLI Example:

```
salt '*' hg.status /path/to/repo
```

`salt.modules.hg.`**`update`**(*cwd*, *rev*, *force=False*, *user=None*)

Update to a given revision

cwd The path to the Mercurial repository

rev The revision to update to

force [False] Force an update

user [None] Run hg as a user other than what the minion runs as

CLI Example:

```
salt devserver1 hg.update /path/to/repo somebranch
```

21.9.158 salt.modules.hosts

Manage the information in the hosts file

`salt.modules.hosts.`**`add_host`**(*ip*, *alias*)

Add a host to an existing entry, if the entry is not in place then create it with the given host

CLI Example:

```
salt '*' hosts.add_host <ip> <alias>
```

`salt.modules.hosts.`**`get_alias`**(*ip*)

Return the list of aliases associated with an ip

Aliases (host names) are returned in the order in which they appear in the hosts file. If there are no aliases associated with the IP, an empty list is returned.

CLI Example:

```
salt '*' hosts.get_alias <ip addr>
```

`salt.modules.hosts.`**`get_ip`**`(`*host*`)`

Return the ip associated with the named host

CLI Example:

```
salt '*' hosts.get_ip <hostname>
```

`salt.modules.hosts.`**`has_pair`**`(`*ip, alias*`)`

Return true if the alias is set

CLI Example:

```
salt '*' hosts.has_pair <ip> <alias>
```

`salt.modules.hosts.`**`list_hosts`**`()`

Return the hosts found in the hosts file in this format:

```
{'<ip addr>': ['alias1', 'alias2', ...]}
```

CLI Example:

```
salt '*' hosts.list_hosts
```

`salt.modules.hosts.`**`rm_host`**`(`*ip, alias*`)`

Remove a host entry from the hosts file

CLI Example:

```
salt '*' hosts.rm_host <ip> <alias>
```

`salt.modules.hosts.`**`set_host`**`(`*ip, alias*`)`

Set the host entry in the hosts file for the given ip, this will overwrite any previous entry for the given ip

Changed in version 2016.3.0: If `alias` does not include any host names (it is the empty string or contains only whitespace), all entries for the given IP address are removed.

CLI Example:

```
salt '*' hosts.set_host <ip> <alias>
```

21.9.159 salt.modules.http

Module for making various web calls. Primarily designed for webhooks and the like, but also useful for basic http testing.

New in version 2015.5.0.

`salt.modules.http.`**`query`**`(`*url, **kwargs*`)`

New in version 2015.5.0.

Query a resource, and decode the return data

Passes through all the parameters described in the *`utils.http.query function`*:

`salt.utils.http.`**`query`**(*url, method='GET', params=None, data=None, data_file=None, header_dict=None, header_list=None, header_file=None, username=None, password=None, auth=None, decode=False, decode_type='auto', status=False, headers=False, text=False, cookies=None, cookie_jar=None, cookie_format='lwp', persist_session=False, session_cookie_jar=None, data_render=False, data_renderer=None, header_render=False, header_renderer=None, template_dict=None, test=False, test_url=None, node='minion', port=80, opts=None, backend=None, ca_bundle=None, verify_ssl=None, cert=None, text_out=None, headers_out=None, decode_out=None, stream=False, streaming_callback=None, header_callback=None, handle=False, agent='Salt/3000.3', hide_fields=None, raise_error=True, formdata=False, formdata_fieldname=None, formdata_filename=None, **kwargs*)

Query a resource, and decode the return data

raise_error [True] If `False`, and if a connection cannot be made, the error will be suppressed and the body of the return will simply be `None`.

CLI Example:

```
salt '*' http.query http://somelink.com/
salt '*' http.query http://somelink.com/ method=POST             params=
↪'key1=val1&key2=val2'
salt '*' http.query http://somelink.com/ method=POST             data='<xml>
↪somecontent</xml>'
```

`salt.modules.http.`**`update_ca_bundle`**(*target=None, source=None, merge_files=None*)

Update the local CA bundle file from a URL

New in version 2015.5.0.

CLI Example:

```
salt '*' http.update_ca_bundle
salt '*' http.update_ca_bundle target=/path/to/cacerts.pem
salt '*' http.update_ca_bundle source=https://example.com/cacerts.pem
```

If the `target` is not specified, it will be pulled from the `ca_cert` configuration variable available to the minion. If it cannot be found there, it will be placed at `<<FILE_ROOTS>>/cacerts.pem`.

If the `source` is not specified, it will be pulled from the `ca_cert_url` configuration variable available to the minion. If it cannot be found, it will be downloaded from the cURL website, using an http (not https) URL. USING THE DEFAULT URL SHOULD BE AVOIDED!

`merge_files` may also be specified, which includes a string or list of strings representing a file or files to be appended to the end of the CA bundle, once it is downloaded.

CLI Example:

```
salt '*' http.update_ca_bundle merge_files=/path/to/mycert.pem
```

`salt.modules.http.`**`wait_for_successful_query`**(*url, wait_for=300, **kwargs*)

Query a resource until a successful response, and decode the return data

CLI Example:

```
salt '*' http.wait_for_successful_query http://somelink.com/ wait_for=160 request_
↪interval=1
```

21.9.160 salt.modules.ifttt

Support for IFTTT

New in version 2015.8.0.

Requires an `api_key` in `/etc/salt/minion`:

```
ifttt:
  secret_key: '280d4699-a817-4719-ba6f-ca56e573e44f'
```

`salt.modules.ifttt.`**`trigger_event`**(*event=None*, ***kwargs*)

Trigger a configured event in IFTTT.

Parameters **`event`** – The name of the event to trigger.

Returns A dictionary with status, text, and error if result was failure.

21.9.161 salt.modules.ilo

Manage HP ILO

depends hponcfg (SmartStart Scripting Toolkit Linux Edition)

`salt.modules.ilo.`**`change_password`**(*username*, *password*)

Reset a users password

CLI Example:

```
salt '*' ilo.change_password damianMyerscough
```

`salt.modules.ilo.`**`change_username`**(*old_username*, *new_username*)

Change a username

CLI Example:

```
salt '*' ilo.change_username damian diana
```

`salt.modules.ilo.`**`configure_network`**(*ip*, *netmask*, *gateway*)

Configure Network Interface

CLI Example:

```
salt '*' ilo.configure_network [IP ADDRESS] [NETMASK] [GATEWAY]
```

`salt.modules.ilo.`**`configure_snmp`**(*community*, *snmp_port=161*, *snmp_trapport=161*)

Configure SNMP

CLI Example:

```
salt '*' ilo.configure_snmp [COMMUNITY STRING] [SNMP PORT] [SNMP TRAP PORT]
```

`salt.modules.ilo.`**`create_user`**(*name*, *password*, **privileges*)

Create user

CLI Example:

```
salt '*' ilo.create_user damian secretagent VIRTUAL_MEDIA_PRIV
```

If no permissions are specify the user will only have a read-only account.

Supported privelges:

- ADMIN_PRIV Enables the user to administer user accounts.
- REMOTE_CONS_PRIV Enables the user to access the Remote Console functionality.
- RESET_SERVER_PRIV Enables the user to remotely manipulate the server power setting.
- VIRTUAL_MEDIA_PRIV Enables the user permission to access the virtual media functionality.
- CONFIG_ILO_PRIV Enables the user to configure iLO settings.

`salt.modules.ilo.`**`delete_ssh_key`**`(`*username*`)`

Delete a users SSH key from the ILO

CLI Example:

```
salt '*' ilo.delete_user_sshkey damian
```

`salt.modules.ilo.`**`delete_user`**`(`*username*`)`

Delete a user

CLI Example:

```
salt '*' ilo.delete_user damian
```

`salt.modules.ilo.`**`disable_dhcp`**`()`

Disable DHCP

CLI Example:

```
salt '*' ilo.disable_dhcp
```

`salt.modules.ilo.`**`disable_ssh`**`()`

Disable the SSH daemon

CLI Example:

```
salt '*' ilo.disable_ssh
```

`salt.modules.ilo.`**`enable_dhcp`**`()`

Enable DHCP

CLI Example:

```
salt '*' ilo.enable_dhcp
```

`salt.modules.ilo.`**`enable_ssh`**`()`

Enable the SSH daemon

CLI Example:

```
salt '*' ilo.enable_ssh
```

`salt.modules.ilo.`**`get_user`**`(`*username*`)`

Returns local user information, excluding the password

CLI Example:

```
salt '*' ilo.get_user damian
```

`salt.modules.ilo.`**`global_settings`**`()`

Show global settings

CLI Example:

```
salt '*' ilo.global_settings
```

`salt.modules.ilo.`**`list_users`**`()`

List all users

CLI Example:

```
salt '*' ilo.list_users
```

`salt.modules.ilo.`**`list_users_info`**`()`

List all users in detail

CLI Example:

```
salt '*' ilo.list_users_info
```

`salt.modules.ilo.`**`network`**`()`

Grab the current network settings

CLI Example:

```
salt '*' ilo.network
```

`salt.modules.ilo.`**`set_http_port`**`(`*port=80*`)`

Configure the port HTTP should listen on

CLI Example:

```
salt '*' ilo.set_http_port 8080
```

`salt.modules.ilo.`**`set_https_port`**`(`*port=443*`)`

Configure the port HTTPS should listen on

CLI Example:

```
salt '*' ilo.set_https_port 4334
```

`salt.modules.ilo.`**`set_ssh_key`**`(`*public_key*`)`

Configure SSH public keys for specific users

CLI Example:

```
salt '*' ilo.set_ssh_key "ssh-dss AAAAB3NzaC1kc3MAAACBA... damian"
```

The SSH public key needs to be DSA and the last argument in the key needs to be the username (case-senstive) of the ILO username.

`salt.modules.ilo.`**`set_ssh_port`**`(`*port=22*`)`

Enable SSH on a user defined port

CLI Example:

```
salt '*' ilo.set_ssh_port 2222
```

21.9.162 salt.modules.icinga2 module

Module to provide icinga2 compatibility to salt.

New in version 2017.7.0.

depends

- icinga2 server

`salt.modules.icinga2.`**`generate_cert`**`(`*domain*`)`

Generate an icinga2 client certificate and key.

Returns:: icinga2 pki new-cert –cn domain.tld –key /etc/icinga2/pki/domain.tld.key –cert /etc/icinga2/pki/domain.tld.crt

CLI Example:

```
salt '*' icinga2.generate_cert domain.tld
```

`salt.modules.icinga2.`**`generate_ticket`**`(`*domain*`)`

Generate and save an icinga2 ticket.

Returns:: icinga2 pki ticket –cn domain.tld

CLI Example:

```
salt '*' icinga2.generate_ticket domain.tld
```

`salt.modules.icinga2.`**`node_setup`**`(`*domain, master, ticket*`)`

Setup the icinga2 node.

Returns:: icinga2 node setup –ticket TICKET_ID –endpoint master.domain.tld –zone domain.tld –master_host master.domain.tld –trustedcert /etc/icinga2/pki/trusted-master.crt

CLI Example:

```
salt '*' icinga2.node_setup domain.tld master.domain.tld TICKET_ID
```

`salt.modules.icinga2.`**`request_cert`**`(`*domain, master, ticket, port*`)`

Request CA cert from master icinga2 node.

Returns:: icinga2 pki request –host master.domain.tld –port 5665 –ticket TICKET_ID –key /etc/icinga2/pki/domain.tld.key –cert /etc/icinga2/pki/domain.tld.crt –trustedcert /etc/icinga2/pki/trusted-master.crt –ca /etc/icinga2/pki/ca.crt

CLI Example:

```
salt '*' icinga2.request_cert domain.tld master.domain.tld TICKET_ID
```

`salt.modules.icinga2.`**`save_cert`**`(`*domain, master*`)`

Save the certificate for master icinga2 node.

Returns:: icinga2 pki save-cert –key /etc/icinga2/pki/domain.tld.key –cert /etc/icinga2/pki/domain.tld.crt –trustedcert /etc/icinga2/pki/trusted-master.crt –host master.domain.tld

CLI Example:

```
salt '*' icinga2.save_cert domain.tld master.domain.tld
```

21.9.163 salt.modules.incron

Work with incron

`salt.modules.incron.`**`list_tab`**`(`*user*`)`

Return the contents of the specified user's incrontab

CLI Example:

```
salt '*' incron.list_tab root
```

`salt.modules.incron.`**`ls`**`(`*user*`)`

This function is an alias of `list_tab`.

> Return the contents of the specified user's incrontab
>
> CLI Example:
>
> ```
> salt '*' incron.list_tab root
> ```

`salt.modules.incron.`**`raw_incron`**`(`*user*`)`

Return the contents of the user's incrontab

CLI Example:

```
salt '*' incron.raw_incron root
```

`salt.modules.incron.`**`raw_system_incron`**`()`

Return the contents of the system wide incrontab

CLI Example:

```
salt '*' incron.raw_system_incron
```

`salt.modules.incron.`**`rm`**`(`*user, path, mask, cmd*`)`

This function is an alias of `rm_job`.

> Remove a incron job for a specified user. If any of the day/time params are specified, the job will only be removed if the specified params match.
>
> CLI Example:
>
> ```
> salt '*' incron.rm_job root /path
> ```

`salt.modules.incron.`**`rm_job`**`(`*user, path, mask, cmd*`)`

Remove a incron job for a specified user. If any of the day/time params are specified, the job will only be removed if the specified params match.

CLI Example:

```
salt '*' incron.rm_job root /path
```

`salt.modules.incron.`**`set_job`**`(`*user, path, mask, cmd*`)`

Sets an incron job up for a specified user.

CLI Example:

```
salt '*' incron.set_job root '/root' 'IN_MODIFY' 'echo "$$ $@ $# $% $&"'
```

`salt.modules.incron.`**`write_incron_file`**`(`*user*, *path*`)`

Writes the contents of a file to a user's incrontab

CLI Example:

```
salt '*' incron.write_incron_file root /tmp/new_incron
```

`salt.modules.incron.`**`write_incron_file_verbose`**`(`*user*, *path*`)`

Writes the contents of a file to a user's incrontab and return error message on error

CLI Example:

```
salt '*' incron.write_incron_file_verbose root /tmp/new_incron
```

21.9.164 salt.modules.influxdbmod

InfluxDB - A distributed time series database

Module to provide InfluxDB compatibility to Salt (compatible with InfluxDB version 0.9+)

depends

- influxdb Python module (>= 3.0.0)

configuration This module accepts connection configuration details either as parameters or as configuration settings in /etc/salt/minion on the relevant minions:

```
influxdb.host: 'localhost'
influxdb.port: 8086
influxdb.user: 'root'
influxdb.password: 'root'
```

This data can also be passed into pillar. Options passed into opts will overwrite options passed into pillar.

Most functions in this module allow you to override or provide some or all of these settings via keyword arguments:

```
salt '*' influxdb.foo_function influxdb_user='influxadmin' influxdb_
↪password='s3cr1t'
```

would override `user` and `password` while still using the defaults for `host` and `port`.

`salt.modules.influxdbmod.`**`alter_retention_policy`**`(`*database*, *name*, *duration*, *replication*, *default=False*, ***client_args*`)`

Modify an existing retention policy.

name Name of the retention policy to modify.

database Name of the database for which the retention policy was defined.

duration New duration of given retention policy.

Durations such as 1h, 90m, 12h, 7d, and 4w, are all supported and mean 1 hour, 90 minutes, 12 hours, 7 day, and 4 weeks, respectively. For infinite retention – meaning the data will never be deleted – use 'INF' for duration. The minimum retention period is 1 hour.

replication New replication of given retention policy.

This determines how many independent copies of each data point are stored in a cluster.

default [False] Whether or not to set the modified policy as default.

CLI Example:

```
salt '*' influxdb.alter_retention_policy metrics default 1d 1
```

`salt.modules.influxdbmod.`**`continuous_query_exists`**`(`*database*, *name*, ***client_args*`)`

Check if continuous query with given name exists on the database.

database Name of the database for which the continuous query was defined.

name Name of the continuous query to check.

CLI Example:

```
salt '*' influxdb.continuous_query_exists metrics default
```

`salt.modules.influxdbmod.`**`create_continuous_query`**`(`*database*, *name*, *query*, *resample_time=None*, *coverage_period=None*, ***client_args*`)`

Create a continuous query.

database Name of the database for which the continuous query will be created on.

name Name of the continuous query to create.

query The continuous query string.

resample_time [None] Duration between continuous query resampling.

coverage_period [None] Duration specifying time period per sample.

CLI Example:

```
salt '*' influxdb.create_continuous_query mydb cq_month 'SELECT mean(*) INTO mydb.
↪a_month.:MEASUREMENT FROM mydb.a_week./.*/ GROUP BY time(5m), *'
```

`salt.modules.influxdbmod.`**`create_db`**`(`*name*, ***client_args*`)`

Create a database.

name Name of the database to create.

CLI Example:

```
salt '*' influxdb.create_db <name>
```

`salt.modules.influxdbmod.`**`create_retention_policy`**`(`*database*, *name*, *duration*, *replication*, *default=False*, ***client_args*`)`

Create a retention policy.

database Name of the database for which the retention policy will be created.

name Name of the new retention policy.

duration Duration of the new retention policy.

> Durations such as 1h, 90m, 12h, 7d, and 4w, are all supported and mean 1 hour, 90 minutes, 12 hours, 7 day, and 4 weeks, respectively. For infinite retention – meaning the data will never be deleted – use 'INF' for duration. The minimum retention period is 1 hour.

replication Replication factor of the retention policy.

> This determines how many independent copies of each data point are stored in a cluster.

default [False] Whether or not the policy as default will be set as default.

CLI Example:

```
salt '*' influxdb.create_retention_policy metrics default 1d 1
```

`salt.modules.influxdbmod.`**`create_user`**`(`*name*, *passwd*, *admin=False*, ***client_args*`)`
Create a user.

name Name of the user to create.

passwd Password of the new user.

admin [False] Whether the user should have cluster administration privileges or not.

CLI Example:

```
salt '*' influxdb.create_user <name> <password>
salt '*' influxdb.create_user <name> <password> admin=True
```

`salt.modules.influxdbmod.`**`db_exists`**`(`*name*, ***client_args*`)`
Checks if a database exists in InfluxDB.

name Name of the database to check.

CLI Example:

```
salt '*' influxdb.db_exists <name>
```

`salt.modules.influxdbmod.`**`drop_continuous_query`**`(`*database*, *name*, ***client_args*`)`
Drop a continuous query.

database Name of the database for which the continuous query will be drop from.

name Name of the continuous query to drop.

CLI Example:

```
salt '*' influxdb.drop_continuous_query mydb my_cq
```

`salt.modules.influxdbmod.`**`drop_db`**`(`*name*, ***client_args*`)`
Drop a database.

name Name of the database to drop.

CLI Example:

```
salt '*' influxdb.drop_db <name>
```

`salt.modules.influxdbmod.`**`drop_retention_policy`**`(`*database*, *name*, ***client_args*`)`
Drop a retention policy.

database Name of the database for which the retention policy will be dropped.

name Name of the retention policy to drop.

CLI Example:

```
salt '*' influxdb.drop_retention_policy mydb mypr
```

`salt.modules.influxdbmod.`**`get_continuous_query`**`(`*database*, *name*, ***client_args*`)`
Get an existing continuous query.

database Name of the database for which the continuous query was defined.

name Name of the continuous query to get.

CLI Example:

```
salt '*' influxdb.get_continuous_query mydb cq_month
```

`salt.modules.influxdbmod.`**`get_retention_policy`**`(`*database*, *name*, ***client_args*`)`

Get an existing retention policy.

database Name of the database for which the retention policy was defined.

name Name of the retention policy.

CLI Example:

```
salt '*' influxdb.get_retention_policy metrics default
```

`salt.modules.influxdbmod.`**`grant_admin_privileges`**`(`*name*, ***client_args*`)`

Grant cluster administration privileges to a user.

name Name of the user to whom admin privileges will be granted.

CLI Example:

```
salt '*' influxdb.grant_admin_privileges <name>
```

`salt.modules.influxdbmod.`**`grant_privilege`**`(`*database*, *privilege*, *username*, ***client_args*`)`

Grant a privilege on a database to a user.

database Name of the database to grant the privilege on.

privilege Privilege to grant. Can be one of 'read', 'write' or 'all'.

username Name of the user to grant the privilege to.

`salt.modules.influxdbmod.`**`list_dbs`**`(`***client_args*`)`

List all InfluxDB databases.

CLI Example:

```
salt '*' influxdb.list_dbs
```

`salt.modules.influxdbmod.`**`list_privileges`**`(`*name*, ***client_args*`)`

List privileges from a user.

name Name of the user from whom privileges will be listed.

CLI Example:

```
salt '*' influxdb.list_privileges <name>
```

`salt.modules.influxdbmod.`**`list_users`**`(`***client_args*`)`

List all users.

CLI Example:

```
salt '*' influxdb.list_users
```

`salt.modules.influxdbmod.`**`query`**`(`*database*, *query*, ***client_args*`)`

Execute a query.

database Name of the database to query on.

query InfluxQL query string.

`salt.modules.influxdbmod.`**`remove_user`**`(`*name*, ***client_args*`)`

Remove a user.

name Name of the user to remove

CLI Example:

```
salt '*' influxdb.remove_user <name>
```

`salt.modules.influxdbmod.`**`retention_policy_exists`**`(`*database*, *name*, ***client_args*`)`

Check if retention policy with given name exists.

database Name of the database for which the retention policy was defined.

name Name of the retention policy to check.

CLI Example:

```
salt '*' influxdb.retention_policy_exists metrics default
```

`salt.modules.influxdbmod.`**`revoke_admin_privileges`**`(`*name*, ***client_args*`)`

Revoke cluster administration privileges from a user.

name Name of the user from whom admin privileges will be revoked.

CLI Example:

```
salt '*' influxdb.revoke_admin_privileges <name>
```

`salt.modules.influxdbmod.`**`revoke_privilege`**`(`*database*, *privilege*, *username*, ***client_args*`)`

Revoke a privilege on a database from a user.

database Name of the database to grant the privilege on.

privilege Privilege to grant. Can be one of 'read', 'write' or 'all'.

username Name of the user to grant the privilege to.

`salt.modules.influxdbmod.`**`set_user_password`**`(`*name*, *passwd*, ***client_args*`)`

Change password of a user.

name Name of the user for whom to set the password.

passwd New password of the user.

CLI Example:

```
salt '*' influxdb.set_user_password <name> <password>
```

`salt.modules.influxdbmod.`**`user_exists`**`(`*name*, ***client_args*`)`

Check if a user exists.

name Name of the user to check.

CLI Example:

```
salt '*' influxdb.user_exists <name>
```

`salt.modules.influxdbmod.`**`user_info`**`(`*name*, ***client_args*`)`

Get information about given user.

name Name of the user for which to get information.

CLI Example:

```
salt '*' influxdb.user_info <name>
```

21.9.165 salt.modules.influxdb08mod

InfluxDB - A distributed time series database

Module to provide InfluxDB compatibility to Salt (compatible with InfluxDB version 0.5-0.8)

New in version 2014.7.0.

depends

- influxdb Python module (>= 1.0.0)

configuration This module accepts connection configuration details either as parameters or as configuration settings in /etc/salt/minion on the relevant minions:

```
influxdb08.host: 'localhost'
influxdb08.port: 8086
influxdb08.user: 'root'
influxdb08.password: 'root'
```

This data can also be passed into pillar. Options passed into opts will overwrite options passed into pillar.

`salt.modules.influxdb08mod.`**`db_create`**(*name, user=None, password=None, host=None, port=None*)

Create a database

name Database name to create

user The user to connect as

password The password of the user

host The host to connect to

port The port to connect to

CLI Example:

```
salt '*' influxdb08.db_create <name>
salt '*' influxdb08.db_create <name> <user> <password> <host> <port>
```

`salt.modules.influxdb08mod.`**`db_exists`**(*name, user=None, password=None, host=None, port=None*)

Checks if a database exists in Influxdb

name Database name to create

user The user to connect as

password The password of the user

host The host to connect to

port The port to connect to

CLI Example:

```
salt '*' influxdb08.db_exists <name>
salt '*' influxdb08.db_exists <name> <user> <password> <host> <port>
```

`salt.modules.influxdb08mod.`**`db_list`**(*user=None, password=None, host=None, port=None*)

List all InfluxDB databases

user The user to connect as

password The password of the user

host The host to connect to

port The port to connect to

CLI Example:

```
salt '*' influxdb08.db_list
salt '*' influxdb08.db_list <user> <password> <host> <port>
```

`salt.modules.influxdb08mod.`**`db_remove`**(*name, user=None, password=None, host=None, port=None*)

Remove a database

name Database name to remove

user The user to connect as

password The password of the user

host The host to connect to

port The port to connect to

CLI Example:

```
salt '*' influxdb08.db_remove <name>
salt '*' influxdb08.db_remove <name> <user> <password> <host> <port>
```

`salt.modules.influxdb08mod.`**`login_test`**(*name, password, database=None, host=None, port=None*)

Checks if a credential pair can log in at all.

If a database is specified: it will check for database user existence. If a database is not specified: it will check for cluster admin existence.

name The user to connect as

password The password of the user

database The database to try to log in to

host The host to connect to

port The port to connect to

CLI Example:

```
salt '*' influxdb08.login_test <name>
salt '*' influxdb08.login_test <name> <database>
salt '*' influxdb08.login_test <name> <database> <user> <password> <host> <port>
```

`salt.modules.influxdb08mod.`**`query`**(*database, query, time_precision='s', chunked=False, user=None, password=None, host=None, port=None*)

Querying data

database The database to query

query Query to be executed

time_precision Time precision to use ('s', 'm', or 'u')

chunked Whether is chunked or not

user The user to connect as

password The password of the user

host The host to connect to

port The port to connect to

CLI Example:

```
salt '*' influxdb08.query <database> <query>
salt '*' influxdb08.query <database> <query> <time_precision> <chunked> <user>
↪<password> <host> <port>
```

`salt.modules.influxdb08mod.`**`retention_policy_add`**`(`*database, name, duration, replication, default=False, user=None, password=None, host=None, port=None*`)`

Add a retention policy.

database The database to operate on.

name Name of the policy to modify.

duration How long InfluxDB keeps the data.

replication How many copies of the data are stored in the cluster.

default Whether this policy should be the default or not. Default is False.

CLI Example:

```
salt '*' influxdb.retention_policy_add metrics default 1d 1
```

`salt.modules.influxdb08mod.`**`retention_policy_alter`**`(`*database, name, duration, replication, default=False, user=None, password=None, host=None, port=None*`)`

Modify an existing retention policy.

database The database to operate on.

name Name of the policy to modify.

duration How long InfluxDB keeps the data.

replication How many copies of the data are stored in the cluster.

default Whether this policy should be the default or not. Default is False.

CLI Example:

```
salt '*' influxdb08.retention_policy_modify metrics default 1d 1
```

`salt.modules.influxdb08mod.`**`retention_policy_exists`**`(`*database, name, user=None, password=None, host=None, port=None*`)`

Check if a retention policy exists.

database The database to operate on.

name Name of the policy to modify.

CLI Example:

```
salt '*' influxdb08.retention_policy_exists metrics default
```

salt.modules.influxdb08mod.**retention_policy_get**(*database, name, user=None, password=None, host=None, port=None*)

Get an existing retention policy.

database The database to operate on.

name Name of the policy to modify.

CLI Example:

```
salt '*' influxdb08.retention_policy_get metrics default
```

salt.modules.influxdb08mod.**user_chpass**(*name, passwd, database=None, user=None, password=None, host=None, port=None*)

Change password for a cluster admin or a database user.

If a database is specified: it will update database user password. If a database is not specified: it will update cluster admin password.

name User name for whom to change the password

passwd New password

database The database on which to operate

user The user to connect as

password The password of the user

host The host to connect to

port The port to connect to

CLI Example:

```
salt '*' influxdb08.user_chpass <name> <passwd>
salt '*' influxdb08.user_chpass <name> <passwd> <database>
salt '*' influxdb08.user_chpass <name> <passwd> <database> <user> <password>
↪<host> <port>
```

salt.modules.influxdb08mod.**user_create**(*name, passwd, database=None, user=None, password=None, host=None, port=None*)

Create a cluster admin or a database user.

If a database is specified: it will create database user. If a database is not specified: it will create a cluster admin.

name User name for the new user to create

passwd Password for the new user to create

database The database to create the user in

user The user to connect as

password The password of the user

host The host to connect to

port The port to connect to

CLI Example:

```
salt '*' influxdb08.user_create <name> <passwd>
salt '*' influxdb08.user_create <name> <passwd> <database>
salt '*' influxdb08.user_create <name> <passwd> <database> <user> <password>
↪<host> <port>
```

`salt.modules.influxdb08mod.`**`user_exists`**(*name*, *database=None*, *user=None*, *password=None*, *host=None*, *port=None*)

Checks if a cluster admin or database user exists.

If a database is specified: it will check for database user existence. If a database is not specified: it will check for cluster admin existence.

name User name

database The database to check for the user to exist

user The user to connect as

password The password of the user

host The host to connect to

port The port to connect to

CLI Example:

```
salt '*' influxdb08.user_exists <name>
salt '*' influxdb08.user_exists <name> <database>
salt '*' influxdb08.user_exists <name> <database> <user> <password> <host> <port>
```

`salt.modules.influxdb08mod.`**`user_list`**(*database=None*, *user=None*, *password=None*, *host=None*, *port=None*)

List cluster admins or database users.

If a database is specified: it will return database users list. If a database is not specified: it will return cluster admins list.

database The database to list the users from

user The user to connect as

password The password of the user

host The host to connect to

port The port to connect to

CLI Example:

```
salt '*' influxdb08.user_list
salt '*' influxdb08.user_list <database>
salt '*' influxdb08.user_list <database> <user> <password> <host> <port>
```

`salt.modules.influxdb08mod.`**`user_remove`**(*name*, *database=None*, *user=None*, *password=None*, *host=None*, *port=None*)

Remove a cluster admin or a database user.

If a database is specified: it will remove the database user. If a database is not specified: it will remove the cluster admin.

name User name to remove

database The database to remove the user from

user User name for the new user to delete

user The user to connect as

password The password of the user

host The host to connect to

port The port to connect to

CLI Example:

```
salt '*' influxdb08.user_remove <name>
salt '*' influxdb08.user_remove <name> <database>
salt '*' influxdb08.user_remove <name> <database> <user> <password> <host> <port>
```

21.9.166 salt.modules.infoblox

This module have been tested on infoblox API v1.2.1, other versions of the API are likly workable.

depends libinfoblox, https://github.com/steverweber/libinfoblox

libinfoblox can be installed using *pip install libinfoblox*

API documents can be found on your infoblox server at:

https://INFOBLOX/wapidoc

configuration The following configuration defaults can be defined (pillar or config files '/etc/salt/master.d/infoblox.conf'):

```
infoblox.config:
    api_sslverify: True
    api_url: 'https://INFOBLOX/wapi/v1.2.1'
    api_user: 'username'
    api_key: 'password'
```

Many of the functions accept *api_opts* to override the API config.

```
salt-call infoblox.get_host name=my.host.com              api_url:
↪'https://INFOBLOX/wapi/v1.2.1'             api_user=admin              
↪api_key=passs
```

salt.modules.infoblox.**create_a**(*data*, ***api_opts*)

Create A record.

This is a helper function to *create_object*. See your infoblox API for full *data* format.

CLI Example:

```
salt-call infoblox.create_a                          data =
            name: 'fastlinux.math.example.ca'
            ipv4addr: '127.0.0.1'
            view: External
```

salt.modules.infoblox.**create_cname**(*data*, ***api_opts*)

Create a cname record.

CLI Example:

```
salt-call infoblox.create_cname data={             "comment": "cname to example⤦
↪server",              "name": "example.example.com",              "zone":
↪"example.com",              "view": "Internal",              "canonical":
↪"example-ha-0.example.com"          }
```

`salt.modules.infoblox.create_host(data, **api_opts)`

Add host record

Avoid race conditions, use func:nextavailableip for ipv[4,6]addrs:

- func:nextavailableip:network/ZG54dfgsrDFEFfsfsLzA:10.0.0.0/8/default
- func:nextavailableip:10.0.0.0/8
- func:nextavailableip:10.0.0.0/8,external
- func:nextavailableip:10.0.0.3-10.0.0.10

See your infoblox API for full *data* format.

CLI Example:

```
salt-call infoblox.create_host             data =
        {'name': 'hostname.example.ca',
        'aliases': ['hostname.math.example.ca'],
    'extattrs': [{'Business Contact': {'value': 'example@example.ca'}},
        {'Pol8 Classification': {'value': 'Restricted'}},
        {'Primary OU': {'value': 'CS'}},
        {'Technical Contact': {'value': 'example@example.ca'}}],
    'ipv4addrs': [{'configure_for_dhcp': True,
        'ipv4addr': 'func:nextavailableip:129.97.139.0/24',
        'mac': '00:50:56:84:6e:ae'}],
    'ipv6addrs': [], }
```

`salt.modules.infoblox.create_ipv4_range(data, **api_opts)`

Create a ipv4 range

This is a helper function to *create_object* See your infoblox API for full *data* format.

CLI Example:

```
salt-call infoblox.create_ipv4_range data={
    start_addr: '129.97.150.160',
    end_addr: '129.97.150.170'}
```

`salt.modules.infoblox.create_object(object_type, data, **api_opts)`

Create raw infoblox object. This is a low level api call.

CLI Example:

```
salt-call infoblox.update_object object_type=record:host  data={}
```

`salt.modules.infoblox.delete_a(name=None, ipv4addr=None, allow_array=False, **api_opts)`

Delete A record

If the A record is used as a round robin you can set `allow_array=True` to delete all records for the hostname.

CLI Examples:

```
salt-call infoblox.delete_a name=abc.example.com
salt-call infoblox.delete_a ipv4addr=192.168.3.5
salt-call infoblox.delete_a name=acname.example.com allow_array=True
```

`salt.modules.infoblox.`**`delete_cname`**(*name=None*, *canonical=None*, ***api_opts*)

Delete CNAME. This is a helper call to delete_object.

If record is not found, return True

CLI Examples:

```
salt-call infoblox.delete_cname name=example.example.com
salt-call infoblox.delete_cname canonical=example-ha-0.example.com
```

`salt.modules.infoblox.`**`delete_host`**(*name=None*, *mac=None*, *ipv4addr=None*, ***api_opts*)

Delete host

CLI Example:

```
salt-call infoblox.delete_host name=example.domain.com
salt-call infoblox.delete_host ipv4addr=123.123.122.12
salt-call infoblox.delete_host ipv4addr=123.123.122.12 mac=00:50:56:84:6e:ae
```

`salt.modules.infoblox.`**`delete_ipv4_range`**(*start_addr=None*, *end_addr=None*, ***api_opts*)

Delete ip range.

CLI Example:

```
salt-call infoblox.delete_ipv4_range start_addr=123.123.122.12
```

`salt.modules.infoblox.`**`delete_object`**(*objref*, ***api_opts*)

Delete infoblox object. This is a low level api call.

CLI Example:

```
salt-call infoblox.delete_object objref=[ref_of_object]
```

`salt.modules.infoblox.`**`diff_objects`**(*obja*, *objb*)

Diff two complex infoblox objects. This is used from salt states to detect changes in objects.

Using `func:nextavailableip` will not cause a diff if the ipaddress is in range

`salt.modules.infoblox.`**`get_a`**(*name=None*, *ipv4addr=None*, *allow_array=True*, ***api_opts*)

Get A record

CLI Examples:

```
salt-call infoblox.get_a name=abc.example.com
salt-call infoblox.get_a ipv4addr=192.168.3.5
```

`salt.modules.infoblox.`**`get_cname`**(*name=None*, *canonical=None*, *return_fields=None*, ***api_opts*)

Get CNAME information.

CLI Examples:

```
salt-call infoblox.get_cname name=example.example.com
salt-call infoblox.get_cname canonical=example-ha-0.example.com
```

`salt.modules.infoblox.`**`get_host`**(*name=None*, *ipv4addr=None*, *mac=None*, *return_fields=None*, ***api_opts*)

Get host information

CLI Examples:

```
salt-call infoblox.get_host hostname.domain.ca
salt-call infoblox.get_host ipv4addr=123.123.122.12
salt-call infoblox.get_host mac=00:50:56:84:6e:ae
```

salt.modules.infoblox.**get_host_advanced**(*name=None*, *ipv4addr=None*, *mac=None*, ***api_opts*)

Get all host information

CLI Example:

```
salt-call infoblox.get_host_advanced hostname.domain.ca
```

salt.modules.infoblox.**get_host_domainname**(*name*, *domains=None*, ***api_opts*)

Get host domain name

If no domains are passed, the hostname is checked for a zone in infoblox, if no zone split on first dot.

If domains are provided, the best match out of the list is returned.

If none are found the return is None

dots at end of names are ignored.

CLI Example:

```
salt-call uwl.get_host_domainname name=localhost.t.domain.com          ⊠
↪domains=['domain.com', 't.domain.com.']

# returns: t.domain.com
```

salt.modules.infoblox.**get_host_hostname**(*name*, *domains=None*, ***api_opts*)

Get hostname

If no domains are passed, the hostname is checked for a zone in infoblox, if no zone split on first dot.

If domains are provided, the best match out of the list is truncated from the fqdn leaving the hostname.

If no matching domains are found the fqdn is returned.

dots at end of names are ignored.

CLI Examples:

```
salt-call infoblox.get_host_hostname fqdn=localhost.xxx.t.domain.com      ⊠
↪domains="['domain.com', 't.domain.com']"
#returns: localhost.xxx

salt-call infoblox.get_host_hostname fqdn=localhost.xxx.t.domain.com
#returns: localhost
```

salt.modules.infoblox.**get_host_ipv4**(*name=None*, *mac=None*, *allow_array=False*, ***api_opts*)

Get ipv4 address from host record.

Use *allow_array* to return possible multiple values.

CLI Examples:

```
salt-call infoblox.get_host_ipv4 host=localhost.domain.com
salt-call infoblox.get_host_ipv4 mac=00:50:56:84:6e:ae
```

`salt.modules.infoblox.`**`get_host_ipv4addr_info`**(*ipv4addr=None, mac=None, discovered_data=None, return_fields=None, **api_opts*)

Get host ipv4addr information

CLI Examples:

```
salt-call infoblox.get_ipv4addr ipv4addr=123.123.122.12
salt-call infoblox.get_ipv4addr mac=00:50:56:84:6e:ae
salt-call infoblox.get_ipv4addr mac=00:50:56:84:6e:ae return_fields=host return_
↪fields='mac,host,configure_for_dhcp,ipv4addr'
```

`salt.modules.infoblox.`**`get_host_ipv6addr_info`**(*ipv6addr=None, mac=None, discovered_data=None, return_fields=None, **api_opts*)

Get host ipv6addr information

CLI Example:

```
salt-call infoblox.get_host_ipv6addr_info
↪ipv6addr=2001:db8:85a3:8d3:1349:8a2e:370:7348
```

`salt.modules.infoblox.`**`get_host_mac`**(*name=None, allow_array=False, **api_opts*)

Get mac address from host record.

Use *allow_array* to return possible multiple values.

CLI Example:

```
salt-call infoblox.get_host_mac host=localhost.domain.com
```

`salt.modules.infoblox.`**`get_ipv4_range`**(*start_addr=None, end_addr=None, return_fields=None, **api_opts*)

Get ip range

CLI Example:

```
salt-call infoblox.get_ipv4_range start_addr=123.123.122.12
```

`salt.modules.infoblox.`**`get_network`**(*ipv4addr=None, network=None, return_fields=None, **api_opts*)

Get list of all networks. This is helpful when looking up subnets to use with func:nextavailableip

This call is offen slow and not cached!

some return_fields comment,network,network_view,ddns_domainname,disable,enable_ddns

CLI Example:

```
salt-call infoblox.get_network
```

`salt.modules.infoblox.`**`get_object`**(*objref, data=None, return_fields=None, max_results=None, ensure_none_or_one_result=False, **api_opts*)

Get raw infoblox object. This is a low level api call.

CLI Example:

```
salt-call infoblox.get_object objref=[_ref of object]
```

`salt.modules.infoblox.`**`is_ipaddr_in_ipfunc_range`**(*ipaddr, ipfunc*)

Return true if the ipaddress is in the range of the nextavailableip function

CLI Example:

```
salt-call infoblox.is_ipaddr_in_ipfunc_range          ipaddr="10.0.2.2" ipfunc=
↪"func:nextavailableip:10.0.0.0/8"
```

`salt.modules.infoblox.`**`update_cname`**`(`*name*, *data*, ***api_opts*`)`

Update CNAME. This is a helper call to update_object.

Find a CNAME `_ref` then call update_object with the record data.

CLI Example:

```
salt-call infoblox.update_cname name=example.example.com data="{
        'canonical':'example-ha-0.example.com',
        'use_ttl':true,
        'ttl':200,
        'comment':'Salt managed CNAME'}"
```

`salt.modules.infoblox.`**`update_host`**`(`*name*, *data*, ***api_opts*`)`

Update host record. This is a helper call to update_object.

Find a hosts `_ref` then call update_object with the record data.

CLI Example:

```
salt-call infoblox.update_host name=fqdn data={}
```

`salt.modules.infoblox.`**`update_object`**`(`*objref*, *data*, ***api_opts*`)`

Update raw infoblox object. This is a low level api call.

CLI Example:

```
salt-call infoblox.update_object objref=[ref_of_object] data={}
```

21.9.167 salt.modules.ini_manage

Edit ini files

maintainer <akilesh1597@gmail.com>

maturity new

depends re

platform all

(for example /etc/sysctl.conf)

`salt.modules.ini_manage.`**`get_ini`**`(`*file_name*, *separator='='*`)`

Retrieve whole structure from an ini file and return it as dictionary.

API Example:

```
import salt
sc = salt.client.get_local_client()
sc.cmd('target', 'ini.get_ini',
       [path_to_ini_file])
```

CLI Example:

```
salt '*' ini.get_ini /path/to/ini
```

salt.modules.ini_manage.**get_option**(*file_name*, *section*, *option*, *separator='='*)

Get value of a key from a section in an ini file. Returns `None` if no matching key was found.

API Example:

```
import salt
sc = salt.client.get_local_client()
sc.cmd('target', 'ini.get_option',
       [path_to_ini_file, section_name, option])
```

CLI Example:

```
salt '*' ini.get_option /path/to/ini section_name option_name
```

salt.modules.ini_manage.**get_section**(*file_name*, *section*, *separator='='*)

Retrieve a section from an ini file. Returns the section as dictionary. If the section is not found, an empty dictionary is returned.

API Example:

```
import salt
sc = salt.client.get_local_client()
sc.cmd('target', 'ini.get_section',
       [path_to_ini_file, section_name])
```

CLI Example:

```
salt '*' ini.get_section /path/to/ini section_name
```

salt.modules.ini_manage.**remove_option**(*file_name*, *section*, *option*, *separator='='*)

Remove a key/value pair from a section in an ini file. Returns the value of the removed key, or `None` if nothing was removed.

API Example:

```
import salt
sc = salt.client.get_local_client()
sc.cmd('target', 'ini.remove_option',
       [path_to_ini_file, section_name, option])
```

CLI Example:

```
salt '*' ini.remove_option /path/to/ini section_name option_name
```

salt.modules.ini_manage.**remove_section**(*file_name*, *section*, *separator='='*)

Remove a section in an ini file. Returns the removed section as dictionary, or `None` if nothing was removed.

API Example:

```
import salt
sc = salt.client.get_local_client()
sc.cmd('target', 'ini.remove_section',
       [path_to_ini_file, section_name])
```

CLI Example:

```
salt '*' ini.remove_section /path/to/ini section_name
```

`salt.modules.ini_manage.`**`set_option`**(*file_name, sections=None, separator='='*)

Edit an ini file, replacing one or more sections. Returns a dictionary containing the changes made.

file_name path of ini_file

sections [None] A dictionary representing the sections to be edited ini file The keys are the section names and the values are the dictionary containing the options If the ini file does not contain sections the keys and values represent the options

separator [=] A character used to separate keys and values. Standard ini files use the "=" character.

New in version 2016.11.0.

API Example:

```
import salt
sc = salt.client.get_local_client()
sc.cmd('target', 'ini.set_option',
       ['path_to_ini_file', '{"section_to_change": {"key": "value"}}'])
```

CLI Example:

```
salt '*' ini.set_option /path/to/ini '{section_foo: {key: value}}'
```

21.9.168 salt.modules.inspectlib package

Submodules

salt.modules.inspectlib.collector module

`class` `salt.modules.inspectlib.collector.`**`Inspector`**(*cachedir=None, piddir=None, pidfilename=None*)

`DEFAULT_MINION_CONFIG_PATH = '/etc/salt/minion'`

`IGNORE_FS_TYPES = ['autofs', 'cifs', 'nfs', 'nfs4']`

`IGNORE_MOUNTS = ['proc', 'sysfs', 'devtmpfs', 'tmpfs', 'fuse.gvfs-fuse-daemon']`

`IGNORE_PATHS = ['/tmp', '/var/tmp', '/lost+found', '/var/run', '/var/lib/rpm', '/.snap`

`MODE = ['configuration', 'payload', 'all']`

`build`(*format='qcow2', path='/tmp'*)

Build an image using Kiwi.

Parameters

- **`format`** –
- **`path`** –

Returns

`create_snapshot`()

Open new snapshot.

Returns

export(*description*, *local=False*, *path='/tmp'*, *format='qcow2'*)
Export description for Kiwi.

Parameters

- **local** –
- **path** –

Returns

request_snapshot(*mode*, *priority=19*, ***kwargs*)
Take a snapshot of the system.

reuse_snapshot()
Open an existing, latest snapshot.

Returns

snapshot(*mode*)
Take a snapshot of the system.

salt.modules.inspectlib.collector.**is_alive**(*pidfile*)
Check if PID is still alive.

salt.modules.inspectlib.collector.**main**(*dbfile*, *pidfile*, *mode*)
Main analyzer routine.

salt.modules.inspectlib.dbhandle module

class salt.modules.inspectlib.dbhandle.**DBHandle**(*path*)

class salt.modules.inspectlib.dbhandle.**DBHandleBase**(*path*)
Handle for the *volatile* database, which serves the purpose of caching the inspected data. This database can be destroyed or corrupted, so it should be simply re-created from scratch.

close()
Close the database connection.

flush(*table*)
Flush the table.

open(*new=False*)
Init the database, if required.

purge()
Purge whole database.

salt.modules.inspectlib.exceptions module

exception salt.modules.inspectlib.exceptions.**InspectorKiwiProcessorException**
Kiwi builder/exporter exception.

exception salt.modules.inspectlib.exceptions.**InspectorQueryException**
Exception that is only for the inspector query.

exception salt.modules.inspectlib.exceptions.**InspectorSnapshotException**
Snapshot exception.

exception salt.modules.inspectlib.exceptions.**SIException**
System information exception.

salt.modules.inspectlib.query module

class `salt.modules.inspectlib.query.`**`Query`**(*scope*, *cachedir=None*)

Query the system. This class is actually puts all Salt features together, so there would be no need to pick it from various places.

`SCOPES = ['changes', 'configuration', 'identity', 'system', 'software', 'services', 'p`

class `salt.modules.inspectlib.query.`**`SysInfo`**(*systype*)

System information.

Module contents

class `salt.modules.inspectlib.`**`EnvLoader`**(*cachedir=None*, *piddir=None*, *pidfilename=None*)

Load environment.

`DB_FILE = '_minion_collector.db'`

`DEFAULT_CACHE_PATH = '/var/cache/salt'`

`DEFAULT_PID_PATH = '/var/run'`

`PID_FILE = '_minion_collector.pid'`

21.9.169 salt.modules.inspectlib.entities module

class `salt.modules.inspectlib.entities.`**`AllowedDir`**

Allowed directories

class `salt.modules.inspectlib.entities.`**`IgnoredDir`**

Ignored directories

class `salt.modules.inspectlib.entities.`**`Package`**

Package.

class `salt.modules.inspectlib.entities.`**`PackageCfgFile`**

Config file, belongs to the package

class `salt.modules.inspectlib.entities.`**`PayloadFile`**

Payload file.

21.9.170 salt.modules.inspectlib.fsdb module

codeauthor Bo Maryniuk <bo@suse.de>

class `salt.modules.inspectlib.fsdb.`**`CsvDB`**(*path*)

File-based CSV database. This database is in-memory operating relatively small plain text csv files.

`close`()

Close the database.

Returns

`create_table_from_object`(*obj*)

Create a table from the object. NOTE: This method doesn't stores anything.

Parameters **`obj`** –

Returns

delete(*obj*, *matches=None*, *mt=None*, *lt=None*, *eq=None*)
Delete object from the database.

Parameters

- **obj** –
- **matches** –
- **mt** –
- **lt** –
- **eq** –

Returns

flush(*table*)
Flush table.

Parameters **table** –

Returns

get(*obj*, *matches=None*, *mt=None*, *lt=None*, *eq=None*)
Get objects from the table.

Parameters

- **table_name** –
- **matches** – Regexp.
- **mt** – More than.
- **lt** – Less than.
- **eq** – Equals.

Returns

is_closed()
Return if the database is closed.

Returns

list()
List all the databases on the given path.

Returns

list_tables()
Load existing tables and their descriptions.

Returns

new()
Create a new database and opens it.

Returns

open(*dbname=None*)
Open database from the path with the name or latest. If there are no yet databases, create a new implicitly.

Returns

purge(*dbid*)
Purge the database.

Parameters **dbid** –

Returns

store(*obj*, *distinct=False*)

Store an object in the table.

Parameters

- **obj** – An object to store
- **distinct** – Store object only if there is none identical of such. If at least one field is different, store it.

Returns

update(*obj*, *matches=None*, *mt=None*, *lt=None*, *eq=None*)

Update object(s) in the database.

Parameters

- **obj** –
- **matches** –
- **mt** –
- **lt** –
- **eq** –

Returns

class `salt.modules.inspectlib.fsdb.`**CsvDBEntity**

Serializable object for the table.

21.9.171 salt.modules.inspectlib.kiwiproc module

class `salt.modules.inspectlib.kiwiproc.`**KiwiExporter**(*grains*, *format*)

Exports system description as Kiwi configuration.

export(*name*)

Export to the Kiwi config.xml as text.

Returns

load(***descr*)

Load data by keys.

Parameters **data** –

Returns

21.9.172 salt.modules.inspector module

Module for full system inspection.

`salt.modules.inspector.`**build**(*format='qcow2'*, *path='/tmp/'*)

Build an image from a current system description. The image is a system image can be output in bootable ISO or QCOW2 formats.

Node uses the image building library Kiwi to perform the actual build.

Parameters:

- **format**: Specifies output format: "qcow2" or "iso. Default: *qcow2*.
- **path**: Specifies output path where to store built image. Default: */tmp*.

CLI Example:

```
salt myminion inspector.build
salt myminion inspector.build format=iso path=/opt/builds/
```

`salt.modules.inspector.`**`delete`**(*all=False*, **databases*)

Remove description snapshots from the system.

::parameter: all. Default: False. Remove all snapshots, if set to True.

CLI example:

```
salt myminion inspector.delete <ID> <ID1> <ID2>..
salt myminion inspector.delete all=True
```

`salt.modules.inspector.`**`export`**(*local=False*, *path='/tmp'*, *format='qcow2'*)

Export an image description for Kiwi.

Parameters:

- **local**: Specifies True or False if the export has to be in the local file. Default: False.
- **path: If *local=True*, then specifies the path where file with the Kiwi description is written.** Default: */tmp*.

CLI Example:

```
salt myminion inspector.export
salt myminion inspector.export format=iso path=/opt/builds/
```

`salt.modules.inspector.`**`inspect`**(*mode='all'*, *priority=19*, ***kwargs*)

Start node inspection and save the data to the database for further query.

Parameters:

- **mode**: Clarify inspection mode: configuration, payload, all (default)

  payload

    - **filter**: Comma-separated directories to track payload.
- **priority**: (advanced) Set priority of the inspection. Default is low priority.

CLI Example:

```
salt '*' inspector.inspect
salt '*' inspector.inspect configuration
salt '*' inspector.inspect payload filter=/opt,/ext/oracle
```

`salt.modules.inspector.`**`query`**(**args*, ***kwargs*)

Query the node for specific information.

Parameters:

- **scope**: Specify scope of the query.
  - **System**: Return system data.
  - **Software**: Return software information.
  - **Services**: Return known services.

  - **Identity: Return user accounts information for this system.**

    accounts Can be either 'local', 'remote' or 'all' (equal to "local,remote"). Remote accounts cannot be resolved on all systems, but only those, which supports 'passwd -S -a'.

    disabled True (or False, default) to return only disabled accounts.

  - **payload: Payload scope parameters:**

    filter Include only results which path starts from the filter string.

    time Display time in Unix ticks or format according to the configured TZ (default) Values: ticks, tz (default)

    size Format size. Values: B, KB, MB, GB

    type Include payload type. Values (comma-separated): directory (or dir), link, file (default) Example (returns everything): type=directory,link,file

    owners Resolve UID/GID to an actual names or leave them numeric (default). Values: name (default), id

    brief Return just a list of payload elements, if True. Default: False.

  - **all**: Return all information (default).

CLI Example:

```
salt '*' inspector.query scope=system
salt '*' inspector.query scope=payload type=file,link filter=/etc size=Kb␣
↪brief=False
```

`salt.modules.inspector.`**`snapshots`**`()`

List current description snapshots.

CLI Example:

```
salt myminion inspector.snapshots
```

21.9.173 salt.modules.introspect

Functions to perform introspection on a minion, and return data in a format usable by Salt States

`salt.modules.introspect.`**`enabled_service_owners`**`()`

Return which packages own each of the services that are currently enabled.

CLI Example:

salt myminion introspect.enabled_service_owners

`salt.modules.introspect.`**`running_service_owners`**`(`*exclude=('/dev', '/home', '/media', '/proc', '/run', '/sys/', '/tmp', '/var')*`)`

Determine which packages own the currently running services. By default, excludes files whose full path starts with `/dev`, `/home`, `/media`, `/proc`, `/run`, `/sys`, `/tmp` and `/var`. This can be overridden by passing in a new list to `exclude`.

CLI Example:

salt myminion introspect.running_service_owners

`salt.modules.introspect.`**`service_highstate`**`(`*requires=True*`)`

Return running and enabled services in a highstate structure. By default also returns package dependencies for those services, which means that package definitions must be created outside this function. To drop the package dependencies, set `requires` to False.

CLI Example:

salt myminion introspect.service_highstate salt myminion introspect.service_highstate requires=False

21.9.174 salt.modules.iosconfig module

Cisco IOS configuration manipulation helpers

New in version 2019.2.0.

This module provides a collection of helper functions for Cisco IOS style configuration manipulation. This module does not have external dependencies and can be used from any Proxy or regular Minion.

`salt.modules.iosconfig.`**`clean`**(*config=None*, *path=None*, *saltenv='base'*)

Return a clean version of the config, without any special signs (such as `!` as an individual line) or empty lines, but just lines with significant value in the configuration of the network device.

config The configuration sent as text. This argument is ignored when `path` is configured.

path Absolute or remote path from where to load the configuration text. This argument allows any URI supported by *`cp.get_url`*), e.g., `salt://`, `https://`, `s3://`, `ftp:/`, etc.

saltenv: **`base`** Salt fileserver environment from which to retrieve the file. Ignored if `path` is not a `salt://` URL.

CLI Example:

```
salt '*' iosconfig.clean path=salt://path/to/my/config.txt
salt '*' iosconfig.clean path=https://bit.ly/2mAdq7z
```

`salt.modules.iosconfig.`**`diff_text`**(*candidate_config=None*, *candidate_path=None*, *running_config=None*, *running_path=None*, *saltenv='base'*)

Return the diff, as text, between the candidate and the running config.

candidate_config The candidate configuration sent as text. This argument is ignored when `candidate_path` is set.

candidate_path Absolute or remote path from where to load the candidate configuration text. This argument allows any URI supported by *`cp.get_url`*), e.g., `salt://`, `https://`, `s3://`, `ftp:/`, etc.

running_config The running configuration sent as text. This argument is ignored when `running_path` is set.

running_path Absolute or remote path from where to load the runing configuration text. This argument allows any URI supported by *`cp.get_url`*), e.g., `salt://`, `https://`, `s3://`, `ftp:/`, etc.

saltenv: **`base`** Salt fileserver environment from which to retrieve the file. Ignored if `candidate_path` or `running_path` is not a `salt://` URL.

CLI Example:

```
salt '*' iosconfig.diff_text candidate_path=salt://path/to/candidate.cfg running_
↪path=salt://path/to/running.cfg
```

`salt.modules.iosconfig.`**`diff_tree`**(*candidate_config=None*, *candidate_path=None*, *running_config=None*, *running_path=None*, *saltenv='base'*)

Return the diff, as Python dictionary, between the candidate and the running configuration.

candidate_config The candidate configuration sent as text. This argument is ignored when `candidate_path` is set.

candidate_path Absolute or remote path from where to load the candidate configuration text. This argument allows any URI supported by *cp.get_url*), e.g., `salt://`, `https://`, `s3://`, `ftp:/`, etc.

running_config The running configuration sent as text. This argument is ignored when `running_path` is set.

running_path Absolute or remote path from where to load the runing configuration text. This argument allows any URI supported by *cp.get_url*), e.g., `salt://`, `https://`, `s3://`, `ftp:/`, etc.

saltenv: **base** Salt fileserver environment from which to retrieve the file. Ignored if `candidate_path` or `running_path` is not a `salt://` URL.

CLI Example:

```
salt '*' iosconfig.diff_tree candidate_path=salt://path/to/candidate.cfg running_
↪path=salt://path/to/running.cfg
```

`salt.modules.iosconfig.`**`merge_diff`**(*initial_config=None*, *initial_path=None*, *merge_config=None*, *merge_path=None*, *saltenv='base'*)

Return the merge diff, as text, after merging the merge config into the initial config.

initial_config The initial configuration sent as text. This argument is ignored when `initial_path` is set.

initial_path Absolute or remote path from where to load the initial configuration text. This argument allows any URI supported by *cp.get_url*), e.g., `salt://`, `https://`, `s3://`, `ftp:/`, etc.

merge_config The config to be merged into the initial config, sent as text. This argument is ignored when `merge_path` is set.

merge_path Absolute or remote path from where to load the merge configuration text. This argument allows any URI supported by *cp.get_url*), e.g., `salt://`, `https://`, `s3://`, `ftp:/`, etc.

saltenv: **base** Salt fileserver environment from which to retrieve the file. Ignored if `initial_path` or `merge_path` is not a `salt://` URL.

CLI Example:

```
salt '*' iosconfig.merge_diff initial_path=salt://path/to/running.cfg merge_
↪path=salt://path/to/merge.cfg
```

`salt.modules.iosconfig.`**`merge_text`**(*initial_config=None*, *initial_path=None*, *merge_config=None*, *merge_path=None*, *saltenv='base'*)

Return the merge result of the `initial_config` with the `merge_config`, as plain text.

initial_config The initial configuration sent as text. This argument is ignored when `initial_path` is set.

initial_path Absolute or remote path from where to load the initial configuration text. This argument allows any URI supported by *cp.get_url*), e.g., `salt://`, `https://`, `s3://`, `ftp:/`, etc.

merge_config The config to be merged into the initial config, sent as text. This argument is ignored when `merge_path` is set.

merge_path Absolute or remote path from where to load the merge configuration text. This argument allows any URI supported by *cp.get_url*), e.g., `salt://`, `https://`, `s3://`, `ftp:/`, etc.

saltenv: **base** Salt fileserver environment from which to retrieve the file. Ignored if `initial_path` or `merge_path` is not a `salt://` URL.

CLI Example:

```
salt '*' iosconfig.merge_text initial_path=salt://path/to/running.cfg merge_
↪path=salt://path/to/merge.cfg
```

`salt.modules.iosconfig.`**`merge_tree`**(*initial_config=None, initial_path=None, merge_config=None, merge_path=None, saltenv='base'*)

Return the merge tree of the `initial_config` with the `merge_config`, as a Python dictionary.

initial_config The initial configuration sent as text. This argument is ignored when `initial_path` is set.

initial_path Absolute or remote path from where to load the initial configuration text. This argument allows any URI supported by *`cp.get_url`*), e.g., `salt://`, `https://`, `s3://`, `ftp:/`, etc.

merge_config The config to be merged into the initial config, sent as text. This argument is ignored when `merge_path` is set.

merge_path Absolute or remote path from where to load the merge configuration text. This argument allows any URI supported by *`cp.get_url`*), e.g., `salt://`, `https://`, `s3://`, `ftp:/`, etc.

saltenv: **`base`** Salt fileserver environment from which to retrieve the file. Ignored if `initial_path` or `merge_path` is not a `salt://` URL.

CLI Example:

```
salt '*' iosconfig.merge_tree initial_path=salt://path/to/running.cfg merge_
↪path=salt://path/to/merge.cfg
```

`salt.modules.iosconfig.`**`tree`**(*config=None, path=None, with_tags=False, saltenv='base'*)

Transform Cisco IOS style configuration to structured Python dictionary. Depending on the value of the `with_tags` argument, this function may provide different views, valuable in different situations.

config The configuration sent as text. This argument is ignored when `path` is configured.

path Absolute or remote path from where to load the configuration text. This argument allows any URI supported by *`cp.get_url`*), e.g., `salt://`, `https://`, `s3://`, `ftp:/`, etc.

with_tags: **`False`** Whether this function should return a detailed view, with tags.

saltenv: **`base`** Salt fileserver environment from which to retrieve the file. Ignored if `path` is not a `salt://` URL.

CLI Example:

```
salt '*' iosconfig.tree path=salt://path/to/my/config.txt
salt '*' iosconfig.tree path=https://bit.ly/2mAdq7z
```

21.9.175 salt.modules.ipmi

Support IPMI commands over LAN. This module does not talk to the local systems hardware through IPMI drivers. It uses a python module *pyghmi*.

depends Python module pyghmi. You can install pyghmi using pip:

```
pip install pyghmi
```

configuration The following configuration defaults can be define (pillar or config files):

```
ipmi.config:
    api_host: 127.0.0.1
    api_user: admin
```

(continues on next page)

(continued from previous page)

```
    api_pass: apassword
    api_port: 623
    api_kg: None
```

Usage can override the config defaults:

```
salt-call ipmi.get_user api_host=myipmienabled.system
                        api_user=admin api_pass=pass
                        uid=1
```

`salt.modules.ipmi.`**`create_user`**(*uid*, *name*, *password*, *channel=14*, *callback=False*, *link_auth=True*, *ipmi_msg=True*, *privilege_level='administrator'*, ***kwargs*)

create/ensure a user is created with provided settings.

Parameters

- **privilege_level** – User Privilege Limit. (Determines the maximum privilege level that the user is allowed to switch to on the specified channel.) * callback * user * operator * administrator * proprietary * no_access
- **kwargs** –
  - api_host=127.0.0.1
  - api_user=admin
  - api_pass=example
  - api_port=623
  - api_kg=None

CLI Examples:

```
salt-call ipmi.create_user uid=2 name=steverweber api_host=172.168.0.7 api_
↪pass=nevertell
```

`salt.modules.ipmi.`**`fast_connect_test`**(***kwargs*)

Returns True if connection success. This uses an aggressive timeout value!

Parameters **kwargs** –

- api_host=127.0.0.1
- api_user=admin
- api_pass=example
- api_port=623
- api_kg=None

CLI Examples:

```
salt-call ipmi.fast_connect_test api_host=172.168.0.9
```

`salt.modules.ipmi.`**`get_bootdev`**(***kwargs*)

Get current boot device override information.

Provides the current requested boot device. Be aware that not all IPMI devices support this. Even in BMCs that claim to, occasionally the BIOS or UEFI fail to honor it. This is usually only applicable to the next reboot.

Parameters **kwargs** –

- api_host=127.0.0.1
- api_user=admin
- api_pass=example
- api_port=623
- api_kg=None

CLI Example:

```
salt-call ipmi.get_bootdev api_host=127.0.0.1 api_user=admin api_pass=pass
```

`salt.modules.ipmi.`**`get_channel_access`**(*channel=14, read_mode='non_volatile', **kwargs*)
:param kwargs:api_host='127.0.0.1' api_user='admin' api_pass='example' api_port=623

Parameters

- **channel** – number [1:7]
- **read_mode** –
  - non_volatile = get non-volatile Channel Access
  - volatile = get present volatile (active) setting of Channel Access
- **kwargs** –
  - api_host=127.0.0.1
  - api_user=admin
  - api_pass=example
  - api_port=623
  - api_kg=None

Return Data

A Python dict with the following keys/values:

```
{
    alerting:
    per_msg_auth:
    user_level_auth:
    access_mode:{ (ONE OF)
        0: 'disabled',
        1: 'pre_boot',
        2: 'always',
        3: 'shared'
    }
    privilege_level: { (ONE OF)
        1: 'callback',
        2: 'user',
        3: 'operator',
        4: 'administrator',
        5: 'proprietary',
    }
}
```

CLI Examples:

```
salt-call ipmi.get_channel_access channel=1
```

salt.modules.ipmi.**get_channel_info**(*channel=14*, ***kwargs*)

Get channel info

Parameters

- **channel** – number [1:7]
- **kwargs** –
  - api_host=127.0.0.1
  - api_user=admin
  - api_pass=example
  - api_port=623
  - api_kg=None

Return Data channel session supports

```
- no_session: channel is session-less
- single: channel is single-session
- multi: channel is multi-session
- auto: channel is session-based (channel could alternate between
    single- and multi-session operation, as can occur with a
    serial/modem channel that supports connection mode auto-detect)
```

CLI Examples:

```
salt-call ipmi.get_channel_info
```

salt.modules.ipmi.**get_channel_max_user_count**(*channel=14*, ***kwargs*)

Get max users in channel

Parameters

- **channel** – number [1:7]
- **kwargs** –
  - api_host=127.0.0.1
  - api_user=admin
  - api_pass=example
  - api_port=623
  - api_kg=None

Returns int – often 16

CLI Examples:

```
salt-call ipmi.get_channel_max_user_count
```

salt.modules.ipmi.**get_health**(***kwargs*)

Get Summarize health

This provides a summary of the health of the managed system. It additionally provides an iterable list of reasons for warning, critical, or failed assessments.

good health: {'badreadings': [], 'health': 0}

Parameters **kwargs** –

- api_host=127.0.0.1
- api_user=admin
- api_pass=example
- api_port=623
- api_kg=None

CLI Example:

```
salt-call ipmi.get_health api_host=127.0.0.1 api_user=admin api_pass=pass
```

salt.modules.ipmi.**get_power**(***kwargs*)

Get current power state

The response, if successful, should contain 'powerstate' key and either 'on' or 'off' to indicate current state.

Parameters **kwargs** –

- api_host=127.0.0.1
- api_user=admin
- api_pass=example
- api_port=623
- api_kg=None

CLI Example:

```
salt-call ipmi.get_power api_host=127.0.0.1 api_user=admin api_pass=pass
```

salt.modules.ipmi.**get_sensor_data**(***kwargs*)

Get sensor readings

Iterates sensor reading objects

Parameters **kwargs** –

- api_host=127.0.0.1
- api_user=admin
- api_pass=example
- api_port=623
- api_kg=None

CLI Example:

```
salt-call ipmi.get_sensor_data api_host=127.0.0.1 api_user=admin api_pass=pass
```

salt.modules.ipmi.**get_user**(*uid, channel=14, **kwargs*)

Get user from uid and access on channel

Parameters

- **uid** – user number [1:16]
- **channel** – number [1:7]

- **kwargs** –
  - api_host=127.0.0.1
  - api_user=admin
  - api_pass=example
  - api_port=623
  - api_kg=None

Return Data

```
name: (str)
uid: (int)
channel: (int)
access:
    - callback (bool)
    - link_auth (bool)
    - ipmi_msg (bool)
    - privilege_level: (str)[callback, user, operatorm administrator,
                            proprietary, no_access]
```

CLI Examples:

```
salt-call ipmi.get_user uid=2
```

`salt.modules.ipmi.`**`get_user_access`**(*uid*, *channel=14*, ***kwargs*)

Get user access

Parameters

- **uid** – user number [1:16]
- **channel** – number [1:7]
- **kwargs** –
  - api_host=127.0.0.1
  - api_user=admin
  - api_pass=example
  - api_port=623
  - api_kg=None

Return Data

```
channel_info:
    - max_user_count = maximum number of user IDs on this channel
    - enabled_users = count of User ID slots presently in use
    - users_with_fixed_names = count of user IDs with fixed names
access:
    - callback
    - link_auth
    - ipmi_msg
    - privilege_level: [reserved, callback, user, operator
                       administrator, proprietary, no_access]
```

CLI Examples:

```
salt-call ipmi.get_user_access uid=2
```

`salt.modules.ipmi.`**`get_user_name`**`(`*uid*, *return_none_on_error=True*, ***kwargs*`)`

Get user name

Parameters

- **`uid`** – user number [1:16]
- **`return_none_on_error`** – return None on error
- **`kwargs`** –
  - api_host=127.0.0.1
  - api_user=admin
  - api_pass=example
  - api_port=623
  - api_kg=None

CLI Examples:

```
salt-call ipmi.get_user_name uid=2
```

`salt.modules.ipmi.`**`get_users`**`(`*channel=14*, ***kwargs*`)`

get list of users and access information

Parameters

- **`channel`** – number [1:7]
- **`kwargs`** –
  - api_host=127.0.0.1
  - api_user=admin
  - api_pass=example
  - api_port=623
  - api_kg=None

Returns

- name: (str)
- uid: (int)
- channel: (int)
- access:
  - callback (bool)
  - link_auth (bool)
  - ipmi_msg (bool)
  - privilege_level: (str)[callback, user, operatorm administrator, proprietary, no_access]

CLI Examples:

```
salt-call ipmi.get_users api_host=172.168.0.7
```

`salt.modules.ipmi.`**`raw_command`**(*netfn*, *command*, *bridge_request=None*, *data=()*, *retry=True*, *delay_xmit=None*, ***kwargs*)

Send raw ipmi command

This allows arbitrary IPMI bytes to be issued. This is commonly used for certain vendor specific commands.

Parameters

- **`netfn`** – Net function number
- **`command`** – Command value
- **`bridge_request`** – The target slave address and channel number for the bridge request.
- **`data`** – Command data as a tuple or list
- **`kwargs`** –
  - api_host=127.0.0.1
  - api_user=admin
  - api_pass=example
  - api_port=623
  - api_kg=None

Returns dict – The response from IPMI device

CLI Examples:

```
salt-call ipmi.raw_command netfn=0x06 command=0x46 data=[0x02]
# this will return the name of the user with id 2 in bytes
```

`salt.modules.ipmi.`**`set_bootdev`**(*bootdev='default'*, *persist=False*, *uefiboot=False*, ***kwargs*)

Set boot device to use on next reboot

Parameters

- **`bootdev`** –
  - network: Request network boot
  - hd: Boot from hard drive
  - safe: Boot from hard drive, requesting 'safe mode'
  - optical: boot from CD/DVD/BD drive
  - setup: Boot into setup utility
  - default: remove any IPMI directed boot device request
- **`persist`** – If true, ask that system firmware use this device beyond next boot. Be aware many systems do not honor this
- **`uefiboot`** – If true, request UEFI boot explicitly. Strictly speaking, the spec suggests that if not set, the system should BIOS boot and offers no "don't care" option. In practice, this flag not being set does not preclude UEFI boot on any system I've encountered.
- **`kwargs`** –
  - api_host=127.0.0.1
  - api_user=admin
  - api_pass=example

  - api_port=623
  - api_kg=None

Returns dict or True – If callback is not provided, the response

CLI Examples:

```
salt-call ipmi.set_bootdev bootdev=network persist=True
```

salt.modules.ipmi.**set_channel_access**(*channel=14, access_update_mode='non_volatile', alerting=False, per_msg_auth=False, user_level_auth=False, access_mode='always', privilege_update_mode='non_volatile', privilege_level='administrator', **kwargs*)

Set channel access

Parameters

- **`channel`** – number [1:7]
- **`access_update_mode`** –
  - 'dont_change' = don't set or change Channel Access
  - 'non_volatile' = set non-volatile Channel Access
  - 'volatile' = set volatile (active) setting of Channel Access
- **`alerting`** – PEF Alerting Enable/Disable
  - True = enable PEF Alerting
  - False = disable PEF Alerting on this channel (Alert Immediate command can still be used to generate alerts)
- **`per_msg_auth`** – Per-message Authentication
  - True = enable
  - False = disable Per-message Authentication. [Authentication required to activate any session on this channel, but authentication not used on subsequent packets for the session.]
- **`user_level_auth`** – User Level Authentication Enable/Disable
  - True = enable User Level Authentication. All User Level commands are to be authenticated per the Authentication Type that was negotiated when the session was activated.
  - False = disable User Level Authentication. Allow User Level commands to be executed without being authenticated. If the option to disable User Level Command authentication is accepted, the BMC will accept packets with Authentication Type set to None if they contain user level commands. For outgoing packets, the BMC returns responses with the same Authentication Type that was used for the request.
- **`access_mode`** – Access Mode for IPMI messaging (PEF Alerting is enabled/disabled separately from IPMI messaging)
  - disabled = disabled for IPMI messaging
  - pre_boot = pre-boot only channel only available when system is in a powered down state or in BIOS prior to start of boot.
  - always = channel always available regardless of system mode. BIOS typically dedicates the serial connection to the BMC.

    - shared = same as always available, but BIOS typically leaves the serial port available for software use.
- **`privilege_update_mode`** – Channel Privilege Level Limit. This value sets the maximum privilege level that can be accepted on the specified channel.
    - dont_change = don't set or change channel Privilege Level Limit
    - non_volatile = non-volatile Privilege Level Limit according
    - volatile = volatile setting of Privilege Level Limit
- **`privilege_level`** – Channel Privilege Level Limit
    - reserved = unused
    - callback
    - user
    - operator
    - administrator
    - proprietary = used by OEM
- **`kwargs`** –
    - api_host=127.0.0.1
    - api_user=admin
    - api_pass=example
    - api_port=623
    - api_kg=None

CLI Examples:

```
salt-call ipmi.set_channel_access privilege_level='administrator'
```

`salt.modules.ipmi.`**`set_identify`**(*on=True*, *duration=600*, ***kwargs*)

Request identify light

Request the identify light to turn off, on for a duration, or on indefinitely. Other than error exceptions,

Parameters

- **`on`** – Set to True to force on or False to force off
- **`duration`** – Set if wanting to request turn on for a duration in seconds, None = indefinitely.
- **`kwargs`** –
    - api_host=127.0.0.1
    - api_user=admin
    - api_pass=example
    - api_port=623
    - api_kg=None

CLI Examples:

```
salt-call ipmi.set_identify
```

`salt.modules.ipmi.`**`set_power`**(*state='power_on', wait=True, **kwargs*)

Request power state change

Parameters

- **`name`** –
  - power_on – system turn on
  - power_off – system turn off (without waiting for OS)
  - shutdown – request OS proper shutdown
  - reset – reset (without waiting for OS)
  - boot – If system is off, then 'on', else 'reset'
- **`ensure`** – If (bool True), do not return until system actually completes requested state change for 300 seconds. If a non-zero (int), adjust the wait time to the requested number of seconds
- **`kwargs`** –
  - api_host=127.0.0.1
  - api_user=admin
  - api_pass=example
  - api_port=623
  - api_kg=None

Returns dict – A dict describing the response retrieved

CLI Examples:

```
salt-call ipmi.set_power state=shutdown wait=True
```

`salt.modules.ipmi.`**`set_user_access`**(*uid, channel=14, callback=True, link_auth=True, ipmi_msg=True, privilege_level='administrator', **kwargs*)

Set user access

Parameters

- **`uid`** – user number [1:16]
- **`channel`** – number [1:7]
- **`callback`** – User Restricted to Callback
  - False = User Privilege Limit is determined by the User Privilege Limit parameter, below, for both callback and non-callback connections.
  - True = User Privilege Limit is determined by the User Privilege Limit parameter for callback connections, but is restricted to Callback level for non-callback connections. Thus, a user can only initiate a Callback when they 'call in' to the BMC, but once the callback connection has been made, the user could potentially establish a session as an Operator.
- **`link_auth`** – User Link authentication enable/disable (used to enable whether this user's name and password information will be used for link authentication, e.g. PPP CHAP) for

the given channel. Link authentication itself is a global setting for the channel and is enabled/disabled via the serial/modem configuration parameters.

- **ipmi_msg** – User IPMI Messaging: (used to enable/disable whether this user's name and password information will be used for IPMI Messaging. In this case, 'IPMI Messaging' refers to the ability to execute generic IPMI commands that are not associated with a particular payload type. For example, if IPMI Messaging is disabled for a user, but that user is enabled for activating the SOL payload type, then IPMI commands associated with SOL and session management, such as Get SOL Configuration Parameters and Close Session are available, but generic IPMI commands such as Get SEL Time are unavailable.)
- **privilege_level** – User Privilege Limit. (Determines the maximum privilege level that the user is allowed to switch to on the specified channel.)
  - callback
  - user
  - operator
  - administrator
  - proprietary
  - no_access
- **kwargs** –
  - api_host=127.0.0.1
  - api_user=admin
  - api_pass=example
  - api_port=623
  - api_kg=None

CLI Examples:

```
salt-call ipmi.set_user_access uid=2 privilege_level='operator'
```

`salt.modules.ipmi.`**`set_user_name`**(*uid*, *name*, ***kwargs*)

Set user name

Parameters

- **uid** – user number [1:16]
- **name** – username (limit of 16bytes)
- **kwargs** –
  - api_host=127.0.0.1
  - api_user=admin
  - api_pass=example
  - api_port=623
  - api_kg=None

CLI Examples:

```
salt-call ipmi.set_user_name uid=2 name='steverweber'
```

`salt.modules.ipmi.`**`set_user_password`**(*uid*, *mode='set_password'*, *password=None*, ***kwargs*)
Set user password and (modes)

Parameters

- **uid** – id number of user. see: get_names_uid()['name']
- **mode** –
  - disable = disable user connections
  - enable = enable user connections
  - set_password = set or ensure password
  - test_password = test password is correct
- **password** – max 16 char string (optional when mode is [disable or enable])
- **kwargs** –
  - api_host=127.0.0.1
  - api_user=admin
  - api_pass=example
  - api_port=623
  - api_kg=None

Returns True on success when mode = test_password, return False on bad password

CLI Example:

```
salt-call ipmi.set_user_password api_host=127.0.0.1 api_user=admin api_pass=pass
                                 uid=1 password=newPass
salt-call ipmi.set_user_password uid=1 mode=enable
```

`salt.modules.ipmi.`**`user_delete`**(*uid*, *channel=14*, ***kwargs*)
Delete user (helper)

Parameters

- **uid** – user number [1:16]
- **channel** – number [1:7]
- **kwargs** –
  - api_host=127.0.0.1
  - api_user=admin
  - api_pass=example
  - api_port=623
  - api_kg=None

CLI Examples:

```
salt-call ipmi.user_delete uid=2
```

21.9.176 salt.modules.ipset

Support for ipset

`salt.modules.ipset.`**`add`**(*setname=None, entry=None, family='ipv4', **kwargs*)

Append an entry to the specified set.

CLI Example:

```
salt '*' ipset.add setname 192.168.1.26

salt '*' ipset.add setname 192.168.0.3,AA:BB:CC:DD:EE:FF
```

`salt.modules.ipset.`**`check`**(*set=None, entry=None, family='ipv4'*)

Check that an entry exists in the specified set.

set The ipset name

entry An entry in the ipset. This parameter can be a single IP address, a range of IP addresses, or a subnet block. Example:

```
192.168.0.1
192.168.0.2-192.168.0.19
192.168.0.0/25
```

family IP protocol version: ipv4 or ipv6

CLI Example:

```
salt '*' ipset.check setname '192.168.0.1 comment "Hello"'
```

`salt.modules.ipset.`**`check_set`**(*set=None, family='ipv4'*)

Check that given ipset set exists.

New in version 2014.7.0.

CLI Example:

```
salt '*' ipset.check_set setname
```

`salt.modules.ipset.`**`delete`**(*set=None, entry=None, family='ipv4', **kwargs*)

Delete an entry from the specified set.

CLI Example:

```
salt '*' ipset.delete setname 192.168.0.3,AA:BB:CC:DD:EE:FF
```

`salt.modules.ipset.`**`delete_set`**(*set=None, family='ipv4'*)

New in version 2014.7.0.

Delete ipset set.

CLI Example:

```
salt '*' ipset.delete_set custom_set

IPv6:
salt '*' ipset.delete_set custom_set family=ipv6
```

`salt.modules.ipset.`**`flush`**(*set=None, family='ipv4'*)

Flush entries in the specified set, Flush all sets if set is not specified.

CLI Example:

```
salt '*' ipset.flush

salt '*' ipset.flush set

IPv6:
salt '*' ipset.flush

salt '*' ipset.flush set
```

`salt.modules.ipset.`**`list_sets`**(*family='ipv4'*)

New in version 2014.7.0.

List all ipset sets.

CLI Example:

```
salt '*' ipset.list_sets
```

`salt.modules.ipset.`**`long_range`**(*start, end*)

`salt.modules.ipset.`**`new_set`**(*set=None, set_type=None, family='ipv4', comment=False, **kwargs*)

New in version 2014.7.0.

Create new custom set

CLI Example:

```
salt '*' ipset.new_set custom_set list:set

salt '*' ipset.new_set custom_set list:set comment=True

IPv6:
salt '*' ipset.new_set custom_set list:set family=ipv6
```

`salt.modules.ipset.`**`rename_set`**(*set=None, new_set=None, family='ipv4'*)

New in version 2014.7.0.

Delete ipset set.

CLI Example:

```
salt '*' ipset.rename_set custom_set new_set=new_set_name

IPv6:
salt '*' ipset.rename_set custom_set new_set=new_set_name family=ipv6
```

`salt.modules.ipset.`**`test`**(*set=None, entry=None, family='ipv4', **kwargs*)

Test if an entry is in the specified set.

CLI Example:

```
salt '*' ipset.test setname 192.168.0.2

IPv6:
salt '*' ipset.test setname fd81:fc56:9ac7::/48
```

`salt.modules.ipset.`**`version`**()

Return version from ipset –version

CLI Example:

```
salt '*' ipset.version
```

21.9.177 salt.modules.iptables

Support for iptables

Configuration Options

The following options can be set in the minion config, grains, pillar, or master config. The configuration is read using `config.get`.

- `iptables.save_filters`: List of REGEX strings to FILTER OUT matching lines

  This is useful for filtering out chains, rules, etc that you do not wish to persist, such as ephemeral Docker rules.

  The default is to not filter out anything.

```
iptables.save_filters:
  - "-j CATTLE_PREROUTING"
  - "-j DOCKER"
  - "-A POSTROUTING"
  - "-A CATTLE_POSTROUTING"
  - "-A FORWARD"
```

`salt.modules.iptables.`**`append`**(*table='filter', chain=None, rule=None, family='ipv4'*)

Append a rule to the specified table/chain.

This function accepts a rule in a standard iptables command format, starting with the chain. Trying to force users to adapt to a new method of creating rules would be irritating at best, and we already have a parser that can handle it.

CLI Example:

```
salt '*' iptables.append filter INPUT \
    rule='-m state --state RELATED,ESTABLISHED -j ACCEPT'

IPv6:
salt '*' iptables.append filter INPUT \
    rule='-m state --state RELATED,ESTABLISHED -j ACCEPT' \
    family=ipv6
```

`salt.modules.iptables.`**`build_rule`**(*table='filter', chain=None, command=None, position='', full=None, family='ipv4', **kwargs*)

Build a well-formatted iptables rule based on kwargs. A *table* and *chain* are not required, unless *full* is True.

If *full* is *True*, then *table*, *chain* and *command* are required. *command* may be specified as either a short option ('I') or a long option (*–insert*). This will return the iptables command, exactly as it would be used from the command line.

If a position is required (as with *-I* or *-D*), it may be specified as *position*. This will only be useful if *full* is True.

If *state* is passed, it will be ignored, use *connstate*. If *connstate* is passed in, it will automatically be changed to *state*.

To pass in jump options that doesn't take arguments, pass in an empty string.

Note: Whereas iptables will accept `-p`, `--proto[c[o[l]]]` as synonyms of `--protocol`, if `--proto` appears in an iptables command after the appearance of `-m policy`, it is interpreted as the `--proto` option of the policy extension (see the iptables-extensions(8) man page).

CLI Examples:

```
salt '*' iptables.build_rule match=state \
    connstate=RELATED,ESTABLISHED jump=ACCEPT

salt '*' iptables.build_rule filter INPUT command=I position=3 \
    full=True match=state connstate=RELATED,ESTABLISHED jump=ACCEPT

salt '*' iptables.build_rule filter INPUT command=A \
    full=True match=state connstate=RELATED,ESTABLISHED \
    source='127.0.0.1' jump=ACCEPT

.. Invert Rules
salt '*' iptables.build_rule filter INPUT command=A \
    full=True match=state connstate=RELATED,ESTABLISHED \
    source='!127.0.0.1' jump=ACCEPT

salt '*' iptables.build_rule filter INPUT command=A \
    full=True match=state connstate=RELATED,ESTABLISHED \
    destination='not 127.0.0.1' jump=ACCEPT

IPv6:
salt '*' iptables.build_rule match=state \
    connstate=RELATED,ESTABLISHED jump=ACCEPT \
    family=ipv6
salt '*' iptables.build_rule filter INPUT command=I position=3 \
    full=True match=state connstate=RELATED,ESTABLISHED jump=ACCEPT \
    family=ipv6
```

`salt.modules.iptables.`**`check`**(*table='filter', chain=None, rule=None, family='ipv4'*)

Check for the existence of a rule in the table and chain

This function accepts a rule in a standard iptables command format, starting with the chain. Trying to force users to adapt to a new method of creating rules would be irritating at best, and we already have a parser that can handle it.

CLI Example:

```
salt '*' iptables.check filter INPUT \
    rule='-m state --state RELATED,ESTABLISHED -j ACCEPT'

IPv6:
salt '*' iptables.check filter INPUT \
    rule='-m state --state RELATED,ESTABLISHED -j ACCEPT' \
    family=ipv6
```

`salt.modules.iptables.`**`check_chain`**(*table='filter', chain=None, family='ipv4'*)

New in version 2014.1.0.

Check for the existence of a chain in the table

CLI Example:

```
salt '*' iptables.check_chain filter INPUT

IPv6:
salt '*' iptables.check_chain filter INPUT family=ipv6
```

`salt.modules.iptables.`**`delete`**(*table*, *chain=None*, *position=None*, *rule=None*, *family='ipv4'*)

Delete a rule from the specified table/chain, specifying either the rule in its entirety, or the rule's position in the chain.

This function accepts a rule in a standard iptables command format, starting with the chain. Trying to force users to adapt to a new method of creating rules would be irritating at best, and we already have a parser that can handle it.

CLI Examples:

```
salt '*' iptables.delete filter INPUT position=3
salt '*' iptables.delete filter INPUT \
    rule='-m state --state RELATED,ESTABLISHED -j ACCEPT'

IPv6:
salt '*' iptables.delete filter INPUT position=3 family=ipv6
salt '*' iptables.delete filter INPUT \
    rule='-m state --state RELATED,ESTABLISHED -j ACCEPT' \
    family=ipv6
```

`salt.modules.iptables.`**`delete_chain`**(*table='filter'*, *chain=None*, *family='ipv4'*)

New in version 2014.1.0.

Delete custom chain to the specified table.

CLI Example:

```
salt '*' iptables.delete_chain filter CUSTOM_CHAIN

IPv6:
salt '*' iptables.delete_chain filter CUSTOM_CHAIN family=ipv6
```

`salt.modules.iptables.`**`flush`**(*table='filter'*, *chain=''*, *family='ipv4'*)

Flush the chain in the specified table, flush all chains in the specified table if not specified chain.

CLI Example:

```
salt '*' iptables.flush filter INPUT

IPv6:
salt '*' iptables.flush filter INPUT family=ipv6
```

`salt.modules.iptables.`**`get_policy`**(*table='filter'*, *chain=None*, *family='ipv4'*)

Return the current policy for the specified table/chain

CLI Example:

```
salt '*' iptables.get_policy filter INPUT

IPv6:
salt '*' iptables.get_policy filter INPUT family=ipv6
```

`salt.modules.iptables.`**`get_rules`**(*family='ipv4'*)

Return a data structure of the current, in-memory rules

CLI Example:

```
salt '*' iptables.get_rules

IPv6:
salt '*' iptables.get_rules family=ipv6
```

`salt.modules.iptables.`**`get_saved_policy`**(*table='filter', chain=None, conf_file=None, family='ipv4'*)

Return the current policy for the specified table/chain

CLI Examples:

```
salt '*' iptables.get_saved_policy filter INPUT
salt '*' iptables.get_saved_policy filter INPUT \
    conf_file=/etc/iptables.saved

IPv6:
salt '*' iptables.get_saved_policy filter INPUT family=ipv6
salt '*' iptables.get_saved_policy filter INPUT \
    conf_file=/etc/iptables.saved family=ipv6
```

`salt.modules.iptables.`**`get_saved_rules`**(*conf_file=None, family='ipv4'*)

Return a data structure of the rules in the conf file

CLI Example:

```
salt '*' iptables.get_saved_rules

IPv6:
salt '*' iptables.get_saved_rules family=ipv6
```

`salt.modules.iptables.`**`insert`**(*table='filter', chain=None, position=None, rule=None, family='ipv4'*)

Insert a rule into the specified table/chain, at the specified position.

This function accepts a rule in a standard iptables command format, starting with the chain. Trying to force users to adapt to a new method of creating rules would be irritating at best, and we already have a parser that can handle it.

If the position specified is a negative number, then the insert will be performed counting from the end of the list. For instance, a position of -1 will insert the rule as the second to last rule. To insert a rule in the last position, use the append function instead.

CLI Examples:

```
salt '*' iptables.insert filter INPUT position=3 \
    rule='-m state --state RELATED,ESTABLISHED -j ACCEPT'

IPv6:
salt '*' iptables.insert filter INPUT position=3 \
    rule='-m state --state RELATED,ESTABLISHED -j ACCEPT' \
    family=ipv6
```

`salt.modules.iptables.`**`new_chain`**(*table='filter', chain=None, family='ipv4'*)

New in version 2014.1.0.

Create new custom chain to the specified table.

CLI Example:

```
salt '*' iptables.new_chain filter CUSTOM_CHAIN

IPv6:
salt '*' iptables.new_chain filter CUSTOM_CHAIN family=ipv6
```

`salt.modules.iptables.`**`save`**(*filename=None, family='ipv4'*)
Save the current in-memory rules to disk

CLI Example:

```
salt '*' iptables.save /etc/sysconfig/iptables

IPv6:
salt '*' iptables.save /etc/sysconfig/iptables family=ipv6
```

`salt.modules.iptables.`**`set_policy`**(*table='filter', chain=None, policy=None, family='ipv4'*)
Set the current policy for the specified table/chain

CLI Example:

```
salt '*' iptables.set_policy filter INPUT ACCEPT

IPv6:
salt '*' iptables.set_policy filter INPUT ACCEPT family=ipv6
```

`salt.modules.iptables.`**`version`**(*family='ipv4'*)
Return version from iptables –version

CLI Example:

```
salt '*' iptables.version

IPv6:
salt '*' iptables.version family=ipv6
```

21.9.178 salt.modules.iwtools module

Support for Wireless Tools for Linux

`salt.modules.iwtools.`**`list_interfaces`**(*style=None*)
List all of the wireless interfaces

CLI Example:

salt minion iwtools.list_interfaces

`salt.modules.iwtools.`**`scan`**(*iface, style=None*)
List networks on a wireless interface

CLI Examples:

salt minion iwtools.scan wlp3s0 salt minion iwtools.scan wlp3s0 list

`salt.modules.iwtools.`**`set_mode`**(*iface, mode*)
List networks on a wireless interface

CLI Example:

salt minion iwtools.set_mode wlp3s0 Managed

21.9.179 salt.modules.jboss7

Module for managing JBoss AS 7 through the CLI interface.

New in version 2015.5.0.

In order to run each function, jboss_config dictionary with the following properties must be passed:

- cli_path: the path to jboss-cli script, for example: '/opt/jboss/jboss-7.0/bin/jboss-cli.sh'
- controller: the IP address and port of controller, for example: 10.11.12.13:9999
- cli_user: username to connect to jboss administration console if necessary
- cli_password: password to connect to jboss administration console if necessary

Example:

```
jboss_config:
   cli_path: '/opt/jboss/jboss-7.0/bin/jboss-cli.sh'
   controller: 10.11.12.13:9999
   cli_user: 'jbossadm'
   cli_password: 'jbossadm'
```

`salt.modules.jboss7.`**`create_datasource`**(*jboss_config*, *name*, *datasource_properties*, *profile=None*)

Create datasource in running jboss instance

jboss_config Configuration dictionary with properties specified above.

name Datasource name

datasource_properties

A dictionary of datasource properties to be created:

- driver-name: mysql
- connection-url: 'jdbc:mysql://localhost:3306/sampleDatabase'
- jndi-name: 'java:jboss/datasources/sampleDS'
- user-name: sampleuser
- password: secret
- min-pool-size: 3
- use-java-context: True

profile The profile name (JBoss domain mode only)

CLI Example:

```
salt '*' jboss7.create_datasource '{"cli_path": "integration.modules.sysmod.
↪SysModuleTest.test_valid_docs", "controller": "10.11.12.13:9999", "cli_user":
↪"jbossadm", "cli_password": "jbossadm"}' 'my_datasource' '{"driver-name": "mysql
↪", "connection-url": "jdbc:mysql://localhost:3306/sampleDatabase", "jndi-name":
↪"java:jboss/datasources/sampleDS", "user-name": "sampleuser", "password":
↪"secret", "min-pool-size": 3, "use-java-context": True}'
```

`salt.modules.jboss7.`**`create_simple_binding`**(*jboss_config*, *binding_name*, *value*, *profile=None*)

Create a simple jndi binding in the running jboss instance

jboss_config Configuration dictionary with properties specified above.

binding_name Binding name to be created

value Binding value

profile The profile name (JBoss domain mode only)

CLI Example:

```
salt '*' jboss7.create_simple_binding \
        '{"cli_path": "integration.modules.sysmod.SysModuleTest.test_valid_docs",↩
↪\
        "controller": "10.11.12.13:9999", "cli_user": "jbossadm", "cli_password":
↪"jbossadm"}' \
        my_binding_name my_binding_value
```

`salt.modules.jboss7.`**`deploy`**(*jboss_config, source_file*)

Deploy the application on the jboss instance from the local file system where minion is running.

jboss_config Configuration dictionary with properties specified above.

source_file Source file to deploy from

CLI Example:

```
salt '*' jboss7.deploy '{"cli_path": "integration.modules.sysmod.SysModuleTest.
↪test_valid_docs", "controller": "10.11.12.13:9999", "cli_user": "jbossadm",
↪"cli_password": "jbossadm"}' /opt/deploy_files/my_deploy
```

`salt.modules.jboss7.`**`list_deployments`**(*jboss_config*)

List all deployments on the jboss instance

jboss_config

Configuration dictionary with properties specified above.

CLI Example:

```
salt '*' jboss7.list_deployments '{"cli_path": "integration.modules.sysmod.
↪SysModuleTest.test_valid_docs", "controller": "10.11.12.13:9999", "cli_user
↪": "jbossadm", "cli_password": "jbossadm"}'
```

`salt.modules.jboss7.`**`read_datasource`**(*jboss_config, name, profile=None*)

Read datasource properties in the running jboss instance.

jboss_config Configuration dictionary with properties specified above.

name Datasource name

profile Profile name (JBoss domain mode only)

CLI Example:

```
salt '*' jboss7.read_datasource '{"cli_path": "integration.modules.sysmod.
↪SysModuleTest.test_valid_docs", "controller": "10.11.12.13:9999", "cli_user":
↪"jbossadm", "cli_password": "jbossadm"}'
```

`salt.modules.jboss7.`**`read_simple_binding`**(*jboss_config, binding_name, profile=None*)

Read jndi binding in the running jboss instance

jboss_config Configuration dictionary with properties specified above.

binding_name Binding name to be created

profile The profile name (JBoss domain mode only)

CLI Example:

salt '*' jboss7.read_simple_binding '{"cli_path": "integration.modules.sysmod.SysModuleTest.test_valid_docs", "controller": "10.11.12.13:9999", "cli_user": "jbossadm", "cli_password": "jbossadm"}' my_binding_name

`salt.modules.jboss7.`**`reload_`**(*jboss_config*, *host=None*)

Reload running jboss instance

jboss_config Configuration dictionary with properties specified above.

host The name of the host. JBoss domain mode only - and required if running in domain mode. The host name is the "name" attribute of the "host" element in host.xml

CLI Example:

```
salt '*' jboss7.reload '{"cli_path": "integration.modules.sysmod.SysModuleTest.
↪test_valid_docs", "controller": "10.11.12.13:9999", "cli_user": "jbossadm",
↪"cli_password": "jbossadm"}'
```

`salt.modules.jboss7.`**`remove_datasource`**(*jboss_config*, *name*, *profile=None*)

Remove an existing datasource from the running jboss instance.

jboss_config Configuration dictionary with properties specified above.

name Datasource name

profile The profile (JBoss domain mode only)

CLI Example:

```
salt '*' jboss7.remove_datasource '{"cli_path": "integration.modules.sysmod.
↪SysModuleTest.test_valid_docs", "controller": "10.11.12.13:9999", "cli_user":
↪"jbossadm", "cli_password": "jbossadm"}' my_datasource_name
```

`salt.modules.jboss7.`**`status`**(*jboss_config*, *host=None*, *server_config=None*)

Get status of running jboss instance.

jboss_config Configuration dictionary with properties specified above.

host The name of the host. JBoss domain mode only - and required if running in domain mode. The host name is the "name" attribute of the "host" element in host.xml

server_config The name of the Server Configuration. JBoss Domain mode only - and required if running in domain mode.

CLI Example:

```
salt '*' jboss7.status '{"cli_path": "integration.modules.sysmod.SysModuleTest.
↪test_valid_docs", "controller": "10.11.12.13:9999", "cli_user": "jbossadm",
↪"cli_password": "jbossadm"}'
```

`salt.modules.jboss7.`**`stop_server`**(*jboss_config*, *host=None*)

Stop running jboss instance

jboss_config Configuration dictionary with properties specified above.

host The name of the host. JBoss domain mode only - and required if running in domain mode. The host name is the "name" attribute of the "host" element in host.xml

CLI Example:

```
salt '*' jboss7.stop_server '{"cli_path": "integration.modules.sysmod.
↪SysModuleTest.test_valid_docs", "controller": "10.11.12.13:9999", "cli_user":
↪"jbossadm", "cli_password": "jbossadm"}'
```

`salt.modules.jboss7.`**`undeploy`**(*jboss_config*, *deployment*)

Undeploy the application from jboss instance

jboss_config Configuration dictionary with properties specified above.

deployment Deployment name to undeploy

CLI Example:

```
salt '*' jboss7.undeploy '{"cli_path": "integration.modules.sysmod.SysModuleTest.
↪test_valid_docs", "controller": "10.11.12.13:9999", "cli_user": "jbossadm",
↪"cli_password": "jbossadm"}' my_deployment
```

`salt.modules.jboss7.`**`update_datasource`**(*jboss_config*, *name*, *new_properties*, *profile=None*)

Update an existing datasource in running jboss instance. If the property doesn't exist if will be created, if it does, it will be updated with the new value

jboss_config Configuration dictionary with properties specified above.

name Datasource name

new_properties

A dictionary of datasource properties to be updated. For example:

- driver-name: mysql
- connection-url: 'jdbc:mysql://localhost:3306/sampleDatabase'
- jndi-name: 'java:jboss/datasources/sampleDS'
- user-name: sampleuser
- password: secret
- min-pool-size: 3
- use-java-context: True

profile The profile name (JBoss domain mode only)

CLI Example:

```
salt '*' jboss7.update_datasource '{"cli_path": "integration.modules.sysmod.
↪SysModuleTest.test_valid_docs", "controller": "10.11.12.13:9999", "cli_user":
↪"jbossadm", "cli_password": "jbossadm"}' 'my_datasource' '{"driver-name": "mysql
↪", "connection-url": "jdbc:mysql://localhost:3306/sampleDatabase", "jndi-name":
↪"java:jboss/datasources/sampleDS", "user-name": "sampleuser", "password":
↪"secret", "min-pool-size": 3, "use-java-context": True}'
```

`salt.modules.jboss7.`**`update_simple_binding`**(*jboss_config*, *binding_name*, *value*, *profile=None*)

Update the simple jndi binding in the running jboss instance

jboss_config Configuration dictionary with properties specified above.

binding_name Binding name to be updated

value New binding value

profile The profile name (JBoss domain mode only)

CLI Example:

```
salt '*' jboss7.update_simple_binding '{"cli_path": "integration.modules.sysmod.
↪SysModuleTest.test_valid_docs", "controller": "10.11.12.13:9999", "cli_user":
↪"jbossadm", "cli_password": "jbossadm"}' my_binding_name my_binding_value
```

21.9.180 salt.modules.jboss7_cli

Module for low-level interaction with JbossAS7 through CLI.

This module exposes two ways of interaction with the CLI, either through commands or operations.

Note: Following JBoss documentation (https://developer.jboss.org/wiki/CommandLineInterface): "Operations are considered a low level but comprehensive way to manage the AS controller, i.e. if it can't be done with operations it can't be done in any other way. Commands, on the other hand, are more user-friendly in syntax, although most of them still translate into operation requests and some of them even into a few composite operation requests, i.e. commands also simplify some management operations from the user's point of view."

The difference between calling a command or operation is in handling the result. Commands return a zero return code if operation is successful or return non-zero return code and print an error to standard output in plain text, in case of an error.

Operations return a json-like structure, that contain more information about the result. In case of a failure, they also return a specific return code. This module parses the output from the operations and returns it as a dictionary so that an execution of an operation can then be verified against specific errors.

In order to run each function, jboss_config dictionary with the following properties must be passed:

- cli_path: the path to jboss-cli script, for example: '/opt/jboss/jboss-7.0/bin/jboss-cli.sh'
- controller: the IP address and port of controller, for example: 10.11.12.13:9999
- cli_user: username to connect to jboss administration console if necessary
- cli_password: password to connect to jboss administration console if necessary

Example:

```
jboss_config:
   cli_path: '/opt/jboss/jboss-7.0/bin/jboss-cli.sh'
   controller: 10.11.12.13:9999
   cli_user: 'jbossadm'
   cli_password: 'jbossadm'
```

`salt.modules.jboss7_cli.`**`run_command`**(*jboss_config*, *command*, *fail_on_error=True*)

Execute a command against jboss instance through the CLI interface.

jboss_config Configuration dictionary with properties specified above.

command Command to execute against jboss instance

fail_on_error (default=True) Is true, raise CommandExecutionError exception if execution fails. If false, 'success' property of the returned dictionary is set to False

CLI Example:

```
salt '*' jboss7_cli.run_command '{"cli_path": "integration.modules.sysmod.
↪SysModuleTest.test_valid_docs", "controller": "10.11.12.13:9999", "cli_user":
↪"jbossadm", "cli_password": "jbossadm"}' my_command
```

`salt.modules.jboss7_cli.`**`run_operation`**(*jboss_config, operation, fail_on_error=True, retries=1*)
Execute an operation against jboss instance through the CLI interface.

jboss_config Configuration dictionary with properties specified above.

operation An operation to execute against jboss instance

fail_on_error (default=True) Is true, raise CommandExecutionError exception if execution fails. If false, 'success' property of the returned dictionary is set to False

retries: Number of retries in case of "JBAS012144: Could not connect to remote" error.

CLI Example:

```
salt '*' jboss7_cli.run_operation '{"cli_path": "integration.modules.sysmod.
↪SysModuleTest.test_valid_docs", "controller": "10.11.12.13:9999", "cli_user":
↪"jbossadm", "cli_password": "jbossadm"}' my_operation
```

21.9.181 salt.modules.jenkinsmod module

Module for controlling Jenkins

depends python-jenkins

New in version 2016.3.0.

depends python-jenkins Python module (not to be confused with jenkins)

configuration This module can be used by either passing an api key and version directly or by specifying both in a configuration profile in the salt master/minion config.

For example:

```
jenkins:
  api_key: peWcBiMOS9HrZG15peWcBiMOS9HrZG15
```

`salt.modules.jenkinsmod.`**`build_job`**(*name=None, parameters=None*)
Initiate a build for the provided job.

Parameters

- **`name`** – The name of the job is check if it exists.
- **`parameters`** – Parameters to send to the job.

Returns True is successful, otherwise raise an exception.

CLI Example:

```
salt '*' jenkins.build_job jobname
```

`salt.modules.jenkinsmod.`**`create_job`**(*name=None, config_xml=None, saltenv='base'*)
Return the configuration file.

Parameters

- **`name`** – The name of the job is check if it exists.
- **`config_xml`** – The configuration file to use to create the job.
- **`saltenv`** – The environment to look for the file in.

Returns The configuration file used for the job.

CLI Example:

```
salt '*' jenkins.create_job jobname

salt '*' jenkins.create_job jobname config_xml='salt://jenkins/config.xml'
```

`salt.modules.jenkinsmod.`**`delete_job`**`(`*name=None*`)`

Return true is job is deleted successfully.

Parameters **`name`** – The name of the job to delete.

Returns Return true if job is deleted successfully.

CLI Example:

```
salt '*' jenkins.delete_job jobname
```

`salt.modules.jenkinsmod.`**`disable_job`**`(`*name=None*`)`

Return true is job is disabled successfully.

Parameters **`name`** – The name of the job to disable.

Returns Return true if job is disabled successfully.

CLI Example:

```
salt '*' jenkins.disable_job jobname
```

`salt.modules.jenkinsmod.`**`enable_job`**`(`*name=None*`)`

Return true is job is enabled successfully.

Parameters **`name`** – The name of the job to enable.

Returns Return true if job is enabled successfully.

CLI Example:

```
salt '*' jenkins.enable_job jobname
```

`salt.modules.jenkinsmod.`**`get_job_config`**`(`*name=None*`)`

Return the current job configuration for the provided job.

Parameters **`name`** – The name of the job to return the configuration for.

Returns The configuration for the job specified.

CLI Example:

```
salt '*' jenkins.get_job_config jobname
```

`salt.modules.jenkinsmod.`**`get_job_info`**`(`*name=None*`)`

Return information about the Jenkins job.

Parameters **`name`** – The name of the job is check if it exists.

Returns Information about the Jenkins job.

CLI Example:

```
salt '*' jenkins.get_job_info jobname
```

`salt.modules.jenkinsmod.`**`get_jobs`**`()`

Return the currently configured jobs.

Returns The currently configured jobs.

CLI Example:

```
salt '*' jenkins.get_jobs
```

`salt.modules.jenkinsmod.`**`get_version`**`()`

Return version of Jenkins

Returns The version of Jenkins

CLI Example:

```
salt '*' jenkins.get_version
```

`salt.modules.jenkinsmod.`**`job_exists`**`(`*name=None*`)`

Check whether the job exists in configured Jenkins jobs.

Parameters **`name`** – The name of the job is check if it exists.

Returns True if job exists, False if job does not exist.

CLI Example:

```
salt '*' jenkins.job_exists jobname
```

`salt.modules.jenkinsmod.`**`job_status`**`(`*name=None*`)`

Return the current status, enabled or disabled, of the job.

Parameters **`name`** – The name of the job to return status for

Returns Return true if enabled or false if disabled.

CLI Example:

```
salt '*' jenkins.job_status jobname
```

`salt.modules.jenkinsmod.`**`plugin_installed`**`(`*name*`)`

New in version 2016.11.0.

Return if the plugin is installed for the provided plugin name.

Parameters **`name`** – The name of the parameter to confirm installation.

Returns True if plugin exists, False if plugin does not exist.

CLI Example:

```
salt '*' jenkins.plugin_installed pluginName
```

`salt.modules.jenkinsmod.`**`run`**`(`*script*`)`

New in version 2017.7.0.

Execute a script on the jenkins master

Parameters **`script`** – The script

CLI Example:

```
salt '*' jenkins.run 'Jenkins.instance.doSafeRestart()'
```

`salt.modules.jenkinsmod.`**`update_job`**`(`*name=None, config_xml=None, saltenv='base'*`)`

Return the updated configuration file.

Parameters

- **`name`** – The name of the job is check if it exists.
- **`config_xml`** – The configuration file to use to create the job.
- **`saltenv`** – The environment to look for the file in.

Returns The configuration file used for the job.

CLI Example:

```
salt '*' jenkins.update_job jobname

salt '*' jenkins.update_job jobname config_xml='salt://jenkins/config.xml'
```

21.9.182 salt.modules.jinja

Module for checking jinja maps and verifying the result of loading JSON/YAML files

New in version 3000.

`salt.modules.jinja.`**`import_json`**`(`*path*`)`

Loads JSON data from the specified path

CLI Example:

```
salt myminion jinja.import_JSON myformula/foo.json
```

`salt.modules.jinja.`**`import_yaml`**`(`*path*`)`

Loads YAML data from the specified path

CLI Example:

```
salt myminion jinja.import_yaml myformula/foo.yaml
```

`salt.modules.jinja.`**`load_map`**`(`*path, value*`)`

Loads the map at the specified path, and returns the specified value from that map.

CLI Example:

```
# Assuming the map is loaded in your formula SLS as follows:
#
# {% from "myformula/map.jinja" import myformula with context %}
#
# the following syntax can be used to load the map and check the
# results:
salt myminion jinja.load_map myformula/map.jinja myformula
```

21.9.183 salt.modules.jira_mod module

JIRA Execution module

New in version 2019.2.0.

Execution module to manipulate JIRA tickets via Salt.

This module requires the `jira` Python library to be installed.

Configuration example:

```
jira:
  server: https://jira.atlassian.org
  username: salt
  password: pass
```

`salt.modules.jira_mod.`**`add_comment`**(*issue_key, comment, visibility=None, is_internal=False, server=None, username=None, password=None*)

Add a comment to an existing ticket. Return `True` when it successfully added the comment.

issue_key The issue ID to add the comment to.

comment The body of the comment to be added.

visibility: `None` A dictionary having two keys:

- `type`: is `role` (or `group` if the JIRA server has configured comment visibility for groups).
- `value`: the name of the role (or group) to which viewing of this comment will be restricted.

is_internal: `False` Whether a comment has to be marked as `Internal` in Jira Service Desk.

CLI Example:

```
salt '*' jira.add_comment NE-123 'This is a comment'
```

`salt.modules.jira_mod.`**`assign_issue`**(*issue_key, assignee, server=None, username=None, password=None*)

Assign the issue to an existing user. Return `True` when the issue has been properly assigned.

issue_key The JIRA ID of the ticket to manipulate.

assignee The name of the user to assign the ticket to.

CLI Example:

salt '*' jira.assign_issue NET-123 example_user

`salt.modules.jira_mod.`**`create_issue`**(*project, summary, description, template_engine='jinja', context=None, defaults=None, saltenv='base', issuetype='Bug', priority='Normal', labels=None, assignee=None, server=None, username=None, password=None, **kwargs*)

Create a JIRA issue using the named settings. Return the JIRA ticket ID.

project The name of the project to attach the JIRA ticket to.

summary The summary (title) of the JIRA ticket. When the `template_engine` argument is set to a proper value of an existing Salt template engine (e.g., `jinja`, `mako`, etc.) it will render the `summary` before creating the ticket.

description The full body description of the JIRA ticket. When the `template_engine` argument is set to a proper value of an existing Salt template engine (e.g., `jinja`, `mako`, etc.) it will render the `description` before creating the ticket.

template_engine: `jinja` The name of the template engine to be used to render the values of the `summary` and `description` arguments. Default: `jinja`.

context: `None` The context to pass when rendering the `summary` and `description`. This argument is ignored when `template_engine` is set as `None`

defaults: `None` Default values to pass to the Salt rendering pipeline for the `summary` and `description` arguments. This argument is ignored when `template_engine` is set as `None`.

saltenv: `base` The Salt environment name (for the rendering system).

issuetype: `Bug` The type of the JIRA ticket. Default: `Bug`.

priority: `Normal` The priority of the JIRA ticket. Default: `Normal`.

labels: `None` A list of labels to add to the ticket.

assignee: `None` The name of the person to assign the ticket to.

CLI Examples:

```
salt '*' jira.create_issue NET 'Ticket title' 'Ticket description'
salt '*' jira.create_issue NET 'Issue on {{ opts.id }}' 'Error detected on {{
↪opts.id }}' template_engine=jinja
```

`salt.modules.jira_mod.`**`issue_closed`**(*issue_key, server=None, username=None, password=None*)

Check if the issue is closed.

issue_key The JIRA iD of the ticket to close.

Returns:

- `True`: the ticket exists and it is closed.
- `False`: the ticket exists and it has not been closed.
- `None`: the ticket does not exist.

CLI Example:

```
salt '*' jira.issue_closed NE-123
```

21.9.184 salt.modules.junos

Module to interact with Junos devices.

maturity new

dependencies junos-eznc, jxmlease

Note: Those who wish to use junos-eznc (PyEZ) version >= 2.1.0, must use the latest salt code from github until the next release.

Refer to *`junos`* for information on connecting to junos proxy.

`salt.modules.junos.`**`cli`**(*command=None, **kwargs*)

Executes the CLI commands and returns the output in specified format. (default is text) The output can also be stored in a file.

command (required) The command to execute on the Junos CLI

format [text] Format in which to get the CLI output (either `text` or `xml`)

dev_timeout [30] The NETCONF RPC timeout (in seconds)

dest Destination file where the RPC output is stored. Note that the file will be stored on the proxy minion. To push the files to the master use *`cp.push`*.

CLI Examples:

```
salt 'device_name' junos.cli 'show system commit'
salt 'device_name' junos.cli 'show version' dev_timeout=40
salt 'device_name' junos.cli 'show system alarms' format=xml dest=/home/user/cli_
↪output.txt
```

`salt.modules.junos.`**`commit`**`(`***kwargs*`)`

To commit the changes loaded in the candidate configuration.

dev_timeout [30] The NETCONF RPC timeout (in seconds)

comment Provide a comment for the commit

confirm Provide time in minutes for commit confirmation. If this option is specified, the commit will be rolled back in the specified amount of time unless the commit is confirmed.

sync [False] When `True`, on dual control plane systems, requests that the candidate configuration on one control plane be copied to the other control plane, checked for correct syntax, and committed on both Routing Engines.

force_sync [False] When `True`, on dual control plane systems, force the candidate configuration on one control plane to be copied to the other control plane.

full When `True`, requires all the daemons to check and evaluate the new configuration.

detail When `True`, return commit detail

CLI Examples:

```
salt 'device_name' junos.commit comment='Commiting via saltstack' detail=True
salt 'device_name' junos.commit dev_timeout=60 confirm=10
salt 'device_name' junos.commit sync=True dev_timeout=90
```

`salt.modules.junos.`**`commit_check`**`()`

Perform a commit check on the configuration

CLI Example:

```
salt 'device_name' junos.commit_check
```

`salt.modules.junos.`**`diff`**`(`***kwargs*`)`

Returns the difference between the candidate and the current configuration

id [0] The rollback ID value (0-49)

CLI Example:

```
salt 'device_name' junos.diff 3
```

`salt.modules.junos.`**`facts`**`()`

Displays the facts gathered during the connection. These facts are also stored in Salt grains.

CLI Example:

```
salt 'device_name' junos.facts
```

`salt.modules.junos.`**`facts_refresh`**`()`

Reload the facts dictionary from the device. Usually only needed if, the device configuration is changed by some other actor. This function will also refresh the facts stored in the salt grains.

CLI Example:

```
salt 'device_name' junos.facts_refresh
```

`salt.modules.junos.`**`file_copy`**(*src=None*, *dest=None*)

Copies the file from the local device to the junos device

src The source path where the file is kept.

dest The destination path on the where the file will be copied

CLI Example:

```
salt 'device_name' junos.file_copy /home/m2/info.txt info_copy.txt
```

`salt.modules.junos.`**`install_config`**(*path=None*, ***kwargs*)

Installs the given configuration file into the candidate configuration. Commits the changes if the commit checks or throws an error.

path (required) Path where the configuration/template file is present. If the file has a `.conf` extension, the content is treated as text format. If the file has a `.xml` extension, the content is treated as XML format. If the file has a `.set` extension, the content is treated as Junos OS `set` commands.

mode [exclusive] The mode in which the configuration is locked. Can be one of `private`, `dynamic`, `batch`, `exclusive`.

dev_timeout [30] Set NETCONF RPC timeout. Can be used for commands which take a while to execute.

overwrite [False] Set to `True` if you want this file is to completely replace the configuration file.

replace [False] Specify whether the configuration file uses `replace:` statements. If `True`, only those statements under the `replace` tag will be changed.

format Determines the format of the contents

update [False] Compare a complete loaded configuration against the candidate configuration. For each hierarchy level or configuration object that is different in the two configurations, the version in the loaded configuration replaces the version in the candidate configuration. When the configuration is later committed, only system processes that are affected by the changed configuration elements parse the new configuration. This action is supported from PyEZ 2.1.

comment Provide a comment for the commit

confirm Provide time in minutes for commit confirmation. If this option is specified, the commit will be rolled back in the specified amount of time unless the commit is confirmed.

diffs_file Path to the file where the diff (difference in old configuration and the committed configuration) will be stored. Note that the file will be stored on the proxy minion. To push the files to the master use *`cp.push`*.

template_vars Variables to be passed into the template processing engine in addition to those present in pillar, the minion configuration, grains, etc. You may reference these variables in your template like so:

```
{{ template_vars["var_name"] }}
```

CLI Examples:

```
salt 'device_name' junos.install_config 'salt://production/network/routers/config.
↪set'
salt 'device_name' junos.install_config 'salt://templates/replace_config.conf'
↪replace=True comment='Committed via SaltStack'
```

(continues on next page)

(continued from previous page)

```
salt 'device_name' junos.install_config 'salt://my_new_configuration.conf' dev_
↪timeout=300 diffs_file='/salt/confs/old_config.conf' overwrite=True
salt 'device_name' junos.install_config 'salt://syslog_template.conf' template_
↪vars='{"syslog_host": "10.180.222.7"}'
```

`salt.modules.junos.`**`install_os`**(*path=None*, ***kwargs*)

Installs the given image on the device. After the installation is complete the device is rebooted, if reboot=True is given as a keyworded argument.

path (required) Path where the image file is present on the proxy minion

remote_path : If the value of path is a file path on the local (Salt host's) filesystem, then the image is copied from the local filesystem to the :remote_path: directory on the target Junos device. The default is `/var/tmp`. If the value of :path: or is a URL, then the value of :remote_path: is unused.

dev_timeout [30] The NETCONF RPC timeout (in seconds). This argument was added since most of the time the "package add" RPC takes a significant amount of time. The default RPC timeout is 30 seconds. So this :timeout: value will be used in the context of the SW installation process. Defaults to 30 minutes (30*60=1800)

reboot [False] Whether to reboot after installation

no_copy [False] If `True` the software package will not be SCP'd to the device

bool validate: When `True` this method will perform a config validation against the new image

bool issu: When `True` allows unified in-service software upgrade (ISSU) feature enables you to upgrade between two different Junos OS releases with no disruption on the control plane and with minimal disruption of traffic.

bool nssu: When `True` allows nonstop software upgrade (NSSU) enables you to upgrade the software running on a Juniper Networks EX Series Virtual Chassis or a Juniper Networks EX Series Ethernet Switch with redundant Routing Engines with a single command and minimal disruption to network traffic.

CLI Examples:

```
salt 'device_name' junos.install_os 'salt://images/junos_image.tgz' reboot=True
salt 'device_name' junos.install_os 'salt://junos_16_1.tgz' dev_timeout=300
```

`salt.modules.junos.`**`load`**(*path=None*, ***kwargs*)

Loads the configuration from the file provided onto the device.

path (required) Path where the configuration/template file is present. If the file has a `.conf` extension, the content is treated as text format. If the file has a `.xml` extension, the content is treated as XML format. If the file has a `.set` extension, the content is treated as Junos OS `set` commands.

overwrite [False] Set to `True` if you want this file is to completely replace the configuration file.

replace [False] Specify whether the configuration file uses `replace:` statements. If `True`, only those statements under the `replace` tag will be changed.

format Determines the format of the contents

update [False] Compare a complete loaded configuration against the candidate configuration. For each hierarchy level or configuration object that is different in the two configurations, the version in the loaded configuration replaces the version in the candidate configuration. When the configuration is later committed, only system processes that are affected by the changed configuration elements parse the new configuration. This action is supported from PyEZ 2.1.

template_vars Variables to be passed into the template processing engine in addition to those present in pillar, the minion configuration, grains, etc. You may reference these variables in your template like so:

```
{{ template_vars["var_name"] }}
```

CLI Examples:

```
salt 'device_name' junos.load 'salt://production/network/routers/config.set'

salt 'device_name' junos.load 'salt://templates/replace_config.conf' replace=True

salt 'device_name' junos.load 'salt://my_new_configuration.conf' overwrite=True

salt 'device_name' junos.load 'salt://syslog_template.conf' template_vars='{
↪"syslog_host": "10.180.222.7"}'
```

`salt.modules.junos.`**`lock`**`()`

Attempts an exclusive lock on the candidate configuration. This is a non-blocking call.

Note: When locking, it is important to remember to call *`junos.unlock`* once finished. If locking during orchestration, remember to include a step in the orchestration job to unlock.

CLI Example:

```
salt 'device_name' junos.lock
```

`salt.modules.junos.`**`ping`**`(`*dest_ip=None*, ***kwargs*`)`

Send a ping RPC to a device

dest_ip The IP of the device to ping

dev_timeout [30] The NETCONF RPC timeout (in seconds)

rapid [False] When `True`, executes ping at 100pps instead of 1pps

ttl Maximum number of IP routers (IP hops) allowed between source and destination

routing_instance Name of the routing instance to use to send the ping

interface Interface used to send traffic

count [5] Number of packets to send

CLI Examples:

```
salt 'device_name' junos.ping '8.8.8.8' count=5
salt 'device_name' junos.ping '8.8.8.8' ttl=1 rapid=True
```

`salt.modules.junos.`**`rollback`**`(`***kwargs*`)`

Roll back the last committed configuration changes and commit

id [0] The rollback ID value (0-49)

dev_timeout [30] The NETCONF RPC timeout (in seconds)

comment Provide a comment for the commit

confirm Provide time in minutes for commit confirmation. If this option is specified, the commit will be rolled back in the specified amount of time unless the commit is confirmed.

diffs_file Path to the file where the diff (difference in old configuration and the committed configuration) will be stored. Note that the file will be stored on the proxy minion. To push the files to the master use *`cp.push`*.

CLI Example:

```
salt 'device_name' junos.rollback 10
```

`salt.modules.junos.`**`rpc`**(*cmd=None*, *dest=None*, ***kwargs*)

This function executes the RPC provided as arguments on the junos device. The returned data can be stored in a file.

cmd The RPC to be executed

dest Destination file where the RPC output is stored. Note that the file will be stored on the proxy minion. To push the files to the master use *`cp.push`*.

format [xml] The format in which the RPC reply is received from the device

dev_timeout [30] The NETCONF RPC timeout (in seconds)

filter Used with the `get-config` RPC to get specific configuration

terse [False] Amount of information you want

interface_name Name of the interface to query

CLI Example:

```
salt 'device' junos.rpc get_config /var/log/config.txt format=text filter='
↪<configuration><system/></configuration>'
salt 'device' junos.rpc get-interface-information /home/user/interface.xml
↪interface_name='lo0' terse=True
salt 'device' junos.rpc get-chassis-inventory
```

`salt.modules.junos.`**`set_hostname`**(*hostname=None*, ***kwargs*)

Set the device's hostname

hostname The name to be set

dev_timeout [30] The NETCONF RPC timeout (in seconds)

comment Provide a comment to the commit

confirm Provide time in minutes for commit confirmation. If this option is specified, the commit will be rolled back in the specified amount of time unless the commit is confirmed.

CLI Example:

```
salt 'device_name' junos.set_hostname salt-device
```

`salt.modules.junos.`**`shutdown`**(***kwargs*)

Shut down (power off) or reboot a device running Junos OS. This includes all Routing Engines in a Virtual Chassis or a dual Routing Engine system.

> **Note:** One of `shutdown` or `reboot` must be set to `True` or no action will be taken.

shutdown [False] Set this to `True` if you want to shutdown the machine. This is a safety mechanism so that the user does not accidentally shutdown the junos device.

reboot [False] If `True`, reboot instead of shutting down

at Used when rebooting, to specify the date and time the reboot should take place. The value of this option must match the JunOS CLI reboot syntax.

in_min Used when shutting down. Specify the delay (in minutes) before the device will be shut down.

CLI Examples:

```
salt 'device_name' junos.shutdown reboot=True
salt 'device_name' junos.shutdown shutdown=True in_min=10
salt 'device_name' junos.shutdown shutdown=True
```

`salt.modules.junos.`**`timeoutDecorator`**`(`*function*`)`

`salt.modules.junos.`**`unlock`**`()`

Unlocks the candidate configuration.

CLI Example:

```
salt 'device_name' junos.unlock
```

`salt.modules.junos.`**`zeroize`**`()`

Resets the device to default factory settings

CLI Example:

```
salt 'device_name' junos.zeroize
```

21.9.185 salt.modules.k8s

Salt module to manage Kubernetes cluster

New in version 2016.3.0.

Roadmap:

- Add creation of K8S objects (pod, rc, service, ...)
- Add replace of K8S objects (pod, rc, service, ...)
- Add deletion of K8S objects (pod, rc, service, ...)
- Add rolling update
- Add (auto)scalling

`salt.modules.k8s.`**`create_namespace`**`(`*name*, *apiserver_url=None*`)`

New in version 2016.3.0.

Create kubernetes namespace from the name, similar to the functionality added to kubectl since v.1.2.0: .. code-block:: bash

> kubectl create namespaces namespace-name

CLI Example:

```
salt '*' k8s.create_namespace namespace_name

salt '*' k8s.create_namespace namespace_name http://kube-master.cluster.local
```

`salt.modules.k8s.`**`create_secret`**`(`*namespace*, *name*, *sources*, *apiserver_url=None*, *force=False*, *update=False*, *saltenv='base'*`)`

New in version 2016.3.0.

Create k8s secrets in the defined namespace from the list of files

CLI Example:

```
salt '*' k8s.create_secret namespace_name secret_name sources

salt '*' k8s.create_secret namespace_name secret_name sources
http://kube-master.cluster.local
```

sources are either dictionary of {name: path, name1: path} pairs or array of strings defining paths.

Example of paths array:

```
```

['/full/path/filename', "file:///full/path/filename", "salt://secret/storage/file.txt", "http://user:password@securesite.com/secret-file.json"]

Example of dictionaries:

```
```

{"nameit": '/full/path/fiename', name2: "salt://secret/storage/file.txt"}

optional parameters accepted:

update=[false] default value is false if set to false, and secret is already present on the cluster - warning will be returned and no changes to the secret will be done. In case it is set to "true" and secret is present but data is differ - secret will be updated.

force=[true] default value is true if the to False, secret will not be created in case one of the files is not valid kubernetes secret. e.g. capital letters in secret name or _ in case force is set to True, wrong files will be skipped but secret will be created any way.

saltenv=['base'] default value is base in case 'salt://' path is used, this parameter can change the visibility of files

`salt.modules.k8s.`**`delete_secret`**(*namespace, name, apiserver_url=None, force=True*)

New in version 2016.3.0.

Delete kubernetes secret in the defined namespace. Namespace is the mandatory parameter as well as name.

CLI Example:

```
salt '*' k8s.delete_secret namespace_name secret_name

salt '*' k8s.delete_secret namespace_name secret_name http://kube-master.cluster.
↪local
```

`salt.modules.k8s.`**`get_labels`**(*node=None, apiserver_url=None*)

New in version 2016.3.0.

Get labels from the current node

CLI Example:

```
salt '*' k8s.get_labels
salt '*' k8s.get_labels kube-node.cluster.local http://kube-master.cluster.local
```

`salt.modules.k8s.`**`get_namespaces`**(*namespace=", apiserver_url=None*)

New in version 2016.3.0.

Get one or all kubernetes namespaces.

If namespace parameter is omitted, all namespaces will be returned back to user, similar to following kubectl example:

```
kubectl get namespaces -o json
```

In case namespace is set by user, the output will be similar to the one from kubectl:

```
kubectl get namespaces namespace_name -o json
```

CLI Example:

```
salt '*' k8s.get_namespaces
salt '*' k8s.get_namespaces namespace_name http://kube-master.cluster.local
```

`salt.modules.k8s.`**`get_secrets`**(*namespace, name=", apiserver_url=None, decode=False, brief=False*)

Get k8s namespaces

CLI Example:

```
salt '*' k8s.get_secrets namespace_name
salt '*' k8s.get_secrets namespace_name secret_name http://kube-master.cluster.
↪local
```

`salt.modules.k8s.`**`label_absent`**(*name, node=None, apiserver_url=None*)

New in version 2016.3.0.

Delete label to the current node

CLI Example:

```
salt '*' k8s.label_absent hw/disktype
salt '*' k8s.label_absent hw/disktype kube-node.cluster.local http://kube-master.
↪cluster.local
```

`salt.modules.k8s.`**`label_folder_absent`**(*name, node=None, apiserver_url=None*)

New in version 2016.3.0.

Delete label folder to the current node

CLI Example:

```
salt '*' k8s.label_folder_absent hw
salt '*' k8s.label_folder_absent hw/ kube-node.cluster.local http://kube-master.
↪cluster.local
```

`salt.modules.k8s.`**`label_present`**(*name, value, node=None, apiserver_url=None*)

New in version 2016.3.0.

Set label to the current node

CLI Example:

```
salt '*' k8s.label_present hw/disktype ssd

salt '*' k8s.label_present hw/disktype ssd kube-node.cluster.local http://kube-
↪master.cluster.local
```

`salt.modules.k8s.`**`update_secret`**(*namespace, name, sources, apiserver_url=None, force=True, saltenv='base'*)

New in version 2016.3.0.

alias to k8s.create_secret with update=true

CLI Example:

```
salt '*' k8s.update_secret namespace_name secret_name sources [apiserver_url]↵
↪[force=true] [update=false] [saltenv='base']
```

sources are either dictionary of {name: path, name1: path} pairs or array of strings defining paths.

Example of paths array:

```
```

['/full/path/filename', "file:///full/path/filename", "salt://secret/storage/file.txt", "http://user:password@securesite.com/secret-file.json"]

Example of dictionaries:

```
```

{"nameit": '/full/path/fiename', name2: "salt://secret/storage/file.txt"}

optional parameters accepted:

force=[true] default value is true if the to False, secret will not be created in case one of the files is not valid kubernetes secret. e.g. capital letters in secret name or _ in case force is set to True, wrong files will be skipped but secret will be created any way.

saltenv=['base'] default value is base in case 'salt://' path is used, this parameter can change the visibility of files

21.9.186 salt.modules.kapacitor module

Kapacitor execution module.

configuration This module accepts connection configuration details either as parameters or as configuration settings in /etc/salt/minion on the relevant minions:

```
kapacitor.host: 'localhost'
kapacitor.port: 9092
```

New in version 2016.11.0.

Also protocol and SSL settings could be configured:

```
kapacitor.unsafe_ssl: 'false'
kapacitor.protocol: 'http'
```

New in version 2019.2.0.

This data can also be passed into pillar. Options passed into opts will overwrite options passed into pillar.

`salt.modules.kapacitor.`**`define_task`**(*name*, *tick_script*, *task_type='stream'*, *database=None*, *retention_policy='default'*, *dbrps=None*)

Define a task. Serves as both create/update.

name Name of the task.

tick_script Path to the TICK script for the task. Can be a salt:// source.

task_type Task type. Defaults to 'stream'

dbrps A list of databases and retention policies in "dbname"."rpname" format to fetch data from. For backward compatibility, the value of 'database' and 'retention_policy' will be merged as part of dbrps.

New in version 2019.2.0.

database Which database to fetch data from.

retention_policy Which retention policy to fetch data from. Defaults to 'default'.

CLI Example:

```
salt '*' kapacitor.define_task cpu salt://kapacitor/cpu.tick database=telegraf
```

salt.modules.kapacitor.**delete_task**(*name*)

Delete a kapacitor task.

name Name of the task to delete.

CLI Example:

```
salt '*' kapacitor.delete_task cpu
```

salt.modules.kapacitor.**disable_task**(*name*)

Disable a kapacitor task.

name Name of the task to disable.

CLI Example:

```
salt '*' kapacitor.disable_task cpu
```

salt.modules.kapacitor.**enable_task**(*name*)

Enable a kapacitor task.

name Name of the task to enable.

CLI Example:

```
salt '*' kapacitor.enable_task cpu
```

salt.modules.kapacitor.**get_task**(*name*)

Get a dict of data on a task.

name Name of the task to get information about.

CLI Example:

```
salt '*' kapacitor.get_task cpu
```

salt.modules.kapacitor.**version**()

Get the kapacitor version.

21.9.187 salt.modules.kerberos

Manage Kerberos KDC

configuration In order to manage your KDC you will need to generate a keytab that can authenticate without requiring a password.

```
# ktadd -k /root/secure.keytab kadmin/admin kadmin/changepw
```

On the KDC minion you will need to add the following to the minion configuration file so Salt knows what keytab to use and what principal to authenticate as.

```
auth_keytab: /root/auth.keytab
auth_principal: kadmin/admin
```

`salt.modules.kerberos.`**`create_keytab`**(*name*, *keytab*, *enctypes=None*)

Create keytab

CLI Example:

```
salt 'kdc.example.com' kerberos.create_keytab host/host1.example.com host1.
↪example.com.keytab
```

`salt.modules.kerberos.`**`create_principal`**(*name*, *enctypes=None*)

Create Principal

CLI Example:

```
salt 'kdc.example.com' kerberos.create_principal host/example.com
```

`salt.modules.kerberos.`**`delete_principal`**(*name*)

Delete Principal

CLI Example:

```
salt 'kdc.example.com' kerberos.delete_principal host/example.com@EXAMPLE.COM
```

`salt.modules.kerberos.`**`get_policy`**(*name*)

Get policy details

CLI Example:

```
salt 'kdc.example.com' kerberos.get_policy my_policy
```

`salt.modules.kerberos.`**`get_principal`**(*name*)

Get princial details

CLI Example:

```
salt 'kdc.example.com' kerberos.get_principal root/admin
```

`salt.modules.kerberos.`**`get_privs`**()

Current privileges

CLI Example:

```
salt 'kdc.example.com' kerberos.get_privs
```

`salt.modules.kerberos.`**`list_policies`**()

List policies

CLI Example:

```
salt 'kdc.example.com' kerberos.list_policies
```

`salt.modules.kerberos.`**`list_principals`**()

Get all principals

CLI Example:

```
salt 'kde.example.com' kerberos.list_principals
```

21.9.188 salt.modules.kernelpkg_linux_apt

Manage Linux kernel packages on APT-based systems

`salt.modules.kernelpkg_linux_apt.`**`active`**`()`

Return the version of the running kernel.

CLI Example:

```
salt '*' kernelpkg.active
```

`salt.modules.kernelpkg_linux_apt.`**`cleanup`**`(`*keep_latest=True*`)`

Remove all unused kernel packages from the system.

keep_latest [True] In the event that the active kernel is not the latest one installed, setting this to True will retain the latest kernel package, in addition to the active one. If False, all kernel packages other than the active one will be removed.

CLI Example:

```
salt '*' kernelpkg.cleanup
```

`salt.modules.kernelpkg_linux_apt.`**`latest_available`**`()`

Return the version of the latest kernel from the package repositories.

CLI Example:

```
salt '*' kernelpkg.latest_available
```

`salt.modules.kernelpkg_linux_apt.`**`latest_installed`**`()`

Return the version of the latest installed kernel.

CLI Example:

```
salt '*' kernelpkg.latest_installed
```

Note: This function may not return the same value as `active()` if a new kernel has been installed and the system has not yet been rebooted. The `needs_reboot()` function exists to detect this condition.

`salt.modules.kernelpkg_linux_apt.`**`list_installed`**`()`

Return a list of all installed kernels.

CLI Example:

```
salt '*' kernelpkg.list_installed
```

`salt.modules.kernelpkg_linux_apt.`**`needs_reboot`**`()`

Detect if a new kernel version has been installed but is not running. Returns True if a new kernel is installed, False otherwise.

CLI Example:

```
salt '*' kernelpkg.needs_reboot
```

`salt.modules.kernelpkg_linux_apt.`**`remove`**`(`*release*`)`

Remove a specific version of the kernel.

release The release number of an installed kernel. This must be the entire release number as returned by *`list_installed()`*, not the package name.

CLI Example:

```
salt '*' kernelpkg.remove 4.4.0-70-generic
```

`salt.modules.kernelpkg_linux_apt.`**`upgrade`**`(`*reboot=False*, *at_time=None*`)`

Upgrade the kernel and optionally reboot the system.

reboot [False] Request a reboot if a new kernel is available.

at_time [immediate] Schedule the reboot at some point in the future. This argument is ignored if `reboot=False`. See *`reboot()`* for more details on this argument.

CLI Example:

```
salt '*' kernelpkg.upgrade
salt '*' kernelpkg.upgrade reboot=True at_time=1
```

Note: An immediate reboot often shuts down the system before the minion has a chance to return, resulting in errors. A minimal delay (1 minute) is useful to ensure the result is delivered to the master.

`salt.modules.kernelpkg_linux_apt.`**`upgrade_available`**`()`

Detect if a new kernel version is available in the repositories. Returns True if a new kernel is available, False otherwise.

CLI Example:

```
salt '*' kernelpkg.upgrade_available
```

21.9.189 salt.modules.kernelpkg_linux_yum

Manage Linux kernel packages on YUM-based systems

`salt.modules.kernelpkg_linux_yum.`**`active`**`()`

Return the version of the running kernel.

CLI Example:

```
salt '*' kernelpkg.active
```

`salt.modules.kernelpkg_linux_yum.`**`cleanup`**`(`*keep_latest=True*`)`

Remove all unused kernel packages from the system.

keep_latest [True] In the event that the active kernel is not the latest one installed, setting this to True will retain the latest kernel package, in addition to the active one. If False, all kernel packages other than the active one will be removed.

CLI Example:

```
salt '*' kernelpkg.cleanup
```

`salt.modules.kernelpkg_linux_yum.`**`latest_available`**`()`

Return the version of the latest kernel from the package repositories.

CLI Example:

```
salt '*' kernelpkg.latest_available
```

`salt.modules.kernelpkg_linux_yum.`**`latest_installed`**`()`

Return the version of the latest installed kernel.

CLI Example:

```
salt '*' kernelpkg.latest_installed
```

Note: This function may not return the same value as `active()` if a new kernel has been installed and the system has not yet been rebooted. The `needs_reboot()` function exists to detect this condition.

`salt.modules.kernelpkg_linux_yum.`**`list_installed`**`()`

Return a list of all installed kernels.

CLI Example:

```
salt '*' kernelpkg.list_installed
```

`salt.modules.kernelpkg_linux_yum.`**`needs_reboot`**`()`

Detect if a new kernel version has been installed but is not running. Returns True if a new kernel is installed, False otherwise.

CLI Example:

```
salt '*' kernelpkg.needs_reboot
```

`salt.modules.kernelpkg_linux_yum.`**`remove`**`(`*release*`)`

Remove a specific version of the kernel.

release The release number of an installed kernel. This must be the entire release number as returned by `list_installed()`, not the package name.

CLI Example:

```
salt '*' kernelpkg.remove 3.10.0-327.el7
```

`salt.modules.kernelpkg_linux_yum.`**`upgrade`**`(`*reboot=False, at_time=None*`)`

Upgrade the kernel and optionally reboot the system.

reboot [False] Request a reboot if a new kernel is available.

at_time [immediate] Schedule the reboot at some point in the future. This argument is ignored if `reboot=False`. See `reboot()` for more details on this argument.

CLI Example:

```
salt '*' kernelpkg.upgrade
salt '*' kernelpkg.upgrade reboot=True at_time=1
```

Note: An immediate reboot often shuts down the system before the minion has a chance to return, resulting in errors. A minimal delay (1 minute) is useful to ensure the result is delivered to the master.

salt.modules.kernelpkg_linux_yum.**upgrade_available**()

Detect if a new kernel version is available in the repositories. Returns True if a new kernel is available, False otherwise.

CLI Example:

```
salt '*' kernelpkg.upgrade_available
```

21.9.190 salt.modules.key

Functions to view the minion's public key information

salt.modules.key.**finger**(*hash_type=None*)

Return the minion's public key fingerprint

hash_type The hash algorithm used to calculate the fingerprint

CLI Example:

```
salt '*' key.finger
```

salt.modules.key.**finger_master**(*hash_type=None*)

Return the fingerprint of the master's public key on the minion.

hash_type The hash algorithm used to calculate the fingerprint

CLI Example:

```
salt '*' key.finger_master
```

21.9.191 salt.modules.keyboard

Module for managing keyboards on supported POSIX-like systems using systemd, or such as Redhat, Debian and Gentoo.

salt.modules.keyboard.**get_sys**()

Get current system keyboard setting

CLI Example:

```
salt '*' keyboard.get_sys
```

salt.modules.keyboard.**get_x**()

Get current X keyboard setting

CLI Example:

```
salt '*' keyboard.get_x
```

salt.modules.keyboard.**set_sys**(*layout*)

Set current system keyboard setting

CLI Example:

```
salt '*' keyboard.set_sys dvorak
```

`salt.modules.keyboard.`**`set_x`**`(`*layout*`)`

Set current X keyboard setting

CLI Example:

```
salt '*' keyboard.set_x dvorak
```

21.9.192 salt.modules.keystone

Module for handling openstack keystone calls.

optdepends

- keystoneclient Python adapter

configuration This module is not usable until the following are specified either in a pillar or in the minion's config file:

```
keystone.user: admin
keystone.password: verybadpass
keystone.tenant: admin
keystone.tenant_id: f80919baedab48ec8931f200c65a50df
keystone.auth_url: 'http://127.0.0.1:5000/v2.0/'
```

OR (for token based authentication)

```
keystone.token: 'ADMIN'
keystone.endpoint: 'http://127.0.0.1:35357/v2.0'
```

If configuration for multiple openstack accounts is required, they can be set up as different configuration profiles. For example:

```
openstack1:
  keystone.user: admin
  keystone.password: verybadpass
  keystone.tenant: admin
  keystone.tenant_id: f80919baedab48ec8931f200c65a50df
  keystone.auth_url: 'http://127.0.0.1:5000/v2.0/'

openstack2:
  keystone.user: admin
  keystone.password: verybadpass
  keystone.tenant: admin
  keystone.tenant_id: f80919baedab48ec8931f200c65a50df
  keystone.auth_url: 'http://127.0.0.2:5000/v2.0/'
```

With this configuration in place, any of the keystone functions can make use of a configuration profile by declaring it explicitly. For example:

```
salt '*' keystone.tenant_list profile=openstack1
```

`salt.modules.keystone.`**`api_version`**`(`*profile=None*, ***connection_args*`)`

Returns the API version derived from endpoint's response.

CLI Example:

```
salt '*' keystone.api_version
```

`salt.modules.keystone.auth(profile=None, **connection_args)`

Set up keystone credentials. Only intended to be used within Keystone-enabled modules.

CLI Example:

```
salt '*' keystone.auth
```

`salt.modules.keystone.ec2_credentials_create(user_id=None, name=None, tenant_id=None, tenant=None, profile=None, **connection_args)`

Create EC2-compatible credentials for user per tenant

CLI Examples:

```
salt '*' keystone.ec2_credentials_create name=admin tenant=admin

salt '*' keystone.ec2_credentials_create         user_
↪id=c965f79c4f864eaaa9c3b41904e67082         tenant_
↪id=722787eb540849158668370dc627ec5f
```

`salt.modules.keystone.ec2_credentials_delete(user_id=None, name=None, access_key=None, profile=None, **connection_args)`

Delete EC2-compatible credentials

CLI Examples:

```
salt '*' keystone.ec2_credentials_delete         860f8c2c38ca4fab989f9bc56a061a64
↪access_key=5f66d2f24f604b8bb9cd28886106f442

salt '*' keystone.ec2_credentials_delete name=admin         access_
↪key=5f66d2f24f604b8bb9cd28886106f442
```

`salt.modules.keystone.ec2_credentials_get(user_id=None, name=None, access=None, profile=None, **connection_args)`

Return ec2_credentials for a user (keystone ec2-credentials-get)

CLI Examples:

```
salt '*' keystone.ec2_credentials_get c965f79c4f864eaaa9c3b41904e67082
↪access=722787eb540849158668370
salt '*' keystone.ec2_credentials_get user_id=c965f79c4f864eaaa9c3b41904e67082
↪access=722787eb540849158668370
salt '*' keystone.ec2_credentials_get name=nova
↪access=722787eb540849158668370dc627ec5f
```

`salt.modules.keystone.ec2_credentials_list(user_id=None, name=None, profile=None, **connection_args)`

Return a list of ec2_credentials for a specific user (keystone ec2-credentials-list)

CLI Examples:

```
salt '*' keystone.ec2_credentials_list 298ce377245c4ec9b70e1c639c89e654
salt '*' keystone.ec2_credentials_list user_id=298ce377245c4ec9b70e1c639c89e654
salt '*' keystone.ec2_credentials_list name=jack
```

`salt.modules.keystone.endpoint_create(service, publicurl=None, internalurl=None, adminurl=None, region=None, profile=None, url=None, interface=None, **connection_args)`

Create an endpoint for an Openstack service

CLI Examples:

```
salt 'v2' keystone.endpoint_create nova 'http://public/url' 'http://internal/url'
↪'http://adminurl/url' region

salt 'v3' keystone.endpoint_create nova url='http://public/url' interface='public
↪' region='RegionOne'
```

`salt.modules.keystone.`**`endpoint_delete`**(*service*, *region=None*, *profile=None*, *interface=None*, ***connection_args*)

Delete endpoints of an Openstack service

CLI Examples:

```
salt 'v2' keystone.endpoint_delete nova [region=RegionOne]

salt 'v3' keystone.endpoint_delete nova interface=admin [region=RegionOne]
```

`salt.modules.keystone.`**`endpoint_get`**(*service*, *region=None*, *profile=None*, *interface=None*, ***connection_args*)

Return a specific endpoint (keystone endpoint-get)

CLI Example:

```
salt 'v2' keystone.endpoint_get nova [region=RegionOne]

salt 'v3' keystone.endpoint_get nova interface=admin [region=RegionOne]
```

`salt.modules.keystone.`**`endpoint_list`**(*profile=None*, ***connection_args*)

Return a list of available endpoints (keystone endpoints-list)

CLI Example:

```
salt '*' keystone.endpoint_list
```

`salt.modules.keystone.`**`project_create`**(*name*, *domain*, *description=None*, *enabled=True*, *profile=None*, ***connection_args*)

Create a keystone project. Overrides keystone tenant_create form api V2. For keystone api V3.

New in version 2016.11.0.

name The project name, which must be unique within the owning domain.

domain The domain name.

description The project description.

enabled Enables or disables the project.

profile Configuration profile - if configuration for multiple openstack accounts required.

CLI Examples:

```
salt '*' keystone.project_create nova default description='Nova Compute Project'
salt '*' keystone.project_create test default enabled=False
```

`salt.modules.keystone.`**`project_delete`**(*project_id=None*, *name=None*, *profile=None*, ***connection_args*)

Delete a project (keystone project-delete). Overrides keystone tenant-delete form api V2. For keystone api V3 only.

New in version 2016.11.0.

project_id The project id.

name The project name.

profile Configuration profile - if configuration for multiple openstack accounts required.

CLI Examples:

```
salt '*' keystone.project_delete c965f79c4f864eaaa9c3b41904e67082
salt '*' keystone.project_delete project_id=c965f79c4f864eaaa9c3b41904e67082
salt '*' keystone.project_delete name=demo
```

`salt.modules.keystone.`**`project_get`**(*project_id=None, name=None, profile=None, **connection_args*)

Return a specific projects (keystone project-get) Overrides keystone tenant-get form api V2. For keystone api V3 only.

New in version 2016.11.0.

project_id The project id.

name The project name.

profile Configuration profile - if configuration for multiple openstack accounts required.

CLI Examples:

```
salt '*' keystone.project_get c965f79c4f864eaaa9c3b41904e67082
salt '*' keystone.project_get project_id=c965f79c4f864eaaa9c3b41904e67082
salt '*' keystone.project_get name=nova
```

`salt.modules.keystone.`**`project_list`**(*profile=None, **connection_args*)

Return a list of available projects (keystone projects-list). Overrides keystone tenants-list form api V2. For keystone api V3 only.

New in version 2016.11.0.

profile Configuration profile - if configuration for multiple openstack accounts required.

CLI Example:

```
salt '*' keystone.project_list
```

`salt.modules.keystone.`**`project_update`**(*project_id=None, name=None, description=None, enabled=None, profile=None, **connection_args*)

Update a tenant's information (keystone project-update) The following fields may be updated: name, description, enabled. Can only update name if targeting by ID

Overrides keystone tenant_update form api V2. For keystone api V3 only.

New in version 2016.11.0.

project_id The project id.

name The project name, which must be unique within the owning domain.

description The project description.

enabled Enables or disables the project.

profile Configuration profile - if configuration for multiple openstack accounts required.

CLI Examples:

```
salt '*' keystone.project_update name=admin enabled=True
salt '*' keystone.project_update c965f79c4f864eaaa9c3b41904e67082 name=admin␣
↪email=admin@domain.com
```

salt.modules.keystone.**role_create**(*name*, *profile=None*, ***connection_args*)
Create a named role.

CLI Example:

```
salt '*' keystone.role_create admin
```

salt.modules.keystone.**role_delete**(*role_id=None*, *name=None*, *profile=None*, ***connection_args*)
Delete a role (keystone role-delete)

CLI Examples:

```
salt '*' keystone.role_delete c965f79c4f864eaaa9c3b41904e67082
salt '*' keystone.role_delete role_id=c965f79c4f864eaaa9c3b41904e67082
salt '*' keystone.role_delete name=admin
```

salt.modules.keystone.**role_get**(*role_id=None*, *name=None*, *profile=None*, ***connection_args*)
Return a specific roles (keystone role-get)

CLI Examples:

```
salt '*' keystone.role_get c965f79c4f864eaaa9c3b41904e67082
salt '*' keystone.role_get role_id=c965f79c4f864eaaa9c3b41904e67082
salt '*' keystone.role_get name=nova
```

salt.modules.keystone.**role_list**(*profile=None*, ***connection_args*)
Return a list of available roles (keystone role-list)

CLI Example:

```
salt '*' keystone.role_list
```

salt.modules.keystone.**service_create**(*name*, *service_type*, *description=None*, *profile=None*, ***connection_args*)
Add service to Keystone service catalog

CLI Examples:

```
salt '*' keystone.service_create nova compute 'OpenStack Compute Service'
```

salt.modules.keystone.**service_delete**(*service_id=None*, *name=None*, *profile=None*, ***connection_args*)
Delete a service from Keystone service catalog

CLI Examples:

```
salt '*' keystone.service_delete c965f79c4f864eaaa9c3b41904e67082
salt '*' keystone.service_delete name=nova
```

salt.modules.keystone.**service_get**(*service_id=None*, *name=None*, *profile=None*, ***connection_args*)
Return a specific services (keystone service-get)

CLI Examples:

```
salt '*' keystone.service_get c965f79c4f864eaaa9c3b41904e67082
salt '*' keystone.service_get service_id=c965f79c4f864eaaa9c3b41904e67082
salt '*' keystone.service_get name=nova
```

`salt.modules.keystone.`**`service_list`**(*profile=None*, ***connection_args*)
Return a list of available services (keystone services-list)

CLI Example:

```
salt '*' keystone.service_list
```

`salt.modules.keystone.`**`tenant_create`**(*name*, *description=None*, *enabled=True*, *profile=None*, ***connection_args*)
Create a keystone tenant

CLI Examples:

```
salt '*' keystone.tenant_create nova description='nova tenant'
salt '*' keystone.tenant_create test enabled=False
```

`salt.modules.keystone.`**`tenant_delete`**(*tenant_id=None*, *name=None*, *profile=None*, ***connection_args*)
Delete a tenant (keystone tenant-delete)

CLI Examples:

```
salt '*' keystone.tenant_delete c965f79c4f864eaaa9c3b41904e67082
salt '*' keystone.tenant_delete tenant_id=c965f79c4f864eaaa9c3b41904e67082
salt '*' keystone.tenant_delete name=demo
```

`salt.modules.keystone.`**`tenant_get`**(*tenant_id=None*, *name=None*, *profile=None*, ***connection_args*)
Return a specific tenants (keystone tenant-get)

CLI Examples:

```
salt '*' keystone.tenant_get c965f79c4f864eaaa9c3b41904e67082
salt '*' keystone.tenant_get tenant_id=c965f79c4f864eaaa9c3b41904e67082
salt '*' keystone.tenant_get name=nova
```

`salt.modules.keystone.`**`tenant_list`**(*profile=None*, ***connection_args*)
Return a list of available tenants (keystone tenants-list)

CLI Example:

```
salt '*' keystone.tenant_list
```

`salt.modules.keystone.`**`tenant_update`**(*tenant_id=None*, *name=None*, *description=None*, *enabled=None*, *profile=None*, ***connection_args*)
Update a tenant's information (keystone tenant-update) The following fields may be updated: name, description, enabled. Can only update name if targeting by ID

CLI Examples:

```
salt '*' keystone.tenant_update name=admin enabled=True
salt '*' keystone.tenant_update c965f79c4f864eaaa9c3b41904e67082 name=admin␣
↪email=admin@domain.com
```

`salt.modules.keystone.`**`token_get`**(*profile=None*, ***connection_args*)
Return the configured tokens (keystone token-get)

CLI Example:

```
salt '*' keystone.token_get c965f79c4f864eaaa9c3b41904e67082
```

`salt.modules.keystone.`**`user_create`**(*name*, *password*, *email*, *tenant_id=None*, *enabled=True*, *profile=None*, *project_id=None*, *description=None*, ***connection_args*)

Create a user (keystone user-create)

CLI Examples:

```
salt '*' keystone.user_create name=jack password=zero email=jack@halloweentown.
↪org         tenant_id=a28a7b5a999a455f84b1f5210264375e enabled=True
```

`salt.modules.keystone.`**`user_delete`**(*user_id=None*, *name=None*, *profile=None*, ***connection_args*)

Delete a user (keystone user-delete)

CLI Examples:

```
salt '*' keystone.user_delete c965f79c4f864eaaa9c3b41904e67082
salt '*' keystone.user_delete user_id=c965f79c4f864eaaa9c3b41904e67082
salt '*' keystone.user_delete name=nova
```

`salt.modules.keystone.`**`user_get`**(*user_id=None*, *name=None*, *profile=None*, ***connection_args*)

Return a specific users (keystone user-get)

CLI Examples:

```
salt '*' keystone.user_get c965f79c4f864eaaa9c3b41904e67082
salt '*' keystone.user_get user_id=c965f79c4f864eaaa9c3b41904e67082
salt '*' keystone.user_get name=nova
```

`salt.modules.keystone.`**`user_list`**(*profile=None*, ***connection_args*)

Return a list of available users (keystone user-list)

CLI Example:

```
salt '*' keystone.user_list
```

`salt.modules.keystone.`**`user_password_update`**(*user_id=None*, *name=None*, *password=None*, *profile=None*, ***connection_args*)

Update a user's password (keystone user-password-update)

CLI Examples:

```
salt '*' keystone.user_password_update c965f79c4f864eaaa9c3b41904e67082
↪password=12345
salt '*' keystone.user_password_update user_id=c965f79c4f864eaaa9c3b41904e67082
↪password=12345
salt '*' keystone.user_password_update name=nova password=12345
```

`salt.modules.keystone.`**`user_role_add`**(*user_id=None*, *user=None*, *tenant_id=None*, *tenant=None*, *role_id=None*, *role=None*, *profile=None*, *project_id=None*, *project_name=None*, ***connection_args*)

Add role for user in tenant (keystone user-role-add)

CLI Examples:

```
salt '*' keystone.user_role_add user_id=298ce377245c4ec9b70e1c639c89e654 tenant_
↪id=7167a092ece84bae8cead4bf9d15bb3b role_id=ce377245c4ec9b70e1c639c89e8cead4
salt '*' keystone.user_role_add user=admin tenant=admin role=admin
```

`salt.modules.keystone.`**`user_role_list`**(*user_id=None, tenant_id=None, user_name=None, tenant_name=None, profile=None, project_id=None, project_name=None, **connection_args*)

Return a list of available user_roles (keystone user-roles-list)

CLI Examples:

```
salt '*' keystone.user_role_list user_id=298ce377245c4ec9b70e1c639c89e654 tenant_
↪id=7167a092ece84bae8cead4bf9d15bb3b
salt '*' keystone.user_role_list user_name=admin tenant_name=admin
```

`salt.modules.keystone.`**`user_role_remove`**(*user_id=None, user=None, tenant_id=None, tenant=None, role_id=None, role=None, profile=None, project_id=None, project_name=None, **connection_args*)

Remove role for user in tenant (keystone user-role-remove)

CLI Examples:

```
salt '*' keystone.user_role_remove user_id=298ce377245c4ec9b70e1c639c89e654␣
↪tenant_id=7167a092ece84bae8cead4bf9d15bb3b role_
↪id=ce377245c4ec9b70e1c639c89e8cead4
salt '*' keystone.user_role_remove user=admin tenant=admin role=admin
```

`salt.modules.keystone.`**`user_update`**(*user_id=None, name=None, email=None, enabled=None, tenant=None, profile=None, project=None, description=None, **connection_args*)

Update a user's information (keystone user-update) The following fields may be updated: name, email, enabled, tenant. Because the name is one of the fields, a valid user id is required.

CLI Examples:

```
salt '*' keystone.user_update user_id=c965f79c4f864eaaa9c3b41904e67082␣
↪name=newname
salt '*' keystone.user_update c965f79c4f864eaaa9c3b41904e67082 name=newname␣
↪email=newemail@domain.com
```

`salt.modules.keystone.`**`user_verify_password`**(*user_id=None, name=None, password=None, profile=None, **connection_args*)

Verify a user's password

CLI Examples:

```
salt '*' keystone.user_verify_password name=test password=foobar
salt '*' keystone.user_verify_password user_id=c965f79c4f864eaaa9c3b41904e67082␣
↪password=foobar
```

21.9.193 salt.modules.keystoneng

Keystone module for interacting with OpenStack Keystone

New in version 2018.3.0.

:depends:shade

Example configuration

```
keystone:
  cloud: default
```

```
keystone:
  auth:
    username: admin
    password: password123
    user_domain_name: mydomain
    project_name: myproject
    project_domain_name: myproject
    auth_url: https://example.org:5000/v3
  identity_api_version: 3
```

`salt.modules.keystoneng.`**`compare_changes`**(*obj*, ***kwargs*)

Compare two dicts returning only keys that exist in the first dict and are different in the second one

`salt.modules.keystoneng.`**`domain_create`**(*auth=None*, ***kwargs*)

Create a domain

CLI Example:

```
salt '*' keystoneng.domain_create name=domain1
```

`salt.modules.keystoneng.`**`domain_delete`**(*auth=None*, ***kwargs*)

Delete a domain

CLI Example:

```
salt '*' keystoneng.domain_delete name=domain1
salt '*' keystoneng.domain_delete name=b62e76fbeeff4e8fb77073f591cf211e
```

`salt.modules.keystoneng.`**`domain_get`**(*auth=None*, ***kwargs*)

Get a single domain

CLI Example:

```
salt '*' keystoneng.domain_get name=domain1
salt '*' keystoneng.domain_get name=b62e76fbeeff4e8fb77073f591cf211e
```

`salt.modules.keystoneng.`**`domain_list`**(*auth=None*, ***kwargs*)

List domains

CLI Example:

```
salt '*' keystoneng.domain_list
```

`salt.modules.keystoneng.`**`domain_search`**(*auth=None*, ***kwargs*)

Search domains

CLI Example:

```
salt '*' keystoneng.domain_search
salt '*' keystoneng.domain_search name=domain1
```

`salt.modules.keystoneng.`**`domain_update`**(*auth=None*, ***kwargs*)

Update a domain

CLI Example:

```
salt '*' keystoneng.domain_update name=domain1 new_name=newdomain
salt '*' keystoneng.domain_update name=domain1 enabled=True description='new␣
↪description'
```

salt.modules.keystoneng.**endpoint_create**(*auth=None*, ***kwargs*)
Create an endpoint

CLI Example:

```
salt '*' keystoneng.endpoint_create interface=admin service=glance url=https://
↪example.org:9292
salt '*' keystoneng.endpoint_create interface=public service=glance␣
↪region=RegionOne url=https://example.org:9292
salt '*' keystoneng.endpoint_create interface=admin service=glance url=https://
↪example.org:9292 enabled=True
```

salt.modules.keystoneng.**endpoint_delete**(*auth=None*, ***kwargs*)
Delete an endpoint

CLI Example:

```
salt '*' keystoneng.endpoint_delete id=3bee4bd8c2b040ee966adfda1f0bfca9
```

salt.modules.keystoneng.**endpoint_get**(*auth=None*, ***kwargs*)
Get a single endpoint

CLI Example:

```
salt '*' keystoneng.endpoint_get id=02cffaa173b2460f98e40eda3748dae5
```

salt.modules.keystoneng.**endpoint_list**(*auth=None*, ***kwargs*)
List endpoints

CLI Example:

```
salt '*' keystoneng.endpoint_list
```

salt.modules.keystoneng.**endpoint_search**(*auth=None*, ***kwargs*)
Search endpoints

CLI Example:

```
salt '*' keystoneng.endpoint_search
salt '*' keystoneng.endpoint_search id=02cffaa173b2460f98e40eda3748dae5
```

salt.modules.keystoneng.**endpoint_update**(*auth=None*, ***kwargs*)
Update an endpoint

CLI Example:

```
salt '*' keystoneng.endpoint_update endpoint_id=4f961ad09d2d48948896bbe7c6a79717␣
↪interface=public enabled=False
salt '*' keystoneng.endpoint_update endpoint_id=4f961ad09d2d48948896bbe7c6a79717␣
↪region=newregion
salt '*' keystoneng.endpoint_update endpoint_id=4f961ad09d2d48948896bbe7c6a79717␣
↪service_name_or_id=glance url=https://example.org:9292
```

salt.modules.keystoneng.**get_entity**(*ent_type*, ***kwargs*)
Attempt to query Keystone for more information about an entity

`salt.modules.keystoneng.`**`get_openstack_cloud`**(*auth=None*)
Return an openstack_cloud

`salt.modules.keystoneng.`**`get_operator_cloud`**(*auth=None*)
Return an operator_cloud

`salt.modules.keystoneng.`**`group_create`**(*auth=None*, ***kwargs*)
Create a group

CLI Example:

```
salt '*' keystoneng.group_create name=group1
salt '*' keystoneng.group_create name=group2 domain=domain1 description='my group2
↪'
```

`salt.modules.keystoneng.`**`group_delete`**(*auth=None*, ***kwargs*)
Delete a group

CLI Example:

```
salt '*' keystoneng.group_delete name=group1
salt '*' keystoneng.group_delete name=group2 domain_
↪id=b62e76fbeeff4e8fb77073f591cf211e
salt '*' keystoneng.group_delete name=0e4febc2a5ab4f2c8f374b054162506d
```

`salt.modules.keystoneng.`**`group_get`**(*auth=None*, ***kwargs*)
Get a single group

CLI Example:

```
salt '*' keystoneng.group_get name=group1
salt '*' keystoneng.group_get name=group2 domain_
↪id=b62e76fbeeff4e8fb77073f591cf211e
salt '*' keystoneng.group_get name=0e4febc2a5ab4f2c8f374b054162506d
```

`salt.modules.keystoneng.`**`group_list`**(*auth=None*, ***kwargs*)
List groups

CLI Example:

```
salt '*' keystoneng.group_list
salt '*' keystoneng.group_list domain_id=b62e76fbeeff4e8fb77073f591cf211e
```

`salt.modules.keystoneng.`**`group_search`**(*auth=None*, ***kwargs*)
Search for groups

CLI Example:

```
salt '*' keystoneng.group_search name=group1
salt '*' keystoneng.group_search domain_id=b62e76fbeeff4e8fb77073f591cf211e
```

`salt.modules.keystoneng.`**`group_update`**(*auth=None*, ***kwargs*)
Update a group

CLI Example:

```
salt '*' keystoneng.group_update name=group1 description='new description'
salt '*' keystoneng.group_create name=group2 domain_
↪id=b62e76fbeeff4e8fb77073f591cf211e new_name=newgroupname
salt '*' keystoneng.group_create name=0e4febc2a5ab4f2c8f374b054162506d new_
↪name=newgroupname
```

salt.modules.keystoneng.**project_create**(*auth=None*, ***kwargs*)

Create a project

CLI Example:

```
salt '*' keystoneng.project_create name=project1
salt '*' keystoneng.project_create name=project2 domain_
↪id=b62e76fbeeff4e8fb77073f591cf211e
salt '*' keystoneng.project_create name=project3 enabled=False description='my⊠
↪project3'
```

salt.modules.keystoneng.**project_delete**(*auth=None*, ***kwargs*)

Delete a project

CLI Example:

```
salt '*' keystoneng.project_delete name=project1
salt '*' keystoneng.project_delete name=project2 domain_
↪id=b62e76fbeeff4e8fb77073f591cf211e
salt '*' keystoneng.project_delete name=f315afcf12f24ad88c92b936c38f2d5a
```

salt.modules.keystoneng.**project_get**(*auth=None*, ***kwargs*)

Get a single project

CLI Example:

```
salt '*' keystoneng.project_get name=project1
salt '*' keystoneng.project_get name=project2 domain_
↪id=b62e76fbeeff4e8fb77073f591cf211e
salt '*' keystoneng.project_get name=f315afcf12f24ad88c92b936c38f2d5a
```

salt.modules.keystoneng.**project_list**(*auth=None*, ***kwargs*)

List projects

CLI Example:

```
salt '*' keystoneng.project_list
salt '*' keystoneng.project_list domain_id=b62e76fbeeff4e8fb77073f591cf211e
```

salt.modules.keystoneng.**project_search**(*auth=None*, ***kwargs*)

Search projects

CLI Example:

```
salt '*' keystoneng.project_search
salt '*' keystoneng.project_search name=project1
salt '*' keystoneng.project_search domain_id=b62e76fbeeff4e8fb77073f591cf211e
```

salt.modules.keystoneng.**project_update**(*auth=None*, ***kwargs*)

Update a project

CLI Example:

```
salt '*' keystoneng.project_update name=project1 new_name=newproject
salt '*' keystoneng.project_update name=project2 enabled=False description='new⊠
↪description'
```

salt.modules.keystoneng.**role_assignment_list**(*auth=None*, ***kwargs*)

List role assignments

CLI Example:

```
salt '*' keystoneng.role_assignment_list
```

`salt.modules.keystoneng.role_create(auth=None, **kwargs)`

Create a role

CLI Example:

```
salt '*' keystoneng.role_create name=role1
salt '*' keystoneng.role_create name=role1 domain_
↪id=b62e76fbeeff4e8fb77073f591cf211e
```

`salt.modules.keystoneng.role_delete(auth=None, **kwargs)`

Delete a role

CLI Example:

```
salt '*' keystoneng.role_delete name=role1 domain_
↪id=b62e76fbeeff4e8fb77073f591cf211e
salt '*' keystoneng.role_delete name=1eb6edd5525e4ac39af571adee673559
```

`salt.modules.keystoneng.role_get(auth=None, **kwargs)`

Get a single role

CLI Example:

```
salt '*' keystoneng.role_get name=role1
salt '*' keystoneng.role_get name=role1 domain_id=b62e76fbeeff4e8fb77073f591cf211e
salt '*' keystoneng.role_get name=1eb6edd5525e4ac39af571adee673559
```

`salt.modules.keystoneng.role_grant(auth=None, **kwargs)`

Grant a role in a project/domain to a user/group

CLI Example:

```
salt '*' keystoneng.role_grant name=role1 user=user1 project=project1
salt '*' keystoneng.role_grant name=ddbe3e0ed74e4c7f8027bad4af03339d group=user1
↪project=project1 domain=domain1
salt '*' keystoneng.role_grant name=ddbe3e0ed74e4c7f8027bad4af03339d
↪group=19573afd5e4241d8b65c42215bae9704 project=1dcac318a83b4610b7a7f7ba01465548
```

`salt.modules.keystoneng.role_list(auth=None, **kwargs)`

List roles

CLI Example:

```
salt '*' keystoneng.role_list
salt '*' keystoneng.role_list domain_id=b62e76fbeeff4e8fb77073f591cf211e
```

`salt.modules.keystoneng.role_revoke(auth=None, **kwargs)`

Grant a role in a project/domain to a user/group

CLI Example:

```
salt '*' keystoneng.role_revoke name=role1 user=user1 project=project1
salt '*' keystoneng.role_revoke name=ddbe3e0ed74e4c7f8027bad4af03339d group=user1
↪project=project1 domain=domain1
salt '*' keystoneng.role_revoke name=ddbe3e0ed74e4c7f8027bad4af03339d
↪group=19573afd5e4241d8b65c42215bae9704 project=1dcac318a83b4610b7a7f7ba01465548
```

salt.modules.keystoneng.**role_search**(*auth=None*, ***kwargs*)

Search roles

CLI Example:

```
salt '*' keystoneng.role_search
salt '*' keystoneng.role_search name=role1
salt '*' keystoneng.role_search domain_id=b62e76fbeeff4e8fb77073f591cf211e
```

salt.modules.keystoneng.**role_update**(*auth=None*, ***kwargs*)

Update a role

CLI Example:

```
salt '*' keystoneng.role_update name=role1 new_name=newrole
salt '*' keystoneng.role_update name=1eb6edd5525e4ac39af571adee673559 new_
↪name=newrole
```

salt.modules.keystoneng.**service_create**(*auth=None*, ***kwargs*)

Create a service

CLI Example:

```
salt '*' keystoneng.service_create name=glance type=image
salt '*' keystoneng.service_create name=glance type=image description="Image"
```

salt.modules.keystoneng.**service_delete**(*auth=None*, ***kwargs*)

Delete a service

CLI Example:

```
salt '*' keystoneng.service_delete name=glance
salt '*' keystoneng.service_delete name=39cc1327cdf744ab815331554430e8ec
```

salt.modules.keystoneng.**service_get**(*auth=None*, ***kwargs*)

Get a single service

CLI Example:

```
salt '*' keystoneng.service_get name=glance
salt '*' keystoneng.service_get name=75a5804638944b3ab54f7fbfcec2305a
```

salt.modules.keystoneng.**service_list**(*auth=None*, ***kwargs*)

List services

CLI Example:

```
salt '*' keystoneng.service_list
```

salt.modules.keystoneng.**service_search**(*auth=None*, ***kwargs*)

Search services

CLI Example:

```
salt '*' keystoneng.service_search
salt '*' keystoneng.service_search name=glance
salt '*' keystoneng.service_search name=135f0403f8e544dc9008c6739ecda860
```

salt.modules.keystoneng.**service_update**(*auth=None*, ***kwargs*)

Update a service

CLI Example:

```
salt '*' keystoneng.service_update name=cinder type=volumev2
salt '*' keystoneng.service_update name=cinder description='new description'
salt '*' keystoneng.service_update name=ab4d35e269f147b3ae2d849f77f5c88f
↪enabled=False
```

salt.modules.keystoneng.**setup_clouds**(*auth=None*)

Call functions to create Shade cloud objects in __context__ to take advantage of Shade's in-memory caching across several states

salt.modules.keystoneng.**user_create**(*auth=None*, ***kwargs*)

Create a user

CLI Example:

```
salt '*' keystoneng.user_create name=user1
salt '*' keystoneng.user_create name=user2 password=1234 enabled=False
salt '*' keystoneng.user_create name=user3 domain_
↪id=b62e76fbeeff4e8fb77073f591cf211e
```

salt.modules.keystoneng.**user_delete**(*auth=None*, ***kwargs*)

Delete a user

CLI Example:

```
salt '*' keystoneng.user_delete name=user1
salt '*' keystoneng.user_delete name=user2 domain_
↪id=b62e76fbeeff4e8fb77073f591cf211e
salt '*' keystoneng.user_delete name=a42cbbfa1e894e839fd0f584d22e321f
```

salt.modules.keystoneng.**user_get**(*auth=None*, ***kwargs*)

Get a single user

CLI Example:

```
salt '*' keystoneng.user_get name=user1
salt '*' keystoneng.user_get name=user1 domain_id=b62e76fbeeff4e8fb77073f591cf211e
salt '*' keystoneng.user_get name=02cffaa173b2460f98e40eda3748dae5
```

salt.modules.keystoneng.**user_list**(*auth=None*, ***kwargs*)

List users

CLI Example:

```
salt '*' keystoneng.user_list
salt '*' keystoneng.user_list domain_id=b62e76fbeeff4e8fb77073f591cf211e
```

salt.modules.keystoneng.**user_search**(*auth=None*, ***kwargs*)

List users

CLI Example:

```
salt '*' keystoneng.user_list
salt '*' keystoneng.user_list domain_id=b62e76fbeeff4e8fb77073f591cf211e
```

salt.modules.keystoneng.**user_update**(*auth=None*, ***kwargs*)

Update a user

CLI Example:

```
salt '*' keystoneng.user_update name=user1 enabled=False description='new␣
↪description'
salt '*' keystoneng.user_update name=user1 new_name=newuser
```

21.9.194 salt.modules.keystore

Module to interact with keystores

`salt.modules.keystore.`**`add`**(*name*, *keystore*, *passphrase*, *certificate*, *private_key=None*)

Adds certificates to an existing keystore or creates a new one if necesssary.

Parameters

- **name** – alias for the certificate
- **keystore** – The path to the keystore file to query
- **passphrase** – The passphrase to use to decode the keystore
- **certificate** – The PEM public certificate to add to keystore. Can be a string for file.
- **private_key** – (Optional for TrustedCert) The PEM private key to add to the keystore

CLI Example:

```
salt '*' keystore.add aliasname /tmp/test.store changeit /tmp/testcert.crt
salt '*' keystore.add aliasname /tmp/test.store changeit certificate="-----BEGIN␣
↪CERTIFICATE-----SIb...BM=-----END CERTIFICATE-----"
salt '*' keystore.add keyname /tmp/test.store changeit /tmp/512.cert private_key=/
↪tmp/512.key
```

`salt.modules.keystore.`**`list`**(*keystore*, *passphrase*, *alias=None*, *return_cert=False*)

Lists certificates in a keytool managed keystore.

Parameters

- **keystore** – The path to the keystore file to query
- **passphrase** – The passphrase to use to decode the keystore
- **alias** – (Optional) If found, displays details on only this key
- **return_certs** – (Optional) Also return certificate PEM.

Warning: There are security implications for using return_cert to return decrypted certificates.

CLI Example:

```
salt '*' keystore.list /usr/lib/jvm/java-8/jre/lib/security/cacerts changeit
salt '*' keystore.list /usr/lib/jvm/java-8/jre/lib/security/cacerts changeit␣
↪debian:verisign_-_g5.pem
```

`salt.modules.keystore.`**`remove`**(*name*, *keystore*, *passphrase*)

Removes a certificate from an existing keystore. Returns True if remove was successful, otherwise False

Parameters

- **name** – alias for the certificate
- **keystore** – The path to the keystore file to query

- **passphrase** – The passphrase to use to decode the keystore

CLI Example:

```
salt '*' keystore.remove aliasname /tmp/test.store changeit
```

21.9.195 salt.modules.kmod

Module to manage Linux kernel modules

`salt.modules.kmod.`**`available`**`()`

Return a list of all available kernel modules

CLI Example:

```
salt '*' kmod.available
```

`salt.modules.kmod.`**`check_available`**`(`*mod*`)`

Check to see if the specified kernel module is available

CLI Example:

```
salt '*' kmod.check_available kvm
```

`salt.modules.kmod.`**`is_loaded`**`(`*mod*`)`

Check to see if the specified kernel module is loaded

CLI Example:

```
salt '*' kmod.is_loaded kvm
```

`salt.modules.kmod.`**`load`**`(`*mod, persist=False*`)`

Load the specified kernel module

mod Name of module to add

persist Write module to /etc/modules to make it load on system reboot

CLI Example:

```
salt '*' kmod.load kvm
```

`salt.modules.kmod.`**`lsmod`**`()`

Return a dict containing information about currently loaded modules

CLI Example:

```
salt '*' kmod.lsmod
```

`salt.modules.kmod.`**`mod_list`**`(`*only_persist=False*`)`

Return a list of the loaded module names

only_persist Only return the list of loaded persistent modules

CLI Example:

```
salt '*' kmod.mod_list
```

`salt.modules.kmod.`**`remove`**`(`*mod, persist=False, comment=True*`)`

Remove the specified kernel module

mod Name of module to remove

persist Also remove module from /etc/modules

comment If persist is set don't remove line from /etc/modules but only comment it

CLI Example:

```
salt '*' kmod.remove kvm
```

21.9.196 salt.modules.kubernetesmod

Module for handling kubernetes calls.

optdepends

- kubernetes Python client

configuration The k8s API settings are provided either in a pillar, in the minion's config file, or in master's config file:

```
kubernetes.kubeconfig: '/path/to/kubeconfig'
kubernetes.kubeconfig-data: '<base64 encoded kubeconfig content'
kubernetes.context: 'context'
```

These settings can be overridden by adding *context and 'kubeconfig* or *kubeconfig_data* parameters when calling a function.

The data format for *kubernetes.kubeconfig-data* value is the content of *kubeconfig* base64 encoded in one line.

Only *kubeconfig* or *kubeconfig-data* should be provided. In case both are provided *kubeconfig* entry is preferred.

```
salt '*' kubernetes.nodes kubeconfig=/etc/salt/k8s/kubeconfig context=minikube
```

Changed in version 2019.2.0.

Warning: Configuration options changed in 2019.2.0. The following configuration options have been removed:

- kubernetes.user
- kubernetes.password
- kubernetes.api_url
- kubernetes.certificate-authority-data/file
- kubernetes.client-certificate-data/file
- kubernetes.client-key-data/file

Please use now:

- kubernetes.kubeconfig or kubernetes.kubeconfig-data
- kubernetes.context

`salt.modules.kubernetesmod.`**`configmaps`**`(`*namespace='default', **kwargs*`)`

Return a list of kubernetes configmaps defined in the namespace

CLI Examples:

```
salt '*' kubernetes.configmaps
salt '*' kubernetes.configmaps namespace=default
```

`salt.modules.kubernetesmod.`**`create_configmap`**(*name, namespace, data, source=None, template=None, saltenv='base', **kwargs*)

Creates the kubernetes configmap as defined by the user.

CLI Examples:

```
salt 'minion1' kubernetes.create_configmap             settings default '{
↪"example.conf": "# example file"}'

salt 'minion2' kubernetes.create_configmap             name=settings⊠
↪namespace=default data='{"example.conf": "# example file"}'
```

`salt.modules.kubernetesmod.`**`create_deployment`**(*name, namespace, metadata, spec, source, template, saltenv, **kwargs*)

Creates the kubernetes deployment as defined by the user.

`salt.modules.kubernetesmod.`**`create_namespace`**(*name, **kwargs*)

Creates a namespace with the specified name.

CLI Example: salt '*' kubernetes.create_namespace salt salt '*' kubernetes.create_namespace name=salt

`salt.modules.kubernetesmod.`**`create_pod`**(*name, namespace, metadata, spec, source, template, saltenv, **kwargs*)

Creates the kubernetes deployment as defined by the user.

`salt.modules.kubernetesmod.`**`create_secret`**(*name, namespace='default', data=None, source=None, template=None, saltenv='base', **kwargs*)

Creates the kubernetes secret as defined by the user.

CLI Examples:

```
salt 'minion1' kubernetes.create_secret             passwords default '{"db":
↪"letmein"}'

salt 'minion2' kubernetes.create_secret             name=passwords⊠
↪namespace=default data='{"db": "letmein"}'
```

`salt.modules.kubernetesmod.`**`create_service`**(*name, namespace, metadata, spec, source, template, saltenv, **kwargs*)

Creates the kubernetes service as defined by the user.

`salt.modules.kubernetesmod.`**`delete_configmap`**(*name, namespace='default', **kwargs*)

Deletes the kubernetes configmap defined by name and namespace

CLI Examples:

```
salt '*' kubernetes.delete_configmap settings default
salt '*' kubernetes.delete_configmap name=settings namespace=default
```

`salt.modules.kubernetesmod.`**`delete_deployment`**(*name, namespace='default', **kwargs*)

Deletes the kubernetes deployment defined by name and namespace

CLI Examples:

```
salt '*' kubernetes.delete_deployment my-nginx
salt '*' kubernetes.delete_deployment name=my-nginx namespace=default
```

`salt.modules.kubernetesmod.`**`delete_namespace`**`(`*name*, ***kwargs*`)`

Deletes the kubernetes namespace defined by name

CLI Examples:

```
salt '*' kubernetes.delete_namespace salt
salt '*' kubernetes.delete_namespace name=salt
```

`salt.modules.kubernetesmod.`**`delete_pod`**`(`*name*, *namespace='default'*, ***kwargs*`)`

Deletes the kubernetes pod defined by name and namespace

CLI Examples:

```
salt '*' kubernetes.delete_pod guestbook-708336848-5nl8c default
salt '*' kubernetes.delete_pod name=guestbook-708336848-5nl8c namespace=default
```

`salt.modules.kubernetesmod.`**`delete_secret`**`(`*name*, *namespace='default'*, ***kwargs*`)`

Deletes the kubernetes secret defined by name and namespace

CLI Examples:

```
salt '*' kubernetes.delete_secret confidential default
salt '*' kubernetes.delete_secret name=confidential namespace=default
```

`salt.modules.kubernetesmod.`**`delete_service`**`(`*name*, *namespace='default'*, ***kwargs*`)`

Deletes the kubernetes service defined by name and namespace

CLI Examples:

```
salt '*' kubernetes.delete_service my-nginx default
salt '*' kubernetes.delete_service name=my-nginx namespace=default
```

`salt.modules.kubernetesmod.`**`deployments`**`(`*namespace='default'*, ***kwargs*`)`

Return a list of kubernetes deployments defined in the namespace

CLI Examples:

```
salt '*' kubernetes.deployments
salt '*' kubernetes.deployments namespace=default
```

`salt.modules.kubernetesmod.`**`namespaces`**`(`***kwargs*`)`

Return the names of the available namespaces

CLI Examples:

```
salt '*' kubernetes.namespaces
salt '*' kubernetes.namespaces kubeconfig=/etc/salt/k8s/kubeconfig
↪context=minikube
```

`salt.modules.kubernetesmod.`**`node`**`(`*name*, ***kwargs*`)`

Return the details of the node identified by the specified name

CLI Examples:

```
salt '*' kubernetes.node name='minikube'
```

`salt.modules.kubernetesmod.`**`node_add_label`**`(`*node_name*, *label_name*, *label_value*, ***kwargs*`)`

Set the value of the label identified by *label_name* to *label_value* on the node identified by the name *node_name*. Creates the lable if not present.

CLI Examples:

```
salt '*' kubernetes.node_add_label node_name="minikube"            label_name=
↪"foo" label_value="bar"
```

`salt.modules.kubernetesmod.`**`node_labels`**`(`*name*, ***kwargs*`)`

Return the labels of the node identified by the specified name

CLI Examples:

```
salt '*' kubernetes.node_labels name="minikube"
```

`salt.modules.kubernetesmod.`**`node_remove_label`**`(`*node_name*, *label_name*, ***kwargs*`)`

Removes the label identified by *label_name* from the node identified by the name *node_name*.

CLI Examples:

```
salt '*' kubernetes.node_remove_label node_name="minikube"            label_name=
↪"foo"
```

`salt.modules.kubernetesmod.`**`nodes`**`(`***kwargs*`)`

Return the names of the nodes composing the kubernetes cluster

CLI Examples:

```
salt '*' kubernetes.nodes
salt '*' kubernetes.nodes kubeconfig=/etc/salt/k8s/kubeconfig context=minikube
```

`salt.modules.kubernetesmod.`**`ping`**`(`***kwargs*`)`

Checks connections with the kubernetes API server. Returns True if the connection can be established, False otherwise.

CLI Example: salt '*' kubernetes.ping

`salt.modules.kubernetesmod.`**`pods`**`(`*namespace='default'*, ***kwargs*`)`

Return a list of kubernetes pods defined in the namespace

CLI Examples:

```
salt '*' kubernetes.pods
salt '*' kubernetes.pods namespace=default
```

`salt.modules.kubernetesmod.`**`replace_configmap`**`(`*name*, *data*, *source=None*, *template=None*, *saltenv='base'*, *namespace='default'*, ***kwargs*`)`

Replaces an existing configmap with a new one defined by name and namespace with the specified data.

CLI Examples:

```
salt 'minion1' kubernetes.replace_configmap             settings default '{
↪"example.conf": "# example file"}'

salt 'minion2' kubernetes.replace_configmap             name=settings
↪namespace=default data='{"example.conf": "# example file"}'
```

`salt.modules.kubernetesmod.`**`replace_deployment`**`(`*name*, *metadata*, *spec*, *source*, *template*, *saltenv*, *namespace='default'*, ***kwargs*`)`

Replaces an existing deployment with a new one defined by name and namespace, having the specificed metadata and spec.

`salt.modules.kubernetesmod.`**`replace_secret`**`(`*name, data, source=None, template=None, saltenv='base', namespace='default', **kwargs*`)`

Replaces an existing secret with a new one defined by name and namespace, having the specificed data.

CLI Examples:

```
salt 'minion1' kubernetes.replace_secret            name=passwords data='{"db":
↪"letmein"}'

salt 'minion2' kubernetes.replace_secret            name=passwords␣
↪namespace=saltstack data='{"db": "passw0rd"}'
```

`salt.modules.kubernetesmod.`**`replace_service`**`(`*name, metadata, spec, source, template, old_service, saltenv, namespace='default', **kwargs*`)`

Replaces an existing service with a new one defined by name and namespace, having the specificed metadata and spec.

`salt.modules.kubernetesmod.`**`secrets`**`(`*namespace='default', **kwargs*`)`

Return a list of kubernetes secrets defined in the namespace

CLI Examples:

```
salt '*' kubernetes.secrets
salt '*' kubernetes.secrets namespace=default
```

`salt.modules.kubernetesmod.`**`services`**`(`*namespace='default', **kwargs*`)`

Return a list of kubernetes services defined in the namespace

CLI Examples:

```
salt '*' kubernetes.services
salt '*' kubernetes.services namespace=default
```

`salt.modules.kubernetesmod.`**`show_configmap`**`(`*name, namespace='default', **kwargs*`)`

Return the kubernetes configmap defined by name and namespace.

CLI Examples:

```
salt '*' kubernetes.show_configmap game-config default
salt '*' kubernetes.show_configmap name=game-config namespace=default
```

`salt.modules.kubernetesmod.`**`show_deployment`**`(`*name, namespace='default', **kwargs*`)`

Return the kubernetes deployment defined by name and namespace

CLI Examples:

```
salt '*' kubernetes.show_deployment my-nginx default
salt '*' kubernetes.show_deployment name=my-nginx namespace=default
```

`salt.modules.kubernetesmod.`**`show_namespace`**`(`*name, **kwargs*`)`

Return information for a given namespace defined by the specified name

CLI Examples:

```
salt '*' kubernetes.show_namespace kube-system
```

`salt.modules.kubernetesmod.`**`show_pod`**`(`*name, namespace='default', **kwargs*`)`

Return POD information for a given pod name defined in the namespace

CLI Examples:

```
salt '*' kubernetes.show_pod guestbook-708336848-fqr2x
salt '*' kubernetes.show_pod guestbook-708336848-fqr2x namespace=default
```

`salt.modules.kubernetesmod.`**`show_secret`**`(`*name*, *namespace='default'*, *decode=False*, ***kwargs*`)`

Return the kubernetes secret defined by name and namespace. The secrets can be decoded if specified by the user. Warning: this has security implications.

CLI Examples:

```
salt '*' kubernetes.show_secret confidential default
salt '*' kubernetes.show_secret name=confidential namespace=default
salt '*' kubernetes.show_secret name=confidential decode=True
```

`salt.modules.kubernetesmod.`**`show_service`**`(`*name*, *namespace='default'*, ***kwargs*`)`

Return the kubernetes service defined by name and namespace

CLI Examples:

```
salt '*' kubernetes.show_service my-nginx default
salt '*' kubernetes.show_service name=my-nginx namespace=default
```

21.9.197 salt.modules.launchctl_service

Module for the management of MacOS systems that use launchd/launchctl

---

Important: If you feel that Salt should be using this module to manage services on a minion, and it is using a different module (or gives an error similar to *'service.start' is not available*), see *here*.

---

depends

- plistlib Python module

`salt.modules.launchctl_service.`**`available`**`(`*job_label*`)`

Check that the given service is available.

CLI Example:

```
salt '*' service.available com.openssh.sshd
```

`salt.modules.launchctl_service.`**`disabled`**`(`*job_label*, *runas=None*`)`

Return True if the named service is disabled, false otherwise

CLI Example:

```
salt '*' service.disabled <service label>
```

`salt.modules.launchctl_service.`**`enabled`**`(`*job_label*, *runas=None*`)`

Return True if the named service is enabled, false otherwise

CLI Example:

```
salt '*' service.enabled <service label>
```

`salt.modules.launchctl_service.`**`get_all`**`()`

Return all installed services

CLI Example:

```
salt '*' service.get_all
```

`salt.modules.launchctl_service.`**`missing`**(*job_label*)

The inverse of service.available Check that the given service is not available.

CLI Example:

```
salt '*' service.missing com.openssh.sshd
```

`salt.modules.launchctl_service.`**`restart`**(*job_label*, *runas=None*)

Restart the named service

CLI Example:

```
salt '*' service.restart <service label>
```

`salt.modules.launchctl_service.`**`start`**(*job_label*, *runas=None*)

Start the specified service

CLI Example:

```
salt '*' service.start <service label>
salt '*' service.start org.ntp.ntpd
salt '*' service.start /System/Library/LaunchDaemons/org.ntp.ntpd.plist
```

`salt.modules.launchctl_service.`**`status`**(*name*, *runas=None*)

Return the status for a service via systemd. If the name contains globbing, a dict mapping service name to True/False values is returned.

Changed in version 2018.3.0: The service name can now be a glob (e.g. `salt*`)

Parameters

- **`name`** (*`str`*) – The name of the service to check
- **`runas`** (*`str`*) – User to run launchctl commands

Returns True if running, False otherwise dict: Maps service name to True if running, False otherwise

Return type bool

CLI Example:

```
salt '*' service.status <service name>
```

`salt.modules.launchctl_service.`**`stop`**(*job_label*, *runas=None*)

Stop the specified service

CLI Example:

```
salt '*' service.stop <service label>
salt '*' service.stop org.ntp.ntpd
salt '*' service.stop /System/Library/LaunchDaemons/org.ntp.ntpd.plist
```

21.9.198 salt.modules.layman

Support for Layman

`salt.modules.layman.`**`add`**`(`*overlay*`)`

Add the given overlay from the cached remote list to your locally installed overlays. Specify 'ALL' to add all overlays from the remote list.

Return a list of the new overlay(s) added:

CLI Example:

```
salt '*' layman.add <overlay name>
```

`salt.modules.layman.`**`delete`**`(`*overlay*`)`

Remove the given overlay from the your locally installed overlays. Specify 'ALL' to remove all overlays.

Return a list of the overlays(s) that were removed:

CLI Example:

```
salt '*' layman.delete <overlay name>
```

`salt.modules.layman.`**`list_all`**`()`

List all overlays, including remote ones.

Return a list of available overlays:

CLI Example:

```
salt '*' layman.list_all
```

`salt.modules.layman.`**`list_local`**`()`

List the locally installed overlays.

Return a list of installed overlays:

CLI Example:

```
salt '*' layman.list_local
```

`salt.modules.layman.`**`sync`**`(`*overlay='ALL'*`)`

Update the specified overlay. Use 'ALL' to synchronize all overlays. This is the default if no overlay is specified.

overlay Name of the overlay to sync. (Defaults to 'ALL')

CLI Example:

```
salt '*' layman.sync
```

21.9.199 salt.modules.ldap3

Query and modify an LDAP database (alternative interface)

New in version 2016.3.0.

This is an alternative to the `ldap` interface provided by the *`ldapmod`* execution module.

depends

- `ldap` Python module

`exception` `salt.modules.ldap3.`**`LDAPError`**`(`*message, cause=None*`)`

Base class of all LDAP exceptions raised by backends.

This is only used for errors encountered while interacting with the LDAP server; usage errors (e.g., invalid backend name) will have a different type.

Variables **`cause`** – backend exception object, if applicable

`salt.modules.ldap3.`**`add`**(*connect_spec*, *dn*, *attributes*)

Add an entry to an LDAP database.

Parameters

- **`connect_spec`** – See the documentation for the `connect_spec` parameter for *`connect()`*.
- **`dn`** – Distinguished name of the entry.
- **`attributes`** – Non-empty dict mapping each of the new entry's attributes to a non-empty iterable of values.

Returns `True` if successful, raises an exception otherwise.

CLI example:

```
salt '*' ldap3.add "{
    'url': 'ldaps://ldap.example.com/',
    'bind': {
        'method': 'simple',
        'password': 'secret',
    },
}" "dn='dc=example,dc=com'" "attributes={'example': 'values'}"
```

`salt.modules.ldap3.`**`change`**(*connect_spec*, *dn*, *before*, *after*)

Modify an entry in an LDAP database.

This does the same thing as *`modify()`*, but with a simpler interface. Instead of taking a list of directives, it takes a before and after view of an entry, determines the differences between the two, computes the directives, and executes them.

Any attribute value present in `before` but missing in `after` is deleted. Any attribute value present in `after` but missing in `before` is added. Any attribute value in the database that is not mentioned in either `before` or `after` is not altered. Any attribute value that is present in both `before` and `after` is ignored, regardless of whether that attribute value exists in the database.

Parameters

- **`connect_spec`** – See the documentation for the `connect_spec` parameter for *`connect()`*.
- **`dn`** – Distinguished name of the entry.
- **`before`** – The expected state of the entry before modification. This is a dict mapping each attribute name to an iterable of values.
- **`after`** – The desired state of the entry after modification. This is a dict mapping each attribute name to an iterable of values.

Returns `True` if successful, raises an exception otherwise.

CLI example:

```
salt '*' ldap3.change "{
    'url': 'ldaps://ldap.example.com/',
    'bind': {
        'method': 'simple',
```

(continues on next page)

(continued from previous page)

```
        'password': 'secret'}
}" dn='cn=admin,dc=example,dc=com'
before="{'example_value': 'before_val'}"
after="{'example_value': 'after_val'}"
```

`salt.modules.ldap3.`**`connect`**`(`*connect_spec=None*`)`

Connect and optionally bind to an LDAP server.

Parameters **`connect_spec`** – This can be an LDAP connection object returned by a previous call to *`connect()`* (in which case the argument is simply returned), `None` (in which case an empty dict is used), or a dict with the following keys:

- **`'backend'`** Optional; default depends on which Python LDAP modules are installed. Name of the Python LDAP module to use. Only `'ldap'` is supported at the moment.
- **`'url'`** Optional; defaults to `'ldapi:///'`. URL to the LDAP server.
- **`'bind'`** Optional; defaults to `None`. Describes how to bind an identity to the LDAP connection. If `None`, an anonymous connection is made. Valid keys:
  - **`'method'`** Optional; defaults to `None`. The authentication method to use. Valid values include but are not necessarily limited to `'simple'`, `'sasl'`, and `None`. If `None`, an anonymous connection is made. Available methods depend on the chosen backend.
  - **`'mechanism'`** Optional; defaults to `'EXTERNAL'`. The SASL mechanism to use. Ignored unless the method is `'sasl'`. Available methods depend on the chosen backend and the server's capabilities.
  - **`'credentials'`** Optional; defaults to `None`. An object specific to the chosen SASL mechanism and backend that represents the authentication credentials. Ignored unless the method is `'sasl'`.

    For the `'ldap'` backend, this is a dictionary. If `None`, an empty dict is used. Keys:

    - **`'args'`** Optional; defaults to an empty list. A list of arguments to pass to the SASL mechanism constructor. See the SASL mechanism constructor documentation in the `ldap.sasl` Python module.
    - **`'kwargs'`** Optional; defaults to an empty dict. A dict of keyword arguments to pass to the SASL mechanism constructor. See the SASL mechanism constructor documentation in the `ldap.sasl` Python module.
  - **`'dn'`** Optional; defaults to an empty string. The distinguished name to bind.
  - **`'password'`** Optional; defaults to an empty string. Password for binding. Ignored if the method is `'sasl'`.
- **`'tls'`** Optional; defaults to `None`. A backend-specific object containing settings to override default TLS behavior.

  For the `'ldap'` backend, this is a dictionary. Not all settings in this dictionary are supported by all versions of `python-ldap` or the underlying TLS library. If `None`, an empty dict is used. Possible keys:

  - **`'starttls'`** If present, initiate a TLS connection using StartTLS. (The value associated with this key is ignored.)
  - **`'cacertdir'`** Set the path of the directory containing CA certificates.
  - **`'cacertfile'`** Set the pathname of the CA certificate file.

  - **'certfile'** Set the pathname of the certificate file.
  - **'cipher_suite'** Set the allowed cipher suite.
  - **'crlcheck'** Set the CRL evaluation strategy. Valid values are `'none'`, `'peer'`, and `'all'`.
  - **'crlfile'** Set the pathname of the CRL file.
  - **'dhfile'** Set the pathname of the file containing the parameters for Diffie-Hellman ephemeral key exchange.
  - **'keyfile'** Set the pathname of the certificate key file.
  - **'newctx'** If present, instruct the underlying TLS library to create a new TLS context. (The value associated with this key is ignored.)
  - **'protocol_min'** Set the minimum protocol version.
  - **'random_file'** Set the pathname of the random file when `/dev/random` and `/dev/urandom` are not available.
  - **'require_cert'** Set the certificate validation policy. Valid values are `'never'`, `'hard'`, `'demand'`, `'allow'`, and `'try'`.

- **'opts'** Optional; defaults to `None`. A backend-specific object containing options for the backend.

  For the `'ldap'` backend, this is a dictionary of OpenLDAP options to set. If `None`, an empty dict is used. Each key is a the name of an OpenLDAP option constant without the `'LDAP_OPT_'` prefix, then converted to lower case.

Returns

an object representing an LDAP connection that can be used as the `connect_spec` argument to any of the functions in this module (to avoid the overhead of making and terminating multiple connections).

This object should be used as a context manager. It is safe to nest `with` statements.

CLI example:

```
salt '*' ldap3.connect "{
    'url': 'ldaps://ldap.example.com/',
    'bind': {
        'method': 'simple',
        'dn': 'cn=admin,dc=example,dc=com',
        'password': 'secret'}
}"
```

`salt.modules.ldap3.`**`delete`**(*connect_spec*, *dn*)

Delete an entry from an LDAP database.

Parameters

- **connect_spec** – See the documentation for the `connect_spec` parameter for *`connect()`*.
- **dn** – Distinguished name of the entry.

Returns `True` if successful, raises an exception otherwise.

CLI example:

```
salt '*' ldap3.delete "{
    'url': 'ldaps://ldap.example.com/',
    'bind': {
        'method': 'simple',
        'password': 'secret'}
}" dn='cn=admin,dc=example,dc=com'
```

`salt.modules.ldap3.`**`modify`**(*connect_spec, dn, directives*)

Modify an entry in an LDAP database.

Parameters

- **`connect_spec`** – See the documentation for the `connect_spec` parameter for *`connect()`*.
- **`dn`** – Distinguished name of the entry.
- **`directives`** – Iterable of directives that indicate how to modify the entry. Each directive is a tuple of the form `(op, attr, vals)`, where:
  - `op` identifies the modification operation to perform. One of:
    - `'add'` to add one or more values to the attribute
    - `'delete'` to delete some or all of the values from the attribute. If no values are specified with this operation, all of the attribute's values are deleted. Otherwise, only the named values are deleted.
    - `'replace'` to replace all of the attribute's values with zero or more new values
  - `attr` names the attribute to modify
  - `vals` is an iterable of values to add or delete

Returns `True` if successful, raises an exception otherwise.

CLI example:

```
salt '*' ldap3.modify "{
    'url': 'ldaps://ldap.example.com/',
    'bind': {
        'method': 'simple',
        'password': 'secret'}
}" dn='cn=admin,dc=example,dc=com'
directives="('add', 'example', ['example_val'])"
```

`salt.modules.ldap3.`**`search`**(*connect_spec, base, scope='subtree', filterstr='(objectClass=*)', attrlist=None, attrsonly=0*)

Search an LDAP database.

Parameters

- **`connect_spec`** – See the documentation for the `connect_spec` parameter for *`connect()`*.
- **`base`** – Distinguished name of the entry at which to start the search.
- **`scope`** – One of the following:
  - **`'subtree'`** Search the base and all of its descendants.
  - **`'base'`** Search only the base itself.
  - **`'onelevel'`** Search only the base's immediate children.
- **`filterstr`** – String representation of the filter to apply in the search.

- **`attrlist`** – Limit the returned attributes to those in the specified list. If `None`, all attributes of each entry are returned.
- **`attrsonly`** – If non-zero, don't return any attribute values.

Returns a dict of results. The dict is empty if there are no results. The dict maps each returned entry's distinguished name to a dict that maps each of the matching attribute names to a list of its values.

CLI example:

```
salt '*' ldap3.search "{
    'url': 'ldaps://ldap.example.com/',
    'bind': {
        'method': 'simple',
        'dn': 'cn=admin,dc=example,dc=com',
        'password': 'secret',
    },
}" "base='dc=example,dc=com'"
```

21.9.200 salt.modules.ldapmod

Salt interface to LDAP commands

depends

- ldap Python module

configuration In order to connect to LDAP, certain configuration is required in the minion config on the LDAP server. The minimum configuration items that must be set are:

```
ldap.basedn: dc=acme,dc=com (example values, adjust to suit)
```

If your LDAP server requires authentication then you must also set:

```
ldap.anonymous: False
ldap.binddn: admin
ldap.bindpw: password
```

In addition, the following optional values may be set:

```
ldap.server: localhost (default=localhost, see warning below)
ldap.port: 389 (default=389, standard port)
ldap.tls: False (default=False, no TLS)
ldap.no_verify: False (default=False, verify TLS)
ldap.anonymous: True (default=True, bind anonymous)
ldap.scope: 2 (default=2, ldap.SCOPE_SUBTREE)
ldap.attrs: [saltAttr] (default=None, return all attributes)
```

Warning: At the moment this module only recommends connection to LDAP services listening on `localhost`. This is deliberate to avoid the potentially dangerous situation of multiple minions sending identical update commands to the same LDAP server. It's easy enough to override this behavior, but badness may ensue - you have been warned.

`salt.modules.ldapmod.`**`search`**(*filter*, *dn=None*, *scope=None*, *attrs=None*, ***kwargs*)

Run an arbitrary LDAP query and return the results.

CLI Example:

```
salt 'ldaphost' ldap.search "filter=cn=myhost"
```

Return data:

```
{'myhost': {'count': 1,
            'results': [['cn=myhost,ou=hosts,o=acme,c=gb',
                         {'saltKeyValue': ['ntpserver=ntp.acme.local',
                                           'foo=myfoo'],
                          'saltState': ['foo', 'bar']}]],
            'time': {'human': '1.2ms', 'raw': '0.00123'}}}
```

Search and connection options can be overridden by specifying the relevant option as key=value pairs, for example:

```
salt 'ldaphost' ldap.search filter=cn=myhost dn=ou=hosts,o=acme,c=gb
scope=1 attrs='' server='localhost' port='7393' tls=True bindpw='ssh'
```

21.9.201 salt.modules.libcloud_compute module

Apache Libcloud Compute Management

Connection module for Apache Libcloud Compute management for a full list of supported clouds, see http://libcloud.readthedocs.io/en/latest/compute/supported_providers.html

Clouds include Amazon EC2, Azure, Google GCE, VMware, OpenStack Nova

New in version 2018.3.0.

configuration This module uses a configuration profile for one or multiple cloud providers

```
libcloud_compute:
    profile_test1:
      driver: google
      key: service-account@googlecloud.net
      secret: /path/to.key.json
    profile_test2:
      driver: arm
      key: 12345
      secret: mysecret
```

depends apache-libcloud

`salt.modules.libcloud_compute.`**`attach_volume`**`(`*node_id, volume_id, profile, device=None, **libcloud_kwargs*`)`

Attaches volume to node.

Parameters

- **node_id** (str) – Node ID to target
- **volume_id** (str) – Volume ID from which to attach
- **profile** (str) – The profile key
- **device** (str) – Where the device is exposed, e.g. '/dev/sdb'
- **libcloud_kwargs** (dict) – Extra arguments for the driver's attach_volume method

CLI Example:

```
salt myminion libcloud_compute.detach_volume vol1 profile1
```

`salt.modules.libcloud_compute.`**`copy_image`**(*source_region, image_id, name, profile, description=None, **libcloud_kwargs*)

Copies an image from a source region to the current region.

Parameters

- **`source_region`** (`str`) – Region to copy the node from.
- **`image_id`** (`str`) – Image to copy.
- **`name`** (`str`) – name for new image.
- **`profile`** (`str`) – The profile key
- **`description`** – description for new image.
- **`libcloud_kwargs`** (`dict`) – Extra arguments for the driver's copy_image method

CLI Example:

```
salt myminion libcloud_compute.copy_image us-east1 image1 'new image' profile1
```

`salt.modules.libcloud_compute.`**`create_image`**(*node_id, name, profile, description=None, **libcloud_kwargs*)

Create an image from a node

Parameters

- **`node_id`** (`str`) – Node to run the task on.
- **`name`** (`str`) – name for new image.
- **`profile`** (`str`) – The profile key
- **`description`** (`description`) – description for new image.
- **`libcloud_kwargs`** (`dict`) – Extra arguments for the driver's create_image method

CLI Example:

```
salt myminion libcloud_compute.create_image server1 my_image profile1
salt myminion libcloud_compute.create_image server1 my_image profile1 description=
↪'test image'
```

`salt.modules.libcloud_compute.`**`create_key_pair`**(*name, profile, **libcloud_kwargs*)

Create a single key pair by name

Parameters

- **`name`** (`str`) – Name of the key pair to create.
- **`profile`** (`str`) – The profile key
- **`libcloud_kwargs`** (`dict`) – Extra arguments for the driver's create_key_pair method

CLI Example:

```
salt myminion libcloud_compute.create_key_pair pair1 profile1
```

`salt.modules.libcloud_compute.`**`create_volume`**(*size, name, profile, location_id=None, **libcloud_kwargs*)

Create a storage volume

Parameters

- **size** (int) – Size of volume in gigabytes (required)
- **name** (str) – Name of the volume to be created
- **location_id** (str) – Which data center to create a volume in. If empty, undefined behavior will be selected. (optional)
- **profile** (str) – The profile key
- **libcloud_kwargs** (dict) – Extra arguments for the driver's list_volumes method

CLI Example:

```
salt myminion libcloud_compute.create_volume 1000 vol1 profile1
```

`salt.modules.libcloud_compute.`**`create_volume_snapshot`**`(`*volume_id, profile, name=None, **libcloud_kwargs*`)`

Create a storage volume snapshot

Parameters

- **volume_id** (str) – Volume ID from which to create the new snapshot.
- **profile** (str) – The profile key
- **name** (str) – Name of the snapshot to be created (optional)
- **libcloud_kwargs** (dict) – Extra arguments for the driver's create_volume_snapshot method

CLI Example:

```
salt myminion libcloud_compute.create_volume_snapshot vol1 profile1
```

`salt.modules.libcloud_compute.`**`delete_image`**`(`*image_id, profile, **libcloud_kwargs*`)`

Delete an image of a node

Parameters

- **image_id** (str) – Image to delete
- **profile** (str) – The profile key
- **libcloud_kwargs** (dict) – Extra arguments for the driver's delete_image method

CLI Example:

```
salt myminion libcloud_compute.delete_image image1 profile1
```

`salt.modules.libcloud_compute.`**`delete_key_pair`**`(`*name, profile, **libcloud_kwargs*`)`

Delete a key pair

Parameters

- **name** (str) – Key pair name.
- **profile** (str) – The profile key
- **libcloud_kwargs** (dict) – Extra arguments for the driver's import_key_pair_from_xxx method

CLI Example:

```
salt myminion libcloud_compute.delete_key_pair pair1 profile1
```

`salt.modules.libcloud_compute.`**`destroy_node`**`(`*node_id, profile, **libcloud_kwargs*`)`

Destroy a node in the cloud

Parameters

- **node_id** (`str`) – Unique ID of the node to destroy
- **profile** (`str`) – The profile key
- **libcloud_kwargs** (`dict`) – Extra arguments for the driver's destroy_node method

CLI Example:

```
salt myminion libcloud_compute.destry_node as-2346 profile1
```

`salt.modules.libcloud_compute.`**`destroy_volume`**(*volume_id*, *profile*, ***libcloud_kwargs*)

Destroy a volume.

Parameters

- **volume_id** (`str`) – Volume ID from which to destroy
- **profile** (`str`) – The profile key
- **libcloud_kwargs** (`dict`) – Extra arguments for the driver's destroy_volume method

CLI Example:

```
salt myminion libcloud_compute.destroy_volume vol1 profile1
```

`salt.modules.libcloud_compute.`**`destroy_volume_snapshot`**(*volume_id*, *snapshot_id*, *profile*, ***libcloud_kwargs*)

Destroy a volume snapshot.

Parameters

- **volume_id** (`str`) – Volume ID from which the snapshot belongs
- **snapshot_id** (`str`) – Volume Snapshot ID from which to destroy
- **profile** (`str`) – The profile key
- **libcloud_kwargs** (`dict`) – Extra arguments for the driver's destroy_volume_snapshot method

CLI Example:

```
salt myminion libcloud_compute.destroy_volume_snapshot snap1 profile1
```

`salt.modules.libcloud_compute.`**`detach_volume`**(*volume_id*, *profile*, ***libcloud_kwargs*)

Detaches a volume from a node.

Parameters

- **volume_id** (`str`) – Volume ID from which to detach
- **profile** (`str`) – The profile key
- **libcloud_kwargs** (`dict`) – Extra arguments for the driver's detach_volume method

CLI Example:

```
salt myminion libcloud_compute.detach_volume vol1 profile1
```

`salt.modules.libcloud_compute.`**`extra`**(*method*, *profile*, ***libcloud_kwargs*)

Call an extended method on the driver

Parameters

- **method** (`str`) – Driver's method name
- **profile** (`str`) – The profile key

- **libcloud_kwargs** (dict) – Extra arguments for the driver's method

CLI Example:

```
salt myminion libcloud_compute.extra ex_get_permissions google container_name=my_
↪container object_name=me.jpg --out=yaml
```

salt.modules.libcloud_compute.**get_image**(*image_id*, *profile*, ***libcloud_kwargs*)

Get an image of a node

Parameters

- **image_id** (str) – Image to fetch
- **profile** (str) – The profile key
- **libcloud_kwargs** (dict) – Extra arguments for the driver's delete_image method

CLI Example:

```
salt myminion libcloud_compute.get_image image1 profile1
```

salt.modules.libcloud_compute.**get_key_pair**(*name*, *profile*, ***libcloud_kwargs*)

Get a single key pair by name

Parameters

- **name** (str) – Name of the key pair to retrieve.
- **profile** (str) – The profile key
- **libcloud_kwargs** (dict) – Extra arguments for the driver's get_key_pair method

CLI Example:

```
salt myminion libcloud_compute.get_key_pair pair1 profile1
```

salt.modules.libcloud_compute.**import_key_pair**(*name*, *key*, *profile*, *key_type=None*, ***libcloud_kwargs*)

Import a new public key from string or a file path

Parameters

- **name** (str) – Key pair name.
- **key** (str or path str) – Public key material, the string or a path to a file
- **profile** (str) – The profile key
- **key_type** (str) – The key pair type, either *FILE* or *STRING*. Will detect if not provided and assume that if the string is a path to an existing path it is a FILE, else STRING.
- **libcloud_kwargs** (dict) – Extra arguments for the driver's import_key_pair_from_xxx method

CLI Example:

```
salt myminion libcloud_compute.import_key_pair pair1 key_value_data123 profile1
salt myminion libcloud_compute.import_key_pair pair1 /path/to/key profile1
```

salt.modules.libcloud_compute.**list_images**(*profile*, *location_id=None*, ***libcloud_kwargs*)

Return a list of images for this cloud

Parameters

- **profile** (str) – The profile key

- **location_id** (`str`) – The location key, from list_locations
- **libcloud_kwargs** (`dict`) – Extra arguments for the driver's list_images method

CLI Example:

```
salt myminion libcloud_compute.list_images profile1
```

`salt.modules.libcloud_compute.`**list_key_pairs**(*profile*, ***libcloud_kwargs*)

List all the available key pair objects.

Parameters

- **profile** (`str`) – The profile key
- **libcloud_kwargs** (`dict`) – Extra arguments for the driver's list_key_pairs method

CLI Example:

```
salt myminion libcloud_compute.list_key_pairs profile1
```

`salt.modules.libcloud_compute.`**list_locations**(*profile*, ***libcloud_kwargs*)

Return a list of locations for this cloud

Parameters

- **profile** (`str`) – The profile key
- **libcloud_kwargs** (`dict`) – Extra arguments for the driver's list_locations method

CLI Example:

```
salt myminion libcloud_compute.list_locations profile1
```

`salt.modules.libcloud_compute.`**list_nodes**(*profile*, ***libcloud_kwargs*)

Return a list of nodes

Parameters

- **profile** (`str`) – The profile key
- **libcloud_kwargs** (`dict`) – Extra arguments for the driver's list_nodes method

CLI Example:

```
salt myminion libcloud_compute.list_nodes profile1
```

`salt.modules.libcloud_compute.`**list_sizes**(*profile*, *location_id=None*, ***libcloud_kwargs*)

Return a list of node sizes

Parameters

- **profile** (`str`) – The profile key
- **location_id** (`str`) – The location key, from list_locations
- **libcloud_kwargs** (`dict`) – Extra arguments for the driver's list_sizes method

CLI Example:

```
salt myminion libcloud_compute.list_sizes profile1
salt myminion libcloud_compute.list_sizes profile1 us-east1
```

`salt.modules.libcloud_compute.`**list_volume_snapshots**(*volume_id*, *profile*, ***libcloud_kwargs*)

Return a list of storage volumes snapshots for this cloud

Parameters

- **volume_id** (str) – The volume identifier
- **profile** (str) – The profile key
- **libcloud_kwargs** (dict) – Extra arguments for the driver's list_volume_snapshots method

CLI Example:

```
salt myminion libcloud_compute.list_volume_snapshots vol1 profile1
```

`salt.modules.libcloud_compute.`**`list_volumes`**(*profile*, ***libcloud_kwargs*)

Return a list of storage volumes for this cloud

Parameters

- **profile** (str) – The profile key
- **libcloud_kwargs** (dict) – Extra arguments for the driver's list_volumes method

CLI Example:

```
salt myminion libcloud_compute.list_volumes profile1
```

`salt.modules.libcloud_compute.`**`reboot_node`**(*node_id*, *profile*, ***libcloud_kwargs*)

Reboot a node in the cloud

Parameters

- **node_id** (str) – Unique ID of the node to reboot
- **profile** (str) – The profile key
- **libcloud_kwargs** (dict) – Extra arguments for the driver's reboot_node method

CLI Example:

```
salt myminion libcloud_compute.reboot_node as-2346 profile1
```

21.9.202 salt.modules.libcloud_dns module

Apache Libcloud DNS Management

Connection module for Apache Libcloud DNS management

New in version 2016.11.0.

configuration This module uses a configuration profile for one or multiple DNS providers

```
libcloud_dns:
    profile_test1:
      driver: cloudflare
      key: 12345
      secret: mysecret
    profile_test2:
      driver: godaddy
      key: 12345
      secret: mysecret
      shopper_id: 12345
```

depends apache-libcloud

`salt.modules.libcloud_dns.`**`create_record`**(*name*, *zone_id*, *type*, *data*, *profile*)

Create a new record.

Parameters

- **name** (`str`) – Record name without the domain name (e.g. www). Note: If you want to create a record for a base domain name, you should specify empty string (") for this argument.
- **zone_id** (`str`) – Zone where the requested record is created.
- **type** (`str`) – DNS record type (A, AAAA, …).
- **data** (`str`) – Data for the record (depends on the record type).
- **profile** (`str`) – The profile key

CLI Example:

```
salt myminion libcloud_dns.create_record www google.com A 12.32.12.2 profile1
```

`salt.modules.libcloud_dns.`**`create_zone`**(*domain*, *profile*, *type='master'*, *ttl=None*)

Create a new zone.

Parameters

- **domain** (`str`) – Zone domain name (e.g. example.com)
- **profile** (`str`) – The profile key
- **type** (`str`) – Zone type (master / slave).
- **ttl** (`int`) – TTL for new records. (optional)

CLI Example:

```
salt myminion libcloud_dns.create_zone google.com profile1
```

`salt.modules.libcloud_dns.`**`delete_record`**(*zone_id*, *record_id*, *profile*)

Delete a record.

Parameters

- **zone_id** (`str`) – Zone to delete.
- **record_id** (`str`) – Record to delete.
- **profile** (`str`) – The profile key

Return type `bool`

CLI Example:

```
salt myminion libcloud_dns.delete_record google.com www profile1
```

`salt.modules.libcloud_dns.`**`delete_zone`**(*zone_id*, *profile*)

Delete a zone.

Parameters

- **zone_id** (`str`) – Zone to delete.
- **profile** (`str`) – The profile key

Return type `bool`

CLI Example:

```
salt myminion libcloud_dns.delete_zone google.com profile1
```

`salt.modules.libcloud_dns.`**`extra`**(*method*, *profile*, ***libcloud_kwargs*)

Call an extended method on the driver

Parameters

- **method** (`str`) – Driver's method name

- **profile** (str) – The profile key
- **libcloud_kwargs** (dict) – Extra arguments for the driver's delete_container method

CLI Example:

```
salt myminion libcloud_dns.extra ex_get_permissions google container_name=my_
↪container object_name=me.jpg --out=yaml
```

salt.modules.libcloud_dns.**get_bind_data**(*zone_id*, *profile*)

Export Zone to the BIND compatible format.

Parameters

- **zone_id** (str) – Zone to export.
- **profile** (str) – The profile key

Returns Zone data in BIND compatible format.

Return type str

CLI Example:

```
salt myminion libcloud_dns.get_bind_data google.com profile1
```

salt.modules.libcloud_dns.**get_record**(*zone_id*, *record_id*, *profile*)

Get record information for the given zone_id on the given profile

Parameters

- **zone_id** (str) – Zone to export.
- **record_id** (str) – Record to delete.
- **profile** (str) – The profile key

CLI Example:

```
salt myminion libcloud_dns.get_record google.com www profile1
```

salt.modules.libcloud_dns.**get_zone**(*zone_id*, *profile*)

Get zone information for the given zone_id on the given profile

Parameters

- **zone_id** (str) – Zone to export.
- **profile** (str) – The profile key

CLI Example:

```
salt myminion libcloud_dns.get_zone google.com profile1
```

salt.modules.libcloud_dns.**list_record_types**(*profile*)

List available record types for the given profile, e.g. A, AAAA

Parameters **profile** (str) – The profile key

CLI Example:

```
salt myminion libcloud_dns.list_record_types profile1
```

salt.modules.libcloud_dns.**list_records**(*zone_id*, *profile*, *type=None*)

List records for the given zone_id on the given profile

Parameters

- **zone_id** (str) – Zone to export.
- **profile** (str) – The profile key

- **type** (str) – The record type, e.g. A, NS

CLI Example:

```
salt myminion libcloud_dns.list_records google.com profile1
```

`salt.modules.libcloud_dns.`**`list_zones`**(*profile*)

List zones for the given profile

Parameters **profile** (str) – The profile key

CLI Example:

```
salt myminion libcloud_dns.list_zones profile1
```

`salt.modules.libcloud_dns.`**`update_zone`**(*zone_id, domain, profile, type='master', ttl=None*)

Update an existing zone.

Parameters

- **zone_id** (str) – Zone ID to update.
- **domain** (str) – Zone domain name (e.g. example.com)
- **profile** (str) – The profile key
- **type** (str) – Zone type (master / slave).
- **ttl** (int) – TTL for new records. (optional)

CLI Example:

```
salt myminion libcloud_dns.update_zone google.com google.com profile1 type=slave
```

21.9.203 salt.modules.libcloud_loadbalancer

Apache Libcloud Load Balancer Management

Connection module for Apache Libcloud Storage load balancer management for a full list of supported clouds, see http://libcloud.readthedocs.io/en/latest/loadbalancer/supported_providers.html

Clouds include Amazon ELB, ALB, Google, Aliyun, CloudStack, Softlayer

New in version 2018.3.0.

configuration This module uses a configuration profile for one or multiple Storage providers

```
libcloud_loadbalancer:
    profile_test1:
      driver: gce
      key: GOOG0123456789ABCXYZ
      secret: mysecret
    profile_test2:
      driver: alb
      key: 12345
      secret: mysecret
```

depends apache-libcloud

`salt.modules.libcloud_loadbalancer.`**`balancer_attach_member`**(*balancer_id, ip, port, profile, extra=None, **libcloud_kwargs*)

Add a new member to the load balancer

Parameters

- **balancer_id** (str) – id of a load balancer you want to fetch
- **ip** (str) – IP address for the new member
- **port** (int) – Port for the new member
- **profile** (str) – The profile key
- **libcloud_kwargs** (dict) – Extra arguments for the driver's balancer_attach_member method

CLI Example:

```
salt myminion libcloud_storage.balancer_attach_member balancer123 1.2.3.4 80␣
↪profile1
```

salt.modules.libcloud_loadbalancer.**balancer_detach_member**(*balancer_id*, *member_id*, *profile*, ***libcloud_kwargs*)

Add a new member to the load balancer

Parameters

- **balancer_id** (str) – id of a load balancer you want to fetch
- **ip** (str) – IP address for the new member
- **port** (int) – Port for the new member
- **profile** (str) – The profile key
- **libcloud_kwargs** (dict) – Extra arguments for the driver's balancer_detach_member method

CLI Example:

```
salt myminion libcloud_storage.balancer_detach_member balancer123 member123␣
↪profile1
```

salt.modules.libcloud_loadbalancer.**create_balancer**(*name*, *port*, *protocol*, *profile*, *algorithm=None*, *members=None*, ***libcloud_kwargs*)

Create a new load balancer instance

Parameters

- **name** (str) – Name of the new load balancer (required)
- **port** (str) – Port the load balancer should listen on, defaults to 80
- **protocol** (str) – Loadbalancer protocol, defaults to http.
- **algorithm** (str) – Load balancing algorithm, defaults to ROUND_ROBIN. See Algorithm type in Libcloud documentation for a full listing.
- **profile** (str) – The profile key
- **libcloud_kwargs** (dict) – Extra arguments for the driver's create_balancer method

Returns The details of the new balancer

CLI Example:

```
salt myminion libcloud_storage.create_balancer my_balancer 80 http profile1
```

salt.modules.libcloud_loadbalancer.**destroy_balancer**(*balancer_id*, *profile*, ***libcloud_kwargs*)

Destroy a load balancer

Parameters

- **balancer_id** (str) – LoadBalancer ID which should be used
- **profile** (str) – The profile key
- **libcloud_kwargs** (dict) – Extra arguments for the driver's destroy_balancer method

Returns True if the destroy was successful, otherwise False.

Return type bool

CLI Example:

```
salt myminion libcloud_storage.destroy_balancer balancer_1 profile1
```

salt.modules.libcloud_loadbalancer.**extra**(*method*, *profile*, ***libcloud_kwargs*)

Call an extended method on the driver

Parameters

- **method** (str) – Driver's method name
- **profile** (str) – The profile key
- **libcloud_kwargs** (dict) – Extra arguments for the driver's method

CLI Example:

```
salt myminion libcloud_loadbalancer.extra ex_get_permissions google container_
↪name=my_container object_name=me.jpg --out=yaml
```

salt.modules.libcloud_loadbalancer.**get_balancer**(*balancer_id*, *profile*, ***libcloud_kwargs*)

Get the details for a load balancer by ID

Parameters

- **balancer_id** (str) – id of a load balancer you want to fetch
- **profile** (str) – The profile key
- **libcloud_kwargs** (dict) – Extra arguments for the driver's get_balancer method

Returns the load balancer details

CLI Example:

```
salt myminion libcloud_storage.get_balancer balancer123 profile1
```

salt.modules.libcloud_loadbalancer.**get_balancer_by_name**(*name*, *profile*, ***libcloud_kwargs*)

Get the details for a load balancer by name

Parameters

- **name** (str) – Name of a load balancer you want to fetch
- **profile** (str) – The profile key
- **libcloud_kwargs** (dict) – Extra arguments for the driver's list_balancers method

Returns the load balancer details

CLI Example:

```
salt myminion libcloud_storage.get_balancer_by_name my_balancer profile1
```

`salt.modules.libcloud_loadbalancer.`**`list_balancer_members`**(*balancer_id*, *profile*, ***libcloud_kwargs*)

List the members of a load balancer

Parameters

- **`balancer_id`** (`str`) – id of a load balancer you want to fetch
- **`profile`** (`str`) – The profile key
- **`libcloud_kwargs`** (`dict`) – Extra arguments for the driver's list_balancer_members method

CLI Example:

```
salt myminion libcloud_storage.list_balancer_members balancer123 profile1
```

`salt.modules.libcloud_loadbalancer.`**`list_balancers`**(*profile*, ***libcloud_kwargs*)

Return a list of load balancers.

Parameters

- **`profile`** (`str`) – The profile key
- **`libcloud_kwargs`** (`dict`) – Extra arguments for the driver's list_balancers method

CLI Example:

```
salt myminion libcloud_storage.list_balancers profile1
```

`salt.modules.libcloud_loadbalancer.`**`list_protocols`**(*profile*, ***libcloud_kwargs*)

Return a list of supported protocols.

Parameters

- **`profile`** (`str`) – The profile key
- **`libcloud_kwargs`** (`dict`) – Extra arguments for the driver's list_protocols method

Returns a list of supported protocols

Return type `list` of `str`

CLI Example:

```
salt myminion libcloud_storage.list_protocols profile1
```

`salt.modules.libcloud_loadbalancer.`**`list_supported_algorithms`**(*profile*, ***libcloud_kwargs*)

Get the supported algorithms for a profile

Parameters

- **`profile`** (`str`) – The profile key
- **`libcloud_kwargs`** (`dict`) – Extra arguments for the driver's list_supported_algorithms method

Returns The supported algorithms

CLI Example:

```
salt myminion libcloud_storage.list_supported_algorithms profile1
```

21.9.204 salt.modules.libcloud_storage

Apache Libcloud Storage Management

Connection module for Apache Libcloud Storage (object/blob) management for a full list of supported clouds, see http://libcloud.readthedocs.io/en/latest/storage/supported_providers.html

Clouds include Amazon S3, Google Storage, Aliyun, Azure Blobs, Ceph, OpenStack swift

New in version 2018.3.0.

configuration This module uses a configuration profile for one or multiple Storage providers

```
libcloud_storage:
    profile_test1:
      driver: google_storage
      key: GOOG0123456789ABCXYZ
      secret: mysecret
    profile_test2:
      driver: s3
      key: 12345
      secret: mysecret
```

depends apache-libcloud

`salt.modules.libcloud_storage.`**`create_container`**(*container_name*, *profile*, ***libcloud_kwargs*)

Create a container in the cloud

Parameters

- **`container_name`** (`str`) – Container name
- **`profile`** (`str`) – The profile key
- **`libcloud_kwargs`** (`dict`) – Extra arguments for the driver's create_container method

CLI Example:

```
salt myminion libcloud_storage.create_container MyFolder profile1
```

`salt.modules.libcloud_storage.`**`delete_container`**(*container_name*, *profile*, ***libcloud_kwargs*)

Delete an object container in the cloud

Parameters

- **`container_name`** (`str`) – Container name
- **`profile`** (`str`) – The profile key
- **`libcloud_kwargs`** (`dict`) – Extra arguments for the driver's delete_container method

Returns True if an object container has been successfully deleted, False otherwise.

Return type `bool`

CLI Example:

```
salt myminion libcloud_storage.delete_container MyFolder profile1
```

`salt.modules.libcloud_storage.`**`delete_object`**(*container_name*, *object_name*, *profile*, ***libcloud_kwargs*)

Delete an object in the cloud

Parameters

- **`container_name`** (`str`) – Container name

- **`object_name`** (`str`) – Object name
- **`profile`** (`str`) – The profile key
- **`libcloud_kwargs`** (`dict`) – Extra arguments for the driver's delete_object method

Returns True if an object has been successfully deleted, False otherwise.

Return type `bool`

CLI Example:

```
salt myminion libcloud_storage.delete_object MyFolder me.jpg profile1
```

`salt.modules.libcloud_storage.`**`download_object`**(*container_name, object_name, destination_path, profile, overwrite_existing=False, delete_on_failure=True, **libcloud_kwargs*)

Download an object to the specified destination path.

Parameters

- **`container_name`** (`str`) – Container name
- **`object_name`** (`str`) – Object name
- **`destination_path`** (`str`) – Full path to a file or a directory where the incoming file will be saved.
- **`profile`** (`str`) – The profile key
- **`overwrite_existing`** (`bool`) – True to overwrite an existing file, defaults to False.
- **`delete_on_failure`** (`bool`) – True to delete a partially downloaded file if the download was not successful (hash mismatch / file size).
- **`libcloud_kwargs`** (`dict`) – Extra arguments for the driver's download_object method

Returns True if an object has been successfully downloaded, False otherwise.

Return type `bool`

CLI Example:

```
salt myminion libcloud_storage.download_object MyFolder me.jpg /tmp/me.jpg
↪profile1
```

`salt.modules.libcloud_storage.`**`extra`**(*method, profile, **libcloud_kwargs*)

Call an extended method on the driver

Parameters

- **`method`** (`str`) – Driver's method name
- **`profile`** (`str`) – The profile key
- **`libcloud_kwargs`** (`dict`) – Extra arguments for the driver's delete_container method

CLI Example:

```
salt myminion libcloud_storage.extra ex_get_permissions google container_name=my_
↪container object_name=me.jpg --out=yaml
```

`salt.modules.libcloud_storage.`**`get_container`**(*container_name, profile, **libcloud_kwargs*)

List container details for the given container_name on the given profile

Parameters

- **container_name** (`str`) – Container name
- **profile** (`str`) – The profile key
- **libcloud_kwargs** (`dict`) – Extra arguments for the driver's get_container method

CLI Example:

```
salt myminion libcloud_storage.get_container MyFolder profile1
```

`salt.modules.libcloud_storage.`**`get_container_object`**`(`*container_name*, *object_name*, *profile*, ***libcloud_kwargs*`)`

Get the details for a container object (file or object in the cloud)

Parameters

- **container_name** (`str`) – Container name
- **object_name** (`str`) – Object name
- **profile** (`str`) – The profile key
- **libcloud_kwargs** (`dict`) – Extra arguments for the driver's get_container_object method

CLI Example:

```
salt myminion libcloud_storage.get_container_object MyFolder MyFile.xyz profile1
```

`salt.modules.libcloud_storage.`**`list_container_objects`**`(`*container_name*, *profile*, ***libcloud_kwargs*`)`

List container objects (e.g. files) for the given container_id on the given profile

Parameters

- **container_name** (`str`) – Container name
- **profile** (`str`) – The profile key
- **libcloud_kwargs** (`dict`) – Extra arguments for the driver's list_container_objects method

CLI Example:

```
salt myminion libcloud_storage.list_container_objects MyFolder profile1
```

`salt.modules.libcloud_storage.`**`list_containers`**`(`*profile*, ***libcloud_kwargs*`)`

Return a list of containers.

Parameters

- **profile** (`str`) – The profile key
- **libcloud_kwargs** (`dict`) – Extra arguments for the driver's list_containers method

CLI Example:

```
salt myminion libcloud_storage.list_containers profile1
```

`salt.modules.libcloud_storage.`**`upload_object`**`(`*file_path*, *container_name*, *object_name*, *profile*, *extra=None*, *verify_hash=True*, *headers=None*, ***libcloud_kwargs*`)`

Upload an object currently located on a disk.

Parameters

- **file_path** (`str`) – Path to the object on disk.
- **container_name** (`str`) – Destination container.
- **object_name** (`str`) – Object name.
- **profile** (`str`) – The profile key
- **verify_hash** (`bool`) – Verify hash
- **extra** (`dict`) – Extra attributes (driver specific). (optional)
- **headers** (`dict`) – (optional) Additional request headers, such as CORS headers. For example: headers = {'Access-Control-Allow-Origin': 'http://mozilla.com'}
- **libcloud_kwargs** (`dict`) – Extra arguments for the driver's upload_object method

Returns The object name in the cloud

Return type `str`

CLI Example:

```
salt myminion libcloud_storage.upload_object /file/to/me.jpg MyFolder me.jpg␣
↪profile1
```

21.9.205 salt.modules.linux_acl

Support for Linux File Access Control Lists

The Linux ACL module requires the *getfacl* and *setfacl* binaries.

`salt.modules.linux_acl.`**`delfacl`**`(`*acl_type, acl_name='', *args, **kwargs*`)`

Remove specific FACL from the specified file(s)

CLI Examples:

```
salt '*' acl.delfacl user myuser /tmp/house/kitchen
salt '*' acl.delfacl default:group mygroup /tmp/house/kitchen
salt '*' acl.delfacl d:u myuser /tmp/house/kitchen
salt '*' acl.delfacl g myuser /tmp/house/kitchen /tmp/house/livingroom
salt '*' acl.delfacl user myuser /tmp/house/kitchen recursive=True
```

`salt.modules.linux_acl.`**`getfacl`**`(`**args, **kwargs*`)`

Return (extremely verbose) map of FACLs on specified file(s)

CLI Examples:

```
salt '*' acl.getfacl /tmp/house/kitchen
salt '*' acl.getfacl /tmp/house/kitchen /tmp/house/livingroom
salt '*' acl.getfacl /tmp/house/kitchen /tmp/house/livingroom recursive=True
```

`salt.modules.linux_acl.`**`modfacl`**`(`*acl_type, acl_name='', perms='', *args, **kwargs*`)`

Add or modify a FACL for the specified file(s)

CLI Examples:

```
salt '*' acl.modfacl user myuser rwx /tmp/house/kitchen
salt '*' acl.modfacl default:group mygroup rx /tmp/house/kitchen
salt '*' acl.modfacl d:u myuser 7 /tmp/house/kitchen
salt '*' acl.modfacl g mygroup 0 /tmp/house/kitchen /tmp/house/livingroom
```

(continues on next page)

(continued from previous page)

```
salt '*' acl.modfacl user myuser rwx /tmp/house/kitchen recursive=True
salt '*' acl.modfacl user myuser rwx /tmp/house/kitchen raise_err=True
```

salt.modules.linux_acl.**version**()

Return facl version from getfacl –version

CLI Example:

```
salt '*' acl.version
```

salt.modules.linux_acl.**wipefacls**(**args*, ***kwargs*)

Remove all FACLs from the specified file(s)

CLI Examples:

```
salt '*' acl.wipefacls /tmp/house/kitchen
salt '*' acl.wipefacls /tmp/house/kitchen /tmp/house/livingroom
salt '*' acl.wipefacls /tmp/house/kitchen /tmp/house/livingroom recursive=True
```

21.9.206 salt.modules.linux_ip module

The networking module for Non-RH/Deb Linux distros

salt.modules.linux_ip.**down**(*iface*, *iface_type=None*)

Shutdown a network interface

CLI Example:

```
salt '*' ip.down eth0
```

salt.modules.linux_ip.**get_interface**(*iface*)

Return the contents of an interface script

CLI Example:

```
salt '*' ip.get_interface eth0
```

salt.modules.linux_ip.**get_routes**(*iface=None*)

Return the current routing table

CLI Examples:

```
salt '*' ip.get_routes
salt '*' ip.get_routes eth0
```

salt.modules.linux_ip.**up**(*iface*, *iface_type=None*)

Start up a network interface

CLI Example:

```
salt '*' ip.up eth0
```

21.9.207 salt.modules.linux_lvm

Support for Linux LVM2

`salt.modules.linux_lvm.`**`fullversion`**`()`

Return all version info from lvm version

CLI Example:

```
salt '*' lvm.fullversion
```

`salt.modules.linux_lvm.`**`lvcreate`**`(`*lvname, vgname, size=None, extents=None, snapshot=None, pv=None, thinvolume=False, thinpool=False, force=False, **kwargs*`)`

Create a new logical volume, with option for which physical volume to be used

CLI Examples:

```
salt '*' lvm.lvcreate new_volume_name     vg_name size=10G
salt '*' lvm.lvcreate new_volume_name     vg_name extents=100 pv=/dev/sdb
salt '*' lvm.lvcreate new_snapshot        vg_name snapshot=volume_name size=3G
```

New in version to_complete.

Support for thin pools and thin volumes

CLI Examples:

```
salt '*' lvm.lvcreate new_thinpool_name   vg_name               size=20G
↪thinpool=True
salt '*' lvm.lvcreate new_thinvolume_name vg_name/thinpool_name size=10G
↪thinvolume=True
```

`salt.modules.linux_lvm.`**`lvdisplay`**`(`*lvname='', quiet=False*`)`

Return information about the logical volume(s)

lvname logical device name

quiet if the logical volume is not present, do not show any error

CLI Examples:

```
salt '*' lvm.lvdisplay
salt '*' lvm.lvdisplay /dev/vg_myserver/root
```

`salt.modules.linux_lvm.`**`lvremove`**`(`*lvname, vgname*`)`

Remove a given existing logical volume from a named existing volume group

CLI Example:

```
salt '*' lvm.lvremove lvname vgname force=True
```

`salt.modules.linux_lvm.`**`lvresize`**`(`*size=None, lvpath=None, extents=None*`)`

Return information about the logical volume(s)

CLI Examples:

```
salt '*' lvm.lvresize +12M /dev/mapper/vg1-test
salt '*' lvm.lvresize lvpath=/dev/mapper/vg1-test extents=+100%FREE
```

`salt.modules.linux_lvm.`**`pvcreate`**`(`*devices, override=True, **kwargs*`)`

Set a physical device to be used as an LVM physical volume

override Skip devices, if they are already LVM physical volumes

CLI Examples:

```
salt mymachine lvm.pvcreate /dev/sdb1,/dev/sdb2
salt mymachine lvm.pvcreate /dev/sdb1 dataalignmentoffset=7s
```

`salt.modules.linux_lvm.`**`pvdisplay`**`(`*pvname=”, real=False, quiet=False*`)`

Return information about the physical volume(s)

pvname physical device name

real dereference any symlinks and report the real device

New in version 2015.8.7.

quiet if the physical volume is not present, do not show any error

CLI Examples:

```
salt '*' lvm.pvdisplay
salt '*' lvm.pvdisplay /dev/md0
```

`salt.modules.linux_lvm.`**`pvremove`**`(`*devices, override=True*`)`

Remove a physical device being used as an LVM physical volume

override Skip devices, if they are already not used as LVM physical volumes

CLI Examples:

```
salt mymachine lvm.pvremove /dev/sdb1,/dev/sdb2
```

`salt.modules.linux_lvm.`**`version`**`()`

Return LVM version from lvm version

CLI Example:

```
salt '*' lvm.version
```

`salt.modules.linux_lvm.`**`vgcreate`**`(`*vgname, devices, **kwargs*`)`

Create an LVM volume group

CLI Examples:

```
salt mymachine lvm.vgcreate my_vg /dev/sdb1,/dev/sdb2
salt mymachine lvm.vgcreate my_vg /dev/sdb1 clustered=y
```

`salt.modules.linux_lvm.`**`vgdisplay`**`(`*vgname=”, quiet=False*`)`

Return information about the volume group(s)

vgname volume group name

quiet if the volume group is not present, do not show any error

CLI Examples:

```
salt '*' lvm.vgdisplay
salt '*' lvm.vgdisplay nova-volumes
```

`salt.modules.linux_lvm.`**`vgextend`**`(`*vgname, devices*`)`

Add physical volumes to an LVM volume group

CLI Examples:

```
salt mymachine lvm.vgextend my_vg /dev/sdb1,/dev/sdb2
salt mymachine lvm.vgextend my_vg /dev/sdb1
```

`salt.modules.linux_lvm.`**`vgremove`**`(`*vgname*`)`

Remove an LVM volume group

CLI Examples:

```
salt mymachine lvm.vgremove vgname
salt mymachine lvm.vgremove vgname force=True
```

21.9.208 salt.modules.linux_sysctl

Module for viewing and modifying sysctl parameters

`salt.modules.linux_sysctl.`**`assign`**`(`*name*, *value*`)`

Assign a single sysctl parameter for this minion

CLI Example:

```
salt '*' sysctl.assign net.ipv4.ip_forward 1
```

`salt.modules.linux_sysctl.`**`default_config`**`()`

Linux hosts using systemd 207 or later ignore `/etc/sysctl.conf` and only load from `/etc/sysctl.d/*.conf`. This function will do the proper checks and return a default config file which will be valid for the Minion. Hosts running systemd >= 207 will use `/etc/sysctl.d/99-salt.conf`.

CLI Example:

```
salt -G 'kernel:Linux' sysctl.default_config
```

`salt.modules.linux_sysctl.`**`get`**`(`*name*`)`

Return a single sysctl parameter for this minion

CLI Example:

```
salt '*' sysctl.get net.ipv4.ip_forward
```

`salt.modules.linux_sysctl.`**`persist`**`(`*name*, *value*, *config=None*`)`

Assign and persist a simple sysctl parameter for this minion. If `config` is not specified, a sensible default will be chosen using *`sysctl.default_config`*.

CLI Example:

```
salt '*' sysctl.persist net.ipv4.ip_forward 1
```

`salt.modules.linux_sysctl.`**`show`**`(`*config_file=False*`)`

Return a list of sysctl parameters for this minion

config: Pull the data from the system configuration file instead of the live data.

CLI Example:

```
salt '*' sysctl.show
```

21.9.209 salt.modules.localemod

Module for managing locales on POSIX-like systems.

`salt.modules.localemod.`**`avail`**`(`*locale*`)`

Check if a locale is available.

New in version 2014.7.0.

CLI Example:

```
salt '*' locale.avail 'en_US.UTF-8'
```

`salt.modules.localemod.`**`gen_locale`**(*locale*, ***kwargs*)

Generate a locale. Options:

New in version 2014.7.0.

Parameters **`locale`** – Any locale listed in /usr/share/i18n/locales or /usr/share/i18n/SUPPORTED for Debian and Gentoo based distributions, which require the charmap to be specified as part of the locale when generating it.

verbose Show extra warnings about errors that are normally ignored.

CLI Example:

```
salt '*' locale.gen_locale en_US.UTF-8
salt '*' locale.gen_locale 'en_IE.UTF-8 UTF-8'    # Debian/Gentoo only
```

`salt.modules.localemod.`**`get_locale`**()

Get the current system locale

CLI Example:

```
salt '*' locale.get_locale
```

`salt.modules.localemod.`**`list_avail`**()

Lists available (compiled) locales

CLI Example:

```
salt '*' locale.list_avail
```

`salt.modules.localemod.`**`set_locale`**(*locale*)

Sets the current system locale

CLI Example:

```
salt '*' locale.set_locale 'en_US.UTF-8'
```

21.9.210 salt.modules.locate

Module for using the locate utilities

`salt.modules.locate.`**`locate`**(*pattern*, *database=''*, *limit=0*, ***kwargs*)

Performs a file lookup. Valid options (and their defaults) are:

```
basename=False
count=False
existing=False
follow=True
ignore=False
nofollow=False
wholename=True
regex=False
database=<locate's default database>
limit=<integer, not set by default>
```

See the manpage for `locate(1)` for further explanation of these options.

CLI Example:

```
salt '*' locate.locate
```

`salt.modules.locate.`**`stats`**`()`

Returns statistics about the locate database

CLI Example:

```
salt '*' locate.stats
```

`salt.modules.locate.`**`updatedb`**`()`

Updates the locate database

CLI Example:

```
salt '*' locate.updatedb
```

`salt.modules.locate.`**`version`**`()`

Returns the version of locate

CLI Example:

```
salt '*' locate.version
```

21.9.211 salt.modules.logadm

Module for managing Solaris logadm based log rotations.

`salt.modules.logadm.`**`list_conf`**`(`*conf_file='/etc/logadm.conf', log_file=None, include_unset=False*`)`

Show parsed configuration

New in version 2018.3.0.

conf_file [string] path to logadm.conf, defaults to /etc/logadm.conf
log_file [string] optional show only one log file
include_unset [boolean] include unset flags in output
CLI Example:

```
salt '*' logadm.list_conf
salt '*' logadm.list_conf log=/var/log/syslog
salt '*' logadm.list_conf include_unset=False
```

`salt.modules.logadm.`**`remove`**`(`*name, conf_file='/etc/logadm.conf'*`)`

Remove log pattern from logadm

CLI Example:

```
salt '*' logadm.remove myapplog
```

`salt.modules.logadm.`**`rotate`**`(`*name, pattern=None, conf_file='/etc/logadm.conf', **kwargs*`)`

Set up pattern for logging.
name [string] alias for entryname
pattern [string] alias for log_file
conf_file [string] optional path to alternative configuration file
kwargs [boolean|string|int] optional additional flags and parameters

---

Note: `name` and `pattern` were kept for backwards compatibility reasons.

`name` is an alias for the `entryname` argument, `pattern` is an alias for `log_file`. These aliases will only be used if the `entryname` and `log_file` arguments are not passed.

For a full list of arguments see ``logadm.show_args``.

CLI Example:

```
salt '*' logadm.rotate myapplog pattern='/var/log/myapp/*.log' count=7
salt '*' logadm.rotate myapplog log_file='/var/log/myapp/*.log' count=4 owner=myappd mode='0700'
```

`salt.modules.logadm.`**`show_args`**`()`

Show which arguments map to which flags and options.

New in version 2018.3.0.

CLI Example:

```
salt '*' logadm.show_args
```

`salt.modules.logadm.`**`show_conf`**`(`*conf_file='/etc/logadm.conf', name=None*`)`

Show configuration

conf_file [string] path to logadm.conf, defaults to /etc/logadm.conf

name [string] optional show only a single entry

CLI Example:

```
salt '*' logadm.show_conf
salt '*' logadm.show_conf name=/var/log/syslog
```

21.9.212 salt.modules.logmod module

On-demand logging

New in version 2017.7.0.

The sole purpose of this module is logging messages in the (proxy) minion. It comes very handy when debugging complex Jinja templates, for example:

```
{%- for var in range(10) %}
  {%- do salt.log.info(var) -%}
{%- endfor %}
```

CLI Example:

```
salt '*' log.error "Please don't do that, this module is not for CLI use!"
```

`salt.modules.logmod.`**`critical`**`(`*message*`)`

Log message at level CRITICAL.

`salt.modules.logmod.`**`debug`**`(`*message*`)`

Log message at level DEBUG.

`salt.modules.logmod.`**`error`**`(`*message*`)`

Log message at level ERROR.

`salt.modules.logmod.`**`exception`**`(`*message*`)`

Log message at level EXCEPTION.

`salt.modules.logmod.`**`info`**(*message*)

Log message at level INFO.

`salt.modules.logmod.`**`warning`**(*message*)

Log message at level WARNING.

21.9.213 salt.modules.logrotate

Module for managing logrotate.

`salt.modules.logrotate.`**`get`**(*key, value=None, conf_file='/etc/logrotate.conf'*)

Get the value for a specific configuration line.

Parameters

- **`key`** (*`str`*) – The command or stanza block to configure.
- **`value`** (*`str`*) – The command value or command of the block specified by the key parameter.
- **`conf_file`** (*`str`*) – The logrotate configuration file.

Returns The value for a specific configuration line.

Return type bool|int|str

CLI Example:

```
salt '*' logrotate.get rotate

salt '*' logrotate.get /var/log/wtmp rotate /etc/logrotate.conf
```

`salt.modules.logrotate.`**`set_`**(*key, value, setting=None, conf_file='/etc/logrotate.conf'*)

Set a new value for a specific configuration line.

Parameters

- **`key`** (*`str`*) – The command or block to configure.
- **`value`** (*`str`*) – The command value or command of the block specified by the key parameter.
- **`setting`** (*`str`*) – The command value for the command specified by the value parameter.
- **`conf_file`** (*`str`*) – The logrotate configuration file.

Returns A boolean representing whether all changes succeeded.

Return type bool

CLI Example:

```
salt '*' logrotate.set rotate 2
```

Can also be used to set a single value inside a multiline configuration block. For instance, to change rotate in the following block:

```
/var/log/wtmp {
    monthly
    create 0664 root root
    rotate 1
}
```

Use the following command:

```
salt '*' logrotate.set /var/log/wtmp rotate 2
```

This module also has the ability to scan files inside an include directory, and make changes in the appropriate file.

`salt.modules.logrotate.`**`show_conf`**`(`*conf_file='/etc/logrotate.conf'*`)`

Show parsed configuration

Parameters **`conf_file`** (*`str`*) – The logrotate configuration file.

Returns The parsed configuration.

Return type dict

CLI Example:

```
salt '*' logrotate.show_conf
```

21.9.214 salt.modules.lvs

Support for LVS (Linux Virtual Server)

`salt.modules.lvs.`**`add_server`**`(`*protocol=None, service_address=None, server_address=None, packet_forward_method='dr', weight=1, **kwargs*`)`

Add a real server to a virtual service.

protocol The service protocol(only support `tcp`, `udp` and `fwmark` service).

service_address The LVS service address.

server_address The real server address.

packet_forward_method The LVS packet forwarding method(`dr` for direct routing, `tunnel` for tunneling, `nat` for network access translation).

weight The capacity of a server relative to the others in the pool.

CLI Example:

```
salt '*' lvs.add_server tcp 1.1.1.1:80 192.168.0.11:8080 nat 1
```

`salt.modules.lvs.`**`add_service`**`(`*protocol=None, service_address=None, scheduler='wlc'*`)`

Add a virtual service.

protocol The service protocol(only support tcp, udp and fwmark service).

service_address The LVS service address.

scheduler Algorithm for allocating TCP connections and UDP datagrams to real servers.

CLI Example:

```
salt '*' lvs.add_service tcp 1.1.1.1:80 rr
```

`salt.modules.lvs.`**`check_server`**`(`*protocol=None, service_address=None, server_address=None, **kwargs*`)`

Check the real server exists in the specified service.

CLI Example:

```
salt '*' lvs.check_server tcp 1.1.1.1:80 192.168.0.11:8080
```

`salt.modules.lvs.`**`check_service`**`(`*protocol=None, service_address=None, **kwargs*`)`

Check the virtual service exists.

CLI Example:

```
salt '*' lvs.check_service tcp 1.1.1.1:80
```

`salt.modules.lvs.`**`clear`**`()`

Clear the virtual server table

CLI Example:

```
salt '*' lvs.clear
```

`salt.modules.lvs.`**`delete_server`**`(`*protocol=None*, *service_address=None*, *server_address=None*`)`

Delete the realserver from the virtual service.

protocol The service protocol(only support `tcp`, `udp` and `fwmark` service).

service_address The LVS service address.

server_address The real server address.

CLI Example:

```
salt '*' lvs.delete_server tcp 1.1.1.1:80 192.168.0.11:8080
```

`salt.modules.lvs.`**`delete_service`**`(`*protocol=None*, *service_address=None*`)`

Delete the virtual service.

protocol The service protocol(only support tcp, udp and fwmark service).

service_address The LVS service address.

CLI Example:

```
salt '*' lvs.delete_service tcp 1.1.1.1:80
```

`salt.modules.lvs.`**`edit_server`**`(`*protocol=None*, *service_address=None*, *server_address=None*, *packet_forward_method=None*, *weight=None*, ***kwargs*`)`

Edit a real server to a virtual service.

protocol The service protocol(only support `tcp`, `udp` and `fwmark` service).

service_address The LVS service address.

server_address The real server address.

packet_forward_method The LVS packet forwarding method(`dr` for direct routing, `tunnel` for tunneling, `nat` for network access translation).

weight The capacity of a server relative to the others in the pool.

CLI Example:

```
salt '*' lvs.edit_server tcp 1.1.1.1:80 192.168.0.11:8080 nat 1
```

`salt.modules.lvs.`**`edit_service`**`(`*protocol=None*, *service_address=None*, *scheduler=None*`)`

Edit the virtual service.

protocol The service protocol(only support tcp, udp and fwmark service).

service_address The LVS service address.

scheduler Algorithm for allocating TCP connections and UDP datagrams to real servers.

CLI Example:

```
salt '*' lvs.edit_service tcp 1.1.1.1:80 rr
```

`salt.modules.lvs.`**`get_rules`**`()`

Get the virtual server rules

CLI Example:

```
salt '*' lvs.get_rules
```

`salt.modules.lvs.`**`list_`**`(`*protocol=None*, *service_address=None*`)`

List the virtual server table if service_address is not specified. If a service_address is selected, list this service only.

CLI Example:

```
salt '*' lvs.list
```

`salt.modules.lvs.`**`zero`**(*protocol=None*, *service_address=None*)

Zero the packet, byte and rate counters in a service or all services.

CLI Example:

```
salt '*' lvs.zero
```

21.9.215 salt.modules.lxc

Control Linux Containers via Salt

depends lxc package for distribution

lxc >= 1.0 (even beta alpha) is required

`salt.modules.lxc.`**`add_veth`**(*name*, *interface_name*, *bridge=None*, *path=None*)

Add a veth to a container. Note : this function doesn't update the container config, just add the interface at runtime
name Name of the container
interface_name Name of the interface in the container
bridge Name of the bridge to attach the interface to (facultative)
CLI Examples:

```
salt '*' lxc.add_veth container_name eth1 br1
salt '*' lxc.add_veth container_name eth1
```

`salt.modules.lxc.`**`apply_network_profile`**(*name*, *network_profile*, *nic_opts=None*, *path=None*)

New in version 2015.5.0.

Apply a network profile to a container
network_profile profile name or default values (dict)
nic_opts values to override in defaults (dict) indexed by nic card names
path path to the container parent

New in version 2015.8.0.

CLI Examples:

```
salt 'minion' lxc.apply_network_profile web1 centos
salt 'minion' lxc.apply_network_profile web1 centos \
      nic_opts="{'eth0': {'mac': 'xx:xx:xx:xx:xx:xx'}}"
salt 'minion' lxc.apply_network_profile web1 \
      "{'eth0': {'mac': 'xx:xx:xx:xx:xx:yy'}}"
      nic_opts="{'eth0': {'mac': 'xx:xx:xx:xx:xx:xx'}}"
```

The special case to disable use of ethernet nics:

```
salt 'minion' lxc.apply_network_profile web1 centos \
      "{eth0: {disable: true}}"
```

`salt.modules.lxc.`**`attachable`**(*name*, *path=None*)

Return True if the named container can be attached to via the lxc-attach command
path path to the container parent default: /var/lib/lxc (system default)

New in version 2015.8.0.

CLI Example:

```
salt 'minion' lxc.attachable ubuntu
```

`salt.modules.lxc.`**`bootstrap`**(*name*, *config=None*, *approve_key=True*, *install=True*, *pub_key=None*, *priv_key=None*, *bootstrap_url=None*, *force_install=False*, *unconditional_install=False*, *path=None*, *bootstrap_delay=None*, *bootstrap_args=None*, *bootstrap_shell=None*)

Install and configure salt in a container.

config Minion configuration options. By default, the `master` option is set to the target host's master.

approve_key Request a pre-approval of the generated minion key. Requires that the salt-master be configured to either auto-accept all keys or expect a signing request from the target host. Default: `True`

path path to the container parent default: /var/lib/lxc (system default)

New in version 2015.8.0.

pub_key Explicit public key to pressed the minion with (optional). This can be either a filepath or a string representing the key

priv_key Explicit private key to pressed the minion with (optional). This can be either a filepath or a string representing the key

bootstrap_delay Delay in seconds between end of container creation and bootstrapping. Useful when waiting for container to obtain a DHCP lease.

New in version 2015.5.0.

bootstrap_url url, content or filepath to the salt bootstrap script

bootstrap_args salt bootstrap script arguments

bootstrap_shell shell to execute the script into

install Whether to attempt a full installation of salt-minion if needed.

force_install Force installation even if salt-minion is detected, this is the way to run vendor bootstrap scripts even if a salt minion is already present in the container

unconditional_install Run the script even if the container seems seeded

CLI Examples:

```
salt 'minion' lxc.bootstrap container_name [config=config_data] \
        [approve_key=(True|False)] [install=(True|False)]
```

`salt.modules.lxc.`**`clone`**(*name*, *orig*, *profile=None*, *network_profile=None*, *nic_opts=None*, ***kwargs*)

Create a new container as a clone of another container

name Name of the container

orig Name of the original container to be cloned

profile Profile to use in container cloning (see *`lxc.get_container_profile`*). Values in a profile will be overridden by the **Container Cloning Arguments** listed below.

path path to the container parent directory default: /var/lib/lxc (system)

New in version 2015.8.0.

Container Cloning Arguments

snapshot Use Copy On Write snapshots (LVM)

size [1G] Size of the volume to create. Only applicable if `backing=lvm`.

backing The type of storage to use. Set to `lvm` to use an LVM group. Defaults to filesystem within /var/lib/lxc.

network_profile Network profile to use for container

New in version 2015.8.0.

nic_opts give extra opts overriding network profile values

New in version 2015.8.0.

CLI Examples:

```
salt '*' lxc.clone myclone orig=orig_container
salt '*' lxc.clone myclone orig=orig_container snapshot=True
```

`salt.modules.lxc.`**`cloud_init`**(*name*, *vm_=None*, ***kwargs*)

Thin wrapper to lxc.init to be used from the saltcloud lxc driver

name Name of the container may be None and then guessed from saltcloud mapping

vm_ saltcloud mapping defaults for the vm

CLI Example:

```
salt '*' lxc.cloud_init foo
```

`salt.modules.lxc.`**`cloud_init_interface`**(*name*, *vm_=None*, ***kwargs*)

Interface between salt.cloud.lxc driver and lxc.init `vm_` is a mapping of vm opts in the salt.cloud format as documented for the lxc driver.

This can be used either:

- from the salt cloud driver
- because you find the argument to give easier here than using directly lxc.init

> **Warning:** BE REALLY CAREFUL CHANGING DEFAULTS !!! IT'S A RETRO COMPATIBLE INTERFACE WITH THE SALT CLOUD DRIVER (ask kiorky).

name name of the lxc container to create

pub_key public key to preseed the minion with. Can be the keycontent or a filepath

priv_key private key to preseed the minion with. Can be the keycontent or a filepath

path path to the container parent directory (default: /var/lib/lxc)

New in version 2015.8.0.

profile *profile* selection

network_profile *network profile* selection

nic_opts per interface settings compatibles with network profile (ipv4/ipv6/link/gateway/mac/netmask)

eg:

```
- {'eth0': {'mac': '00:16:3e:01:29:40',
            'gateway': None, (default)
            'link': 'br0', (default)
            'gateway': None, (default)
            'netmask': '', (default)
            'ip': '22.1.4.25'}}
```

unconditional_install given to lxc.bootstrap (see relative doc)

force_install given to lxc.bootstrap (see relative doc)

config any extra argument for the salt minion config

dnsservers list of DNS servers to set inside the container

dns_via_dhcp do not set the dns servers, let them be set by the dhcp.

autostart autostart the container at boot time

password administrative password for the container

bootstrap_delay delay before launching bootstrap script at Container init

> **Warning:** Legacy but still supported options:
>
> **from_container** which container we use as a template when running lxc.clone
>
> **image** which template do we use when we are using lxc.create. This is the default mode unless you specify something in from_container
>
> **backing** which backing store to use. Values can be: overlayfs, dir(default), lvm, zfs, brtfs
>
> **fstype** When using a blockdevice level backing store, which filesystem to use on

size When using a blockdevice level backing store, which size for the filesystem to use on

snapshot Use snapshot when cloning the container source

vgname if using LVM: vgname

lvname if using LVM: lvname

thinpool: if using LVM: thinpool

ip ip for the primary nic

mac mac address for the primary nic

netmask netmask for the primary nic (24) = `vm_.get('netmask', '24')`

bridge bridge for the primary nic (lxcbr0)

gateway network gateway for the container

additional_ips additional ips which will be wired on the main bridge (br0) which is connected to internet. Be aware that you may use manual virtual mac addresses providen by you provider (online, ovh, etc). This is a list of mappings {ip: '', mac: '', netmask:''} Set gateway to None and an interface with a gateway to escape from another interface that eth0. eg:

```
- {'mac': '00:16:3e:01:29:40',
   'gateway': None, (default)
   'link': 'br0', (default)
   'netmask': '', (default)
   'ip': '22.1.4.25'}
```

users administrative users for the container default: [root] and [root, ubuntu] on ubuntu

default_nic name of the first interface, you should really not override this

CLI Example:

```
salt '*' lxc.cloud_init_interface foo
```

`salt.modules.lxc.`**`copy_to`**(*name*, *source*, *dest*, *overwrite=False*, *makedirs=False*, *path=None*)

Changed in version 2015.8.0: Function renamed from `lxc.cp` to `lxc.copy_to` for consistency with other container types. `lxc.cp` will continue to work, however. For versions 2015.2.x and earlier, use `lxc.cp`.

Copy a file or directory from the host into a container
name Container name
source File to be copied to the container
path path to the container parent default: /var/lib/lxc (system default)

New in version 2015.8.0.
dest Destination on the container. Must be an absolute path.

Changed in version 2015.5.0: If the destination is a directory, the file will be copied into that directory.
overwrite [False] Unless this option is set to `True`, then if a file exists at the location specified by the `dest` argument, an error will be raised.

New in version 2015.8.0.
makedirs : False
Create the parent directory on the container if it does not already exist.

New in version 2015.5.0.
CLI Example:

```
salt 'minion' lxc.copy_to /tmp/foo /root/foo
salt 'minion' lxc.cp /tmp/foo /root/foo
```

`salt.modules.lxc.`**`create`**(*name*, *config=None*, *profile=None*, *network_profile=None*, *nic_opts=None*, ***kwargs*)

Create a new container.

name Name of the container

config The config file to use for the container. Defaults to system-wide config (usually in /etc/lxc/lxc.conf).

profile Profile to use in container creation (see *`lxc.get_container_profile`*). Values in a profile will be overridden by the **Container Creation Arguments** listed below.

network_profile Network profile to use for container

New in version 2015.5.0.

Container Creation Arguments

template The template to use. For example, `ubuntu` or `fedora`. For a full list of available templates, check out the *`lxc.templates`* function.

Conflicts with the `image` argument.

Note: The `download` template requires the following three parameters to be defined in `options`:

- **dist** - The name of the distribution
- **release** - Release name/version
- **arch** - Architecture of the container

The available images can be listed using the *`lxc.images`* function.

options Template-specific options to pass to the lxc-create command. These correspond to the long options (ones beginning with two dashes) that the template script accepts. For example:

```
options='{"dist": "centos", "release": "6", "arch": "amd64"}'
```

For available template options, refer to the lxc template scripts which are ususally located under `/usr/share/lxc/templates`, or run `lxc-create -t <template> -h`.

image A tar archive to use as the rootfs for the container. Conflicts with the `template` argument.

backing The type of storage to use. Set to `lvm` to use an LVM group. Defaults to filesystem within /var/lib/lxc.

fstype Filesystem type to use on LVM logical volume

size [1G] Size of the volume to create. Only applicable if `backing=lvm`.

vgname [lxc] Name of the LVM volume group in which to create the volume for this container. Only applicable if `backing=lvm`.

lvname Name of the LVM logical volume in which to create the volume for this container. Only applicable if `backing=lvm`.

thinpool Name of a pool volume that will be used for thin-provisioning this container. Only applicable if `backing=lvm`.

nic_opts give extra opts overriding network profile values

path parent path for the container creation (default: /var/lib/lxc)

zfsroot Name of the ZFS root in which to create the volume for this container. Only applicable if `backing=zfs`. (default: tank/lxc)

New in version 2015.8.0.

`salt.modules.lxc.`**`destroy`**(*name*, *stop=False*, *path=None*)

Destroy the named container.

Warning: Destroys all data associated with the container.

path path to the container parent directory (default: /var/lib/lxc)

New in version 2015.8.0.

stop [False] If `True`, the container will be destroyed even if it is running/frozen.

Changed in version 2015.5.0: Default value changed to `False`. This more closely matches the behavior of `lxc-destroy(1)`, and also makes it less likely that an accidental command will destroy a running container that was being used for important things.

CLI Examples:

```
salt '*' lxc.destroy foo
salt '*' lxc.destroy foo stop=True
```

`salt.modules.lxc.`**`edit_conf`**`(`*conf_file, out_format='simple', read_only=False, lxc_config=None, **kwargs*`)`

Edit an LXC configuration file. If a setting is already present inside the file, its value will be replaced. If it does not exist, it will be appended to the end of the file. Comments and blank lines will be kept in-tact if they already exist in the file.

out_format: Set to simple if you need backward compatibility (multiple items for a simple key is not supported)

read_only: return only the edited configuration without applying it to the underlying lxc configuration file

lxc_config: List of dict containning lxc configuration items For network configuration, you also need to add the device it belongs to, otherwise it will default to eth0. Also, any change to a network parameter will result in the whole network reconfiguration to avoid mismatchs, be aware of that !

After the file is edited, its contents will be returned. By default, it will be returned in `simple` format, meaning an unordered dict (which may not represent the actual file order). Passing in an `out_format` of `commented` will return a data structure which accurately represents the order and content of the file.

CLI Example:

```
salt 'minion' lxc.edit_conf /etc/lxc/mycontainer.conf \
    out_format=commented lxc.network.type=veth
salt 'minion' lxc.edit_conf /etc/lxc/mycontainer.conf \
    out_format=commented \
    lxc_config="[{'lxc.network.name': 'eth0', \
                  'lxc.network.ipv4': '1.2.3.4'},
                 {'lxc.network.name': 'eth2', \
                  'lxc.network.ipv4': '1.2.3.5',\
                  'lxc.network.gateway': '1.2.3.1'}]"
```

`salt.modules.lxc.`**`exists`**`(`*name, path=None*`)`

Returns whether the named container exists.

path path to the container parent directory (default: /var/lib/lxc)

New in version 2015.8.0.

CLI Example:

```
salt '*' lxc.exists name
```

`salt.modules.lxc.`**`freeze`**`(`*name, **kwargs*`)`

Freeze the named container

path path to the container parent directory default: /var/lib/lxc (system)

New in version 2015.8.0.

start [False] If `True` and the container is stopped, the container will be started before attempting to freeze.

New in version 2015.5.0.

use_vt run the command through VT

New in version 2015.8.0.

CLI Example:

```
salt '*' lxc.freeze name
```

`salt.modules.lxc.`**`get_container_profile`**`(`*name=None*, ***kwargs*`)`

New in version 2015.5.0.

Gather a pre-configured set of container configuration parameters. If no arguments are passed, an empty profile is returned.

Profiles can be defined in the minion or master config files, or in pillar or grains, and are loaded using *`config.get`*. The key under which LXC profiles must be configured is `lxc.container_profile.profile_name`. An example container profile would be as follows:

```
lxc.container_profile:
  ubuntu:
    template: ubuntu
    backing: lvm
    vgname: lxc
    size: 1G
```

Parameters set in a profile can be overridden by passing additional container creation arguments (such as the ones passed to *`lxc.create`*) to this function.

A profile can be defined either as the name of the profile, or a dictionary of variable names and values. See the *LXC Tutorial* for more information on how to use LXC profiles.

CLI Example:

```
salt-call lxc.get_container_profile centos
salt-call lxc.get_container_profile ubuntu template=ubuntu backing=overlayfs
```

`salt.modules.lxc.`**`get_network_profile`**`(`*name=None*, ***kwargs*`)`

New in version 2015.5.0.

Gather a pre-configured set of network configuration parameters. If no arguments are passed, the following default profile is returned:

```
{'eth0': {'link': 'br0', 'type': 'veth', 'flags': 'up'}}
```

Profiles can be defined in the minion or master config files, or in pillar or grains, and are loaded using *`config.get`*. The key under which LXC profiles must be configured is `lxc.network_profile`. An example network profile would be as follows:

```
lxc.network_profile.centos:
  eth0:
    link: br0
    type: veth
    flags: up
```

To disable networking entirely:

```
lxc.network_profile.centos:
  eth0:
    disable: true
```

Parameters set in a profile can be overridden by passing additional arguments to this function.

A profile can be passed either as the name of the profile, or a dictionary of variable names and values. See the *LXC Tutorial* for more information on how to use network profiles.

> **Warning:** The `ipv4`, `ipv6`, `gateway`, and `link` (bridge) settings in network profiles will only work if the container doesn't redefine the network configuration (for example in `/etc/sysconfig/network-scripts/ifcfg-<interface_name>` on RHEL/CentOS, or `/etc/network/interfaces` on Debian/Ubuntu/etc.)

CLI Example:

```
salt-call lxc.get_network_profile default
```

`salt.modules.lxc.`**`get_parameter`**(*name*, *parameter*, *path=None*)

Returns the value of a cgroup parameter for a container

path path to the container parent directory default: /var/lib/lxc (system)

New in version 2015.8.0.

CLI Example:

```
salt '*' lxc.get_parameter container_name memory.limit_in_bytes
```

`salt.modules.lxc.`**`get_pid`**(*name*, *path=None*)

Returns a container pid. Throw an exception if the container isn't running.

CLI Example:

```
salt '*' lxc.get_pid name
```

`salt.modules.lxc.`**`get_root_path`**(*path*)

Get the configured lxc root for containers

New in version 2015.8.0.

CLI Example:

```
salt '*' lxc.get_root_path
```

`salt.modules.lxc.`**`images`**(*dist=None*)

New in version 2015.5.0.

List the available images for LXC's `download` template.

dist [None] Filter results to a single Linux distribution

CLI Examples:

```
salt myminion lxc.images
salt myminion lxc.images dist=centos
```

`salt.modules.lxc.`**`info`**(*name*, *path=None*)

Returns information about a container

path path to the container parent directory default: /var/lib/lxc (system)

New in version 2015.8.0.

CLI Example:

```
salt '*' lxc.info name
```

`salt.modules.lxc.`**`init`**(*name*, *config=None*, *cpuset=None*, *cpushare=None*, *memory=None*, *profile=None*, *network_profile=None*, *nic_opts=None*, *cpu=None*, *autostart=True*, *password=None*, *password_encrypted=None*, *users=None*, *dnsservers=None*, *searchdomains=None*, *bridge=None*, *gateway=None*, *pub_key=None*, *priv_key=None*, *force_install=False*, *unconditional_install=False*, *bootstrap_delay=None*, *bootstrap_args=None*, *bootstrap_shell=None*, *bootstrap_url=None*, ***kwargs*)

Initialize a new container.

This is a partial idempotent function as if it is already provisioned, we will reset a bit the lxc configuration file but much of the hard work will be escaped as markers will prevent re-execution of harmful tasks.

name Name of the container

image A tar archive to use as the rootfs for the container. Conflicts with the `template` argument.

cpus Select a random number of cpu cores and assign it to the cpuset, if the cpuset option is set then this option will be ignored

cpuset Explicitly define the cpus this container will be bound to

cpushare cgroups cpu shares

autostart autostart container on reboot

memory cgroups memory limit, in MB

Changed in version 2015.5.0: If no value is passed, no limit is set. In earlier Salt versions, not passing this value causes a 1024MB memory limit to be set, and it was necessary to pass `memory=0` to set no limit.

gateway the ipv4 gateway to use the default does nothing more than lxcutils does

bridge the bridge to use the default does nothing more than lxcutils does

network_profile Network profile to use for the container

New in version 2015.5.0.

nic_opts Extra options for network interfaces, will override

`{"eth0": {"hwaddr": "aa:bb:cc:dd:ee:ff", "ipv4": "10.1.1.1", "ipv6": "2001:db8::ff00:42:8329"}}`

or

`{"eth0": {"hwaddr": "aa:bb:cc:dd:ee:ff", "ipv4": "10.1.1.1/24", "ipv6": "2001:db8::ff00:42:8329"}}`

users Users for which the password defined in the `password` param should be set. Can be passed as a comma separated list or a python list. Defaults to just the `root` user.

password Set the initial password for the users defined in the `users` parameter

password_encrypted [False] Set to `True` to denote a password hash instead of a plaintext password

New in version 2015.5.0.

profile A LXC profile (defined in config or pillar). This can be either a real profile mapping or a string to retrieve it in configuration

start Start the newly-created container

dnsservers list of dns servers to set in the container, default [] (no setting)

seed Seed the container with the minion config. Default: `True`

install If salt-minion is not already installed, install it. Default: `True`

config Optional config parameters. By default, the id is set to the name of the container.

master salt master (default to minion's master)

master_port salt master port (default to minion's master port)

pub_key Explicit public key to preseed the minion with (optional). This can be either a filepath or a string representing the key

priv_key Explicit private key to preseed the minion with (optional). This can be either a filepath or a string representing the key

approve_key If explicit preseeding is not used; Attempt to request key approval from the master. Default: `True`

path path to the container parent directory default: /var/lib/lxc (system)

New in version 2015.8.0.

clone_from Original from which to use a clone operation to create the container. Default: `None`

bootstrap_delay Delay in seconds between end of container creation and bootstrapping. Useful when waiting for container to obtain a DHCP lease.

New in version 2015.5.0.

bootstrap_url See lxc.bootstrap

bootstrap_shell See lxc.bootstrap

bootstrap_args See lxc.bootstrap

force_install Force installation even if salt-minion is detected, this is the way to run vendor bootstrap scripts even if a salt minion is already present in the container

unconditional_install Run the script even if the container seems seeded

CLI Example:

```
salt 'minion' lxc.init name [cpuset=cgroups_cpuset] \
        [cpushare=cgroups_cpushare] [memory=cgroups_memory] \
        [nic=nic_profile] [profile=lxc_profile] \
        [nic_opts=nic_opts] [start=(True|False)] \
        [seed=(True|False)] [install=(True|False)] \
        [config=minion_config] [approve_key=(True|False) \
        [clone_from=original] [autostart=True] \
        [priv_key=/path_or_content] [pub_key=/path_or_content] \
        [bridge=lxcbr0] [gateway=10.0.3.1] \
        [dnsservers[dns1,dns2]] \
        [users=[foo]] [password='secret'] \
        [password_encrypted=(True|False)]
```

`salt.modules.lxc.`**`list_`**(*extra=False*, *limit=None*, *path=None*)

List containers classified by state

extra Also get per-container specific info. This will change the return data. Instead of returning a list of containers, a dictionary of containers and each container's output from *`lxc.info`*.

path path to the container parent directory default: /var/lib/lxc (system)

New in version 2015.8.0.

limit Return output matching a specific state (**frozen**, **running**, or **stopped**).

New in version 2015.5.0.

CLI Examples:

```
salt '*' lxc.list
salt '*' lxc.list extra=True
salt '*' lxc.list limit=running
```

`salt.modules.lxc.`**`ls_`**(*active=None*, *cache=True*, *path=None*)

Return a list of the containers available on the minion

path path to the container parent directory default: /var/lib/lxc (system)

New in version 2015.8.0.

active If `True`, return only active (i.e. running) containers

New in version 2015.5.0.

CLI Example:

```
salt '*' lxc.ls
salt '*' lxc.ls active=True
```

`salt.modules.lxc.`**`read_conf`**(*conf_file*, *out_format='simple'*)

Read in an LXC configuration file. By default returns a simple, unsorted dict, but can also return a more detailed structure including blank lines and comments.

out_format: set to 'simple' if you need the old and unsupported behavior. This won't support the multiple lxc values (eg: multiple network nics)

CLI Examples:

```
salt 'minion' lxc.read_conf /etc/lxc/mycontainer.conf
salt 'minion' lxc.read_conf /etc/lxc/mycontainer.conf out_format=commented
```

`salt.modules.lxc.`**`reboot`**(*name*, *path=None*)

Reboot a container.

path path to the container parent default: /var/lib/lxc (system default)

New in version 2015.8.0.

CLI Examples:

```
salt 'minion' lxc.reboot myvm
```

`salt.modules.lxc.`**`reconfigure`**(*name*, *cpu=None*, *cpuset=None*, *cpushare=None*, *memory=None*, *profile=None*, *network_profile=None*, *nic_opts=None*, *bridge=None*, *gateway=None*, *autostart=None*, *utsname=None*, *rootfs=None*, *path=None*, ***kwargs*)

Reconfigure a container.

This only applies to a few property

name Name of the container.

utsname utsname of the container.

New in version 2016.3.0.

rootfs rootfs of the container.

New in version 2016.3.0.

cpu Select a random number of cpu cores and assign it to the cpuset, if the cpuset option is set then this option will be ignored

cpuset Explicitly define the cpus this container will be bound to

cpushare cgroups cpu shares.

autostart autostart container on reboot

memory cgroups memory limit, in MB. (0 for nolimit, None for old default 1024MB)

gateway the ipv4 gateway to use the default does nothing more than lxcutils does

bridge the bridge to use the default does nothing more than lxcutils does

nic Network interfaces profile (defined in config or pillar).

nic_opts Extra options for network interfaces, will override

`{"eth0": {"mac": "aa:bb:cc:dd:ee:ff", "ipv4": "10.1.1.1", "ipv6": "2001:db8::ff00:42:8329"}}`

or

`{"eth0": {"mac": "aa:bb:cc:dd:ee:ff", "ipv4": "10.1.1.1/24", "ipv6": "2001:db8::ff00:42:8329"}}`

path path to the container parent

New in version 2015.8.0.

CLI Example:

```
salt-call -lall mc_lxc_fork.reconfigure foobar nic_opts="{'eth1': {'mac':
↪'00:16:3e:dd:ee:44'}}" memory=4
```

`salt.modules.lxc.`**`restart`**(*name*, *path=None*, *lxc_config=None*, *force=False*)

New in version 2015.5.0.

Restart the named container. If the container was not running, the container will merely be started.

name The name of the container

path path to the container parent directory default: /var/lib/lxc (system)

New in version 2015.8.0.

lxc_config path to a lxc config file config file will be guessed from container name otherwise

New in version 2015.8.0.

force [False] If `True`, the container will be force-stopped instead of gracefully shut down

CLI Example:

```
salt myminion lxc.restart name
```

`salt.modules.lxc.`**`retcode`**(*name*, *cmd*, *no_start=False*, *preserve_state=True*, *stdin=None*, *python_shell=True*, *output_loglevel='debug'*, *use_vt=False*, *path=None*, *ignore_retcode=False*, *chroot_fallback=False*, *keep_env='http_proxy*, *https_proxy*, *no_proxy'*)

New in version 2015.5.0.

Run `cmd.retcode` within a container

Warning: Many shell builtins do not work, failing with stderr similar to the following:

```
lxc_container: No such file or directory - failed to exec 'command'
```

The same error will be displayed in stderr if the command being run does not exist. If the retcode is nonzero and not what was expected, try using `lxc.run_stderr` or `lxc.run_all`.

name Name of the container in which to run the command

cmd Command to run

no_start [False] If the container is not running, don't start it

preserve_state [True] After running the command, return the container to its previous state

path path to the container parent default: /var/lib/lxc (system default)

New in version 2015.8.0.

stdin [None] Standard input to be used for the command

output_loglevel [debug] Level at which to log the output from the command. Set to `quiet` to suppress logging.

use_vt [False] Use SaltStack's utils.vt to stream output to console `output=all`.

keep_env [http_proxy,https_proxy,no_proxy] A list of env vars to preserve. May be passed as commma-delimited list.

chroot_fallback if the container is not running, try to run the command using chroot default: false

CLI Example:

```
salt myminion lxc.retcode mycontainer 'ip addr show'
```

`salt.modules.lxc.`**`run`**(*name*, *cmd*, *no_start=False*, *preserve_state=True*, *stdin=None*, *python_shell=True*, *output_loglevel='debug'*, *use_vt=False*, *path=None*, *ignore_retcode=False*, *chroot_fallback=False*, *keep_env='http_proxy*, *https_proxy*, *no_proxy'*)

New in version 2015.8.0.

Run `cmd.run` within a container

Warning: Many shell builtins do not work, failing with stderr similar to the following:

```
lxc_container: No such file or directory - failed to exec 'command'
```

The same error will be displayed in stderr if the command being run does not exist. If no output is returned using this function, try using *lxc.run_stderr* or *lxc.run_all*.

name Name of the container in which to run the command
cmd Command to run
path path to the container parent default: /var/lib/lxc (system default)

New in version 2015.8.0.

no_start [False] If the container is not running, don't start it
preserve_state [True] After running the command, return the container to its previous state
stdin [None] Standard input to be used for the command
output_loglevel [debug] Level at which to log the output from the command. Set to `quiet` to suppress logging.
use_vt [False] Use SaltStack's utils.vt to stream output to console. Assumes `output=all`.
chroot_fallback if the container is not running, try to run the command using chroot default: false
keep_env [http_proxy,https_proxy,no_proxy] A list of env vars to preserve. May be passed as commma-delimited list.

CLI Example:

```
salt myminion lxc.run mycontainer 'ifconfig -a'
```

`salt.modules.lxc.`**`run_all`**`(`*name, cmd, no_start=False, preserve_state=True, stdin=None, python_shell=True, output_loglevel='debug', use_vt=False, path=None, ignore_retcode=False, chroot_fallback=False, keep_env='http_proxy, https_proxy, no_proxy'*`)`

New in version 2015.5.0.

Run *cmd.run_all* within a container

Note: While the command is run within the container, it is initiated from the host. Therefore, the PID in the return dict is from the host, not from the container.

Warning: Many shell builtins do not work, failing with stderr similar to the following:

```
lxc_container: No such file or directory - failed to exec 'command'
```

The same error will be displayed in stderr if the command being run does not exist.

name Name of the container in which to run the command
path path to the container parent default: /var/lib/lxc (system default)

New in version 2015.8.0.

cmd Command to run
no_start [False] If the container is not running, don't start it
preserve_state [True] After running the command, return the container to its previous state
stdin [None] Standard input to be used for the command
output_loglevel [debug] Level at which to log the output from the command. Set to `quiet` to suppress logging.
use_vt [False] Use SaltStack's utils.vt to stream output to console `output=all`.

keep_env [http_proxy,https_proxy,no_proxy] A list of env vars to preserve. May be passed as commma-delimited list.

chroot_fallback if the container is not running, try to run the command using chroot default: false

CLI Example:

```
salt myminion lxc.run_all mycontainer 'ip addr show'
```

`salt.modules.lxc.`**`run_stderr`**(*name, cmd, no_start=False, preserve_state=True, stdin=None, python_shell=True, output_loglevel='debug', use_vt=False, path=None, ignore_retcode=False, chroot_fallback=False, keep_env='http_proxy, https_proxy, no_proxy'*)

New in version 2015.5.0.

Run *`cmd.run_stderr`* within a container

> **Warning:** Many shell builtins do not work, failing with stderr similar to the following:
>
> ```
> lxc_container: No such file or directory - failed to exec 'command'
> ```
>
> The same error will be displayed if the command being run does not exist.

name Name of the container in which to run the command

cmd Command to run

path path to the container parent default: /var/lib/lxc (system default)

New in version 2015.8.0.

no_start [False] If the container is not running, don't start it

preserve_state [True] After running the command, return the container to its previous state

stdin [None] Standard input to be used for the command

output_loglevel [debug] Level at which to log the output from the command. Set to `quiet` to suppress logging.

use_vt [False] Use SaltStack's utils.vt to stream output to console `output=all`.

keep_env [http_proxy,https_proxy,no_proxy] A list of env vars to preserve. May be passed as commma-delimited list.

chroot_fallback if the container is not running, try to run the command using chroot default: false

CLI Example:

```
salt myminion lxc.run_stderr mycontainer 'ip addr show'
```

`salt.modules.lxc.`**`run_stdout`**(*name, cmd, no_start=False, preserve_state=True, stdin=None, python_shell=True, output_loglevel='debug', use_vt=False, path=None, ignore_retcode=False, chroot_fallback=False, keep_env='http_proxy, https_proxy, no_proxy'*)

New in version 2015.5.0.

Run *`cmd.run_stdout`* within a container

> **Warning:** Many shell builtins do not work, failing with stderr similar to the following:
>
> ```
> lxc_container: No such file or directory - failed to exec 'command'
> ```
>
> The same error will be displayed in stderr if the command being run does not exist. If no output is returned using this function, try using *`lxc.run_stderr`* or *`lxc.run_all`*.

name Name of the container in which to run the command

cmd Command to run

path path to the container parent default: /var/lib/lxc (system default)

New in version 2015.8.0.

no_start [False] If the container is not running, don't start it

preserve_state [True] After running the command, return the container to its previous state

stdin [None] Standard input to be used for the command

output_loglevel [debug] Level at which to log the output from the command. Set to `quiet` to suppress logging.

use_vt [False] Use SaltStack's utils.vt to stream output to console `output=all`.

keep_env [http_proxy,https_proxy,no_proxy] A list of env vars to preserve. May be passed as commma-delimited list.

chroot_fallback if the container is not running, try to run the command using chroot default: false

CLI Example:

```
salt myminion lxc.run_stdout mycontainer 'ifconfig -a'
```

`salt.modules.lxc.`**`running_systemd`**`(`*name*, *cache=True*, *path=None*`)`

Determine if systemD is running

path path to the container parent

New in version 2015.8.0.

CLI Example:

```
salt '*' lxc.running_systemd ubuntu
```

`salt.modules.lxc.`**`search_lxc_bridge`**`()`

Search the first bridge which is potentially available as LXC bridge

CLI Example:

```
salt '*' lxc.search_lxc_bridge
```

`salt.modules.lxc.`**`search_lxc_bridges`**`()`

Search which bridges are potentially available as LXC bridges

CLI Example:

```
salt '*' lxc.search_lxc_bridges
```

`salt.modules.lxc.`**`set_dns`**`(`*name*, *dnsservers=None*, *searchdomains=None*, *path=None*`)`

Changed in version 2015.5.0: The `dnsservers` and `searchdomains` parameters can now be passed as a comma-separated list.

Update /etc/resolv.confo

path

path to the container parent default: /var/lib/lxc (system default)

New in version 2015.8.0.

CLI Example:

```
salt myminion lxc.set_dns ubuntu "['8.8.8.8', '4.4.4.4']"
```

`salt.modules.lxc.`**`set_parameter`**`(`*name*, *parameter*, *value*, *path=None*`)`

Set the value of a cgroup parameter for a container.

path path to the container parent directory default: /var/lib/lxc (system)

New in version 2015.8.0.

CLI Example:

```
salt '*' lxc.set_parameter name parameter value
```

`salt.modules.lxc.`**`set_password`**(*name, users, password, encrypted=True, path=None*)

Changed in version 2015.5.0: Function renamed from `set_pass` to `set_password`. Additionally, this function now supports (and defaults to using) a password hash instead of a plaintext password.

Set the password of one or more system users inside containers

users Comma-separated list (or python list) of users to change password

password Password to set for the specified user(s)

encrypted [True] If true, `password` must be a password hash. Set to `False` to set a plaintext password (not recommended).

New in version 2015.5.0.

path path to the container parent directory default: /var/lib/lxc (system)

New in version 2015.8.0.

CLI Example:

```
salt '*' lxc.set_pass container-name root '$6$uJ2uAyLU$KoI67t8As/
↪0fXtJOPcHKGXmUpcoYUcVR2K6x93walnShTCQvjRwq25yIkiCBOqgbfdKQSFnAo28/ek6716vEV1'
salt '*' lxc.set_pass container-name root foo encrypted=False
```

`salt.modules.lxc.`**`start`**(*name, **kwargs*)

Start the named container

path path to the container parent directory default: /var/lib/lxc (system)

New in version 2015.8.0.

lxc_config path to a lxc config file config file will be guessed from container name otherwise

New in version 2015.8.0.

use_vt run the command through VT

New in version 2015.8.0.

CLI Example:

```
salt myminion lxc.start name
```

`salt.modules.lxc.`**`state`**(*name, path=None*)

Returns the state of a container.

path path to the container parent directory (default: /var/lib/lxc)

New in version 2015.8.0.

CLI Example:

```
salt '*' lxc.state name
```

`salt.modules.lxc.`**`stop`**(*name, kill=False, path=None, use_vt=None*)

Stop the named container

path path to the container parent directory default: /var/lib/lxc (system)

New in version 2015.8.0.

kill: False Do not wait for the container to stop, kill all tasks in the container. Older LXC versions will stop containers like this irrespective of this argument.

Changed in version 2015.5.0: Default value changed to `False`

use_vt run the command through VT

New in version 2015.8.0.

CLI Example:

```
salt myminion lxc.stop name
```

`salt.modules.lxc.`**`systemd_running_state`**`(`*name*, *path=None*`)`

Get the operational state of a systemd based container

path path to the container parent default: /var/lib/lxc (system default)

New in version 2015.8.0.

CLI Example:

```
salt myminion lxc.systemd_running_state ubuntu
```

`salt.modules.lxc.`**`templates`**`()`

New in version 2015.5.0.

List the available LXC template scripts installed on the minion

CLI Examples:

```
salt myminion lxc.templates
```

`salt.modules.lxc.`**`test_bare_started_state`**`(`*name*, *path=None*`)`

Test if a non systemd container is fully started For now, it consists only to test if the container is attachable

path path to the container parent default: /var/lib/lxc (system default)

New in version 2015.8.0.

CLI Example:

```
salt myminion lxc.test_bare_started_state ubuntu
```

`salt.modules.lxc.`**`test_sd_started_state`**`(`*name*, *path=None*`)`

Test if a systemd container is fully started

path path to the container parent default: /var/lib/lxc (system default)

New in version 2015.8.0.

CLI Example:

```
salt myminion lxc.test_sd_started_state ubuntu
```

`salt.modules.lxc.`**`unfreeze`**`(`*name*, *path=None*, *use_vt=None*`)`

Unfreeze the named container.

path path to the container parent directory default: /var/lib/lxc (system)

New in version 2015.8.0.

use_vt run the command through VT

New in version 2015.8.0.

CLI Example:

```
salt '*' lxc.unfreeze name
```

`salt.modules.lxc.`**`update_lxc_conf`**`(`*name*, *lxc_conf*, *lxc_conf_unset*, *path=None*`)`

Edit LXC configuration options

path path to the container parent default: /var/lib/lxc (system default)

New in version 2015.8.0.

CLI Example:

```
salt myminion lxc.update_lxc_conf ubuntu \
        lxc_conf="[{'network.ipv4.ip':'10.0.3.5'}]" \
        lxc_conf_unset="['lxc.utsname']"
```

`salt.modules.lxc.`**`version`**`()`

Return the actual lxc client version

New in version 2015.8.0.

CLI Example:

```
salt '*' lxc.version
```

`salt.modules.lxc.`**`wait_started`**`(`*name, path=None, timeout=300*`)`

Check that the system has fully inited

This is actually very important for systemD based containers

see https://github.com/saltstack/salt/issues/23847

path path to the container parent default: /var/lib/lxc (system default)

New in version 2015.8.0.

CLI Example:

```
salt myminion lxc.wait_started ubuntu
```

`salt.modules.lxc.`**`write_conf`**`(`*conf_file, conf*`)`

Write out an LXC configuration file

This is normally only used internally. The format of the data structure must match that which is returned from `lxc.read_conf()`, with `out_format` set to `commented`.

An example might look like:

```
[
    {'lxc.utsname': '$CONTAINER_NAME'},
    '# This is a commented line\n',
    '\n',
    {'lxc.mount': '$CONTAINER_FSTAB'},
    {'lxc.rootfs': {'comment': 'This is another test',
                    'value': 'This is another test'}},
    '\n',
    {'lxc.network.type': 'veth'},
    {'lxc.network.flags': 'up'},
    {'lxc.network.link': 'br0'},
    {'lxc.network.mac': '$CONTAINER_MACADDR'},
    {'lxc.network.ipv4': '$CONTAINER_IPADDR'},
    {'lxc.network.name': '$CONTAINER_DEVICENAME'},
]
```

CLI Example:

```
salt 'minion' lxc.write_conf /etc/lxc/mycontainer.conf \
    out_format=commented
```

21.9.216 salt.modules.lxd

Module for managing the LXD daemon and its containers.

New in version 2019.2.0.

LXD(1) is a container "hypervisor". This execution module provides several functions to help manage it and its containers.

Note:

- pylxd(2) version >=2.2.5 is required to let this work, currently only available via pip.

  To install on Ubuntu:

  $ apt-get install libssl-dev python-pip $ pip install -U pylxd

- you need lxd installed on the minion for the init() and version() methods.
- for the config_get() and config_get() methods you need to have lxd-client installed.

---

maintainer René Jochum <rene@jochums.at>

maturity new

depends python-pylxd

platform Linux

`salt.modules.lxd.`**`authenticate`**(*remote_addr*, *password*, *cert*, *key*, *verify_cert=True*)

Authenticate with a remote LXDaemon.

remote_addr : An URL to a remote Server, you also have to give cert and key if you provide remote_addr and its a TCP Address!

Examples: https://myserver.lan:8443

password : The password of the remote.

cert : PEM Formatted SSL Certificate.

Examples: ~/.config/lxc/client.crt

key : PEM Formatted SSL Key.

Examples: ~/.config/lxc/client.key

verify_cert [True] Wherever to verify the cert, this is by default True but in the most cases you want to set it off as LXD normaly uses self-signed certificates.

CLI Example:

```
$ salt '*' lxd.authenticate https://srv01:8443 <yourpass> ~/.config/lxc/client.
↪crt ~/.config/lxc/client.key false
```

See the requests-docs for the SSL stuff.

`salt.modules.lxd.`**`config_get`**(*key*)

Get an LXD daemon config option

key : The key of the config value to retrieve

CLI Examples:

```
salt '*' lxd.config_get core.https_address
```

`salt.modules.lxd.`**`config_set`**(*key*, *value*)

Set an LXD daemon config option

CLI Examples:

To listen on IPv4 and IPv6 port 8443, you can omit the :8443 its the default:

```
salt '*' lxd.config_set core.https_address [::]:8443
```

To set the server trust password:

```
salt '*' lxd.config_set core.trust_password blah
```

salt.modules.lxd.**container_config_delete**(*name*, *config_key*, *remote_addr=None*, *cert=None*, *key=None*, *verify_cert=True*)

Delete a container config value

name : Name of the container

config_key : The config key to delete

remote_addr : An URL to a remote Server, you also have to give cert and key if you provide remote_addr and its a TCP Address!

Examples: https://myserver.lan:8443 /var/lib/mysocket.sock

cert : PEM Formatted SSL Certificate.

Examples: ~/.config/lxc/client.crt

key : PEM Formatted SSL Key.

Examples: ~/.config/lxc/client.key

verify_cert [True] Wherever to verify the cert, this is by default True but in the most cases you want to set it off as LXD normaly uses self-signed certificates.

salt.modules.lxd.**container_config_get**(*name*, *config_key*, *remote_addr=None*, *cert=None*, *key=None*, *verify_cert=True*)

Get a container config value

name : Name of the container

config_key : The config key to retrieve

remote_addr : An URL to a remote Server, you also have to give cert and key if you provide remote_addr and its a TCP Address!

Examples: https://myserver.lan:8443 /var/lib/mysocket.sock

cert : PEM Formatted SSL Certificate.

Examples: ~/.config/lxc/client.crt

key : PEM Formatted SSL Key.

Examples: ~/.config/lxc/client.key

verify_cert [True] Wherever to verify the cert, this is by default True but in the most cases you want to set it off as LXD normaly uses self-signed certificates.

salt.modules.lxd.**container_config_set**(*name*, *config_key*, *config_value*, *remote_addr=None*, *cert=None*, *key=None*, *verify_cert=True*)

Set a container config value

name : Name of the container

config_key : The config key to set

config_value : The config value to set

remote_addr : An URL to a remote Server, you also have to give cert and key if you provide remote_addr and its a TCP Address!

Examples: https://myserver.lan:8443 /var/lib/mysocket.sock

cert : PEM Formatted SSL Certificate.

Examples: ~/.config/lxc/client.crt

key : PEM Formatted SSL Key.

Examples: ~/.config/lxc/client.key

verify_cert [True] Wherever to verify the cert, this is by default True but in the most cases you want to set it off as LXD normaly uses self-signed certificates.

salt.modules.lxd.**container_create**(*name*, *source*, *profiles=None*, *config=None*, *devices=None*, *architecture='x86_64'*, *ephemeral=False*, *wait=True*, *remote_addr=None*, *cert=None*, *key=None*, *verify_cert=True*, *_raw=False*)

Create a container

name : The name of the container

source :
  Can be either a string containing an image alias: "xenial/amd64"
  or an dict with type "image" with alias:

    {"type": "image", "alias": "xenial/amd64"}
  or image with "fingerprint":

    {"type": "image", "fingerprint": "SHA-256"}
  or image with "properties":

    {"type": "image",

      "properties": { "os": "ubuntu", "release": "14.04", "architecture": "x86_64"}}
  or none: {"type": "none"}
  or copy:

    {"type": "copy", "source": "my-old-container"}

profiles [['default']] List of profiles to apply on this container

config : A config dict or None (None = unset).
  Can also be a list:

    [{'key': 'boot.autostart', 'value': 1}, {'key': 'security.privileged', 'value': '1'}]

devices : A device dict or None (None = unset).

architecture ['x86_64']
  Can be one of the following:

- unknown
- i686
- x86_64
- armv7l
- aarch64
- ppc
- ppc64
- ppc64le
- s390x

ephemeral [False] Destroy this container after stop?

remote_addr : An URL to a remote Server, you also have to give cert and key if you provide remote_addr and its a TCP Address!
  Examples: https://myserver.lan:8443 /var/lib/mysocket.sock

cert : PEM Formatted SSL Certificate.
  Examples: ~/.config/lxc/client.crt

key : PEM Formatted SSL Key.
  Examples: ~/.config/lxc/client.key

verify_cert [True] Wherever to verify the cert, this is by default True but in the most cases you want to set it off as LXD normaly uses self-signed certificates.

_raw [False] Return the raw pyxld object or a dict?

CLI Examples:

```
salt '*' lxd.container_create test xenial/amd64
```

See also the rest-api-docs.

`salt.modules.lxd.`**`container_delete`**`(`*name, remote_addr=None, cert=None, key=None, verify_cert=True*`)`

Delete a container

name : Name of the container to delete

remote_addr : An URL to a remote Server, you also have to give cert and key if you provide remote_addr and its a TCP Address!

Examples: https://myserver.lan:8443 /var/lib/mysocket.sock

cert : PEM Formatted SSL Certificate.

Examples: ~/.config/lxc/client.crt

key : PEM Formatted SSL Key.

Examples: ~/.config/lxc/client.key

verify_cert [True] Wherever to verify the cert, this is by default True but in the most cases you want to set it off as LXD normaly uses self-signed certificates.

`salt.modules.lxd.`**`container_device_add`**`(`*name, device_name, device_type='disk', remote_addr=None, cert=None, key=None, verify_cert=True, **kwargs*`)`

Add a container device

name : Name of the container

device_name : The device name to add

device_type : Type of the device

** **kwargs** : Additional device args

remote_addr : An URL to a remote Server, you also have to give cert and key if you provide remote_addr and its a TCP Address!

Examples: https://myserver.lan:8443 /var/lib/mysocket.sock

cert : PEM Formatted SSL Certificate.

Examples: ~/.config/lxc/client.crt

key : PEM Formatted SSL Key.

Examples: ~/.config/lxc/client.key

verify_cert [True] Wherever to verify the cert, this is by default True but in the most cases you want to set it off as LXD normaly uses self-signed certificates.

`salt.modules.lxd.`**`container_device_delete`**`(`*name, device_name, remote_addr=None, cert=None, key=None, verify_cert=True*`)`

Delete a container device

name : Name of the container

device_name : The device name to delete

remote_addr : An URL to a remote Server, you also have to give cert and key if you provide remote_addr and its a TCP Address!

Examples: https://myserver.lan:8443 /var/lib/mysocket.sock

cert : PEM Formatted SSL Certificate.

Examples: ~/.config/lxc/client.crt

key : PEM Formatted SSL Key.

Examples: ~/.config/lxc/client.key

verify_cert [True] Wherever to verify the cert, this is by default True but in the most cases you want to set it off as LXD normaly uses self-signed certificates.

`salt.modules.lxd.`**`container_device_get`**`(`*name, device_name, remote_addr=None, cert=None, key=None, verify_cert=True*`)`

Get a container device

name : Name of the container

device_name : The device name to retrieve

remote_addr : An URL to a remote Server, you also have to give cert and key if you provide remote_addr and its a TCP Address!

Examples: https://myserver.lan:8443 /var/lib/mysocket.sock

cert : PEM Formatted SSL Certificate.

Examples: ~/.config/lxc/client.crt

key : PEM Formatted SSL Key.

Examples: ~/.config/lxc/client.key

verify_cert [True] Wherever to verify the cert, this is by default True but in the most cases you want to set it off as LXD normaly uses self-signed certificates.

`salt.modules.lxd.`**`container_execute`**`(`*name, cmd, remote_addr=None, cert=None, key=None, verify_cert=True*`)`

Execute a command list on a container.

name : Name of the container

cmd : Command to be executed (as a list)

Example : '["ls", "-l"]'

remote_addr : An URL to a remote Server, you also have to give cert and key if you provide remote_addr and its a TCP Address!

Examples: https://myserver.lan:8443 /var/lib/mysocket.sock

cert : PEM Formatted SSL Certificate.

Examples: ~/.config/lxc/client.crt

key : PEM Formatted SSL Key.

Examples: ~/.config/lxc/client.key

verify_cert [True] Wherever to verify the cert, this is by default True but in the most cases you want to set it off as LXD normaly uses self-signed certificates.

CLI Example:

```
salt '*' lxd.container_execute <container name> '["ls", "-l"]'
```

`salt.modules.lxd.`**`container_file_get`**`(`*name, src, dst, overwrite=False, mode=None, uid=None, gid=None, remote_addr=None, cert=None, key=None, verify_cert=True*`)`

Get a file from a container

name : Name of the container

src : The source file or directory

dst : The destination file or directory

mode : Set file mode to octal number

uid : Set file uid (owner)

gid : Set file gid (group)

remote_addr : An URL to a remote Server, you also have to give cert and key if you provide remote_addr and its a TCP Address!

Examples: https://myserver.lan:8443 /var/lib/mysocket.sock

cert : PEM Formatted SSL Certificate.

Examples: ~/.config/lxc/client.crt

key : PEM Formatted SSL Key.

Examples: ~/.config/lxc/client.key

verify_cert [True] Wherever to verify the cert, this is by default True but in the most cases you want to set it off as LXD normaly uses self-signed certificates.

`salt.modules.lxd.`**`container_file_put`**`(`*name, src, dst, recursive=False, overwrite=False, mode=None, uid=None, gid=None, saltenv='base', remote_addr=None, cert=None, key=None, verify_cert=True*`)`

Put a file into a container

name : Name of the container

src : The source file or directory

dst : The destination file or directory

recursive : Decent into src directory

overwrite : Replace destination if it exists

mode : Set file mode to octal number

uid : Set file uid (owner)

gid : Set file gid (group)

remote_addr : An URL to a remote Server, you also have to give cert and key if you provide remote_addr and its a TCP Address!

Examples: https://myserver.lan:8443 /var/lib/mysocket.sock

cert : PEM Formatted SSL Certificate.

Examples: ~/.config/lxc/client.crt

key : PEM Formatted SSL Key.

Examples: ~/.config/lxc/client.key

verify_cert [True] Wherever to verify the cert, this is by default True but in the most cases you want to set it off as LXD normaly uses self-signed certificates.

CLI Example:

```
salt '*' lxd.container_file_put <container name> /var/tmp/foo /var/tmp/
```

salt.modules.lxd.**container_freeze**(*name, remote_addr=None, cert=None, key=None, verify_cert=True*)

Freeze a container

name : Name of the container to freeze

remote_addr : An URL to a remote Server, you also have to give cert and key if you provide remote_addr and its a TCP Address!

Examples: https://myserver.lan:8443 /var/lib/mysocket.sock

cert : PEM Formatted SSL Certificate.

Examples: ~/.config/lxc/client.crt

key : PEM Formatted SSL Key.

Examples: ~/.config/lxc/client.key

verify_cert [True] Wherever to verify the cert, this is by default True but in the most cases you want to set it off as LXD normaly uses self-signed certificates.

salt.modules.lxd.**container_get**(*name=None, remote_addr=None, cert=None, key=None, verify_cert=True, _raw=False*)

Gets a container from the LXD

name : The name of the container to get.

remote_addr : An URL to a remote Server, you also have to give cert and key if you provide remote_addr and its a TCP Address!

Examples: https://myserver.lan:8443 /var/lib/mysocket.sock

cert : PEM Formatted SSL Certificate.

Examples: ~/.config/lxc/client.crt

key : PEM Formatted SSL Key.

Examples: ~/.config/lxc/client.key

verify_cert [True] Wherever to verify the cert, this is by default True but in the most cases you want to set it off as LXD normaly uses self-signed certificates.

_raw : Return the pylxd object, this is internal and by states in use.

salt.modules.lxd.**container_list**(*list_names=False, remote_addr=None, cert=None, key=None, verify_cert=True*)

Lists containers

list_names [False] Only return a list of names when True

remote_addr : An URL to a remote Server, you also have to give cert and key if you provide remote_addr and its a TCP Address!

Examples: https://myserver.lan:8443 /var/lib/mysocket.sock

cert : PEM Formatted SSL Certificate.

Examples: ~/.config/lxc/client.crt

key : PEM Formatted SSL Key.

Examples: ~/.config/lxc/client.key

verify_cert [True] Wherever to verify the cert, this is by default True but in the most cases you want to set it off as LXD normaly uses self-signed certificates.

CLI Examples:

Full dict with all available informations:

```
salt '*' lxd.container_list
```

For a list of names:

```
salt '*' lxd.container_list true
```

See also container-attributes.

`salt.modules.lxd.`**`container_migrate`**(*name*, *stop_and_start=False*, *remote_addr=None*, *cert=None*, *key=None*, *verify_cert=True*, *src_remote_addr=None*, *src_cert=None*, *src_key=None*, *src_verify_cert=None*)

Migrate a container.

If the container is running, it either must be shut down first (use stop_and_start=True) or criu must be installed on the source and destination machines.

For this operation both certs need to be authenticated, use `lxd.authenticate <salt.modules.lxd.authenticate` to authenticate your cert(s).

name : Name of the container to migrate

stop_and_start : Stop the container on the source and start it on dest

remote_addr : An URL to a remote Server, you also have to give cert and key if you provide remote_addr and its a TCP Address!

Examples: https://myserver.lan:8443 /var/lib/mysocket.sock

cert : PEM Formatted SSL Certificate.

Examples: ~/.config/lxc/client.crt

key : PEM Formatted SSL Key.

Examples: ~/.config/lxc/client.key

verify_cert [True] Wherever to verify the cert, this is by default True but in the most cases you want to set it off as LXD normaly uses self-signed certificates.

CLI Example:

```
# Authorize
salt '*' lxd.authenticate https://srv01:8443 <yourpass> ~/.config/lxc/client.crt ~
↪/.config/lxc/client.key false
salt '*' lxd.authenticate https://srv02:8443 <yourpass> ~/.config/lxc/client.crt ~
↪/.config/lxc/client.key false

# Migrate phpmyadmin from srv01 to srv02
salt '*' lxd.container_migrate phpmyadmin stop_and_start=true remote_addr=https://
↪srv02:8443 cert=~/.config/lxc/client.crt key=~/.config/lxc/client.key verify_
↪cert=False src_remote_addr=https://srv01:8443
```

`salt.modules.lxd.`**`container_rename`**(*name*, *newname*, *remote_addr=None*, *cert=None*, *key=None*, *verify_cert=True*)

Rename a container

name : Name of the container to Rename

newname : The new name of the contianer

remote_addr : An URL to a remote Server, you also have to give cert and key if you provide remote_addr and its a TCP Address!

Examples: https://myserver.lan:8443 /var/lib/mysocket.sock

cert : PEM Formatted SSL Certificate.

Examples: ~/.config/lxc/client.crt

key : PEM Formatted SSL Key.
  Examples: ~/.config/lxc/client.key

verify_cert [True] Wherever to verify the cert, this is by default True but in the most cases you want to set it off as LXD normaly uses self-signed certificates.

`salt.modules.lxd.`**`container_restart`**(*name*, *remote_addr=None*, *cert=None*, *key=None*, *verify_cert=True*)

Restart a container

name : Name of the container to restart

remote_addr : An URL to a remote Server, you also have to give cert and key if you provide remote_addr and its a TCP Address!
  Examples: https://myserver.lan:8443 /var/lib/mysocket.sock

cert : PEM Formatted SSL Certificate.
  Examples: ~/.config/lxc/client.crt

key : PEM Formatted SSL Key.
  Examples: ~/.config/lxc/client.key

verify_cert [True] Wherever to verify the cert, this is by default True but in the most cases you want to set it off as LXD normaly uses self-signed certificates.

`salt.modules.lxd.`**`container_start`**(*name*, *remote_addr=None*, *cert=None*, *key=None*, *verify_cert=True*)

Start a container

name : Name of the container to start

remote_addr : An URL to a remote Server, you also have to give cert and key if you provide remote_addr and its a TCP Address!
  Examples: https://myserver.lan:8443 /var/lib/mysocket.sock

cert : PEM Formatted SSL Certificate.
  Examples: ~/.config/lxc/client.crt

key : PEM Formatted SSL Key.
  Examples: ~/.config/lxc/client.key

verify_cert [True] Wherever to verify the cert, this is by default True but in the most cases you want to set it off as LXD normaly uses self-signed certificates.

`salt.modules.lxd.`**`container_state`**(*name=None*, *remote_addr=None*, *cert=None*, *key=None*, *verify_cert=True*)

Get container state

remote_addr : An URL to a remote Server, you also have to give cert and key if you provide remote_addr and its a TCP Address!
  Examples: https://myserver.lan:8443 /var/lib/mysocket.sock

cert : PEM Formatted SSL Certificate.
  Examples: ~/.config/lxc/client.crt

key : PEM Formatted SSL Key.
  Examples: ~/.config/lxc/client.key

verify_cert [True] Wherever to verify the cert, this is by default True but in the most cases you want to set it off as LXD normaly uses self-signed certificates.

`salt.modules.lxd.`**`container_stop`**(*name*, *timeout=30*, *force=True*, *remote_addr=None*, *cert=None*, *key=None*, *verify_cert=True*)

Stop a container

name : Name of the container to stop

remote_addr : An URL to a remote Server, you also have to give cert and key if you provide remote_addr and its a TCP Address!
  Examples: https://myserver.lan:8443 /var/lib/mysocket.sock

cert : PEM Formatted SSL Certificate.
  Examples: ~/.config/lxc/client.crt

key : PEM Formatted SSL Key.
  Examples: ~/.config/lxc/client.key

verify_cert [True] Wherever to verify the cert, this is by default True but in the most cases you want to set it off as LXD normaly uses self-signed certificates.

`salt.modules.lxd.`**`container_unfreeze`**(*name*, *remote_addr=None*, *cert=None*, *key=None*, *verify_cert=True*)

Unfreeze a container

name : Name of the container to unfreeze

remote_addr : An URL to a remote Server, you also have to give cert and key if you provide remote_addr and its a TCP Address!

Examples: https://myserver.lan:8443 /var/lib/mysocket.sock

cert : PEM Formatted SSL Certificate.

Examples: ~/.config/lxc/client.crt

key : PEM Formatted SSL Key.

Examples: ~/.config/lxc/client.key

verify_cert [True] Wherever to verify the cert, this is by default True but in the most cases you want to set it off as LXD normaly uses self-signed certificates.

`salt.modules.lxd.`**`image_alias_add`**(*image*, *alias*, *description=''*, *remote_addr=None*, *cert=None*, *key=None*, *verify_cert=True*)

Create an alias on the given image

image : An image alias, a fingerprint or a image object

alias : The alias to add

description : Description of the alias

remote_addr : An URL to a remote Server, you also have to give cert and key if you provide remote_addr and its a TCP Address!

Examples: https://myserver.lan:8443 /var/lib/mysocket.sock

cert : PEM Formatted SSL Certificate.

Examples: ~/.config/lxc/client.crt

key : PEM Formatted SSL Key.

Examples: ~/.config/lxc/client.key

verify_cert [True] Wherever to verify the cert, this is by default True but in the most cases you want to set it off as LXD normaly uses self-signed certificates.

CLI Examples:

```
$ salt '*' lxd.image_alias_add xenial/amd64 x "Short version of xenial/amd64"
```

`salt.modules.lxd.`**`image_alias_delete`**(*image*, *alias*, *remote_addr=None*, *cert=None*, *key=None*, *verify_cert=True*)

Delete an alias (this is currently not restricted to the image)

image : An image alias, a fingerprint or a image object

alias : The alias to delete

remote_addr : An URL to a remote Server, you also have to give cert and key if you provide remote_addr and its a TCP Address!

Examples: https://myserver.lan:8443 /var/lib/mysocket.sock

cert : PEM Formatted SSL Certificate.

Examples: ~/.config/lxc/client.crt

key : PEM Formatted SSL Key.

Examples: ~/.config/lxc/client.key

verify_cert [True] Wherever to verify the cert, this is by default True but in the most cases you want to set it off as LXD normaly uses self-signed certificates.

CLI Examples:

```
$ salt '*' lxd.image_alias_add xenial/amd64 x "Short version of xenial/amd64"
```

`salt.modules.lxd.`**`image_copy_lxd`**(*source, src_remote_addr, src_cert, src_key, src_verify_cert, remote_addr, cert, key, verify_cert=True, aliases=None, public=None, auto_update=None, _raw=False*)

Copy an image from another LXD instance

source : An alias or a fingerprint of the source.

src_remote_addr : An URL to the source remote daemon

Examples: https://mysourceserver.lan:8443

src_cert : PEM Formatted SSL Certificate for the source

Examples: ~/.config/lxc/client.crt

src_key : PEM Formatted SSL Key for the source

Examples: ~/.config/lxc/client.key

src_verify_cert [True] Wherever to verify the cert, this is by default True but in the most cases you want to set it off as LXD normaly uses self-signed certificates.

remote_addr : Address of the destination daemon

Examples: https://mydestserver.lan:8443

cert : PEM Formatted SSL Certificate for the destination

Examples: ~/.config/lxc/client.crt

key : PEM Formatted SSL Key for the destination

Examples: ~/.config/lxc/client.key

verify_cert [True] Wherever to verify the cert, this is by default True but in the most cases you want to set it off as LXD normaly uses self-signed certificates.

aliases [[]] List of aliases to append to the copied image

public [None] Make this image public available, None = copy source

auto_update [None] Wherever to auto-update from the original source, None = copy source

_raw [False] Return the raw pylxd object or a dict of the destination image?

CLI Examples:

```
$ salt '*' lxd.image_copy_lxd xenial/amd64 https://srv01:8443 ~/.config/lxc/
↪client.crt ~/.config/lxc/client.key false https://srv02:8443 ~/.config/lxc/
↪client.crt ~/.config/lxc/client.key false aliases="['xenial/amd64']"
```

`salt.modules.lxd.`**`image_delete`**(*image, remote_addr=None, cert=None, key=None, verify_cert=True*)

Delete an image by an alias or fingerprint

name : The alias or fingerprint of the image to delete, can be a obj for the states.

remote_addr : An URL to a remote Server, you also have to give cert and key if you provide remote_addr and its a TCP Address!

Examples: https://myserver.lan:8443 /var/lib/mysocket.sock

cert : PEM Formatted SSL Certificate.

Examples: ~/.config/lxc/client.crt

key : PEM Formatted SSL Key.

Examples: ~/.config/lxc/client.key

verify_cert [True] Wherever to verify the cert, this is by default True but in the most cases you want to set it off as LXD normaly uses self-signed certificates.

CLI Examples:

..code-block:: bash

$ salt '*' lxd.image_delete xenial/amd64

`salt.modules.lxd.`**`image_from_file`**(*filename, remote_addr=None, cert=None, key=None, verify_cert=True, aliases=None, public=False, saltenv='base', _raw=False*)

Create an image from a file

filename : The filename of the rootfs

remote_addr : An URL to a remote Server, you also have to give cert and key if you provide remote_addr and its a TCP Address!

Examples: https://myserver.lan:8443 /var/lib/mysocket.sock

cert : PEM Formatted SSL Certificate.

Examples: ~/.config/lxc/client.crt

key : PEM Formatted SSL Key.

Examples: ~/.config/lxc/client.key

verify_cert [True] Wherever to verify the cert, this is by default True but in the most cases you want to set it off as LXD normaly uses self-signed certificates.

aliases [[]] List of aliases to append to the copied image

public [False] Make this image public available

saltenv [base] The saltenv to use for salt:// copies

_raw [False] Return the raw pylxd object or a dict of the image?

CLI Examples:

..code-block:: bash

$ salt '*' lxd.image_from_file salt://lxd/files/busybox.tar.xz aliases=["busybox-amd64"]

`salt.modules.lxd.`**`image_from_simplestreams`**(*server*, *alias*, *remote_addr=None*, *cert=None*, *key=None*, *verify_cert=True*, *aliases=None*, *public=False*, *auto_update=False*, *_raw=False*)

Create an image from simplestreams

server : Simplestreams server URI

alias : The alias of the image to retrieve

remote_addr : An URL to a remote Server, you also have to give cert and key if you provide remote_addr and its a TCP Address!

Examples: https://myserver.lan:8443 /var/lib/mysocket.sock

cert : PEM Formatted SSL Certificate.

Examples: ~/.config/lxc/client.crt

key : PEM Formatted SSL Key.

Examples: ~/.config/lxc/client.key

verify_cert [True] Wherever to verify the cert, this is by default True but in the most cases you want to set it off as LXD normaly uses self-signed certificates.

aliases [[]] List of aliases to append to the copied image

public [False] Make this image public available

auto_update [False] Should LXD auto update that image?

_raw [False] Return the raw pylxd object or a dict of the image?

CLI Examples:

..code-block:: bash

$ salt '*' lxd.image_from_simplestreams "https://cloud-images.ubuntu.com/releases" "trusty/amd64" aliases='["t", "trusty/amd64"]' auto_update=True

`salt.modules.lxd.`**`image_from_url`**(*url*, *remote_addr=None*, *cert=None*, *key=None*, *verify_cert=True*, *aliases=None*, *public=False*, *auto_update=False*, *_raw=False*)

Create an image from an url

url : The URL from where to download the image

remote_addr : An URL to a remote Server, you also have to give cert and key if you provide remote_addr and its a TCP Address!

Examples: https://myserver.lan:8443 /var/lib/mysocket.sock

cert : PEM Formatted SSL Certificate.

Examples: ~/.config/lxc/client.crt

key : PEM Formatted SSL Key.

Examples: ~/.config/lxc/client.key

verify_cert [True] Wherever to verify the cert, this is by default True but in the most cases you want to set it off as LXD normaly uses self-signed certificates.

aliases [[]] List of aliases to append to the copied image

public [False] Make this image public available

auto_update [False] Should LXD auto update that image?

_raw [False] Return the raw pylxd object or a dict of the image?

CLI Examples:

..code-block:: bash

$ salt '*' lxd.image_from_url https://dl.stgraber.org/lxd aliases='["busybox-amd64"]'

`salt.modules.lxd.`**`image_get`**(*fingerprint*, *remote_addr=None*, *cert=None*, *key=None*, *verify_cert=True*, *_raw=False*)

Get an image by its fingerprint

fingerprint : The fingerprint of the image to retrieve

remote_addr : An URL to a remote Server, you also have to give cert and key if you provide remote_addr and its a TCP Address!

Examples: https://myserver.lan:8443 /var/lib/mysocket.sock

cert : PEM Formatted SSL Certificate.

Examples: ~/.config/lxc/client.crt

key : PEM Formatted SSL Key.

Examples: ~/.config/lxc/client.key

verify_cert [True] Wherever to verify the cert, this is by default True but in the most cases you want to set it off as LXD normaly uses self-signed certificates.

_raw [False] Return the raw pylxd object or a dict of it?

CLI Examples:

..code-block:: bash

$ salt '*' lxd.image_get <fingerprint>

`salt.modules.lxd.`**`image_get_by_alias`**(*alias*, *remote_addr=None*, *cert=None*, *key=None*, *verify_cert=True*, *_raw=False*)

Get an image by an alias

alias : The alias of the image to retrieve

remote_addr : An URL to a remote Server, you also have to give cert and key if you provide remote_addr and its a TCP Address!

Examples: https://myserver.lan:8443 /var/lib/mysocket.sock

cert : PEM Formatted SSL Certificate.

Examples: ~/.config/lxc/client.crt

key : PEM Formatted SSL Key.

Examples: ~/.config/lxc/client.key

verify_cert [True] Wherever to verify the cert, this is by default True but in the most cases you want to set it off as LXD normaly uses self-signed certificates.

_raw [False] Return the raw pylxd object or a dict of it?

CLI Examples:

..code-block:: bash

$ salt '*' lxd.image_get_by_alias xenial/amd64

`salt.modules.lxd.`**`image_list`**(*list_aliases=False*, *remote_addr=None*, *cert=None*, *key=None*, *verify_cert=True*)

Lists all images from the LXD.

list_aliases :

Return a dict with the fingerprint as key and a list of aliases as value instead.

remote_addr : An URL to a remote Server, you also have to give cert and key if you provide remote_addr and its a TCP Address!

Examples: https://myserver.lan:8443 /var/lib/mysocket.sock

cert : PEM Formatted SSL Certificate.
Examples: ~/.config/lxc/client.crt
key : PEM Formatted SSL Key.
Examples: ~/.config/lxc/client.key
verify_cert [True] Wherever to verify the cert, this is by default True but in the most cases you want to set it off as LXD normaly uses self-signed certificates.

CLI Examples:

```
$ salt '*' lxd.image_list true --out=json
$ salt '*' lxd.image_list --out=json
```

`salt.modules.lxd.`**`init`**(*storage_backend='dir', trust_password=None, network_address=None, network_port=None, storage_create_device=None, storage_create_loop=None, storage_pool=None*)

Calls lxd init –auto – opts
storage_backend : Storage backend to use (zfs or dir, default: dir)
trust_password : Password required to add new clients
network_address [None] Address to bind LXD to (default: none)
network_port [None] Port to bind LXD to (Default: 8443)
storage_create_device [None] Setup device based storage using this DEVICE
storage_create_loop [None] Setup loop based storage with this SIZE in GB
storage_pool [None] Storage pool to use or create
CLI Examples:

To listen on all IPv4/IPv6 Addresses:

```
salt '*' lxd.init dir PaSsW0rD [::]
```

To not listen on Network:

```
salt '*' lxd.init
```

`salt.modules.lxd.`**`normalize_input_values`**(*config, devices*)

normalize config input so returns can be put into mongodb, which doesn't like .

This is not meant to be used on the commandline.

CLI Examples:

```
salt '*' lxd.normalize_input_values config={} devices={}
```

`salt.modules.lxd.`**`profile_config_delete`**(*name, config_key, remote_addr=None, cert=None, key=None, verify_cert=True*)

Delete a profile config item.
name : The name of the profile to delete the config item.
config_key : The config key for the value to retrieve.
remote_addr : An URL to a remote Server, you also have to give cert and key if you provide remote_addr and its a TCP Address!
Examples: https://myserver.lan:8443 /var/lib/mysocket.sock
cert : PEM Formatted SSL Certificate.
Examples: ~/.config/lxc/client.crt
key : PEM Formatted SSL Key.
Examples: ~/.config/lxc/client.key
verify_cert [True] Wherever to verify the cert, this is by default True but in the most cases you want to set it off as LXD normaly uses self-signed certificates.
CLI Example:

```
$ salt '*' lxd.profile_config_delete autostart boot.autostart.delay
```

salt.modules.lxd.**profile_config_get**(*name*, *config_key*, *remote_addr=None*, *cert=None*, *key=None*, *verify_cert=True*)

Get a profile config item.

name : The name of the profile to get the config item from.

config_key : The key for the item to retrieve.

remote_addr : An URL to a remote Server, you also have to give cert and key if you provide remote_addr and its a TCP Address!

Examples: https://myserver.lan:8443 /var/lib/mysocket.sock

cert : PEM Formatted SSL Certificate.

Examples: ~/.config/lxc/client.crt

key : PEM Formatted SSL Key.

Examples: ~/.config/lxc/client.key

verify_cert [True] Wherever to verify the cert, this is by default True but in the most cases you want to set it off as LXD normaly uses self-signed certificates.

CLI Example:

```
$ salt '*' lxd.profile_config_get autostart boot.autostart
```

salt.modules.lxd.**profile_config_set**(*name*, *config_key*, *config_value*, *remote_addr=None*, *cert=None*, *key=None*, *verify_cert=True*)

Set a profile config item.

name : The name of the profile to set the config item to.

config_key : The items key.

config_value : Its items value.

remote_addr : An URL to a remote Server, you also have to give cert and key if you provide remote_addr and its a TCP Address!

Examples: https://myserver.lan:8443 /var/lib/mysocket.sock

cert : PEM Formatted SSL Certificate.

Examples: ~/.config/lxc/client.crt

key : PEM Formatted SSL Key.

Examples: ~/.config/lxc/client.key

verify_cert [True] Wherever to verify the cert, this is by default True but in the most cases you want to set it off as LXD normaly uses self-signed certificates.

CLI Example:

```
$ salt '*' lxd.profile_config_set autostart boot.autostart 0
```

salt.modules.lxd.**profile_create**(*name*, *config=None*, *devices=None*, *description=None*, *remote_addr=None*, *cert=None*, *key=None*, *verify_cert=True*)

Creates a profile.

name : The name of the profile to get.

config : A config dict or None (None = unset).

Can also be a list:

[{'**key**': '**boot.autostart**', '**value**': **1**}, {'key': 'security.privileged', 'value': '1'}]

devices : A device dict or None (None = unset).

description : A description string or None (None = unset).

remote_addr : An URL to a remote Server, you also have to give cert and key if you provide remote_addr and its a TCP Address!

Examples: https://myserver.lan:8443 /var/lib/mysocket.sock

cert : PEM Formatted SSL Certificate.

Examples: ~/.config/lxc/client.crt

key : PEM Formatted SSL Key.

Examples: ~/.config/lxc/client.key

verify_cert [True] Wherever to verify the cert, this is by default True but in the most cases you want to set it off as LXD normaly uses self-signed certificates.

CLI Examples:

```
$ salt '*' lxd.profile_create autostart config="{boot.autostart: 1, boot.
↪autostart.delay: 2, boot.autostart.priority: 1}"
$ salt '*' lxd.profile_create shared_mounts devices="{shared_mount: {type: 'disk',
↪ source: '/home/shared', path: '/home/shared'}}"
```

See the lxd-docs for the details about the config and devices dicts.

`salt.modules.lxd.`**`profile_delete`**`(`*name, remote_addr=None, cert=None, key=None, verify_cert=True*`)`

Deletes a profile.

name : The name of the profile to delete.

remote_addr : An URL to a remote Server, you also have to give cert and key if you provide remote_addr and its a TCP Address!

Examples: https://myserver.lan:8443 /var/lib/mysocket.sock

cert : PEM Formatted SSL Certificate.

Examples: ~/.config/lxc/client.crt

key : PEM Formatted SSL Key.

Examples: ~/.config/lxc/client.key

verify_cert [True] Wherever to verify the cert, this is by default True but in the most cases you want to set it off as LXD normaly uses self-signed certificates.

CLI Example:

```
$ salt '*' lxd.profile_delete shared_mounts
```

`salt.modules.lxd.`**`profile_device_delete`**`(`*name, device_name, remote_addr=None, cert=None, key=None, verify_cert=True*`)`

Delete a profile device.

name : The name of the profile to delete the device.

device_name : The name of the device to delete.

remote_addr : An URL to a remote Server, you also have to give cert and key if you provide remote_addr and its a TCP Address!

Examples: https://myserver.lan:8443 /var/lib/mysocket.sock

cert : PEM Formatted SSL Certificate.

Examples: ~/.config/lxc/client.crt

key : PEM Formatted SSL Key.

Examples: ~/.config/lxc/client.key

verify_cert [True] Wherever to verify the cert, this is by default True but in the most cases you want to set it off as LXD normaly uses self-signed certificates.

CLI Example:

```
$ salt '*' lxd.profile_device_delete autostart eth1
```

`salt.modules.lxd.`**`profile_device_get`**`(`*name, device_name, remote_addr=None, cert=None, key=None, verify_cert=True*`)`

Get a profile device.

name : The name of the profile to get the device from.

device_name : The name of the device to retrieve.

remote_addr : An URL to a remote Server, you also have to give cert and key if you provide remote_addr and its a TCP Address!

Examples: https://myserver.lan:8443 /var/lib/mysocket.sock

cert : PEM Formatted SSL Certificate.

Examples: ~/.config/lxc/client.crt

key : PEM Formatted SSL Key.

Examples: ~/.config/lxc/client.key

verify_cert [True] Wherever to verify the cert, this is by default True but in the most cases you want to set it off as LXD normaly uses self-signed certificates.

CLI Example:

```
$ salt '*' lxd.profile_device_get default eth0
```

`salt.modules.lxd.`**`profile_device_set`**(*name, device_name, device_type='disk', remote_addr=None, cert=None, key=None, verify_cert=True, **kwargs*)

Set a profile device.

name : The name of the profile to set the device to.

device_name : The name of the device to set.

remote_addr : An URL to a remote Server, you also have to give cert and key if you provide remote_addr and its a TCP Address!

Examples: https://myserver.lan:8443 /var/lib/mysocket.sock

cert : PEM Formatted SSL Certificate.

Examples: ~/.config/lxc/client.crt

key : PEM Formatted SSL Key.

Examples: ~/.config/lxc/client.key

verify_cert [True] Wherever to verify the cert, this is by default True but in the most cases you want to set it off as LXD normaly uses self-signed certificates.

CLI Example:

```
$ salt '*' lxd.profile_device_set autostart eth1 nic nictype=bridged parent=lxdbr0
```

`salt.modules.lxd.`**`profile_get`**(*name, remote_addr=None, cert=None, key=None, verify_cert=True, _raw=False*)

Gets a profile from the LXD

name : The name of the profile to get.

remote_addr : An URL to a remote Server, you also have to give cert and key if you provide remote_addr and its a TCP Address!

Examples: https://myserver.lan:8443 /var/lib/mysocket.sock

cert : PEM Formatted SSL Certificate.

Examples: ~/.config/lxc/client.crt

key : PEM Formatted SSL Key.

Examples: ~/.config/lxc/client.key

verify_cert [True] Wherever to verify the cert, this is by default True but in the most cases you want to set it off as LXD normaly uses self-signed certificates.

_raw : Return the pylxd object, this is internal and by states in use.

CLI Examples:

```
$ salt '*' lxd.profile_get autostart
```

`salt.modules.lxd.`**`profile_list`**(*list_names=False, remote_addr=None, cert=None, key=None, verify_cert=True*)

Lists all profiles from the LXD.

list_names :

Return a list of names instead of full blown dicts.

remote_addr : An URL to a remote Server, you also have to give cert and key if you provide remote_addr and its a TCP Address!

Examples: https://myserver.lan:8443 /var/lib/mysocket.sock

cert : PEM Formatted SSL Certificate.
Examples: ~/.config/lxc/client.crt

key : PEM Formatted SSL Key.
Examples: ~/.config/lxc/client.key

verify_cert [True] Wherever to verify the cert, this is by default True but in the most cases you want to set it off as LXD normaly uses self-signed certificates.

CLI Examples:

```
salt '*' lxd.profile_list true --out=json
salt '*' lxd.profile_list --out=json
```

`salt.modules.lxd.`**`pylxd_client_get`**(*remote_addr=None, cert=None, key=None, verify_cert=True*)

Get an pyxld client, this is not ment to be runned over the CLI.

remote_addr : An URL to a remote Server, you also have to give cert and key if you provide remote_addr and its a TCP Address!
Examples: https://myserver.lan:8443 /var/lib/mysocket.sock

cert : PEM Formatted SSL Certificate.
Examples: ~/.config/lxc/client.crt

key : PEM Formatted SSL Key.
Examples: ~/.config/lxc/client.key

verify_cert [True] Wherever to verify the cert, this is by default True but in the most cases you want to set it off as LXD normaly uses self-signed certificates.

See the requests-docs for the SSL stuff.

`salt.modules.lxd.`**`pylxd_save_object`**(*obj*)

Saves an object (profile/image/container) and translate its execpetion on failure

obj : The object to save

This is an internal method, no CLI Example.

`salt.modules.lxd.`**`pylxd_version`**()

Returns the actual pylxd version.

CLI Example:

```
salt '*' lxd.pylxd_version
```

`salt.modules.lxd.`**`snapshots_all`**(*container, remote_addr=None, cert=None, key=None, verify_cert=True*)

Get all snapshots for a container

container : The name of the container to get.

remote_addr : An URL to a remote server. The 'cert' and 'key' fields must also be provided if 'remote_addr' is defined.
Examples: https://myserver.lan:8443 /var/lib/mysocket.sock

cert : PEM Formatted SSL Certificate.
Examples: ~/.config/lxc/client.crt

key : PEM Formatted SSL Key.
Examples: ~/.config/lxc/client.key

verify_cert [True] Verify the ssl certificate. Default: True

CLI Examples:

```
$ salt '*' lxd.snapshots_all test-container
```

`salt.modules.lxd.`**`snapshots_create`**(*container, name=None, remote_addr=None, cert=None, key=None, verify_cert=True*)

Create a snapshot for a container

container : The name of the container to get.

name : The name of the snapshot.
remote_addr : An URL to a remote server. The 'cert' and 'key' fields must also be provided if 'remote_addr' is defined.
Examples: https://myserver.lan:8443 /var/lib/mysocket.sock
cert : PEM Formatted SSL Certificate.
Examples: ~/.config/lxc/client.crt
key : PEM Formatted SSL Key.
Examples: ~/.config/lxc/client.key
verify_cert [True] Verify the ssl certificate. Default: True
CLI Examples:

```
$ salt '*' lxd.snapshots_create test-container test-snapshot
```

`salt.modules.lxd.`**`snapshots_delete`**(*container*, *name*, *remote_addr=None*, *cert=None*, *key=None*, *verify_cert=True*)

Delete a snapshot for a container
container : The name of the container to get.
name : The name of the snapshot.
remote_addr : An URL to a remote server. The 'cert' and 'key' fields must also be provided if 'remote_addr' is defined.
Examples: https://myserver.lan:8443 /var/lib/mysocket.sock
cert : PEM Formatted SSL Certificate.
Examples: ~/.config/lxc/client.crt
key : PEM Formatted SSL Key.
Examples: ~/.config/lxc/client.key
verify_cert [True] Verify the ssl certificate. Default: True
CLI Examples:

```
$ salt '*' lxd.snapshots_delete test-container test-snapshot
```

`salt.modules.lxd.`**`snapshots_get`**(*container*, *name*, *remote_addr=None*, *cert=None*, *key=None*, *verify_cert=True*)

Get information about snapshot for a container
container : The name of the container to get.
name : The name of the snapshot.
remote_addr : An URL to a remote server. The 'cert' and 'key' fields must also be provided if 'remote_addr' is defined.
Examples: https://myserver.lan:8443 /var/lib/mysocket.sock
cert : PEM Formatted SSL Certificate.
Examples: ~/.config/lxc/client.crt
key : PEM Formatted SSL Key.
Examples: ~/.config/lxc/client.key
verify_cert [True] Verify the ssl certificate. Default: True
CLI Examples:

```
$ salt '*' lxd.snapshots_get test-container test-snapshot
```

`salt.modules.lxd.`**`sync_config_devices`**(*obj*, *newconfig*, *newdevices*, *test=False*)

Syncs the given config and devices with the object (a profile or a container) returns a changes dict with all changes made.
obj : The object to sync with / or just test with.
newconfig: The new config to check with the obj.
newdevices: The new devices to check with the obj.
test: Wherever to not change anything and give "Would change" message.

`salt.modules.lxd.`**`version`**()

Returns the actual lxd version.

CLI Example:

```
salt '*' lxd.version
```

21.9.217 salt.modules.mac_assistive module

This module allows you to manage assistive access on macOS minions with 10.9+

New in version 2016.3.0.

```
salt '*' assistive.install /usr/bin/osascript
```

`salt.modules.mac_assistive.`**`enable`**`(`*app_id*, *enabled=True*`)`

Enable or disable an existing assistive access application.

app_id The bundle ID or command to set assistive access status.

enabled Sets enabled or disabled status. Default is `True`.

CLI Example:

```
salt '*' assistive.enable /usr/bin/osascript
salt '*' assistive.enable com.smileonmymac.textexpander enabled=False
```

`salt.modules.mac_assistive.`**`enabled`**`(`*app_id*`)`

Check if a bundle ID or command is listed in assistive access and enabled.

app_id The bundle ID or command to retrieve assistive access status.

CLI Example:

```
salt '*' assistive.enabled /usr/bin/osascript
salt '*' assistive.enabled com.smileonmymac.textexpander
```

`salt.modules.mac_assistive.`**`install`**`(`*app_id*, *enable=True*`)`

Install a bundle ID or command as being allowed to use assistive access.

app_id The bundle ID or command to install for assistive access.

enabled Sets enabled or disabled status. Default is `True`.

CLI Example:

```
salt '*' assistive.install /usr/bin/osascript
salt '*' assistive.install com.smileonmymac.textexpander
```

`salt.modules.mac_assistive.`**`installed`**`(`*app_id*`)`

Check if a bundle ID or command is listed in assistive access. This will not check to see if it's enabled.

app_id The bundle ID or command to check installed status.

CLI Example:

```
salt '*' assistive.installed /usr/bin/osascript
salt '*' assistive.installed com.smileonmymac.textexpander
```

`salt.modules.mac_assistive.`**`remove`**`(`*app_id*`)`

Remove a bundle ID or command as being allowed to use assistive access.

app_id The bundle ID or command to remove from assistive access list.

CLI Example:

```
salt '*' assistive.remove /usr/bin/osascript
salt '*' assistive.remove com.smileonmymac.textexpander
```

21.9.218 salt.modules.mac_brew_pkg

Homebrew for macOS

Important: If you feel that Salt should be using this module to manage packages on a minion, and it is using a different module (or gives an error similar to *'pkg.install' is not available*), see *here*.

`salt.modules.mac_brew_pkg.`**`available_version`**`(`**names*, ***kwargs*`)`

This function is an alias of `latest_version`.

Return the latest version of the named package available for upgrade or installation

Currently chooses stable versions, falling back to devel if that does not exist.

CLI Example:

```
salt '*' pkg.latest_version <package name>
salt '*' pkg.latest_version <package1> <package2> <package3>
```

`salt.modules.mac_brew_pkg.`**`info_installed`**`(`**names*`)`

Return the information of the named package(s) installed on the system.

New in version 2016.3.1.

names The names of the packages for which to return information.

CLI example:

```
salt '*' pkg.info_installed <package1>
salt '*' pkg.info_installed <package1> <package2> <package3> ...
```

`salt.modules.mac_brew_pkg.`**`install`**`(`*name=None*, *pkgs=None*, *taps=None*, *options=None*, ***kwargs*`)`

Install the passed package(s) with `brew install`

name The name of the formula to be installed. Note that this parameter is ignored if "pkgs" is passed.

CLI Example:

```
salt '*' pkg.install <package name>
```

taps Unofficial GitHub repos to use when updating and installing formulas.

CLI Example:

```
salt '*' pkg.install <package name> tap='<tap>'
salt '*' pkg.install zlib taps='homebrew/dupes'
salt '*' pkg.install php54 taps='["josegonzalez/php", "homebrew/dupes"]'
```

options Options to pass to brew. Only applies to initial install. Due to how brew works, modifying chosen options requires a full uninstall followed by a fresh install. Note that if "pkgs" is used, all options will be passed to all packages. Unrecognized options for a package will be silently ignored by brew.

CLI Example:

```
salt '*' pkg.install <package name> tap='<tap>'
salt '*' pkg.install php54 taps='["josegonzalez/php", "homebrew/dupes"]'
↪options='["--with-fpm"]'
```

Multiple Package Installation Options:

pkgs A list of formulas to install. Must be passed as a python list.

CLI Example:

```
salt '*' pkg.install pkgs='["foo","bar"]'
```

Returns a dict containing the new package names and versions:

```
{'<package>': {'old': '<old-version>',
               'new': '<new-version>'}}
```

CLI Example:

```
salt '*' pkg.install 'package package package'
```

`salt.modules.mac_brew_pkg.`**`latest_version`**(**names*, ***kwargs*)

Return the latest version of the named package available for upgrade or installation

Currently chooses stable versions, falling back to devel if that does not exist.

CLI Example:

```
salt '*' pkg.latest_version <package name>
salt '*' pkg.latest_version <package1> <package2> <package3>
```

`salt.modules.mac_brew_pkg.`**`list_pkgs`**(*versions_as_list=False*, ***kwargs*)

List the packages currently installed in a dict:

```
{'<package_name>': '<version>'}
```

CLI Example:

```
salt '*' pkg.list_pkgs
```

`salt.modules.mac_brew_pkg.`**`list_upgrades`**(*refresh=True*, ***kwargs*)

Check whether or not an upgrade is available for all packages

CLI Example:

```
salt '*' pkg.list_upgrades
```

`salt.modules.mac_brew_pkg.`**`refresh_db`**()

Update the homebrew package repository.

CLI Example:

```
salt '*' pkg.refresh_db
```

`salt.modules.mac_brew_pkg.`**`remove`**(*name=None*, *pkgs=None*, ***kwargs*)

Removes packages with `brew uninstall`.

name The name of the package to be deleted.

Multiple Package Options:

pkgs A list of packages to delete. Must be passed as a python list. The `name` parameter will be ignored if this option is passed.

New in version 0.16.0.

Returns a dict containing the changes.

CLI Example:

```
salt '*' pkg.remove <package name>
salt '*' pkg.remove <package1>,<package2>,<package3>
salt '*' pkg.remove pkgs='["foo", "bar"]'
```

`salt.modules.mac_brew_pkg.`**`upgrade`**`(`*refresh=True*`)`

Upgrade outdated, unpinned brews.

refresh Fetch the newest version of Homebrew and all formulae from GitHub before installing.

Returns a dictionary containing the changes:

```
{'<package>':  {'old': '<old-version>',
                'new': '<new-version>'}}
```

CLI Example:

```
salt '*' pkg.upgrade
```

`salt.modules.mac_brew_pkg.`**`upgrade_available`**`(`*pkg*`)`

Check whether or not an upgrade is available for a given package

CLI Example:

```
salt '*' pkg.upgrade_available <package name>
```

`salt.modules.mac_brew_pkg.`**`version`**`(`**names, **kwargs*`)`

Returns a string representing the package version or an empty string if not installed. If more than one package name is specified, a dict of name/version pairs is returned.

CLI Example:

```
salt '*' pkg.version <package name>
salt '*' pkg.version <package1> <package2> <package3>
```

21.9.219 salt.modules.macdefaults

Set defaults on Mac OS

`salt.modules.macdefaults.`**`delete`**`(`*domain, key, user=None*`)`

Delete a default from the system

CLI Example:

```
salt '*' macdefaults.delete com.apple.CrashReporter DialogType

salt '*' macdefaults.delete NSGlobalDomain ApplePersistence
```

domain The name of the domain to delete from

key The key of the given domain to delete

user The user to delete the defaults with

`salt.modules.macdefaults.`**`read`**`(`*domain, key, user=None*`)`

Write a default to the system

CLI Example:

```
salt '*' macdefaults.read com.apple.CrashReporter DialogType

salt '*' macdefaults.read NSGlobalDomain ApplePersistence
```

domain The name of the domain to read from

key The key of the given domain to read from

user The user to write the defaults to `salt.modules.macdefaults.`**`write`**(*domain, key, value, type='string', user=None*)

Write a default to the system

CLI Example:

```
salt '*' macdefaults.write com.apple.CrashReporter DialogType Server

salt '*' macdefaults.write NSGlobalDomain ApplePersistence True type=bool
```

domain The name of the domain to write to
key The key of the given domain to write to
value The value to write to the given key
type The type of value to be written, valid types are string, data, int[eger], float, bool[ean], date, array, array-add, dict, dict-add
user The user to write the defaults to

21.9.220 salt.modules.mac_desktop module

macOS implementations of various commands in the "desktop" interface

`salt.modules.mac_desktop.`**`get_output_volume`**()

Get the output volume (range 0 to 100)

CLI Example:

```
salt '*' desktop.get_output_volume
```

`salt.modules.mac_desktop.`**`lock`**()

Lock the desktop session

CLI Example:

```
salt '*' desktop.lock
```

`salt.modules.mac_desktop.`**`say`**(**words*)

Say some words.
words The words to execute the say command with.
CLI Example:

```
salt '*' desktop.say <word0> <word1> ... <wordN>
```

`salt.modules.mac_desktop.`**`screensaver`**()

Launch the screensaver.

CLI Example:

```
salt '*' desktop.screensaver
```

`salt.modules.mac_desktop.`**`set_output_volume`**(*volume*)

Set the volume of sound.
volume The level of volume. Can range from 0 to 100.
CLI Example:

```
salt '*' desktop.set_output_volume <volume>
```

21.9.221 salt.modules.mac_group

Manage groups on Mac OS 10.7+

`salt.modules.mac_group.`**`add`**(*name*, *gid=None*, ***kwargs*)

Add the specified group

CLI Example:

```
salt '*' group.add foo 3456
```

`salt.modules.mac_group.`**`adduser`**(*group*, *name*)

Add a user in the group.

CLI Example:

```
salt '*' group.adduser foo bar
```

Verifies if a valid username 'bar' as a member of an existing group 'foo', if not then adds it.

`salt.modules.mac_group.`**`chgid`**(*name*, *gid*)

Change the gid for a named group

CLI Example:

```
salt '*' group.chgid foo 4376
```

`salt.modules.mac_group.`**`delete`**(*name*)

Remove the named group

CLI Example:

```
salt '*' group.delete foo
```

`salt.modules.mac_group.`**`deluser`**(*group*, *name*)

Remove a user from the group

New in version 2016.3.0.

CLI Example:

```
salt '*' group.deluser foo bar
```

Removes a member user 'bar' from a group 'foo'. If group is not present then returns True.

`salt.modules.mac_group.`**`getent`**(*refresh=False*)

Return info on all groups

CLI Example:

```
salt '*' group.getent
```

`salt.modules.mac_group.`**`info`**(*name*)

Return information about a group

CLI Example:

```
salt '*' group.info foo
```

`salt.modules.mac_group.`**`members`**(*name*, *members_list*)

Replaces members of the group with a provided list.

New in version 2016.3.0.

CLI Example:
salt '*' group.members foo 'user1,user2,user3,...'
Replaces a membership list for a local group 'foo'.

21.9.222 salt.modules.mac_keychain module

Install certificates into the keychain on Mac OS

New in version 2016.3.0.

salt.modules.mac_keychain.**get_default_keychain**(*user=None, domain='user'*)
Get the default keychain
user The user to check the default keychain of
domain The domain to use valid values are user|system|common|dynamic, the default is user
CLI Example:

```
salt '*' keychain.get_default_keychain
```

salt.modules.mac_keychain.**get_friendly_name**(*cert, password*)
Get the friendly name of the given certificate
cert The certificate to install
password The password for the certificate being installed formatted in the way described for openssl command in the PASS PHRASE ARGUMENTS section

Note: The password given here will show up as plaintext in the returned job info.
CLI Example:

```
salt '*' keychain.get_friendly_name /tmp/test.p12 test123
```

salt.modules.mac_keychain.**get_hash**(*name, password=None*)
Returns the hash of a certificate in the keychain.
name The name of the certificate (which you can get from keychain.get_friendly_name) or the location of a p12 file.
password The password that is used in the certificate. Only required if your passing a p12 file. Note: This will be outputted to logs
CLI Example:

```
salt '*' keychain.get_hash /tmp/test.p12 test123
```

salt.modules.mac_keychain.**install**(*cert, password, keychain='/Library/Keychains/System.keychain', allow_any=False, keychain_password=None*)
Install a certificate
cert The certificate to install
password The password for the certificate being installed formatted in the way described for openssl command in the PASS PHRASE ARGUMENTS section.

Note: The password given here will show up as plaintext in the job returned info.
keychain The keychain to install the certificate to, this defaults to /Library/Keychains/System.keychain
allow_any Allow any application to access the imported certificate without warning
keychain_password If your keychain is likely to be locked pass the password and it will be unlocked before running the import

Note: The password given here will show up as plaintext in the returned job info.
CLI Example:

```
salt '*' keychain.install test.p12 test123
```

`salt.modules.mac_keychain.`**`list_certs`**`(`*keychain='/Library/Keychains/System.keychain'*`)`

List all of the installed certificates

keychain The keychain to install the certificate to, this defaults to /Library/Keychains/System.keychain

CLI Example:

```
salt '*' keychain.list_certs
```

`salt.modules.mac_keychain.`**`set_default_keychain`**`(`*keychain, domain='user', user=None*`)`

Set the default keychain

keychain The location of the keychain to set as default

domain The domain to use valid values are user|system|common|dynamic, the default is user

user The user to set the default keychain as

CLI Example:

```
salt '*' keychain.set_keychain /Users/fred/Library/Keychains/login.keychain
```

`salt.modules.mac_keychain.`**`uninstall`**`(`*cert_name, keychain='/Library/Keychains/System.keychain', keychain_password=None*`)`

Uninstall a certificate from a keychain

cert_name The name of the certificate to remove

keychain The keychain to install the certificate to, this defaults to /Library/Keychains/System.keychain

keychain_password If your keychain is likely to be locked pass the password and it will be unlocked before running the import

Note: The password given here will show up as plaintext in the returned job info.

CLI Example:

```
salt '*' keychain.install test.p12 test123
```

`salt.modules.mac_keychain.`**`unlock_keychain`**`(`*keychain, password*`)`

Unlock the given keychain with the password

keychain The keychain to unlock

password The password to use to unlock the keychain.

Note: The password given here will show up as plaintext in the returned job info.

CLI Example:

```
salt '*' keychain.unlock_keychain /tmp/test.p12 test123
```

21.9.223 salt.modules.macpackage

Install pkg, dmg and .app applications on macOS minions.

`salt.modules.macpackage.`**`get_mpkg_ids`**`(`*mpkg*`)`

Attempt to get the package IDs from a mounted .mpkg file

Parameters **`mpkg`** (`str`) – The location of the mounted mpkg file

Returns List of package IDs

Return type list

CLI Example:

```
salt '*' macpackage.get_mpkg_ids /dev/disk2
```

`salt.modules.macpackage.`**`get_pkg_id`**`(`*pkg*`)`

Attempt to get the package ID from a .pkg file

Parameters **`pkg`** (*`str`*) – The location of the pkg file

Returns List of all of the package IDs

Return type list

CLI Example:

```
salt '*' macpackage.get_pkg_id /tmp/test.pkg
```

`salt.modules.macpackage.`**`install`**`(`*pkg, target='LocalSystem', store=False, allow_untrusted=False*`)`

Install a pkg file

Parameters

- **`pkg`** (*`str`*) – The package to install
- **`target`** (*`str`*) – The target in which to install the package to
- **`store`** (*`bool`*) – Should the package be installed as if it was from the store?
- **`allow_untrusted`** (*`bool`*) – Allow the installation of untrusted packages?

Returns A dictionary containing the results of the installation

Return type dict

CLI Example:

```
salt '*' macpackage.install test.pkg
```

`salt.modules.macpackage.`**`install_app`**`(`*app, target='/Applications/'*`)`

Install an app file by moving it into the specified Applications directory

Parameters

- **`app`** (*`str`*) – The location of the .app file
- **`target`** (*`str`*) – The target in which to install the package to Default is "/Applications/"

Returns The results of the rsync command

Return type str

CLI Example:

```
salt '*' macpackage.install_app /tmp/tmp.app /Applications/
```

`salt.modules.macpackage.`**`installed_pkgs`**`()`

Return the list of installed packages on the machine

Returns List of installed packages

Return type list

CLI Example:

```
salt '*' macpackage.installed_pkgs
```

`salt.modules.macpackage.`**`mount`**`(`*dmg*`)`

Attempt to mount a dmg file to a temporary location and return the location of the pkg file inside

Parameters **`dmg`** (*`str`*) – The location of the dmg file to mount

Returns

Tuple containing the results of the command along with the mount point

Return type tuple

CLI Example:

```
salt '*' macpackage.mount /tmp/software.dmg
```

`salt.modules.macpackage.`**`uninstall_app`**`(`*app*`)`

Uninstall an app file by removing it from the Applications directory

Parameters **`app`** (*`str`*) – The location of the .app file

Returns True if successful, otherwise False

Return type bool

CLI Example:

```
salt '*' macpackage.uninstall_app /Applications/app.app
```

`salt.modules.macpackage.`**`unmount`**`(`*mountpoint*`)`

Attempt to unmount a dmg file from a temporary location

Parameters **`mountpoint`** (*`str`*) – The location of the mount point

Returns The results of the hdutil detach command

Return type str

CLI Example:

```
salt '*' macpackage.unmount /dev/disk2
```

21.9.224 salt.modules.mac_pkgutil module

Installer support for macOS.

Installer is the native .pkg/.mpkg package manager for macOS.

`salt.modules.mac_pkgutil.`**`forget`**`(`*package_id*`)`

New in version 2016.3.0.

Remove the receipt data about the specified package. Does not remove files.

Warning: DO NOT use this command to fix broken package design

Parameters **`package_id`** (*`str`*) – The name of the package to forget

Returns True if successful, otherwise False

Return type bool

CLI Example:

```
salt '*' pkgutil.forget com.apple.pkg.gcc4.2Leo
```

`salt.modules.mac_pkgutil.`**`install`**`(`*source, package_id*`)`

Install a .pkg from an URI or an absolute path.

Parameters

- **`source`** (*`str`*) – The path to a package.
- **`package_id`** (*`str`*) – The package ID

Returns True if successful, otherwise False

Return type bool

CLI Example:

```
salt '*' pkgutil.install source=/vagrant/build_essentials.pkg package_id=com.
↪apple.pkg.gcc4.2Leo
```

`salt.modules.mac_pkgutil.`**`is_installed`**`(`*package_id*`)`

Returns whether a given package id is installed.

Returns True if installed, otherwise False

Return type bool

CLI Example:

```
salt '*' pkgutil.is_installed com.apple.pkg.gcc4.2Leo
```

`salt.modules.mac_pkgutil.`**`list_`**`()`

List the installed packages.

Returns A list of installed packages

Return type list

CLI Example:

```
salt '*' pkgutil.list
```

21.9.225 salt.modules.mac_portspkg

Support for MacPorts under macOS.

This module has some caveats.

1. Updating the database of available ports is quite resource-intensive. However, *refresh=True* is the default for all operations that need an up-to-date copy of available ports. Consider *refresh=False* when you are sure no db update is needed.

2. In some cases MacPorts doesn't always realize when another copy of itself is running and will gleefully tromp all over the available ports database. This makes MacPorts behave in undefined ways until a fresh complete copy is retrieved.

Because of 1 and 2 it is possible to get the salt-minion into a state where *salt mac-machine pkg./something/* won't want to return. Use

salt-run jobs.active

on the master to check for potentially long-running calls to *port*.

Finally, ports database updates are always handled with *port selfupdate* as opposed to *port sync*. This makes sense in the MacPorts user commmunity but may confuse experienced Linux admins as Linux package managers don't upgrade the packaging software when doing a package database update. In other words *salt mac-machine pkg.refresh_db* is more like *apt-get update; apt-get upgrade dpkg apt-get* than simply *apt-get update*.

`salt.modules.mac_portspkg.`**`available_version`**`(`**names*, ***kwargs*`)`

This function is an alias of `latest_version`.

Return the latest version of the named package available for upgrade or installation

Options:

refresh Update ports with `port selfupdate`

CLI Example:

```
salt '*' pkg.latest_version <package name>
salt '*' pkg.latest_version <package1> <package2> <package3>
```

`salt.modules.mac_portspkg.`**`install`**`(`*name=None*, *refresh=False*, *pkgs=None*, ***kwargs*`)`

Install the passed package(s) with `port install`

name The name of the formula to be installed. Note that this parameter is ignored if "pkgs" is passed.

CLI Example:

```
salt '*' pkg.install <package name>
```

version Specify a version to pkg to install. Ignored if pkgs is specified.

CLI Example:

```
salt '*' pkg.install <package name>
salt '*' pkg.install git-core version='1.8.5.5'
```

variant Specify a variant to pkg to install. Ignored if pkgs is specified.

CLI Example:

```
salt '*' pkg.install <package name>
salt '*' pkg.install git-core version='1.8.5.5' variant='+credential_
↪osxkeychain+doc+pcre'
```

Multiple Package Installation Options:

pkgs A list of formulas to install. Must be passed as a python list.

CLI Example:

```
salt '*' pkg.install pkgs='["foo","bar"]'
salt '*' pkg.install pkgs='["foo@1.2","bar"]'
salt '*' pkg.install pkgs='["foo@1.2+ssl","bar@2.3"]'
```

Returns a dict containing the new package names and versions:

```
{'<package>': {'old': '<old-version>',
               'new': '<new-version>'}}
```

CLI Example:

```
salt '*' pkg.install 'package package package'
```

`salt.modules.mac_portspkg.`**`latest_version`**(**names*, ***kwargs*)

Return the latest version of the named package available for upgrade or installation

Options:

refresh Update ports with `port selfupdate`

CLI Example:

```
salt '*' pkg.latest_version <package name>
salt '*' pkg.latest_version <package1> <package2> <package3>
```

`salt.modules.mac_portspkg.`**`list_pkgs`**(*versions_as_list=False*, ***kwargs*)

List the packages currently installed in a dict:

```
{'<package_name>': '<version>'}
```

CLI Example:

```
salt '*' pkg.list_pkgs
```

`salt.modules.mac_portspkg.`**`list_upgrades`**(*refresh=True*, ***kwargs*)

Check whether or not an upgrade is available for all packages

Options:

refresh Update ports with `port selfupdate`

CLI Example:

```
salt '*' pkg.list_upgrades
```

`salt.modules.mac_portspkg.`**`refresh_db`**`()`

Update ports with `port selfupdate`

CLI Example:

```
salt mac pkg.refresh_db
```

`salt.modules.mac_portspkg.`**`remove`**`(`*name=None*, *pkgs=None*, ***kwargs*`)`

Removes packages with `port uninstall`.

name The name of the package to be deleted.

Multiple Package Options:

pkgs A list of packages to delete. Must be passed as a python list. The `name` parameter will be ignored if this option is passed.

New in version 0.16.0.

Returns a dict containing the changes.

CLI Example:

```
salt '*' pkg.remove <package name>
salt '*' pkg.remove <package1>,<package2>,<package3>
salt '*' pkg.remove pkgs='["foo", "bar"]'
```

`salt.modules.mac_portspkg.`**`upgrade`**`(`*refresh=True*`)`

Run a full upgrade using MacPorts 'port upgrade outdated'

Options:

refresh Update ports with `port selfupdate`

Returns a dictionary containing the changes:

```
{'<package>':  {'old': '<old-version>',
                'new': '<new-version>'}}
```

CLI Example:

```
salt '*' pkg.upgrade
```

`salt.modules.mac_portspkg.`**`upgrade_available`**`(`*pkg*, *refresh=True*`)`

Check whether or not an upgrade is available for a given package

CLI Example:

```
salt '*' pkg.upgrade_available <package name>
```

`salt.modules.mac_portspkg.`**`version`**`(`**names*, ***kwargs*`)`

Returns a string representing the package version or an empty string if not installed. If more than one package name is specified, a dict of name/version pairs is returned.

CLI Example:

```
salt '*' pkg.version <package name>
salt '*' pkg.version <package1> <package2> <package3>
```

21.9.226 salt.modules.mac_power module

Module for editing power settings on macOS

New in version 2016.3.0.

`salt.modules.mac_power.`**`get_computer_sleep`**`()`

Display the amount of idle time until the computer sleeps.

Returns A string representing the sleep settings for the computer

Return type str

CLI Example:

..code-block:: bash

salt '*' power.get_computer_sleep

`salt.modules.mac_power.`**`get_display_sleep`**`()`

Display the amount of idle time until the display sleeps.

Returns A string representing the sleep settings for the displey

Return type str

CLI Example:

..code-block:: bash

salt '*' power.get_display_sleep

`salt.modules.mac_power.`**`get_harddisk_sleep`**`()`

Display the amount of idle time until the hard disk sleeps.

Returns A string representing the sleep settings for the hard disk

Return type str

CLI Example:

..code-block:: bash

salt '*' power.get_harddisk_sleep

`salt.modules.mac_power.`**`get_restart_freeze`**`()`

Displays whether 'restart on freeze' is on or off if supported

Returns A string value representing the "restart on freeze" settings

Return type string

CLI Example:

```
salt '*' power.get_restart_freeze
```

`salt.modules.mac_power.`**`get_restart_power_failure`**`()`

Displays whether 'restart on power failure' is on or off if supported

Returns A string value representing the "restart on power failure" settings

Return type string

CLI Example:

```
salt '*' power.get_restart_power_failure
```

`salt.modules.mac_power.`**`get_sleep`**`()`

Displays the amount of idle time until the machine sleeps. Settings for Computer, Display, and Hard Disk are displayed.

Returns A dictionary containing the sleep status for Computer, Display, and Hard Disk

Return type dict

CLI Example:

```
salt '*' power.get_sleep
```

`salt.modules.mac_power.`**`get_sleep_on_power_button`**`()`

Displays whether 'allow power button to sleep computer' is on or off if supported

Returns A string value representing the "allow power button to sleep computer" settings

Return type string

CLI Example:

```
salt '*' power.get_sleep_on_power_button
```

`salt.modules.mac_power.`**`get_wake_on_modem`**`()`

Displays whether 'wake on modem' is on or off if supported

Returns A string value representing the "wake on modem" settings

Return type str

CLI Example:

```
salt '*' power.get_wake_on_modem
```

`salt.modules.mac_power.`**`get_wake_on_network`**`()`

Displays whether 'wake on network' is on or off if supported

Returns A string value representing the "wake on network" settings

Return type string

CLI Example:

```
salt '*' power.get_wake_on_network
```

`salt.modules.mac_power.`**`set_computer_sleep`**`(`*minutes*`)`

Set the amount of idle time until the computer sleeps. Pass "Never" of "Off" to never sleep.

Parameters **`minutes`** – Can be an integer between 1 and 180 or "Never" or "Off"

Ptype int, str

Returns True if successful, False if not

Return type bool

CLI Example:

```
salt '*' power.set_computer_sleep 120
salt '*' power.set_computer_sleep off
```

`salt.modules.mac_power.`**`set_display_sleep`**`(`*minutes*`)`

Set the amount of idle time until the display sleeps. Pass "Never" of "Off" to never sleep.

Parameters **`minutes`** – Can be an integer between 1 and 180 or "Never" or "Off"

Ptype int, str

Returns True if successful, False if not

Return type bool

CLI Example:

```
salt '*' power.set_display_sleep 120
salt '*' power.set_display_sleep off
```

`salt.modules.mac_power.`**`set_harddisk_sleep`**`(`*minutes*`)`

Set the amount of idle time until the harddisk sleeps. Pass "Never" of "Off" to never sleep.

Parameters **`minutes`** – Can be an integer between 1 and 180 or "Never" or "Off"

Ptype int, str

Returns True if successful, False if not

Return type bool

CLI Example:

```
salt '*' power.set_harddisk_sleep 120
salt '*' power.set_harddisk_sleep off
```

`salt.modules.mac_power.`**`set_restart_freeze`**`(`*enabled*`)`

Specifies whether the server restarts automatically after a system freeze. This setting doesn't seem to be editable. The command completes successfully but the setting isn't actually updated. This is probably a macOS. The functions remains in case they ever fix the bug.

Parameters **enabled** (*bool*) – True to enable, False to disable. "On" and "Off" are also acceptable values. Additionally you can pass 1 and 0 to represent True and False respectively

Returns True if successful, False if not

Return type bool

CLI Example:

```
salt '*' power.set_restart_freeze True
```

salt.modules.mac_power.**set_restart_power_failure**(*enabled*)

Set whether or not the computer will automatically restart after a power failure.

Parameters **enabled** (*bool*) – True to enable, False to disable. "On" and "Off" are also acceptable values. Additionally you can pass 1 and 0 to represent True and False respectively

Returns True if successful, False if not

Return type bool

CLI Example:

```
salt '*' power.set_restart_power_failure True
```

salt.modules.mac_power.**set_sleep**(*minutes*)

Sets the amount of idle time until the machine sleeps. Sets the same value for Computer, Display, and Hard Disk. Pass "Never" or "Off" for computers that should never sleep.

Parameters **minutes** – Can be an integer between 1 and 180 or "Never" or "Off"

Ptype int, str

Returns True if successful, False if not

Return type bool

CLI Example:

```
salt '*' power.set_sleep 120
salt '*' power.set_sleep never
```

salt.modules.mac_power.**set_sleep_on_power_button**(*enabled*)

Set whether or not the power button can sleep the computer.

Parameters **enabled** (*bool*) – True to enable, False to disable. "On" and "Off" are also acceptable values. Additionally you can pass 1 and 0 to represent True and False respectively

Returns True if successful, False if not

Return type bool

CLI Example:

```
salt '*' power.set_sleep_on_power_button True
```

salt.modules.mac_power.**set_wake_on_modem**(*enabled*)

Set whether or not the computer will wake from sleep when modem activity is detected.

Parameters **enabled** (*bool*) – True to enable, False to disable. "On" and "Off" are also acceptable values. Additionally you can pass 1 and 0 to represent True and False respectively

Returns True if successful, False if not

Return type bool

CLI Example:

```
salt '*' power.set_wake_on_modem True
```

salt.modules.mac_power.**set_wake_on_network**(*enabled*)

Set whether or not the computer will wake from sleep when network activity is detected.

Parameters **enabled** (*bool*) – True to enable, False to disable. "On" and "Off" are also acceptable values. Additionally you can pass 1 and 0 to represent True and False respectively

Returns True if successful, False if not

Return type bool

CLI Example:

```
salt '*' power.set_wake_on_network True
```

21.9.227 salt.modules.mac_service module

The service module for macOS

New in version 2016.3.0.

This module has support for services in the following locations.

```
/System/Library/LaunchDaemons/
/System/Library/LaunchAgents/
/Library/LaunchDaemons/
/Library/LaunchAgents/

# As of version "2019.2.0" support for user-specific services were added.
/Users/foo/Library/LaunchAgents/
```

Note: As of the 2019.2.0 release, if a service is located in a `LaunchAgent` path and a `runas` user is NOT specified, the current console user will be used to properly interact with the service.

`salt.modules.mac_service.`**`available`**(*name*)

Check that the given service is available.

Parameters **`name`** (*`str`*) – The name of the service

Returns True if the service is available, otherwise False

Return type bool

CLI Example:

```
salt '*' service.available com.openssh.sshd
```

`salt.modules.mac_service.`**`disable`**(*name*, *runas=None*)

Disable a launchd service. Raises an error if the service fails to be disabled

Parameters

- **`name`** (*`str`*) – Service label, file name, or full path
- **`runas`** (*`str`*) – User to run launchctl commands

Returns `True` if successful or if the service is already disabled

Return type bool

CLI Example:

```
salt '*' service.disable org.cups.cupsd
```

`salt.modules.mac_service.`**`disabled`**(*name*, *runas=None*, *domain='system'*)

Check if the specified service is not enabled. This is the opposite of `service.enabled`

Parameters

- **`name`** (*`str`*) – The name to look up
- **`runas`** (*`str`*) – User to run launchctl commands
- **`domain`** (*`str`*) – domain to check for disabled services. Default is system.

Returns True if the specified service is NOT enabled, otherwise False

Return type bool

CLI Example:

```
salt '*' service.disabled org.cups.cupsd
```

`salt.modules.mac_service.`**`enable`**(*name*, *runas=None*)

Enable a launchd service. Raises an error if the service fails to be enabled

Parameters

- **name** (*str*) – Service label, file name, or full path
- **runas** (*str*) – User to run launchctl commands

Returns `True` if successful or if the service is already enabled

Return type bool

CLI Example:

```
salt '*' service.enable org.cups.cupsd
```

`salt.modules.mac_service.`**`enabled`**(*name*, *runas=None*)

Check if the specified service is enabled

Parameters

- **name** (*str*) – The name of the service to look up
- **runas** (*str*) – User to run launchctl commands

Returns True if the specified service enabled, otherwise False

Return type bool

CLI Example:

```
salt '*' service.enabled org.cups.cupsd
```

`salt.modules.mac_service.`**`get_all`**(*runas=None*)

Return a list of services that are enabled or available. Can be used to find the name of a service.

Parameters **runas** (*str*) – User to run launchctl commands

Returns A list of all the services available or enabled

Return type list

CLI Example:

```
salt '*' service.get_all
```

`salt.modules.mac_service.`**`get_enabled`**(*runas=None*)

Return a list of all services that are enabled. Can be used to find the name of a service.

Parameters **runas** (*str*) – User to run launchctl commands

Returns A list of all the services enabled on the system

Return type list

CLI Example:

```
salt '*' service.get_enabled
```

`salt.modules.mac_service.`**`launchctl`**(*sub_cmd*, **args*, ***kwargs*)

Run a launchctl command and raise an error if it fails

Parameters

- **sub_cmd** (*str*) – Sub command supplied to launchctl
- **args** (*tuple*) – Tuple containing additional arguments to pass to launchctl
- **kwargs** (*dict*) – Dictionary containing arguments to pass to `cmd.run_all`
- **return_stdout** (*bool*) – A keyword argument. If true return the stdout of the launchctl command

Returns `True` if successful, raise `CommandExecutionError` if not, or the stdout of the launchctl command if requested

Return type bool, str

CLI Example:

```
salt '*' service.launchctl debug org.cups.cupsd
```

`salt.modules.mac_service.`**`list_`**(*name=None*, *runas=None*)

Run launchctl list and return the output

Parameters

- **`name`** (*`str`*) – The name of the service to list
- **`runas`** (*`str`*) – User to run launchctl commands

Returns If a name is passed returns information about the named service, otherwise returns a list of all services and pids

Return type str

CLI Example:

```
salt '*' service.list
salt '*' service.list org.cups.cupsd
```

`salt.modules.mac_service.`**`missing`**(*name*)

The inverse of service.available Check that the given service is not available.

Parameters **`name`** (*`str`*) – The name of the service

Returns True if the service is not available, otherwise False

Return type bool

CLI Example:

```
salt '*' service.missing com.openssh.sshd
```

`salt.modules.mac_service.`**`restart`**(*name*, *runas=None*)

Unloads and reloads a launchd service. Raises an error if the service fails to reload

Parameters

- **`name`** (*`str`*) – Service label, file name, or full path
- **`runas`** (*`str`*) – User to run launchctl commands

Returns `True` if successful

Return type bool

CLI Example:

```
salt '*' service.restart org.cups.cupsd
```

`salt.modules.mac_service.`**`show`**(*name*)

Show properties of a launchctl service

Parameters **`name`** (*`str`*) – Service label, file name, or full path

Returns The service information if the service is found

Return type dict

CLI Example:

```
salt '*' service.show org.cups.cupsd  # service label
salt '*' service.show org.cups.cupsd.plist  # file name
salt '*' service.show /System/Library/LaunchDaemons/org.cups.cupsd.plist  # full
↪path
```

`salt.modules.mac_service.`**`start`**(*name*, *runas=None*)

Start a launchd service. Raises an error if the service fails to start

---

Note: To start a service in macOS the service must be enabled first. Use `service.enable` to enable the service.

Parameters

- **`name`** (*`str`*) – Service label, file name, or full path
- **`runas`** (*`str`*) – User to run launchctl commands

Returns `True` if successful or if the service is already running

Return type bool

CLI Example:

```
salt '*' service.start org.cups.cupsd
```

`salt.modules.mac_service.`**`status`**`(`*name*, *sig=None*, *runas=None*`)`

Return the status for a service.

Parameters

- **`name`** (*`str`*) – Used to find the service from launchctl. Can be any part of the service name or a regex expression.
- **`sig`** (*`str`*) – Find the service with status.pid instead. Note that `name` must still be provided.
- **`runas`** (*`str`*) – User to run launchctl commands

Returns The PID for the service if it is running, or 'loaded' if the service should not always have a PID, or otherwise an empty string

Return type str

CLI Example:

```
salt '*' service.status cups
```

`salt.modules.mac_service.`**`stop`**`(`*name*, *runas=None*`)`

Stop a launchd service. Raises an error if the service fails to stop

---

Note: Though `service.stop` will unload a service in macOS, the service will start on next boot unless it is disabled. Use `service.disable` to disable the service

---

Parameters

- **`name`** (*`str`*) – Service label, file name, or full path
- **`runas`** (*`str`*) – User to run launchctl commands

Returns `True` if successful or if the service is already stopped

Return type bool

CLI Example:

```
salt '*' service.stop org.cups.cupsd
```

21.9.228 salt.modules.mac_shadow module

Manage macOS local directory passwords and policies

New in version 2016.3.0.

Note that it is usually better to apply password policies through the creation of a configuration profile.

`salt.modules.mac_shadow.`**`del_password`**`(`*name*`)`

Deletes the account password

Parameters **`name`** (*str*) – The user name of the account

Returns True if successful, otherwise False

Return type bool

Raises CommandExecutionError on user not found or any other unknown error

CLI Example:

```
salt '*' shadow.del_password username
```

`salt.modules.mac_shadow.`**`get_account_created`**`(`*name*`)`

Get the date/time the account was created

Parameters **`name`** (*str*) – The username of the account

Returns The date/time the account was created (yyyy-mm-dd hh:mm:ss)

Return type str

Raises CommandExecutionError on user not found or any other unknown error

CLI Example:

```
salt '*' shadow.get_account_created admin
```

`salt.modules.mac_shadow.`**`get_change`**`(`*name*`)`

Gets the date on which the password expires

Parameters **`name`** (*str*) – The name of the user account

Returns The date the password will expire

Return type str

Raises CommandExecutionError on user not found or any other unknown error

CLI Example:

```
salt '*' shadow.get_change username
```

`salt.modules.mac_shadow.`**`get_expire`**`(`*name*`)`

Gets the date on which the account expires

Parameters **`name`** (*str*) – The name of the user account

Returns The date the account expires

Return type str

Raises CommandExecutionError on user not found or any other unknown error

CLI Example:

```
salt '*' shadow.get_expire username
```

`salt.modules.mac_shadow.`**`get_last_change`**`(`*name*`)`

Get the date/time the account was changed

Parameters **`name`** (*str*) – The username of the account

Returns The date/time the account was modified (yyyy-mm-dd hh:mm:ss)

Return type str

Raises CommandExecutionError on user not found or any other unknown error

CLI Example:

```
salt '*' shadow.get_last_change admin
```

`salt.modules.mac_shadow.`**`get_login_failed_count`**`(`*name*`)`

Get the the number of failed login attempts

Parameters **`name`** (*str*) – The username of the account

Returns The number of failed login attempts

Return type int

Raises CommandExecutionError on user not found or any other unknown error

CLI Example:

```
salt '*' shadow.get_login_failed_count admin
```

`salt.modules.mac_shadow.`**`get_login_failed_last`**`(`*name*`)`

Get the date/time of the last failed login attempt

Parameters **name** (*str*) – The username of the account

Returns The date/time of the last failed login attempt on this account (yyyy-mm-dd hh:mm:ss)

Return type str

Raises CommandExecutionError on user not found or any other unknown error

CLI Example:

```
salt '*' shadow.get_login_failed_last admin
```

`salt.modules.mac_shadow.`**`get_maxdays`**`(`*name*`)`

Get the maximum age of the password

Parameters **name** (*str*) – The username of the account

Returns The maximum age of the password in days

Return type int

Raises CommandExecutionError on user not found or any other unknown error

CLI Example:

```
salt '*' shadow.get_maxdays admin 90
```

`salt.modules.mac_shadow.`**`info`**`(`*name*`)`

Return information for the specified user

Parameters **name** (*str*) – The username

Returns A dictionary containing the user's shadow information

Return type dict

CLI Example:

```
salt '*' shadow.info admin
```

`salt.modules.mac_shadow.`**`set_change`**`(`*name, date*`)`

Sets the date on which the password expires. The user will be required to change their password. Format is mm/dd/yyyy

Parameters

- **name** (*str*) – The name of the user account
- **date** (*date*) – The date the password will expire. Must be in mm/dd/yyyy format.

Returns True if successful, otherwise False

Return type bool

Raises CommandExecutionError on user not found or any other unknown error

CLI Example:

```
salt '*' shadow.set_change username 09/21/2016
```

`salt.modules.mac_shadow.`**`set_expire`**`(`*name, date*`)`

Sets the date on which the account expires. The user will not be able to login after this date. Date format is mm/dd/yyyy

Parameters

- **name** (*str*) – The name of the user account

- **date** (*datetime*) – The date the account will expire. Format must be mm/dd/yyyy.

Returns True if successful, False if not

Return type bool

Raises CommandExecutionError on user not found or any other unknown error

CLI Example:

```
salt '*' shadow.set_expire username 07/23/2015
```

salt.modules.mac_shadow.**set_inactdays**(*name*, *days*)

Set the number if inactive days before the account is locked. Not available in macOS

Parameters

- **name** (*str*) – The user name
- **days** (*int*) – The number of days

Returns Will always return False until macOS supports this feature.

Return type bool

CLI Example:

```
salt '*' shadow.set_inactdays admin 90
```

salt.modules.mac_shadow.**set_maxdays**(*name*, *days*)

Set the maximum age of the password in days

Parameters

- **name** (*str*) – The username of the account
- **days** (*int*) – The maximum age of the account in days

Returns True if successful, False if not

Return type bool

Raises CommandExecutionError on user not found or any other unknown error

CLI Example:

```
salt '*' shadow.set_maxdays admin 90
```

salt.modules.mac_shadow.**set_mindays**(*name*, *days*)

Set the minimum password age in days. Not available in macOS.

Parameters

- **name** (*str*) – The user name
- **days** (*int*) – The number of days

Returns Will always return False until macOS supports this feature.

Return type bool

CLI Example:

```
salt '*' shadow.set_mindays admin 90
```

salt.modules.mac_shadow.**set_password**(*name*, *password*)

Set the password for a named user (insecure, the password will be in the process list while the command is running)

Parameters

- **name** (*str*) – The name of the local user, which is assumed to be in the local directory service
- **password** (*str*) – The plaintext password to set

Returns True if successful, otherwise False

Return type bool

Raises CommandExecutionError on user not found or any other unknown error

CLI Example:

```
salt '*' mac_shadow.set_password macuser macpassword
```

`salt.modules.mac_shadow.`**`set_warndays`**`(`*name*, *days*`)`

Set the number of days before the password expires that the user will start to see a warning. Not available in macOS

Parameters

- **name** (*str*) – The user name
- **days** (*int*) – The number of days

Returns Will always return False until macOS supports this feature.

Return type bool

CLI Example:

```
salt '*' shadow.set_warndays admin 90
```

21.9.229 salt.modules.mac_softwareupdate module

Support for the softwareupdate command on MacOS.

`salt.modules.mac_softwareupdate.`**`download`**`(`*name*`)`

Download a named update so that it can be installed later with the `update` or `update_all` functions

Parameters **name** (*str*) – The update to download.

Returns True if successful, otherwise False

Return type bool

CLI Example:

```
salt '*' softwareupdate.download <update name>
```

`salt.modules.mac_softwareupdate.`**`download_all`**`(`*recommended=False*, *restart=True*`)`

Download all available updates so that they can be installed later with the `update` or `update_all` functions. It returns a list of updates that are now downloaded.

Parameters

- **recommended** (*bool*) – If set to True, only install the recommended updates. If set to False (default) all updates are installed.
- **restart** (*bool*) – Set this to False if you do not want to install updates that require a restart. Default is True

Returns A list containing all downloaded updates on the system.

Return type list

CLI Example:

```
salt '*' softwareupdate.download_all
```

`salt.modules.mac_softwareupdate.`**`get_catalog`**`()`

New in version 2016.3.0.

Get the current catalog being used for update lookups. Will return a url if a custom catalog has been specified. Otherwise the word 'Default' will be returned

Returns The catalog being used for update lookups

Return type str

CLI Example:

```
salt '*' softwareupdates.get_catalog
```

`salt.modules.mac_softwareupdate.`**`ignore`**`(`*name*`)`

Ignore a specific program update. When an update is ignored the '-' and version number at the end will be omitted, so "SecUpd2014-001-1.0" becomes "SecUpd2014-001". It will be removed automatically if present. An update is successfully ignored when it no longer shows up after list_updates.

Parameters **`name`** – The name of the update to add to the ignore list.

Ptype str

Returns True if successful, False if not

Return type bool

CLI Example:

```
salt '*' softwareupdate.ignore <update-name>
```

`salt.modules.mac_softwareupdate.`**`list_available`**`(`*recommended=False, restart=False*`)`

List all available updates.

Parameters

- **`recommended`** (*`bool`*) – Show only recommended updates.
- **`restart`** (*`bool`*) – Show only updates that require a restart.

Returns Returns a dictionary containing the updates

Return type dict

CLI Example:

```
salt '*' softwareupdate.list_available
```

`salt.modules.mac_softwareupdate.`**`list_downloads`**`()`

Return a list of all updates that have been downloaded locally.

Returns A list of updates that have been downloaded

Return type list

CLI Example:

```
salt '*' softwareupdate.list_downloads
```

`salt.modules.mac_softwareupdate.`**`list_ignored`**`()`

List all updates that have been ignored. Ignored updates are shown without the '-' and version number at the end, this is how the softwareupdate command works.

Returns The list of ignored updates

Return type list

CLI Example:

```
salt '*' softwareupdate.list_ignored
```

`salt.modules.mac_softwareupdate.`**`reset_catalog`**`()`

New in version 2016.3.0.

Reset the Software Update Catalog to the default.

Returns True if successful, False if not

Return type bool

CLI Example:

```
salt '*' softwareupdates.reset_catalog
```

`salt.modules.mac_softwareupdate.`**`reset_ignored`**`()`

Make sure the ignored updates are not ignored anymore, returns a list of the updates that are no longer ignored.

Returns True if the list was reset, Otherwise False

Return type bool

CLI Example:

```
salt '*' softwareupdate.reset_ignored
```

salt.modules.mac_softwareupdate.**schedule_enable**(*enable*)

Enable/disable automatic update scheduling.

Parameters **enable** – True/On/Yes/1 to turn on automatic updates. False/No/Off/0 to turn off automatic updates. If this value is empty, the current status will be returned.

Type bool str

Returns True if scheduling is enabled, False if disabled

Return type bool

CLI Example:

```
salt '*' softwareupdate.schedule_enable on|off
```

salt.modules.mac_softwareupdate.**schedule_enabled**()

Check the status of automatic update scheduling.

Returns True if scheduling is enabled, False if disabled

Return type bool

CLI Example:

```
salt '*' softwareupdate.schedule_enabled
```

salt.modules.mac_softwareupdate.**set_catalog**(*url*)

New in version 2016.3.0.

Set the Software Update Catalog to the URL specified

Parameters **url** (*str*) – The url to the update catalog

Returns True if successful, False if not

Return type bool

CLI Example:

```
salt '*' softwareupdates.set_catalog http://swupd.local:8888/index.sucatalog
```

salt.modules.mac_softwareupdate.**update**(*name*)

Install a named update.

Parameters **name** (*str*) – The name of the of the update to install.

Returns True if successfully updated, otherwise False

Return type bool

CLI Example:

```
salt '*' softwareupdate.update <update-name>
```

salt.modules.mac_softwareupdate.**update_all**(*recommended=False*, *restart=True*)

Install all available updates. Returns a dictionary containing the name of the update and the status of its installation.

Parameters

- **recommended** (*bool*) – If set to True, only install the recommended updates. If set to False (default) all updates are installed.
- **restart** (*bool*) – Set this to False if you do not want to install updates that require a restart. Default is True

Returns A dictionary containing the updates that were installed and the status of its installation. If no updates were installed an empty dictionary is returned.

Return type dict

CLI Example:

```
salt '*' softwareupdate.update_all
```

`salt.modules.mac_softwareupdate.`**`update_available`**`(`*name*`)`

Check whether or not an update is available with a given name.

Parameters **`name`** (*`str`*) – The name of the update to look for

Returns True if available, False if not

Return type bool

CLI Example:

```
salt '*' softwareupdate.update_available <update-name>
salt '*' softwareupdate.update_available "<update with whitespace>"
```

21.9.230 salt.modules.mac_sysctl module

Module for viewing and modifying sysctl parameters

`salt.modules.mac_sysctl.`**`assign`**`(`*name*, *value*`)`

Assign a single sysctl parameter for this minion

name The name of the sysctl value to edit.

value The sysctl value to apply.

CLI Example:

```
salt '*' sysctl.assign net.inet.icmp.icmplim 50
```

`salt.modules.mac_sysctl.`**`get`**`(`*name*`)`

Return a single sysctl parameter for this minion

name The name of the sysctl value to display.

CLI Example:

```
salt '*' sysctl.get hw.physmem
```

`salt.modules.mac_sysctl.`**`persist`**`(`*name*, *value*, *config='/etc/sysctl.conf'*, *apply_change=False*`)`

Assign and persist a simple sysctl parameter for this minion

name The name of the sysctl value to edit.

value The sysctl value to apply.

config The location of the sysctl configuration file.

apply_change Default is False; Default behavior only creates or edits the sysctl.conf file. If apply is set to True, the changes are applied to the system.

CLI Example:

```
salt '*' sysctl.persist net.inet.icmp.icmplim 50
salt '*' sysctl.persist coretemp_load NO config=/etc/sysctl.conf
```

`salt.modules.mac_sysctl.`**`show`**`(`*config_file=False*`)`

Return a list of sysctl parameters for this minion

CLI Example:

```
salt '*' sysctl.show
```

21.9.231 salt.modules.mac_system module

System module for sleeping, restarting, and shutting down the system on Mac OS X

New in version 2016.3.0.

> **Warning:** Using this module will enable `atrun` on the system if it is disabled.

salt.modules.mac_system.**get_boot_arch**()

Get the kernel architecture setting from `com.apple.Boot.plist`

Returns A string value representing the boot architecture setting

Return type str

CLI Example:

```
salt '*' system.get_boot_arch
```

salt.modules.mac_system.**get_computer_name**()

Gets the computer name

Returns The computer name

Return type str

CLI Example:

```
salt '*' system.get_computer_name
```

salt.modules.mac_system.**get_disable_keyboard_on_lock**()

Get whether or not the keyboard should be disabled when the X Serve enclosure lock is engaged.

Returns True if disable keyboard on lock is on, False if off

Return type bool

CLI Example:

..code-block:: bash

salt '*' system.get_disable_keyboard_on_lock

salt.modules.mac_system.**get_remote_events**()

Displays whether remote apple events are on or off.

Returns True if remote apple events are on, False if off

Return type bool

CLI Example:

```
salt '*' system.get_remote_events
```

salt.modules.mac_system.**get_remote_login**()

Displays whether remote login (SSH) is on or off.

Returns True if remote login is on, False if off

Return type bool

CLI Example:

```
salt '*' system.get_remote_login
```

salt.modules.mac_system.**get_restart_delay**()

Get the number of seconds after which the computer will start up after a power failure.

Returns A string value representing the number of seconds the system will delay restart after power loss

Return type str

CLI Example:

```
salt '*' system.get_restart_delay
```

`salt.modules.mac_system.`**`get_startup_disk`**`()`

Displays the current startup disk

Returns The current startup disk

Return type str

CLI Example:

```
salt '*' system.get_startup_disk
```

`salt.modules.mac_system.`**`get_subnet_name`**`()`

Gets the local subnet name

Returns The local subnet name

Return type str

CLI Example:

```
salt '*' system.get_subnet_name
```

`salt.modules.mac_system.`**`halt`**`(`*at_time=None*`)`

Halt a running system

Parameters **at_time** (*str*) – Any valid *at* expression. For example, some valid at expressions could be:

- noon
- midnight
- fri
- 9:00 AM
- 2:30 PM tomorrow
- now + 10 minutes

Note: If you pass a time only, with no 'AM/PM' designation, you have to double quote the parameter on the command line. For example: '"14:00"'

CLI Example:

```
salt '*' system.halt
salt '*' system.halt 'now + 10 minutes'
```

`salt.modules.mac_system.`**`list_startup_disks`**`()`

List all valid startup disks on the system.

Returns A list of valid startup disks

Return type list

CLI Example:

```
salt '*' system.list_startup_disks
```

`salt.modules.mac_system.`**`restart`**`(`*at_time=None*`)`

Restart the system

Parameters **at_time** (*str*) – Any valid *at* expression. For example, some valid at expressions could be:

- noon

  - midnight
  - fri
  - 9:00 AM
  - 2:30 PM tomorrow
  - now + 10 minutes

Note: If you pass a time only, with no 'AM/PM' designation, you have to double quote the parameter on the command line. For example: '"14:00"'

CLI Example:

```
salt '*' system.restart
salt '*' system.restart '12:00 PM fri'
```

`salt.modules.mac_system.`**`set_boot_arch`**(*arch='default'*)

Set the kernel to boot in 32 or 64 bit mode on next boot.

Note: When this function fails with the error `changes to kernel architecture failed to save!`, then the boot arch is not updated. This is either an Apple bug, not available on the test system, or a result of system files being locked down in macOS (SIP Protection).

Parameters **`arch`** (*`str`*) – A string representing the desired architecture. If no value is passed, default is assumed. Valid values include:

  - i386
  - x86_64
  - default

Returns True if successful, False if not

Return type bool

CLI Example:

```
salt '*' system.set_boot_arch i386
```

`salt.modules.mac_system.`**`set_computer_name`**(*name*)

Set the computer name

Parameters **`name`** (*`str`*) – The new computer name

Returns True if successful, False if not

Return type bool

CLI Example:

```
salt '*' system.set_computer_name "Mike's Mac"
```

`salt.modules.mac_system.`**`set_disable_keyboard_on_lock`**(*enable*)

Get whether or not the keyboard should be disabled when the X Serve enclosure lock is engaged.

Parameters **`enable`** (*`bool`*) – True to enable, False to disable. "On" and "Off" are also acceptable values. Additionally you can pass 1 and 0 to represent True and False respectively

Returns True if successful, False if not

Return type bool

CLI Example:

```
salt '*' system.set_disable_keyboard_on_lock False
```

`salt.modules.mac_system.`**`set_remote_events`**`(`*enable*`)`

Set whether the server responds to events sent by other computers (such as AppleScripts)

Parameters **enable** (*bool*) – True to enable, False to disable. "On" and "Off" are also acceptable values. Additionally you can pass 1 and 0 to represent True and False respectively

Returns True if successful, False if not

Return type bool

CLI Example:

```
salt '*' system.set_remote_events On
```

`salt.modules.mac_system.`**`set_remote_login`**`(`*enable*`)`

Set the remote login (SSH) to either on or off.

Parameters **enable** (*bool*) – True to enable, False to disable. "On" and "Off" are also acceptable values. Additionally you can pass 1 and 0 to represent True and False respectively

Returns True if successful, False if not

Return type bool

CLI Example:

```
salt '*' system.set_remote_login True
```

`salt.modules.mac_system.`**`set_restart_delay`**`(`*seconds*`)`

Set the number of seconds after which the computer will start up after a power failure.

Warning: This command fails with the following error:

```
Error, IOServiceOpen returned 0x10000003
```

The setting is not updated. This is an apple bug. It seems like it may only work on certain versions of Mac Server X. This article explains the issue in more detail, though it is quite old.

http://lists.apple.com/archives/macos-x-server/2006/Jul/msg00967.html

Parameters **seconds** (*int*) – The number of seconds. Must be a multiple of 30

Returns True if successful, False if not

Return type bool

CLI Example:

```
salt '*' system.set_restart_delay 180
```

`salt.modules.mac_system.`**`set_startup_disk`**`(`*path*`)`

Set the current startup disk to the indicated path. Use `system.list_startup_disks` to find valid startup disks on the system.

Parameters **path** (*str*) – The valid startup disk path

Returns True if successful, False if not

Return type bool

CLI Example:

```
salt '*' system.set_startup_disk /System/Library/CoreServices
```

`salt.modules.mac_system.`**`set_subnet_name`**`(`*name*`)`

Set the local subnet name

Parameters **name** (*str*) – The new local subnet name

Note: Spaces are changed to dashes. Other special characters are removed.

Returns True if successful, False if not

Return type bool

CLI Example:

```
The following will be set as 'Mikes-Mac'
salt '*' system.set_subnet_name "Mike's Mac"
```

`salt.modules.mac_system.`**`shutdown`**`(`*at_time=None*`)`

Shutdown the system

Parameters **`at_time`** (*str*) – Any valid *at* expression. For example, some valid at expressions could be:

- noon
- midnight
- fri
- 9:00 AM
- 2:30 PM tomorrow
- now + 10 minutes

Note: If you pass a time only, with no 'AM/PM' designation, you have to double quote the parameter on the command line. For example: '"14:00"'

CLI Example:

```
salt '*' system.shutdown
salt '*' system.shutdown 'now + 1 hour'
```

`salt.modules.mac_system.`**`sleep`**`(`*at_time=None*`)`

Sleep the system. If a user is active on the system it will likely fail to sleep.

Parameters **`at_time`** (*str*) – Any valid *at* expression. For example, some valid at expressions could be:

- noon
- midnight
- fri
- 9:00 AM
- 2:30 PM tomorrow
- now + 10 minutes

Note: If you pass a time only, with no 'AM/PM' designation, you have to double quote the parameter on the command line. For example: '"14:00"'

CLI Example:

```
salt '*' system.sleep
salt '*' system.sleep '10:00 PM'
```

21.9.232 salt.modules.mac_timezone module

Module for editing date/time settings on macOS

New in version 2016.3.0.

`salt.modules.mac_timezone.`**`get_date`**`()`

Displays the current date

Returns the system date

Return type str

CLI Example:

```
salt '*' timezone.get_date
```

`salt.modules.mac_timezone.`**`get_hwclock`**`()`

Get current hardware clock setting (UTC or localtime)

CLI Example:

```
salt '*' timezone.get_hwclock
```

`salt.modules.mac_timezone.`**`get_offset`**`()`

Displays the current time zone offset

Returns The current time zone offset

Return type str

CLI Example:

```
salt '*' timezone.get_offset
```

`salt.modules.mac_timezone.`**`get_time`**`()`

Get the current system time.

Returns The current time in 24 hour format

Return type str

CLI Example:

```
salt '*' timezone.get_time
```

`salt.modules.mac_timezone.`**`get_time_server`**`()`

Display the currently set network time server.

Returns the network time server

Return type str

CLI Example:

```
salt '*' timezone.get_time_server
```

`salt.modules.mac_timezone.`**`get_using_network_time`**`()`

Display whether network time is on or off

Returns True if network time is on, False if off

Return type bool

CLI Example:

```
salt '*' timezone.get_using_network_time
```

`salt.modules.mac_timezone.`**`get_zone`**`()`

Displays the current time zone

Returns The current time zone

Return type str

CLI Example:

```
salt '*' timezone.get_zone
```

`salt.modules.mac_timezone.`**`get_zonecode`**`()`

Displays the current time zone abbreviated code

Returns The current time zone code

Return type str

CLI Example:

```
salt '*' timezone.get_zonecode
```

`salt.modules.mac_timezone.`**`list_zones`**`()`

Displays a list of available time zones. Use this list when setting a time zone using `timezone.set_zone`

Returns a list of time zones

Return type list

CLI Example:

```
salt '*' timezone.list_zones
```

`salt.modules.mac_timezone.`**`set_date`**`(`*date*`)`

Set the current month, day, and year

Parameters **`date`** (*`str`*) – The date to set. Valid date formats are:

- %m:%d:%y
- %m:%d:%Y
- %m/%d/%y
- %m/%d/%Y

Returns True if successful, False if not

Return type bool

Raises SaltInvocationError on Invalid Date format

Raises CommandExecutionError on failure

CLI Example:

```
salt '*' timezone.set_date 1/13/2016
```

`salt.modules.mac_timezone.`**`set_hwclock`**`(`*clock*`)`

Sets the hardware clock to be either UTC or localtime

CLI Example:

```
salt '*' timezone.set_hwclock UTC
```

`salt.modules.mac_timezone.`**`set_time`**`(`*time*`)`

Sets the current time. Must be in 24 hour format.

Parameters **`time`** (*`str`*) – The time to set in 24 hour format. The value must be double quoted. ie: '"17:46"'

Returns True if successful, False if not

Return type bool

Raises SaltInvocationError on Invalid Time format

Raises CommandExecutionError on failure

CLI Example:

```
salt '*' timezone.set_time '"17:34"'
```

`salt.modules.mac_timezone.`**`set_time_server`**(*time_server='time.apple.com'*)

Designates a network time server. Enter the IP address or DNS name for the network time server.

Parameters **`time_server`** – IP or DNS name of the network time server. If nothing is passed the time server will be set to the macOS default of 'time.apple.com'

Type str

Returns True if successful, False if not

Return type bool

Raises CommandExecutionError on failure

CLI Example:

```
salt '*' timezone.set_time_server time.acme.com
```

`salt.modules.mac_timezone.`**`set_using_network_time`**(*enable*)

Set whether network time is on or off.

Parameters **`enable`** – True to enable, False to disable. Can also use 'on' or 'off'

Type str bool

Returns True if successful, False if not

Return type bool

Raises CommandExecutionError on failure

CLI Example:

```
salt '*' timezone.set_using_network_time True
```

`salt.modules.mac_timezone.`**`set_zone`**(*time_zone*)

Set the local time zone. Use `timezone.list_zones` to list valid time_zone arguments

Parameters **`time_zone`** (*`str`*) – The time zone to apply

Returns True if successful, False if not

Return type bool

Raises SaltInvocationError on Invalid Timezone

Raises CommandExecutionError on failure

CLI Example:

```
salt '*' timezone.set_zone America/Denver
```

`salt.modules.mac_timezone.`**`zone_compare`**(*time_zone*)

Compares the given timezone name with the system timezone name.

Returns True if they are the same, False if not

Return type bool

CLI Example:

```
salt '*' timezone.zone_compare America/Boise
```

21.9.233 salt.modules.mac_user

Manage users on Mac OS 10.7+

Important: If you feel that Salt should be using this module to manage users on a minion, and it is using a different module (or gives an error similar to *'user.info' is not available*), see *here*.

`salt.modules.mac_user.`**`add`**(*name, uid=None, gid=None, groups=None, home=None, shell=None, fullname=None, createhome=True, **kwargs*)

Add a user to the minion

CLI Example:

```
salt '*' user.add name <uid> <gid> <groups> <home> <shell>
```

`salt.modules.mac_user.`**`chfullname`**(*name*, *fullname*)

Change the user's Full Name

CLI Example:

```
salt '*' user.chfullname foo 'Foo Bar'
```

`salt.modules.mac_user.`**`chgid`**(*name*, *gid*)

Change the default group of the user

CLI Example:

```
salt '*' user.chgid foo 4376
```

`salt.modules.mac_user.`**`chgroups`**(*name*, *groups*, *append=False*)

Change the groups to which the user belongs. Note that the user's primary group does not have to be one of the groups passed, membership in the user's primary group is automatically assumed.

groups Groups to which the user should belong, can be passed either as a python list or a comma-separated string

append Instead of removing user from groups not included in the `groups` parameter, just add user to any groups for which they are not members

CLI Example:

```
salt '*' user.chgroups foo wheel,root
```

`salt.modules.mac_user.`**`chhome`**(*name*, *home*, ***kwargs*)

Change the home directory of the user

CLI Example:

```
salt '*' user.chhome foo /Users/foo
```

`salt.modules.mac_user.`**`chshell`**(*name*, *shell*)

Change the default shell of the user

CLI Example:

```
salt '*' user.chshell foo /bin/zsh
```

`salt.modules.mac_user.`**`chuid`**(*name*, *uid*)

Change the uid for a named user

CLI Example:

```
salt '*' user.chuid foo 4376
```

`salt.modules.mac_user.`**`delete`**(*name*, *remove=False*, *force=False*)

Remove a user from the minion

CLI Example:

```
salt '*' user.delete name remove=True force=True
```

`salt.modules.mac_user.`**`disable_auto_login`**()

New in version 2016.3.0.

Disables auto login on the machine

Returns `True` if successful, otherwise `False`

Return type bool

CLI Example:

```
salt '*' user.disable_auto_login
```

`salt.modules.mac_user.`**`enable_auto_login`**`(`*name, password*`)`

New in version 2016.3.0.

Configures the machine to auto login with the specified user

Parameters

- **name** (*str*) – The user account use for auto login
- **password** (*str*) – The password to user for auto login

  New in version 2017.7.3.

Returns `True` if successful, otherwise `False`

Return type bool

CLI Example:

```
salt '*' user.enable_auto_login stevej
```

`salt.modules.mac_user.`**`get_auto_login`**`()`

New in version 2016.3.0.

Gets the current setting for Auto Login

Returns If enabled, returns the user name, otherwise returns False

Return type str, bool

CLI Example:

```
salt '*' user.get_auto_login
```

`salt.modules.mac_user.`**`getent`**`(`*refresh=False*`)`

Return the list of all info for all users

CLI Example:

```
salt '*' user.getent
```

`salt.modules.mac_user.`**`info`**`(`*name*`)`

Return user information

CLI Example:

```
salt '*' user.info root
```

`salt.modules.mac_user.`**`list_groups`**`(`*name*`)`

Return a list of groups the named user belongs to.

name

The name of the user for which to list groups. Starting in Salt 2016.11.0, all groups for the user, including groups beginning with an underscore will be listed.

Changed in version 2016.11.0.

CLI Example:

```
salt '*' user.list_groups foo
```

`salt.modules.mac_user.`**`list_users`**`()`

Return a list of all users

CLI Example:

```
salt '*' user.list_users
```

`salt.modules.mac_user.`**`primary_group`**`(`*name*`)`

Return the primary group of the named user

New in version 2016.3.0.

CLI Example:

```
salt '*' user.primary_group saltadmin
```

`salt.modules.mac_user.`**`rename`**`(`*name*, *new_name*`)`

Change the username for a named user

CLI Example:

```
salt '*' user.rename name new_name
```

21.9.234 salt.modules.mac_xattr module

This module allows you to manage extended attributes on files or directories

```
salt '*' xattr.list /path/to/file
```

`salt.modules.mac_xattr.`**`clear`**`(`*path*`)`

Causes the all attributes on the file/directory to be removed

Parameters **`path`** (*`str`*) – The file(s) to get attributes from

Returns True if successful, otherwise False

Raises CommandExecutionError on file not found or any other unknown error

CLI Example:

```
salt '*' xattr.delete /path/to/file "com.test.attr"
```

`salt.modules.mac_xattr.`**`delete`**`(`*path*, *attribute*`)`

Removes the given attribute from the file

Parameters

- **`path`** (*`str`*) – The file(s) to get attributes from
- **`attribute`** (*`str`*) – The attribute name to be deleted from the file/directory

Returns True if successful, otherwise False

Return type bool

Raises CommandExecutionError on file not found, attribute not found, and any other unknown error

CLI Example:

```
salt '*' xattr.delete /path/to/file "com.test.attr"
```

`salt.modules.mac_xattr.`**`list_`**`(`*path*, ***kwargs*`)`

List all of the extended attributes on the given file/directory

Parameters

- **`path`** (*`str`*) – The file(s) to get attributes from

- **hex** (*bool*) – Return the values with forced hexadecimal values

Returns A dictionary containing extended attributes and values for the given file

Return type dict

Raises CommandExecutionError on file not found or any other unknown error

CLI Example:

```
salt '*' xattr.list /path/to/file
salt '*' xattr.list /path/to/file hex=True
```

`salt.modules.mac_xattr.`**`read`**`(`*path, attribute, **kwargs*`)`

Read the given attributes on the given file/directory

Parameters

- **path** (*str*) – The file to get attributes from
- **attribute** (*str*) – The attribute to read
- **hex** (*bool*) – Return the values with forced hexadecimal values

Returns A string containing the value of the named attribute

Return type str

Raises CommandExecutionError on file not found, attribute not found, and any other unknown error

CLI Example:

```
salt '*' xattr.read /path/to/file com.test.attr
salt '*' xattr.read /path/to/file com.test.attr hex=True
```

`salt.modules.mac_xattr.`**`write`**`(`*path, attribute, value, **kwargs*`)`

Causes the given attribute name to be assigned the given value

Parameters

- **path** (*str*) – The file(s) to get attributes from
- **attribute** (*str*) – The attribute name to be written to the file/directory
- **value** (*str*) – The value to assign to the given attribute
- **hex** (*bool*) – Set the values with forced hexadecimal values

Returns True if successful, otherwise False

Return type bool

Raises CommandExecutionError on file not found or any other unknown error

CLI Example:

```
salt '*' xattr.write /path/to/file "com.test.attr" "value"
```

21.9.235 salt.modules.makeconf

Support for modifying make.conf under Gentoo

`salt.modules.makeconf.`**`append_cflags`**`(`*value*`)`

Add to or create a new CFLAGS in the make.conf

Return a dict containing the new value for variable:

```
{'<variable>': {'old': '<old-value>',
                'new': '<new-value>'}}
```

CLI Example:

```
salt '*' makeconf.append_cflags '-pipe'
```

`salt.modules.makeconf.`**`append_cxxflags`**`(`*value*`)`

Add to or create a new CXXFLAGS in the make.conf

Return a dict containing the new value for variable:

```
{'<variable>': {'old': '<old-value>',
                'new': '<new-value>'}}
```

CLI Example:

```
salt '*' makeconf.append_cxxflags '-pipe'
```

`salt.modules.makeconf.`**`append_emerge_default_opts`**`(`*value*`)`

Add to or create a new EMERGE_DEFAULT_OPTS in the make.conf

Return a dict containing the new value for variable:

```
{'<variable>': {'old': '<old-value>',
                'new': '<new-value>'}}
```

CLI Example:

```
salt '*' makeconf.append_emerge_default_opts '--jobs'
```

`salt.modules.makeconf.`**`append_features`**`(`*value*`)`

Add to or create a new FEATURES in the make.conf

Return a dict containing the new value for variable:

```
{'<variable>': {'old': '<old-value>',
                'new': '<new-value>'}}
```

CLI Example:

```
salt '*' makeconf.append_features 'webrsync-gpg'
```

`salt.modules.makeconf.`**`append_gentoo_mirrors`**`(`*value*`)`

Add to or create a new GENTOO_MIRRORS in the make.conf

Return a dict containing the new value for variable:

```
{'<variable>': {'old': '<old-value>',
                'new': '<new-value>'}}
```

CLI Example:

```
salt '*' makeconf.append_gentoo_mirrors 'http://distfiles.gentoo.org'
```

`salt.modules.makeconf.`**`append_makeopts`**`(`*value*`)`

Add to or create a new MAKEOPTS in the make.conf

Return a dict containing the new value for variable:

```
{'<variable>': {'old': '<old-value>',
                'new': '<new-value>'}}
```

CLI Example:

```
salt '*' makeconf.append_makeopts '-j3'
```

salt.modules.makeconf.**append_var**(*var*, *value*)
Add to or create a new variable in the make.conf

Return a dict containing the new value for variable:

```
{'<variable>': {'old': '<old-value>',
                'new': '<new-value>'}}
```

CLI Example:

```
salt '*' makeconf.append_var 'LINGUAS' 'en'
```

salt.modules.makeconf.**cflags_contains**(*value*)
Verify if CFLAGS variable contains a value in make.conf

Return True if value is set for var

CLI Example:

```
salt '*' makeconf.cflags_contains '-pipe'
```

salt.modules.makeconf.**chost_contains**(*value*)
Verify if CHOST variable contains a value in make.conf

Return True if value is set for var

CLI Example:

```
salt '*' makeconf.chost_contains 'x86_64-pc-linux-gnu'
```

salt.modules.makeconf.**cxxflags_contains**(*value*)
Verify if CXXFLAGS variable contains a value in make.conf

Return True if value is set for var

CLI Example:

```
salt '*' makeconf.cxxflags_contains '-pipe'
```

salt.modules.makeconf.**emerge_default_opts_contains**(*value*)
Verify if EMERGE_DEFAULT_OPTS variable contains a value in make.conf

Return True if value is set for var

CLI Example:

```
salt '*' makeconf.emerge_default_opts_contains '--jobs'
```

salt.modules.makeconf.**features_contains**(*value*)
Verify if FEATURES variable contains a value in make.conf

Return True if value is set for var

CLI Example:

```
salt '*' makeconf.features_contains 'webrsync-gpg'
```

`salt.modules.makeconf.`**`gentoo_mirrors_contains`**`(`*value*`)`

Verify if GENTOO_MIRRORS variable contains a value in make.conf

Return True if value is set for var

CLI Example:

```
salt '*' makeconf.gentoo_mirrors_contains 'http://distfiles.gentoo.org'
```

`salt.modules.makeconf.`**`get_cflags`**`()`

Get the value of CFLAGS variable in the make.conf

Return the value of the variable or None if the variable is not in the make.conf

CLI Example:

```
salt '*' makeconf.get_cflags
```

`salt.modules.makeconf.`**`get_chost`**`()`

Get the value of CHOST variable in the make.conf

Return the value of the variable or None if the variable is not in the make.conf

CLI Example:

```
salt '*' makeconf.get_chost
```

`salt.modules.makeconf.`**`get_cxxflags`**`()`

Get the value of CXXFLAGS variable in the make.conf

Return the value of the variable or None if the variable is not in the make.conf

CLI Example:

```
salt '*' makeconf.get_cxxflags
```

`salt.modules.makeconf.`**`get_emerge_default_opts`**`()`

Get the value of EMERGE_DEFAULT_OPTS variable in the make.conf

Return the value of the variable or None if the variable is not in the make.conf

CLI Example:

```
salt '*' makeconf.get_emerge_default_opts
```

`salt.modules.makeconf.`**`get_features`**`()`

Get the value of FEATURES variable in the make.conf

Return the value of the variable or None if the variable is not in the make.conf

CLI Example:

```
salt '*' makeconf.get_features
```

`salt.modules.makeconf.`**`get_gentoo_mirrors`**`()`

Get the value of GENTOO_MIRRORS variable in the make.conf

Return the value of the variable or None if the variable is not in the make.conf

CLI Example:

```
salt '*' makeconf.get_gentoo_mirrors
```

salt.modules.makeconf.**get_makeopts**()

Get the value of MAKEOPTS variable in the make.conf

Return the value of the variable or None if the variable is not in the make.conf

CLI Example:

```
salt '*' makeconf.get_makeopts
```

salt.modules.makeconf.**get_sync**()

Get the value of SYNC variable in the make.conf

Return the value of the variable or None if the variable is not in the make.conf

CLI Example:

```
salt '*' makeconf.get_sync
```

salt.modules.makeconf.**get_var**(*var*)

Get the value of a variable in make.conf

Return the value of the variable or None if the variable is not in make.conf

CLI Example:

```
salt '*' makeconf.get_var 'LINGUAS'
```

salt.modules.makeconf.**makeopts_contains**(*value*)

Verify if MAKEOPTS variable contains a value in make.conf

Return True if value is set for var

CLI Example:

```
salt '*' makeconf.makeopts_contains '-j3'
```

salt.modules.makeconf.**remove_var**(*var*)

Remove a variable from the make.conf

Return a dict containing the new value for the variable:

```
{'<variable>': {'old': '<old-value>',
                'new': '<new-value>'}}
```

CLI Example:

```
salt '*' makeconf.remove_var 'LINGUAS'
```

salt.modules.makeconf.**set_cflags**(*value*)

Set the CFLAGS variable

Return a dict containing the new value for variable:

```
{'<variable>': {'old': '<old-value>',
                'new': '<new-value>'}}
```

CLI Example:

```
salt '*' makeconf.set_cflags '-march=native -O2 -pipe'
```

salt.modules.makeconf.**set_chost**(*value*)

Set the CHOST variable

Return a dict containing the new value for variable:

```
{'<variable>': {'old': '<old-value>',
                'new': '<new-value>'}}
```

CLI Example:

```
salt '*' makeconf.set_chost 'x86_64-pc-linux-gnu'
```

salt.modules.makeconf.**set_cxxflags**(*value*)

Set the CXXFLAGS variable

Return a dict containing the new value for variable:

```
{'<variable>': {'old': '<old-value>',
                'new': '<new-value>'}}
```

CLI Example:

```
salt '*' makeconf.set_cxxflags '-march=native -O2 -pipe'
```

salt.modules.makeconf.**set_emerge_default_opts**(*value*)

Set the EMERGE_DEFAULT_OPTS variable

Return a dict containing the new value for variable:

```
{'<variable>': {'old': '<old-value>',
                'new': '<new-value>'}}
```

CLI Example:

```
salt '*' makeconf.set_emerge_default_opts '--jobs'
```

salt.modules.makeconf.**set_gentoo_mirrors**(*value*)

Set the GENTOO_MIRRORS variable

Return a dict containing the new value for variable:

```
{'<variable>': {'old': '<old-value>',
                'new': '<new-value>'}}
```

CLI Example:

```
salt '*' makeconf.set_gentoo_mirrors 'http://distfiles.gentoo.org'
```

salt.modules.makeconf.**set_makeopts**(*value*)

Set the MAKEOPTS variable

Return a dict containing the new value for variable:

```
{'<variable>': {'old': '<old-value>',
                'new': '<new-value>'}}
```

CLI Example:

```
salt '*' makeconf.set_makeopts '-j3'
```

salt.modules.makeconf.**set_sync**(*value*)

Set the SYNC variable

Return a dict containing the new value for variable:

```
{'<variable>': {'old': '<old-value>',
                'new': '<new-value>'}}
```

CLI Example:

```
salt '*' makeconf.set_sync 'rsync://rsync.namerica.gentoo.org/gentoo-portage'
```

salt.modules.makeconf.**set_var**(*var, value*)

Set a variable in the make.conf

Return a dict containing the new value for variable:

```
{'<variable>': {'old': '<old-value>',
                'new': '<new-value>'}}
```

CLI Example:

```
salt '*' makeconf.set_var 'LINGUAS' 'en'
```

salt.modules.makeconf.**sync_contains**(*value*)

Verify if SYNC variable contains a value in make.conf

Return True if value is set for var

CLI Example:

```
salt '*' makeconf.sync_contains 'rsync://rsync.namerica.gentoo.org/gentoo-portage'
```

salt.modules.makeconf.**trim_cflags**(*value*)

Remove a value from CFLAGS variable in the make.conf

Return a dict containing the new value for variable:

```
{'<variable>': {'old': '<old-value>',
                'new': '<new-value>'}}
```

CLI Example:

```
salt '*' makeconf.trim_cflags '-pipe'
```

salt.modules.makeconf.**trim_cxxflags**(*value*)

Remove a value from CXXFLAGS variable in the make.conf

Return a dict containing the new value for variable:

```
{'<variable>': {'old': '<old-value>',
                'new': '<new-value>'}}
```

CLI Example:

```
salt '*' makeconf.trim_cxxflags '-pipe'
```

`salt.modules.makeconf.`**`trim_emerge_default_opts`**`(`*value*`)`

Remove a value from EMERGE_DEFAULT_OPTS variable in the make.conf

Return a dict containing the new value for variable:

```
{'<variable>': {'old': '<old-value>',
                'new': '<new-value>'}}
```

CLI Example:

```
salt '*' makeconf.trim_emerge_default_opts '--jobs'
```

`salt.modules.makeconf.`**`trim_features`**`(`*value*`)`

Remove a value from FEATURES variable in the make.conf

Return a dict containing the new value for variable:

```
{'<variable>': {'old': '<old-value>',
                'new': '<new-value>'}}
```

CLI Example:

```
salt '*' makeconf.trim_features 'webrsync-gpg'
```

`salt.modules.makeconf.`**`trim_gentoo_mirrors`**`(`*value*`)`

Remove a value from GENTOO_MIRRORS variable in the make.conf

Return a dict containing the new value for variable:

```
{'<variable>': {'old': '<old-value>',
                'new': '<new-value>'}}
```

CLI Example:

```
salt '*' makeconf.trim_gentoo_mirrors 'http://distfiles.gentoo.org'
```

`salt.modules.makeconf.`**`trim_makeopts`**`(`*value*`)`

Remove a value from MAKEOPTS variable in the make.conf

Return a dict containing the new value for variable:

```
{'<variable>': {'old': '<old-value>',
                'new': '<new-value>'}}
```

CLI Example:

```
salt '*' makeconf.trim_makeopts '-j3'
```

`salt.modules.makeconf.`**`trim_var`**`(`*var, value*`)`

Remove a value from a variable in the make.conf

Return a dict containing the new value for variable:

```
{'<variable>': {'old': '<old-value>',
                'new': '<new-value>'}}
```

CLI Example:

```
salt '*' makeconf.trim_var 'LINGUAS' 'en'
```

`salt.modules.makeconf.`**`var_contains`**(*var*, *value*)

Verify if variable contains a value in make.conf

Return True if value is set for var

CLI Example:

```
salt '*' makeconf.var_contains 'LINGUAS' 'en'
```

21.9.236 salt.modules.mandrill

Mandrill

Send out emails using the Mandrill API.

In the minion configuration file, the following block is required:

```
mandrill:
  key: <API_KEY>
```

New in version 2018.3.0.

`salt.modules.mandrill.`**`send`**(*message*, *asynchronous=False*, *ip_pool=None*, *send_at=None*, *api_url=None*, *api_version=None*, *api_key=None*, ***kwargs*)

Send out the email using the details from the `message` argument.

message The information on the message to send. This argument must be sent as dictionary with at fields as specified in the Mandrill API documentation.

asynchronous: `False` Enable a background sending mode that is optimized for bulk sending. In asynchronous mode, messages/send will immediately return a status of "queued" for every recipient. To handle rejections when sending in asynchronous mode, set up a webhook for the 'reject' event. Defaults to false for messages with no more than 10 recipients; messages with more than 10 recipients are always sent asynchronously, regardless of the value of asynchronous.

ip_pool The name of the dedicated ip pool that should be used to send the message. If you do not have any dedicated IPs, this parameter has no effect. If you specify a pool that does not exist, your default pool will be used instead.

send_at When this message should be sent as a UTC timestamp in `YYYY-MM-DD HH:MM:SS` format. If you specify a time in the past, the message will be sent immediately. An additional fee applies for scheduled email, and this feature is only available to accounts with a positive balance.

Note: Fur further details please consult the API documentation.

CLI Example:

```
$ salt '*' mandrill.send message="{'subject': 'Hi', 'from_email': 'test@example.
↪com', 'to': [{'email': 'recv@example.com', 'type': 'to'}]}"
```

`message` structure example (as YAML for readability):

```
message:
    text: |
        This is the body of the email.
        This is the second line.
```

(continues on next page)

(continued from previous page)

```
    subject: Email subject
    from_name: Test At Example Dot Com
    from_email: test@example.com
    to:
      - email: recv@example.com
        type: to
        name: Recv At Example Dot Com
      - email: cc@example.com
        type: cc
        name: CC At Example Dot Com
    important: true
    track_clicks: true
    track_opens: true
    attachments:
      - type: text/x-yaml
        name: yaml_file.yml
        content: aV9hbV9zdXBlcl9jdXJpb3VzOiB0cnVl
```

Output example:

```
minion:
    ----------
    comment:
    out:
        |_
          ----------
          _id:
              c4353540a3c123eca112bbdd704ab6
          email:
              recv@example.com
          reject_reason:
              None
          status:
              sent
    result:
        True
```

21.9.237 salt.modules.marathon module

Module providing a simple management interface to a marathon cluster.

Currently this only works when run through a proxy minion.

New in version 2015.8.2.

`salt.modules.marathon.`**`app`**`(`*`id`*`)`

Return the current server configuration for the specified app.

CLI Example:

```
salt marathon-minion-id marathon.app my-app
```

`salt.modules.marathon.`**`apps`**`()`

Return a list of the currently installed app ids.

CLI Example:

```
salt marathon-minion-id marathon.apps
```

salt.modules.marathon.**has_app**(*id*)

Return whether the given app id is currently configured.

CLI Example:

```
salt marathon-minion-id marathon.has_app my-app
```

salt.modules.marathon.**info**()

Return configuration and status information about the marathon instance.

CLI Example:

```
salt marathon-minion-id marathon.info
```

salt.modules.marathon.**restart_app**(*id*, *restart=False*, *force=True*)

Restart the current server configuration for the specified app.

Parameters

- **restart** – Restart the app
- **force** – Override the current deployment

CLI Example:

```
salt marathon-minion-id marathon.restart_app my-app
```

By default, this will only check if the app exists in marathon. It does not check if there are any tasks associated with it or if the app is suspended.

```
salt marathon-minion-id marathon.restart_app my-app true true
```

The restart option needs to be set to True to actually issue a rolling restart to marathon.

The force option tells marathon to ignore the current app deployment if there is one.

salt.modules.marathon.**rm_app**(*id*)

Remove the specified app from the server.

CLI Example:

```
salt marathon-minion-id marathon.rm_app my-app
```

salt.modules.marathon.**update_app**(*id*, *config*)

Update the specified app with the given configuration.

CLI Example:

```
salt marathon-minion-id marathon.update_app my-app '<config yaml>'
```

21.9.238 salt.modules.match

The match module allows for match routines to be run and determine target specs

salt.modules.match.**compound**(*tgt*, *minion_id=None*)

Return True if the minion ID matches the given compound target

minion_id Specify the minion ID to match against the target expression

New in version 2014.7.0.

CLI Example:

```
salt '*' match.compound 'L@cheese,foo and *'
```

`salt.modules.match.`**`data`**(*tgt*)

Return True if the minion matches the given data target

CLI Example:

```
salt '*' match.data 'spam:eggs'
```

`salt.modules.match.`**`filter_by`**(*lookup*, *tgt_type='compound'*, *minion_id=None*, *default='default'*)

Return the first match in a dictionary of target patterns

New in version 2014.7.0.

CLI Example:

```
salt '*' match.filter_by '{foo*: Foo!, bar*: Bar!}' minion_id=bar03
```

Pillar Example:

```
# Filter the data for the current minion into a variable:
{% set roles = salt['match.filter_by']({
    'web*': ['app', 'caching'],
    'db*': ['db'],
}, default='web*') %}

# Make the filtered data available to Pillar:
roles: {{ roles | yaml() }}
```

`salt.modules.match.`**`glob`**(*tgt*, *minion_id=None*)

Return True if the minion ID matches the given glob target

minion_id Specify the minion ID to match against the target expression

New in version 2014.7.0.

CLI Example:

```
salt '*' match.glob '*'
```

`salt.modules.match.`**`grain`**(*tgt*, *delimiter=':'*)

Return True if the minion matches the given grain target. The `delimiter` argument can be used to specify a different delimiter.

CLI Example:

```
salt '*' match.grain 'os:Ubuntu'
salt '*' match.grain 'ipv6|2001:db8::ff00:42:8329' delimiter='|'
```

delimiter Specify an alternate delimiter to use when traversing a nested dict

New in version 2014.7.0.

delim Specify an alternate delimiter to use when traversing a nested dict

New in version 0.16.4.

Deprecated since version 2015.8.0.

`salt.modules.match.`**`grain_pcre`**(*tgt*, *delimiter=':'*)

Return True if the minion matches the given grain_pcre target. The `delimiter` argument can be used to specify a different delimiter.

CLI Example:

```
salt '*' match.grain_pcre 'os:Fedo.*'
salt '*' match.grain_pcre 'ipv6|2001:.*' delimiter='|'
```

delimiter Specify an alternate delimiter to use when traversing a nested dict

New in version 2014.7.0.

delim Specify an alternate delimiter to use when traversing a nested dict

New in version 0.16.4.

Deprecated since version 2015.8.0.

`salt.modules.match.`**`ipcidr`**(*tgt*)

Return True if the minion matches the given ipcidr target

CLI Example:

```
salt '*' match.ipcidr '192.168.44.0/24'
```

delimiter Pillar Example:

```
'172.16.0.0/12':
  - match: ipcidr
  - nodeclass: internal
```

`salt.modules.match.`**`list_`**(*tgt, minion_id=None*)

Return True if the minion ID matches the given list target

minion_id Specify the minion ID to match against the target expression

New in version 2014.7.0.

CLI Example:

```
salt '*' match.list 'server1,server2'
```

`salt.modules.match.`**`pcre`**(*tgt, minion_id=None*)

Return True if the minion ID matches the given pcre target

minion_id Specify the minion ID to match against the target expression

New in version 2014.7.0.

CLI Example:

```
salt '*' match.pcre '.*'
```

`salt.modules.match.`**`pillar`**(*tgt, delimiter=':'*)

Return True if the minion matches the given pillar target. The `delimiter` argument can be used to specify a different delimiter.

CLI Example:

```
salt '*' match.pillar 'cheese:foo'
salt '*' match.pillar 'clone_url|https://github.com/saltstack/salt.git' delimiter=
↪'|'
```

delimiter Specify an alternate delimiter to use when traversing a nested dict

New in version 2014.7.0.

delim Specify an alternate delimiter to use when traversing a nested dict

New in version 0.16.4.

Deprecated since version 2015.8.0.

`salt.modules.match.`**`pillar_pcre`**(*tgt*, *delimiter=':'*)

Return True if the minion matches the given pillar_pcre target. The `delimiter` argument can be used to specify a different delimiter.

CLI Example:

```
salt '*' match.pillar_pcre 'cheese:(swiss|american)'
salt '*' match.pillar_pcre 'clone_url|https://github\.com/.*\.git' delimiter='|'
```

delimiter Specify an alternate delimiter to use when traversing a nested dict

New in version 2014.7.0.

delim Specify an alternate delimiter to use when traversing a nested dict

New in version 0.16.4.

Deprecated since version 2015.8.0.

`salt.modules.match.`**`search_by`**(*lookup*, *tgt_type='compound'*, *minion_id=None*)

Search a dictionary of target strings for matching targets

This is the inverse of *`match.filter_by`* and allows matching values instead of matching keys. A minion can be matched by multiple entries.

New in version 2017.7.0.

CLI Example:

```
salt '*' match.search_by '{web: [node1, node2], db: [node2, node]}'
```

Pillar Example:

```
{% set roles = salt.match.search_by({
    'web': ['G@os_family:Debian not nodeX'],
    'db': ['L@node2,node3 and G@datacenter:west'],
    'caching': ['node3', 'node4'],
}) %}

# Make the filtered data available to Pillar:
roles: {{ roles | yaml() }}
```

21.9.239 salt.modules.mattermost module

Module for sending messages to Mattermost

New in version 2017.7.0.

configuration This module can be used by either passing an api_url and hook directly or by specifying both in a configuration profile in the salt master/minion config. For example:

```
mattermost:
  hook: peWcBiMOS9HrZG15peWcBiMOS9HrZG15
  api_url: https://example.com
```

`salt.modules.mattermost.`**`post_message`**(*message*, *channel=None*, *username=None*, *api_url=None*, *hook=None*)

Send a message to a Mattermost channel.

Parameters

- **channel** – The channel name, either will work.
- **username** – The username of the poster.
- **message** – The message to send to the Mattermost channel.
- **api_url** – The Mattermost api url, if not specified in the configuration.
- **hook** – The Mattermost hook, if not specified in the configuration.

Returns Boolean if message was sent successfully.

CLI Example:

```
salt '*' mattermost.post_message message='Build is done'
```

21.9.240 salt.modules.mdadm_raid

Salt module to manage RAID arrays with mdadm

`salt.modules.mdadm_raid.`**`add`**`(`*name*, *device*`)`

Add new device to RAID array.

CLI Example:

```
salt '*' raid.add /dev/md0 /dev/sda1
```

`salt.modules.mdadm_raid.`**`assemble`**`(`*name*, *devices*, *test_mode=False*, ***kwargs*`)`

Assemble a RAID device.

CLI Examples:

```
salt '*' raid.assemble /dev/md0 ['/dev/xvdd', '/dev/xvde']
```

Note: Adding `test_mode=True` as an argument will print out the mdadm command that would have been run.

name The name of the array to assemble.

devices The list of devices comprising the array to assemble.

kwargs Optional arguments to be passed to mdadm.

returns

- **test_mode=True:** Prints out the full command.
- **test_mode=False (Default):** Executes command on the host(s) and prints out the mdadm output.

For more info, read the `mdadm` manpage.

`salt.modules.mdadm_raid.`**`create`**`(`*name*, *level*, *devices*, *metadata='default'*, *test_mode=False*, ***kwargs*`)`

Create a RAID device.

Changed in version 2014.7.0.

Warning: Use with CAUTION, as this function can be very destructive if not used properly!

CLI Examples:

```
salt '*' raid.create /dev/md0 level=1 chunk=256 devices="['/dev/xvdd', '/dev/xvde
↪']" test_mode=True
```

Note: Adding `test_mode=True` as an argument will print out the mdadm command that would have been run.

---

name The name of the array to create.
level The RAID level to use when creating the raid.
devices A list of devices used to build the array.
metadata Version of metadata to use when creating the array.
kwargs Optional arguments to be passed to mdadm.
returns
  test_mode=True: Prints out the full command.
  test_mode=False (Default): Executes command on remote the host(s) and Prints out the mdadm output.

Note: It takes time to create a RAID array. You can check the progress in "resync_status:" field of the results from the following command:

```
salt '*' raid.detail /dev/md0
```

For more info, read the `mdadm(8)` manpage

`salt.modules.mdadm_raid.`**`destroy`**`(`*device*`)`
Destroy a RAID device.

WARNING This will zero the superblock of all members of the RAID array..

CLI Example:

```
salt '*' raid.destroy /dev/md0
```

`salt.modules.mdadm_raid.`**`detail`**`(`*device='/dev/md0'*`)`
Show detail for a specified RAID device

CLI Example:

```
salt '*' raid.detail '/dev/md0'
```

`salt.modules.mdadm_raid.`**`examine`**`(`*device, quiet=False*`)`
Show detail for a specified RAID component device
device Device to examine, that is part of the RAID
quiet If the device is not part of the RAID, do not show any error
CLI Example:

```
salt '*' raid.examine '/dev/sda1'
```

`salt.modules.mdadm_raid.`**`list_`**`()`
List the RAID devices.

CLI Example:

```
salt '*' raid.list
```

`salt.modules.mdadm_raid.`**`save_config`**`()`
Save RAID configuration to config file.

Same as: mdadm –detail –scan >> /etc/mdadm/mdadm.conf

Fixes this issue with Ubuntu REF: http://askubuntu.com/questions/209702/why-is-my-raid-dev-md1-showing-up-as-dev-md126-is-mdadm-conf-being-ignored

CLI Example:

```
salt '*' raid.save_config
```

`salt.modules.mdadm_raid.`**`stop`**`()`

Shut down all arrays that can be shut down (i.e. are not currently in use).

CLI Example:

```
salt '*' raid.stop
```

21.9.241 salt.modules.mdata

Module for managaging metadata in SmartOS Zones

New in version 2016.3.0.

maintainer Jorge Schrauwen <sjorge@blackdot.be>

maturity new

platform smartos

`salt.modules.mdata.`**`delete_`**`(`**keyname*`)`

Delete metadata
prop [string] name of property
CLI Example:

```
salt '*' mdata.get salt:role
salt '*' mdata.get user-script salt:role
```

`salt.modules.mdata.`**`get_`**`(`**keyname*`)`

Get metadata
keyname [string] name of key

Note: If no keynames are specified, we get all (public) properties

CLI Example:

```
salt '*' mdata.get salt:role
salt '*' mdata.get user-script salt:role
```

`salt.modules.mdata.`**`list_`**`()`

List available metadata

CLI Example:

```
salt '*' mdata.list
```

`salt.modules.mdata.`**`put_`**`(`*keyname, val*`)`

Put metadata
prop [string] name of property
val [string] value to set
CLI Example:

```
salt '*' mdata.list
```

21.9.242 salt.modules.memcached

Module for Management of Memcached Keys

New in version 2014.1.0.

`salt.modules.memcached.`**`add`**`(`*key, value, host='127.0.0.1', port=11211, time=0, min_compress_len=0*`)`

Add a key to the memcached server, but only if it does not exist. Returns False if the key already exists.

CLI Example:

```
salt '*' memcached.add <key> <value>
```

`salt.modules.memcached.`**`decrement`**`(`*key, delta=1, host='127.0.0.1', port=11211*`)`

Decrement the value of a key

CLI Example:

```
salt '*' memcached.decrement <key>
salt '*' memcached.decrement <key> 2
```

`salt.modules.memcached.`**`delete`**`(`*key, host='127.0.0.1', port=11211, time=0*`)`

Delete a key from memcache server

CLI Example:

```
salt '*' memcached.delete <key>
```

`salt.modules.memcached.`**`get`**`(`*key, host='127.0.0.1', port=11211*`)`

Retrieve value for a key

CLI Example:

```
salt '*' memcached.get <key>
```

`salt.modules.memcached.`**`increment`**`(`*key, delta=1, host='127.0.0.1', port=11211*`)`

Increment the value of a key

CLI Example:

```
salt '*' memcached.increment <key>
salt '*' memcached.increment <key> 2
```

`salt.modules.memcached.`**`replace`**`(`*key, value, host='127.0.0.1', port=11211, time=0, min_compress_len=0*`)`

Replace a key on the memcached server. This only succeeds if the key already exists. This is the opposite of *`memcached.add`*

CLI Example:

```
salt '*' memcached.replace <key> <value>
```

`salt.modules.memcached.`**`set_`**`(`*key, value, host='127.0.0.1', port=11211, time=0, min_compress_len=0*`)`

Set a key on the memcached server, overwriting the value if it exists.

CLI Example:

```
salt '*' memcached.set <key> <value>
```

`salt.modules.memcached.`**`status`**(*host='127.0.0.1'*, *port=11211*)

Get memcached status

CLI Example:

```
salt '*' memcached.status
```

21.9.243 salt.modules.mine

The function cache system allows for data to be stored on the master so it can be easily read by other minions

`salt.modules.mine.`**`delete`**(*fun*)

Remove specific function contents of minion.

Parameters **fun** (*str*) – The name of the function.

Return type bool

Returns True on success.

CLI Example:

```
salt '*' mine.delete 'network.interfaces'
```

`salt.modules.mine.`**`flush`**()

Remove all mine contents of minion.

Return type bool

Returns True on success

CLI Example:

```
salt '*' mine.flush
```

`salt.modules.mine.`**`get`**(*tgt*, *fun*, *tgt_type='glob'*, *exclude_minion=False*)

Get data from the mine.

Parameters

- **tgt** (*str*) – Target whose mine data to get.
- **fun** (*str or list*) – Function to get the mine data of. You can specify multiple functions to retrieve using either a list or a comma-separated string of functions.
- **tgt_type** (*str*) – Default `glob`. Target type to use with `tgt`. See *Targeting Minions* for more information. Note that all pillar matches, whether using the compound matching system or the pillar matching system, will be exact matches, with globbing disabled.
- **exclude_minion** (*bool*) – Excludes the current minion from the result set.

CLI Example:

```
salt '*' mine.get '*' network.interfaces
salt '*' mine.get 'os:Fedora' network.interfaces grain
salt '*' mine.get 'G@os:Fedora and S@192.168.5.0/24' network.ipaddrs compound
```

See also:

Retrieving Mine data from Pillar and Orchestrate

This execution module is intended to be executed on minions. Master-side operations such as Pillar or Orchestrate that require Mine data should use the *`Mine Runner module`* instead; it can be invoked from a Pillar SLS file using the *`saltutil.runner`* module. For example:

```
{% set minion_ips = salt.saltutil.runner('mine.get',
    tgt='*',
    fun='network.ip_addrs',
    tgt_type='glob') %}
```

`salt.modules.mine.`**`get_docker`**(*interfaces=None*, *cidrs=None*, *with_container_id=False*)

Changed in version 2017.7.8,2018.3.3: When *`docker.update_mine`* is set to `False` for a given minion, no mine data will be populated for that minion, and thus none will be returned for it.

Changed in version 2019.2.0: *`docker.update_mine`* now defaults to `False`

Get all mine data for *`docker.ps`* and run an aggregation routine. The `interfaces` parameter allows for specifying the network interfaces from which to select IP addresses. The `cidrs` parameter allows for specifying a list of subnets which the IP address must match.

with_container_id Boolean, to expose container_id in the list of results

New in version 2015.8.2.

CLI Example:

```
salt '*' mine.get_docker
salt '*' mine.get_docker interfaces='eth0'
salt '*' mine.get_docker interfaces='["eth0", "eth1"]'
salt '*' mine.get_docker cidrs='107.170.147.0/24'
salt '*' mine.get_docker cidrs='["107.170.147.0/24", "172.17.42.0/24"]'
salt '*' mine.get_docker interfaces='["eth0", "eth1"]' cidrs='["107.170.147.0/24",
↪ "172.17.42.0/24"]'
```

`salt.modules.mine.`**`send`**(*name*, **args*, ***kwargs*)

Send a specific function and its result to the salt mine. This gets stored in either the local cache, or the salt master's cache.

Parameters **`name`** (*`str`*) – Name of the function to add to the mine.

The following pameters are extracted from kwargs if present:

Parameters

- **`mine_function`** (*`str`*) – The name of the execution_module.function to run and whose value will be stored in the salt mine. Defaults to `name`.
- **`allow_tgt`** (*`str`*) – Targeting specification for ACL. Specifies which minions are allowed to access this function. Please note both your master and minion need to be on, at least, version 3000 for this to work properly.
- **`allow_tgt_type`** (*`str`*) – Type of the targeting specification. This value will be ignored if `allow_tgt` is not specified. Please note both your master and minion need to be on, at least, version 3000 for this to work properly.

Remaining args and kwargs will be passed on to the function to run.

Return type bool

Returns Whether executing the function and storing the information was succesful.

Changed in version 3000: Added `allow_tgt`- and `allow_tgt_type`-parameters to specify which minions are allowed to access this function. See *Targeting Minions* for more information about targeting.

CLI Example:

```
salt '*' mine.send network.ip_addrs eth0
salt '*' mine.send eth0_ip_addrs mine_function=network.ip_addrs eth0
salt '*' mine.send eth0_ip_addrs mine_function=network.ip_addrs eth0 allow_tgt=
↪'G@grain:value' allow_tgt_type=compound
```

`salt.modules.mine.`**`update`**(*clear=False*, *mine_functions=None*)

Call the configured functions and send the data back up to the master. The functions to be called are merged from the master config, pillar and minion config under the option *mine_functions*:

```
mine_functions:
  network.ip_addrs:
    - eth0
  disk.usage: []
```

This function accepts the following arguments:

Parameters

- **`clear`** (*`bool`*) – Default: `False` Specifies whether updating will clear the existing values (`True`), or whether it will update them (`False`).
- **`mine_functions`** (*`dict`*) – Update (or clear, see `clear`) the mine data on these functions only. This will need to have the structure as defined on https://docs.saltstack.com/en/latest/topics/mine/index.html#mine-functions

  This feature can be used when updating the mine for functions that require a refresh at different intervals than the rest of the functions specified under *mine_functions* in the minion/master config or pillar. A potential use would be together with the *scheduler*, for example:

  ```
  schedule:
    lldp_mine_update:
      function: mine.update
      kwargs:
          mine_functions:
            net.lldp: []
      hours: 12
  ```

  In the example above, the mine for *net.lldp* would be refreshed every 12 hours, while *network.ip_addrs* would continue to be updated as specified in *mine_interval*.

The function cache will be populated with information from executing these functions

CLI Example:

```
salt '*' mine.update
```

`salt.modules.mine.`**`valid`**`()`

List valid entries in mine configuration.

CLI Example:

```
salt '*' mine.valid
```

21.9.244 salt.modules.minion module

Module to provide information about minions

`salt.modules.minion.`**`kill`**`(`*timeout=15*`)`

Kill the salt minion.

timeout int seconds to wait for the minion to die.

If you have a monitor that restarts `salt-minion` when it dies then this is a great way to restart after a minion upgrade.

CLI example:

```
>$ salt minion[12] minion.kill
minion1:
    ----------
    killed:
        7874
    retcode:
        0
minion2:
    ----------
    killed:
        29071
    retcode:
        0
```

The result of the salt command shows the process ID of the minions and the results of a kill signal to the minion in as the `retcode` value: `0` is success, anything else is a failure.

salt.modules.minion.**list_**()

Return a list of accepted, denied, unaccepted and rejected keys. This is the same output as *salt-key -L*

CLI Example:

```
salt 'master' minion.list
```

salt.modules.minion.**restart**()

Kill and restart the salt minion.

The configuration key `minion_restart_command` is an argv list for the command to restart the minion. If `minion_restart_command` is not specified or empty then the `argv` of the current process will be used.

if the configuration value `minion_restart_command` is not set and the `-d` (daemonize) argument is missing from `argv` then the minion *will* be killed but will *not* be restarted and will require the parent process to perform the restart. This behavior is intended for managed salt minion processes.

CLI example:

```
>$ salt minion[12] minion.restart
minion1:
    ----------
    comment:
        - Restart using process argv:
        -     /home/omniture/install/bin/salt-minion
        -     -d
        -     -c
        -     /home/omniture/install/etc/salt
    killed:
        10070
    restart:
        ----------
        stderr:
        stdout:
    retcode:
        0
minion2:
    ----------
    comment:
        - Using configuration minion_restart_command:
        -     /home/omniture/install/bin/salt-minion
```

(continues on next page)

(continued from previous page)

```
        -     --not-an-option
        -     -d
        -     -c
        -     /home/omniture/install/etc/salt
        - Restart failed
    killed:
        10896
    restart:
        ----------
        stderr:
            Usage: salt-minion

            salt-minion: error: no such option: --not-an-option
        stdout:
    retcode:
        64
```

The result of the command shows the process ID of `minion1` that is shutdown (killed) and the results of the restart. If there is a failure in the restart it will be reflected in a non-zero `retcode` and possibly output in the `stderr` and/or `stdout` values along with addition information in the `comment` field as is demonstrated with `minion2`.

21.9.245 salt.modules.mod_random

Provides access to randomness generators.

New in version 2014.7.0.

`salt.modules.mod_random.`**`get_str`**`(`*length=20*`)`

New in version 2014.7.0.

Returns a random string of the specified length.
length [20] Any valid number of bytes.
CLI Example:

```
salt '*' random.get_str 128
```

`salt.modules.mod_random.`**`hash`**`(`*value, algorithm='sha512'*`)`

New in version 2014.7.0.

Encodes a value with the specified encoder.
value The value to be hashed.
algorithm [sha512] The algorithm to use. May be any valid algorithm supported by hashlib.
CLI Example:

```
salt '*' random.hash 'I am a string' md5
```

`salt.modules.mod_random.`**`rand_int`**`(`*start=1, end=10, seed=None*`)`

Returns a random integer number between the start and end number.
start [1] Any valid integer number
end [10] Any valid integer number
seed : Optional hashable object
Changed in version 2019.2.0: Added seed argument. Will return the same result when run with the same seed.

CLI Example:

```
salt '*' random.rand_int 1 10
```

`salt.modules.mod_random.`**`seed`**(*range=10*, *hash=None*)

Returns a random number within a range. Optional hash argument can be any hashable object. If hash is omitted or None, the id of the minion is used.

hash: **None** Any hashable object.

range: **10** Any valid integer number

CLI Example:

```
salt '*' random.seed 10 hash=None
```

`salt.modules.mod_random.`**`shadow_hash`**(*crypt_salt=None*, *password=None*, *algorithm='sha512'*)

Generates a salted hash suitable for /etc/shadow.

crypt_salt [None] Salt to be used in the generation of the hash. If one is not provided, a random salt will be generated.

password [None] Value to be salted and hashed. If one is not provided, a random password will be generated.

algorithm [sha512] Hash algorithm to use.

CLI Example:

```
salt '*' random.shadow_hash 'My5alT' 'MyP@asswd' md5
```

`salt.modules.mod_random.`**`str_encode`**(*value*, *encoder='base64'*)

New in version 2014.7.0.

value The value to be encoded.

encoder [base64] The encoder to use on the subsequent string.

CLI Example:

```
salt '*' random.str_encode 'I am a new string' base64
```

21.9.246 salt.modules.modjk

Control Modjk via the Apache Tomcat "Status" worker (http://tomcat.apache.org/connectors-doc/reference/status.html)

Below is an example of the configuration needed for this module. This configuration data can be placed either in *grains* or *pillar*.

If using grains, this can be accomplished *statically* or via a *grain module*.

If using pillar, the yaml configuration can be placed directly into a pillar SLS file, making this both the easier and more dynamic method of configuring this module.

```
modjk:
  default:
    url: http://localhost/jkstatus
    user: modjk
    pass: secret
    realm: authentication realm for digest passwords
    timeout: 5
  otherVhost:
    url: http://otherVhost/jkstatus
    user: modjk
    pass: secret2
    realm: authentication realm2 for digest passwords
    timeout: 600
```

`salt.modules.modjk.`**`bulk_activate`**(*workers, lbn, profile='default'*)

Activate all the given workers in the specific load balancer

CLI Examples:

```
salt '*' modjk.bulk_activate node1,node2,node3 loadbalancer1
salt '*' modjk.bulk_activate node1,node2,node3 loadbalancer1 other-profile

salt '*' modjk.bulk_activate ["node1","node2","node3"] loadbalancer1
salt '*' modjk.bulk_activate ["node1","node2","node3"] loadbalancer1 other-profile
```

`salt.modules.modjk.`**`bulk_disable`**(*workers, lbn, profile='default'*)

Disable all the given workers in the specific load balancer

CLI Examples:

```
salt '*' modjk.bulk_disable node1,node2,node3 loadbalancer1
salt '*' modjk.bulk_disable node1,node2,node3 loadbalancer1 other-profile

salt '*' modjk.bulk_disable ["node1","node2","node3"] loadbalancer1
salt '*' modjk.bulk_disable ["node1","node2","node3"] loadbalancer1 other-profile
```

`salt.modules.modjk.`**`bulk_recover`**(*workers, lbn, profile='default'*)

Recover all the given workers in the specific load balancer

CLI Examples:

```
salt '*' modjk.bulk_recover node1,node2,node3 loadbalancer1
salt '*' modjk.bulk_recover node1,node2,node3 loadbalancer1 other-profile

salt '*' modjk.bulk_recover ["node1","node2","node3"] loadbalancer1
salt '*' modjk.bulk_recover ["node1","node2","node3"] loadbalancer1 other-profile
```

`salt.modules.modjk.`**`bulk_stop`**(*workers, lbn, profile='default'*)

Stop all the given workers in the specific load balancer

CLI Examples:

```
salt '*' modjk.bulk_stop node1,node2,node3 loadbalancer1
salt '*' modjk.bulk_stop node1,node2,node3 loadbalancer1 other-profile

salt '*' modjk.bulk_stop ["node1","node2","node3"] loadbalancer1
salt '*' modjk.bulk_stop ["node1","node2","node3"] loadbalancer1 other-profile
```

`salt.modules.modjk.`**`dump_config`**(*profile='default'*)

Dump the original configuration that was loaded from disk

CLI Examples:

```
salt '*' modjk.dump_config
salt '*' modjk.dump_config other-profile
```

`salt.modules.modjk.`**`get_running`**(*profile='default'*)

Get the current running config (not from disk)

CLI Examples:

```
salt '*' modjk.get_running
salt '*' modjk.get_running other-profile
```

`salt.modules.modjk.`**`lb_edit`**(*lbn*, *settings*, *profile='default'*)

Edit the loadbalancer settings

Note: http://tomcat.apache.org/connectors-doc/reference/status.html Data Parameters for the standard Update Action

CLI Examples:

```
salt '*' modjk.lb_edit loadbalancer1 "{'vlr': 1, 'vlt': 60}"
salt '*' modjk.lb_edit loadbalancer1 "{'vlr': 1, 'vlt': 60}" other-profile
```

`salt.modules.modjk.`**`list_configured_members`**(*lbn*, *profile='default'*)

Return a list of member workers from the configuration files

CLI Examples:

```
salt '*' modjk.list_configured_members loadbalancer1
salt '*' modjk.list_configured_members loadbalancer1 other-profile
```

`salt.modules.modjk.`**`recover_all`**(*lbn*, *profile='default'*)

Set the all the workers in lbn to recover and activate them if they are not

CLI Examples:

```
salt '*' modjk.recover_all loadbalancer1
salt '*' modjk.recover_all loadbalancer1 other-profile
```

`salt.modules.modjk.`**`reset_stats`**(*lbn*, *profile='default'*)

Reset all runtime statistics for the load balancer

CLI Examples:

```
salt '*' modjk.reset_stats loadbalancer1
salt '*' modjk.reset_stats loadbalancer1 other-profile
```

`salt.modules.modjk.`**`version`**(*profile='default'*)

Return the modjk version

CLI Examples:

```
salt '*' modjk.version
salt '*' modjk.version other-profile
```

`salt.modules.modjk.`**`worker_activate`**(*worker*, *lbn*, *profile='default'*)

Set the worker to activate state in the lbn load balancer

CLI Examples:

```
salt '*' modjk.worker_activate node1 loadbalancer1
salt '*' modjk.worker_activate node1 loadbalancer1 other-profile
```

`salt.modules.modjk.`**`worker_disable`**(*worker*, *lbn*, *profile='default'*)

Set the worker to disable state in the lbn load balancer

CLI Examples:

```
salt '*' modjk.worker_disable node1 loadbalancer1
salt '*' modjk.worker_disable node1 loadbalancer1 other-profile
```

`salt.modules.modjk.`**`worker_edit`**(*worker, lbn, settings, profile='default'*)

Edit the worker settings

Note: http://tomcat.apache.org/connectors-doc/reference/status.html Data Parameters for the standard Update Action

CLI Examples:

```
salt '*' modjk.worker_edit node1 loadbalancer1 "{'vwf': 500, 'vwd': 60}"
salt '*' modjk.worker_edit node1 loadbalancer1 "{'vwf': 500, 'vwd': 60}" other-
↪profile
```

`salt.modules.modjk.`**`worker_recover`**(*worker, lbn, profile='default'*)

Set the worker to recover this module will fail if it is in OK state

CLI Examples:

```
salt '*' modjk.worker_recover node1 loadbalancer1
salt '*' modjk.worker_recover node1 loadbalancer1 other-profile
```

`salt.modules.modjk.`**`worker_status`**(*worker, profile='default'*)

Return the state of the worker

CLI Examples:

```
salt '*' modjk.worker_status node1
salt '*' modjk.worker_status node1 other-profile
```

`salt.modules.modjk.`**`worker_stop`**(*worker, lbn, profile='default'*)

Set the worker to stopped state in the lbn load balancer

CLI Examples:

```
salt '*' modjk.worker_activate node1 loadbalancer1
salt '*' modjk.worker_activate node1 loadbalancer1 other-profile
```

`salt.modules.modjk.`**`workers`**(*profile='default'*)

Return a list of member workers and their status

CLI Examples:

```
salt '*' modjk.workers
salt '*' modjk.workers other-profile
```

21.9.247 salt.modules.mongodb

Module to provide MongoDB functionality to Salt

configuration This module uses PyMongo, and accepts configuration details as parameters as well as configuration settings:

```
mongodb.host: 'localhost'
mongodb.port: 27017
mongodb.user: ''
mongodb.password: ''
```

This data can also be passed into pillar. Options passed into opts will overwrite options passed into pillar.

`salt.modules.mongodb.`**`db_exists`**(*name*, *user=None*, *password=None*, *host=None*, *port=None*, *authdb=None*)

Checks if a database exists in MongoDB

CLI Example:

```
salt '*' mongodb.db_exists <name> <user> <password> <host> <port>
```

`salt.modules.mongodb.`**`db_list`**(*user=None*, *password=None*, *host=None*, *port=None*, *authdb=None*)

List all MongoDB databases

CLI Example:

```
salt '*' mongodb.db_list <user> <password> <host> <port>
```

`salt.modules.mongodb.`**`db_remove`**(*name*, *user=None*, *password=None*, *host=None*, *port=None*, *authdb=None*)

Remove a MongoDB database

CLI Example:

```
salt '*' mongodb.db_remove <name> <user> <password> <host> <port>
```

`salt.modules.mongodb.`**`find`**(*collection*, *query=None*, *user=None*, *password=None*, *host=None*, *port=None*, *database='admin'*, *authdb=None*)

Find an object or list of objects in a collection

CLI Example:

```
salt '*' mongodb.find mycollection '[{"foo": "FOO", "bar": "BAR"}]' <user>
↪<password> <host> <port> <database>
```

`salt.modules.mongodb.`**`insert`**(*objects*, *collection*, *user=None*, *password=None*, *host=None*, *port=None*, *database='admin'*, *authdb=None*)

Insert an object or list of objects into a collection

CLI Example:

```
salt '*' mongodb.insert '[{"foo": "FOO", "bar": "BAR"}, {"foo": "BAZ", "bar": "BAM
↪"}]' mycollection <user> <password> <host> <port> <database>
```

`salt.modules.mongodb.`**`remove`**(*collection*, *query=None*, *user=None*, *password=None*, *host=None*, *port=None*, *database='admin'*, *w=1*, *authdb=None*)

Remove an object or list of objects into a collection

CLI Example:

```
salt '*' mongodb.remove mycollection '[{"foo": "FOO", "bar": "BAR"}, {"foo": "BAZ
↪", "bar": "BAM"}]' <user> <password> <host> <port> <database>
```

`salt.modules.mongodb.`**`update_one`**(*objects*, *collection*, *user=None*, *password=None*, *host=None*, *port=None*, *database='admin'*, *authdb=None*)

Update an object into a collection http://api.mongodb.com/python/current/api/pymongo/collection.html#pymongo.collection.Collection.update_one

New in version 2016.11.0.

CLI Example:

```
salt '*' mongodb.update_one '{"_id": "my_minion"} {"bar": "BAR"}' mycollection
↪<user> <password> <host> <port> <database>
```

salt.modules.mongodb.**user_create**(*name*, *passwd*, *user=None*, *password=None*, *host=None*, *port=None*, *database='admin'*, *authdb=None*, *roles=None*)

Create a MongoDB user

CLI Example:

```
salt '*' mongodb.user_create <user_name> <user_password> <roles> <user> <password>
↪ <host> <port> <database>
```

salt.modules.mongodb.**user_exists**(*name*, *user=None*, *password=None*, *host=None*, *port=None*, *database='admin'*, *authdb=None*)

Checks if a user exists in MongoDB

CLI Example:

```
salt '*' mongodb.user_exists <name> <user> <password> <host> <port> <database>
```

salt.modules.mongodb.**user_find**(*name*, *user=None*, *password=None*, *host=None*, *port=None*, *database='admin'*, *authdb=None*)

Get single user from MongoDB

CLI Example:

```
salt '*' mongodb.user_find <name> <user> <password> <host> <port> <database>
↪<authdb>
```

salt.modules.mongodb.**user_grant_roles**(*name*, *roles*, *database*, *user=None*, *password=None*, *host=None*, *port=None*, *authdb=None*)

Grant one or many roles to a MongoDB user

CLI Examples:

```
salt '*' mongodb.user_grant_roles johndoe '["readWrite"]' dbname admin adminpwd
↪localhost 27017
```

```
salt '*' mongodb.user_grant_roles janedoe '[{"role": "readWrite", "db": "dbname" }
↪, {"role": "read", "db": "otherdb"}]' dbname admin adminpwd localhost 27017
```

salt.modules.mongodb.**user_list**(*user=None*, *password=None*, *host=None*, *port=None*, *database='admin'*, *authdb=None*)

List users of a MongoDB database

CLI Example:

```
salt '*' mongodb.user_list <user> <password> <host> <port> <database>
```

salt.modules.mongodb.**user_remove**(*name*, *user=None*, *password=None*, *host=None*, *port=None*, *database='admin'*, *authdb=None*)

Remove a MongoDB user

CLI Example:

```
salt '*' mongodb.user_remove <name> <user> <password> <host> <port> <database>
```

salt.modules.mongodb.**user_revoke_roles**(*name*, *roles*, *database*, *user=None*, *password=None*, *host=None*, *port=None*, *authdb=None*)

Revoke one or many roles to a MongoDB user

CLI Examples:

```
salt '*' mongodb.user_revoke_roles johndoe '["readWrite"]' dbname admin adminpwd
↪localhost 27017
```

```
salt '*' mongodb.user_revoke_roles janedoe '[{"role": "readWrite", "db": "dbname"
↪}, {"role": "read", "db": "otherdb"}]' dbname admin adminpwd localhost 27017
```

salt.modules.mongodb.**user_roles_exists**(*name*, *roles*, *database*, *user=None*, *password=None*, *host=None*, *port=None*, *authdb=None*)

Checks if a user of a MongoDB database has specified roles

CLI Examples:

```
salt '*' mongodb.user_roles_exists johndoe '["readWrite"]' dbname admin adminpwd
↪localhost 27017
```

```
salt '*' mongodb.user_roles_exists johndoe '[{"role": "readWrite", "db": "dbname"
↪}, {"role": "read", "db": "otherdb"}]' dbname admin adminpwd localhost 27017
```

salt.modules.mongodb.**version**(*user=None*, *password=None*, *host=None*, *port=None*, *database='admin'*, *authdb=None*)

Get MongoDB instance version

CLI Example:

```
salt '*' mongodb.version <user> <password> <host> <port> <database>
```

21.9.248 salt.modules.monit

Monit service module. This module will create a monit type service watcher.

salt.modules.monit.**configtest**()

New in version 2016.3.0.

Test monit configuration syntax

CLI Example:

```
salt '*' monit.configtest
```

salt.modules.monit.**id_**(*reset=False*)

New in version 2016.3.0.

Return monit unique id.
reset [False] Reset current id and generate a new id when it's True.
CLI Example:

```
salt '*' monit.id [reset=True]
```

salt.modules.monit.**monitor**(*name*)

monitor service via monit

CLI Example:

```
salt '*' monit.monitor <service name>
```

salt.modules.monit.**reload_**()

New in version 2016.3.0.

Reload monit configuration

CLI Example:

```
salt '*' monit.reload
```

salt.modules.monit.restart(*name*)

Restart service via monit

CLI Example:

```
salt '*' monit.restart <service name>
```

salt.modules.monit.start(*name*)

CLI Example:

```
salt '*' monit.start <service name>
```

salt.modules.monit.status(*svc_name=''*)

Display a process status from monit

CLI Example:

```
salt '*' monit.status
salt '*' monit.status <service name>
```

salt.modules.monit.stop(*name*)

Stops service via monit

CLI Example:

```
salt '*' monit.stop <service name>
```

salt.modules.monit.summary(*svc_name=''*)

Display a summary from monit

CLI Example:

```
salt '*' monit.summary
salt '*' monit.summary <service name>
```

salt.modules.monit.unmonitor(*name*)

Unmonitor service via monit

CLI Example:

```
salt '*' monit.unmonitor <service name>
```

salt.modules.monit.validate()

New in version 2016.3.0.

Check all services

CLI Example:

```
salt '*' monit.validate
```

salt.modules.monit.version()

New in version 2016.3.0.

Return version from monit -V

CLI Example:

```
salt '*' monit.version
```

21.9.249 salt.modules.moosefs

Module for gathering and managing information about MooseFS

`salt.modules.moosefs.`**`dirinfo`**`(`*path, opts=None*`)`

Return information on a directory located on the Moose

CLI Example:

```
salt '*' moosefs.dirinfo /path/to/dir/ [-[n][h|H]]
```

`salt.modules.moosefs.`**`fileinfo`**`(`*path*`)`

Return information on a file located on the Moose

CLI Example:

```
salt '*' moosefs.fileinfo /path/to/dir/
```

`salt.modules.moosefs.`**`getgoal`**`(`*path, opts=None*`)`

Return goal(s) for a file or directory

CLI Example:

```
salt '*' moosefs.getgoal /path/to/file [-[n][h|H]]
salt '*' moosefs.getgoal /path/to/dir/ [-[n][h|H][r]]
```

`salt.modules.moosefs.`**`mounts`**`()`

Return a list of current MooseFS mounts

CLI Example:

```
salt '*' moosefs.mounts
```

21.9.250 salt.modules.mount

Salt module to manage Unix mounts and the fstab file

`salt.modules.mount.`**`active`**`(`*extended=False*`)`

List the active mounts.

CLI Example:

```
salt '*' mount.active
```

`salt.modules.mount.`**`automaster`**`(`*config='/etc/auto_salt'*`)`

List the contents of the auto master

CLI Example:

```
salt '*' mount.automaster
```

`salt.modules.mount.`**`delete_mount_cache`**`(`*real_name*`)`

New in version 2018.3.0.

Provide information if the path is mounted

CLI Example:

```
salt '*' mount.delete_mount_cache /mnt/share
```

`salt.modules.mount.`**`filesystems`**`(`*config='/etc/filesystems'*`)`

New in version 2018.3.3.

List the contents of the filesystems

CLI Example:

```
salt '*' mount.filesystems
```

`salt.modules.mount.`**`fstab`**`(`*config='/etc/fstab'*`)`

Changed in version 2016.3.2.

List the contents of the fstab

CLI Example:

```
salt '*' mount.fstab
```

`salt.modules.mount.`**`is_fuse_exec`**`(`*cmd*`)`

Returns true if the command passed is a fuse mountable application.

CLI Example:

```
salt '*' mount.is_fuse_exec sshfs
```

`salt.modules.mount.`**`is_mounted`**`(`*name*`)`

New in version 2014.7.0.

Provide information if the path is mounted

CLI Example:

```
salt '*' mount.is_mounted /mnt/share
```

`salt.modules.mount.`**`mount`**`(`*name, device, mkmnt=False, fstype='', opts='defaults', user=None, util='mount'*`)`

Mount a device

CLI Example:

```
salt '*' mount.mount /mnt/foo /dev/sdz1 True
```

`salt.modules.mount.`**`read_mount_cache`**`(`*name*`)`

New in version 2018.3.0.

Provide information if the path is mounted

CLI Example:

```
salt '*' mount.read_mount_cache /mnt/share
```

`salt.modules.mount.`**`remount`**`(`*name, device, mkmnt=False, fstype='', opts='defaults', user=None*`)`

Attempt to remount a device, if the device is not already mounted, mount is called

CLI Example:

```
salt '*' mount.remount /mnt/foo /dev/sdz1 True
```

`salt.modules.mount.`**`rm_automaster`**`(`*name, device, config='/etc/auto_salt'*`)`

Remove the mount point from the auto_master

CLI Example:

```
salt '*' mount.rm_automaster /mnt/foo /dev/sdg
```

`salt.modules.mount.`**`rm_filesystems`**`(`*name, device, config='/etc/filesystems'*`)`

New in version 2018.3.3.

Remove the mount point from the filesystems

CLI Example:

```
salt '*' mount.rm_filesystems /mnt/foo /dev/sdg
```

`salt.modules.mount.`**`rm_fstab`**`(`*name, device, config='/etc/fstab'*`)`

Changed in version 2016.3.2.

Remove the mount point from the fstab

CLI Example:

```
salt '*' mount.rm_fstab /mnt/foo /dev/sdg
```

`salt.modules.mount.`**`rm_vfstab`**`(`*name, device, config='/etc/vfstab'*`)`

New in version 2016.3.2.

Remove the mount point from the vfstab

CLI Example:

```
salt '*' mount.rm_vfstab /mnt/foo /device/c0t0d0p0
```

`salt.modules.mount.`**`set_automaster`**`(`*name, device, fstype, opts='', config='/etc/auto_salt', test=False, not_change=False, **kwargs*`)`

Verify that this mount is represented in the auto_salt, change the mount to match the data passed, or add the mount if it is not present.

CLI Example:

```
salt '*' mount.set_automaster /mnt/foo /dev/sdz1 ext4
```

`salt.modules.mount.`**`set_filesystems`**`(`*name, device, vfstype, opts='-', mount='true', config='/etc/filesystems', test=False, match_on='auto', not_change=False, **kwargs*`)`

New in version 2018.3.3.

Verify that this mount is represented in the filesystems, change the mount to match the data passed, or add the mount if it is not present on AIX

If the entry is found via *match_on* and *not_change* is True, the current line will be preserved.

Provide information if the path is mounted

Parameters

- **`name`** – The name of the mount point where the device is mounted.

- **device** – The device that is being mounted.
- **vfstype** – The file system that is used (AIX has two fstypes, fstype and vfstype - similar to Linux fstype)
- **opts** – Additional options used when mounting the device.
- **mount** – Mount if not mounted, default True.
- **config** – Configuration file, default /etc/filesystems.
- **match** – File systems type to match on, default auto

CLI Example:

```
salt '*' mount.set_filesystems /mnt/foo /dev/sdz1 jfs2
```

salt.modules.mount.**set_fstab**(*name, device, fstype, opts='defaults', dump=0, pass_num=0, config='/etc/fstab', test=False, match_on='auto', not_change=False, **kwargs*)

Verify that this mount is represented in the fstab, change the mount to match the data passed, or add the mount if it is not present.

If the entry is found via *match_on* and *not_change* is True, the current line will be preserved.

CLI Example:

```
salt '*' mount.set_fstab /mnt/foo /dev/sdz1 ext4
```

salt.modules.mount.**set_vfstab**(*name, device, fstype, opts='-', device_fsck='-', pass_fsck='-', mount_at_boot='yes', config='/etc/vfstab', test=False, match_on='auto', not_change=False, **kwargs*)

..verionadded:: 2016.3.2 Verify that this mount is represented in the fstab, change the mount to match the data passed, or add the mount if it is not present.

If the entry is found via *match_on* and *not_change* is True, the current line will be preserved.

CLI Example:

```
salt '*' mount.set_vfstab /mnt/foo /device/c0t0d0p0 ufs
```

salt.modules.mount.**swapoff**(*name*)

Deactivate a named swap mount

Changed in version 2016.3.2.

CLI Example:

```
salt '*' mount.swapoff /root/swapfile
```

salt.modules.mount.**swapon**(*name, priority=None*)

Activate a swap disk

Changed in version 2016.3.2.

CLI Example:

```
salt '*' mount.swapon /root/swapfile
```

salt.modules.mount.**swaps**()

Return a dict containing information on active swap

Changed in version 2016.3.2.

CLI Example:

```
salt '*' mount.swaps
```

`salt.modules.mount.`**`umount`**(*name*, *device=None*, *user=None*, *util='mount'*)

Attempt to unmount a device by specifying the directory it is mounted on

CLI Example:

```
salt '*' mount.umount /mnt/foo
```

New in version 2015.5.0.

```
salt '*' mount.umount /mnt/foo /dev/xvdc1
```

`salt.modules.mount.`**`vfstab`**(*config='/etc/vfstab'*)

New in version 2016.3.2.

List the contents of the vfstab

CLI Example:

```
salt '*' mount.vfstab
```

`salt.modules.mount.`**`write_mount_cache`**(*real_name*, *device*, *mkmnt*, *fstype*, *mount_opts*)

New in version 2018.3.0.

Provide information if the path is mounted

Parameters

- **`real_name`** – The real name of the mount point where the device is mounted.
- **`device`** – The device that is being mounted.
- **`mkmnt`** – Whether or not the mount point should be created.
- **`fstype`** – The file system that is used.
- **`mount_opts`** – Additional options used when mounting the device.

Returns Boolean if message was sent successfully.

CLI Example:

```
salt '*' mount.write_mount_cache /mnt/share /dev/sda1 False ext4 defaults,nosuid
```

21.9.251 salt.modules.mssql

Module to provide MS SQL Server compatibility to salt.

depends

- FreeTDS
- pymssql Python module

configuration In order to connect to MS SQL Server, certain configuration is required in minion configs/pillars on the relevant minions. Some sample pillars might look like:

```
mssql.server: 'localhost'
mssql.port:   1433
mssql.user:   'sysdba'
```

(continues on next page)

(continued from previous page)

```
mssql.password:   'Some preferable complex password'
mssql.database: ''
```

The default for the port is '1433' and for the database is " (empty string); in most cases they can be left at the default setting. Options that are directly passed into functions will overwrite options from configs or pillars.

`salt.modules.mssql.`**`db_create`**(*database*, *containment='NONE'*, *new_database_options=None*, ***kwargs*)

Creates a new database. Does not update options of existing databases. new_database_options can only be a list of strings

CLI Example:

```
salt minion mssql.db_create DB_NAME
```

`salt.modules.mssql.`**`db_exists`**(*database_name*, ***kwargs*)

Find if a specific database exists on the MS SQL server.

CLI Example:

```
salt minion mssql.db_exists database_name='DBNAME'
```

`salt.modules.mssql.`**`db_list`**(***kwargs*)

Return the database list created on a MS SQL server.

CLI Example:

```
salt minion mssql.db_list
```

`salt.modules.mssql.`**`db_remove`**(*database_name*, ***kwargs*)

Drops a specific database from the MS SQL server. It will not drop any of 'master', 'model', 'msdb' or 'tempdb'.

CLI Example:

```
salt minion mssql.db_remove database_name='DBNAME'
```

`salt.modules.mssql.`**`login_create`**(*login*, *new_login_password=None*, *new_login_domain="*, *new_login_roles=None*, *new_login_options=None*, ***kwargs*)

Creates a new login. Does not update password of existing logins. For Windows authentication, provide `new_login_domain`. For SQL Server authentication, prvide `new_login_password`. Since hashed passwords are *varbinary* values, if the `new_login_password` is 'int / long', it will be considered to be HASHED.

new_login_roles a list of SERVER roles

new_login_options a list of strings

CLI Example:

```
salt minion mssql.login_create LOGIN_NAME database=DBNAME [new_login_
↪password=PASSWORD]
```

`salt.modules.mssql.`**`login_exists`**(*login*, *domain="*, ***kwargs*)

Find if a login exists in the MS SQL server. domain, if provided, will be prepended to login

CLI Example:

```
salt minion mssql.login_exists 'LOGIN'
```

`salt.modules.mssql.`**`login_remove`**(*login*, ***kwargs*)

Removes an login.

CLI Example:

```
salt minion mssql.login_remove LOGINNAME
```

`salt.modules.mssql.`**`role_create`**(*role*, *owner=None*, *grants=None*, ***kwargs*)

Creates a new database role. If no owner is specified, the role will be owned by the user that executes CREATE ROLE, which is the user argument or mssql.user option. grants is list of strings.

CLI Example:

```
salt minion mssql.role_create role=product01 owner=sysdba grants='["SELECT",
↪"INSERT", "UPDATE", "DELETE", "EXECUTE"]'
```

`salt.modules.mssql.`**`role_exists`**(*role*, ***kwargs*)

Checks if a role exists.

CLI Example:

```
salt minion mssql.role_exists db_owner
```

`salt.modules.mssql.`**`role_list`**(***kwargs*)

Lists database roles.

CLI Example:

```
salt minion mssql.role_list
```

`salt.modules.mssql.`**`role_remove`**(*role*, ***kwargs*)

Remove a database role.

CLI Example:

```
salt minion mssql.role_create role=test_role01
```

`salt.modules.mssql.`**`tsql_query`**(*query*, ***kwargs*)

Run a SQL query and return query result as list of tuples, or a list of dictionaries if as_dict was passed, or an empty list if no data is available.

CLI Example:

```
salt minion mssql.tsql_query 'SELECT @@version as version' as_dict=True
```

`salt.modules.mssql.`**`user_create`**(*username*, *login=None*, *domain=''*, *database=None*, *roles=None*, *options=None*, ***kwargs*)

Creates a new user. If login is not specified, the user will be created without a login. domain, if provided, will be prepended to username. options can only be a list of strings

CLI Example:

```
salt minion mssql.user_create USERNAME database=DBNAME
```

`salt.modules.mssql.`**`user_exists`**(*username*, *domain=''*, *database=None*, ***kwargs*)

Find if an user exists in a specific database on the MS SQL server. domain, if provided, will be prepended to username

CLI Example:

```
salt minion mssql.user_exists 'USERNAME' [database='DBNAME']
```

`salt.modules.mssql.`**`user_list`**`(`*`**kwargs`*`)`

Get the user list for a specific database on the MS SQL server.

CLI Example:

```
salt minion mssql.user_list [database='DBNAME']
```

`salt.modules.mssql.`**`user_remove`**`(`*`username, **kwargs`*`)`

Removes an user.

CLI Example:

```
salt minion mssql.user_remove USERNAME database=DBNAME
```

`salt.modules.mssql.`**`version`**`(`*`**kwargs`*`)`

Return the version of a MS SQL server.

CLI Example:

```
salt minion mssql.version
```

21.9.252 salt.modules.msteams module

Module for sending messages to MS Teams

New in version 2017.7.0.

configuration This module can be used by either passing a hook_url directly or by specifying it in a configuration profile in the salt master/minion config. For example:

```
msteams:
  hook_url: https://outlook.office.com/webhook/837
```

`salt.modules.msteams.`**`post_card`**`(`*`message, hook_url=None, title=None, theme_color=None`*`)`

Send a message to an MS Teams channel. :param message: The message to send to the MS Teams channel. :param hook_url: The Teams webhook URL, if not specified in the configuration. :param title: Optional title for the posted card :param theme_color: Optional hex color highlight for the posted card :return: Boolean if message was sent successfully.

CLI Example:

```
salt '*' msteams.post_card message="Build is done"
```

21.9.253 salt.modules.munin

Run munin plugins/checks from salt and format the output as data.

`salt.modules.munin.`**`list_plugins`**`()`

List all the munin plugins

CLI Example:

```
salt '*' munin.list_plugins
```

`salt.modules.munin.`**`run`**`(`*`plugins`*`)`

Run one or more named munin plugins

CLI Example:

```
salt '*' munin.run uptime
salt '*' munin.run uptime,cpu,load,memory
```

`salt.modules.munin.`**`run_all`**`()`

Run all the munin plugins

CLI Example:

```
salt '*' munin.run_all
```

21.9.254 salt.modules.mysql

Module to provide MySQL compatibility to salt.

depends

- MySQLdb Python module

Note: On CentOS 5 (and possibly RHEL 5) both MySQL-python and python26-mysqldb need to be installed.

---

configuration In order to connect to MySQL, certain configuration is required in /etc/salt/minion on the relevant minions. Some sample configs might look like:

```
mysql.host: 'localhost'
mysql.port: 3306
mysql.user: 'root'
mysql.pass: ''
mysql.db: 'mysql'
mysql.unix_socket: '/tmp/mysql.sock'
mysql.charset: 'utf8'
```

You can also use a defaults file:

```
mysql.default_file: '/etc/mysql/debian.cnf'
```

Changed in version 2014.1.0: 'charset' connection argument added. This is a MySQL charset, not a python one.

Changed in version 0.16.2: Connection arguments from the minion config file can be overridden on the CLI by using the arguments defined `here`. Additionally, it is now possible to setup a user with no password.

`salt.modules.mysql.`**`alter_db`**`(`*name, character_set=None, collate=None, **connection_args*`)`

Modify database using `ALTER DATABASE %(dbname)s CHARACTER SET %(charset)s COLLATE %(collation)s;` query.

CLI Example:

```
salt '*' mysql.alter_db testdb charset='latin1'
```

`salt.modules.mysql.`**`db_check`**`(`*name, table=None, **connection_args*`)`

Repairs the full database or just a given table

CLI Example:

```
salt '*' mysql.db_check dbname
salt '*' mysql.db_check dbname dbtable
```

`salt.modules.mysql.`**`db_create`**(*name*, *character_set=None*, *collate=None*, ***connection_args*)

Adds a databases to the MySQL server.

name The name of the database to manage

character_set The character set, if left empty the MySQL default will be used

collate The collation, if left empty the MySQL default will be used

CLI Example:

```
salt '*' mysql.db_create 'dbname'
salt '*' mysql.db_create 'dbname' 'utf8' 'utf8_general_ci'
```

`salt.modules.mysql.`**`db_exists`**(*name*, ***connection_args*)

Checks if a database exists on the MySQL server.

CLI Example:

```
salt '*' mysql.db_exists 'dbname'
```

`salt.modules.mysql.`**`db_get`**(*name*, ***connection_args*)

Return a list of databases of a MySQL server using the output from the `SELECT DEFAULT_CHARACTER_SET_NAME, DEFAULT_COLLATION_NAME FROM INFORMATION_SCHEMA.SCHEMATA WHERE SCHEMA_NAME='dbname';` query.

CLI Example:

```
salt '*' mysql.db_get test
```

`salt.modules.mysql.`**`db_list`**(***connection_args*)

Return a list of databases of a MySQL server using the output from the `SHOW DATABASES` query.

CLI Example:

```
salt '*' mysql.db_list
```

`salt.modules.mysql.`**`db_optimize`**(*name*, *table=None*, ***connection_args*)

Optimizes the full database or just a given table

CLI Example:

```
salt '*' mysql.db_optimize dbname
```

`salt.modules.mysql.`**`db_remove`**(*name*, ***connection_args*)

Removes a databases from the MySQL server.

CLI Example:

```
salt '*' mysql.db_remove 'dbname'
```

`salt.modules.mysql.`**`db_repair`**(*name*, *table=None*, ***connection_args*)

Repairs the full database or just a given table

CLI Example:

```
salt '*' mysql.db_repair dbname
```

`salt.modules.mysql.`**`db_tables`**(*name*, ***connection_args*)

Shows the tables in the given MySQL database (if exists)

CLI Example:

```
salt '*' mysql.db_tables 'database'
```

`salt.modules.mysql.`**`file_query`**`(`*database, file_name, **connection_args*`)`

Run an arbitrary SQL query from the specified file and return the the number of affected rows.

New in version 2017.7.0.

database
    database to run script inside

file_name
    File name of the script. This can be on the minion, or a file that is reachable by the fileserver

CLI Example:

```
salt '*' mysql.file_query mydb file_name=/tmp/sqlfile.sql
salt '*' mysql.file_query mydb file_name=salt://sqlfile.sql
```

Return data:

```
{'query time': {'human': '39.0ms', 'raw': '0.03899'}, 'rows affected': 1L}
```

`salt.modules.mysql.`**`free_slave`**`(`***connection_args*`)`

Frees a slave from its master. This is a WIP, do not use.

CLI Example:

```
salt '*' mysql.free_slave
```

`salt.modules.mysql.`**`get_master_status`**`(`***connection_args*`)`

Retrieves the master status from the minion.

Returns:

```
{'host.domain.com': {'Binlog_Do_DB': '',
                     'Binlog_Ignore_DB': '',
                     'File': 'mysql-bin.000021',
                     'Position': 107}}
```

CLI Example:

```
salt '*' mysql.get_master_status
```

`salt.modules.mysql.`**`get_slave_status`**`(`***connection_args*`)`

Retrieves the slave status from the minion.

Returns:

```
{'host.domain.com': {'Connect_Retry': 60,
                   'Exec_Master_Log_Pos': 107,
                   'Last_Errno': 0,
                   'Last_Error': '',
                   'Last_IO_Errno': 0,
                   'Last_IO_Error': '',
                   'Last_SQL_Errno': 0,
                   'Last_SQL_Error': '',
                   'Master_Host': 'comet.scion-eng.com',
                   'Master_Log_File': 'mysql-bin.000021',
                   'Master_Port': 3306,
                   'Master_SSL_Allowed': 'No',
```

(continues on next page)

(continued from previous page)

```
                    'Master_SSL_CA_File': '',
                    'Master_SSL_CA_Path': '',
                    'Master_SSL_Cert': '',
                    'Master_SSL_Cipher': '',
                    'Master_SSL_Key': '',
                    'Master_SSL_Verify_Server_Cert': 'No',
                    'Master_Server_Id': 1,
                    'Master_User': 'replu',
                    'Read_Master_Log_Pos': 107,
                    'Relay_Log_File': 'klo-relay-bin.000071',
                    'Relay_Log_Pos': 253,
                    'Relay_Log_Space': 553,
                    'Relay_Master_Log_File': 'mysql-bin.000021',
                    'Replicate_Do_DB': '',
                    'Replicate_Do_Table': '',
                    'Replicate_Ignore_DB': '',
                    'Replicate_Ignore_Server_Ids': '',
                    'Replicate_Ignore_Table': '',
                    'Replicate_Wild_Do_Table': '',
                    'Replicate_Wild_Ignore_Table': '',
                    'Seconds_Behind_Master': 0,
                    'Skip_Counter': 0,
                    'Slave_IO_Running': 'Yes',
                    'Slave_IO_State': 'Waiting for master to send event',
                    'Slave_SQL_Running': 'Yes',
                    'Until_Condition': 'None',
                    'Until_Log_File': '',
                    'Until_Log_Pos': 0}}
```

CLI Example:

```
salt '*' mysql.get_slave_status
```

`salt.modules.mysql.`**`grant_add`**(*grant*, *database*, *user*, *host='localhost'*, *grant_option=False*, *escape=True*, *ssl_option=False*, ***connection_args*)

Adds a grant to the MySQL server.

For database, make sure you specify database.table or database.*

CLI Example:

```
salt '*' mysql.grant_add             'SELECT,INSERT,UPDATE,...' 'database.*'
↪'frank' 'localhost'
```

`salt.modules.mysql.`**`grant_exists`**(*grant*, *database*, *user*, *host='localhost'*, *grant_option=False*, *escape=True*, ***connection_args*)

Checks to see if a grant exists in the database

CLI Example:

```
salt '*' mysql.grant_exists             'SELECT,INSERT,UPDATE,...' 'database.*'
↪'frank' 'localhost'
```

`salt.modules.mysql.`**`grant_revoke`**(*grant*, *database*, *user*, *host='localhost'*, *grant_option=False*, *escape=True*, ***connection_args*)

Removes a grant from the MySQL server.

CLI Example:

```
salt '*' mysql.grant_revoke            'SELECT,INSERT,UPDATE' 'database.*' 'frank
↪' 'localhost'
```

`salt.modules.mysql.`**`plugin_add`**`(`*name, soname=None, **connection_args*`)`

Add a plugina.

CLI Example:

```
salt '*' mysql.plugin_add auth_socket
```

`salt.modules.mysql.`**`plugin_remove`**`(`*name, **connection_args*`)`

Remove a plugin.

CLI Example:

```
salt '*' mysql.plugin_remove auth_socket
```

`salt.modules.mysql.`**`plugin_status`**`(`*name, **connection_args*`)`

Return the status of a plugin.

CLI Example:

```
salt '*' mysql.plugin_status auth_socket
```

`salt.modules.mysql.`**`plugins_list`**`(`***connection_args*`)`

Return a list of plugins and their status from the `SHOW PLUGINS` query.

CLI Example:

```
salt '*' mysql.plugins_list
```

`salt.modules.mysql.`**`processlist`**`(`***connection_args*`)`

Retrieves the processlist from the MySQL server via "SHOW FULL PROCESSLIST".

Returns: a list of dicts, with each dict representing a process:

```
{'Command': 'Query',
'Host': 'localhost',
'Id': 39,
'Info': 'SHOW FULL PROCESSLIST',
'Rows_examined': 0,
'Rows_read': 1,
'Rows_sent': 0,
'State': None,
'Time': 0,
'User': 'root',
'db': 'mysql'}
```

CLI Example:

```
salt '*' mysql.processlist
```

`salt.modules.mysql.`**`query`**`(`*database, query, **connection_args*`)`

Run an arbitrary SQL query and return the results or the number of affected rows.

CLI Example:

```
salt '*' mysql.query mydb "UPDATE mytable set myfield=1 limit 1"
```

Return data:

```
{'query time': {'human': '39.0ms', 'raw': '0.03899'}, 'rows affected': 1L}
```

CLI Example:

```
salt '*' mysql.query mydb "SELECT id,name,cash from users limit 3"
```

Return data:

```
{'columns': ('id', 'name', 'cash'),
    'query time': {'human': '1.0ms', 'raw': '0.001'},
    'results': ((1L, 'User 1', Decimal('110.000000')),
                (2L, 'User 2', Decimal('215.636756')),
                (3L, 'User 3', Decimal('0.040000'))),
    'rows returned': 3L}
```

CLI Example:

```
salt '*' mysql.query mydb 'INSERT into users values (null,"user 4", 5)'
```

Return data:

```
{'query time': {'human': '25.6ms', 'raw': '0.02563'}, 'rows affected': 1L}
```

CLI Example:

```
salt '*' mysql.query mydb 'DELETE from users where id = 4 limit 1'
```

Return data:

```
{'query time': {'human': '39.0ms', 'raw': '0.03899'}, 'rows affected': 1L}
```

Jinja Example: Run a query on `mydb` and use row 0, column 0's data.

```
{{ salt['mysql.query']('mydb', 'SELECT info from mytable limit 1')['results
↪'][0][0] }}
```

`salt.modules.mysql.`**`quote_identifier`**(*identifier*, *for_grants=False*)

Return an identifier name (column, table, database, etc) escaped for MySQL

This means surrounded by ”‘” character and escaping this character inside. It also means doubling the ’%’ character for MySQLdb internal usage.

Parameters

- **`identifier`** – the table, column or database identifier
- **`for_grants`** – is False by default, when using database names on grant queries you should set it to True to also escape ”_” and ”%” characters as requested by MySQL. Note that theses characters should only be escaped when requesting grants on the database level (my_%db.*) but not for table level grants (my_%db.‘foo‘)

CLI Example:

```
salt '*' mysql.quote_identifier 'foo`bar'
```

`salt.modules.mysql.`**`showglobal`**(***connection_args*)

Retrieves the show global variables from the minion.

Returns:: show global variables full dict

CLI Example:

```
salt '*' mysql.showglobal
```

`salt.modules.mysql.`**`showvariables`**`(`***connection_args*`)`

Retrieves the show variables from the minion.

Returns:: show variables full dict

CLI Example:

```
salt '*' mysql.showvariables
```

`salt.modules.mysql.`**`slave_lag`**`(`***connection_args*`)`

Return the number of seconds that a slave SQL server is lagging behind the master, if the host is not a slave it will return -1. If the server is configured to be a slave for replication but slave IO is not running then -2 will be returned. If there was an error connecting to the database or checking the slave status, -3 will be returned.

CLI Example:

```
salt '*' mysql.slave_lag
```

`salt.modules.mysql.`**`status`**`(`***connection_args*`)`

Return the status of a MySQL server using the output from the `SHOW STATUS` query.

CLI Example:

```
salt '*' mysql.status
```

`salt.modules.mysql.`**`tokenize_grant`**`(`*grant*`)`

External wrapper function :param grant: :return: dict

CLI Example:

```
salt '*' mysql.tokenize_grant             "GRANT SELECT, INSERT ON testdb.* TO
↪'testuser'@'localhost'"
```

`salt.modules.mysql.`**`user_chpass`**`(`*user, host='localhost', password=None, password_hash=None, allow_passwordless=False, unix_socket=None, password_column=None, **connection_args*`)`

Change password for a MySQL user

host Host for which this user/password combo applies

password The password to set for the new user. Will take precedence over the `password_hash` option if both are specified.

password_hash The password in hashed form. Be sure to quote the password because YAML doesn't like the `*`. A password hash can be obtained from the mysql command-line client like so:

```
mysql> SELECT PASSWORD('mypass');
+-------------------------------------------+
| PASSWORD('mypass')                        |
+-------------------------------------------+
| *6C8989366EAF75BB670AD8EA7A7FC1176A95CEF4 |
+-------------------------------------------+
1 row in set (0.00 sec)
```

allow_passwordless If `True`, then `password` and `password_hash` can be omitted (or set to `None`) to permit a passwordless login.

New in version 0.16.2: The `allow_passwordless` option was added.

CLI Examples:

```
salt '*' mysql.user_chpass frank localhost newpassword
salt '*' mysql.user_chpass frank localhost password_hash='hash'
salt '*' mysql.user_chpass frank localhost allow_passwordless=True
```

`salt.modules.mysql.`**`user_create`**(*user, host='localhost', password=None, password_hash=None, allow_passwordless=False, unix_socket=False, password_column=None, auth_plugin='mysql_native_password', **connection_args*)

Creates a MySQL user

host Host for which this user/password combo applies

password The password to use for the new user. Will take precedence over the `password_hash` option if both are specified.

password_hash The password in hashed form. Be sure to quote the password because YAML doesn't like the `*`. A password hash can be obtained from the mysql command-line client like so:

```
mysql> SELECT PASSWORD('mypass');
+-------------------------------------------+
| PASSWORD('mypass')                        |
+-------------------------------------------+
| *6C8989366EAF75BB670AD8EA7A7FC1176A95CEF4 |
+-------------------------------------------+
1 row in set (0.00 sec)
```

allow_passwordless If `True`, then `password` and `password_hash` can be omitted (or set to `None`) to permit a passwordless login.

unix_socket If `True` and allow_passwordless is `True` then will be used unix_socket auth plugin.

password_column The password column to use in the user table.

auth_plugin The authentication plugin to use, default is to use the mysql_native_password plugin.

New in version 0.16.2: The `allow_passwordless` option was added.

CLI Examples:

```
salt '*' mysql.user_create 'username' 'hostname' 'password'
salt '*' mysql.user_create 'username' 'hostname' password_hash='hash'
salt '*' mysql.user_create 'username' 'hostname' allow_passwordless=True
```

`salt.modules.mysql.`**`user_exists`**(*user, host='localhost', password=None, password_hash=None, passwordless=False, unix_socket=False, password_column=None, **connection_args*)

Checks if a user exists on the MySQL server. A login can be checked to see if passwordless login is permitted by omitting `password` and `password_hash`, and using `passwordless=True`.

New in version 0.16.2: The `passwordless` option was added.

CLI Example:

```
salt '*' mysql.user_exists 'username' 'hostname' 'password'
salt '*' mysql.user_exists 'username' 'hostname' password_hash='hash'
salt '*' mysql.user_exists 'username' passwordless=True
salt '*' mysql.user_exists 'username' password_column='authentication_string'
```

`salt.modules.mysql.`**`user_grants`**(*user, host='localhost', **connection_args*)

Shows the grants for the given MySQL user (if it exists)

CLI Example:

```
salt '*' mysql.user_grants 'frank' 'localhost'
```

`salt.modules.mysql.`**`user_info`**`(`*user, host='localhost', **connection_args*`)`

Get full info on a MySQL user

CLI Example:

```
salt '*' mysql.user_info root localhost
```

`salt.modules.mysql.`**`user_list`**`(`***connection_args*`)`

Return a list of users on a MySQL server

CLI Example:

```
salt '*' mysql.user_list
```

`salt.modules.mysql.`**`user_remove`**`(`*user, host='localhost', **connection_args*`)`

Delete MySQL user

CLI Example:

```
salt '*' mysql.user_remove frank localhost
```

`salt.modules.mysql.`**`verify_login`**`(`*user, password=None, **connection_args*`)`

Attempt to login using the provided credentials. If successful, return true. Otherwise, return False.

CLI Example:

```
salt '*' mysql.verify_login root password
```

`salt.modules.mysql.`**`version`**`(`***connection_args*`)`

Return the version of a MySQL server using the output from the `SELECT VERSION()` query.

CLI Example:

```
salt '*' mysql.version
```

21.9.255 salt.modules.nacl

This module helps include encrypted passwords in pillars, grains and salt state files.

depends libnacl, https://github.com/saltstack/libnacl

This is often useful if you wish to store your pillars in source control or share your pillar data with others that you trust. I don't advise making your pillars public regardless if they are encrypted or not.

When generating keys and encrypting passwords use –local when using salt-call for extra security. Also consider using just the salt runner nacl when encrypting pillar passwords.

configuration The following configuration defaults can be define (pillar or config files) Avoid storing private keys in pillars! Ensure master does not have *pillar_opts=True*:

```
# cat /etc/salt/master.d/nacl.conf
nacl.config:
    # NOTE: `key` and `key_file` have been renamed to `sk`, `sk_file`
    # also `box_type` default changed from secretbox to sealedbox.
    box_type: sealedbox                     (default)
    sk_file: /etc/salt/pki/master/nacl      (default)
    pk_file: /etc/salt/pki/master/nacl.pub  (default)
    sk: None
    pk: None
```

Usage can override the config defaults:

```
salt-call nacl.enc sk_file=/etc/salt/pki/master/nacl pk_file=/etc/salt/
↪pki/master/nacl.pub
```

The nacl lib uses 32byte keys, these keys are base64 encoded to make your life more simple. To generate your *sk_file* and *pk_file* use:

```
salt-call --local nacl.keygen sk_file=/etc/salt/pki/master/nacl
# or if you want to work without files.
salt-call --local nacl.keygen
local:
    ----------
    pk:
        /kfGX7PbWeu099702PBbKWLpG/9p06IQRswkdWHCDk0=
    sk:
        SVWut5SqNpuPeNzb1b9y6b2eXg2PLIog43GBzp48Sow=
```

Now with your keypair, you can encrypt data:

You have two option, *sealedbox* or *secretbox*.

SecretBox is data encrypted using private key *pk*. Sealedbox is encrypted using public key *pk*.

Recommend using Sealedbox because the one way encryption permits developers to encrypt data for source control but not decrypt. Sealedbox only has one key that is for both encryption and decryption.

```
salt-call --local nacl.enc asecretpass pk=/kfGX7PbWeu099702PBbKWLpG/9p06IQRswkdWHCDk0=
tqXzeIJnTAM9Xf0mdLcpEdklMbfBGPj2oTKmlgrm3S1DTVVHNnh9h8mU1GKllGq/+cYsk6m5WhGdk58=
```

To decrypt the data:

```
salt-call --local nacl.dec data=
↪'tqXzeIJnTAM9Xf0mdLcpEdklMbfBGPj2oTKmlgrm3S1DTVVHNnh9h8mU1GKllGq/+cYsk6m5WhGdk58='
↪       sk='SVWut5SqNpuPeNzb1b9y6b2eXg2PLIog43GBzp48Sow='
```

When the keys are defined in the master config you can use them from the nacl runner without extra parameters:

```
# cat /etc/salt/master.d/nacl.conf
nacl.config:
    sk_file: /etc/salt/pki/master/nacl
    pk: 'cTIqXwnUiD1ulg4kXsbeCE7/NoeKEzd4nLeYcCFpd9k='
```

```
salt-run nacl.enc 'asecretpass'
salt-run nacl.dec 'tqXzeIJnTAM9Xf0mdLcpEdklMbfBGPj2oTKmlgrm3S1DTVVHNnh9h8mU1GKllGq/
↪+cYsk6m5WhGdk58='
```

```
# a salt developers minion could have pillar data that includes a nacl public key
nacl.config:
    pk: '/kfGX7PbWeu099702PBbKWLpG/9p06IQRswkdWHCDk0='
```

The developer can then use a less-secure system to encrypt data.

```
salt-call --local nacl.enc apassword
```

Pillar files can include protected data that the salt master decrypts:

```
pillarexample:
    user: root
    password1: {{salt.nacl.dec('DRB7Q6/X5gGSRCTpZyxS6hlbWj0llUA+uaVyvou3vJ4=')|json}}
    cert_key: {{salt.nacl.dec_file('/srv/salt/certs/example.com/key.nacl')|json}}
    cert_key2: {{salt.nacl.dec_file('salt:///certs/example.com/key.nacl')|json}}
```

Larger files like certificates can be encrypted with:

```
salt-call nacl.enc_file /tmp/cert.crt out=/tmp/cert.nacl
# or more advanced
cert=$(cat /tmp/cert.crt)
salt-call --out=newline_values_only nacl.enc_pub data="$cert" > /tmp/cert.nacl
```

In pillars rended with jinja be sure to include *|json* so line breaks are encoded:

```
cert: "{{salt.nacl.dec('S2uogToXkgENz9...085KYt')|json}}"
```

In states rendered with jinja it is also good pratice to include *|json*:

```
{{sls}} private key:
    file.managed:
        - name: /etc/ssl/private/cert.key
        - mode: 700
        - contents: "{{pillar['pillarexample']['cert_key']|json}}"
```

Optional small program to encrypt data without needing salt modules.

```
#!/bin/python3
import sys, base64, libnacl.sealed
pk = base64.b64decode('YOURPUBKEY')
b = libnacl.sealed.SealedBox(pk)
data = sys.stdin.buffer.read()
print(base64.b64encode(b.encrypt(data)).decode())
```

```
echo 'apassword' | nacl_enc.py
```

`salt.modules.nacl.`**`dec`**(*data*, ***kwargs*)

Alias to *{box_type}_decrypt*

box_type: secretbox, sealedbox(default)

`salt.modules.nacl.`**`dec_file`**(*name*, *out=None*, ***kwargs*)

This is a helper function to decrypt a file and return its contents.

You can provide an optional output file using *out*

name can be a local file or when not using *salt-run* can be a url like *salt://*, *https://* etc.

CLI Examples:

```
salt-run nacl.dec_file name=/tmp/id_rsa.nacl
salt-call nacl.dec_file name=salt://crt/mycert.nacl out=/tmp/id_rsa
salt-run nacl.dec_file name=/tmp/id_rsa.nacl box_type=secretbox             sk_
↪file=/etc/salt/pki/master/nacl.pub
```

`salt.modules.nacl.`**`enc`**(*data*, ***kwargs*)

Alias to *{box_type}_encrypt*

box_type: secretbox, sealedbox(default)

`salt.modules.nacl.`**`enc_file`**(*name*, *out=None*, ***kwargs*)

This is a helper function to encrypt a file and return its contents.

You can provide an optional output file using *out*

name can be a local file or when not using *salt-run* can be a url like *salt://*, *https://* etc.

CLI Examples:

```
salt-run nacl.enc_file name=/tmp/id_rsa
salt-call nacl.enc_file name=salt://crt/mycert out=/tmp/cert
salt-run nacl.enc_file name=/tmp/id_rsa box_type=secretbox             sk_file=/
↪etc/salt/pki/master/nacl.pub
```

`salt.modules.nacl.`**`keygen`**(*sk_file=None*, *pk_file=None*, ***kwargs*)

Use libnacl to generate a keypair.

If no *sk_file* is defined return a keypair.

If only the *sk_file* is defined *pk_file* will use the same name with a postfix *.pub*.

When the *sk_file* is already existing, but *pk_file* is not. The *pk_file* will be generated using the *sk_file*.

CLI Examples:

```
salt-call nacl.keygen
salt-call nacl.keygen sk_file=/etc/salt/pki/master/nacl
salt-call nacl.keygen sk_file=/etc/salt/pki/master/nacl pk_file=/etc/salt/pki/
↪master/nacl.pub
salt-call --local nacl.keygen
```

`salt.modules.nacl.`**`sealedbox_decrypt`**(*data*, ***kwargs*)

Decrypt data using a secret key that was encrypted using a public key with *nacl.sealedbox_encrypt*.

CLI Examples:

```
salt-call nacl.sealedbox_decrypt pEXHQM6cuaF7A=
salt-call --local nacl.sealedbox_decrypt data='pEXHQM6cuaF7A=' sk_file=/etc/salt/
↪pki/master/nacl
salt-call --local nacl.sealedbox_decrypt data='pEXHQM6cuaF7A=' sk='YmFkcGFzcwo='
```

`salt.modules.nacl.`**`sealedbox_encrypt`**(*data*, ***kwargs*)

Encrypt data using a public key generated from *nacl.keygen*. The encryptd data can be decrypted using *nacl.sealedbox_decrypt* only with the secret key.

CLI Examples:

```
salt-run nacl.sealedbox_encrypt datatoenc
salt-call --local nacl.sealedbox_encrypt datatoenc pk_file=/etc/salt/pki/master/
↪nacl.pub
salt-call --local nacl.sealedbox_encrypt datatoenc pk=
↪'vrwQF7cNiNAVQVAiS3bvcbJUnF0cN6fU9YTZD9mBfzQ='
```

`salt.modules.nacl.`**`secretbox_decrypt`**(*data*, ***kwargs*)

Decrypt data that was encrypted using *nacl.secretbox_encrypt* using the secret key that was generated from *nacl.keygen*.

CLI Examples:

```
salt-call nacl.secretbox_decrypt pEXHQM6cuaF7A=
salt-call --local nacl.secretbox_decrypt data='pEXHQM6cuaF7A=' sk_file=/etc/salt/
↪pki/master/nacl
salt-call --local nacl.secretbox_decrypt data='pEXHQM6cuaF7A=' sk='YmFkcGFzcwo='
```

`salt.modules.nacl.`**`secretbox_encrypt`**(*data*, ***kwargs*)

Encrypt data using a secret key generated from *nacl.keygen*. The same secret key can be used to decrypt the data using *nacl.secretbox_decrypt*.

CLI Examples:

```
salt-run nacl.secretbox_encrypt datatoenc
salt-call --local nacl.secretbox_encrypt datatoenc sk_file=/etc/salt/pki/master/
↪nacl
salt-call --local nacl.secretbox_encrypt datatoenc sk='YmFkcGFzcwo='
```

21.9.256 salt.modules.nagios

Run nagios plugins/checks from salt and get the return as data.

`salt.modules.nagios.`**`list_plugins`**()

List all the nagios plugins

CLI Example:

```
salt '*' nagios.list_plugins
```

`salt.modules.nagios.`**`retcode`**(*plugin*, *args=''*, *key_name=None*)

Run one nagios plugin and return retcode of the execution

`salt.modules.nagios.`**`retcode_pillar`**(*pillar_name*)

Run one or more nagios plugins from pillar data and get the result of cmd.retcode The pillar have to be in this format:

```
------
webserver:
    Ping_google:
        - check_icmp: 8.8.8.8
        - check_icmp: google.com
    Load:
        - check_load: -w 0.8 -c 1
    APT:
        - check_apt
-------
```

webserver is the role to check, the next keys are the group and the items the check with the arguments if needed

You must to group different checks(one o more) and always it will return the highest value of all the checks

CLI Example:

```
salt '*' nagios.retcode webserver
```

`salt.modules.nagios.`**`run`**(*plugin*, *args=''*)

Run nagios plugin and return all the data execution with cmd.run

CLI Example:

```
salt '*' nagios.run check_apt
salt '*' nagios.run check_icmp '8.8.8.8'
```

salt.modules.nagios.**run_all**(*plugin*, *args=''*)
Run nagios plugin and return all the data execution with cmd.run_all

salt.modules.nagios.**run_all_pillar**(*pillar_name*)
Run one or more nagios plugins from pillar data and get the result of cmd.run_all The pillar have to be in this format:

```
------
webserver:
    Ping_google:
        - check_icmp: 8.8.8.8
        - check_icmp: google.com
    Load:
        - check_load: -w 0.8 -c 1
    APT:
        - check_apt
-------
```

webserver is the role to check, the next keys are the group and the items the check with the arguments if needed

You have to group different checks in a group

CLI Example:

```
salt '*' nagios.run webserver
```

salt.modules.nagios.**run_pillar**(*pillar_name*)
Run one or more nagios plugins from pillar data and get the result of cmd.run The pillar have to be in this format:

```
------
webserver:
    Ping_google:
        - check_icmp: 8.8.8.8
        - check_icmp: google.com
    Load:
        - check_load: -w 0.8 -c 1
    APT:
        - check_apt
-------
```

webserver is the role to check, the next keys are the group and the items the check with the arguments if needed

You have to group different checks in a group

CLI Example:

```
salt '*' nagios.run webserver
```

21.9.257 salt.modules.nagios_rpc

Check Host & Service status from Nagios via JSON RPC.

New in version 2015.8.0.

`salt.modules.nagios_rpc.`**`host_status`**`(`*hostname=None*, ***kwargs*`)`

Check status of a particular host By default statuses are returned in a numeric format.

Parameters:

hostname The hostname to check the status of the service in Nagios.

numeric Turn to false in order to return status in text format ('OK' instead of 0, 'Warning' instead of 1 etc)

Returns status: 'OK', 'Warning', 'Critical' or 'Unknown'

CLI Example:

```
salt '*' nagios_rpc.host_status hostname=webserver.domain.com
salt '*' nagios_rpc.host_status hostname=webserver.domain.com numeric=False
```

`salt.modules.nagios_rpc.`**`service_status`**`(`*hostname=None*, *service=None*, ***kwargs*`)`

Check status of a particular service on a host on it in Nagios. By default statuses are returned in a numeric format.

Parameters:

hostname The hostname to check the status of the service in Nagios.

service The service to check the status of in Nagios.

numeric Turn to false in order to return status in text format ('OK' instead of 0, 'Warning' instead of 1 etc)

Returns status: 'OK', 'Warning', 'Critical' or 'Unknown'

CLI Example:

```
salt '*' nagios_rpc.service_status hostname=webserver.domain.com service='HTTP'
salt '*' nagios_rpc.service_status hostname=webserver.domain.com service='HTTP'
↪numeric=False
```

21.9.258 salt.modules.namecheap_domains_dns

Namecheap DNS Management

New in version 2017.7.0.

Prerequisites

This module uses the `requests` Python module to communicate to the namecheap API.

Configuration

The Namecheap username, API key and URL should be set in the minion configuration file, or in the Pillar data.

```
namecheap.name: companyname
namecheap.key: a1b2c3d4e5f67a8b9c0d1e2f3
namecheap.client_ip: 162.155.30.172
#Real url
namecheap.url: https://api.namecheap.com/xml.response
#Sandbox url
#namecheap.url: https://api.sandbox.namecheap.xml.response
```

`salt.modules.namecheap_domains_dns.`**`get_hosts`**(*sld*, *tld*)

Retrieves DNS host record settings for the requested domain.

returns a dictionary of information about the requested domain
sld SLD of the domain name
tld TLD of the domain name
CLI Example:

```
salt 'my-minion' namecheap_domains_dns.get_hosts sld tld
```

`salt.modules.namecheap_domains_dns.`**`get_list`**(*sld*, *tld*)

Gets a list of DNS servers associated with the requested domain.

returns a dictionary of information about requested domain
sld SLD of the domain name
tld TLD of the domain name
CLI Example:

```
salt 'my-minion' namecheap_domains_dns.get_list sld tld
```

`salt.modules.namecheap_domains_dns.`**`set_custom`**(*sld*, *tld*, *nameservers*)

Sets domain to use custom DNS servers.

returns True if the custom nameservers were set successfully
sld SLD of the domain name
tld TLD of the domain name
nameservers array of strings List of nameservers to be associated with this domain
CLI Example:

```
salt 'my-minion' namecheap_domains_dns.set_custom sld tld nameserver
```

`salt.modules.namecheap_domains_dns.`**`set_default`**(*sld*, *tld*)

Sets domain to use namecheap default DNS servers. Required for free services like Host record management, URL forwarding, email forwarding, dynamic DNS and other value added services.
sld SLD of the domain name
tld TLD of the domain name
Returns `True` if the domain was successfully pointed at the default DNS servers.

CLI Example:

```
salt 'my-minion' namecheap_domains_dns.set_default sld tld
```

`salt.modules.namecheap_domains_dns.`**`set_hosts`**(*sld*, *tld*, *hosts*)

Sets DNS host records settings for the requested domain.

returns True if the host records were set successfully
sld SLD of the domain name
tld TLD of the domain name
hosts Must be passed as a list of Python dictionaries, with each dictionary containing the following keys:

- **hostname**
- **recordtype** - One of `A`, `AAAA`, `CNAME`, `MX`, `MXE`, `TXT`, `URL`, `URL301`, or `FRAME`
- **address** - URL or IP address
- **ttl** - An integer between 60 and 60000 (default: `1800`)

Additonally, the `mxpref` key can be present, but must be accompanied by an `emailtype` key.

CLI Example:

```
salt 'my-minion' namecheap_domains_dns.set_hosts sld tld hosts
```

21.9.259 salt.modules.namecheap_domains module

Namecheap Domain Management

New in version 2017.7.0.

Prerequisites

This module uses the `requests` Python module to communicate to the namecheap API.

Configuration

The Namecheap username, API key and URL should be set in the minion configuration file, or in the Pillar data.

```
namecheap.name: companyname
namecheap.key: a1b2c3d4e5f67a8b9c0d1e2f3
namecheap.client_ip: 162.155.30.172
#Real url
namecheap.url: https://api.namecheap.com/xml.response
#Sandbox url
#namecheap.url: https://api.sandbox.namecheap.xml.response
```

`salt.modules.namecheap_domains.`**`check`**(**domains_to_check*)

Checks the availability of domains

domains_to_check array of strings List of domains to check

Returns a dictionary mapping the each domain name to a boolean denoting whether or not it is available.

CLI Example:

```
salt 'my-minion' namecheap_domains.check domain-to-check
```

`salt.modules.namecheap_domains.`**`create`**(*domain_name, years, **kwargs*)

Try to register the specified domain name

domain_name The domain name to be registered

years Number of years to register

Returns the following information:

- Whether or not the domain was renewed successfully
- Whether or not WhoisGuard is enabled
- Whether or not registration is instant
- The amount charged for registration
- The domain ID
- The order ID
- The transaction ID

CLI Example:

```
salt 'my-minion' namecheap_domains.create my-domain-name 2
```

`salt.modules.namecheap_domains.`**`get_info`**(*domain_name*)

Returns information about the requested domain

returns a dictionary of information about the domain_name

domain_name string Domain name to get information about

CLI Example:

```
salt 'my-minion' namecheap_domains.get_info my-domain-name
```

`salt.modules.namecheap_domains.`**`get_list`**`(`*list_type=None*, *search_term=None*, *page=None*, *page_size=None*, *sort_by=None*`)`

Returns a list of domains for the particular user as a list of objects offset by `page` length of `page_size`

list_type [ALL] One of `ALL`, `EXPIRING`, `EXPIRED`

search_term Keyword to look for on the domain list

page [1] Number of result page to return

page_size [20] Number of domains to be listed per page (minimum: `10`, maximum: `100`)

sort_by One of `NAME`, `NAME_DESC`, `EXPIREDATE`, `EXPIREDATE_DESC`, `CREATEDATE`, or `CREATEDATE_DESC`

CLI Example:

```
salt 'my-minion' namecheap_domains.get_list
```

`salt.modules.namecheap_domains.`**`get_tld_list`**`()`

Returns a list of TLDs as objects

CLI Example:

```
salt 'my-minion' namecheap_domains.get_tld_list
```

`salt.modules.namecheap_domains.`**`reactivate`**`(`*domain_name*`)`

Try to reactivate the expired domain name

Returns the following information:

- Whether or not the domain was reactivated successfully
- The amount charged for reactivation
- The order ID
- The transaction ID

CLI Example:

```
salt 'my-minion' namecheap_domains.reactivate my-domain-name
```

`salt.modules.namecheap_domains.`**`renew`**`(`*domain_name*, *years*, *promotion_code=None*`)`

Try to renew the specified expiring domain name for a specified number of years

domain_name The domain name to be renewed

years Number of years to renew

Returns the following information:

- Whether or not the domain was renewed successfully
- The domain ID
- The order ID
- The transaction ID
- The amount charged for renewal

CLI Example:

```
salt 'my-minion' namecheap_domains.renew my-domain-name 5
```

21.9.260 salt.modules.namecheap_domains_ns

Namecheap Nameserver Management

New in version 2017.7.0.

Prerequisites

This module uses the `requests` Python module to communicate to the namecheap API.

Configuration

The Namecheap username, API key and URL should be set in the minion configuration file, or in the Pillar data.

```
namecheap.name: companyname
namecheap.key: a1b2c3d4e5f67a8b9c0d1e2f3
namecheap.client_ip: 162.155.30.172
#Real url
namecheap.url: https://api.namecheap.com/xml.response
#Sandbox url
#namecheap.url: https://api.sandbox.namecheap.xml.response
```

`salt.modules.namecheap_domains_ns.`**`create`**(*sld*, *tld*, *nameserver*, *ip*)

Creates a new nameserver. Returns `True` if the nameserver was created successfully.

sld SLD of the domain name

tld TLD of the domain name

nameserver Nameserver to create

ip Nameserver IP address

CLI Example:

```
salt '*' namecheap_domains_ns.create sld tld nameserver ip
```

`salt.modules.namecheap_domains_ns.`**`delete`**(*sld*, *tld*, *nameserver*)

Deletes a nameserver. Returns `True` if the nameserver was deleted successfully

sld SLD of the domain name

tld TLD of the domain name

nameserver Nameserver to delete

CLI Example:

```
salt '*' namecheap_domains_ns.delete sld tld nameserver
```

`salt.modules.namecheap_domains_ns.`**`get_info`**(*sld*, *tld*, *nameserver*)

Retrieves information about a registered nameserver. Returns the following information:

- IP Address set for the nameserver
- Domain name which was queried
- A list of nameservers and their statuses

sld SLD of the domain name

tld TLD of the domain name

nameserver Nameserver to retrieve

CLI Example:

```
salt '*' namecheap_domains_ns.get_info sld tld nameserver
```

`salt.modules.namecheap_domains_ns.`**`update`**(*sld*, *tld*, *nameserver*, *old_ip*, *new_ip*)

Deletes a nameserver. Returns `True` if the nameserver was updated successfully.

sld SLD of the domain name

tld TLD of the domain name

nameserver Nameserver to create

old_ip Current ip address

new_ip New ip address

CLI Example:

```
salt '*' namecheap_domains_ns.update sld tld nameserver old_ip new_ip
```

21.9.261 salt.modules.namecheap_ssl module

Namecheap SSL Certificate Management

New in version 2017.7.0.

Prerequisites

This module uses the `requests` Python module to communicate to the namecheap API.

Configuration

The Namecheap username, API key and URL should be set in the minion configuration file, or in the Pillar data.

```
namecheap.name: companyname
namecheap.key: a1b2c3d4e5f67a8b9c0d1e2f3
namecheap.client_ip: 162.155.30.172
#Real url
namecheap.url: https://api.namecheap.com/xml.response
#Sandbox url
#namecheap.url: https://api.sandbox.namecheap.xml.response
```

`salt.modules.namecheap_ssl.`**`activate`**(*csr_file, certificate_id, web_server_type, approver_email=None, http_dc_validation=False, **kwargs*)

Activates a newly-purchased SSL certificate. Returns a dictionary of result values.

csr_file Path to Certificate Signing Request file

certificate_id Unique ID of the SSL certificate you wish to activate

web_server_type The type of certificate format to return. Possible values include:

- apache2
- apacheapachessl
- apacheopenssl
- apacheraven
- apachessl
- apachessleay
- c2net
- cobaltseries
- cpanel
- domino
- dominogo4625
- dominogo4626
- ensim
- hsphere
- ibmhttp
- iis
- iis4
- iis5
- iplanet
- ipswitch

- netscape
- other
- plesk
- tomcat
- weblogic
- website
- webstar
- zeusv3

approver_email The email ID which is on the approver email list.

> **Note:** `http_dc_validation` must be set to `False` if this option is used.

http_dc_validation [False] Whether or not to activate using HTTP-based validation.

Note: For other parameters which may be required, see here.

CLI Example:

```
salt 'my-minion' namecheap_ssl.activate my-csr-file my-cert-id apachessl
```

`salt.modules.namecheap_ssl.`**`create`**`(`*years*, *certificate_type*, *promotion_code=None*, *sans_to_add=None*`)`

Creates a new SSL certificate. Returns the following information:

- Whether or not the SSL order was successful
- The certificate ID
- The order ID
- The transaction ID
- The amount charged for the order
- The date on which the certificate was created
- The date on which the certificate will expire
- The type of SSL certificate
- The number of years for which the certificate was purchased
- The current status of the SSL certificate

years [1] Number of years to register

certificate_type Type of SSL Certificate. Possible values include:

- EV Multi Domain SSL
- EV SSL
- EV SSL SGC
- EssentialSSL
- EssentialSSL Wildcard
- InstantSSL
- InstantSSL Pro
- Multi Domain SSL
- PositiveSSL
- PositiveSSL Multi Domain
- PositiveSSL Wildcard
- PremiumSSL
- PremiumSSL Wildcard
- QuickSSL Premium
- RapidSSL
- RapidSSL Wildcard
- SGC Supercert
- SSL Web Server

- SSL Webserver EV
- SSL123
- Secure Site
- Secure Site Pro
- Secure Site Pro with EV
- Secure Site with EV
- True BusinessID
- True BusinessID Multi Domain
- True BusinessID Wildcard
- True BusinessID with EV
- True BusinessID with EV Multi Domain
- Unified Communications

promotional_code An optional promo code to use when creating the certificate

sans_to_add [0] This parameter defines the number of add-on domains to be purchased in addition to the default number of domains included with a multi-domain certificate. Each certificate that supports SANs has the default number of domains included. You may check the default number of domains included and the maximum number of domains that can be added to it in the table below.

Provider	Product name	Default number of domains (domain from CSR is counted here)	Maximum number of total domains	Maximum number of domains that can be passed in sans_to_add parameter
Comodo	PositiveSSL Multi-Domain	3	100	97
Comodo	Multi-Domain SSL	3	100	97
Comodo	EV Multi- Domain SSL	3	100	97
Comodo	Unified Communications	3	100	97
GeoTrust	QuickSSL Premium	1	1 domain + 4 subdomains	The only supported value is 4
GeoTrust	True BusinessID with EV Multi-Domain	5	25	20
GeoTrust	True Business ID Multi- Domain	5	25	20
Thawte	SSL Web Server	1	25	24
Thawte	SSL Web Server with EV	1	25	24
Thawte	SGC Supercerts	1	25	24
Symantec	Secure Site Pro with EV	1	25	24
Symantec	Secure Site with EV	1	25	24
Symantec	Secure Site	1	25	24
Symantec	Secure Site Pro	1	25	24

CLI Example:

```
salt 'my-minion' namecheap_ssl.create 2 RapidSSL
```

`salt.modules.namecheap_ssl.`**`get_info`**`(`*certificate_id*, *returncertificate=False*, *returntype=None*`)`

Retrieves information about the requested SSL certificate. Returns a dictionary of information about the SSL certificate with two keys:

- **ssl** - Contains the metadata information
- **certificate** - Contains the details for the certificate such as the CSR, Approver, and certificate data

certificate_id Unique ID of the SSL certificate

returncertificate [False] Set to `True` to ask for the certificate in response

returntype Optional type for the returned certificate. Can be either "Individual" (for X.509 format) or "PKCS7"

> **Note:** Required if `returncertificate` is `True`

CLI Example:

```
salt 'my-minion' namecheap_ssl.get_info my-cert-id
```

`salt.modules.namecheap_ssl.`**`get_list`**`(`***kwargs*`)`

Returns a list of SSL certificates for a particular user

ListType [All] Possible values:

- All
- Processing
- EmailSent
- TechnicalProblem
- InProgress
- Completed
- Deactivated
- Active
- Cancelled
- NewPurchase
- NewRenewal

SearchTerm Keyword to look for on the SSL list

Page [1] Page number to return

PageSize [20] Total number of SSL certificates to display per page (minimum: `10`, maximum: `100`)

SoryBy One of `PURCHASEDATE`, `PURCHASEDATE_DESC`, `SSLTYPE`, `SSLTYPE_DESC`, `EXPIREDATETIME`, `EXPIREDATETIME_DESC`, `Host_Name`, or `Host_Name_DESC`

CLI Example:

```
salt 'my-minion' namecheap_ssl.get_list Processing
```

`salt.modules.namecheap_ssl.`**`parse_csr`**`(`*csr_file*, *certificate_type*, *http_dc_validation=False*`)`

Parses the CSR. Returns a dictionary of result values.

csr_file Path to Certificate Signing Request file

certificate_type Type of SSL Certificate. Possible values include:

- EV Multi Domain SSL
- EV SSL
- EV SSL SGC
- EssentialSSL
- EssentialSSL Wildcard
- InstantSSL
- InstantSSL Pro
- Multi Domain SSL
- PositiveSSL
- PositiveSSL Multi Domain
- PositiveSSL Wildcard

- PremiumSSL
- PremiumSSL Wildcard
- QuickSSL Premium
- RapidSSL
- RapidSSL Wildcard
- SGC Supercert
- SSL Web Server
- SSL Webserver EV
- SSL123
- Secure Site
- Secure Site Pro
- Secure Site Pro with EV
- Secure Site with EV
- True BusinessID
- True BusinessID Multi Domain
- True BusinessID Wildcard
- True BusinessID with EV
- True BusinessID with EV Multi Domain
- Unified Communications

http_dc_validation [False] Set to `True` if a Comodo certificate and validation should be done with files instead of emails and to return the info to do so

CLI Example:

```
salt 'my-minion' namecheap_ssl.parse_csr my-csr-file PremiumSSL
```

`salt.modules.namecheap_ssl.`**`reissue`**(*csr_file*, *certificate_id*, *web_server_type*, *approver_email=None*, *http_dc_validation=False*, ***kwargs*)

Reissues a purchased SSL certificate. Returns a dictionary of result values.

csr_file Path to Certificate Signing Request file

certificate_id Unique ID of the SSL certificate you wish to activate

web_server_type The type of certificate format to return. Possible values include:

- apache2
- apacheapachessl
- apacheopenssl
- apacheraven
- apachessl
- apachessleay
- c2net
- cobaltseries
- cpanel
- domino
- dominogo4625
- dominogo4626
- ensim
- hsphere
- ibmhttp
- iis
- iis4
- iis5
- iplanet
- ipswitch
- netscape
- other
- plesk

  - tomcat
  - weblogic
  - website
  - webstar
  - zeusv3

approver_email The email ID which is on the approver email list.

> **Note:** `http_dc_validation` must be set to `False` if this option is used.

http_dc_validation [False] Whether or not to activate using HTTP-based validation.

Note: For other parameters which may be required, see here.

CLI Example:

```
salt 'my-minion' namecheap_ssl.reissue my-csr-file my-cert-id apachessl
```

`salt.modules.namecheap_ssl.`**`renew`**(*years*, *certificate_id*, *certificate_type*, *promotion_code=None*)

Renews an SSL certificate if it is ACTIVE and Expires <= 30 days. Returns the following information:

- The certificate ID
- The order ID
- The transaction ID
- The amount charged for the order

years [1] Number of years to register

certificate_id Unique ID of the SSL certificate you wish to renew

certificate_type Type of SSL Certificate. Possible values include:

  - EV Multi Domain SSL
  - EV SSL
  - EV SSL SGC
  - EssentialSSL
  - EssentialSSL Wildcard
  - InstantSSL
  - InstantSSL Pro
  - Multi Domain SSL
  - PositiveSSL
  - PositiveSSL Multi Domain
  - PositiveSSL Wildcard
  - PremiumSSL
  - PremiumSSL Wildcard
  - QuickSSL Premium
  - RapidSSL
  - RapidSSL Wildcard
  - SGC Supercert
  - SSL Web Server
  - SSL Webserver EV
  - SSL123
  - Secure Site
  - Secure Site Pro
  - Secure Site Pro with EV
  - Secure Site with EV
  - True BusinessID
  - True BusinessID Multi Domain
  - True BusinessID Wildcard

- True BusinessID with EV
- True BusinessID with EV Multi Domain
- Unified Communications

promotional_code An optional promo code to use when renewing the certificate

CLI Example:

```
salt 'my-minion' namecheap_ssl.renew 1 my-cert-id RapidSSL
```

21.9.262 salt.modules.namecheap_users module

Namecheap User Management

New in version 2017.7.0.

Prerequisites

This module uses the `requests` Python module to communicate to the namecheap API.

Configuration

The Namecheap username, API key and URL should be set in the minion configuration file, or in the Pillar data.

```
namecheap.name: companyname
namecheap.key: a1b2c3d4e5f67a8b9c0d1e2f3
namecheap.client_ip: 162.155.30.172
#Real url
namecheap.url: https://api.namecheap.com/xml.response
#Sandbox url
#namecheap.url: https://api.sandbox.namecheap.xml.response
```

`salt.modules.namecheap_users.`**`check_balances`**`(`*minimum=100*`)`

Checks if the provided minimum value is present in the user's account.

Returns a boolean. Returns `False` if the user's account balance is less than the provided minimum or `True` if greater than the minimum.

minimum [100] The value to check

CLI Example:

```
salt 'my-minion' namecheap_users.check_balances
salt 'my-minion' namecheap_users.check_balances minimum=150
```

`salt.modules.namecheap_users.`**`get_balances`**`()`

Gets information about fund in the user's account. This method returns the following information: Available Balance, Account Balance, Earned Amount, Withdrawable Amount and Funds Required for AutoRenew.

Note: If a domain setup with automatic renewal is expiring within the next 90 days, the FundsRequiredForAutoRenew attribute shows the amount needed in your Namecheap account to complete auto renewal.

CLI Example:

```
salt 'my-minion' namecheap_users.get_balances
```

21.9.263 salt.modules.napalm_mod module

NAPALM helpers

Helpers for the NAPALM modules.

New in version 2017.7.0.

`salt.modules.napalm_mod.`**`alive`**`(`*`**kwargs`*`)`

Returns the alive status of the connection layer. The output is a dictionary under the usual dictionary output of the NAPALM modules.

CLI Example:

```
salt '*' napalm.alive
```

Output Example:

```
result: True
out:
    is_alive: False
comment: ''
```

`salt.modules.napalm_mod.`**`call`**`(`*`method, *args, **kwargs`*`)`

Execute arbitrary methods from the NAPALM library. To see the expected output, please consult the NAPALM documentation.

Note: This feature is not recommended to be used in production. It should be used for testing only!

CLI Example:

```
salt '*' napalm.call get_lldp_neighbors
salt '*' napalm.call get_firewall_policies
salt '*' napalm.call get_bgp_config group='my-group'
```

`salt.modules.napalm_mod.`**`compliance_report`**`(`*`filepath=None, string=None, renderer='jinja|yaml', **kwargs`*`)`

Return the compliance report.

filepath The absolute path to the validation file.

Changed in version 2019.2.0.

Beginning with release codename `2019.2.0`, this function has been enhanced, to be able to leverage the multi-engine template rendering of Salt, besides the possibility to retrieve the file source from remote systems, the URL schemes supported being:

- `salt://`
- `http://` and `https://`
- `ftp://`
- `s3://`
- `swift:/`

Or on the local file system (on the Minion).

Note: The rendering result does not necessarily need to be YAML, instead it can be any format interpreted by Salt's rendering pipeline (including pure Python).

string New in version 2019.2.0.

The compliance report send as inline string, to be used as the file to send through the renderer system. Note, not all renderer modules can work with strings; the 'py' renderer requires a file, for example.

renderer: **`jinja|yaml`** New in version 2019.2.0.

The renderer pipe to send the file through; this is overridden by a "she-bang" at the top of the file.

kwargs Changed in version 2019.2.0.

Keyword args to pass to Salt's compile_template() function.

CLI Example:

```
salt '*' napalm.compliance_report ~/validate.yml
salt '*' napalm.compliance_report salt://path/to/validator.sls
```

Validation File Example (pure YAML):

```
- get_facts:
    os_version: 4.17

- get_interfaces_ip:
    Management1:
      ipv4:
        10.0.2.14:
          prefix_length: 24
        _mode: strict
```

Validation File Example (as Jinja + YAML):

```
- get_facts:
    os_version: {{ grains.version }}
- get_interfaces_ip:
    Loopback0:
      ipv4:
        {{ grains.lo0.ipv4 }}:
          prefix_length: 24
        _mode: strict
- get_bgp_neighbors: {{ pillar.bgp.neighbors }}
```

Output Example:

```
device1:
    ----------
    comment:
    out:
        ----------
        complies:
            False
        get_facts:
            ----------
            complies:
                False
            extra:
            missing:
            present:
                ----------
                os_version:
                    ----------
                    actual_value:
```

(continues on next page)

(continued from previous page)

```
                        15.1F6-S1.4
                    complies:
                        False
                    nested:
                        False
        get_interfaces_ip:
            ----------
            complies:
                False
            extra:
            missing:
                - Management1
            present:
                ----------
        skipped:
    result:
        True
```

`salt.modules.napalm_mod.`**`config_diff_text`**(*source1='candidate'*, *candidate_path=None*, *source2='running'*, *running_path=None*)

New in version 2019.2.0.

Return the diff, as text, between the two different configuration sources. The sources can be either specified using the `source1` and `source2` arguments when retrieving from the managed network device.

source1: **`candidate`** The source from where to retrieve the configuration to be compared with. Available options: `candidate`, `running`, `startup`. Default: `candidate`.

candidate_path Absolute or remote path from where to load the candidate configuration text. This argument allows any URI supported by *`cp.get_url`*), e.g., `salt://`, `https://`, `s3://`, `ftp:/`, etc.

source2: **`running`** The source from where to retrieve the configuration to compare with. Available options: `candidate`, `running`, `startup`. Default: `running`.

running_path Absolute or remote path from where to load the runing configuration text. This argument allows any URI supported by *`cp.get_url`*), e.g., `salt://`, `https://`, `s3://`, `ftp:/`, etc.

saltenv: **`base`** Salt fileserver environment from which to retrieve the file. Ignored if `candidate_path` or `running_path` is not a `salt://` URL.

CLI Example:

```
salt '*' napalm.config_diff_text
salt '*' napalm.config_diff_text candidate_path=https://bit.ly/2mAdq7z
# Would compare the running config with the configuration available at
# https://bit.ly/2mAdq7z
```

`salt.modules.napalm_mod.`**`config_diff_tree`**(*source1='candidate'*, *candidate_path=None*, *source2='running'*, *running_path=None*)

New in version 2019.2.0.

Return the diff, as Python dictionary, between two different sources. The sources can be either specified using the `source1` and `source2` arguments when retrieving from the managed network device.

source1: **`candidate`** The source from where to retrieve the configuration to be compared with. Available options: `candidate`, `running`, `startup`. Default: `candidate`.

candidate_path Absolute or remote path from where to load the candidate configuration text. This argument allows any URI supported by *`cp.get_url`*), e.g., `salt://`, `https://`, `s3://`, `ftp:/`, etc.

source2: **`running`** The source from where to retrieve the configuration to compare with. Available options: `candidate`, `running`, `startup`. Default: `running`.

running_path Absolute or remote path from where to load the runing configuration text. This argument allows any URI supported by *`cp.get_url`*), e.g., `salt://`, `https://`, `s3://`, `ftp:/`, etc.

saltenv: **`base`** Salt fileserver environment from which to retrieve the file. Ignored if `candidate_path` or `running_path` is not a `salt://` URL.

CLI Example:

```
salt '*' napalm.config_diff_text
salt '*' napalm.config_diff_text candidate_path=https://bit.ly/2mAdq7z
# Would compare the running config with the configuration available at
# https://bit.ly/2mAdq7z
```

CLI Example:

```
salt '*' napalm.config_diff_tree
salt '*' napalm.config_diff_tree running startup
```

`salt.modules.napalm_mod.`**`config_filter_lines`**(*parent_regex*, *child_regex*, *source='running'*)

New in version 2019.2.0.

Return a list of detailed matches, for the configuration blocks (parent-child relationship) whose parent respects the regular expressions configured via the `parent_regex` argument, and the child matches the `child_regex` regular expression. The result is a list of dictionaries with the following keys:

- `match`: a boolean value that tells whether `child_regex` matched any children lines.
- `parent`: the parent line (as text).
- `child`: the child line (as text). If no child line matched, this field will be `None`.

Note: This function is only available only when the underlying library ciscoconfparse is installed. See *`ciscoconfparse module`* for more details.

parent_regex The regular expression to match the parent configuration lines against.

child_regex The regular expression to match the child configuration lines against.

source: `running` The configuration type to retrieve from the network device. Default: `running`. Available options: `running`, `startup`, `candidate`.

CLI Example:

```
salt '*' napalm.config_filter_lines '^interface' 'ip address'
salt '*' napalm.config_filter_lines '^interface' 'shutdown' source=candidate
```

`salt.modules.napalm_mod.`**`config_find_lines`**(*regex*, *source='running'*)

New in version 2019.2.0.

Return the configuration lines that match the regular expressions from the `regex` argument. The configuration is read from the network device interrogated.

regex The regular expression to match the configuration lines against.

source: `running` The configuration type to retrieve from the network device. Default: `running`. Available options: `running`, `startup`, `candidate`.

CLI Example:

```
salt '*' napalm.config_find_lines '^interface Ethernet1\d'
```

`salt.modules.napalm_mod.`**`config_lines_w_child`**(*parent_regex*, *child_regex*, *source='running'*)

New in version 2019.2.0.

Return the configuration lines that match the regular expressions from the `parent_regex` argument, having child lines matching `child_regex`. The configuration is read from the network device interrogated.

Note: This function is only available only when the underlying library ciscoconfparse is installed. See *ciscoconfparse module* for more details.

---

parent_regex The regular expression to match the parent configuration lines against.
child_regex The regular expression to match the child configuration lines against.
source: `running` The configuration type to retrieve from the network device. Default: `running`. Available options: `running`, `startup`, `candidate`.

CLI Example:

```
salt '*' napalm.config_lines_w_child '^interface' 'ip address'
salt '*' napalm.config_lines_w_child '^interface' 'shutdown' source=candidate
```

`salt.modules.napalm_mod.`**`config_lines_wo_child`**(*parent_regex*, *child_regex*, *source='running'*)

New in version 2019.2.0.

Return the configuration lines that match the regular expressions from the `parent_regex` argument, having the child lines *not* matching `child_regex`. The configuration is read from the network device interrogated.

Note: This function is only available only when the underlying library ciscoconfparse is installed. See *ciscoconfparse module* for more details.

---

parent_regex The regular expression to match the parent configuration lines against.
child_regex The regular expression to match the child configuration lines against.
source: `running` The configuration type to retrieve from the network device. Default: `running`. Available options: `running`, `startup`, `candidate`.

CLI Example:

```
salt '*' napalm.config_lines_wo_child '^interface' 'ip address'
salt '*' napalm.config_lines_wo_child '^interface' 'shutdown' source=candidate
```

`salt.modules.napalm_mod.`**`config_merge_diff`**(*source='running'*, *merge_config=None*, *merge_path=None*, *saltenv='base'*)

New in version 2019.2.0.

Return the merge diff, as text, after merging the merge config into the configuration source requested (without loading the config on the device).

source: `running` The configuration type to retrieve from the network device. Default: `running`. Available options: `running`, `startup`, `candidate`.
merge_config The config to be merged into the initial config, sent as text. This argument is ignored when `merge_path` is set.
merge_path Absolute or remote path from where to load the merge configuration text. This argument allows any URI supported by *cp.get_url*), e.g., `salt://`, `https://`, `s3://`, `ftp:/`, etc.
saltenv: `base` Salt fileserver environment from which to retrieve the file. Ignored if `merge_path` is not a `salt://` URL.

CLI Example:

```
salt '*' napalm.config_merge_diff merge_path=salt://path/to/merge.cfg
```

`salt.modules.napalm_mod.`**`config_merge_text`**(*source='running'*, *merge_config=None*, *merge_path=None*, *saltenv='base'*)

New in version 2019.2.0.

Return the merge result of the configuration from `source` with the merge configuration, as plain text (without loading the config on the device).

source: `running` The configuration type to retrieve from the network device. Default: `running`. Available options: `running`, `startup`, `candidate`.

merge_config The config to be merged into the initial config, sent as text. This argument is ignored when `merge_path` is set.

merge_path Absolute or remote path from where to load the merge configuration text. This argument allows any URI supported by *cp.get_url*), e.g., `salt://`, `https://`, `s3://`, `ftp:/`, etc.

saltenv: `base` Salt fileserver environment from which to retrieve the file. Ignored if `merge_path` is not a `salt://` URL.

CLI Example:

```
salt '*' napalm.config_merge_text merge_path=salt://path/to/merge.cfg
```

`salt.modules.napalm_mod.`**`config_merge_tree`**(*source='running', merge_config=None, merge_path=None, saltenv='base'*)

New in version 2019.2.0.

Return the merge tree of the `initial_config` with the `merge_config`, as a Python dictionary.

source: `running` The configuration type to retrieve from the network device. Default: `running`. Available options: `running`, `startup`, `candidate`.

merge_config The config to be merged into the initial config, sent as text. This argument is ignored when `merge_path` is set.

merge_path Absolute or remote path from where to load the merge configuration text. This argument allows any URI supported by *cp.get_url*), e.g., `salt://`, `https://`, `s3://`, `ftp:/`, etc.

saltenv: `base` Salt fileserver environment from which to retrieve the file. Ignored if `merge_path` is not a `salt://` URL.

CLI Example:

```
salt '*' napalm.config_merge_tree merge_path=salt://path/to/merge.cfg
```

`salt.modules.napalm_mod.`**`config_tree`**(*source='running', with_tags=False*)

New in version 2019.2.0.

Transform Cisco IOS style configuration to structured Python dictionary. Depending on the value of the `with_tags` argument, this function may provide different views, valuable in different situations.

source: `running` The configuration type to retrieve from the network device. Default: `running`. Available options: `running`, `startup`, `candidate`.

with_tags: `False` Whether this function should return a detailed view, with tags.

CLI Example:

```
salt '*' napalm.config_tree
```

`salt.modules.napalm_mod.`**`junos_call`**(*fun, *args, **kwargs*)

New in version 2019.2.0.

Execute an arbitrary function from the `junos execution module`. To check what `args` and `kwargs` you must send to the function, please consult the appropriate documentation.

fun The name of the function. E.g., `set_hostname`.

args List of arguments to send to the `junos` function invoked.

kwargs Dictionary of key-value arguments to send to the `juno` function invoked.

CLI Example:

```
salt '*' napalm.junos_fun cli 'show system commit'
```

`salt.modules.napalm_mod.`**`junos_cli`**(*command, format=None, dev_timeout=None, dest=None, **kwargs*)

New in version 2019.2.0.

Execute a CLI command and return the output in the specified format.

command The command to execute on the Junos CLI.

format: **`text`** Format in which to get the CLI output (either `text` or `xml`).

dev_timeout: **`30`** The NETCONF RPC timeout (in seconds).

dest Destination file where the RPC output is stored. Note that the file will be stored on the Proxy Minion. To push the files to the Master, use *`cp.push`*.

CLI Example:

```
salt '*' napalm.junos_cli 'show lldp neighbors'
```

`salt.modules.napalm_mod.`**`junos_commit`**(***kwargs*)

New in version 2019.2.0.

Commit the changes loaded in the candidate configuration.

dev_timeout: **`30`** The NETCONF RPC timeout (in seconds).

comment Provide a comment for the commit.

confirm Provide time in minutes for commit confirmation. If this option is specified, the commit will be rolled back in the specified amount of time unless the commit is confirmed.

sync: **`False`** When `True`, on dual control plane systems, requests that the candidate configuration on one control plane be copied to the other control plane, checked for correct syntax, and committed on both Routing Engines.

force_sync: **`False`** When `True`, on dual control plane systems, force the candidate configuration on one control plane to be copied to the other control plane.

full When `True`, requires all the daemons to check and evaluate the new configuration.

detail When `True`, return commit detail.

CLI Examples:

```
salt '*' napalm.junos_commit comment='Commitiing via Salt' detail=True
salt '*' napalm.junos_commit dev_timeout=60 confirm=10
salt '*' napalm.junos_commit sync=True dev_timeout=90
```

`salt.modules.napalm_mod.`**`junos_copy_file`**(*src, dst, **kwargs*)

New in version 2019.2.0.

Copies the file on the remote Junos device.

src The source file path. This argument accepts the usual Salt URIs (e.g., `salt://`, `http://`, `https://`, `s3://`, `ftp://`, etc.).

dst The destination path on the device where to copy the file.

CLI Example:

```
salt '*' napalm.junos_copy_file https://example.com/junos.cfg /var/tmp/myjunos.cfg
```

`salt.modules.napalm_mod.`**`junos_facts`**(***kwargs*)

New in version 2019.2.0.

The complete list of Junos facts collected by `junos-eznc`.

CLI Example:

```
salt '*' napalm.junos_facts
```

`salt.modules.napalm_mod.`**`junos_install_os`**(*path=None, **kwargs*)

New in version 2019.2.0.

Installs the given image on the device.

path The image file source. This argument supports the following URIs:

- Absolute path on the Minion.
- `salt://` to fetch from the Salt fileserver.
- `http://` and `https://`

- `ftp://`
- `swift:/`
- `s3://`

dev_timeout: **`30`** The NETCONF RPC timeout (in seconds)
reboot: **`False`** Whether to reboot the device after the installation is complete.
no_copy: **`False`** If `True` the software package will not be copied to the remote device.
CLI Example:

```
salt '*' napalm.junos_install_os salt://images/junos_16_1.tgz reboot=True
```

`salt.modules.napalm_mod.`**`junos_rpc`**`(`*cmd=None, dest=None, format=None, **kwargs*`)`
New in version 2019.2.0.

Execute an RPC request on the remote Junos device.

cmd The RPC request to the executed. To determine the RPC request, you can check the from the command line of the device, by executing the usual command followed by `| display xml rpc`, e.g., `show lldp neighbors | display xml rpc`.

dest Destination file where the RPC output is stored. Note that the file will be stored on the Proxy Minion. To push the files to the Master, use *`cp.push`* Execution function.

format: **`xml`** The format in which the RPC reply is received from the device.
dev_timeout: **`30`** The NETCONF RPC timeout.
filter Used with the `get-config` RPC request to filter out the config tree.
terse: **`False`** Whether to return terse output.

Note: Some RPC requests may not support this argument.

interface_name Name of the interface to query.
CLI Example:

```
salt '*' napalm.junos_rpc get-lldp-neighbors-information
salt '*' napalm.junos_rcp get-config <configuration><system><ntp/></system></
↪configuration>
```

`salt.modules.napalm_mod.`**`netmiko_args`**`(`***kwargs*`)`
New in version 2019.2.0.

Return the key-value arguments used for the authentication arguments for the netmiko module.

When running in a non-native NAPALM driver (e.g., `panos`, *f5‘*, `mos` - either from https://github.com/napalm-automation-community or defined in user's own environment, one can specify the Netmiko device type (the `device_type` argument) via the `netmiko_device_type_map` configuration option / Pillar key, e.g.,

```
netmiko_device_type_map:
  f5: f5_ltm
  dellos10: dell_os10
```

The configuration above defines the mapping between the NAPALM `os` Grain and the Netmiko `device_type`, e.g., when the NAPALM Grain is `f5`, it would use the `f5_ltm` SSH Netmiko driver to execute commands over SSH on the remote network device.

CLI Example:

```
salt '*' napalm.netmiko_args
```

`salt.modules.napalm_mod.`**`netmiko_call`**`(`*method, *args, **kwargs*`)`
New in version 2019.2.0.

Execute an arbitrary Netmiko method, passing the authentication details from the existing NAPALM connection.

method The name of the Netmiko method to execute.

args List of arguments to send to the Netmiko method specified in `method`.

kwargs Key-value arguments to send to the execution function specified in `method`.

CLI Example:

```
salt '*' napalm.netmiko_call send_command 'show version'
```

`salt.modules.napalm_mod.`**`netmiko_commands`**`(`**commands*, ***kwargs*`)`

New in version 2019.2.0.

Invoke one or more commands to be executed on the remote device, via Netmiko. Returns a list of strings, with the output from each command.

commands A list of commands to be executed.

expect_string Regular expression pattern to use for determining end of output. If left blank will default to being based on router prompt.

delay_factor: **`1`** Multiplying factor used to adjust delays (default: `1`).

max_loops: **`500`** Controls wait time in conjunction with delay_factor. Will default to be based upon self.timeout.

auto_find_prompt: **`True`** Whether it should try to auto-detect the prompt (default: `True`).

strip_prompt: **`True`** Remove the trailing router prompt from the output (default: `True`).

strip_command: **`True`** Remove the echo of the command from the output (default: `True`).

normalize: **`True`** Ensure the proper enter is sent at end of command (default: `True`).

use_textfsm: **`False`** Process command output through TextFSM template (default: `False`).

CLI Example:

```
salt '*' napalm.netmiko_commands 'show version' 'show interfaces'
```

`salt.modules.napalm_mod.`**`netmiko_config`**`(`**config_commands*, ***kwargs*`)`

New in version 2019.2.0.

Load a list of configuration commands on the remote device, via Netmiko.

> **Warning:** Please remember that `netmiko` does not have any rollback safeguards and any configuration change will be directly loaded into the running config if the platform doesn't have the concept of `candidate` config.
>
> On Junos, or other platforms that have this capability, the changes will not be loaded into the running config, and the user must set the `commit` argument to `True` to transfer the changes from the candidate into the running config before exiting.

config_commands A list of configuration commands to be loaded on the remote device.

config_file Read the configuration commands from a file. The file can equally be a template that can be rendered using the engine of choice (see `template_engine`).

This can be specified using the absolute path to the file, or using one of the following URL schemes:

- `salt://`, to fetch the file from the Salt fileserver.
- `http://` or `https://`
- `ftp://`
- `s3://`
- `swift://`

exit_config_mode: **`True`** Determines whether or not to exit config mode after complete.

delay_factor: **`1`** Factor to adjust delays.

max_loops: **`150`** Controls wait time in conjunction with delay_factor (default: `150`).

strip_prompt: **`False`** Determines whether or not to strip the prompt (default: `False`).

strip_command: `False` Determines whether or not to strip the command (default: `False`).

config_mode_command The command to enter into config mode.

commit: `False` Commit the configuration changes before exiting the config mode. This option is by default disabled, as many platforms don't have this capability natively.

CLI Example:

```
salt '*' napalm.netmiko_config 'set system ntp peer 1.2.3.4' commit=True
salt '*' napalm.netmiko_config https://bit.ly/2sgljCB
```

`salt.modules.napalm_mod.`**`netmiko_conn`**`(`*`**kwargs`*`)`

New in version 2019.2.0.

Return the connection object with the network device, over Netmiko, passing the authentication details from the existing NAPALM connection.

> **Warning:** This function is not suitable for CLI usage, more rather to be used in various Salt modules.

USAGE Example:

```
conn = __salt__['napalm.netmiko_conn']()
res = conn.send_command('show interfaces')
conn.disconnect()
```

`salt.modules.napalm_mod.`**`netmiko_fun`**`(`*fun, *args, **kwargs*`)`

New in version 2019.2.0.

Call an arbitrary function from the *`Netmiko`* module, passing the authentication details from the existing NAPALM connection.

fun The name of the function from the *`Netmiko`* to invoke.

args List of arguments to send to the execution function specified in `fun`.

kwargs Key-value arguments to send to the execution function specified in `fun`.

CLI Example:

```
salt '*' napalm.netmiko_fun send_command 'show version'
```

`salt.modules.napalm_mod.`**`netmiko_multi_call`**`(`**methods, **kwargs*`)`

New in version 2019.2.0.

Execute a list of arbitrary Netmiko methods, passing the authentication details from the existing NAPALM connection.

methods List of dictionaries with the following keys:

- `name`: the name of the Netmiko function to invoke.
- `args`: list of arguments to send to the `name` method.
- `kwargs`: key-value arguments to send to the `name` method.

CLI Example:

```
salt '*' napalm.netmiko_multi_call "{'name': 'send_command', 'args': ['show
→version']}" "{'name': 'send_command', 'args': ['show interfaces']}"
```

`salt.modules.napalm_mod.`**`nxos_api_config`**`(`*commands=None, config_file=None, template_engine='jinja', context=None, defaults=None, saltenv='base', **kwargs*`)`

New in version 2019.2.0.

Configures the Nexus switch with the specified commands, via the NX-API.

commands The list of configuration commands to load on the Nexus switch.

Note: This argument is ignored when `config_file` is specified.

---

config_file The source file with the configuration commands to be sent to the device.

The file can also be a template that can be rendered using the template engine of choice. This can be specified using the absolute path to the file, or using one of the following URL schemes:

- `salt://`
- `https://`
- `ftp:/`
- `s3:/`
- `swift://`

template_engine: `jinja` The template engine to use when rendering the source file. Default: `jinja`. To simply fetch the file without attempting to render, set this argument to `None`.

context: `None` Variables to add to the template context.

defaults: `None` Default values of the `context` dict.

saltenv: `base` Salt fileserver environment from which to retrieve the file. Ignored if `config_file` is not a `salt://` URL.

CLI Example:

```
salt '*' napalm.nxos_api_config 'spanning-tree mode mstp'
salt '*' napalm.nxos_api_config config_file=https://bit.ly/2LGLcDy context="{
↪'servers': ['1.2.3.4']}"
```

`salt.modules.napalm_mod.`**`nxos_api_rpc`**(*commands*, *method='cli'*, ***kwargs*)

New in version 2019.2.0.

Execute an arbitrary RPC request via the Nexus API.

commands The RPC commands to be executed.

method: `cli` The type of the response, i.e., raw text (`cli_ascii`) or structured document (`cli`). Defaults to `cli` (structured data).

CLI Example:

```
salt '*' napalm.nxos_api_rpc 'show version'
```

`salt.modules.napalm_mod.`**`nxos_api_show`**(*commands*, *raw_text=True*, ***kwargs*)

New in version 2019.2.0.

Execute one or more show (non-configuration) commands.

commands The commands to be executed.

raw_text: `True` Whether to return raw text or structured data.

CLI Example:

```
salt '*' napalm.nxos_api_show 'show version'
salt '*' napalm.nxos_api_show 'show bgp sessions' 'show processes' raw_text=False
```

`salt.modules.napalm_mod.`**`pyeapi_call`**(*method*, **args*, ***kwargs*)

New in version 2019.2.0.

Invoke an arbitrary method from the `pyeapi` library. This function forwards the existing connection details to the `pyeapi.run_commands` execution function.

method The name of the `pyeapi` method to invoke.

kwargs Key-value arguments to send to the `pyeapi` method.

CLI Example:

```
salt '*' napalm.pyeapi_call run_commands 'show version' encoding=text
salt '*' napalm.pyeapi_call get_config as_string=True
```

`salt.modules.napalm_mod.`**`pyeapi_config`**`(`*commands=None, config_file=None, template_engine='jinja', context=None, defaults=None, saltenv='base', **kwargs*`)`

New in version 2019.2.0.

Configures the Arista switch with the specified commands, via the `pyeapi` library. This function forwards the existing connection details to the `pyeapi.run_commands` execution function.

commands The list of configuration commands to load on the Arista switch.

> **Note:** This argument is ignored when `config_file` is specified.

config_file The source file with the configuration commands to be sent to the device.

The file can also be a template that can be rendered using the template engine of choice. This can be specified using the absolute path to the file, or using one of the following URL schemes:

- `salt://`
- `https://`
- `ftp:/`
- `s3:/`
- `swift://`

template_engine: **`jinja`** The template engine to use when rendering the source file. Default: `jinja`. To simply fetch the file without attempting to render, set this argument to `None`.

context: **`None`** Variables to add to the template context.

defaults: **`None`** Default values of the `context` dict.

saltenv: **`base`** Salt fileserver environment from which to retrieve the file. Ignored if `config_file` is not a `salt://` URL.

CLI Example:

```
salt '*' napalm.pyeapi_config 'ntp server 1.2.3.4'
```

`salt.modules.napalm_mod.`**`pyeapi_conn`**`(`***kwargs*`)`

New in version 2019.2.0.

Return the connection object with the Arista switch, over `pyeapi`, passing the authentication details from the existing NAPALM connection.

> **Warning:** This function is not suitable for CLI usage, more rather to be used in various Salt modules, to reusing the established connection, as in opposite to opening a new connection for each task.

Usage example:

```
conn = __salt__['napalm.pyeapi_conn']()
res1 = conn.run_commands('show version')
res2 = conn.get_config(as_string=True)
```

`salt.modules.napalm_mod.`**`pyeapi_nxos_api_args`**`(`***prev_kwargs*`)`

New in version 2019.2.0.

Return the key-value arguments used for the authentication arguments for the `pyeapi execution module`.

CLI Example:

```
salt '*' napalm.pyeapi_nxos_api_args
```

`salt.modules.napalm_mod.`**`pyeapi_run_commands`**(**commands*, ***kwargs*)

Execute a list of commands on the Arista switch, via the `pyeapi` library. This function forwards the existing connection details to the `pyeapi.run_commands` execution function.

commands A list of commands to execute.

encoding: `json` The requested encoding of the command output. Valid values for encoding are `json` (default) or `text`.

CLI Example:

```
salt '*' napalm.pyeapi_run_commands 'show version' encoding=text
salt '*' napalm.pyeapi_run_commands 'show ip bgp neighbors'
```

`salt.modules.napalm_mod.`**`reconnect`**(*force=False*, ***kwargs*)

Reconnect the NAPALM proxy when the connection is dropped by the network device. The connection can be forced to be restarted using the `force` argument.

Note: This function can be used only when running proxy minions.

CLI Example:

```
salt '*' napalm.reconnect
salt '*' napalm.reconnect force=True
```

`salt.modules.napalm_mod.`**`rpc`**(*command*, ***kwargs*)

New in version 2019.2.0.

This is a wrapper to execute RPC requests on various network operating systems supported by NAPALM, invoking the following functions for the NAPALM native drivers:

- *`napalm.junos_rpc`* for `junos`
- *`napalm.pyeapi_run_commands`* for `eos`
- *`napalm.nxos_api_rpc`* for `nxos`
- *`napalm.netmiko_commands`* for `ios`, `iosxr`, and `nxos_ssh`

command The RPC command to execute. This depends on the nature of the operating system.

kwargs Key-value arguments to be sent to the underlying Execution function.

The function capabilities are extensible in the user environment via the `napalm_rpc_map` configuration option / Pillar, e.g.,

```
napalm_rpc_map:
  f5: napalm.netmiko_commands
  panos: panos.call
```

The mapping above reads: when the NAPALM `os` Grain is `f5`, then call `napalm.netmiko_commands` for RPC requests.

By default, if the user does not specify any map, non-native NAPALM drivers will invoke the `napalm.netmiko_commands` Execution function.

CLI Example:

```
salt '*' napalm.rpc 'show version'
salt '*' napalm.rpc get-interfaces
```

`salt.modules.napalm_mod.`**`scp_get`**`(`*remote_path*, *local_path=''*, *recursive=False*, *preserve_times=False*, ***kwargs*`)`

New in version 2019.2.0.

Transfer files and directories from remote network device to the localhost of the Minion.

Note: This function is only available only when the underlying library scp is installed. See *`scp module`* for more details.

remote_path Path to retrieve from remote host. Since this is evaluated by scp on the remote host, shell wildcards and environment variables may be used.
recursive: `False` Transfer files and directories recursively.
preserve_times: `False` Preserve `mtime` and `atime` of transferred files and directories.
passphrase Used for decrypting private keys.
pkey An optional private key to use for authentication.
key_filename The filename, or list of filenames, of optional private key(s) and/or certificates to try for authentication.
timeout An optional timeout (in seconds) for the TCP connect.
socket_timeout: `10` The channel socket timeout in seconds.
buff_size: `16384` The size of the SCP send buffer.
allow_agent: `True` Set to `False` to disable connecting to the SSH agent.
look_for_keys: `True` Set to `False` to disable searching for discoverable private key files in `~/.ssh/`
banner_timeout An optional timeout (in seconds) to wait for the SSH banner to be presented.
auth_timeout An optional timeout (in seconds) to wait for an authentication response.
auto_add_policy: `False` Automatically add the host to the `known_hosts`.

CLI Example:

```
salt '*' napalm.scp_get /var/tmp/file /tmp/file auto_add_policy=True
```

`salt.modules.napalm_mod.`**`scp_put`**`(`*files*, *remote_path=None*, *recursive=False*, *preserve_times=False*, *saltenv='base'*, ***kwargs*`)`

New in version 2019.2.0.

Transfer files and directories to remote network device.

Note: This function is only available only when the underlying library scp is installed. See *`scp module`* for more details.

files A single path or a list of paths to be transferred.
remote_path The path on the remote device where to store the files.
recursive: `True` Transfer files and directories recursively.
preserve_times: `False` Preserve `mtime` and `atime` of transferred files and directories.
saltenv: `base` The name of the Salt environment. Ignored when `files` is not a `salt://` URL.
hostname The hostname of the remote device.
port: `22` The port of the remote device.
username The username required for SSH authentication on the device.
password Used for password authentication. It is also used for private key decryption if `passphrase` is not given.
passphrase Used for decrypting private keys.
pkey An optional private key to use for authentication.
key_filename The filename, or list of filenames, of optional private key(s) and/or certificates to try for authentication.
timeout An optional timeout (in seconds) for the TCP connect.

socket_timeout: **`10`** The channel socket timeout in seconds.
buff_size: **`16384`** The size of the SCP send buffer.
allow_agent: **`True`** Set to `False` to disable connecting to the SSH agent.
look_for_keys: **`True`** Set to `False` to disable searching for discoverable private key files in `~/.ssh/`
banner_timeout An optional timeout (in seconds) to wait for the SSH banner to be presented.
auth_timeout An optional timeout (in seconds) to wait for an authentication response.
auto_add_policy: **`False`** Automatically add the host to the `known_hosts`.

CLI Example:

```
salt '*' napalm.scp_put /path/to/file /var/tmp/file auto_add_policy=True
```

21.9.264 salt.modules.napalm_netacl

NAPALM ACL

Generate and load ACL (firewall) configuration on network devices.

New in version 2017.7.0.

codeauthor Mircea Ulinic <ping@mirceaulinic.net>

maturity new

depends capirca, napalm

platform unix

Dependencies

The firewall configuration is generated by Capirca.

To install Capirca, execute: `pip install capirca`.

To be able to load configuration on network devices, it requires NAPALM library to be installed: `pip install napalm`. Please check Installation for complete details.

`salt.modules.napalm_netacl.`**`get_filter_pillar`**(*filter_name*, *pillar_key='acl'*, *pillarenv=None*, *saltenv=None*)

Helper that can be used inside a state SLS, in order to get the filter configuration given its name.
filter_name The name of the filter.
pillar_key The root key of the whole policy config.
pillarenv Query the master to generate fresh pillar data on the fly, specifically from the requested pillar environment.
saltenv Included only for compatibility with *`pillarenv_from_saltenv`*, and is otherwise ignored.

`salt.modules.napalm_netacl.`**`get_term_pillar`**(*filter_name*, *term_name*, *pillar_key='acl'*, *pillarenv=None*, *saltenv=None*)

Helper that can be used inside a state SLS, in order to get the term configuration given its name, under a certain filter uniquely identified by its name.
filter_name The name of the filter.
term_name The name of the term.
pillar_key: **`acl`** The root key of the whole policy config. Default: `acl`.
pillarenv Query the master to generate fresh pillar data on the fly, specifically from the requested pillar environment.
saltenv Included only for compatibility with *`pillarenv_from_saltenv`*, and is otherwise ignored.

`salt.modules.napalm_netacl.`**`load_filter_config`**(*filter_name, filter_options=None, terms=None, prepend=True, pillar_key='acl', pillarenv=None, saltenv=None, merge_pillar=True, only_lower_merge=False, revision_id=None, revision_no=None, revision_date=True, revision_date_format='%Y/%m/%d', test=False, commit=True, debug=False, **kwargs*)

Generate and load the configuration of a policy filter.

Note: The order of the terms is very important. The configuration loaded on the device respects the order defined in the `terms` and/or inside the pillar.

When merging the `terms` with the pillar data, consider the `prepend` argument to make sure the order is correct!

---

filter_name The name of the policy filter.

filter_options Additional filter options. These options are platform-specific. See the complete list of options.

terms List of terms for this policy filter. If not specified or empty, will try to load the configuration from the pillar, unless `merge_pillar` is set as `False`.

prepend: `True` When `merge_pillar` is set as `True`, the final list of terms generated by merging the terms from `terms` with those defined in the pillar (if any): new terms are prepended at the beginning, while existing ones will preserve the position. To add the new terms at the end of the list, set this argument to `False`.

pillar_key: `acl` The key in the pillar containing the default attributes values. Default: `acl`.

pillarenv Query the master to generate fresh pillar data on the fly, specifically from the requested pillar environment.

saltenv Included only for compatibility with *`pillarenv_from_saltenv`*, and is otherwise ignored.

merge_pillar: `True` Merge the CLI variables with the pillar. Default: `True`.

The merge logic depends on the `prepend` argument and the CLI has higher priority than the pillar.

only_lower_merge: `False` Specify if it should merge only the terms fields. Otherwise it will try to merge also filters fields. Default: `False`. This option requires `merge_pillar`, otherwise it is ignored.

revision_id Add a comment in the filter config having the description for the changes applied.

revision_no The revision count.

revision_date: `True` Boolean flag: display the date when the filter configuration was generated. Default: `True`.

revision_date_format: `%Y/%m/%d` The date format to be used when generating the perforce data. Default: `%Y/%m/%d` (<year>/<month>/<day>).

test: `False` Dry run? If set as `True`, will apply the config, discard and return the changes. Default: `False` and will commit the changes on the device.

commit: `True` Commit? Default: `True`.

debug: `False` Debug mode. Will insert a new key under the output dictionary, as `loaded_config` containing the raw configuration loaded on the device.

The output is a dictionary having the same form as *`net.load_config`*.

CLI Example:

```
salt 'edge01.bjm01' netacl.load_filter_config my-filter pillar_key=netacl
↪debug=True
```

Output Example:

```
edge01.bjm01:
    ----------
    already_configured:
        False
    comment:
    diff:
        [edit firewall]
        +    family inet {
        +        /*
        +         ** $Date: 2017/03/22 $
        +         **
        +         */
        +        filter my-filter {
        +            interface-specific;
        +            term my-term {
        +                from {
        +                    source-port [ 1234 1235 ];
        +                }
        +                then {
        +                    reject;
        +                }
        +            }
        +            term my-other-term {
        +                from {
        +                    protocol tcp;
        +                    source-port 5678-5680;
        +                }
        +                then accept;
        +            }
        +        }
        +    }
    loaded_config:
        firewall {
            family inet {
                replace:
                /*
                ** $Date: 2017/03/22 $
                **
                */
                filter my-filter {
                    interface-specific;
                    term my-term {
                        from {
                            source-port [ 1234 1235 ];
                        }
                        then {
                            reject;
                        }
                    }
                    term my-other-term {
                        from {
                            protocol tcp;
                            source-port 5678-5680;
                        }
                        then accept;
                    }
                }
```

(continues on next page)

(continued from previous page)

```
            }
        }
    result:
        True
```

The filter configuration has been loaded from the pillar, having the following structure:

```
netacl:
  - my-filter:
      terms:
        - my-term:
            source_port:
             - 1234
             - 1235
            action: reject
        - my-other-term:
            source_port:
              - - 5678
                - 5680
            protocol: tcp
            action: accept
```

salt.modules.napalm_netacl.**load_policy_config**(*filters=None, prepend=True, pillar_key='acl', pillarenv=None, saltenv=None, merge_pillar=True, only_lower_merge=False, revision_id=None, revision_no=None, revision_date=True, revision_date_format='%Y/%m/%d', test=False, commit=True, debug=False, **kwargs*)

Generate and load the configuration of the whole policy.

Note: The order of the filters and their terms is very important. The configuration loaded on the device respects the order defined in the `filters` and/or inside the pillar.

When merging the `filters` with the pillar data, consider the `prepend` argument to make sure the order is correct!

filters List of filters for this policy. If not specified or empty, will try to load the configuration from the pillar, unless `merge_pillar` is set as `False`.

prepend: **`True`** When `merge_pillar` is set as `True`, the final list of filters generated by merging the filters from `filters` with those defined in the pillar (if any): new filters are prepended at the beginning, while existing ones will preserve the position. To add the new filters at the end of the list, set this argument to `False`.

pillar_key: **`acl`** The key in the pillar containing the default attributes values. Default: `acl`.

pillarenv Query the master to generate fresh pillar data on the fly, specifically from the requested pillar environment.

saltenv Included only for compatibility with *`pillarenv_from_saltenv`*, and is otherwise ignored.

merge_pillar: **`True`** Merge the CLI variables with the pillar. Default: `True`.

The merge logic depends on the `prepend` argument and the CLI has higher priority than the pillar.

only_lower_merge: **`False`** Specify if it should merge only the filters and terms fields. Otherwise it will try to merge everything at the policy level. Default: `False`. This option requires `merge_pillar`, otherwise it is ignored.

revision_id Add a comment in the policy config having the description for the changes applied.

revision_no The revision count.

revision_date: True Boolean flag: display the date when the policy configuration was generated. Default: `True`.

revision_date_format: %Y/%m/%d The date format to be used when generating the perforce data. Default: `%Y/%m/%d` (<year>/<month>/<day>).

test: False Dry run? If set as `True`, will apply the config, discard and return the changes. Default: `False` and will commit the changes on the device.

commit: True Commit? Default: `True`.

debug: False Debug mode. Will insert a new key under the output dictionary, as `loaded_config` containing the raw configuration loaded on the device.

The output is a dictionary having the same form as *`net.load_config`*.

CLI Example:

```
salt 'edge01.flw01' netacl.load_policy_config debug=True
```

Output Example:

```
edge01.flw01:
    ----------
    already_configured:
        False
    comment:
    diff:
        ---
        +++
        @@ -1228,9 +1228,24 @@
         !
        +ipv4 access-list my-filter
        + 10 remark my-term
        + 20 deny tcp host 1.2.3.4 eq 1234 any
        + 30 deny udp host 1.2.3.4 eq 1234 any
        + 40 deny tcp host 1.2.3.4 eq 1235 any
        + 50 deny udp host 1.2.3.4 eq 1235 any
        + 60 remark my-other-term
        + 70 permit tcp any range 5678 5680 any
        +!
        +!
        +ipv4 access-list block-icmp
        + 10 remark first-term
        + 20 deny icmp any any
         !
    loaded_config:
        ! $Date: 2017/03/22 $
        no ipv4 access-list my-filter
        ipv4 access-list my-filter
         remark my-term
         deny tcp host 1.2.3.4 eq 1234 any
         deny udp host 1.2.3.4 eq 1234 any
         deny tcp host 1.2.3.4 eq 1235 any
         deny udp host 1.2.3.4 eq 1235 any
         remark my-other-term
         permit tcp any range 5678 5680 any
        exit
        no ipv4 access-list block-icmp
        ipv4 access-list block-icmp
```

(continues on next page)

(continued from previous page)

```
          remark first-term
          deny icmp any any
        exit
    result:
        True
```

The policy configuration has been loaded from the pillar, having the following structure:

```
acl:
  - my-filter:
      terms:
        - my-term:
            source_port:
             - 1234
             - 1235
            protocol:
              - tcp
              - udp
            source_address: 1.2.3.4
            action: reject
        - my-other-term:
            source_port:
              - [5678, 5680]
            protocol: tcp
            action: accept
  - block-icmp:
      terms:
        - first-term:
            protocol:
              - icmp
            action: reject
```

salt.modules.napalm_netacl.**load_term_config**(*filter_name, term_name, filter_options=None, pillar_key='acl', pillarenv=None, saltenv=None, merge_pillar=True, revision_id=None, revision_no=None, revision_date=True, revision_date_format='%Y/%m/%d', test=False, commit=True, debug=False, source_service=None, destination_service=None, **term_fields*)

Generate and load the configuration of a policy term.

filter_name The name of the policy filter.

term_name The name of the term.

filter_options Additional filter options. These options are platform-specific. See the complete list of options.

pillar_key: `acl` The key in the pillar containing the default attributes values. Default: `acl`. If the pillar contains the following structure:

```
firewall:
  - my-filter:
      terms:
        - my-term:
            source_port: 1234
            source_address:
                - 1.2.3.4/32
                - 5.6.7.8/32
```

The `pillar_key` field would be specified as `firewall`.

pillarenv Query the master to generate fresh pillar data on the fly, specifically from the requested pillar environment.

saltenv Included only for compatibility with *`pillarenv_from_saltenv`*, and is otherwise ignored.

merge_pillar: `True` Merge the CLI variables with the pillar. Default: `True`.

The properties specified through the CLI have higher priority than the pillar.

revision_id Add a comment in the term config having the description for the changes applied.

revision_no The revision count.

revision_date: `True` Boolean flag: display the date when the term configuration was generated. Default: `True`.

revision_date_format: `%Y/%m/%d` The date format to be used when generating the perforce data. Default: `%Y/%m/%d` (<year>/<month>/<day>).

test: `False` Dry run? If set as `True`, will apply the config, discard and return the changes. Default: `False` and will commit the changes on the device.

commit: `True` Commit? Default: `True`.

debug: `False` Debug mode. Will insert a new key under the output dictionary, as `loaded_config` containing the raw configuration loaded on the device.

source_service A special service to choose from. This is a helper so the user is able to select a source just using the name, instead of specifying a source_port and protocol.

As this module is available on Unix platforms only, it reads the IANA port assignment from /etc/services.

If the user requires additional shortcuts to be referenced, they can add entries under /etc/services, which can be managed using the *`file state`*.

destination_service A special service to choose from. This is a helper so the user is able to select a source just using the name, instead of specifying a destination_port and protocol. Allows the same options as `source_service`.

term_fields Term attributes. To see what fields are supported, please consult the list of supported keywords. Some platforms have a few other optional keywords.

Note: The following fields are accepted (some being platform-specific):

- action
- address
- address_exclude
- comment
- counter
- expiration
- destination_address
- destination_address_exclude
- destination_port
- destination_prefix
- forwarding_class
- forwarding_class_except
- logging
- log_name
- loss_priority
- option
- policer
- port
- precedence
- principals
- protocol
- protocol_except
- qos

- pan_application
- routing_instance
- source_address
- source_address_exclude
- source_port
- source_prefix
- verbatim
- packet_length
- fragment_offset
- hop_limit
- icmp_type
- ether_type
- traffic_class_count
- traffic_type
- translated
- dscp_set
- dscp_match
- dscp_except
- next_ip
- flexible_match_range
- source_prefix_except
- destination_prefix_except
- vpn
- source_tag
- destination_tag
- source_interface
- destination_interface
- flattened
- flattened_addr
- flattened_saddr
- flattened_daddr
- priority

Note: The following fields can be also a single value and a list of values:

- action
- address
- address_exclude
- comment
- destination_address
- destination_address_exclude
- destination_port
- destination_prefix
- forwarding_class
- forwarding_class_except
- logging
- option
- port
- precedence
- principals
- protocol
- protocol_except
- pan_application

- source_address
- source_address_exclude
- source_port
- source_prefix
- verbatim
- icmp_type
- ether_type
- traffic_type
- dscp_match
- dscp_except
- flexible_match_range
- source_prefix_except
- destination_prefix_except
- source_tag
- destination_tag
- source_service
- destination_service

Example: `destination_address` can be either defined as:

```
destination_address: 172.17.17.1/24
```

or as a list of destination IP addresses:

```
destination_address:
    - 172.17.17.1/24
    - 172.17.19.1/24
```

or a list of services to be matched:

```
source_service:
    - ntp
    - snmp
    - ldap
    - bgpd
```

Note: The port fields `source_port` and `destination_port` can be used as above to select either a single value, either a list of values, but also they can select port ranges. Example:

```
source_port:
    - - 1000
      - 2000
    - - 3000
      - 4000
```

With the configuration above, the user is able to select the 1000-2000 and 3000-4000 source port ranges.

The output is a dictionary having the same form as *`net.load_config`*.

CLI Example:

```
salt 'edge01.bjm01' netacl.load_term_config filter-name term-name source_
→address=1.2.3.4 destination_address=5.6.7.8 action=accept test=True debug=True
```

Output Example:

```
edge01.bjm01:
    ----------
    already_configured:
        False
    comment:
        Configuration discarded.
    diff:
        [edit firewall]
        +    family inet {
        +        /*
        +         ** $Date: 2017/03/22 $
        +         **
        +         */
        +        filter filter-name {
        +            interface-specific;
        +            term term-name {
        +                from {
        +                    source-address {
        +                        1.2.3.4/32;
        +                    }
        +                    destination-address {
        +                        5.6.7.8/32;
        +                    }
        +                }
        +                then accept;
        +            }
        +        }
        +    }
    loaded_config:
        firewall {
            family inet {
                replace:
                /*
                ** $Date: 2017/03/22 $
                **
                */
                filter filter-name {
                    interface-specific;
                    term term-name {
                        from {
                            source-address {
                                1.2.3.4/32;
                            }
                            destination-address {
                                5.6.7.8/32;
                            }
                        }
                        then accept;
                    }
                }
            }
        }
    result:
        True
```

21.9.265 salt.modules.napalm_bgp module

NAPALM BGP

Manages BGP configuration on network devices and provides statistics.

codeauthor Mircea Ulinic <ping@mirceaulinic.net> & Jerome Fleury <jf@cloudflare.com>

maturity new

depends napalm

platform unix

Dependencies

- *napalm proxy minion*

New in version 2016.11.0.

`salt.modules.napalm_bgp.`**`config`**(*group=None*, *neighbor=None*, ***kwargs*)

Provides the BGP configuration on the device.

Parameters

- **`group`** – Name of the group selected to display the configuration.
- **`neighbor`** – IP Address of the neighbor to display the configuration. If the group parameter is not specified, the neighbor setting will be ignored.

Returns A dictionary containing the BGP configuration from the network device. The keys of the main dictionary are the group names.

Each group has the following properties:

- type (string)
- description (string)
- apply_groups (string list)
- multihop_ttl (int)
- multipath (True/False)
- local_address (string)
- local_as (int)
- remote_as (int)
- import_policy (string)
- export_policy (string)
- remove_private_as (True/False)
- prefix_limit (dictionary)
- neighbors (dictionary)

Each neighbor in the dictionary of neighbors provides:

- description (string)
- import_policy (string)
- export_policy (string)
- local_address (string)
- local_as (int)
- remote_as (int)
- authentication_key (string)
- prefix_limit (dictionary)
- route_reflector_client (True/False)
- nhs (True/False)

CLI Example:

```
salt '*' bgp.config # entire BGP config
salt '*' bgp.config PEERS-GROUP-NAME # provides detail only about BGP group PEERS-
↪GROUP-NAME
salt '*' bgp.config PEERS-GROUP-NAME 172.17.17.1 # provides details only about
↪BGP neighbor 172.17.17.1,
# configured in the group PEERS-GROUP-NAME
```

Output Example:

```
{
    'PEERS-GROUP-NAME':{
        'type'          : 'external',
        'description'   : 'Here we should have a nice description',
        'apply_groups'  : ['BGP-PREFIX-LIMIT'],
        'import_policy' : 'PUBLIC-PEER-IN',
        'export_policy' : 'PUBLIC-PEER-OUT',
        'remove_private': True,
        'multipath'     : True,
        'multihop_ttl'  : 30,
        'neighbors'     : {
            '192.168.0.1': {
                'description'   : 'Facebook [CDN]',
                'prefix_limit'  : {
                    'inet': {
                        'unicast': {
                            'limit': 100,
                            'teardown': {
                                'threshold' : 95,
                                'timeout'   : 5
                            }
                        }
                    }
                }
                'peer-as'        : 32934,
                'route_reflector': False,
                'nhs'            : True
            },
            '172.17.17.1': {
                'description'   : 'Twitter [CDN]',
                'prefix_limit'  : {
                    'inet': {
                        'unicast': {
                            'limit': 500,
                            'no-validate': 'IMPORT-FLOW-ROUTES'
                        }
                    }
                }
                'peer_as'        : 13414
                'route_reflector': False,
                'nhs'            : False
            }
        }
    }
}
```

`salt.modules.napalm_bgp.`**`neighbors`**(*neighbor=None*, ***kwargs*)

Provides details regarding the BGP sessions configured on the network device.

Parameters **`neighbor`** – IP Address of a specific neighbor.

Returns

A dictionary with the statistics of the all/selected BGP neighbors. Outer dictionary keys represent the VRF name. Keys of inner dictionary represent the AS numbers, while the values are lists of dictionaries, having the following keys:

- up (True/False)
- local_as (int)
- remote_as (int)
- local_address (string)
- routing_table (string)
- local_address_configured (True/False)
- local_port (int)
- remote_address (string)
- remote_port (int)
- multihop (True/False)
- multipath (True/False)
- remove_private_as (True/False)
- import_policy (string)
- export_policy (string)
- input_messages (int)
- output_messages (int)
- input_updates (int)
- output_updates (int)
- messages_queued_out (int)
- connection_state (string)
- previous_connection_state (string)
- last_event (string)
- suppress_4byte_as (True/False)
- local_as_prepend (True/False)
- holdtime (int)
- configured_holdtime (int)
- keepalive (int)
- configured_keepalive (int)
- active_prefix_count (int)
- received_prefix_count (int)
- accepted_prefix_count (int)
- suppressed_prefix_count (int)
- advertised_prefix_count (int)

- flap_count (int)

CLI Example:

```
salt '*' bgp.neighbors  # all neighbors
salt '*' bgp.neighbors 172.17.17.1  # only session with BGP neighbor(s) 172.17.17.1
```

Output Example:

```
{
    'default': {
        8121: [
            {
                'up'                        : True,
                'local_as'                  : 13335,
                'remote_as'                 : 8121,
                'local_address'             : '172.101.76.1',
                'local_address_configured'  : True,
                'local_port'                : 179,
                'remote_address'            : '192.247.78.0',
                'router_id'                 : '192.168.0.1',
                'remote_port'               : 58380,
                'multihop'                  : False,
                'import_policy'             : '4-NTT-TRANSIT-IN',
                'export_policy'             : '4-NTT-TRANSIT-OUT',
                'input_messages'            : 123,
                'output_messages'           : 13,
                'input_updates'             : 123,
                'output_updates'            : 5,
                'messages_queued_out'       : 23,
                'connection_state'          : 'Established',
                'previous_connection_state' : 'EstabSync',
                'last_event'                : 'RecvKeepAlive',
                'suppress_4byte_as'         : False,
                'local_as_prepend'          : False,
                'holdtime'                  : 90,
                'configured_holdtime'       : 90,
                'keepalive'                 : 30,
                'configured_keepalive'      : 30,
                'active_prefix_count'       : 132808,
                'received_prefix_count'     : 566739,
                'accepted_prefix_count'     : 566479,
                'suppressed_prefix_count'   : 0,
                'advertise_prefix_count'    : 0,
                'flap_count'                : 27
            }
        ]
    }
}
```

21.9.266 salt.modules.napalm_formula module

NAPALM Formula helpers

New in version 2019.2.0.

This is an Execution Module providing helpers for various NAPALM formulas, e.g., napalm-interfaces-formula, napalm-bgp-formula, napalm-ntp-formula etc., meant to provide various helper functions to make the templates more readable.

`salt.modules.napalm_formula.`**`container_path`**(*model*, *key=None*, *container=None*, *delim=':'*)

Return the list of all the possible paths in a container, down to the `config` container. This function can be used to verify that the `model` is a Python object correctly structured and respecting the OpenConfig hierarchy.

model The OpenConfig-structured object to inspect.

delim: **`:`** The key delimiter. In particular cases, it is indicated to use `//` as `:` might be already used in various cases, e.g., IPv6 addresses, interface name (e.g., Juniper QFX series), etc.

CLI Example:

```
salt '*' napalm_formula.container_path "{'interfaces': {'interface': {'Ethernet1
↪': {'config': {'name': 'Ethernet1'}}}}}"
```

The example above would return a list with the following element: `interfaces:interface:Ethernet1:config` which is the only possible path in that hierarchy.

Other output examples:

```
- interfaces:interface:Ethernet1:config
- interfaces:interface:Ethernet1:subinterfaces:subinterface:0:config
- interfaces:interface:Ethernet2:config
```

`salt.modules.napalm_formula.`**`defaults`**(*model*, *defaults_*, *delim='//'*, *flipped_merge=False*)

Apply the defaults to a Python dictionary having the structure as described in the OpenConfig standards.

model The OpenConfig model to apply the defaults to.

defaults The dictionary of defaults. This argument must equally be structured with respect to the OpenConfig standards.

For ease of use, the keys of these support glob matching, therefore we don't have to provide the defaults for each entity but only for the entity type. See an example below.

delim: **`//`** The key delimiter to use. Generally, `//` should cover all the possible cases, and you don't need to override this value.

flipped_merge: **`False`** Whether should merge the model into the defaults, or the defaults into the model. Default: `False` (merge the model into the defaults, i.e., any defaults would be overridden by the values from the `model`).

CLI Example:

```
salt '*' napalm_formula.defaults "{'interfaces': {'interface': {'Ethernet1': {
↪'config': {'name': 'Ethernet1'}}}}}" "{'interfaces': {'interface': {'*': {
↪'config': {'enabled': True}}}}}"
```

As one can notice in the example above, the `*` corresponds to the interface name, therefore, the defaults will be applied on all the interfaces.

`salt.modules.napalm_formula.`**`dictupdate`**(*dest*, *upd*, *recursive_update=True*, *merge_lists=False*)

Recursive version of the default dict.update

Merges upd recursively into dest

If recursive_update=False, will use the classic dict.update, or fall back on a manual merge (helpful for non-dict types like `FunctionWrapper`).

If `merge_lists=True`, will aggregate list object types instead of replace. The list in `upd` is added to the list in `dest`, so the resulting list is `dest[key] + upd[key]`. This behaviour is only activated when `recursive_update=True`. By default `merge_lists=False`.

`salt.modules.napalm_formula.`**`render_field`**(*dictionary*, *field*, *prepend=None*, *append=None*, *quotes=False*, ***opts*)

Render a field found under the `field` level of the hierarchy in the `dictionary` object. This is useful to render a field in a Jinja template without worrying that the hierarchy might not exist. For example if we do the following in Jinja: `{{ interfaces.interface.Ethernet5.config.description }}` for the following object: `{'interfaces': {'interface': {'Ethernet1': {'config': {'enabled': True}}}}}` it would error, as the `Ethernet5` key does not exist. With this helper, we can skip this and avoid existence checks. This must be however used with care.

dictionary The dictionary to traverse.

field The key name or part to traverse in the `dictionary`.

prepend: `None` The text to prepend in front of the text. Usually, we need to have the name of the field too when generating the configuration.

append: `None` Text to append at the end.

quotes: `False` Whether should wrap the text around quotes.

CLI Example:

```
salt '*' napalm_formula.render_field "{'enabled': True}" enabled
# This would return the value of the ``enabled`` leaf key
salt '*' napalm_formula.render_field "{'enabled': True}" description
# This would not error
```

Jinja usage example:

```
{%- set config = {'enabled': True, 'description': 'Interface description'} %}
{{ salt.napalm_formula.render_field(config, 'description', quotes=True) }}
```

The example above would be rendered on Arista / Cisco as:

```
description "Interface description"
```

While on Junos (the semicolon is important to be added, otherwise the configuration won't be accepted by Junos):

```
description "Interface description";
```

`salt.modules.napalm_formula.`**`render_fields`**(*dictionary*, **fields*, ***opts*)

This function works similarly to *render_field* but for a list of fields from the same dictionary, rendering, indenting and distributing them on separate lines.

dictionary The dictionary to traverse.

fields A list of field names or paths in the dictionary.

indent: `0` The indentation to use, prepended to the rendered field.

separator: `\n` The separator to use between fields.

CLI Example:

```
salt '*' napalm_formula.render_fields "{'mtu': 68, 'description': 'Interface
→description'}" mtu description
```

Jinja usage example:

```
{%- set config={'mtu': 68, 'description': 'Interface description'} %}
{{ salt.napalm_formula.render_fields(config, 'mtu', 'description', quotes=True) }}
```

The Jinja example above would generate the following configuration:

```
mtu "68"
description "Interface description"
```

`salt.modules.napalm_formula.`**`setval`**(*key*, *val*, *dict_=None*, *delim=':'*)

Set a value under the dictionary hierarchy identified under the key. The target 'foo/bar/baz' returns the dictionary hierarchy {'foo': {'bar': {'baz': {}}}}.

Note: Currently this doesn't work with integers, i.e. cannot build lists dynamically.

CLI Example:

```
salt '*' formula.setval foo:baz:bar True
```

`salt.modules.napalm_formula.`**`traverse`**(*data*, *key*, *default=None*, *delimiter=':'*)

Traverse a dict or list using a colon-delimited (or otherwise delimited, using the `delimiter` param) target string. The target `foo:bar:0` will return `data['foo']['bar'][0]` if this value exists, and will otherwise return the dict in the default argument. Function will automatically determine the target type. The target `foo:bar:0` will return data['foo']['bar'][0] if data like `{'foo':{'bar':['baz']}}` , if data like `{'foo':{'bar':{'0':'baz'}}}` then `return data['foo']['bar']['0']`

CLI Example:

```
salt '*' napalm_formula.traverse "{'foo': {'bar': {'baz': True}}}" foo:baz:bar
```

21.9.267 salt.modules.napalm_network module

NAPALM Network

Basic methods for interaction with the network device through the virtual proxy 'napalm'.

codeauthor Mircea Ulinic <ping@mirceaulinic.net> & Jerome Fleury <jf@cloudflare.com>

maturity new

depends napalm

platform unix

Dependencies

- *napalm proxy minion*

New in version 2016.11.0.

Changed in version 2017.7.0.

`salt.modules.napalm_network.`**`arp`**(*interface=''*, *ipaddr=''*, *macaddr=''*, ***kwargs*)

NAPALM returns a list of dictionaries with details of the ARP entries.

Parameters

- **`interface`** – interface name to filter on
- **`ipaddr`** – IP address to filter on
- **`macaddr`** – MAC address to filter on

Returns List of the entries in the ARP table

CLI Example:

```
salt '*' net.arp
salt '*' net.arp macaddr='5c:5e:ab:da:3c:f0'
```

Example output:

```
[
    {
        'interface' : 'MgmtEth0/RSP0/CPU0/0',
        'mac'       : '5c:5e:ab:da:3c:f0',
        'ip'        : '172.17.17.1',
        'age'       : 1454496274.84
    },
    {
        'interface': 'MgmtEth0/RSP0/CPU0/0',
        'mac'       : '66:0e:94:96:e0:ff',
        'ip'        : '172.17.17.2',
        'age'       : 1435641582.49
    }
]
```

`salt.modules.napalm_network.`**`blockreplace`**(*marker_start, marker_end, content='', append_if_not_found=False, prepend_if_not_found=False, show_changes=True, append_newline=False, source='running', path=None, test=False, commit=True, debug=False, replace=True*)

New in version 2019.2.0.

Replace content of the configuration source, delimited by the line markers.

A block of content delimited by comments can help you manage several lines without worrying about old entries removal.

marker_start The line content identifying a line as the start of the content block. Note that the whole line containing this marker will be considered, so whitespace or extra content before or after the marker is included in final output.

marker_end The line content identifying a line as the end of the content block. Note that the whole line containing this marker will be considered, so whitespace or extra content before or after the marker is included in final output.

content The content to be used between the two lines identified by `marker_start` and `marker_stop`.

append_if_not_found: **`False`** If markers are not found and set to True then, the markers and content will be appended to the file.

prepend_if_not_found: **`False`** If markers are not found and set to True then, the markers and content will be prepended to the file.

append_newline: **`False`** Controls whether or not a newline is appended to the content block. If the value of this argument is `True` then a newline will be added to the content block. If it is `False`, then a newline will not be added to the content block. If it is `None` then a newline will only be added to the content block if it does not already end in a newline.

show_changes: **`True`** Controls how changes are presented. If `True`, this function will return the of the changes made. If `False`, then it will return a boolean (`True` if any changes were made, otherwise False).

source: **`running`** The configuration source. Choose from: `running`, `candidate`, or `startup`. Default: `running`.

path: **`None`** Save the temporary configuration to a specific path, then read from there. This argument is optional, can be used when you prefers a particular location of the temporary file.

test: **`False`** Dry run? If set as `True`, will apply the config, discard and return the changes. Default: `False` and will commit the changes on the device.

commit: **`True`** Commit the configuration changes? Default: `True`.

debug: **`False`** Debug mode. Will insert a new key in the output dictionary, as `loaded_config` containing the raw configuration loaded on the device.

replace: `True` Load and replace the configuration. Default: `True`.
CLI Example:

```
salt '*' net.blockreplace 'ntp' 'interface' ''
```

`salt.modules.napalm_network.`**`cancel_commit`**(*jid*)
New in version 2019.2.0.

Cancel a commit scheduled to be executed via the `commit_in` and `commit_at` arguments from the *`net.load_template`* or *`net.load_config`* execution functions. The commit ID is displayed when the commit is scheduled via the functions named above.

CLI Example:

```
salt '*' net.cancel_commit 20180726083540640360
```

`salt.modules.napalm_network.`**`cli`**(**commands*, ***kwargs*)
Returns a dictionary with the raw output of all commands passed as arguments.
commands List of commands to be executed on the device.
textfsm_parse: `False` Try parsing the outputs using the TextFSM templates.

New in version 2018.3.0.

Note: This option can be also specified in the minion configuration file or pillar as `napalm_cli_textfsm_parse`.

textfsm_path The path where the TextFSM templates can be found. This option implies the usage of the TextFSM index file. `textfsm_path` can be either absolute path on the server, either specified using the following URL mschemes: `file://`, `salt://`, `http://`, `https://`, `ftp://`, `s3://`, `swift://`.

New in version 2018.3.0.

Note: This needs to be a directory with a flat structure, having an index file (whose name can be specified using the `index_file` option) and a number of TextFSM templates.

Note: This option can be also specified in the minion configuration file or pillar as `textfsm_path`.

textfsm_template The path to a certain the TextFSM template. This can be specified using the absolute path to the file, or using one of the following URL schemes:

- `salt://`, to fetch the template from the Salt fileserver.
- `http://` or `https://`
- `ftp://`
- `s3://`
- `swift://`

New in version 2018.3.0.
textfsm_template_dict A dictionary with the mapping between a command and the corresponding TextFSM path to use to extract the data. The TextFSM paths can be specified as in `textfsm_template`.

New in version 2018.3.0.

Note: This option can be also specified in the minion configuration file or pillar as `napalm_cli_textfsm_template_dict`.

platform_grain_name: **`os`** The name of the grain used to identify the platform name in the TextFSM index file. Default: `os`.

New in version 2018.3.0.

Note: This option can be also specified in the minion configuration file or pillar as `textfsm_platform_grain`.

---

platform_column_name: **`Platform`** The column name used to identify the platform, exactly as specified in the TextFSM index file. Default: `Platform`.

New in version 2018.3.0.

Note: This is field is case sensitive, make sure to assign the correct value to this option, exactly as defined in the index file.

Note: This option can be also specified in the minion configuration file or pillar as `textfsm_platform_column_name`.

---

index_file: **`index`** The name of the TextFSM index file, under the `textfsm_path`. Default: `index`.

New in version 2018.3.0.

Note: This option can be also specified in the minion configuration file or pillar as `textfsm_index_file`.

---

saltenv: **`base`** Salt fileserver envrionment from which to retrieve the file. Ignored if `textfsm_path` is not a `salt://` URL.

New in version 2018.3.0.

include_empty: **`False`** Include empty files under the `textfsm_path`.

New in version 2018.3.0.

include_pat Glob or regex to narrow down the files cached from the given path. If matching with a regex, the regex must be prefixed with `E@`, otherwise the expression will be interpreted as a glob.

New in version 2018.3.0.

exclude_pat Glob or regex to exclude certain files from being cached from the given path. If matching with a regex, the regex must be prefixed with `E@`, otherwise the expression will be interpreted as a glob.

New in version 2018.3.0.

Note: If used with `include_pat`, files matching this pattern will be excluded from the subset of files defined by `include_pat`.

CLI Example:

```
salt '*' net.cli "show version" "show chassis fan"
```

CLI Example with TextFSM template:

```
salt '*' net.cli textfsm_parse=True textfsm_path=salt://textfsm/
```

Example output:

```
{
    'show version and haiku':  'Hostname: re0.edge01.arn01
                                  Model: mx480
                                  Junos: 13.3R6.5
                                    Help me, Obi-Wan
                                    I just saw Episode Two
                                    You're my only hope
                                 ',
    'show chassis fan' :    'Item                      Status   RPM     Measurement
                              Top Rear Fan              OK       3840     Spinning⊠
↪at intermediate-speed
                              Bottom Rear Fan           OK       3840     Spinning⊠
↪at intermediate-speed
                              Top Middle Fan            OK       3900     Spinning⊠
↪at intermediate-speed
                              Bottom Middle Fan         OK       3840     Spinning⊠
↪at intermediate-speed
                              Top Front Fan             OK       3810     Spinning⊠
↪at intermediate-speed
                              Bottom Front Fan          OK       3840     Spinning⊠
↪at intermediate-speed
                             '
}
```

Example output with TextFSM parsing:

```
{
  "comment": "",
  "result": true,
  "out": {
    "sh ver": [
      {
        "kernel": "9.1S3.5",
        "documentation": "9.1S3.5",
        "boot": "9.1S3.5",
        "crypto": "9.1S3.5",
        "chassis": "",
        "routing": "9.1S3.5",
        "base": "9.1S3.5",
        "model": "mx960"
      }
    ]
  }
}
```

`salt.modules.napalm_network.`**`commit`**(*inherit_napalm_device=None*, ***kwargs*)

Commits the configuration changes made on the network device.

CLI Example:

```
salt '*' net.commit
```

`salt.modules.napalm_network.`**`compare_config`**(*inherit_napalm_device=None*, ***kwargs*)

Returns the difference between the running config and the candidate config.

CLI Example:

```
salt '*' net.compare_config
```

`salt.modules.napalm_network.`**`config`**`(`*source=None*, ***kwargs*`)`

New in version 2017.7.0.

Return the whole configuration of the network device. By default, it will return all possible configuration sources supported by the network device. At most, there will be:

- running config
- startup config
- candidate config

To return only one of the configurations, you can use the `source` argument.

source Which configuration type you want to display, default is all of them.

Options:

- running
- candidate
- startup

Returns

The object returned is a dictionary with the following keys:

- running (string): Representation of the native running configuration.
- **candidate (string): Representation of the native candidate configuration**. If the device doesn't differentiate between running and startup configuration this will an empty string.
- **startup (string): Representation of the native startup configuration**. If the device doesn't differentiate between running and startup configuration this will an empty string.

CLI Example:

```
salt '*' net.config
salt '*' net.config source=candidate
```

`salt.modules.napalm_network.`**`config_changed`**`(`*inherit_napalm_device=None*, ***kwargs*`)`

Will prompt if the configuration has been changed.

Returns A tuple with a boolean that specifies if the config was changed on the device. And a string that provides more details of the reason why the configuration was not changed.

CLI Example:

```
salt '*' net.config_changed
```

`salt.modules.napalm_network.`**`config_control`**`(`*inherit_napalm_device=None*, ***kwargs*`)`

Will check if the configuration was changed. If differences found, will try to commit. In case commit unsuccessful, will try to rollback.

Returns A tuple with a boolean that specifies if the config was changed/committed/rollbacked on the device. And a string that provides more details of the reason why the configuration was not committed properly.

CLI Example:

```
salt '*' net.config_control
```

`salt.modules.napalm_network.`**`confirm_commit`**`(`*jid*`)`

New in version 2019.2.0.

Confirm a commit scheduled to be reverted via the `revert_in` and `revert_at` arguments from the *`net.load_template`* or *`net.load_config`* execution functions. The commit ID is displayed when the commit confirmed is scheduled via the functions named above.

CLI Example:

```
salt '*' net.confirm_commit 20180726083540640360
```

`salt.modules.napalm_network.`**`connected`**`(`*`**kwargs`*`)`

Specifies if the connection to the device succeeded.

CLI Example:

```
salt '*' net.connected
```

`salt.modules.napalm_network.`**`discard_config`**`(`*`inherit_napalm_device=None, **kwargs`*`)`

Discards the changes applied.

CLI Example:

```
salt '*' net.discard_config
```

`salt.modules.napalm_network.`**`environment`**`(`*`**kwargs`*`)`

Returns the environment of the device.

CLI Example:

```
salt '*' net.environment
```

Example output:

```
{
    'fans': {
        'Bottom Rear Fan': {
            'status': True
        },
        'Bottom Middle Fan': {
            'status': True
        },
        'Top Middle Fan': {
            'status': True
        },
        'Bottom Front Fan': {
            'status': True
        },
        'Top Front Fan': {
            'status': True
        },
        'Top Rear Fan': {
            'status': True
        }
    },
    'memory': {
        'available_ram': 16349,
        'used_ram': 4934
    },
    'temperature': {
       'FPC 0 Exhaust A': {
            'is_alert': False,
```

(continues on next page)

(continued from previous page)

```
                'temperature': 35.0,
                'is_critical': False
            }
        },
        'cpu': {
            '1': {
                '%usage': 19.0
            },
            '0': {
                '%usage': 35.0
            }
        }
    }
```

`salt.modules.napalm_network.`**`facts`**`(`*`**kwargs`*`)`

Returns characteristics of the network device. :return: a dictionary with the following keys:

- uptime - Uptime of the device in seconds.
- vendor - Manufacturer of the device.
- model - Device model.
- hostname - Hostname of the device
- fqdn - Fqdn of the device
- os_version - String with the OS version running on the device.
- serial_number - Serial number of the device
- interface_list - List of the interfaces of the device

CLI Example:

```
salt '*' net.facts
```

Example output:

```
{
    'os_version': '13.3R6.5',
    'uptime': 10117140,
    'interface_list': [
        'lc-0/0/0',
        'pfe-0/0/0',
        'pfh-0/0/0',
        'xe-0/0/0',
        'xe-0/0/1',
        'xe-0/0/2',
        'xe-0/0/3',
        'gr-0/0/10',
        'ip-0/0/10'
    ],
    'vendor': 'Juniper',
    'serial_number': 'JN131356FBFA',
    'model': 'MX480',
    'hostname': 're0.edge05.syd01',
    'fqdn': 're0.edge05.syd01'
}
```

`salt.modules.napalm_network.`**`interfaces`**`(`*`**kwargs`*`)`

Returns details of the interfaces on the device.

Returns Returns a dictionary of dictionaries. The keys for the first dictionary will be the interfaces in the devices.

CLI Example:

```
salt '*' net.interfaces
```

Example output:

```
{
    'Management1': {
        'is_up': False,
        'is_enabled': False,
        'description': '',
        'last_flapped': -1,
        'speed': 1000,
        'mac_address': 'dead:beef:dead',
    },
    'Ethernet1':{
        'is_up': True,
        'is_enabled': True,
        'description': 'foo',
        'last_flapped': 1429978575.1554043,
        'speed': 1000,
        'mac_address': 'beef:dead:beef',
    }
}
```

`salt.modules.napalm_network.`**`ipaddrs`**`(`*`**kwargs`*`)`

Returns IP addresses configured on the device.

Returns A dictionary with the IPv4 and IPv6 addresses of the interfaces. Returns all configured IP addresses on all interfaces as a dictionary of dictionaries. Keys of the main dictionary represent the name of the interface. Values of the main dictionary represent are dictionaries that may consist of two keys 'ipv4' and 'ipv6' (one, both or none) which are themselvs dictionaries with the IP addresses as keys.

CLI Example:

```
salt '*' net.ipaddrs
```

Example output:

```
{
    'FastEthernet8': {
        'ipv4': {
            '10.66.43.169': {
                'prefix_length': 22
            }
        }
    },
    'Loopback555': {
        'ipv4': {
            '192.168.1.1': {
                'prefix_length': 24
            }
        },
        'ipv6': {
            '1::1': {
                'prefix_length': 64
            },
            '2001:DB8:1::1': {
                'prefix_length': 64
            },
```

(continues on next page)

(continued from previous page)

```
                'FE80::3': {
                    'prefix_length': 'N/A'
                }
            }
        }
}
```

`salt.modules.napalm_network.`**`lldp`**(*interface=”, **kwargs*)

Returns a detailed view of the LLDP neighbors.

Parameters **`interface`** – interface name to filter on

Returns A dictionary with the LLDL neighbors. The keys are the interfaces with LLDP activated on.

CLI Example:

```
salt '*' net.lldp
salt '*' net.lldp interface='TenGigE0/0/0/8'
```

Example output:

```
{
    'TenGigE0/0/0/8': [
        {
            'parent_interface': 'Bundle-Ether8',
            'interface_description': 'TenGigE0/0/0/8',
            'remote_chassis_id': '8c60.4f69.e96c',
            'remote_system_name': 'switch',
            'remote_port': 'Eth2/2/1',
            'remote_port_description': 'Ethernet2/2/1',
            'remote_system_description': 'Cisco Nexus Operating System (NX-OS)
↪Software 7.1(0)N1(1a)
                  TAC support: http://www.cisco.com/tac
                  Copyright (c) 2002-2015, Cisco Systems, Inc. All rights
↪reserved.',
            'remote_system_capab': 'B, R',
            'remote_system_enable_capab': 'B'
        }
    ]
}
```

`salt.modules.napalm_network.`**`load_config`**(*filename=None, text=None, test=False, commit=True, debug=False, replace=False, commit_in=None, commit_at=None, revert_in=None, revert_at=None, commit_jid=None, inherit_napalm_device=None, saltenv='base', **kwargs*)

Applies configuration changes on the device. It can be loaded from a file or from inline string. If you send both a filename and a string containing the configuration, the file has higher precedence.

By default this function will commit the changes. If there are no changes, it does not commit and the flag `already_configured` will be set as `True` to point this out.

To avoid committing the configuration, set the argument `test` to `True` and will discard (dry run).

To keep the changes but not commit, set `commit` to `False`.

To replace the config, set `replace` to `True`.

filename Path to the file containing the desired configuration. This can be specified using the absolute path to the file, or using one of the following URL schemes:

- `salt://`, to fetch the template from the Salt fileserver.
- `http://` or `https://`
- `ftp://`
- `s3://`
- `swift://`

Changed in version 2018.3.0.

text String containing the desired configuration. This argument is ignored when `filename` is specified.

test: False Dry run? If set as `True`, will apply the config, discard and return the changes. Default: `False` and will commit the changes on the device.

commit: True Commit? Default: `True`.

debug: False Debug mode. Will insert a new key under the output dictionary, as `loaded_config` containing the raw configuration loaded on the device.

New in version 2016.11.2.

replace: False Load and replace the configuration. Default: `False`.

New in version 2016.11.2.

commit_in: `None` Commit the changes in a specific number of minutes / hours. Example of accepted formats: `5` (commit in 5 minutes), `2m` (commit in 2 minutes), `1h` (commit the changes in 1 hour)`, `5h30m` (commit the changes in 5 hours and 30 minutes).

Note: This feature works on any platforms, as it does not rely on the native features of the network operating system.

Note: After the command is executed and the `diff` is not satisfactory, or for any other reasons you have to discard the commit, you are able to do so using the *`net.cancel_commit`* execution function, using the commit ID returned by this function.

Warning: Using this feature, Salt will load the exact configuration you expect, however the diff may change in time (i.e., if an user applies a manual configuration change, or a different process or command changes the configuration in the meanwhile).

New in version 2019.2.0.

commit_at: `None` Commit the changes at a specific time. Example of accepted formats: `1am` (will commit the changes at the next 1AM), `13:20` (will commit at 13:20), `1:20am`, etc.

Note: This feature works on any platforms, as it does not rely on the native features of the network operating system.

Note: After the command is executed and the `diff` is not satisfactory, or for any other reasons you have to discard the commit, you are able to do so using the *`net.cancel_commit`* execution function, using the commit ID returned by this function.

Warning: Using this feature, Salt will load the exact configuration you expect, however the diff may change in time (i.e., if an user applies a manual configuration change, or a different process or command changes the configuration in the meanwhile).

New in version 2019.2.0.

revert_in: **`None`** Commit and revert the changes in a specific number of minutes / hours. Example of accepted formats: `5` (revert in 5 minutes), `2m` (revert in 2 minutes), `1h` (revert the changes in 1 hour)`, `5h30m` (revert the changes in 5 hours and 30 minutes).

Note: To confirm the commit, and prevent reverting the changes, you will have to execute the *`net.confirm_commit`* function, using the commit ID returned by this function.

Warning: This works on any platform, regardless if they have or don't have native capabilities to confirming a commit. However, please be *very* cautious when using this feature: on Junos (as it is the only NAPALM core platform supporting this natively) it executes a commit confirmed as you would do from the command line. All the other platforms don't have this capability natively, therefore the revert is done via Salt. That means, your device needs to be reachable at the moment when Salt will attempt to revert your changes. Be cautious when pushing configuration changes that would prevent you reach the device.

Similarly, if an user or a different process apply other configuration changes in the meanwhile (between the moment you commit and till the changes are reverted), these changes would be equally reverted, as Salt cannot be aware of them.

New in version 2019.2.0.

revert_at: **`None`** Commit and revert the changes at a specific time. Example of accepted formats: `1am` (will commit and revert the changes at the next 1AM), `13:20` (will commit and revert at 13:20), `1:20am`, etc.

Note: To confirm the commit, and prevent reverting the changes, you will have to execute the *`net.confirm_commit`* function, using the commit ID returned by this function.

Warning: This works on any platform, regardless if they have or don't have native capabilities to confirming a commit. However, please be *very* cautious when using this feature: on Junos (as it is the only NAPALM core platform supporting this natively) it executes a commit confirmed as you would do from the command line. All the other platforms don't have this capability natively, therefore the revert is done via Salt. That means, your device needs to be reachable at the moment when Salt will attempt to revert your changes. Be cautious when pushing configuration changes that would prevent you reach the device.

Similarly, if an user or a different process apply other configuration changes in the meanwhile (between the moment you commit and till the changes are reverted), these changes would be equally reverted, as Salt cannot be aware of them.

New in version 2019.2.0.

saltenv: **`base`** Specifies the Salt environment name.

New in version 2018.3.0.

Returns a dictionary having the following keys:

- result (bool): if the config was applied successfully. It is `False` only in case of failure. In case there are no changes to be applied and successfully performs all operations it is still `True` and so will be the `already_configured` flag (example below)
- comment (str): a message for the user

- already_configured (bool): flag to check if there were no changes applied
- loaded_config (str): the configuration loaded on the device. Requires `debug` to be set as `True`
- diff (str): returns the config changes applied

CLI Example:

```
salt '*' net.load_config text='ntp peer 192.168.0.1'
salt '*' net.load_config filename='/absolute/path/to/your/file'
salt '*' net.load_config filename='/absolute/path/to/your/file' test=True
salt '*' net.load_config filename='/absolute/path/to/your/file' commit=False
```

Example output:

```
{
    'comment': 'Configuration discarded.',
    'already_configured': False,
    'result': True,
    'diff': '[edit interfaces xe-0/0/5]+   description "Adding a description";'
}
```

`salt.modules.napalm_network.`**`load_template`**(*template_name=None, template_source=None, context=None, defaults=None, template_engine='jinja', saltenv='base', template_hash=None, template_hash_name=None, skip_verify=False, test=False, commit=True, debug=False, replace=False, commit_in=None, commit_at=None, revert_in=None, revert_at=None, inherit_napalm_device=None, **template_vars*)

Renders a configuration template (default: Jinja) and loads the result on the device.

By default this function will commit the changes. If there are no changes, it does not commit, discards he config and the flag `already_configured` will be set as `True` to point this out.

To avoid committing the configuration, set the argument `test` to `True` and will discard (dry run).

To preserve the changes, set `commit` to `False`. However, this is recommended to be used only in exceptional cases when there are applied few consecutive states and/or configuration changes. Otherwise the user might forget that the config DB is locked and the candidate config buffer is not cleared/merged in the running config.

To replace the config, set `replace` to `True`.

template_name Identifies path to the template source. The template can be either stored on the local machine, either remotely. The recommended location is under the `file_roots` as specified in the master config file. For example, let's suppose the `file_roots` is configured as:

```
file_roots:
  base:
    - /etc/salt/states
```

Placing the template under `/etc/salt/states/templates/example.jinja`, it can be used as `salt://templates/example.jinja`. Alternatively, for local files, the user can specify the absolute path. If remotely, the source can be retrieved via `http`, `https` or `ftp`.

Examples:

- `salt://my_template.jinja`
- `/absolute/path/to/my_template.jinja`
- `http://example.com/template.cheetah`
- `https:/example.com/template.mako`
- `ftp://example.com/template.py`

Changed in version 2019.2.0: This argument can now support a list of templates to be rendered. The resulting configuration text is loaded at once, as a single configuration chunk.

template_source: None Inline config template to be rendered and loaded on the device.

template_hash: None Hash of the template file. Format: `{hash_type: 'md5', 'hsum': <md5sum>}`

New in version 2016.11.2.

context: None Overrides default context variables passed to the template.

New in version 2019.2.0.

template_hash_name: None When `template_hash` refers to a remote file, this specifies the filename to look for in that file.

New in version 2016.11.2.

saltenv: `base` Specifies the template environment. This will influence the relative imports inside the templates.

New in version 2016.11.2.

template_engine: jinja The following templates engines are supported:

- *`cheetah`*
- *`genshi`*
- *`jinja`*
- *`mako`*
- *`py`*
- *`wempy`*

New in version 2016.11.2.

skip_verify: True If `True`, hash verification of remote file sources (`http://`, `https://`, `ftp://`) will be skipped, and the `source_hash` argument will be ignored.

New in version 2016.11.2.

test: False Dry run? If set to `True`, will apply the config, discard and return the changes. Default: `False` and will commit the changes on the device.

commit: True Commit? (default: `True`)

debug: False Debug mode. Will insert a new key under the output dictionary, as `loaded_config` containing the raw result after the template was rendered.

New in version 2016.11.2.

replace: False Load and replace the configuration.

New in version 2016.11.2.

commit_in: `None` Commit the changes in a specific number of minutes / hours. Example of accepted formats: `5` (commit in 5 minutes), `2m` (commit in 2 minutes), `1h` (commit the changes in 1 hour)', `5h30m` (commit the changes in 5 hours and 30 minutes).

Note: This feature works on any platforms, as it does not rely on the native features of the network operating system.

Note: After the command is executed and the `diff` is not satisfactory, or for any other reasons you have to discard the commit, you are able to do so using the *`net.cancel_commit`* execution function, using the commit ID returned by this function.

Warning: Using this feature, Salt will load the exact configuration you expect, however the diff may change in time (i.e., if an user applies a manual configuration change, or a different process or command changes the configuration in the meanwhile).

New in version 2019.2.0.

commit_at: **None** Commit the changes at a specific time. Example of accepted formats: `1am` (will commit the changes at the next 1AM), `13:20` (will commit at 13:20), `1:20am`, etc.

Note: This feature works on any platforms, as it does not rely on the native features of the network operating system.

Note: After the command is executed and the `diff` is not satisfactory, or for any other reasons you have to discard the commit, you are able to do so using the *`net.cancel_commit`* execution function, using the commit ID returned by this function.

Warning: Using this feature, Salt will load the exact configuration you expect, however the diff may change in time (i.e., if an user applies a manual configuration change, or a different process or command changes the configuration in the meanwhile).

New in version 2019.2.0.

revert_in: **None** Commit and revert the changes in a specific number of minutes / hours. Example of accepted formats: `5` (revert in 5 minutes), `2m` (revert in 2 minutes), `1h` (revert the changes in 1 hour)`, `5h30m` (revert the changes in 5 hours and 30 minutes).

Note: To confirm the commit, and prevent reverting the changes, you will have to execute the *`net.confirm_commit`* function, using the commit ID returned by this function.

Warning: This works on any platform, regardless if they have or don't have native capabilities to confirming a commit. However, please be *very* cautious when using this feature: on Junos (as it is the only NAPALM core platform supporting this natively) it executes a commit confirmed as you would do from the command line. All the other platforms don't have this capability natively, therefore the revert is done via Salt. That means, your device needs to be reachable at the moment when Salt will attempt to revert your changes. Be cautious when pushing configuration changes that would prevent you reach the device.

Similarly, if an user or a different process apply other configuration changes in the meanwhile (between the moment you commit and till the changes are reverted), these changes would be equally reverted, as Salt cannot be aware of them.

New in version 2019.2.0.

revert_at: **None** Commit and revert the changes at a specific time. Example of accepted formats: `1am` (will commit and revert the changes at the next 1AM), `13:20` (will commit and revert at 13:20), `1:20am`, etc.

Note: To confirm the commit, and prevent reverting the changes, you will have to execute the *`net.confirm_commit`* function, using the commit ID returned by this function.

> **Warning:** This works on any platform, regardless if they have or don't have native capabilities to confirming a commit. However, please be *very* cautious when using this feature: on Junos (as it is the only NAPALM core platform supporting this natively) it executes a commit confirmed as you would do from the command line. All the other platforms don't have this capability natively, therefore the revert is done via Salt. That means, your device needs to be reachable at the moment when Salt will attempt to revert your changes. Be cautious when pushing configuration changes that would prevent you reach the device.
>
> Similarly, if an user or a different process apply other configuration changes in the meanwhile (between the moment you commit and till the changes are reverted), these changes would be equally reverted, as Salt cannot be aware of them.

New in version 2019.2.0.

defaults: None Default variables/context passed to the template.

New in version 2016.11.2.

template_vars Dictionary with the arguments/context to be used when the template is rendered.

> **Note:** Do not explicitly specify this argument. This represents any other variable that will be sent to the template rendering system. Please see the examples below!

> **Note:** It is more recommended to use the `context` argument to avoid conflicts between CLI arguments and template variables.

Returns a dictionary having the following keys:

- result (bool): if the config was applied successfully. It is `False` only in case of failure. In case there are no changes to be applied and successfully performs all operations it is still `True` and so will be the `already_configured` flag (example below)
- comment (str): a message for the user
- already_configured (bool): flag to check if there were no changes applied
- loaded_config (str): the configuration loaded on the device, after rendering the template. Requires `debug` to be set as `True`
- diff (str): returns the config changes applied

The template can use variables from the `grains`, `pillar` or `opts`, for example:

```
{% set router_model = grains.get('model') -%}
{% set router_vendor = grains.get('vendor') -%}
{% set os_version = grains.get('version') -%}
{% set hostname = pillar.get('proxy', {}).get('host') -%}
{% if router_vendor|lower == 'juniper' %}
system {
    host-name {{hostname}};
}
{% elif router_vendor|lower == 'cisco' %}
hostname {{hostname}}
{% endif %}
```

CLI Examples:

```
salt '*' net.load_template set_ntp_peers peers=[192.168.0.1]  # uses NAPALM
↪default templates

# inline template:
salt -G 'os:junos' net.load_template template_source='system { host-name {{host_
↪name}}; }'         host_name='MX480.lab'

# inline template using grains info:
salt -G 'os:junos' net.load_template         template_source='system { host-name {
↪{grains.model}}.lab; }'
# if the device is a MX480, the command above will set the hostname as: MX480.lab

# inline template using pillar data:
salt -G 'os:junos' net.load_template template_source='system { host-name {{pillar.
↪proxy.host}}; }'

salt '*' net.load_template https://bit.ly/2OhSgqP hostname=example  # will commit
salt '*' net.load_template https://bit.ly/2OhSgqP hostname=example test=True  #
↪dry run

salt '*' net.load_template salt://templates/example.jinja debug=True  # Using the
↪salt:// URI

# render a mako template:
salt '*' net.load_template salt://templates/example.mako template_engine=mako
↪debug=True

# render remote template
salt -G 'os:junos' net.load_template http://bit.ly/2fReJg7 test=True debug=True
↪peers=['192.168.0.1']
salt -G 'os:ios' net.load_template http://bit.ly/2gKOj20 test=True debug=True
↪peers=['192.168.0.1']

# render multiple templates at once
salt '*' net.load_template "['https://bit.ly/2OhSgqP', 'salt://templates/example.
↪jinja']" context="{'hostname': 'example'}"
```

Example output:

```
{
    'comment': '',
    'already_configured': False,
    'result': True,
    'diff': '[edit system]+  host-name edge01.bjm01',
    'loaded_config': 'system { host-name edge01.bjm01; }''
}
```

`salt.modules.napalm_network.`**`mac`**(*address=”, interface=”, vlan=0, **kwargs*)

Returns the MAC Address Table on the device.

Parameters

- **`address`** – MAC address to filter on
- **`interface`** – Interface name to filter on
- **`vlan`** – VLAN identifier

Returns A list of dictionaries representing the entries in the MAC Address Table

CLI Example:

```
salt '*' net.mac
salt '*' net.mac vlan=10
```

Example output:

```
[
    {
        'mac'       : '00:1c:58:29:4a:71',
        'interface' : 'xe-3/0/2',
        'static'    : False,
        'active'    : True,
        'moves'     : 1,
        'vlan'      : 10,
        'last_move' : 1454417742.58
    },
    {
        'mac'       : '8c:60:4f:58:e1:c1',
        'interface' : 'xe-1/0/1',
        'static'    : False,
        'active'    : True,
        'moves'     : 2,
        'vlan'      : 42,
        'last_move' : 1453191948.11
    }
]
```

`salt.modules.napalm_network.`**`optics`**`(`*`**kwargs`*`)`

New in version 2017.7.0.

Fetches the power usage on the various transceivers installed on the network device (in dBm), and returns a view that conforms with the OpenConfig model openconfig-platform-transceiver.yang.

Returns

Returns a dictionary where the keys are as listed below:

- **intf_name (unicode)**
  - **physical_channels**
    - **channels (list of dicts)**
      - index (int)
      - **state**

        input_power

        instant (float)

        avg (float)

        min (float)

        max (float)

        output_power

        instant (float)

        avg (float)

        min (float)

        max (float)

laser_bias_current

instant (float)

avg (float)

min (float)

max (float)

CLI Example:

```
salt '*' net.optics
```

salt.modules.napalm_network.**patch**(*patchfile*, *options=''*, *saltenv='base'*, *source_hash=None*, *show_changes=True*, *source='running'*, *path=None*, *test=False*, *commit=True*, *debug=False*, *replace=True*)

New in version 2019.2.0.

Apply a patch to the configuration source, and load the result into the running config of the device.

patchfile A patch file to apply to the configuration source.

options Options to pass to patch.

source_hash If the patch file (specified via the `patchfile` argument) is an HTTP(S) or FTP URL and the file exists in the minion's file cache, this option can be passed to keep the minion from re-downloading the file if the cached copy matches the specified hash.

show_changes: **`True`** Controls how changes are presented. If `True`, this function will return the of the changes made. If `False`, then it will return a boolean (`True` if any changes were made, otherwise False).

source: **`running`** The configuration source. Choose from: `running`, `candidate`, or `startup`. Default: `running`.

path: **`None`** Save the temporary configuration to a specific path, then read from there. This argument is optional, can the user prefers a particular location of the temporary file.

test: **`False`** Dry run? If set as `True`, will apply the config, discard and return the changes. Default: `False` and will commit the changes on the device.

commit: **`True`** Commit the configuration changes? Default: `True`.

debug: **`False`** Debug mode. Will insert a new key in the output dictionary, as `loaded_config` containing the raw configuration loaded on the device.

replace: **`True`** Load and replace the configuration. Default: `True`.

CLI Example:

```
salt '*' net.patch https://example.com/running_config.patch
```

salt.modules.napalm_network.**ping**(*destination*, *source=None*, *ttl=None*, *timeout=None*, *size=None*, *count=None*, *vrf=None*, ***kwargs*)

Executes a ping on the network device and returns a dictionary as a result.

destination Hostname or IP address of remote host

source Source address of echo request

ttl IP time-to-live value (IPv6 hop-limit value) (1..255 hops)

timeout Maximum wait time after sending final packet (seconds)

size Size of request packets (0..65468 bytes)

count Number of ping requests to send (1..2000000000 packets)

vrf VRF (routing instance) for ping attempt

New in version 2016.11.4.

CLI Example:

```
salt '*' net.ping 8.8.8.8
salt '*' net.ping 8.8.8.8 ttl=3 size=65468
salt '*' net.ping 8.8.8.8 source=127.0.0.1 timeout=1 count=100
```

`salt.modules.napalm_network.`**`replace_pattern`**(*pattern, repl, count=0, flags=8, bufsize=1, append_if_not_found=False, prepend_if_not_found=False, not_found_content=None, search_only=False, show_changes=True, backslash_literal=False, source=None, path=None, test=False, replace=True, debug=False, commit=True*)

New in version 2019.2.0.

Replace occurrences of a pattern in the configuration source. If `show_changes` is `True`, then a diff of what changed will be returned, otherwise a `True` will be returned when changes are made, and `False` when no changes are made. This is a pure Python implementation that wraps Python's `sub()`.

pattern A regular expression, to be matched using Python's `search()`.

repl The replacement text.

count: `0` Maximum number of pattern occurrences to be replaced. If count is a positive integer `n`, only `n` occurrences will be replaced, otherwise all occurrences will be replaced.

flags (list or int): `8` A list of flags defined in the `re` module documentation from the Python standard library. Each list item should be a string that will correlate to the human-friendly flag name. E.g., `['IGNORECASE', 'MULTILINE']`. Optionally, `flags` may be an int, with a value corresponding to the XOR (|) of all the desired flags. Defaults to 8 (which supports 'MULTILINE').

bufsize (int or str): `1` How much of the configuration to buffer into memory at once. The default value `1` processes one line at a time. The special value `file` may be specified which will read the entire file into memory before processing.

append_if_not_found: `False` If set to `True`, and pattern is not found, then the content will be appended to the file.

prepend_if_not_found: `False` If set to `True` and pattern is not found, then the content will be prepended to the file.

not_found_content Content to use for append/prepend if not found. If None (default), uses `repl`. Useful when `repl` uses references to group in pattern.

search_only: `False` If set to true, this no changes will be performed on the file, and this function will simply return `True` if the pattern was matched, and `False` if not.

show_changes: `True` If `True`, return a diff of changes made. Otherwise, return `True` if changes were made, and `False` if not.

backslash_literal: `False` Interpret backslashes as literal backslashes for the repl and not escape characters. This will help when using append/prepend so that the backslashes are not interpreted for the repl on the second run of the state.

source: `running` The configuration source. Choose from: `running`, `candidate`, or `startup`. Default: `running`.

path Save the temporary configuration to a specific path, then read from there.

test: `False` Dry run? If set as `True`, will apply the config, discard and return the changes. Default: `False` and will commit the changes on the device.

commit: `True` Commit the configuration changes? Default: `True`.

debug: `False` Debug mode. Will insert a new key in the output dictionary, as `loaded_config` containing the raw configuration loaded on the device.

replace: `True` Load and replace the configuration. Default: `True`.

If an equal sign (=) appears in an argument to a Salt command it is interpreted as a keyword argument in the format `key=val`. That processing can be bypassed in order to pass an equal sign through to the remote shell command by manually specifying the kwarg:

```
salt '*' net.replace_pattern "bind-address\s*=" "bind-address:"
```

CLI Example:

```
salt '*' net.replace_pattern PREFIX-LIST_NAME new-prefix-list-name
salt '*' net.replace_pattern bgp-group-name new-bgp-group-name count=1
```

`salt.modules.napalm_network.`**`rollback`**(*inherit_napalm_device=None*, ***kwargs*)

Rollbacks the configuration.

CLI Example:

```
salt '*' net.rollback
```

`salt.modules.napalm_network.`**`save_config`**(*source=None*, *path=None*)

New in version 2019.2.0.

Save the configuration to a file on the local file system.

source: **`running`** The configuration source. Choose from: `running`, `candidate`, `startup`. Default: `running`.

path Absolute path to file where to save the configuration. To push the files to the Master, use *`cp.push`* Execution function.

CLI Example:

```
salt '*' net.save_config source=running
```

`salt.modules.napalm_network.`**`traceroute`**(*destination*, *source=None*, *ttl=None*, *timeout=None*, *vrf=None*, ***kwargs*)

Calls the method traceroute from the NAPALM driver object and returns a dictionary with the result of the traceroute command executed on the device.

destination Hostname or address of remote host

source Source address to use in outgoing traceroute packets

ttl IP maximum time-to-live value (or IPv6 maximum hop-limit value)

timeout Number of seconds to wait for response (seconds)

vrf VRF (routing instance) for traceroute attempt

New in version 2016.11.4.

CLI Example:

```
salt '*' net.traceroute 8.8.8.8
salt '*' net.traceroute 8.8.8.8 source=127.0.0.1 ttl=5 timeout=1
```

21.9.268 salt.modules.napalm_ntp module

NAPALM NTP

Manages NTP on network devices.

codeauthor Mircea Ulinic <ping@mirceaulinic.net> & Jerome Fleury <jf@cloudflare.com>

maturity new

depends napalm

platform unix

Dependencies

- *`NAPALM proxy minion`*
- *`NET basic features`*

See also:

NTP peers management state

New in version 2016.11.0.

`salt.modules.napalm_ntp.`**`delete_peers`**`(`**peers*, ***options*`)`

Removes NTP peers configured on the device.

Parameters

- **`peers`** – list of IP Addresses/Domain Names to be removed as NTP peers
- **`(bool)`** (*`commit`*) – discard loaded config. By default `test` is False (will not dicard the changes)
- **`(bool)`** – commit loaded config. By default `commit` is True (will commit the changes). Useful when the user does not want to commit after each change, but after a couple.

By default this function will commit the config changes (if any). To load without committing, use the `commit` option. For a dry run, use the `test` argument.

CLI Example:

```
salt '*' ntp.delete_peers 8.8.8.8 time.apple.com
salt '*' ntp.delete_peers 172.17.17.1 test=True  # only displays the diff
salt '*' ntp.delete_peers 192.168.0.1 commit=False  # preserves the changes, but
↪does not commit
```

`salt.modules.napalm_ntp.`**`delete_servers`**`(`**servers*, ***options*`)`

Removes NTP servers configured on the device.

Parameters

- **`servers`** – list of IP Addresses/Domain Names to be removed as NTP servers
- **`(bool)`** (*`commit`*) – discard loaded config. By default `test` is False (will not dicard the changes)
- **`(bool)`** – commit loaded config. By default `commit` is True (will commit the changes). Useful when the user does not want to commit after each change, but after a couple.

By default this function will commit the config changes (if any). To load without committing, use the `commit` option. For dry run use the `test` argument.

CLI Example:

```
salt '*' ntp.delete_servers 8.8.8.8 time.apple.com
salt '*' ntp.delete_servers 172.17.17.1 test=True  # only displays the diff
salt '*' ntp.delete_servers 192.168.0.1 commit=False  # preserves the changes,
↪but does not commit
```

`salt.modules.napalm_ntp.`**`peers`**`(`***kwargs*`)`

Returns a list the NTP peers configured on the network device.

Returns configured NTP peers as list.

CLI Example:

```
salt '*' ntp.peers
```

Example output:

```
[
    '192.168.0.1',
    '172.17.17.1',
    '172.17.17.2',
```

(continues on next page)

(continued from previous page)

```
    '2400:cb00:6:1024::c71b:840a'
]
```

`salt.modules.napalm_ntp.`**`servers`**`(`*`**kwargs`*`)`

Returns a list of the configured NTP servers on the device.

CLI Example:

```
salt '*' ntp.servers
```

Example output:

```
[
    '192.168.0.1',
    '172.17.17.1',
    '172.17.17.2',
    '2400:cb00:6:1024::c71b:840a'
]
```

`salt.modules.napalm_ntp.`**`set_peers`**`(`*`*peers, **options`*`)`

Configures a list of NTP peers on the device.

Parameters

- **`peers`** – list of IP Addresses/Domain Names
- **`(bool)`** (*`test`*) – discard loaded config. By default `test` is False (will not dicard the changes)

Commit commit (bool) commit loaded config. By default `commit` is True (will commit the changes). Useful when the user does not want to commit after each change, but after a couple.

By default this function will commit the config changes (if any). To load without committing, use the *commit* option. For dry run use the *test* argument.

CLI Example:

```
salt '*' ntp.set_peers 192.168.0.1 172.17.17.1 time.apple.com
salt '*' ntp.set_peers 172.17.17.1 test=True  # only displays the diff
salt '*' ntp.set_peers 192.168.0.1 commit=False  # preserves the changes, but
↪does not commit
```

`salt.modules.napalm_ntp.`**`set_servers`**`(`*`*servers, **options`*`)`

Configures a list of NTP servers on the device.

Parameters

- **`servers`** – list of IP Addresses/Domain Names
- **`(bool)`** (*`test`*) – discard loaded config. By default `test` is False (will not dicard the changes)

Commit commit (bool) commit loaded config. By default `commit` is True (will commit the changes). Useful when the user does not want to commit after each change, but after a couple.

By default this function will commit the config changes (if any). To load without committing, use the *commit* option. For dry run use the *test* argument.

CLI Example:

```
salt '*' ntp.set_servers 192.168.0.1 172.17.17.1 time.apple.com
salt '*' ntp.set_servers 172.17.17.1 test=True  # only displays the diff
salt '*' ntp.set_servers 192.168.0.1 commit=False  # preserves the changes, but
↪does not commit
```

`salt.modules.napalm_ntp.`**`stats`**`(`*peer=None*, ***kwargs*`)`

Returns a dictionary containing synchronization details of the NTP peers.

Parameters **`peer`** – Returns only the details of a specific NTP peer.

Returns

a list of dictionaries, with the following keys:

- remote
- referenceid
- synchronized
- stratum
- type
- when
- hostpoll
- reachability
- delay
- offset
- jitter

CLI Example:

```
salt '*' ntp.stats
```

Example output:

```
[
    {
        'remote'         : '188.114.101.4',
        'referenceid'    : '188.114.100.1',
        'synchronized'   : True,
        'stratum'        : 4,
        'type'           : '-',
        'when'           : '107',
        'hostpoll'       : 256,
        'reachability'   : 377,
        'delay'          : 164.228,
        'offset'         : -13.866,
        'jitter'         : 2.695
    }
]
```

21.9.269 salt.modules.napalm_probes module

NAPALM Probes

Manages RPM/SLA probes on the network device.

codeauthor Mircea Ulinic <ping@mirceaulinic.net> & Jerome Fleury <jf@cloudflare.com>

maturity new

depends napalm

platform unix

Dependencies

- *`napalm proxy minion`*

- *NET basic features*

See also:

Probes configuration management state

New in version 2016.11.0.

`salt.modules.napalm_probes.`**`config`**`(`*`**kwargs`*`)`

Returns the configuration of the RPM probes.

Returns A dictionary containing the configuration of the RPM/SLA probes.

CLI Example:

```
salt '*' probes.config
```

Output Example:

```
{
    'probe1':{
        'test1': {
            'probe_type'   : 'icmp-ping',
            'target'       : '192.168.0.1',
            'source'       : '192.168.0.2',
            'probe_count'  : 13,
            'test_interval': 3
        },
        'test2': {
            'probe_type'   : 'http-ping',
            'target'       : '172.17.17.1',
            'source'       : '192.17.17.2',
            'probe_count'  : 5,
            'test_interval': 60
        }
    }
}
```

`salt.modules.napalm_probes.`**`delete_probes`**`(`*`probes, test=False, commit=True, **kwargs`*`)`

Removes RPM/SLA probes from the network device. Calls the configuration template 'delete_probes' from the NAPALM library, providing as input a rich formatted dictionary with the configuration details of the probes to be removed from the configuration of the device.

Parameters

- **`probes`** – Dictionary with a similar format as the output dictionary of the function config(), where the details are not necessary.
- **`test`** – Dry run? If set as True, will apply the config, discard and return the changes. Default: False
- **`commit`** – Commit? (default: True) Sometimes it is not needed to commit the config immediately after loading the changes. E.g.: a state loads a couple of parts (add / remove / update) and would not be optimal to commit after each operation. Also, from the CLI when the user needs to apply the similar changes before committing, can specify commit=False and will not discard the config.

Raises **`MergeConfigException`** – If there is an error on the configuration sent.

Returns A dictionary having the following keys:

- result (bool): if the config was applied successfully. It is *False* only in case of failure. In case there are no changes to be applied and successfully performs all operations it is still *True* and so will be the *already_configured* flag (example below)
- comment (str): a message for the user
- already_configured (bool): flag to check if there were no changes applied

- diff (str): returns the config changes applied

Input example:

```
probes = {
    'existing_probe':{
        'existing_test1': {},
        'existing_test2': {}
    }
}
```

`salt.modules.napalm_probes.`**`results`**`(`***kwargs*`)`

Provides the results of the measurements of the RPM/SLA probes.

:return a dictionary with the results of the probes.

CLI Example:

```
salt '*' probes.results
```

Output example:

```
{
    'probe1':  {
        'test1': {
            'last_test_min_delay'   : 63.120,
            'global_test_min_delay' : 62.912,
            'current_test_avg_delay': 63.190,
            'global_test_max_delay' : 177.349,
            'current_test_max_delay': 63.302,
            'global_test_avg_delay' : 63.802,
            'last_test_avg_delay'   : 63.438,
            'last_test_max_delay'   : 65.356,
            'probe_type'            : 'icmp-ping',
            'rtt'                   : 63.138,
            'last_test_loss'        : 0,
            'round_trip_jitter'     : -59.0,
            'target'                : '192.168.0.1',
            'source'                : '192.168.0.2',
            'probe_count'           : 15,
            'current_test_min_delay': 63.138
        },
        'test2': {
            'last_test_min_delay'   : 176.384,
            'global_test_min_delay' : 169.226,
            'current_test_avg_delay': 177.098,
            'global_test_max_delay' : 292.628,
            'current_test_max_delay': 180.055,
            'global_test_avg_delay' : 177.959,
            'last_test_avg_delay'   : 177.178,
            'last_test_max_delay'   : 184.671,
            'probe_type'            : 'icmp-ping',
            'rtt'                   : 176.449,
            'last_test_loss'        : 0,
            'round_trip_jitter'     : -34.0,
            'target'                : '172.17.17.1',
            'source'                : '172.17.17.2',
            'probe_count'           : 15,
            'current_test_min_delay': 176.402
```

(continues on next page)

(continued from previous page)

```
        }
    }
}
```

`salt.modules.napalm_probes.`**`schedule_probes`**(*probes*, *test=False*, *commit=True*, ***kwargs*)

Will schedule the probes. On Cisco devices, it is not enough to define the probes, it is also necessary to schedule them.

This function calls the configuration template `schedule_probes` from the NAPALM library, providing as input a rich formatted dictionary with the names of the probes and the tests to be scheduled.

Parameters

- **`probes`** – Dictionary with a similar format as the output dictionary of the function config(), where the details are not necessary.
- **`test`** – Dry run? If set as True, will apply the config, discard and return the changes. Default: False
- **`commit`** – Commit? (default: True) Sometimes it is not needed to commit the config immediately after loading the changes. E.g.: a state loads a couple of parts (add / remove / update) and would not be optimal to commit after each operation. Also, from the CLI when the user needs to apply the similar changes before committing, can specify commit=False and will not discard the config.

Raises **`MergeConfigException`** – If there is an error on the configuration sent.

Returns a dictionary having the following keys:

- result (bool): if the config was applied successfully. It is *False* only in case of failure. In case there are no changes to be applied and successfully performs all operations it is still *True* and so will be the *already_configured* flag (example below)
- comment (str): a message for the user
- already_configured (bool): flag to check if there were no changes applied
- diff (str): returns the config changes applied

Input example:

```
probes = {
    'new_probe':{
        'new_test1': {},
        'new_test2': {}
    }
}
```

`salt.modules.napalm_probes.`**`set_probes`**(*probes*, *test=False*, *commit=True*, ***kwargs*)

Configures RPM/SLA probes on the device. Calls the configuration template 'set_probes' from the NAPALM library, providing as input a rich formatted dictionary with the configuration details of the probes to be configured.

Parameters

- **`probes`** – Dictionary formatted as the output of the function config()
- **`test`** – Dry run? If set as True, will apply the config, discard and return the changes. Default: False
- **`commit`** – Commit? (default: True) Sometimes it is not needed to commit the config immediately after loading the changes. E.g.: a state loads a couple of parts (add / remove / update) and would not be optimal to commit after each operation. Also, from the CLI when the user needs to apply the similar changes before committing, can specify commit=False and will not discard the config.

Raises **`MergeConfigException`** – If there is an error on the configuration sent.

Return a dictionary having the following keys

- result (bool): if the config was applied successfully. It is *False* only in case of failure.

In case there are no changes to be applied and successfully performs all operations it is still *True* and so will be the *already_configured* flag (example below)

- comment (str): a message for the user
- already_configured (bool): flag to check if there were no changes applied
- diff (str): returns the config changes applied

Input example - via state/script:

```
probes = {
    'new_probe':{
        'new_test1': {
            'probe_type'   : 'icmp-ping',
            'target'       : '192.168.0.1',
            'source'       : '192.168.0.2',
            'probe_count'  : 13,
            'test_interval': 3
        },
        'new_test2': {
            'probe_type'   : 'http-ping',
            'target'       : '172.17.17.1',
            'source'       : '192.17.17.2',
            'probe_count'  : 5,
            'test_interval': 60
        }
    }
}
set_probes(probes)
```

CLI Example - to push cahnges on the fly (not recommended):

```
salt 'junos_minion' probes.set_probes "{'new_probe':{'new_test1':{'probe_type':
↪'icmp-ping',            'target':'192.168.0.1','source':'192.168.0.2','probe_
↪count':13,'test_interval':3}}}" test=True
```

Output example - for the CLI example above:

```
junos_minion:
    ----------
    already_configured:
        False
    comment:
        Configuration discarded.
    diff:
        [edit services rpm]
             probe transit { ... }
        +    probe new_probe {
        +        test new_test1 {
        +            probe-type icmp-ping;
        +            target address 192.168.0.1;
        +            probe-count 13;
        +            test-interval 3;
        +            source-address 192.168.0.2;
        +        }
        +    }
    result:
        True
```

21.9.270 salt.modules.napalm_route module

NAPALM Route

Retrieves route details from network devices.

codeauthor Mircea Ulinic <ping@mirceaulinic.net>

maturity new

depends napalm

platform unix

Dependencies

- *NAPALM proxy minion*

New in version 2016.11.0.

`salt.modules.napalm_route.`**`show`**`(`*destination*, *protocol=None*, ***kwargs*`)`

Displays all details for a certain route learned via a specific protocol. If the protocol is not specified, will return all possible routes.

Note: This function return the routes from the RIB. In case the destination prefix is too short, there may be too many routes matched. Therefore in cases of devices having a very high number of routes it may be necessary to adjust the prefix length and request using a longer prefix.

---

destination destination prefix.
protocol (optional) protocol used to learn the routes to the destination.

Changed in version 2017.7.0.

CLI Example:

```
salt 'my_router' route.show 172.16.0.0/25
salt 'my_router' route.show 172.16.0.0/25 bgp
```

Output example:

```
{
    '172.16.0.0/25': [
        {
            'protocol': 'BGP',
            'last_active': True,
            'current_active': True,
            'age': 1178693,
            'routing_table': 'inet.0',
            'next_hop': '192.168.0.11',
            'outgoing_interface': 'xe-1/1/1.100',
            'preference': 170,
            'selected_next_hop': False,
            'protocol_attributes': {
                'remote_as': 65001,
                'metric': 5,
                'local_as': 13335,
                'as_path': '',
```

(continues on next page)

(continued from previous page)

```
                'remote_address': '192.168.0.11',
                'metric2': 0,
                'local_preference': 0,
                'communities': [
                    '0:2',
                    'no-export'
                ],
                'preference2': -1
            },
            'inactive_reason': ''
        },
        {
            'protocol': 'BGP',
            'last_active': False,
            'current_active': False,
            'age': 2359429,
            'routing_table': 'inet.0',
            'next_hop': '192.168.0.17',
            'outgoing_interface': 'xe-1/1/1.100',
            'preference': 170,
            'selected_next_hop': True,
            'protocol_attributes': {
                'remote_as': 65001,
                'metric': 5,
                'local_as': 13335,
                'as_path': '',
                'remote_address': '192.168.0.17',
                'metric2': 0,
                'local_preference': 0,
                'communities': [
                    '0:3',
                    'no-export'
                ],
                'preference2': -1
            },
            'inactive_reason': 'Not Best in its group - Router ID'
        }
    ]
}
```

21.9.271 salt.modules.napalm_snmp module

NAPALM SNMP

Manages SNMP on network devices.

codeauthor Mircea Ulinic <ping@mirceaulinic.net>

maturity new

depends napalm

platform unix

Dependencies

- *NAPALM proxy minion*
- *NET basic features*

See also:

SNMP configuration management state

New in version 2016.11.0.

`salt.modules.napalm_snmp.`**`config`**`(`***kwargs*`)`

Returns the SNMP configuration

CLI Example:

```
salt '*' snmp.config
```

`salt.modules.napalm_snmp.`**`remove_config`**`(`*chassis_id=None, community=None, contact=None, location=None, test=False, commit=True, **kwargs*`)`

Removes a configuration element from the SNMP configuration.

Parameters

- **`chassis_id`** – (optional) Chassis ID
- **`community`** – (optional) A dictionary having the following optional keys:

- acl (if any policy / ACL need to be set)
- mode: rw or ro. Default: ro

Parameters

- **`contact`** – Contact details
- **`location`** – Location
- **`test`** – Dry run? If set as True, will apply the config, discard and return the changes. Default: False
- **`commit`** – Commit? (default: True) Sometimes it is not needed to commit the config immediately after loading the changes. E.g.: a state loads a couple of parts (add / remove / update) and would not be optimal to commit after each operation. Also, from the CLI when the user needs to apply the similar changes before committing, can specify commit=False and will not discard the config.

Raises **`MergeConfigException`** – If there is an error on the configuration sent.

Returns A dictionary having the following keys:

- result (bool): if the config was applied successfully. It is *False* only in case of failure. In case there are no changes to be applied and successfully performs all operations it is still *True* and so will be the *already_configured* flag (example below)
- comment (str): a message for the user
- already_configured (bool): flag to check if there were no changes applied
- diff (str): returns the config changes applied

CLI Example:

```
salt '*' snmp.remove_config community='abcd'
```

`salt.modules.napalm_snmp.`**`update_config`**`(`*chassis_id=None, community=None, contact=None, location=None, test=False, commit=True, **kwargs*`)`

Updates the SNMP configuration.

Parameters

- **`chassis_id`** – (optional) Chassis ID
- **`community`** – (optional) A dictionary having the following optional keys:

- acl (if any policy / ACL need to be set)
- mode: rw or ro. Default: ro

Parameters

- **`contact`** – Contact details
- **`location`** – Location
- **`test`** – Dry run? If set as True, will apply the config, discard and return the changes. Default: False
- **`commit`** – Commit? (default: True) Sometimes it is not needed to commit the config immediately after loading the changes. E.g.: a state loads a couple of parts (add / remove / update) and would not be optimal to commit after each operation. Also, from the CLI when the user needs to apply the similar changes before committing, can specify commit=False and will not discard the config.

Raises **`MergeConfigException`** – If there is an error on the configuration sent.

Return a dictionary having the following keys

- result (bool): if the config was applied successfully. It is *False* only in case of failure. In case there are no changes to be applied and successfully performs all operations it is still *True* and so will be the *already_configured* flag (example below)
- comment (str): a message for the user
- already_configured (bool): flag to check if there were no changes applied
- diff (str): returns the config changes applied

CLI Example:

```
salt 'edge01.lon01' snmp.update_config location="Greenwich, UK" test=True
```

Output example (for the CLI example above):

```
edge01.lon01:
    ----------
    already_configured:
        False
    comment:
        Configuration discarded.
    diff:
        [edit snmp]
        -  location "London, UK";
        +  location "Greenwich, UK";
    result:
        True
```

21.9.272 salt.modules.napalm_users module

NAPALM Users

Manages the configuration of the users on network devices.

codeauthor Mircea Ulinic <ping@mirceaulinic.net>

maturity new

depends napalm

platform unix

Dependencies

- *NAPALM proxy minion*

See also:

Users management state

New in version 2016.11.0.

`salt.modules.napalm_users.`**`config`**(***kwargs*)

Returns the configuration of the users on the device

CLI Example:

```
salt '*' users.config
```

Output example:

```
{
    'mircea': {
        'level': 15,
        'password': '$1$0P70xKPa$4jt5/10cBTckk6I/w/',
        'sshkeys': [
            'ssh-rsa⊠
↪AAAAB3NzaC1yc2EAAAADAQABAAABAQC4pFn+shPwTb2yELO4L7NtQrKOJXNeCl1je           ⊠
↪     l9STXVaGnRAnuc2PXl35vnWmcUq6YbUEcgUTRzzXfmelJKuVJTJIlMXii7h2xkbQp0YZIEs4P ⊠
↪                  8ipwnRBAxFfk/ZcDsN3mjep4/yjN56ejk345jhk345jk345jk341p3A/
↪9LIL7l6YewLBCwJj6                   D+fWSJ0/
↪YW+7oH17Fk2HH+tw0L5PcWLHkwA4t60iXn16qDbIk/ze6jv2hDGdCdz7oYQeCE55C           ⊠
↪     CHOHMJWYfN3jcL4s0qv8/u6Ka1FVkV7iMmro7ChThoV/5snI4Ljf2wKqgHH7TfNaCfpU0WvHA ⊠
↪                  nTs8zhOrGScSrtb mircea@master-roshi'
        ]
    }
}
```

`salt.modules.napalm_users.`**`delete_users`**(*users*, *test=False*, *commit=True*, ***kwargs*)

Removes users from the configuration of network devices.

Parameters

- **`users`** – Dictionary formatted as the output of the function config()
- **`test`** – Dry run? If set as True, will apply the config, discard and return the changes. Default: False
- **`commit`** – Commit? (default: True) Sometimes it is not needed to commit the config immediately after loading the changes. E.g.: a state loads a couple of parts (add / remove / update) and would not be optimal to commit after each operation. Also, from the CLI when the user needs to apply the similar changes before committing, can specify commit=False and will not discard the config.

Raises **`MergeConfigException`** – If there is an error on the configuration sent.

Return a dictionary having the following keys

- result (bool): if the config was applied successfully. It is *False* only in case of failure. In case there are no changes to be applied and successfully performs all operations it is still *True* and so will be the *already_configured* flag (example below)
- comment (str): a message for the user
- already_configured (bool): flag to check if there were no changes applied
- diff (str): returns the config changes applied

CLI Example:

```
salt '*' users.delete_users "{'mircea': {}}"
```

`salt.modules.napalm_users.`**`set_users`**(*users*, *test=False*, *commit=True*, ***kwargs*)

Configures users on network devices.

Parameters

- **`users`** – Dictionary formatted as the output of the function config()
- **`test`** – Dry run? If set as True, will apply the config, discard and return the changes. Default: False
- **`commit`** – Commit? (default: True) Sometimes it is not needed to commit the config immediately after loading the changes. E.g.: a state loads a couple of parts (add / remove / update) and would not be optimal to commit after each operation. Also, from the CLI when the user needs to apply the similar changes before committing, can specify commit=False and will not discard the config.

Raises **`MergeConfigException`** – If there is an error on the configuration sent.

Return a dictionary having the following keys

- result (bool): if the config was applied successfully. It is *False* only in case of failure. In case there are no changes to be applied and successfully performs all operations it is still *True* and so will be the *already_configured* flag (example below)
- comment (str): a message for the user
- already_configured (bool): flag to check if there were no changes applied
- diff (str): returns the config changes applied

CLI Example:

```
salt '*' users.set_users "{'mircea': {}}"
```

21.9.273 salt.modules.napalm_yang_mod module

NAPALM YANG

NAPALM YANG basic operations.

New in version 2017.7.0.

`salt.modules.napalm_yang_mod.`**`compliance_report`**(*data*, **models*, ***kwargs*)

Return the compliance report using YANG objects.

data Dictionary structured with respect to the models referenced.

models A list of models to be used when generating the config.

filepath The absolute path to the validation file.

CLI Example:

```
salt '*' napalm_yang.compliance_report {} models.openconfig_interfaces filepath=~/
↪validate.yml
```

Output Example:

```
{
  "skipped": [],
  "complies": true,
  "get_interfaces_ip": {
    "missing": [],
    "complies": true,
    "present": {
      "ge-0/0/0.0": {
```

(continues on next page)

(continued from previous page)

```
        "complies": true,
        "nested": true
      }
    },
    "extra": []
  }
}
```

salt.modules.napalm_yang_mod.**diff**(*candidate*, *running*, **models*)

Returns the difference between two configuration entities structured according to the YANG model.

Note: This function is recommended to be used mostly as a state helper.

---

candidate First model to compare.
running Second model to compare.
models A list of models to be used when comparing.

CLI Example:

```
salt '*' napalm_yang.diff {} {} models.openconfig_interfaces
```

Output Example:

```
{
    "interfaces": {
        "interface": {
            "both": {
                "Port-Channel1": {
                    "config": {
                        "mtu": {
                            "first": "0",
                            "second": "9000"
                        }
                    }
                }
            },
            "first_only": [
                "Loopback0"
            ],
            "second_only": [
                "Loopback1"
            ]
        }
    }
}
```

salt.modules.napalm_yang_mod.**get_config**(*data*, **models*, ***kwargs*)

Return the native config.

data Dictionary structured with respect to the models referenced.
models A list of models to be used when generating the config.
profiles: `None` Use certain profiles to generate the config. If not specified, will use the platform default profile(s).

CLI Example:

```
salt '*' napalm_yang.get_config {} models.openconfig_interfaces
```

Output Example:

```
interface et1
    ip address 192.168.1.1/24
    description Uplink1
    mtu 9000
interface et2
    ip address 192.168.2.1/24
    description Uplink2
    mtu 9000
```

`salt.modules.napalm_yang_mod.`**`load_config`**(*data*, **models*, ***kwargs*)

Generate and load the config on the device using the OpenConfig or IETF models and device profiles.

data Dictionary structured with respect to the models referenced.

models A list of models to be used when generating the config.

profiles: **`None`** Use certain profiles to generate the config. If not specified, will use the platform default profile(s).

test: **`False`** Dry run? If set as `True`, will apply the config, discard and return the changes. Default: `False` and will commit the changes on the device.

commit: **`True`** Commit? Default: `True`.

debug: **`False`** Debug mode. Will insert a new key under the output dictionary, as `loaded_config` containing the raw configuration loaded on the device.

replace: **`False`** Should replace the config with the new generate one?

CLI Example:

```
salt '*' napalm_yang.load_config {} models.openconfig_interfaces test=True
↪debug=True
```

Output Example:

```
device1:
    ----------
    already_configured:
        False
    comment:
    diff:
        [edit interfaces ge-0/0/0]
        -   mtu 1400;
        [edit interfaces ge-0/0/0 unit 0 family inet]
        -       dhcp;
        [edit interfaces lo0]
        -    unit 0 {
        -        description lo0.0;
        -    }
        +    unit 1 {
        +        description "new loopback";
        +    }
    loaded_config:
        <configuration>
          <interfaces replace="replace">
            <interface>
              <name>ge-0/0/0</name>
              <unit>
                <name>0</name>
```

(continues on next page)

(continued from previous page)

```
                <family>
                  <inet/>
                </family>
                <description>ge-0/0/0.0</description>
              </unit>
              <description>management interface</description>
            </interface>
            <interface>
              <name>ge-0/0/1</name>
              <disable/>
              <description>ge-0/0/1</description>
            </interface>
            <interface>
              <name>ae0</name>
              <unit>
                <name>0</name>
                <vlan-id>100</vlan-id>
                <family>
                  <inet>
                    <address>
                      <name>192.168.100.1/24</name>
                    </address>
                    <address>
                      <name>172.20.100.1/24</name>
                    </address>
                  </inet>
                </family>
                <description>a description</description>
              </unit>
              <vlan-tagging/>
              <unit>
                <name>1</name>
                <vlan-id>1</vlan-id>
                <family>
                  <inet>
                    <address>
                      <name>192.168.101.1/24</name>
                    </address>
                  </inet>
                </family>
                <disable/>
                <description>ae0.1</description>
              </unit>
              <vlan-tagging/>
              <unit>
                <name>2</name>
                <vlan-id>2</vlan-id>
                <family>
                  <inet>
                    <address>
                      <name>192.168.102.1/24</name>
                    </address>
                  </inet>
                </family>
                <description>ae0.2</description>
              </unit>
```

(continues on next page)

(continued from previous page)

```
                <vlan-tagging/>
              </interface>
              <interface>
                <name>lo0</name>
                <unit>
                  <name>1</name>
                  <description>new loopback</description>
                </unit>
                <description>lo0</description>
              </interface>
            </interfaces>
          </configuration>
    result:
        True
```

salt.modules.napalm_yang_mod.**parse**(**models*, ***kwargs*)

Parse configuration from the device.

models A list of models to be used when parsing.

config: False Parse config.

state: False Parse state.

profiles: None Use certain profiles to parse. If not specified, will use the device default profile(s).

CLI Example:

```
salt '*' napalm_yang.parse models.openconfig_interfaces
```

Output Example:

```
{
    "interfaces": {
        "interface": {
            ".local.": {
                "name": ".local.",
                "state": {
                    "admin-status": "UP",
                    "counters": {
                        "in-discards": 0,
                        "in-errors": 0,
                        "out-errors": 0
                    },
                    "enabled": True,
                    "ifindex": 0,
                    "last-change": 0,
                    "oper-status": "UP",
                    "type": "softwareLoopback"
                },
                "subinterfaces": {
                    "subinterface": {
                        ".local..0": {
                            "index": ".local..0",
                            "state": {
                                "ifindex": 0,
                                "name": ".local..0"
                            }
                        }
                    }
                }
```

(continues on next page)

(continued from previous page)

```
            },
            "ae0": {
                "name": "ae0",
                "state": {
                    "admin-status": "UP",
                    "counters": {
                        "in-discards": 0,
                        "in-errors": 0,
                        "out-errors": 0
                    },
                    "enabled": True,
                    "ifindex": 531,
                    "last-change": 255203,
                    "mtu": 1518,
                    "oper-status": "DOWN"
                },
                "subinterfaces": {
                    "subinterface": {
                        "ae0.0": {
                            "index": "ae0.0",
                            "state": {
                                "description": "ASDASDASD",
                                "ifindex": 532,
                                "name": "ae0.0"
                            }
                        }
                        "ae0.32767": {
                            "index": "ae0.32767",
                            "state": {
                                "ifindex": 535,
                                "name": "ae0.32767"
                            }
                        }
                    }
                }
            },
            "dsc": {
                "name": "dsc",
                "state": {
                    "admin-status": "UP",
                    "counters": {
                        "in-discards": 0,
                        "in-errors": 0,
                        "out-errors": 0
                    },
                    "enabled": True,
                    "ifindex": 5,
                    "last-change": 0,
                    "oper-status": "UP"
                }
            },
            "ge-0/0/0": {
                "name": "ge-0/0/0",
                "state": {
                    "admin-status": "UP",
                    "counters": {
```

(continues on next page)

(continued from previous page)

```
                    "in-broadcast-pkts": 0,
                    "in-discards": 0,
                    "in-errors": 0,
                    "in-multicast-pkts": 0,
                    "in-unicast-pkts": 16877,
                    "out-broadcast-pkts": 0,
                    "out-errors": 0,
                    "out-multicast-pkts": 0,
                    "out-unicast-pkts": 15742
                },
                "description": "management interface",
                "enabled": True,
                "ifindex": 507,
                "last-change": 258467,
                "mtu": 1400,
                "oper-status": "UP"
            },
            "subinterfaces": {
                "subinterface": {
                    "ge-0/0/0.0": {
                        "index": "ge-0/0/0.0",
                        "state": {
                            "description": "ge-0/0/0.0",
                            "ifindex": 521,
                            "name": "ge-0/0/0.0"
                        }
                    }
                }
            }
        }
        "irb": {
            "name": "irb",
            "state": {
                "admin-status": "UP",
                "counters": {
                    "in-discards": 0,
                    "in-errors": 0,
                    "out-errors": 0
                },
                "enabled": True,
                "ifindex": 502,
                "last-change": 0,
                "mtu": 1514,
                "oper-status": "UP",
                "type": "ethernetCsmacd"
            }
        },
        "lo0": {
            "name": "lo0",
            "state": {
                "admin-status": "UP",
                "counters": {
                    "in-discards": 0,
                    "in-errors": 0,
                    "out-errors": 0
                },
```

(continues on next page)

(continued from previous page)

```
                    "description": "lo0",
                    "enabled": True,
                    "ifindex": 6,
                    "last-change": 0,
                    "oper-status": "UP",
                    "type": "softwareLoopback"
                },
                "subinterfaces": {
                    "subinterface": {
                        "lo0.0": {
                            "index": "lo0.0",
                            "state": {
                                "description": "lo0.0",
                                "ifindex": 16,
                                "name": "lo0.0"
                            }
                        },
                        "lo0.16384": {
                            "index": "lo0.16384",
                            "state": {
                                "ifindex": 21,
                                "name": "lo0.16384"
                            }
                        },
                        "lo0.16385": {
                            "index": "lo0.16385",
                            "state": {
                                "ifindex": 22,
                                "name": "lo0.16385"
                            }
                        },
                        "lo0.32768": {
                            "index": "lo0.32768",
                            "state": {
                                "ifindex": 248,
                                "name": "lo0.32768"
                            }
                        }
                    }
                }
            }
        }
    }
}
```

21.9.274 salt.modules.netaddress

Module for getting information about network addresses.

New in version 2016.3.0.

depends netaddr

salt.modules.netaddress.**cidr_broadcast**(*cidr*)

Get the broadcast address associated with a CIDR address.

CLI example:

```
salt myminion netaddress.cidr_netmask 192.168.0.0/20
```

`salt.modules.netaddress.`**`cidr_netmask`**`(`*cidr*`)`

Get the netmask address associated with a CIDR address.

CLI example:

```
salt myminion netaddress.cidr_netmask 192.168.0.0/20
```

`salt.modules.netaddress.`**`list_cidr_ips`**`(`*cidr*`)`

Get a list of IP addresses from a CIDR.

CLI example:

```
salt myminion netaddress.list_cidr_ips 192.168.0.0/20
```

`salt.modules.netaddress.`**`list_cidr_ips_ipv6`**`(`*cidr*`)`

Get a list of IPv6 addresses from a CIDR.

CLI example:

```
salt myminion netaddress.list_cidr_ips_ipv6 192.168.0.0/20
```

21.9.275 salt.modules.netbox module

NetBox

Module to query NetBox

codeauthor Zach Moody <zmoody@do.co>

maturity new

depends pynetbox

The following config should be in the minion config file. In order to work with `secrets` you should provide a token and path to your private key file:

```
netbox:
  url: <NETBOX_URL>
  token: <NETBOX_USERNAME_API_TOKEN (OPTIONAL)>
  keyfile: </PATH/TO/NETBOX/KEY (OPTIONAL)>
```

New in version 2018.3.0.

`salt.modules.netbox.`**`create_circuit`**`(`*name, provider_id, circuit_type, description=None*`)`

New in version 2019.2.0.

Create a new Netbox circuit
name Name of the circuit
provider_id The netbox id of the circuit provider
circuit_type The name of the circuit type
asn The ASN of the circuit provider
description The description of the circuit
CLI Example:

```
salt myminion netbox.create_circuit NEW_CIRCUIT_01 Telia Transit 1299 "New Telia
↪circuit"
```

`salt.modules.netbox.`**`create_circuit_provider`**(*name*, *asn=None*)

New in version 2019.2.0.

Create a new Netbox circuit provider
name The name of the circuit provider
asn The ASN of the circuit provider
CLI Example:

```
salt myminion netbox.create_circuit_provider Telia 1299
```

`salt.modules.netbox.`**`create_circuit_termination`**(*circuit*, *interface*, *device*, *speed*, *xconnect_id=None*, *term_side='A'*)

New in version 2019.2.0.

Terminate a circuit on an interface
circuit The name of the circuit
interface The name of the interface to terminate on
device The name of the device the interface belongs to
speed The speed of the circuit, in Kbps
xconnect_id The cross-connect identifier
term_side The side of the circuit termination
CLI Example:

```
salt myminion netbox.create_circuit_termination NEW_CIRCUIT_01 xe-0/0/1 myminion␣
↪10000 xconnect_id=XCON01
```

`salt.modules.netbox.`**`create_circuit_type`**(*name*)

New in version 2019.2.0.

Create a new Netbox circuit type.
name The name of the circuit type
CLI Example:

```
salt myminion netbox.create_circuit_type Transit
```

`salt.modules.netbox.`**`create_device`**(*name*, *role*, *model*, *manufacturer*, *site*)

New in version 2019.2.0.

Create a new device with a name, role, model, manufacturer and site. All these components need to be already in Netbox.
name The name of the device, e.g., `edge_router`
role String of device role, e.g., `router`
model String of device model, e.g., `MX480`
manufacturer String of device manufacturer, e.g., `Juniper`
site String of device site, e.g., `BRU`
CLI Example:

```
salt myminion netbox.create_device edge_router router MX480 Juniper BRU
```

`salt.modules.netbox.`**`create_device_role`**(*role*, *color*)

New in version 2019.2.0.

Create a device role
role String of device role, e.g., `router`
CLI Example:

```
salt myminion netbox.create_device_role router
```

`salt.modules.netbox.`**`create_device_type`**(*model*, *manufacturer*)

New in version 2019.2.0.

Create a device type. If the manufacturer doesn't exist, create a new manufacturer.

model String of device model, e.g., `MX480`

manufacturer String of device manufacturer, e.g., `Juniper`

CLI Example:

```
salt myminion netbox.create_device_type MX480 Juniper
```

`salt.modules.netbox.`**`create_interface`**(*device_name*, *interface_name*, *mac_address=None*, *description=None*, *enabled=None*, *lag=None*, *lag_parent=None*, *form_factor=None*)

New in version 2019.2.0.

Attach an interface to a device. If not all arguments are provided, they will default to Netbox defaults.

device_name The name of the device, e.g., `edge_router`

interface_name The name of the interface, e.g., `TenGigE0/0/0/0`

mac_address String of mac address, e.g., `50:87:89:73:92:C8`

description String of interface description, e.g., `NTT`

enabled String of boolean interface status, e.g., `True`

lag: Boolean of interface lag status, e.g., `True`

lag_parent String of interface lag parent name, e.g., `ae13`

form_factor Integer of form factor id, obtained through _choices API endpoint, e.g., `200`

CLI Example:

```
salt myminion netbox.create_interface edge_router ae13 description="Core uplink"
```

`salt.modules.netbox.`**`create_interface_connection`**(*interface_a*, *interface_b*)

New in version 2019.2.0.

Create an interface connection between 2 interfaces

interface_a Interface id for Side A

interface_b Interface id for Side B

CLI Example:

```
salt myminion netbox.create_interface_connection 123 456
```

`salt.modules.netbox.`**`create_inventory_item`**(*device_name*, *item_name*, *manufacturer_name=None*, *serial=''*, *part_id=''*, *description=''*)

New in version 2019.2.0.

Add an inventory item to an existing device.

device_name The name of the device, e.g., `edge_router`.

item_name String of inventory item name, e.g., `Transceiver`.

manufacturer_name String of inventory item manufacturer, e.g., `Fiberstore`.

serial String of inventory item serial, e.g., `FS1238931`.

part_id String of inventory item part id, e.g., `740-01234`.

description String of inventory item description, e.g., `SFP+-10G-LR`.

CLI Example:

```
salt myminion netbox.create_inventory_item edge_router Transceiver part_id=740-
↪01234
```

`salt.modules.netbox.`**`create_ipaddress`**(*ip_address*, *family*, *device=None*, *interface=None*)

New in version 2019.2.0.

Add an IP address, and optionally attach it to an interface.
ip_address The IP address and CIDR, e.g., `192.168.1.1/24`
family Integer of IP family, e.g., `4`
device The name of the device to attach IP to, e.g., `edge_router`
interface The name of the interface to attach IP to, e.g., `ae13`
CLI Example:

```
salt myminion netbox.create_ipaddress 192.168.1.1/24 4 device=edge_router␣
↪interface=ae13
```

`salt.modules.netbox.`**`create_manufacturer`**`(`*name*`)`
New in version 2019.2.0.

Create a device manufacturer.
name The name of the manufacturer, e.g., `Juniper`
CLI Example:

```
salt myminion netbox.create_manufacturer Juniper
```

`salt.modules.netbox.`**`create_platform`**`(`*platform*`)`
New in version 2019.2.0.

Create a new device platform
platform String of device platform, e.g., `junos`
CLI Example:

```
salt myminion netbox.create_platform junos
```

`salt.modules.netbox.`**`create_site`**`(`*site*`)`
New in version 2019.2.0.

Create a new device site
site String of device site, e.g., `BRU`
CLI Example:

```
salt myminion netbox.create_site BRU
```

`salt.modules.netbox.`**`delete_interface`**`(`*device_name, interface_name*`)`
New in version 2019.2.0.

Delete an interface from a device.
device_name The name of the device, e.g., `edge_router`.
interface_name The name of the interface, e.g., `ae13`
CLI Example:

```
salt myminion netbox.delete_interface edge_router ae13
```

`salt.modules.netbox.`**`delete_inventory_item`**`(`*item_id*`)`
New in version 2019.2.0.

Remove an item from a devices inventory. Identified by the netbox id
item_id Integer of item to be deleted
CLI Example:

```
salt myminion netbox.delete_inventory_item 1354
```

`salt.modules.netbox.`**`delete_ipaddress`**`(`*ipaddr_id*`)`
New in version 2019.2.0.

Delete an IP address. IP addresses in Netbox are a combination of address and the interface it is assigned to.
id The Netbox id for the IP address.
CLI Example:

```
salt myminion netbox.delete_ipaddress 9002
```

`salt.modules.netbox.`**`filter_`**`(`*app*, *endpoint*, ***kwargs*`)`
Get a list of items from NetBox.
app String of netbox app, e.g., `dcim`, `circuits`, `ipam`
endpoint String of app endpoint, e.g., `sites`, `regions`, `devices`
kwargs Optional arguments that can be used to filter. All filter keywords are available in Netbox, which can be found by surfing to the corresponding API endpoint, and clicking Filters. e.g., `role=router`
Returns a list of dictionaries

```
salt myminion netbox.filter dcim devices status=1 role=router
```

`salt.modules.netbox.`**`get_`**`(`*app*, *endpoint*, *id=None*, ***kwargs*`)`
Get a single item from NetBox.
app String of netbox app, e.g., `dcim`, `circuits`, `ipam`
endpoint String of app endpoint, e.g., `sites`, `regions`, `devices`
Returns a single dictionary

To get an item based on ID.

```
salt myminion netbox.get dcim devices id=123
```

Or using named arguments that correspond with accepted filters on the NetBox endpoint.

```
salt myminion netbox.get dcim devices name=my-router
```

`salt.modules.netbox.`**`get_circuit_provider`**`(`*name*, *asn=None*`)`
New in version 2019.2.0.

Get a circuit provider with a given name and optional ASN.
name The name of the circuit provider
asn The ASN of the circuit provider
CLI Example:

```
salt myminion netbox.get_circuit_provider Telia 1299
```

`salt.modules.netbox.`**`get_interfaces`**`(`*device_name=None*, ***kwargs*`)`
New in version 2019.2.0.

Returns interfaces for a specific device using arbitrary netbox filters
device_name The name of the device, e.g., `edge_router`
kwargs Optional arguments to be used for filtering
CLI Example:

```
salt myminion netbox.get_interfaces edge_router name="et-0/0/5"
```

`salt.modules.netbox.`**`get_ipaddresses`**`(`*device_name=None*, ***kwargs*`)`
New in version 2019.2.0.

Filters for an IP address using specified filters
device_name The name of the device to check for the IP address
kwargs Optional arguments that can be used to filter, e.g., `family=4`
CLI Example:

```
salt myminion netbox.get_ipaddresses device_name family=4
```

`salt.modules.netbox.`**`make_interface_child`**`(`*device_name, interface_name, parent_name*`)`

New in version 2019.2.0.

Set an interface as part of a LAG.

device_name The name of the device, e.g., `edge_router`.

interface_name The name of the interface to be attached to LAG, e.g., `xe-1/0/2`.

parent_name The name of the LAG interface, e.g., `ae13`.

CLI Example:

```
salt myminion netbox.make_interface_child xe-1/0/2 ae13
```

`salt.modules.netbox.`**`make_interface_lag`**`(`*device_name, interface_name*`)`

New in version 2019.2.0.

Update an interface to be a LAG.

device_name The name of the device, e.g., `edge_router`.

interface_name The name of the interface, e.g., `ae13`.

CLI Example:

```
salt myminion netbox.make_interface_lag edge_router ae13
```

`salt.modules.netbox.`**`openconfig_interfaces`**`(`*device_name=None*`)`

New in version 2019.2.0.

Return a dictionary structured as standardised in the openconfig-interfaces YANG model, containing physical and configuration data available in Netbox, e.g., IP addresses, MTU, enabled / disabled, etc.

device_name: **`None`** The name of the device to query the interface data for. If not provided, will use the Minion ID.

CLI Example:

```
salt '*' netbox.openconfig_interfaces
salt '*' netbox.openconfig_interfaces device_name=cr1.thn.lon
```

`salt.modules.netbox.`**`openconfig_lacp`**`(`*device_name=None*`)`

New in version 2019.2.0.

Return a dictionary structured as standardised in the openconfig-lacp YANG model, with configuration data for Link Aggregation Control Protocol (LACP) for aggregate interfaces.

Note: The `interval` and `lacp_mode` keys have the values set as `SLOW` and `ACTIVE` respectively, as this data is not currently available in Netbox, therefore defaulting to the values defined in the standard. See interval and lacp-mode for further details.

device_name: **`None`** The name of the device to query the LACP information for. If not provided, will use the Minion ID.

CLI Example:

```
salt '*' netbox.openconfig_lacp
salt '*' netbox.openconfig_lacp device_name=cr1.thn.lon
```

`salt.modules.netbox.`**`slugify`**`(`*value*`)`

’ Slugify given value. Credit to Djangoproject https://docs.djangoproject.com/en/2.0/_modules/django/utils/text/#slugify `salt.modules.netbox.`**`update_device`**(*name*, ***kwargs*)

New in version 2019.2.0.

Add attributes to an existing device, identified by name.

name The name of the device, e.g., `edge_router`

kwargs Arguments to change in device, e.g., `serial=JN2932930`

CLI Example:

```
salt myminion netbox.update_device edge_router serial=JN2932920
```

`salt.modules.netbox.`**`update_interface`**(*device_name*, *interface_name*, ***kwargs*)

New in version 2019.2.0.

Update an existing interface with new attributes.

device_name The name of the device, e.g., `edge_router`

interface_name The name of the interface, e.g., `ae13`

kwargs Arguments to change in interface, e.g., `mac_address=50:87:69:53:32:D0`

CLI Example:

```
salt myminion netbox.update_interface edge_router ae13 mac_
↪address=50:87:69:53:32:D0
```

21.9.276 salt.modules.netbsd_sysctl

Module for viewing and modifying sysctl parameters

`salt.modules.netbsd_sysctl.`**`assign`**(*name*, *value*)

Assign a single sysctl parameter for this minion

CLI Example:

```
salt '*' sysctl.assign net.inet.icmp.icmplim 50
```

`salt.modules.netbsd_sysctl.`**`get`**(*name*)

Return a single sysctl parameter for this minion

CLI Example:

```
salt '*' sysctl.get hw.physmem
```

`salt.modules.netbsd_sysctl.`**`persist`**(*name*, *value*, *config='/etc/sysctl.conf'*)

Assign and persist a simple sysctl parameter for this minion

CLI Example:

```
salt '*' sysctl.persist net.inet.icmp.icmplim 50
```

`salt.modules.netbsd_sysctl.`**`show`**(*config_file=False*)

Return a list of sysctl parameters for this minion

CLI Example:

```
salt '*' sysctl.show
```

21.9.277 salt.modules.netbsdservice

The service module for NetBSD

Important: If you feel that Salt should be using this module to manage services on a minion, and it is using a different module (or gives an error similar to *'service.start' is not available*), see *here*.

`salt.modules.netbsdservice.`**`available`**`(`*name*`)`

Returns `True` if the specified service is available, otherwise returns `False`.

CLI Example:

```
salt '*' service.available sshd
```

`salt.modules.netbsdservice.`**`disable`**`(`*name*, ***kwargs*`)`

Disable the named service to start at boot

CLI Example:

```
salt '*' service.disable <service name>
```

`salt.modules.netbsdservice.`**`disabled`**`(`*name*`)`

Return True if the named service is enabled, false otherwise

CLI Example:

```
salt '*' service.disabled <service name>
```

`salt.modules.netbsdservice.`**`enable`**`(`*name*, ***kwargs*`)`

Enable the named service to start at boot

CLI Example:

```
salt '*' service.enable <service name>
```

`salt.modules.netbsdservice.`**`enabled`**`(`*name*, ***kwargs*`)`

Return True if the named service is enabled, false otherwise

CLI Example:

```
salt '*' service.enabled <service name>
```

`salt.modules.netbsdservice.`**`force_reload`**`(`*name*`)`

Force-reload the named service

CLI Example:

```
salt '*' service.force_reload <service name>
```

`salt.modules.netbsdservice.`**`get_all`**`()`

Return all available boot services

CLI Example:

```
salt '*' service.get_all
```

`salt.modules.netbsdservice.`**`get_disabled`**`()`

Return a set of services that are installed but disabled

CLI Example:

```
salt '*' service.get_disabled
```

`salt.modules.netbsdservice.`**`get_enabled`**`()`

Return a list of service that are enabled on boot

CLI Example:

```
salt '*' service.get_enabled
```

`salt.modules.netbsdservice.`**`missing`**`(`*name*`)`

The inverse of service.available. Returns `True` if the specified service is not available, otherwise returns `False`.

CLI Example:

```
salt '*' service.missing sshd
```

`salt.modules.netbsdservice.`**`reload_`**`(`*name*`)`

Reload the named service

CLI Example:

```
salt '*' service.reload <service name>
```

`salt.modules.netbsdservice.`**`restart`**`(`*name*`)`

Restart the named service

CLI Example:

```
salt '*' service.restart <service name>
```

`salt.modules.netbsdservice.`**`start`**`(`*name*`)`

Start the specified service

CLI Example:

```
salt '*' service.start <service name>
```

`salt.modules.netbsdservice.`**`status`**`(`*name, sig=None*`)`

Return the status for a service. If the name contains globbing, a dict mapping service name to True/False values is returned.

Changed in version 2018.3.0: The service name can now be a glob (e.g. `salt*`)

Parameters
- **`name`** (*`str`*) – The name of the service to check
- **`sig`** (*`str`*) – Signature to use to find the service via ps

Returns True if running, False otherwise dict: Maps service name to True if running, False otherwise

Return type bool

CLI Example:

```
salt '*' service.status <service name> [service signature]
```

`salt.modules.netbsdservice.`**`stop`**`(`*name*`)`

Stop the specified service

CLI Example:

```
salt '*' service.stop <service name>
```

21.9.278 salt.modules.netmiko_mod

Netmiko Execution Module

New in version 2019.2.0.

Execution module to interface the connection with a remote network device. It is flexible enough to execute the commands both when running under a Netmiko Proxy Minion, as well as running under a Regular Minion by specifying the connection arguments, i.e., `device_type`, `ip`, `username`, `password` etc.

codeauthor Mircea Ulinic <ping@mirceaulinic.net> & Kirk Byers <ktbyers@twb-tech.com>

maturity new

depends netmiko

platform unix

Dependencies

The `netmiko` proxy modules requires Netmiko to be installed: `pip install netmiko`.

Usage

This module can equally be used via the *`netmiko`* Proxy module (check documentation), or directly from an arbitrary (Proxy) Minion that is running on a server (computer) having access to the network device, and has the `netmiko` library installed.

When running outside of the *`netmiko Proxy`* (i.e., from another Proxy Minion type, or regular Minion), the netmiko connection arguments can be either specified from the CLI when executing the command, or in a configuration block under the `netmiko` key in the configuration opts (i.e., (Proxy) Minion configuration file), or Pillar. The module supports these simultaneously. These fields are the exact same supported by the `netmiko` Proxy Module:

- `device_type` - Class selection based on device type. Supported options:
  - `a10`: A10 Networks
  - `accedian`: Accedian Networks
  - `alcatel_aos`: Alcatel AOS
  - `alcatel_sros`: Alcatel SROS
  - `apresia_aeos`: Apresia AEOS
  - `arista_eos`: Arista EOS
  - `aruba_os`: Aruba
  - `avaya_ers`: Avaya ERS
  - `avaya_vsp`: Avaya VSP
  - `brocade_fastiron`: Brocade Fastiron
  - `brocade_netiron`: Brocade Netiron

- `brocade_nos`: Brocade NOS
- `brocade_vdx`: Brocade NOS
- `brocade_vyos`: VyOS
- `checkpoint_gaia`: Check Point GAiA
- `calix_b6`: Calix B6
- `ciena_saos`: Ciena SAOS
- `cisco_asa`: Cisco SA
- `cisco_ios`: Cisco IOS
- `cisco_nxos`: Cisco NX-oS
- `cisco_s300`: Cisco S300
- `cisco_tp`: Cisco TpTcCe
- `cisco_wlc`: Cisco WLC
- `cisco_xe`: Cisco IOS
- `cisco_xr`: Cisco XR
- `coriant`: Coriant
- `dell_force10`: Dell Force10
- `dell_os10`: Dell OS10
- `dell_powerconnect`: Dell PowerConnect
- `eltex`: Eltex
- `enterasys`: Enterasys
- `extreme`: Extreme
- `extreme_wing`: Extreme Wing
- `f5_ltm`: F5 LTM
- `fortinet`: Fortinet
- `generic_termserver`: TerminalServer
- `hp_comware`: HP Comware
- `hp_procurve`: HP Procurve
- `huawei`: Huawei
- `huawei_vrpv8`: Huawei VRPV8
- `juniper`: Juniper Junos
- `juniper_junos`: Juniper Junos
- `linux`: Linux
- `mellanox`: Mellanox
- `mrv_optiswitch`: MrvOptiswitch
- `netapp_cdot`: NetAppcDot
- `netscaler`: Netscaler

  - `ovs_linux`: OvsLinux
  - `paloalto_panos`: PaloAlto Panos
  - `pluribus`: Pluribus
  - `quanta_mesh`: Quanta Mesh
  - `ruckus_fastiron`: Ruckus Fastiron
  - `ubiquiti_edge`: Ubiquiti Edge
  - `ubiquiti_edgeswitch`: Ubiquiti Edge
  - `vyatta_vyos`: VyOS
  - `vyos`: VyOS
  - `brocade_fastiron_telnet`: Brocade Fastiron over Telnet
  - `brocade_netiron_telnet`: Brocade Netiron over Telnet
  - `cisco_ios_telnet`: Cisco IOS over Telnet
  - `apresia_aeos_telnet`: Apresia AEOS over Telnet
  - `arista_eos_telnet`: Arista EOS over Telnet
  - `hp_procurve_telnet`: HP Procurve over Telnet
  - `hp_comware_telnet`: HP Comware over Telnet
  - `juniper_junos_telnet`: Juniper Junos over Telnet
  - `calix_b6_telnet`: Calix B6 over Telnet
  - `dell_powerconnect_telnet`: Dell PowerConnect over Telnet
  - `generic_termserver_telnet`: TerminalServer over Telnet
  - `extreme_telnet`: Extreme Networks over Telnet
  - `ruckus_fastiron_telnet`: Ruckus Fastiron over Telnet
  - `cisco_ios_serial`: Cisco IOS over serial port
- `ip` - IP address of target device (not required if `host` is provided)
- `host` - Hostname of target device (not required if `ip` is provided)
- `username` - Username to authenticate against target device, if required
- `password` - Password to authenticate against target device, if required
- `secret` - The enable password if target device requires one
- `port` - The destination port used to connect to the target device
- `global_delay_factor` - Multiplication factor affecting Netmiko delays (default: `1`)
- `use_keys` - Connect to target device using SSH keys (default: `False`)
- `key_file` - Filename path of the SSH key file to use
- `allow_agent` - Enable use of SSH key-agent
- `ssh_strict` - Automatically reject unknown SSH host keys (default: `False`, which means unknown SSH host keys will be accepted)
- `system_host_keys` - Load host keys from the user's "known_hosts" file (default: `False`)

- `alt_host_keys` - If `True`, host keys will be loaded from the file specified in `alt_key_file` (default: `False`)
- `alt_key_file` - SSH host key file to use (if `alt_host_keys=True`)
- `ssh_config_file` - File name of OpenSSH configuration file
- `timeout` - Connection timeout, in seconds (default: `90`)
- `session_timeout` - Set a timeout for parallel requests, in seconds (default: `60`)
- `keepalive` - Send SSH keepalive packets at a specific interval, in seconds. Currently defaults to `0`, for backwards compatibility (it will not attempt to keep the connection alive using the KEEPALIVE packets).
- `default_enter` - Character(s) to send to correspond to enter key (default: `\n`)
- `response_return` - Character(s) to use in normalized return data to represent enter key (default: `\n`)

Example (when not running in a `netmiko` Proxy Minion):

```
netmiko:
  username: test
  password: test
```

In case the `username` and `password` are the same on any device you are targeting, the block above (besides other parameters specific to your environment you might need) should suffice to be able to execute commands from outside a `netmiko` Proxy, e.g.:

```
salt '*' netmiko.send_command 'show version' host=router1.example.com device_
↪type=juniper
salt '*' netmiko.send_config https://bit.ly/2sgljCB host=sw2.example.com device_
↪type=cisco_ios
```

Note: Remember that the above applies only when not running in a `netmiko` Proxy Minion. If you want to use the `<salt.proxy.netmiko_px>`, please follow the documentation notes for a proper setup.

`salt.modules.netmiko_mod.`**`call`**`(`*method, *args, **kwargs*`)`

Invoke an arbitrary Netmiko method.

method The name of the Netmiko method to invoke.

args A list of arguments to send to the method invoked.

kwargs Key-value dictionary to send to the method invoked.

`salt.modules.netmiko_mod.`**`commit`**`(`***kwargs*`)`

Commit the configuration changes.

Warning: This function is supported only on the platforms that support the `commit` operation.

CLI Example:

```
salt '*' netmiko.commit
```

`salt.modules.netmiko_mod.`**`enter_config_mode`**`(`***kwargs*`)`

Enter into config mode.

config_command Configuration command to send to the device.

pattern Pattern to terminate reading of channel.

CLI Example:

```
salt '*' netmiko.enter_config_mode
salt '*' netmiko.enter_config_mode device_type='juniper_junos' ip='192.168.0.1'␣
↪username='example'
```

`salt.modules.netmiko_mod.`**`exit_config_mode`**`(`*`**kwargs`*`)`

Exit from configuration mode.

exit_config Command to exit configuration mode.

pattern Pattern to terminate reading of channel.

CLI Example:

```
salt '*' netmiko.exit_config_mode
salt '*' netmiko.exit_config_mode device_type='juniper' ip='192.168.0.1' username=
↪'example'
```

`salt.modules.netmiko_mod.`**`get_connection`**`(`*`**kwargs`*`)`

Return the Netmiko connection object.

> **Warning:** This function returns an unserializable object, hence it is not meant to be used on the CLI. This should mainly be used when invoked from other modules for the low level connection with the network device.

kwargs Key-value dictionary with the authentication details.

USAGE Example:

```
conn = __salt__['netmiko.get_connection'](host='router1.example.com',
                                          username='example',
                                          password='example')
show_if = conn.send_command('show interfaces')
conn.disconnect()
```

`salt.modules.netmiko_mod.`**`multi_call`**`(`*`*methods, **kwargs`*`)`

Invoke multiple Netmiko methods at once, and return their output, as list.

methods A list of dictionaries with the following keys:

- `name`: the name of the Netmiko method to be executed.
- `args`: list of arguments to be sent to the Netmiko method.
- `kwargs`: dictionary of arguments to be sent to the Netmiko method.

kwargs Key-value dictionary with the connection details (when not running under a Proxy Minion).

`salt.modules.netmiko_mod.`**`send_command`**`(`*`command_string, **kwargs`*`)`

Execute command_string on the SSH channel using a pattern-based mechanism. Generally used for show commands. By default this method will keep waiting to receive data until the network device prompt is detected. The current network device prompt will be determined automatically.

command_string The command to be executed on the remote device.

expect_string Regular expression pattern to use for determining end of output. If left blank will default to being based on router prompt.

delay_factor: `1` Multiplying factor used to adjust delays (default: `1`).

max_loops: `500` Controls wait time in conjunction with delay_factor. Will default to be based upon self.timeout.

auto_find_prompt: `True` Whether it should try to auto-detect the prompt (default: `True`).

strip_prompt: `True` Remove the trailing router prompt from the output (default: `True`).

strip_command: `True` Remove the echo of the command from the output (default: `True`).

normalize: `True` Ensure the proper enter is sent at end of command (default: `True`).

use_textfsm: `False` Process command output through TextFSM template (default: `False`).

CLI Example:

```
salt '*' netmiko.send_command 'show version'
salt '*' netmiko.send_command 'show_version' host='router1.example.com' username=
↪'example' device_type='cisco_ios'
```

salt.modules.netmiko_mod.**send_command_timing**(*command_string*, ***kwargs*)

Execute command_string on the SSH channel using a delay-based mechanism. Generally used for show commands.

command_string The command to be executed on the remote device.

delay_factor: **1** Multiplying factor used to adjust delays (default: 1).

max_loops: **500** Controls wait time in conjunction with delay_factor. Will default to be based upon self.timeout.

strip_prompt: **True** Remove the trailing router prompt from the output (default: `True`).

strip_command: **True** Remove the echo of the command from the output (default: `True`).

normalize: **True** Ensure the proper enter is sent at end of command (default: `True`).

use_textfsm: **False** Process command output through TextFSM template (default: `False`).

CLI Example:

```
salt '*' netmiko.send_command_timing 'show version'
salt '*' netmiko.send_command_timing 'show version' host='router1.example.com'
↪username='example' device_type='arista_eos'
```

salt.modules.netmiko_mod.**send_config**(*config_file=None*, *config_commands=None*, *template_engine='jinja'*, *commit=False*, *context=None*, *defaults=None*, *saltenv='base'*, ***kwargs*)

Send configuration commands down the SSH channel. Return the configuration lines sent to the device.

The function is flexible to send the configuration from a local or remote file, or simply the commands as list.

config_file The source file with the configuration commands to be sent to the device.

The file can also be a template that can be rendered using the template engine of choice.

This can be specified using the absolute path to the file, or using one of the following URL schemes:

- `salt://`, to fetch the file from the Salt fileserver.
- `http://` or `https://`
- `ftp://`
- `s3://`
- `swift://`

config_commands Multiple configuration commands to be sent to the device.

Note: This argument is ignored when `config_file` is specified.

template_engine: **jinja** The template engine to use when rendering the source file. Default: `jinja`. To simply fetch the file without attempting to render, set this argument to `None`.

commit: **False** Commit the configuration changes before exiting the config mode. This option is by default disabled, as many platforms don't have this capability natively.

context Variables to add to the template context.

defaults Default values of the context_dict.

exit_config_mode: **True** Determines whether or not to exit config mode after complete.

delay_factor: **1** Factor to adjust delays.

max_loops: **150** Controls wait time in conjunction with delay_factor (default: `150`).

strip_prompt: **False** Determines whether or not to strip the prompt (default: `False`).

strip_command: **False** Determines whether or not to strip the command (default: `False`).

config_mode_command The command to enter into config mode.

CLI Example:

```
salt '*' netmiko.send_config config_commands="['interface GigabitEthernet3', 'no␣
↪ip address']"
salt '*' netmiko.send_config config_commands="['snmp-server location {{ grains.
↪location }}']"
salt '*' netmiko.send_config config_file=salt://config.txt
salt '*' netmiko.send_config config_file=https://bit.ly/2sgljCB device_type=
↪'cisco_ios' ip='1.2.3.4' username='example'
```

21.9.279 salt.modules.netscaler

Module to provide Citrix Netscaler compatibility to Salt (compatible with netscaler 9.2+)

New in version 2015.2.0.

depends

- nsnitro Python module

Note: You can install nsnitro using:

```
pip install nsnitro
```

---

configuration This module accepts connection configuration details either as parameters, or as configuration settings in /etc/salt/minion on the relevant minions

```
netscaler.host: 1.2.3.4
netscaler.user: user
netscaler.pass: password
```

This data can also be passed into pillar. Options passed into opts will overwrite options passed into pillar.

CLI Examples Calls relying on configuration passed using /etc/salt/minion, grains, or pillars: .. code-block:: bash

salt-call netscaler.server_exists server_name

Calls passing configuration as opts .. code-block:: bash

salt-call netscaler.server_exists server_name netscaler_host=1.2.3.4 netscaler_user=username netscaler_pass=password salt-call netscaler.server_exists server_name netscaler_host=1.2.3.5 netscaler_user=username2 netscaler_pass=password2 salt-call netscaler.server_enable server_name2 netscaler_host=1.2.3.5 salt-call netscaler.server_up server_name3 netscaler_host=1.2.3.6 netscaler_useSSL=False

`salt.modules.netscaler.`**`server_add`**(*s_name*, *s_ip*, *s_state=None*, ***connection_args*)

Add a server Note: The default server state is ENABLED

CLI Example:

```
salt '*' netscaler.server_add 'serverName' 'serverIpAddress'
salt '*' netscaler.server_add 'serverName' 'serverIpAddress' 'serverState'
```

`salt.modules.netscaler.`**`server_delete`**(*s_name*, ***connection_args*)

Delete a server

CLI Example:

```
salt '*' netscaler.server_delete 'serverName'
```

`salt.modules.netscaler.`**`server_disable`**(*s_name*, ***connection_args*)

Disable a server globally

CLI Example:

```
salt '*' netscaler.server_disable 'serverName'
```

`salt.modules.netscaler.`**`server_enable`**(*s_name*, ***connection_args*)

Enables a server globally

CLI Example:

```
salt '*' netscaler.server_enable 'serverName'
```

`salt.modules.netscaler.`**`server_enabled`**(*s_name*, ***connection_args*)

Check if a server is enabled globally

CLI Example:

```
salt '*' netscaler.server_enabled 'serverName'
```

`salt.modules.netscaler.`**`server_exists`**(*s_name*, *ip=None*, *s_state=None*, ***connection_args*)

Checks if a server exists

CLI Example:

```
salt '*' netscaler.server_exists 'serverName'
```

`salt.modules.netscaler.`**`server_update`**(*s_name*, *s_ip*, ***connection_args*)

Update a server's attributes

CLI Example:

```
salt '*' netscaler.server_update 'serverName' 'serverIP'
```

`salt.modules.netscaler.`**`service_disable`**(*s_name*, *s_delay=None*, ***connection_args*)

Disable a service

CLI Example:

```
salt '*' netscaler.service_disable 'serviceName'
salt '*' netscaler.service_disable 'serviceName' 'delayInSeconds'
```

`salt.modules.netscaler.`**`service_enable`**(*s_name*, ***connection_args*)

Enable a service

CLI Example:

```
salt '*' netscaler.service_enable 'serviceName'
```

`salt.modules.netscaler.`**`service_exists`**(*s_name*, ***connection_args*)

Checks if a service exists

CLI Example:

```
salt '*' netscaler.service_exists 'serviceName'
```

`salt.modules.netscaler.`**`service_up`**(*s_name*, ***connection_args*)

Checks if a service is UP

CLI Example:

```
salt '*' netscaler.service_up 'serviceName'
```

`salt.modules.netscaler.`**`servicegroup_add`**(*sg_name*, *sg_type='HTTP'*, ***connection_args*)

Add a new service group If no service type is specified, HTTP will be used. Most common service types: HTTP, SSL, and SSL_BRIDGE

CLI Example:

```
salt '*' netscaler.servicegroup_add 'serviceGroupName'
salt '*' netscaler.servicegroup_add 'serviceGroupName' 'serviceGroupType'
```

`salt.modules.netscaler.`**`servicegroup_delete`**(*sg_name*, ***connection_args*)

Delete a new service group

CLI Example:

```
salt '*' netscaler.servicegroup_delete 'serviceGroupName'
```

`salt.modules.netscaler.`**`servicegroup_exists`**(*sg_name*, *sg_type=None*, ***connection_args*)

Checks if a service group exists

CLI Example:

```
salt '*' netscaler.servicegroup_exists 'serviceGroupName'
```

`salt.modules.netscaler.`**`servicegroup_server_add`**(*sg_name*, *s_name*, *s_port*, ***connection_args*)

Add a server:port member to a servicegroup

CLI Example:

```
salt '*' netscaler.servicegroup_server_add 'serviceGroupName' 'serverName'
↪'serverPort'
```

`salt.modules.netscaler.`**`servicegroup_server_delete`**(*sg_name*, *s_name*, *s_port*, ***connection_args*)

Remove a server:port member from a servicegroup

CLI Example:

```
salt '*' netscaler.servicegroup_server_delete 'serviceGroupName' 'serverName'
↪'serverPort'
```

`salt.modules.netscaler.`**`servicegroup_server_disable`**(*sg_name*, *s_name*, *s_port*, ***connection_args*)

Disable a server:port member of a servicegroup

CLI Example:

```
salt '*' netscaler.servicegroup_server_disable 'serviceGroupName' 'serverName'
↪'serverPort'
```

`salt.modules.netscaler.`**`servicegroup_server_enable`**`(`*sg_name, s_name, s_port, **connection_args*`)`

Enable a server:port member of a servicegroup

CLI Example:

```
salt '*' netscaler.servicegroup_server_enable 'serviceGroupName' 'serverName'
↪'serverPort'
```

`salt.modules.netscaler.`**`servicegroup_server_exists`**`(`*sg_name, s_name, s_port=None, **connection_args*`)`

Check if a server:port combination is a member of a servicegroup

CLI Example:

```
salt '*' netscaler.servicegroup_server_exists 'serviceGroupName' 'serverName'
↪'serverPort'
```

`salt.modules.netscaler.`**`servicegroup_server_up`**`(`*sg_name, s_name, s_port, **connection_args*`)`

Check if a server:port combination is in state UP in a servicegroup

CLI Example:

```
salt '*' netscaler.servicegroup_server_up 'serviceGroupName' 'serverName'
↪'serverPort'
```

`salt.modules.netscaler.`**`vserver_add`**`(`*v_name, v_ip, v_port, v_type, **connection_args*`)`

Add a new lb vserver

CLI Example:

```
salt '*' netscaler.vserver_add 'vserverName' 'vserverIP' 'vserverPort'
↪'vserverType'
salt '*' netscaler.vserver_add 'alex.patate.chaude.443' '1.2.3.4' '443' 'SSL'
```

`salt.modules.netscaler.`**`vserver_delete`**`(`*v_name, **connection_args*`)`

Delete a lb vserver

CLI Example:

```
salt '*' netscaler.vserver_delete 'vserverName'
```

`salt.modules.netscaler.`**`vserver_exists`**`(`*v_name, v_ip=None, v_port=None, v_type=None, **connection_args*`)`

Checks if a vserver exists

CLI Example:

```
salt '*' netscaler.vserver_exists 'vserverName'
```

`salt.modules.netscaler.`**`vserver_servicegroup_add`**`(`*v_name, sg_name, **connection_args*`)`

Bind a servicegroup to a vserver

CLI Example:

```
salt '*' netscaler.vserver_servicegroup_add 'vserverName' 'serviceGroupName'
```

`salt.modules.netscaler.`**`vserver_servicegroup_delete`**`(`*v_name, sg_name, **connection_args*`)`

Unbind a servicegroup from a vserver

CLI Example:

```
salt '*' netscaler.vserver_servicegroup_delete 'vserverName' 'serviceGroupName'
```

`salt.modules.netscaler.`**`vserver_servicegroup_exists`**`(`*v_name, sg_name, **connection_args*`)`

Checks if a servicegroup is tied to a vserver

CLI Example:

```
salt '*' netscaler.vserver_servicegroup_exists 'vserverName' 'serviceGroupName'
```

`salt.modules.netscaler.`**`vserver_sslcert_add`**`(`*v_name, sc_name, **connection_args*`)`

Binds a SSL certificate to a vserver

CLI Example:

```
salt '*' netscaler.vserver_sslcert_add 'vserverName' 'sslCertificateName'
```

`salt.modules.netscaler.`**`vserver_sslcert_delete`**`(`*v_name, sc_name, **connection_args*`)`

Unbinds a SSL certificate from a vserver

CLI Example:

```
salt '*' netscaler.vserver_sslcert_delete 'vserverName' 'sslCertificateName'
```

`salt.modules.netscaler.`**`vserver_sslcert_exists`**`(`*v_name, sc_name, **connection_args*`)`

Checks if a SSL certificate is tied to a vserver

CLI Example:

```
salt '*' netscaler.vserver_sslcert_exists 'vserverName' 'sslCertificateName'
```

21.9.280 salt.modules.network

Module for gathering and managing network information

`salt.modules.network.`**`active_tcp`**`()`

Return a dict containing information on all of the running TCP connections (currently linux and solaris only)

Changed in version 2015.8.4: Added support for SunOS

CLI Example:

```
salt '*' network.active_tcp
```

`salt.modules.network.`**`arp`**`()`

Return the arp table from the minion

Changed in version 2015.8.0: Added support for SunOS

CLI Example:

```
salt '*' network.arp
```

`salt.modules.network.`**`calc_net`**`(`*ip_addr, netmask=None*`)`

Returns the CIDR of a subnet based on an IP address (CIDR notation supported) and optional netmask.

CLI Example:

```
salt '*' network.calc_net 172.17.0.5 255.255.255.240
salt '*' network.calc_net 2a02:f6e:a000:80:84d8:8332:7866:4e07/64
```

New in version 2015.8.0.

`salt.modules.network.`**`connect`**(*host*, *port=None*, ***kwargs*)

Test connectivity to a host using a particular port from the minion.

New in version 2014.7.0.

CLI Example:

```
salt '*' network.connect archlinux.org 80

salt '*' network.connect archlinux.org 80 timeout=3

salt '*' network.connect archlinux.org 80 timeout=3 family=ipv4

salt '*' network.connect google-public-dns-a.google.com port=53 proto=udp␣
↪timeout=3
```

`salt.modules.network.`**`convert_cidr`**(*cidr*)

returns the network and subnet mask of a cidr addr

New in version 2016.3.0.

CLI Example:

```
salt '*' network.convert_cidr 172.31.0.0/16
```

`salt.modules.network.`**`default_route`**(*family=None*)

Return default route(s) from routing table

Changed in version 2015.8.0: Added support for SunOS (Solaris 10, Illumos, SmartOS)

Changed in version 2016.11.4: Added support for AIX

CLI Example:

```
salt '*' network.default_route
```

`salt.modules.network.`**`dig`**(*host*)

Performs a DNS lookup with dig

CLI Example:

```
salt '*' network.dig archlinux.org
```

`salt.modules.network.`**`get_bufsize`**(*iface*)

Return network buffer sizes as a dict (currently linux only)

CLI Example:

```
salt '*' network.get_bufsize eth0
```

`salt.modules.network.`**`get_fqdn`**()

Get fully qualified domain name

CLI Example:

```
salt '*' network.get_fqdn
```

salt.modules.network.**get_hostname**()
Get hostname

CLI Example:

```
salt '*' network.get_hostname
```

salt.modules.network.**get_route**(*ip*)
Return routing information for given destination ip

New in version 2015.5.3.

Changed in version 2015.8.0: Added support for SunOS (Solaris 10, Illumos, SmartOS) Added support for OpenBSD

Changed in version 2016.11.4: Added support for AIX

CLI Example:

```
salt '*' network.get_route 10.10.10.10
```

salt.modules.network.**hw_addr**(*iface*)
Return the hardware address (a.k.a. MAC address) for a given interface

CLI Example:

```
salt '*' network.hw_addr eth0
```

salt.modules.network.**hwaddr**(*iface*)
This function is an alias of `hw_addr`.

> Return the hardware address (a.k.a. MAC address) for a given interface
>
> CLI Example:
>
> ```
> salt '*' network.hw_addr eth0
> ```

salt.modules.network.**ifacestartswith**(*cidr*)
Retrieve the interface name from a specific CIDR

New in version 2016.11.0.

CLI Example:

```
salt '*' network.ifacestartswith 10.0
```

salt.modules.network.**in_subnet**(*cidr*)
Returns True if host is within specified subnet, otherwise False.

CLI Example:

```
salt '*' network.in_subnet 10.0.0.0/16
```

salt.modules.network.**interface**(*iface*)
Return the inet address for a given interface

New in version 2014.7.0.

CLI Example:

```
salt '*' network.interface eth0
```

`salt.modules.network.`**`interface_ip`**`(`*iface*`)`

Return the inet address for a given interface

New in version 2014.7.0.

CLI Example:

```
salt '*' network.interface_ip eth0
```

`salt.modules.network.`**`interfaces`**`()`

Return a dictionary of information about all the interfaces on the minion

CLI Example:

```
salt '*' network.interfaces
```

`salt.modules.network.`**`ip_addrs`**`(`*interface=None*, *include_loopback=False*, *cidr=None*, *type=None*`)`

Returns a list of IPv4 addresses assigned to the host. 127.0.0.1 is ignored, unless 'include_loopback=True' is indicated. If 'interface' is provided, then only IP addresses from that interface will be returned. Providing a CIDR via 'cidr="10.0.0.0/8"' will return only the addresses which are within that subnet. If 'type' is 'public', then only public addresses will be returned. Ditto for 'type'='private'.

CLI Example:

```
salt '*' network.ip_addrs
```

`salt.modules.network.`**`ip_addrs6`**`(`*interface=None*, *include_loopback=False*, *cidr=None*`)`

Returns a list of IPv6 addresses assigned to the host. ::1 is ignored, unless 'include_loopback=True' is indicated. If 'interface' is provided, then only IP addresses from that interface will be returned. Providing a CIDR via 'cidr="2000::/3"' will return only the addresses which are within that subnet.

CLI Example:

```
salt '*' network.ip_addrs6
```

`salt.modules.network.`**`ip_in_subnet`**`(`*ip_addr*, *cidr*`)`

Returns True if given IP is within specified subnet, otherwise False.

CLI Example:

```
salt '*' network.ip_in_subnet 172.17.0.4 172.16.0.0/12
```

`salt.modules.network.`**`ipaddrs`**`(`*interface=None*, *include_loopback=False*, *cidr=None*, *type=None*`)`

This function is an alias of `ip_addrs`.

> Returns a list of IPv4 addresses assigned to the host. 127.0.0.1 is ignored, unless 'include_loopback=True' is indicated. If 'interface' is provided, then only IP addresses from that interface will be returned. Providing a CIDR via 'cidr="10.0.0.0/8"' will return only the addresses which are within that subnet. If 'type' is 'public', then only public addresses will be returned. Ditto for 'type'='private'.
>
> CLI Example:
>
> ```
> salt '*' network.ip_addrs
> ```

`salt.modules.network.`**`ipaddrs6`**`(`*interface=None*, *include_loopback=False*, *cidr=None*`)`

This function is an alias of `ip_addrs6`.

Returns a list of IPv6 addresses assigned to the host. ::1 is ignored, unless 'include_loopback=True' is indicated. If 'interface' is provided, then only IP addresses from that interface will be returned. Providing a CIDR via 'cidr="2000::/3"' will return only the addresses which are within that subnet.

CLI Example:

```
salt '*' network.ip_addrs6
```

`salt.modules.network.`**`iphexval`**`(`*ip*`)`

Retrieve the hexadecimal representation of an IP address

New in version 2016.11.0.

CLI Example:

```
salt '*' network.iphexval 10.0.0.1
```

`salt.modules.network.`**`is_loopback`**`(`*ip_addr*`)`

Check if the given IP address is a loopback address

New in version 2014.7.0.

Changed in version 2015.8.0: IPv6 support

CLI Example:

```
salt '*' network.is_loopback 127.0.0.1
```

`salt.modules.network.`**`is_private`**`(`*ip_addr*`)`

Check if the given IP address is a private address

New in version 2014.7.0.

Changed in version 2015.8.0: IPv6 support

CLI Example:

```
salt '*' network.is_private 10.0.0.3
```

`salt.modules.network.`**`mod_bufsize`**`(`*iface, *args, **kwargs*`)`

Modify network interface buffers (currently linux only)

CLI Example:

```
salt '*' network.mod_bufsize tx=<val> rx=<val> rx-mini=<val> rx-jumbo=<val>
```

`salt.modules.network.`**`mod_hostname`**`(`*hostname*`)`

Modify hostname

Changed in version 2015.8.0: Added support for SunOS (Solaris 10, Illumos, SmartOS)

CLI Example:

```
salt '*' network.mod_hostname master.saltstack.com
```

`salt.modules.network.`**`netstat`**`()`

Return information on open ports and states

Note: On BSD minions, the output contains PID info (where available) for each netstat entry, fetched from sockstat/fstat output.

Changed in version 2014.1.4: Added support for OpenBSD, FreeBSD, and NetBSD

Changed in version 2015.8.0: Added support for SunOS

Changed in version 2016.11.4: Added support for AIX

CLI Example:

```
salt '*' network.netstat
```

`salt.modules.network.`**`ping`**(*host*, *timeout=False*, *return_boolean=False*)

Performs an ICMP ping to a host

Changed in version 2015.8.0: Added support for SunOS

CLI Example:

```
salt '*' network.ping archlinux.org
```

New in version 2015.5.0.

Return a True or False instead of ping output.

```
salt '*' network.ping archlinux.org return_boolean=True
```

Set the time to wait for a response in seconds.

```
salt '*' network.ping archlinux.org timeout=3
```

`salt.modules.network.`**`reverse_ip`**(*ip_addr*)

Returns the reversed IP address

Changed in version 2015.8.0: IPv6 support

CLI Example:

```
salt '*' network.reverse_ip 172.17.0.4
```

`salt.modules.network.`**`routes`**(*family=None*)

Return currently configured routes from routing table

Changed in version 2015.8.0: Added support for SunOS (Solaris 10, Illumos, SmartOS)

Changed in version 2016.11.4: Added support for AIX

CLI Example:

```
salt '*' network.routes
```

`salt.modules.network.`**`subnets`**(*interfaces=None*)

Returns a list of IPv4 subnets to which the host belongs

CLI Example:

```
salt '*' network.subnets
salt '*' network.subnets interfaces=eth1
```

`salt.modules.network.`**`subnets6`**()

Returns a list of IPv6 subnets to which the host belongs

CLI Example:

```
salt '*' network.subnets
```

`salt.modules.network.`**`traceroute`**`(`*host*`)`

Performs a traceroute to a 3rd party host

Changed in version 2015.8.0: Added support for SunOS

Changed in version 2016.11.4: Added support for AIX

CLI Example:

```
salt '*' network.traceroute archlinux.org
```

`salt.modules.network.`**`wol`**`(`*mac, bcast='255.255.255.255', destport=9*`)`

Send Wake On Lan packet to a host

CLI Example:

```
salt '*' network.wol 08-00-27-13-69-77
salt '*' network.wol 080027136977 255.255.255.255 7
salt '*' network.wol 08:00:27:13:69:77 255.255.255.255 7
```

21.9.281 salt.modules.neutron

Module for handling OpenStack Neutron calls

depends

- neutronclient Python module

configuration This module is not usable until the user, password, tenant, and auth URL are specified either in a pillar or in the minion's config file. For example:

```
keystone.user: 'admin'
keystone.password: 'password'
keystone.tenant: 'admin'
keystone.auth_url: 'http://127.0.0.1:5000/v2.0/'
keystone.region_name: 'RegionOne'
keystone.service_type: 'network'
```

If configuration for multiple OpenStack accounts is required, they can be set up as different configuration profiles: For example:

```
openstack1:
  keystone.user: 'admin'
  keystone.password: 'password'
  keystone.tenant: 'admin'
  keystone.auth_url: 'http://127.0.0.1:5000/v2.0/'
  keystone.region_name: 'RegionOne'
  keystone.service_type: 'network'

openstack2:
  keystone.user: 'admin'
  keystone.password: 'password'
  keystone.tenant: 'admin'
  keystone.auth_url: 'http://127.0.0.2:5000/v2.0/'
  keystone.region_name: 'RegionOne'
  keystone.service_type: 'network'
```

With this configuration in place, any of the neutron functions can make use of a configuration profile by declaring it explicitly. For example:

```
salt '*' neutron.network_list profile=openstack1
```

To use keystoneauth1 instead of keystoneclient, include the *use_keystoneauth* option in the pillar or minion config.

Note: this is required to use keystone v3 as for authentication.

```
keystone.user: admin
keystone.password: verybadpass
keystone.tenant: admin
keystone.auth_url: 'http://127.0.0.1:5000/v3/'
keystone.region_name: 'RegionOne'
keystone.service_type: 'network'
keystone.use_keystoneauth: true
keystone.verify: '/path/to/custom/certs/ca-bundle.crt'
```

Note: by default the neutron module will attempt to verify its connection utilizing the system certificates. If you need to verify against another bundle of CA certificates or want to skip verification altogether you will need to specify the *verify* option. You can specify True or False to verify (or not) against system certificates, a path to a bundle or CA certs to check against, or None to allow keystoneauth to search for the certificates on its own.(defaults to True)

`salt.modules.neutron.`**`add_gateway_router`**(*router*, *ext_network*, *profile=None*)
Adds an external network gateway to the specified router

CLI Example:

```
salt '*' neutron.add_gateway_router router-name ext-network-name
```

Parameters
- **router** – ID or name of the router
- **ext_network** – ID or name of the external network the gateway
- **profile** – Profile to build on (Optional)

Returns Added Gateway router information

`salt.modules.neutron.`**`add_interface_router`**(*router*, *subnet*, *profile=None*)
Adds an internal network interface to the specified router

CLI Example:

```
salt '*' neutron.add_interface_router router-name subnet-name
```

Parameters
- **router** – ID or name of the router
- **subnet** – ID or name of the subnet
- **profile** – Profile to build on (Optional)

Returns Added interface information

`salt.modules.neutron.`**`create_firewall_rule`**(*protocol*, *action*, *profile=None*, ***kwargs*)
Creates a new firewall rule

CLI Example:

```
salt '*' neutron.create_firewall_rule protocol action
        tenant_id=TENANT_ID name=NAME description=DESCRIPTION ip_version=IP_
↪VERSION
        source_ip_address=SOURCE_IP_ADDRESS destination_ip_address=DESTINATION_IP_
↪ADDRESS source_port=SOURCE_PORT
        destination_port=DESTINATION_PORT shared=SHARED enabled=ENABLED
```

Parameters

- **protocol** – Protocol for the firewall rule, choose "tcp","udp","icmp" or "None".
- **action** – Action for the firewall rule, choose "allow" or "deny".
- **tenant_id** – The owner tenant ID. (Optional)
- **name** – Name for the firewall rule. (Optional)
- **description** – Description for the firewall rule. (Optional)
- **ip_version** – IP protocol version, default: 4. (Optional)
- **source_ip_address** – Source IP address or subnet. (Optional)
- **destination_ip_address** – Destination IP address or subnet. (Optional)
- **source_port** – Source port (integer in [1, 65535] or range in a:b). (Optional)
- **destination_port** – Destination port (integer in [1, 65535] or range in a:b). (Optional)
- **shared** – Set shared to True, default: False. (Optional)
- **enabled** – To enable this rule, default: True. (Optional)

`salt.modules.neutron.`**`create_floatingip`**(*floating_network, port=None, profile=None*)

Creates a new floatingIP

CLI Example:

```
salt '*' neutron.create_floatingip network-name port-name
```

Parameters

- **floating_network** – Network name or ID to allocate floatingIP from
- **port** – Of the port to be associated with the floatingIP (Optional)
- **profile** – Profile to build on (Optional)

Returns Created floatingIP information

`salt.modules.neutron.`**`create_ikepolicy`**(*name, profile=None, **kwargs*)

Creates a new IKEPolicy

CLI Example:

```
salt '*' neutron.create_ikepolicy ikepolicy-name
        phase1_negotiation_mode=main auth_algorithm=sha1
        encryption_algorithm=aes-128 pfs=group5
```

Parameters

- **name** – Name of the IKE policy
- **phase1_negotiation_mode** – IKE Phase1 negotiation mode in lowercase, default: main (Optional)
- **auth_algorithm** – Authentication algorithm in lowercase, default: sha1 (Optional)
- **encryption_algorithm** – Encryption algorithm in lowercase. default:aes-128 (Optional)
- **pfs** – Prefect Forward Security in lowercase, default: group5 (Optional)
- **units** – IKE lifetime attribute. default: seconds (Optional)
- **value** – IKE lifetime attribute. default: 3600 (Optional)
- **ike_version** – IKE version in lowercase, default: v1 (Optional)
- **profile** – Profile to build on (Optional)

- **kwargs** –

Returns Created IKE policy information

`salt.modules.neutron.`**`create_ipsec_site_connection`**(*name, ipsecpolicy, ikepolicy, vpnservice, peer_cidrs, peer_address, peer_id, psk, admin_state_up=True, profile=None, **kwargs*)

Creates a new IPsecSiteConnection

CLI Example:

```
salt '*' neutron.show_ipsec_site_connection connection-name
        ipsec-policy-name ikepolicy-name vpnservice-name
        192.168.XXX.XXX/24 192.168.XXX.XXX 192.168.XXX.XXX secret
```

Parameters

- **name** – Set friendly name for the connection
- **ipsecpolicy** – IPSec policy ID or name associated with this connection
- **ikepolicy** – IKE policy ID or name associated with this connection
- **vpnservice** – VPN service instance ID or name associated with this connection
- **peer_cidrs** – Remote subnet(s) in CIDR format
- **peer_address** – Peer gateway public IPv4/IPv6 address or FQDN
- **peer_id** – Peer router identity for authentication Can be IPv4/IPv6 address, e-mail address, key id, or FQDN
- **psk** – Pre-shared key string
- **initiator** – Initiator state in lowercase, default:bi-directional
- **admin_state_up** – Set admin state up to true or false, default: True (Optional)
- **mtu** – size for the connection, default:1500 (Optional)
- **dpd_action** – Dead Peer Detection attribute: hold/clear/disabled/ restart/restart-by-peer (Optional)
- **dpd_interval** – Dead Peer Detection attribute (Optional)
- **dpd_timeout** – Dead Peer Detection attribute (Optional)
- **profile** – Profile to build on (Optional)

Returns Created IPSec site connection information

`salt.modules.neutron.`**`create_ipsecpolicy`**(*name, profile=None, **kwargs*)

Creates a new IPsecPolicy

CLI Example:

```
salt '*' neutron.create_ipsecpolicy ipsecpolicy-name
        transform_protocol=esp auth_algorithm=sha1
        encapsulation_mode=tunnel encryption_algorithm=aes-128
```

Parameters

- **name** – Name of the IPSec policy
- **transform_protocol** – Transform protocol in lowercase, default: esp (Optional)
- **auth_algorithm** – Authentication algorithm in lowercase, default: sha1 (Optional)
- **encapsulation_mode** – Encapsulation mode in lowercase, default: tunnel (Optional)
- **encryption_algorithm** – Encryption algorithm in lowercase, default:aes-128 (Optional)
- **pfs** – Prefect Forward Security in lowercase, default: group5 (Optional)
- **units** – IPSec lifetime attribute. default: seconds (Optional)
- **value** – IPSec lifetime attribute. default: 3600 (Optional)
- **profile** – Profile to build on (Optional)

Returns Created IPSec policy information

`salt.modules.neutron.`**`create_network`**(*name, router_ext=None, admin_state_up=True, network_type=None, physical_network=None, segmentation_id=None, shared=None, profile=None*)

Creates a new network

CLI Example:

```
salt '*' neutron.create_network network-name
salt '*' neutron.create_network network-name profile=openstack1
```

Parameters

- **`name`** – Name of network to create
- **`admin_state_up`** – should the state of the network be up? default: True (Optional)
- **`router_ext`** – True then if create the external network (Optional)
- **`network_type`** – the Type of network that the provider is such as GRE, VXLAN, VLAN, FLAT, or LOCAL (Optional)
- **`physical_network`** – the name of the physical network as neutron knows it (Optional)
- **`segmentation_id`** – the vlan id or GRE id (Optional)
- **`shared`** – is the network shared or not (Optional)
- **`profile`** – Profile to build on (Optional)

Returns Created network information

`salt.modules.neutron.`**`create_port`**(*name, network, device_id=None, admin_state_up=True, profile=None*)

Creates a new port

CLI Example:

```
salt '*' neutron.create_port network-name port-name
```

Parameters

- **`name`** – Name of port to create
- **`network`** – Network name or ID
- **`device_id`** – ID of device (Optional)
- **`admin_state_up`** – Set admin state up to true or false, default: true (Optional)
- **`profile`** – Profile to build on (Optional)

Returns Created port information

`salt.modules.neutron.`**`create_router`**(*name, ext_network=None, admin_state_up=True, profile=None*)

Creates a new router

CLI Example:

```
salt '*' neutron.create_router new-router-name
```

Parameters

- **`name`** – Name of router to create (must be first)
- **`ext_network`** – ID or name of the external for the gateway (Optional)
- **`admin_state_up`** – Set admin state up to true or false, default:true (Optional)
- **`profile`** – Profile to build on (Optional)

Returns Created router information

`salt.modules.neutron.`**`create_security_group`**(*name=None, description=None, profile=None*)

Creates a new security group

CLI Example:

```
salt '*' neutron.create_security_group security-group-name ␣
↪description='Security group for servers'
```

Parameters

- **name** – Name of security group (Optional)
- **description** – Description of security group (Optional)
- **profile** – Profile to build on (Optional)

Returns Created security group information

salt.modules.neutron.**create_security_group_rule**(*security_group, remote_group_id=None, direction='ingress', protocol=None, port_range_min=None, port_range_max=None, ethertype='IPv4', profile=None*)

Creates a new security group rule

CLI Example:

```
salt '*' neutron.show_security_group_rule security-group-rule-id
```

Parameters

- **security_group** – Security group name or ID to add rule
- **remote_group_id** – Remote security group name or ID to apply rule (Optional)
- **direction** – Direction of traffic: ingress/egress, default: ingress (Optional)
- **protocol** – Protocol of packet: null/icmp/tcp/udp, default: null (Optional)
- **port_range_min** – Starting port range (Optional)
- **port_range_max** – Ending port range (Optional)
- **ethertype** – IPv4/IPv6, default: IPv4 (Optional)
- **profile** – Profile to build on (Optional)

Returns Created security group rule information

salt.modules.neutron.**create_subnet**(*network, cidr, name=None, ip_version=4, profile=None*)

Creates a new subnet

CLI Example:

```
salt '*' neutron.create_subnet network-name 192.168.1.0/24
```

Parameters

- **network** – Network ID or name this subnet belongs to
- **cidr** – CIDR of subnet to create (Ex. '192.168.1.0/24')
- **name** – Name of the subnet to create (Optional)
- **ip_version** – Version to use, default is 4(IPv4) (Optional)
- **profile** – Profile to build on (Optional)

Returns Created subnet information

salt.modules.neutron.**create_vpnservice**(*subnet, router, name, admin_state_up=True, profile=None*)

Creates a new VPN service

CLI Example:

```
salt '*' neutron.create_vpnservice router-name name
```

Parameters

- **subnet** – Subnet unique identifier for the VPN service deployment
- **router** – Router unique identifier for the VPN service
- **name** – Set a name for the VPN service
- **admin_state_up** – Set admin state up to true or false, default:True (Optional)
- **profile** – Profile to build on (Optional)

Returns Created VPN service information

salt.modules.neutron.**delete_firewall_rule**(*firewall_rule*, *profile=None*)

Deletes the specified firewall_rule

CLI Example:

```
salt '*' neutron.delete_firewall_rule firewall-rule
```

Parameters
- **firewall_rule** – ID or name of firewall rule to delete
- **profile** – Profile to build on (Optional)

Returns True(Succeed) or False

salt.modules.neutron.**delete_floatingip**(*floatingip_id*, *profile=None*)

Deletes the specified floating IP

CLI Example:

```
salt '*' neutron.delete_floatingip floatingip-id
```

Parameters
- **floatingip_id** – ID of floatingIP to delete
- **profile** – Profile to build on (Optional)

Returns True(Succeed) or False

salt.modules.neutron.**delete_ikepolicy**(*ikepolicy*, *profile=None*)

Deletes the specified IKEPolicy

CLI Example:

```
salt '*' neutron.delete_ikepolicy ikepolicy-name
```

Parameters
- **ikepolicy** – ID or name of IKE policy to delete
- **profile** – Profile to build on (Optional)

Returns True(Succeed) or False

salt.modules.neutron.**delete_ipsec_site_connection**(*ipsec_site_connection*, *profile=None*)

Deletes the specified IPsecSiteConnection

CLI Example:

```
salt '*' neutron.delete_ipsec_site_connection connection-name
```

Parameters
- **ipsec_site_connection** – ID or name of ipsec site connection to delete
- **profile** – Profile to build on (Optional)

Returns True(Succeed) or False

salt.modules.neutron.**delete_ipsecpolicy**(*ipsecpolicy*, *profile=None*)

Deletes the specified IPsecPolicy

CLI Example:

```
salt '*' neutron.delete_ipsecpolicy ipsecpolicy-name
```

Parameters

- **`ipsecpolicy`** – ID or name of IPSec policy to delete
- **`profile`** – Profile to build on (Optional)

Returns True(Succeed) or False

`salt.modules.neutron.`**`delete_network`**(*network*, *profile=None*)

Deletes the specified network

CLI Example:

```
salt '*' neutron.delete_network network-name
salt '*' neutron.delete_network network-name profile=openstack1
```

Parameters

- **`network`** – ID or name of network to delete
- **`profile`** – Profile to build on (Optional)

Returns True(Succeed) or False

`salt.modules.neutron.`**`delete_port`**(*port*, *profile=None*)

Deletes the specified port

CLI Example:

```
salt '*' neutron.delete_network port-name
salt '*' neutron.delete_network port-name profile=openstack1
```

Parameters

- **`port`** – port name or ID
- **`profile`** – Profile to build on (Optional)

Returns True(Succeed) or False

`salt.modules.neutron.`**`delete_quota`**(*tenant_id*, *profile=None*)

Delete the specified tenant's quota value

CLI Example:

```
salt '*' neutron.update_quota tenant-id
salt '*' neutron.update_quota tenant-id profile=openstack1
```

Parameters

- **`tenant_id`** – ID of tenant to quota delete
- **`profile`** – Profile to build on (Optional)

Returns True(Delete succeed) or False(Delete failed)

`salt.modules.neutron.`**`delete_router`**(*router*, *profile=None*)

Delete the specified router

CLI Example:

```
salt '*' neutron.delete_router router-name
```

Parameters

- **`router`** – ID or name of router to delete
- **`profile`** – Profile to build on (Optional)

Returns True(Succeed) or False

`salt.modules.neutron.`**`delete_security_group`**(*security_group, profile=None*)

Deletes the specified security group

CLI Example:

```
salt '*' neutron.delete_security_group security-group-name
```

Parameters
- **`security_group`** – ID or name of security group to delete
- **`profile`** – Profile to build on (Optional)

Returns True(Succeed) or False

`salt.modules.neutron.`**`delete_security_group_rule`**(*security_group_rule_id, profile=None*)

Deletes the specified security group rule

CLI Example:

```
salt '*' neutron.delete_security_group_rule security-group-rule-id
```

Parameters
- **`security_group_rule_id`** – ID of security group rule to delete
- **`profile`** – Profile to build on (Optional)

Returns True(Succeed) or False

`salt.modules.neutron.`**`delete_subnet`**(*subnet, profile=None*)

Deletes the specified subnet

CLI Example:

```
salt '*' neutron.delete_subnet subnet-name
salt '*' neutron.delete_subnet subnet-name profile=openstack1
```

Parameters
- **`subnet`** – ID or name of subnet to delete
- **`profile`** – Profile to build on (Optional)

Returns True(Succeed) or False

`salt.modules.neutron.`**`delete_vpnservice`**(*vpnservice, profile=None*)

Deletes the specified VPN service

CLI Example:

```
salt '*' neutron.delete_vpnservice vpnservice-name
```

Parameters
- **`vpnservice`** – ID or name of vpn service to delete
- **`profile`** – Profile to build on (Optional)

Returns True(Succeed) or False

`salt.modules.neutron.`**`get_quotas_tenant`**(*profile=None*)

Fetches tenant info in server's context for following quota operation

CLI Example:

```
salt '*' neutron.get_quotas_tenant
salt '*' neutron.get_quotas_tenant profile=openstack1
```

Parameters **`profile`** – Profile to build on (Optional)

Returns Quotas information

`salt.modules.neutron.`**`list_agents`**(*profile=None*)

List agents.

CLI Example:

```
salt '*' neutron.list_agents
```

Parameters **`profile`** – Profile to build on (Optional)
Returns agents message.

`salt.modules.neutron.`**`list_extensions`**(*profile=None*)

Fetches a list of all extensions on server side

CLI Example:

```
salt '*' neutron.list_extensions
salt '*' neutron.list_extensions profile=openstack1
```

Parameters **`profile`** – Profile to build on (Optional)
Returns List of extensions

`salt.modules.neutron.`**`list_firewall_rules`**(*profile=None*)

Fetches a list of all firewall rules for a tenant CLI Example:

```
salt '*' neutron.list_firewall_rules
```

Parameters **`profile`** – Profile to build on (Optional)
Returns List of firewall rules

`salt.modules.neutron.`**`list_firewalls`**(*profile=None*)

Fetches a list of all firewalls for a tenant CLI Example:

```
salt '*' neutron.list_firewalls
```

Parameters **`profile`** – Profile to build on (Optional)
Returns List of firewalls

`salt.modules.neutron.`**`list_floatingips`**(*profile=None*)

Fetch a list of all floatingIPs for a tenant

CLI Example:

```
salt '*' neutron.list_floatingips
salt '*' neutron.list_floatingips profile=openstack1
```

Parameters **`profile`** – Profile to build on (Optional)
Returns List of floatingIP

`salt.modules.neutron.`**`list_ikepolicies`**(*profile=None*)

Fetches a list of all configured IKEPolicies for a tenant

CLI Example:

```
salt '*' neutron.list_ikepolicies
salt '*' neutron.list_ikepolicies profile=openstack1
```

Parameters **`profile`** – Profile to build on (Optional)
Returns List of IKE policy

`salt.modules.neutron.`**`list_ipsec_site_connections`**(*profile=None*)

Fetches all configured IPsec Site Connections for a tenant

CLI Example:

```
salt '*' neutron.list_ipsec_site_connections
salt '*' neutron.list_ipsec_site_connections profile=openstack1
```

Parameters **`profile`** – Profile to build on (Optional)
Returns List of IPSec site connection

`salt.modules.neutron.`**`list_ipsecpolicies`**(*profile=None*)
Fetches a list of all configured IPsecPolicies for a tenant

CLI Example:

```
salt '*' neutron.list_ipsecpolicies ipsecpolicy-name
salt '*' neutron.list_ipsecpolicies ipsecpolicy-name profile=openstack1
```

Parameters **`profile`** – Profile to build on (Optional)
Returns List of IPSec policy

`salt.modules.neutron.`**`list_l3_agent_hosting_routers`**(*router, profile=None*)
List L3 agents hosting a router.

CLI Example:

```
salt '*' neutron.list_l3_agent_hosting_routers router
```

:param router:router name or ID to query. :param profile: Profile to build on (Optional) :return: L3 agents message.

`salt.modules.neutron.`**`list_networks`**(*profile=None*)
Fetches a list of all networks for a tenant

CLI Example:

```
salt '*' neutron.list_networks
salt '*' neutron.list_networks profile=openstack1
```

Parameters **`profile`** – Profile to build on (Optional)
Returns List of network

`salt.modules.neutron.`**`list_ports`**(*profile=None*)
Fetches a list of all networks for a tenant

CLI Example:

```
salt '*' neutron.list_ports
salt '*' neutron.list_ports profile=openstack1
```

Parameters **`profile`** – Profile to build on (Optional)
Returns List of port

`salt.modules.neutron.`**`list_quotas`**(*profile=None*)
Fetches all tenants quotas

CLI Example:

```
salt '*' neutron.list_quotas
salt '*' neutron.list_quotas profile=openstack1
```

Parameters **`profile`** – Profile to build on (Optional)
Returns List of quotas

`salt.modules.neutron.`**`list_routers`**(*profile=None*)

Fetches a list of all routers for a tenant

CLI Example:

```
salt '*' neutron.list_routers
salt '*' neutron.list_routers profile=openstack1
```

Parameters **`profile`** – Profile to build on (Optional)

Returns List of router

`salt.modules.neutron.`**`list_security_group_rules`**(*profile=None*)

Fetches a list of all security group rules for a tenant

CLI Example:

```
salt '*' neutron.list_security_group_rules
salt '*' neutron.list_security_group_rules profile=openstack1
```

Parameters **`profile`** – Profile to build on (Optional)

Returns List of security group rule

`salt.modules.neutron.`**`list_security_groups`**(*profile=None*)

Fetches a list of all security groups for a tenant

CLI Example:

```
salt '*' neutron.list_security_groups
salt '*' neutron.list_security_groups profile=openstack1
```

Parameters **`profile`** – Profile to build on (Optional)

Returns List of security group

`salt.modules.neutron.`**`list_subnets`**(*profile=None*)

Fetches a list of all networks for a tenant

CLI Example:

```
salt '*' neutron.list_subnets
salt '*' neutron.list_subnets profile=openstack1
```

Parameters **`profile`** – Profile to build on (Optional)

Returns List of subnet

`salt.modules.neutron.`**`list_vpnservices`**(*retrieve_all=True, profile=None, **kwargs*)

Fetches a list of all configured VPN services for a tenant

CLI Example:

```
salt '*' neutron.list_vpnservices
```

Parameters

- **`retrieve_all`** – True or False, default: True (Optional)
- **`profile`** – Profile to build on (Optional)

Returns List of VPN service

`salt.modules.neutron.`**`remove_gateway_router`**(*router, profile=None*)

Removes an external network gateway from the specified router

CLI Example:

```
salt '*' neutron.remove_gateway_router router-name
```

Parameters
- **`router`** – ID or name of router
- **`profile`** – Profile to build on (Optional)

Returns True(Succeed) or False

`salt.modules.neutron.`**`remove_interface_router`**(*router*, *subnet*, *profile=None*)

Removes an internal network interface from the specified router

CLI Example:

```
salt '*' neutron.remove_interface_router router-name subnet-name
```

Parameters
- **`router`** – ID or name of the router
- **`subnet`** – ID or name of the subnet
- **`profile`** – Profile to build on (Optional)

Returns True(Succeed) or False

`salt.modules.neutron.`**`show_firewall`**(*firewall*, *profile=None*)

Fetches information of a specific firewall rule

CLI Example:

```
salt '*' neutron.show_firewall firewall
```

Parameters
- **`firewall`** – ID or name of firewall to look up
- **`profile`** – Profile to build on (Optional)

Returns firewall information

`salt.modules.neutron.`**`show_firewall_rule`**(*firewall_rule*, *profile=None*)

Fetches information of a specific firewall rule

CLI Example:

```
salt '*' neutron.show_firewall_rule firewall-rule-name
```

Parameters
- **`ipsecpolicy`** – ID or name of firewall rule to look up
- **`profile`** – Profile to build on (Optional)

Returns firewall rule information

`salt.modules.neutron.`**`show_floatingip`**(*floatingip_id*, *profile=None*)

Fetches information of a certain floatingIP

CLI Example:

```
salt '*' neutron.show_floatingip floatingip-id
```

Parameters
- **`floatingip_id`** – ID of floatingIP to look up
- **`profile`** – Profile to build on (Optional)

Returns Floating IP information

`salt.modules.neutron.`**`show_ikepolicy`**(*ikepolicy*, *profile=None*)

Fetches information of a specific IKEPolicy

CLI Example:

```
salt '*' neutron.show_ikepolicy ikepolicy-name
```

Parameters
- **ikepolicy** – ID or name of ikepolicy to look up
- **profile** – Profile to build on (Optional)

Returns IKE policy information

salt.modules.neutron.**show_ipsec_site_connection**(*ipsec_site_connection*, *profile=None*)

Fetches information of a specific IPsecSiteConnection

CLI Example:

```
salt '*' neutron.show_ipsec_site_connection connection-name
```

Parameters
- **ipsec_site_connection** – ID or name of ipsec site connection to look up
- **profile** – Profile to build on (Optional)

Returns IPSec site connection information

salt.modules.neutron.**show_ipsecpolicy**(*ipsecpolicy*, *profile=None*)

Fetches information of a specific IPsecPolicy

CLI Example:

```
salt '*' neutron.show_ipsecpolicy ipsecpolicy-name
```

Parameters
- **ipsecpolicy** – ID or name of IPSec policy to look up
- **profile** – Profile to build on (Optional)

Returns IPSec policy information

salt.modules.neutron.**show_network**(*network*, *profile=None*)

Fetches information of a certain network

CLI Example:

```
salt '*' neutron.show_network network-name
salt '*' neutron.show_network network-name profile=openstack1
```

Parameters
- **network** – ID or name of network to look up
- **profile** – Profile to build on (Optional)

Returns Network information

salt.modules.neutron.**show_port**(*port*, *profile=None*)

Fetches information of a certain port

CLI Example:

```
salt '*' neutron.show_port port-id
salt '*' neutron.show_port port-id profile=openstack1
```

Parameters
- **port** – ID or name of port to look up
- **profile** – Profile to build on (Optional)

Returns Port information

salt.modules.neutron.**show_quota**(*tenant_id*, *profile=None*)

Fetches information of a certain tenant's quotas

CLI Example:

```
salt '*' neutron.show_quota tenant-id
salt '*' neutron.show_quota tenant-id profile=openstack1
```

Parameters
- **`tenant_id`** – ID of tenant
- **`profile`** – Profile to build on (Optional)

Returns Quota information

`salt.modules.neutron.`**`show_router`**(*router*, *profile=None*)

Fetches information of a certain router

CLI Example:

```
salt '*' neutron.show_router router-name
```

Parameters
- **`router`** – ID or name of router to look up
- **`profile`** – Profile to build on (Optional)

Returns Router information

`salt.modules.neutron.`**`show_security_group`**(*security_group*, *profile=None*)

Fetches information of a certain security group

CLI Example:

```
salt '*' neutron.show_security_group security-group-name
```

Parameters
- **`security_group`** – ID or name of security group to look up
- **`profile`** – Profile to build on (Optional)

Returns Security group information

`salt.modules.neutron.`**`show_security_group_rule`**(*security_group_rule_id*, *profile=None*)

Fetches information of a certain security group rule

CLI Example:

```
salt '*' neutron.show_security_group_rule security-group-rule-id
```

Parameters
- **`security_group_rule_id`** – ID of security group rule to look up
- **`profile`** – Profile to build on (Optional)

Returns Security group rule information

`salt.modules.neutron.`**`show_subnet`**(*subnet*, *profile=None*)

Fetches information of a certain subnet

CLI Example:

```
salt '*' neutron.show_subnet subnet-name
```

Parameters
- **`subnet`** – ID or name of subnet to look up
- **`profile`** – Profile to build on (Optional)

Returns Subnet information

`salt.modules.neutron.`**`show_vpnservice`**(*vpnservice*, *profile=None*, ***kwargs*)

Fetches information of a specific VPN service

CLI Example:

```
salt '*' neutron.show_vpnservice vpnservice-name
```

Parameters
- **`vpnservice`** – ID or name of vpn service to look up
- **`profile`** – Profile to build on (Optional)

Returns VPN service information

`salt.modules.neutron.`**`update_firewall_rule`**(*firewall_rule, protocol=None, action=None, name=None, description=None, ip_version=None, source_ip_address=None, destination_ip_address=None, source_port=None, destination_port=None, shared=None, enabled=None, profile=None*)

Update a firewall rule

CLI Example:

```
salt '*' neutron.update_firewall_rule firewall_rule protocol=PROTOCOL⊠
↪action=ACTION
        name=NAME description=DESCRIPTION ip_version=IP_VERSION
        source_ip_address=SOURCE_IP_ADDRESS destination_ip_address=DESTINATION_IP_
↪ADDRESS
        source_port=SOURCE_PORT destination_port=DESTINATION_PORT shared=SHARED⊠
↪enabled=ENABLED
```

Parameters
- **`firewall_rule`** – ID or name of firewall rule to update.
- **`protocol`** – Protocol for the firewall rule, choose "tcp","udp","icmp" or "None". (Optional)
- **`action`** – Action for the firewall rule, choose "allow" or "deny". (Optional)
- **`name`** – Name for the firewall rule. (Optional)
- **`description`** – Description for the firewall rule. (Optional)
- **`ip_version`** – IP protocol version, default: 4. (Optional)
- **`source_ip_address`** – Source IP address or subnet. (Optional)
- **`destination_ip_address`** – Destination IP address or subnet. (Optional)
- **`source_port`** – Source port (integer in [1, 65535] or range in a:b). (Optional)
- **`destination_port`** – Destination port (integer in [1, 65535] or range in a:b). (Optional)
- **`shared`** – Set shared to True, default: False. (Optional)
- **`enabled`** – To enable this rule, default: True. (Optional)
- **`profile`** – Profile to build on (Optional)

`salt.modules.neutron.`**`update_floatingip`**(*floatingip_id, port=None, profile=None*)

Updates a floatingIP

CLI Example:

```
salt '*' neutron.update_floatingip network-name port-name
```

Parameters
- **`floatingip_id`** – ID of floatingIP
- **`port`** – ID or name of port, to associate floatingip to *None* or do not specify to disassociate the floatingip (Optional)
- **`profile`** – Profile to build on (Optional)

Returns Value of updated floating IP information `salt.modules.neutron.`**`update_network`**(*network*, *name*, *profile=None*)

Updates a network

CLI Example:

```
salt '*' neutron.update_network network-name new-network-name
```

Parameters
- **`network`** – ID or name of network to update
- **`name`** – Name of this network
- **`profile`** – Profile to build on (Optional)

Returns Value of updated network information

`salt.modules.neutron.`**`update_port`**(*port*, *name*, *admin_state_up=True*, *profile=None*)

Updates a port

CLI Example:

```
salt '*' neutron.update_port port-name network-name new-port-name
```

Parameters
- **`port`** – Port name or ID
- **`name`** – Name of this port
- **`admin_state_up`** – Set admin state up to true or false, default: true (Optional)
- **`profile`** – Profile to build on (Optional)

Returns Value of updated port information

`salt.modules.neutron.`**`update_quota`**(*tenant_id*, *subnet=None*, *router=None*, *network=None*, *floatingip=None*, *port=None*, *security_group=None*, *security_group_rule=None*, *profile=None*)

Update a tenant's quota

CLI Example:

```
salt '*' neutron.update_quota tenant-id subnet=40 router=50
                            network=10 floatingip=30 port=30
```

Parameters
- **`tenant_id`** – ID of tenant
- **`subnet`** – Value of subnet quota (Optional)
- **`router`** – Value of router quota (Optional)
- **`network`** – Value of network quota (Optional)
- **`floatingip`** – Value of floatingip quota (Optional)
- **`port`** – Value of port quota (Optional)
- **`security_group`** – Value of security group (Optional)
- **`security_group_rule`** – Value of security group rule (Optional)
- **`profile`** – Profile to build on (Optional)

Returns Value of updated quota

`salt.modules.neutron.`**`update_router`**(*router*, *name=None*, *admin_state_up=None*, *profile=None*, ***kwargs*)

Updates a router

CLI Example:

```
salt '*' neutron.update_router router_id name=new-router-name
        admin_state_up=True
```

Parameters

- **router** – ID or name of router to update
- **name** – Name of this router
- **ext_network** – ID or name of the external for the gateway (Optional)
- **admin_state_up** – Set admin state up to true or false, default: true (Optional)
- **profile** – Profile to build on (Optional)
- **kwargs** –

Returns Value of updated router information

`salt.modules.neutron.`**`update_security_group`**`(`*security_group, name=None, description=None, profile=None*`)`

Updates a security group

CLI Example:

```
salt '*' neutron.update_security_group security-group-name                 new-
↪security-group-name
```

Parameters

- **security_group** – ID or name of security group to update
- **name** – Name of this security group (Optional)
- **description** – Description of security group (Optional)
- **profile** – Profile to build on (Optional)

Returns Value of updated security group information

`salt.modules.neutron.`**`update_subnet`**`(`*subnet, name, profile=None*`)`

Updates a subnet

CLI Example:

```
salt '*' neutron.update_subnet subnet-name new-subnet-name
```

Parameters

- **subnet** – ID or name of subnet to update
- **name** – Name of this subnet
- **profile** – Profile to build on (Optional)

Returns Value of updated subnet information

`salt.modules.neutron.`**`update_vpnservice`**`(`*vpnservice, desc, profile=None*`)`

Updates a VPN service

CLI Example:

```
salt '*' neutron.update_vpnservice vpnservice-name desc='VPN Service1'
```

Parameters

- **vpnservice** – ID or name of vpn service to update
- **desc** – Set a description for the VPN service
- **profile** – Profile to build on (Optional)

Returns Value of updated VPN service information

21.9.282 salt.modules.neutronng

Neutron module for interacting with OpenStack Neutron

New in version 2018.3.0.

:depends:shade

Example configuration

```
neutron:
  cloud: default
```

```
neutron:
  auth:
    username: admin
    password: password123
    user_domain_name: mydomain
    project_name: myproject
    project_domain_name: myproject
    auth_url: https://example.org:5000/v3
  identity_api_version: 3
```

`salt.modules.neutronng.`**`compare_changes`**(*obj*, ***kwargs*)
Compare two dicts returning only keys that exist in the first dict and are different in the second one

`salt.modules.neutronng.`**`get_openstack_cloud`**(*auth=None*)
Return an openstack_cloud

`salt.modules.neutronng.`**`get_operator_cloud`**(*auth=None*)
Return an operator_cloud

`salt.modules.neutronng.`**`list_networks`**(*auth=None*, ***kwargs*)
List networks

filters A Python dictionary of filter conditions to push down

CLI Example:

```
salt '*' neutronng.list_networks
salt '*' neutronng.list_networks          filters='{"tenant_id":
↪"1dcac318a83b4610b7a7f7ba01465548"}'
```

`salt.modules.neutronng.`**`list_subnets`**(*auth=None*, ***kwargs*)
List subnets

filters A Python dictionary of filter conditions to push down

CLI Example:

```
salt '*' neutronng.list_subnets
salt '*' neutronng.list_subnets          filters='{"tenant_id":
↪"1dcac318a83b4610b7a7f7ba01465548"}'
```

`salt.modules.neutronng.`**`network_create`**(*auth=None*, ***kwargs*)
Create a network

name Name of the network being created

shared [False] If `True`, set the network as shared

admin_state_up [True] If `True`, Set the network administrative state to "up"

external [False] Control whether or not this network is externally accessible

provider An optional Python dictionary of network provider options

project_id The project ID on which this network will be created

CLI Example:

```
salt '*' neutronng.network_create name=network2           shared=True admin_state_
↪up=True external=True

salt '*' neutronng.network_create name=network3           provider='{"network_type
↪": "vlan",                     "segmentation_id": "4010",
↪"physical_network": "provider"}'           project_
↪id=1dcac318a83b4610b7a7f7ba01465548
```

`salt.modules.neutronng.`**`network_delete`**(*auth=None*, ***kwargs*)

Delete a network

name_or_id Name or ID of the network being deleted

CLI Example:

```
salt '*' neutronng.network_delete name_or_id=network1
salt '*' neutronng.network_delete name_or_id=1dcac318a83b4610b7a7f7ba01465548
```

`salt.modules.neutronng.`**`network_get`**(*auth=None*, ***kwargs*)

Get a single network

filters A Python dictionary of filter conditions to push down

CLI Example:

```
salt '*' neutronng.network_get name=XLB4
```

`salt.modules.neutronng.`**`security_group_create`**(*auth=None*, ***kwargs*)

Create a security group. Use security_group_get to create default.

project_id The project ID on which this security group will be created

CLI Example:

```
salt '*' neutronng.security_group_create name=secgroup1           description=
↪"Very secure security group"
salt '*' neutronng.security_group_create name=secgroup1           description=
↪"Very secure security group"           project_
↪id=1dcac318a83b4610b7a7f7ba01465548
```

`salt.modules.neutronng.`**`security_group_delete`**(*auth=None*, ***kwargs*)

Delete a security group

name_or_id The name or unique ID of the security group

CLI Example:

```
salt '*' neutronng.security_group_delete name_or_id=secgroup1
```

`salt.modules.neutronng.`**`security_group_get`**(*auth=None*, ***kwargs*)

Get a single security group. This will create a default security group if one does not exist yet for a particular project id.

filters A Python dictionary of filter conditions to push down

CLI Example:

```
salt '*' neutronng.security_group_get          ☒
↪name=1dcac318a83b4610b7a7f7ba01465548

salt '*' neutronng.security_group_get           name=default          filters='{
↪"tenant_id":"2e778bb64ca64a199eb526b5958d8710"}'
```

`salt.modules.neutronng.`**`security_group_rule_create`**(*auth=None*, ***kwargs*)

Create a rule in a security group

secgroup_name_or_id The security group name or ID to associate with this security group rule. If a non-unique group name is given, an exception is raised.

port_range_min The minimum port number in the range that is matched by the security group rule. If the protocol is TCP or UDP, this value must be less than or equal to the port_range_max attribute value. If nova is used by the cloud provider for security groups, then a value of None will be transformed to -1.

port_range_max The maximum port number in the range that is matched by the security group rule. The port_range_min attribute constrains the port_range_max attribute. If nova is used by the cloud provider for security groups, then a value of None will be transformed to -1.

protocol The protocol that is matched by the security group rule. Valid values are `None`, `tcp`, `udp`, and `icmp`.

remote_ip_prefix The remote IP prefix to be associated with this security group rule. This attribute matches the specified IP prefix as the source IP address of the IP packet.

remote_group_id The remote group ID to be associated with this security group rule

direction Either `ingress` or `egress`; the direction in which the security group rule is applied. For a compute instance, an ingress security group rule is applied to incoming (ingress) traffic for that instance. An egress rule is applied to traffic leaving the instance

ethertype Must be IPv4 or IPv6, and addresses represented in CIDR must match the ingress or egress rules

project_id Specify the project ID this security group will be created on (admin-only)

CLI Example:

```
salt '*' neutronng.security_group_rule_create         secgroup_name_or_
↪id=secgroup1

salt '*' neutronng.security_group_rule_create         secgroup_name_or_
↪id=secgroup2 port_range_min=8080         port_range_max=8080 direction='egress'

salt '*' neutronng.security_group_rule_create         secgroup_name_or_
↪id=c0e1d1ce-7296-405e-919d-1c08217be529         protocol=icmp project_
↪id=1dcac318a83b4610b7a7f7ba01465548
```

salt.modules.neutronng.**security_group_rule_delete**(*auth=None*, ***kwargs*)

Delete a security group

name_or_id The unique ID of the security group rule

CLI Example:

```
salt '*' neutronng.security_group_rule_delete name_or_
↪id=1dcac318a83b4610b7a7f7ba01465548
```

salt.modules.neutronng.**security_group_update**(*secgroup=None*, *auth=None*, ***kwargs*)

Update a security group

secgroup Name, ID or Raw Object of the security group to update

name New name for the security group

description New description for the security group

CLI Example:

```
salt '*' neutronng.security_group_update secgroup=secgroup1          description=
↪"Very secure security group"
salt '*' neutronng.security_group_update secgroup=secgroup1          description=
↪"Very secure security group"         project_
↪id=1dcac318a83b4610b7a7f7ba01465548
```

salt.modules.neutronng.**setup_clouds**(*auth=None*)

Call functions to create Shade cloud objects in __context__ to take advantage of Shade's in-memory caching across several states

salt.modules.neutronng.**subnet_create**(*auth=None*, ***kwargs*)

Create a subnet

network_name_or_id The unique name or ID of the attached network. If a non-unique name is supplied, an exception is raised.

cidr The CIDR

ip_version The IP version, which is 4 or 6.

enable_dhcp [False] Set to `True` if DHCP is enabled and `False` if disabled

subnet_name The name of the subnet

tenant_id The ID of the tenant who owns the network. Only administrative users can specify a tenant ID other than their own.

allocation_pools A list of dictionaries of the start and end addresses for the allocation pools.

gateway_ip The gateway IP address. When you specify both `allocation_pools` and `gateway_ip`, you must ensure that the gateway IP does not overlap with the specified allocation pools.

disable_gateway_ip [False] Set to `True` if gateway IP address is disabled and `False` if enabled. It is not allowed with `gateway_ip`.

dns_nameservers A list of DNS name servers for the subnet

host_routes A list of host route dictionaries for the subnet

ipv6_ra_mode IPv6 Router Advertisement mode. Valid values are `dhcpv6-stateful`, `dhcpv6-stateless`, or `slaac`.

ipv6_address_mode IPv6 address mode. Valid values are `dhcpv6-stateful`, `dhcpv6-stateless`, or `slaac`.

use_default_subnetpool If `True`, use the default subnetpool for `ip_version` to obtain a CIDR. It is required to pass `None` to the `cidr` argument when enabling this option.

CLI Example:

```
salt '*' neutronng.subnet_create network_name_or_id=network1
  subnet_name=subnet1

salt '*' neutronng.subnet_create subnet_name=subnet2          network_name_or_
↪id=network2 enable_dhcp=True           allocation_pools='[{"start": "192.168.
↪199.2",                              "end": "192.168.199.254"}]'         ⊠
↪gateway_ip='192.168.199.1' cidr=192.168.199.0/24

salt '*' neutronng.subnet_create network_name_or_id=network1           subnet_
↪name=subnet1 dns_nameservers='["8.8.8.8", "8.8.8.7"]'
```

salt.modules.neutronng.**subnet_delete**(*auth=None*, ***kwargs*)

Delete a subnet

name Name or ID of the subnet to update

CLI Example:

```
salt '*' neutronng.subnet_delete name=subnet1
salt '*' neutronng.subnet_delete           name=1dcac318a83b4610b7a7f7ba01465548
```

salt.modules.neutronng.**subnet_get**(*auth=None*, ***kwargs*)

Get a single subnet

filters A Python dictionary of filter conditions to push down

CLI Example:

```
salt '*' neutronng.subnet_get name=subnet1
```

salt.modules.neutronng.**subnet_update**(*auth=None*, ***kwargs*)

Update a subnet

name_or_id Name or ID of the subnet to update

subnet_name The new name of the subnet

enable_dhcp Set to `True` if DHCP is enabled and `False` if disabled

gateway_ip The gateway IP address. When you specify both allocation_pools and gateway_ip, you must ensure that the gateway IP does not overlap with the specified allocation pools.

disable_gateway_ip [False] Set to `True` if gateway IP address is disabled and False if enabled. It is not allowed with `gateway_ip`.

allocation_pools A list of dictionaries of the start and end addresses for the allocation pools.

dns_nameservers A list of DNS name servers for the subnet

host_routes A list of host route dictionaries for the subnet

```
salt '*' neutronng.subnet_update name=subnet1 subnet_name=subnet2
salt '*' neutronng.subnet_update name=subnet1 dns_nameservers='["8.8.8.8", "8.8.8.
↪7"]'
```

21.9.283 salt.modules.nfs3

Module for managing NFS version 3.

`salt.modules.nfs3.`**`add_export`**`(`*exports='/etc/exports', path=None, hosts=None, options=None*`)`

Add an export

CLI Example:

```
salt '*' nfs3.add_export path='/srv/test' hosts='127.0.0.1' options=['rw']
```

`salt.modules.nfs3.`**`del_export`**`(`*exports='/etc/exports', path=None*`)`

Remove an export

CLI Example:

```
salt '*' nfs.del_export /media/storage
```

`salt.modules.nfs3.`**`list_exports`**`(`*exports='/etc/exports'*`)`

List configured exports

CLI Example:

```
salt '*' nfs.list_exports
```

`salt.modules.nfs3.`**`reload_exports`**`()`

Trigger a reload of the exports file to apply changes

CLI Example:

```
salt '*' nfs3.reload_exports
```

21.9.284 salt.modules.nftables

Support for nftables

`salt.modules.nftables.`**`append`**`(`*table='filter', chain=None, rule=None, family='ipv4'*`)`

Append a rule to the specified table & chain.

This function accepts a rule in a standard nftables command format, starting with the chain. Trying to force users to adapt to a new method of creating rules would be irritating at best, and we already have a parser that can handle it.

CLI Example:

```
salt '*' nftables.append filter input \
    rule='tcp dport 22 log accept'

IPv6:
salt '*' nftables.append filter input \
    rule='tcp dport 22 log accept' \
    family=ipv6
```

`salt.modules.nftables.`**`build_rule`**(*table=None, chain=None, command=None, position=", full=None, family='ipv4', **kwargs*)

Build a well-formatted nftables rule based on kwargs. A *table* and *chain* are not required, unless *full* is True.

If *full* is *True*, then *table*, *chain* and *command* are required. *command* may be specified as either insert, append, or delete. This will return the nftables command, exactly as it would be used from the command line.

If a position is required (as with *insert* or *delete*), it may be specified as *position*. This will only be useful if *full* is True.

If *connstate* is passed in, it will automatically be changed to *state*.

CLI Examples:

```
salt '*' nftables.build_rule match=state \
    connstate=RELATED,ESTABLISHED jump=ACCEPT
salt '*' nftables.build_rule filter input command=insert position=3 \
    full=True match=state state=related,established jump=accept

IPv6:
salt '*' nftables.build_rule match=state \
    connstate=related,established jump=accept \
    family=ipv6
salt '*' nftables.build_rule filter input command=insert position=3 \
    full=True match=state state=related,established jump=accept \
    family=ipv6
```

`salt.modules.nftables.`**`check`**(*table='filter', chain=None, rule=None, family='ipv4'*)

Check for the existence of a rule in the table and chain

This function accepts a rule in a standard nftables command format, starting with the chain. Trying to force users to adapt to a new method of creating rules would be irritating at best, and we already have a parser that can handle it.

CLI Example:

```
salt '*' nftables.check filter input \
    rule='tcp dport 22 log accept'

IPv6:
salt '*' nftables.check filter input \
    rule='tcp dport 22 log accept' \
    family=ipv6
```

`salt.modules.nftables.`**`check_chain`**(*table='filter', chain=None, family='ipv4'*)

New in version 2014.7.0.

Check for the existence of a chain in the table

CLI Example:

```
salt '*' nftables.check_chain filter input

IPv6:
salt '*' nftables.check_chain filter input family=ipv6
```

`salt.modules.nftables.`**`check_table`**(*table=None, family='ipv4'*)

Check for the existence of a table

CLI Example:

```
salt '*' nftables.check_table nat
```

`salt.modules.nftables.`**`delete`**`(`*table, chain=None, position=None, rule=None, family='ipv4'*`)`

Delete a rule from the specified table & chain, specifying either the rule in its entirety, or the rule's position in the chain.

This function accepts a rule in a standard nftables command format, starting with the chain. Trying to force users to adapt to a new method of creating rules would be irritating at best, and we already have a parser that can handle it.

CLI Examples:

```
salt '*' nftables.delete filter input position=3

salt '*' nftables.delete filter input \
    rule='tcp dport 22 log accept'

IPv6:
salt '*' nftables.delete filter input position=3 family=ipv6

salt '*' nftables.delete filter input \
    rule='tcp dport 22 log accept' \
    family=ipv6
```

`salt.modules.nftables.`**`delete_chain`**`(`*table='filter', chain=None, family='ipv4'*`)`

New in version 2014.7.0.

Delete the chain from the specified table.

CLI Example:

```
salt '*' nftables.delete_chain filter input

salt '*' nftables.delete_chain filter foo

IPv6:
salt '*' nftables.delete_chain filter input family=ipv6

salt '*' nftables.delete_chain filter foo family=ipv6
```

`salt.modules.nftables.`**`delete_table`**`(`*table, family='ipv4'*`)`

New in version 2014.7.0.

Create new custom table.

CLI Example:

```
salt '*' nftables.delete_table filter

IPv6:
salt '*' nftables.delete_table filter family=ipv6
```

`salt.modules.nftables.`**`flush`**`(`*table='filter', chain='', family='ipv4'*`)`

Flush the chain in the specified table, flush all chains in the specified table if chain is not specified.

CLI Example:

```
salt '*' nftables.flush filter

salt '*' nftables.flush filter input
```

(continues on next page)

(continued from previous page)

```
IPv6:
salt '*' nftables.flush filter input family=ipv6
```

salt.modules.nftables.**get_rule_handle**(*table='filter'*, *chain=None*, *rule=None*, *family='ipv4'*)
Get the handle for a particular rule

This function accepts a rule in a standard nftables command format, starting with the chain. Trying to force users to adapt to a new method of creating rules would be irritating at best, and we already have a parser that can handle it.

CLI Example:

```
salt '*' nftables.get_rule_handle filter input \
    rule='tcp dport 22 log accept'

IPv6:
salt '*' nftables.get_rule_handle filter input \
    rule='tcp dport 22 log accept' \
    family=ipv6
```

salt.modules.nftables.**get_rules**(*family='ipv4'*)
Return a data structure of the current, in-memory rules

CLI Example:

```
salt '*' nftables.get_rules

salt '*' nftables.get_rules family=ipv6
```

salt.modules.nftables.**get_saved_rules**(*conf_file=None*)
Return a data structure of the rules in the conf file

CLI Example:

```
salt '*' nftables.get_saved_rules
```

salt.modules.nftables.**insert**(*table='filter'*, *chain=None*, *position=None*, *rule=None*, *family='ipv4'*)
Insert a rule into the specified table & chain, at the specified position.

If position is not specified, rule will be inserted in first position.

This function accepts a rule in a standard nftables command format, starting with the chain. Trying to force users to adapt to a new method of creating rules would be irritating at best, and we already have a parser that can handle it.

CLI Examples:

```
salt '*' nftables.insert filter input \
    rule='tcp dport 22 log accept'

salt '*' nftables.insert filter input position=3 \
    rule='tcp dport 22 log accept'

IPv6:
salt '*' nftables.insert filter input \
    rule='tcp dport 22 log accept' \
    family=ipv6

salt '*' nftables.insert filter input position=3 \
```

(continues on next page)

(continued from previous page)

```
    rule='tcp dport 22 log accept' \
    family=ipv6
```

`salt.modules.nftables.`**`list_tables`**(*family='ipv4'*)

Return a data structure of the current, in-memory tables

CLI Example:

```
salt '*' nftables.list_tables

salt '*' nftables.list_tables family=ipv6
```

`salt.modules.nftables.`**`new_chain`**(*table='filter', chain=None, table_type=None, hook=None, priority=None, family='ipv4'*)

New in version 2014.7.0.

Create new chain to the specified table.

CLI Example:

```
salt '*' nftables.new_chain filter input

salt '*' nftables.new_chain filter input \
        table_type=filter hook=input priority=0

salt '*' nftables.new_chain filter foo

IPv6:
salt '*' nftables.new_chain filter input family=ipv6

salt '*' nftables.new_chain filter input \
        table_type=filter hook=input priority=0 family=ipv6

salt '*' nftables.new_chain filter foo family=ipv6
```

`salt.modules.nftables.`**`new_table`**(*table, family='ipv4'*)

New in version 2014.7.0.

Create new custom table.

CLI Example:

```
salt '*' nftables.new_table filter

IPv6:
salt '*' nftables.new_table filter family=ipv6
```

`salt.modules.nftables.`**`save`**(*filename=None, family='ipv4'*)

Save the current in-memory rules to disk

CLI Example:

```
salt '*' nftables.save /etc/nftables
```

`salt.modules.nftables.`**`version`**()

Return version from nftables –version

CLI Example:

```
salt '*' nftables.version
```

21.9.285 salt.modules.nginx

Support for nginx

`salt.modules.nginx.`**`build_info`**`()`

Return server and build arguments

CLI Example:

```
salt '*' nginx.build_info
```

`salt.modules.nginx.`**`configtest`**`()`

test configuration and exit

CLI Example:

```
salt '*' nginx.configtest
```

`salt.modules.nginx.`**`signal`**`(`*signal=None*`)`

Signals nginx to start, reload, reopen or stop.

CLI Example:

```
salt '*' nginx.signal reload
```

`salt.modules.nginx.`**`status`**`(`*url='http://127.0.0.1/status'*`)`

Return the data from an Nginx status page as a dictionary. http://wiki.nginx.org/HttpStubStatusModule

url The URL of the status page. Defaults to 'http://127.0.0.1/status'

CLI Example:

```
salt '*' nginx.status
```

`salt.modules.nginx.`**`version`**`()`

Return server version from nginx -v

CLI Example:

```
salt '*' nginx.version
```

21.9.286 salt.modules.nilrt_ip module

The networking module for NI Linux Real-Time distro

`salt.modules.nilrt_ip.`**`apply_network_settings`**`(`***settings*`)`

Apply global network configuration.

CLI Example:

```
salt '*' ip.apply_network_settings
```

`salt.modules.nilrt_ip.`**`build_interface`**`(`*iface, iface_type, enabled, **settings*`)`

Build an interface script for a network interface.

CLI Example:

```
salt '*' ip.build_interface eth0 eth <settings>
```

`salt.modules.nilrt_ip.`**`build_network_settings`**`(`*`**settings`*`)`

Build the global network script.

CLI Example:

```
salt '*' ip.build_network_settings <settings>
```

`salt.modules.nilrt_ip.`**`disable`**`(`*interface*`)`

Disable the specified interface

Change adapter mode to Disabled. If previous adapter mode was EtherCAT, the target will need reboot.

Parameters **`interface`** (`str`) – interface label

Returns True if the service was disabled, otherwise an exception will be thrown.

Return type bool

CLI Example:

```
salt '*' ip.disable interface-label
```

`salt.modules.nilrt_ip.`**`down`**`(`*interface, iface_type=None*`)`

Disable the specified interface

Change adapter mode to Disabled. If previous adapter mode was EtherCAT, the target will need reboot.

Parameters **`interface`** (`str`) – interface label

Returns True if the service was disabled, otherwise an exception will be thrown.

Return type bool

CLI Example:

```
salt '*' ip.down interface-label
```

`salt.modules.nilrt_ip.`**`enable`**`(`*interface*`)`

Enable the specified interface

Change adapter mode to TCP/IP. If previous adapter mode was EtherCAT, the target will need reboot.

Parameters **`interface`** (`str`) – interface label

Returns True if the service was enabled, otherwise an exception will be thrown.

Return type bool

CLI Example:

```
salt '*' ip.enable interface-label
```

`salt.modules.nilrt_ip.`**`get_interface`**`(`*iface*`)`

Returns details about given interface.

CLI Example:

```
salt '*' ip.get_interface eth0
```

`salt.modules.nilrt_ip.`**`get_interfaces_details`**`()`

Get details about all the interfaces on the minion

Returns information about all interfaces omitting loopback

Return type dictionary

CLI Example:

```
salt '*' ip.get_interfaces_details
```

`salt.modules.nilrt_ip.`**`get_network_settings`**`()`

Return the contents of the global network script.

CLI Example:

```
salt '*' ip.get_network_settings
```

`salt.modules.nilrt_ip.`**`set_dhcp_linklocal_all`**`(`*interface*`)`

Configure specified adapter to use DHCP with linklocal fallback

Change adapter mode to TCP/IP. If previous adapter mode was EtherCAT, the target will need reboot.

Parameters **`interface`** (*`str`*) – interface label

Returns True if the settings were applied, otherwise an exception will be thrown.

Return type bool

CLI Example:

```
salt '*' ip.set_dhcp_linklocal_all interface-label
```

`salt.modules.nilrt_ip.`**`set_dhcp_only_all`**`(`*interface*`)`

Configure specified adapter to use DHCP only

Change adapter mode to TCP/IP. If previous adapter mode was EtherCAT, the target will need reboot.

Parameters **`interface`** (*`str`*) – interface label

Returns True if the settings were applied, otherwise an exception will be thrown.

Return type bool

CLI Example:

```
salt '*' ip.dhcp_only_all interface-label
```

`salt.modules.nilrt_ip.`**`set_ethercat`**`(`*interface*, *master_id*`)`

Configure specified adapter to use EtherCAT adapter mode. If successful, the target will need reboot if it doesn't already use EtherCAT adapter mode, otherwise will return true.

Parameters

- **`interface`** – interface label
- **`master_id`** – EtherCAT Master ID

Returns True if the settings were applied, otherwise an exception will be thrown.

CLI Example:

```
salt '*' ip.set_ethercat interface-label master-id
```

`salt.modules.nilrt_ip.`**`set_linklocal_only_all`**`(`*interface*`)`

Configure specified adapter to use linklocal only

Change adapter mode to TCP/IP. If previous adapter mode was EtherCAT, the target will need reboot.

Parameters **`interface`** (*`str`*) – interface label

Returns True if the settings were applied, otherwise an exception will be thrown.

Return type bool

CLI Example:

```
salt '*' ip.linklocal_only_all interface-label
```

`salt.modules.nilrt_ip.`**`set_static_all`**`(`*interface*, *address*, *netmask*, *gateway*, *nameservers*`)`

Configure specified adapter to use ipv4 manual settings

Change adapter mode to TCP/IP. If previous adapter mode was EtherCAT, the target will need reboot.

Parameters

- **`interface`** (*`str`*) – interface label
- **`address`** (*`str`*) – ipv4 address

- **`netmask`** (*`str`*) – ipv4 netmask
- **`gateway`** (*`str`*) – ipv4 gateway
- **`nameservers`** (*`str`*) – list of nameservers servers separated by spaces

Returns True if the settings were applied, otherwise an exception will be thrown.

Return type bool

CLI Example:

```
salt '*' ip.set_static_all interface-label address netmask gateway nameservers
```

`salt.modules.nilrt_ip.`**`up`**`(`*interface, iface_type=None*`)`

Enable the specified interface

Change adapter mode to TCP/IP. If previous adapter mode was EtherCAT, the target will need reboot.

Parameters **`interface`** (*`str`*) – interface label

Returns True if the service was enabled, otherwise an exception will be thrown.

Return type bool

CLI Example:

```
salt '*' ip.up interface-label
```

21.9.287 salt.modules.nix

Work with Nix packages

New in version 2017.7.0.

Does not require the machine to be Nixos, just have Nix installed and available to use for the user running this command. Their profile must be located in their home, under `$HOME/.nix-profile/`, and the nix store, unless specially set up, should be in `/nix`. To easily use this with multiple users or a root user, set up the nix-daemon.

This module exposes most of the common nix operations. Currently not meant to be run as a `pkg` module, but explicitly as `nix.*`.

For more information on nix, see the nix documentation.

`salt.modules.nix.`**`collect_garbage`**`()`

Completely removed all currently 'uninstalled' packages in the nix store.

Tells the user how many store paths were removed and how much space was freed.

Returns How much space was freed and how many derivations were removed

Return type str

Warning: This is a destructive action on the nix store.

```
salt '*' nix.collect_garbage
```

`salt.modules.nix.`**`install`**`(`**pkgs, **kwargs*`)`

Installs a single or multiple packages via nix

Parameters

- **`pkgs`** (*`list(str)`*) – packages to update
- **`attributes`** (*`bool`*) – Pass the list of packages or single package as attribues, not package names. default: False

Returns Installed packages. Example element: `gcc-3.3.2`

Return type list(str)

```
salt '*' nix.install package [package2 ...]
salt '*' nix.install attributes=True attr.name [attr.name2 ...]
```

`salt.modules.nix.`**`list_pkgs`**(*installed=True, attributes=True*)

Lists installed packages. Due to how nix works, it defaults to just doing a `nix-env -q`.

Parameters

- **`installed`** (*bool*) – list only installed packages. This can be a very long list (12,000+ elements), so caution is advised. Default: True
- **`attributes`** (*bool*) – show the attributes of the packages when listing all packages. Default: True

Returns Packages installed or available, along with their attributes.

Return type list(list(str))

```
salt '*' nix.list_pkgs
salt '*' nix.list_pkgs installed=False
```

`salt.modules.nix.`**`uninstall`**(**pkgs*)

Erases a package from the current nix profile. Nix uninstalls work differently than other package managers, and the symlinks in the profile are removed, while the actual package remains. There is also a `nix.purge` function, to clear the package cache of unused packages.

Parameters **`pkgs`** (*list(str)*) – List, single package to uninstall

Returns Packages that have been uninstalled

Return type list(str)

```
salt '*' nix.uninstall pkg1 [pkg2 ...]
```

`salt.modules.nix.`**`upgrade`**(**pkgs*)

Runs an update operation on the specified packages, or all packages if none is specified.

Parameters **`pkgs`** (*list(str)*) – List of packages to update

Returns The upgraded packages. Example element: `['libxslt-1.1.0', 'libxslt-1.1.10']`

Return type list(tuple(str, str))

```
salt '*' nix.update
salt '*' nix.update pkgs=one,two
```

21.9.288 salt.modules.nova

Module for handling OpenStack Nova calls

depends

- novaclient Python module

configuration This module is not usable until the user, password, tenant, and auth URL are specified either in a pillar or in the minion's config file. For example:

```
keystone.user: admin
keystone.password: verybadpass
keystone.tenant: admin
keystone.auth_url: 'http://127.0.0.1:5000/v2.0/'
# Optional
keystone.region_name: 'RegionOne'
```

If configuration for multiple OpenStack accounts is required, they can be set up as different configuration profiles: For example:

```
openstack1:
  keystone.user: admin
  keystone.password: verybadpass
  keystone.tenant: admin
  keystone.auth_url: 'http://127.0.0.1:5000/v2.0/'

openstack2:
  keystone.user: admin
  keystone.password: verybadpass
  keystone.tenant: admin
  keystone.auth_url: 'http://127.0.0.2:5000/v2.0/'
```

With this configuration in place, any of the nova functions can make use of a configuration profile by declaring it explicitly. For example:

```
salt '*' nova.flavor_list profile=openstack1
```

To use keystoneauth1 instead of keystoneclient, include the *use_keystoneauth* option in the pillar or minion config.

Note: This is required to use keystone v3 as for authentication.

```
keystone.user: admin
keystone.password: verybadpass
keystone.tenant: admin
keystone.auth_url: 'http://127.0.0.1:5000/v3/'
keystone.use_keystoneauth: true
keystone.verify: '/path/to/custom/certs/ca-bundle.crt'
```

Note: By default the nova module will attempt to verify its connection utilizing the system certificates. If you need to verify against another bundle of CA certificates or want to skip verification altogether you will need to specify the *verify* option. You can specify True or False to verify (or not) against system certificates, a path to a bundle or CA certs to check against, or None to allow keystoneauth to search for the certificates on its own. (defaults to True)

`salt.modules.nova.`**`boot`**(*name*, *flavor_id=0*, *image_id=0*, *profile=None*, *timeout=300*)

Boot (create) a new instance

name Name of the new instance (must be first)

flavor_id Unique integer ID for the flavor

image_id Unique integer ID for the image

timeout How long to wait, after creating the instance, for the provider to return information about it (default 300 seconds).

New in version 2014.1.0.

CLI Example:

```
salt '*' nova.boot myinstance flavor_id=4596 image_id=2
```

The flavor_id and image_id are obtained from nova.flavor_list and nova.image_list

```
salt '*' nova.flavor_list
salt '*' nova.image_list
```

`salt.modules.nova.`**`delete`**(*instance_id*, *profile=None*)

Delete an instance

instance_id ID of the instance to be deleted

CLI Example:

```
salt '*' nova.delete 1138
```

`salt.modules.nova.`**`flavor_create`**(*name*, *flavor_id=0*, *ram=0*, *disk=0*, *vcpus=1*, *profile=None*)

Add a flavor to nova (nova flavor-create). The following parameters are required:

name Name of the new flavor (must be first)

flavor_id Unique integer ID for the new flavor

ram Memory size in MB

disk Disk size in GB

vcpus Number of vcpus

CLI Example:

```
salt '*' nova.flavor_create myflavor flavor_id=6 ram=4096 disk=10 vcpus=1
```

`salt.modules.nova.`**`flavor_delete`**(*flavor_id*, *profile=None*)

Delete a flavor from nova by id (nova flavor-delete)

CLI Example:

```
salt '*' nova.flavor_delete 7
```

`salt.modules.nova.`**`flavor_list`**(*profile=None*)

Return a list of available flavors (nova flavor-list)

CLI Example:

```
salt '*' nova.flavor_list
```

`salt.modules.nova.`**`image_list`**(*name=None*, *profile=None*)

Return a list of available images (nova images-list + nova image-show) If a name is provided, only that image will be displayed.

CLI Examples:

```
salt '*' nova.image_list
salt '*' nova.image_list myimage
```

`salt.modules.nova.`**`image_meta_delete`**(*image_id=None*, *name=None*, *keys=None*, *profile=None*)

Delete a key=value pair from the metadata for an image (nova image-meta set)

CLI Examples:

```
salt '*' nova.image_meta_delete 6f52b2ff-0b31-4d84-8fd1-af45b84824f6 keys=cheese
salt '*' nova.image_meta_delete name=myimage keys=salad,beans
```

`salt.modules.nova.`**`image_meta_set`**(*image_id=None*, *name=None*, *profile=None*, ***kwargs*)

Sets a key=value pair in the metadata for an image (nova image-meta set)

CLI Examples:

```
salt '*' nova.image_meta_set 6f52b2ff-0b31-4d84-8fd1-af45b84824f6 cheese=gruyere
salt '*' nova.image_meta_set name=myimage salad=pasta beans=baked
```

`salt.modules.nova.`**`keypair_add`**(*name*, *pubfile=None*, *pubkey=None*, *profile=None*)

Add a keypair to nova (nova keypair-add)

CLI Examples:

```
salt '*' nova.keypair_add mykey pubfile=/home/myuser/.ssh/id_rsa.pub
salt '*' nova.keypair_add mykey pubkey='ssh-rsa <key> myuser@mybox'
```

`salt.modules.nova.`**`keypair_delete`**(*name*, *profile=None*)

Add a keypair to nova (nova keypair-delete)

CLI Example:

```
salt '*' nova.keypair_delete mykey
```

`salt.modules.nova.`**`keypair_list`**(*profile=None*)

Return a list of available keypairs (nova keypair-list)

CLI Example:

```
salt '*' nova.keypair_list
```

`salt.modules.nova.`**`list_`**(*profile=None*)

To maintain the feel of the nova command line, this function simply calls the server_list function.

CLI Example:

```
salt '*' nova.list
```

`salt.modules.nova.`**`lock`**(*instance_id*, *profile=None*)

Lock an instance

instance_id ID of the instance to be locked

CLI Example:

```
salt '*' nova.lock 1138
```

`salt.modules.nova.`**`resume`**(*instance_id*, *profile=None*)

Resume an instance

instance_id ID of the instance to be resumed

CLI Example:

```
salt '*' nova.resume 1138
```

`salt.modules.nova.`**`secgroup_create`**(*name*, *description*, *profile=None*)

Add a secgroup to nova (nova secgroup-create)

CLI Example:

```
salt '*' nova.secgroup_create mygroup 'This is my security group'
```

`salt.modules.nova.`**`secgroup_delete`**(*name*, *profile=None*)

Delete a secgroup to nova (nova secgroup-delete)

CLI Example:

```
salt '*' nova.secgroup_delete mygroup
```

`salt.modules.nova.`**`secgroup_list`**(*profile=None*)

Return a list of available security groups (nova items-list)

CLI Example:

```
salt '*' nova.secgroup_list
```

`salt.modules.nova.`**`server_by_name`**(*name*, *profile=None*)

Return information about a server

name Server Name

CLI Example:

```
salt '*' nova.server_by_name myserver profile=openstack
```

`salt.modules.nova.`**`server_list`**(*profile=None*)

Return list of active servers

CLI Example:

```
salt '*' nova.server_list
```

`salt.modules.nova.`**`server_list_detailed`**(*profile=None*)

Return detailed list of active servers

CLI Example:

```
salt '*' nova.server_list_detailed
```

`salt.modules.nova.`**`server_show`**(*server_id*, *profile=None*)

Return detailed information for an active server

CLI Example:

```
salt '*' nova.server_show <server_id>
```

`salt.modules.nova.`**`show`**(*server_id*, *profile=None*)

To maintain the feel of the nova command line, this function simply calls the server_show function.

CLI Example:

```
salt '*' nova.show
```

`salt.modules.nova.`**`suspend`**(*instance_id*, *profile=None*)

Suspend an instance

instance_id ID of the instance to be suspended

CLI Example:

```
salt '*' nova.suspend 1138
```

`salt.modules.nova.`**`volume_attach`**(*name*, *server_name*, *device='/dev/xvdb'*, *profile=None*, *timeout=300*)

Attach a block storage volume

name Name of the new volume to attach

server_name Name of the server to attach to

device Name of the device on the server

profile Profile to build on

CLI Example:

```
salt '*' nova.volume_attach myblock slice.example.com profile=openstack
salt '*' nova.volume_attach myblock server.example.com device='/dev/xvdb'
↪profile=openstack
```

`salt.modules.nova.`**`volume_create`**(*name, size=100, snapshot=None, voltype=None, profile=None*)

Create a block storage volume

name Name of the new volume (must be first)

size Volume size

snapshot Block storage snapshot id

voltype Type of storage

profile Profile to build on

CLI Example:

```
salt '*' nova.volume_create myblock size=300 profile=openstack
```

`salt.modules.nova.`**`volume_delete`**(*name, profile=None*)

Destroy the volume

name Name of the volume

profile Profile to build on

CLI Example:

```
salt '*' nova.volume_delete myblock profile=openstack
```

`salt.modules.nova.`**`volume_detach`**(*name, profile=None, timeout=300*)

Attach a block storage volume

name Name of the new volume to attach

server_name Name of the server to detach from

profile Profile to build on

CLI Example:

```
salt '*' nova.volume_detach myblock profile=openstack
```

`salt.modules.nova.`**`volume_list`**(*search_opts=None, profile=None*)

List storage volumes

search_opts Dictionary of search options

profile Profile to use

CLI Example:

```
salt '*' nova.volume_list search_opts='{"display_name": "myblock"}' ⏎
↪profile=openstack
```

`salt.modules.nova.`**`volume_show`**(*name, profile=None*)

Create a block storage volume

name Name of the volume

profile Profile to use

CLI Example:

```
salt '*' nova.volume_show myblock profile=openstack
```

21.9.289 salt.modules.npm

Manage and query NPM packages.

`salt.modules.npm.`**`cache_clean`**(*path=None, runas=None, env=None, force=False*)

Clean cached NPM packages.

If no path for a specific package is provided the entire cache will be cleared.

path The cache subpath to delete, or None to clear the entire cache

runas The user to run NPM with

env Environment variables to set when invoking npm. Uses the same `env` format as the *cmd.run* execution function.

force Force cleaning of cache. Required for npm@5 and greater

New in version 2016.11.6.

CLI Example:

```
salt '*' npm.cache_clean force=True
```

`salt.modules.npm.`**`cache_list`**(*path=None*, *runas=None*, *env=None*)

List NPM cached packages.

If no path for a specific package is provided this will list all the cached packages.

path The cache subpath to list, or None to list the entire cache

runas The user to run NPM with

env Environment variables to set when invoking npm. Uses the same `env` format as the *cmd.run* execution function.

CLI Example:

```
salt '*' npm.cache_clean
```

`salt.modules.npm.`**`cache_path`**(*runas=None*, *env=None*)

List path of the NPM cache directory.

runas The user to run NPM with

env Environment variables to set when invoking npm. Uses the same `env` format as the *cmd.run* execution function.

CLI Example:

```
salt '*' npm.cache_path
```

`salt.modules.npm.`**`install`**(*pkg=None*, *pkgs=None*, *dir=None*, *runas=None*, *registry=None*, *env=None*, *dry_run=False*, *silent=True*)

Install an NPM package.

If no directory is specified, the package will be installed globally. If no package is specified, the dependencies (from package.json) of the package in the given directory will be installed.

pkg A package name in any format accepted by NPM, including a version identifier

pkgs A list of package names in the same format as the `name` parameter

New in version 2014.7.0.

dir The target directory in which to install the package, or None for global installation

runas The user to run NPM with

registry The NPM registry to install the package from.

New in version 2014.7.0.

env Environment variables to set when invoking npm. Uses the same `env` format as the *cmd.run* execution function.

New in version 2014.7.0.

silent Whether or not to run NPM install with –silent flag.

New in version 2016.3.0.

dry_run Whether or not to run NPM install with –dry-run flag.

New in version 2015.8.4.

silent Whether or not to run NPM install with –silent flag.

New in version 2015.8.5.

CLI Example:

```
salt '*' npm.install coffee-script

salt '*' npm.install coffee-script@1.0.1
```

`salt.modules.npm.`**`list_`**`(`*pkg=None*, *dir=None*, *runas=None*, *env=None*, *depth=None*`)`
List installed NPM packages.

If no directory is specified, this will return the list of globally- installed packages.
pkg Limit package listing by name
dir The directory whose packages will be listed, or None for global installation
runas The user to run NPM with

New in version 2014.7.0.

env Environment variables to set when invoking npm. Uses the same `env` format as the *`cmd.run`* execution function.

New in version 2014.7.0.

depth Limit the depth of the packages listed

New in version 2016.11.6,2017.7.0.

CLI Example:

```
salt '*' npm.list
```

`salt.modules.npm.`**`uninstall`**`(`*pkg*, *dir=None*, *runas=None*, *env=None*`)`
Uninstall an NPM package.

If no directory is specified, the package will be uninstalled globally.
pkg A package name in any format accepted by NPM
dir The target directory from which to uninstall the package, or None for global installation
runas The user to run NPM with
env Environment variables to set when invoking npm. Uses the same `env` format as the *`cmd.run`* execution function.

New in version 2015.5.3.

CLI Example:

```
salt '*' npm.uninstall coffee-script
```

21.9.290 salt.modules.nspawn

Manage nspawn containers

New in version 2015.8.0.

systemd-nspawn(1) is a tool used to manage lightweight namespace containers. This execution module provides several functions to help manage these containers.

Minions running systemd >= 219 will place new containers in `/var/lib/machines`, while those running systemd < 219 will place them in `/var/lib/container`.

`salt.modules.nspawn.`**`bootstrap_container`**`(`*name*, *dist=None*, *version=None*`)`
Bootstrap a container from package servers, if dist is None the os the minion is running as will be created, otherwise the needed bootstrapping tools will need to be available on the host.

CLI Example:

```
salt myminion nspawn.bootstrap_container <name>
```

`salt.modules.nspawn.`**`bootstrap_salt`**`(`*name, config=None, approve_key=True, install=True, pub_key=None, priv_key=None, bootstrap_url=None, force_install=False, unconditional_install=False, bootstrap_delay=None, bootstrap_args=None, bootstrap_shell=None*`)`

Bootstrap a container from package servers, if dist is None the os the minion is running as will be created, otherwise the needed bootstrapping tools will need to be available on the host.

CLI Example:

```
salt '*' nspawn.bootstrap_salt arch1
```

`salt.modules.nspawn.`**`copy_to`**`(`*name, source, dest, overwrite=False, makedirs=False*`)`

Copy a file from the host into a container

name Container name

source File to be copied to the container

dest Destination on the container. Must be an absolute path.

overwrite [False] Unless this option is set to `True`, then if a file exists at the location specified by the `dest` argument, an error will be raised.

makedirs : False

Create the parent directory on the container if it does not already exist.

CLI Example:

```
salt 'minion' nspawn.copy_to /tmp/foo /root/foo
```

`salt.modules.nspawn.`**`disable`**`(`*name*`)`

Set the named container to *not* be launched at boot

CLI Example:

```
salt myminion nspawn.enable <name>
```

`salt.modules.nspawn.`**`enable`**`(`*name*`)`

Set the named container to be launched at boot

CLI Example:

```
salt myminion nspawn.enable <name>
```

`salt.modules.nspawn.`**`exists`**`(`*name*`)`

Returns true if the named container exists

CLI Example:

```
salt myminion nspawn.exists <name>
```

`salt.modules.nspawn.`**`info`**`(`*name, **kwargs*`)`

Return info about a container

Note: The container must be running for `machinectl` to gather information about it. If the container is stopped, then this function will start it.

start [False] If `True`, then the container will be started to retrieve the info. A `Started` key will be in the return data if the container was started.

CLI Example:

```
salt myminion nspawn.info arch1
salt myminion nspawn.info arch1 force_start=False
```

`salt.modules.nspawn.`**`list_all`**`()`

Lists all nspawn containers

CLI Example:

```
salt myminion nspawn.list_all
```

`salt.modules.nspawn.`**`list_running`**`()`

Lists running nspawn containers

Note: `nspawn.list` also works to list running containers

CLI Example:

```
salt myminion nspawn.list_running
salt myminion nspawn.list
```

`salt.modules.nspawn.`**`list_stopped`**`()`

Lists stopped nspawn containers

CLI Example:

```
salt myminion nspawn.list_stopped
```

`salt.modules.nspawn.`**`pid`**`(`*name*`)`

Returns the PID of a container

name Container name

CLI Example:

```
salt myminion nspawn.pid arch1
```

`salt.modules.nspawn.`**`poweroff`**`(`*name*`)`

Issue a clean shutdown to the container. Equivalent to running `machinectl poweroff` on the named container.

For convenience, running `` nspawn.stop``(as shown in the CLI examples below) is equivalent to running ``nspawn.poweroff ``.

Note: `machinectl poweroff` is only supported in systemd >= 219. On earlier systemd versions, running this function will simply issue a clean shutdown via `systemctl`.

CLI Examples:

```
salt myminion nspawn.poweroff arch1
salt myminion nspawn.stop arch1
```

`salt.modules.nspawn.`**`pull_dkr`**`(`*url*, *name*, *index*`)`

Execute a `machinectl pull-dkr` to download a docker image and add it to /var/lib/machines as a new container.

Note: Requires systemd >= 219

---

url URL from which to download the container

name Name for the new container

index URL of the Docker index server from which to pull (must be an `http://` or `https://` URL).

CLI Examples:

```
salt myminion nspawn.pull_dkr centos/centos6 cent6 index=https://get.docker.com
salt myminion nspawn.pull_docker centos/centos6 cent6 index=https://get.docker.com
```

`salt.modules.nspawn.`**`pull_raw`**(*url*, *name*, *verify=False*)

Execute a `machinectl pull-raw` to download a .qcow2 or raw disk image, and add it to /var/lib/machines as a new container.

Note: Requires systemd >= 219

---

url URL from which to download the container

name Name for the new container

verify [False] Perform signature or checksum verification on the container. See the `machinectl(1)` man page (section titled "Image Transfer Commands") for more information on requirements for image verification. To perform signature verification, use `verify=signature`. For checksum verification, use `verify=checksum`. By default, no verification will be performed.

CLI Examples:

```
salt myminion nspawn.pull_raw http://ftp.halifax.rwth-aachen.de/fedora/linux/
↪releases/21/Cloud/Images/x86_64/Fedora-Cloud-Base-20141203-21.x86_64.raw.xz
↪fedora21
```

`salt.modules.nspawn.`**`pull_tar`**(*url*, *name*, *verify=False*)

Execute a `machinectl pull-raw` to download a .tar container image, and add it to /var/lib/machines as a new container.

Note: Requires systemd >= 219

---

url URL from which to download the container

name Name for the new container

verify [False] Perform signature or checksum verification on the container. See the `machinectl(1)` man page (section titled "Image Transfer Commands") for more information on requirements for image verification. To perform signature verification, use `verify=signature`. For checksum verification, use `verify=checksum`. By default, no verification will be performed.

CLI Examples:

```
salt myminion nspawn.pull_tar http://foo.domain.tld/containers/archlinux-2015.02.
↪01.tar.gz arch2
```

`salt.modules.nspawn.`**`reboot`**(*name*, *kill=False*)

Reboot the container by sending a SIGINT to its init process. Equivalent to running `machinectl reboot` on the named container.

For convenience, running `nspawn.restart` (as shown in the CLI examples below) is equivalent to running `nspawn.reboot`.

Note: `machinectl reboot` is only supported in systemd >= 219. On earlier systemd versions, running this function will instead restart the container via `systemctl`.

CLI Examples:

```
salt myminion nspawn.reboot arch1
salt myminion nspawn.restart arch1
```

`salt.modules.nspawn.`**`remove`**(*name*, *stop=False*)

Remove the named container

Warning: This function will remove all data associated with the container. It will not, however, remove the btrfs subvolumes created by pulling container images (*`nspawn.pull_raw`*, *`nspawn.pull_tar`*, *`nspawn.pull_dkr`*).

stop [False] If `True`, the container will be destroyed even if it is running/frozen.

CLI Examples:

```
salt '*' nspawn.remove foo
salt '*' nspawn.remove foo stop=True
```

`salt.modules.nspawn.`**`retcode`**(*name*, *cmd*, *no_start=False*, *preserve_state=True*, *stdin=None*, *python_shell=True*, *output_loglevel='debug'*, *use_vt=False*, *ignore_retcode=False*, *keep_env=None*)

Run *`cmd.retcode`* within a container

name Name of the container in which to run the command

cmd Command to run

no_start [False] If the container is not running, don't start it

preserve_state [True] After running the command, return the container to its previous state

stdin [None] Standard input to be used for the command

output_loglevel [debug] Level at which to log the output from the command. Set to `quiet` to suppress logging.

use_vt [False] Use SaltStack's utils.vt to stream output to console. Assumes `output=all`.

keep_env [None] If not passed, only a sane default PATH environment variable will be set. If `True`, all environment variables from the container's host will be kept. Otherwise, a comma-separated list (or Python list) of environment variable names can be passed, and those environment variables will be kept.

CLI Example:

```
salt myminion nspawn.retcode mycontainer 'ip addr show'
```

`salt.modules.nspawn.`**`run`**(*name*, *cmd*, *no_start=False*, *preserve_state=True*, *stdin=None*, *python_shell=True*, *output_loglevel='debug'*, *use_vt=False*, *ignore_retcode=False*, *keep_env=None*)

Run *`cmd.run`* within a container

name Name of the container in which to run the command

cmd Command to run

no_start [False] If the container is not running, don't start it

preserve_state [True] After running the command, return the container to its previous state

stdin [None] Standard input to be used for the command

output_loglevel [debug] Level at which to log the output from the command. Set to `quiet` to suppress logging.

use_vt [False] Use SaltStack's utils.vt to stream output to console.

keep_env [None] If not passed, only a sane default PATH environment variable will be set. If `True`, all environment variables from the container's host will be kept. Otherwise, a comma-separated list (or Python list) of environment variable names can be passed, and those environment variables will be kept.

CLI Example:

```
salt myminion nspawn.run mycontainer 'ifconfig -a'
```

`salt.modules.nspawn.`**`run_all`**(*name, cmd, no_start=False, preserve_state=True, stdin=None, python_shell=True, output_loglevel='debug', use_vt=False, ignore_retcode=False, keep_env=None*)

Run *`cmd.run_all`* within a container

Note: While the command is run within the container, it is initiated from the host. Therefore, the PID in the return dict is from the host, not from the container.

---

name Name of the container in which to run the command

cmd Command to run

no_start [False] If the container is not running, don't start it

preserve_state [True] After running the command, return the container to its previous state

stdin [None] Standard input to be used for the command

output_loglevel [debug] Level at which to log the output from the command. Set to `quiet` to suppress logging.

use_vt [False] Use SaltStack's utils.vt to stream output to console. Assumes `output=all`.

keep_env [None] If not passed, only a sane default PATH environment variable will be set. If `True`, all environment variables from the container's host will be kept. Otherwise, a comma-separated list (or Python list) of environment variable names can be passed, and those environment variables will be kept.

CLI Example:

```
salt myminion nspawn.run_all mycontainer 'ip addr show'
```

`salt.modules.nspawn.`**`run_stderr`**(*name, cmd, no_start=False, preserve_state=True, stdin=None, python_shell=True, output_loglevel='debug', use_vt=False, ignore_retcode=False, keep_env=None*)

Run *`cmd.run_stderr`* within a container

name Name of the container in which to run the command

cmd Command to run

no_start [False] If the container is not running, don't start it

preserve_state [True] After running the command, return the container to its previous state

stdin [None] Standard input to be used for the command

output_loglevel [debug] Level at which to log the output from the command. Set to `quiet` to suppress logging.

use_vt [False] Use SaltStack's utils.vt to stream output to console. Assumes `output=all`.

keep_env [None] If not passed, only a sane default PATH environment variable will be set. If `True`, all environment variables from the container's host will be kept. Otherwise, a comma-separated list (or Python list) of environment variable names can be passed, and those environment variables will be kept.

CLI Example:

```
salt myminion nspawn.run_stderr mycontainer 'ip addr show'
```

`salt.modules.nspawn.`**`run_stdout`**`(`*name, cmd, no_start=False, preserve_state=True, stdin=None, python_shell=True, output_loglevel='debug', use_vt=False, ignore_retcode=False, keep_env=None*`)`

Run *`cmd.run_stdout`* within a container

name Name of the container in which to run the command

cmd Command to run

no_start [False] If the container is not running, don't start it

preserve_state [True] After running the command, return the container to its previous state

stdin [None] Standard input to be used for the command

output_loglevel [debug] Level at which to log the output from the command. Set to `quiet` to suppress logging.

use_vt [False] Use SaltStack's utils.vt to stream output to console. Assumes `output=all`.

keep_env [None] If not passed, only a sane default PATH environment variable will be set. If `True`, all environment variables from the container's host will be kept. Otherwise, a comma-separated list (or Python list) of environment variable names can be passed, and those environment variables will be kept.

CLI Example:

```
salt myminion nspawn.run_stdout mycontainer 'ifconfig -a'
```

`salt.modules.nspawn.`**`start`**`(`*name*`)`

Start the named container

CLI Example:

```
salt myminion nspawn.start <name>
```

`salt.modules.nspawn.`**`state`**`(`*name*`)`

Return state of container (running or stopped)

CLI Example:

```
salt myminion nspawn.state <name>
```

`salt.modules.nspawn.`**`terminate`**`(`*name*`)`

Kill all processes in the container without issuing a clean shutdown. Equivalent to running `machinectl terminate` on the named container.

For convenience, running `nspawn.stop` and passing `kill=True` (as shown in the CLI examples below) is equivalent to running `nspawn.terminate`.

Note: `machinectl terminate` is only supported in systemd >= 219. On earlier systemd versions, running this function will simply issue a clean shutdown via `systemctl`.

CLI Examples:

```
salt myminion nspawn.terminate arch1
salt myminion nspawn.stop arch1 kill=True
```

21.9.291 salt.modules.nxos module

Execution module for Cisco NX OS Switches Proxy minions

New in version 2016.11.0.

For documentation on setting up the nxos proxy minion look in the documentation for *`salt.proxy.nxos`*.

`salt.modules.nxos.`**`cmd`**`(`*command*, **args*, ***kwargs*`)`

run commands from __proxy__ *salt.proxy.nxos*

command function from *salt.proxy.nxos* to run

args positional args to pass to *command* function

kwargs key word arguments to pass to *command* function

```
salt '*' nxos.cmd sendline 'show ver'
salt '*' nxos.cmd show_run
salt '*' nxos.cmd check_password username=admin password='$5$lkjsdfoi$blahblahblah
↪' encrypted=True
```

`salt.modules.nxos.`**`system_info`**`()`

Return system information for grains of the NX OS proxy minion

```
salt '*' nxos.system_info
```

21.9.292 salt.modules.nxos_api module

Execution module to manage Cisco Nexus Switches (NX-OS) over the NX-API

New in version 2019.2.0.

Execution module used to interface the interaction with a remote or local Nexus switch whether we're running in a Proxy Minion or regular Minion (or regular Minion running directly on the Nexus switch).

codeauthor Mircea Ulinic <ping@mirceaulinic.net>

maturity new

platform any

Note: To be able to use this module you need to enable to NX-API on your switch, by executing `feature nxapi` in configuration mode.

Configuration example:

```
switch# conf t
switch(config)# feature nxapi
```

To check that NX-API is properly enabled, execute `show nxapi`.

Output example:

```
switch# show nxapi
nxapi enabled
HTTPS Listen on port 443
```

Note: NX-API requires modern NXOS distributions, typically at least 7.0 depending on the hardware. Due to reliability reasons it is recommended to run the most recent version.

Check https://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus7000/sw/programmability/guide/b_Cisco_Nexus_7000_Series_NX-OS_Programmability_Guide/b_Cisco_Nexus_7000_Series_NX-OS_Programmability_Guide_chapter_0101.html for more details.

Usage

This module can equally be used via the *nxos_api* Proxy module or directly from an arbitrary (Proxy) Minion that is running on a machine having access to the network device API. Given that there are no external dependencies, this module can very well used when using the regular Salt Minion directly installed on the switch.

When running outside of the *nxos_api Proxy* (i.e., from another Proxy Minion type, or regular Minion), the NX-API connection arguments can be either specified from the CLI when executing the command, or in a configuration block under the `nxos_api` key in the configuration opts (i.e., (Proxy) Minion configuration file), or Pillar. The module supports these simultaneously. These fields are the exact same supported by the `nxos_api` Proxy Module:

transport: `https` Specifies the type of connection transport to use. Valid values for the connection are `http`, and `https`.

host: `localhost` The IP address or DNS host name of the connection device.

username: `admin` The username to pass to the device to authenticate the NX-API connection.

password The password to pass to the device to authenticate the NX-API connection.

port The TCP port of the endpoint for the NX-API connection. If this keyword is not specified, the default value is automatically determined by the transport type (`80` for `http`, or `443` for `https`).

timeout: `60` Time in seconds to wait for the device to respond. Default: 60 seconds.

verify: `True` Either a boolean, in which case it controls whether we verify the NX-API TLS certificate, or a string, in which case it must be a path to a CA bundle to use. Defaults to `True`.

When there is no certificate configuration on the device and this option is set as `True` (default), the commands will fail with the following error: `SSLError: [SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed (_ssl.c:581)`. In this case, you either need to configure a proper certificate on the device (*recommended*), or bypass the checks setting this argument as `False` with all the security risks considered.

Check https://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus3000/sw/programmability/6_x/b_Cisco_Nexus_3000_Series_NX-OS_Programmability_Guide/b_Cisco_Nexus_3000_Series_NX-OS_Programmability_Guide_chapter_01.html to see how to properly configure the certificate.

Example (when not running in a `nxos_api` Proxy Minion):

```
nxos_api:
  username: test
  password: test
```

In case the `username` and `password` are the same on any device you are targeting, the block above (besides other parameters specific to your environment you might need) should suffice to be able to execute commands from outside a `nxos_api` Proxy, e.g.:

```
salt-call --local nxos_api.show 'show lldp neighbors' raw_text
# The command above is available when running in a regular Minion where Salt is
↪installed

salt '*' nxos_api.show 'show version' raw_text=False
```

Note: Remember that the above applies only when not running in a `nxos_api` Proxy Minion. If you want to use the *nxos_api Proxy*, please follow the documentation notes for a proper setup.

`salt.modules.nxos_api.`**`config`**(*commands=None*, *config_file=None*, *template_engine='jinja'*, *context=None*, *defaults=None*, *saltenv='base'*, ***kwargs*)

Configures the Nexus switch with the specified commands.

This method is used to send configuration commands to the switch. It will take either a string or a list and prepend the necessary commands to put the session into config mode.

> **Warning:** All the commands will be applied directly into the running-config.

config_file The source file with the configuration commands to be sent to the device.

The file can also be a template that can be rendered using the template engine of choice.

This can be specified using the absolute path to the file, or using one of the following URL schemes:

- `salt://`, to fetch the file from the Salt fileserver.
- `http://` or `https://`
- `ftp://`
- `s3://`
- `swift://`

commands The commands to send to the switch in config mode. If the commands argument is a string it will be cast to a list. The list of commands will also be prepended with the necessary commands to put the session in config mode.

> **Note:** This argument is ignored when `config_file` is specified.

template_engine: `jinja` The template engine to use when rendering the source file. Default: `jinja`. To simply fetch the file without attempting to render, set this argument to `None`.

context Variables to add to the template context.

defaults Default values of the context_dict.

transport: `https` Specifies the type of connection transport to use. Valid values for the connection are `http`, and `https`.

host: `localhost` The IP address or DNS host name of the connection device.

username: `admin` The username to pass to the device to authenticate the NX-API connection.

password The password to pass to the device to authenticate the NX-API connection.

port The TCP port of the endpoint for the NX-API connection. If this keyword is not specified, the default value is automatically determined by the transport type (`80` for `http`, or `443` for `https`).

timeout: `60` Time in seconds to wait for the device to respond. Default: 60 seconds.

verify: `True` Either a boolean, in which case it controls whether we verify the NX-API TLS certificate, or a string, in which case it must be a path to a CA bundle to use. Defaults to `True`.

CLI Example:

```
salt '*' nxos_api.config commands="['spanning-tree mode mstp']"
salt '*' nxos_api.config config_file=salt://config.txt
salt '*' nxos_api.config config_file=https://bit.ly/2LGLcDy context="{'servers': [
↪'1.2.3.4']}"
```

`salt.modules.nxos_api.`**`rpc`**(*commands*, *method='cli'*, ***kwargs*)

Execute an arbitrary RPC request via the Nexus API.

commands The commands to be executed.

method: `cli` The type of the response, i.e., raw text (`cli_ascii`) or structured document (`cli`). Defaults to `cli` (structured data).

transport: `https` Specifies the type of connection transport to use. Valid values for the connection are `http`, and `https`.

host: `localhost` The IP address or DNS host name of the connection device.

username: `admin` The username to pass to the device to authenticate the NX-API connection.

password The password to pass to the device to authenticate the NX-API connection.

port The TCP port of the endpoint for the NX-API connection. If this keyword is not specified, the default value is automatically determined by the transport type (`80` for `http`, or `443` for `https`).

timeout: `60` Time in seconds to wait for the device to respond. Default: 60 seconds.

verify: `True` Either a boolean, in which case it controls whether we verify the NX-API TLS certificate, or a string, in which case it must be a path to a CA bundle to use. Defaults to `True`.

CLI Example:

```
salt-call --local nxps_api.rpc 'show version'
```

`salt.modules.nxos_api.`**`show`**`(`*commands*, *raw_text=True*, ***kwargs*`)`

Execute one or more show (non-configuration) commands.

commands The commands to be executed.

raw_text: `True` Whether to return raw text or structured data.

transport: `https` Specifies the type of connection transport to use. Valid values for the connection are `http`, and `https`.

host: `localhost` The IP address or DNS host name of the connection device.

username: `admin` The username to pass to the device to authenticate the NX-API connection.

password The password to pass to the device to authenticate the NX-API connection.

port The TCP port of the endpoint for the NX-API connection. If this keyword is not specified, the default value is automatically determined by the transport type (`80` for `http`, or `443` for `https`).

timeout: `60` Time in seconds to wait for the device to respond. Default: 60 seconds.

verify: `True` Either a boolean, in which case it controls whether we verify the NX-API TLS certificate, or a string, in which case it must be a path to a CA bundle to use. Defaults to `True`.

CLI Example:

```
salt-call --local nxos_api.show 'show version'
salt '*' nxos_api.show 'show bgp sessions' 'show processes' raw_text=False
salt 'regular-minion' nxos_api.show 'show interfaces' host=sw01.example.com
↪username=test password=test
```

21.9.293 salt.modules.omapi

This module interacts with an ISC DHCP Server via OMAPI. server_ip and server_port params may be set in the minion config or pillar:

```
omapi.server_ip: 127.0.0.1
omapi.server_port: 7991
```

depends pypureomapi Python module

`salt.modules.omapi.`**`add_host`**`(`*mac*, *name=None*, *ip=None*, *ddns=False*, *group=None*, *supersede_host=False*`)`

Add a host object for the given mac.

CLI Example:

```
salt dhcp-server omapi.add_host ab:ab:ab:ab:ab:ab name=host1
```

Add ddns-hostname and a fixed-ip statements:

```
salt dhcp-server omapi.add_host ab:ab:ab:ab:ab:ab name=host1 ip=10.1.1.1 ddns=true
```

`salt.modules.omapi.`**`delete_host`**`(`*mac=None*, *name=None*`)`

Delete the host with the given mac or name.

CLI Examples:

```
salt dhcp-server omapi.delete_host name=host1
salt dhcp-server omapi.delete_host mac=ab:ab:ab:ab:ab:ab
```

21.9.294 salt.modules.openbsd_sysctl

Module for viewing and modifying OpenBSD sysctl parameters

`salt.modules.openbsd_sysctl.`**`assign`**`(`*name*, *value*`)`

Assign a single sysctl parameter for this minion

CLI Example:

```
salt '*' sysctl.assign net.inet.ip.forwarding 1
```

`salt.modules.openbsd_sysctl.`**`get`**`(`*name*`)`

Return a single sysctl parameter for this minion

CLI Example:

```
salt '*' sysctl.get hw.physmem
```

`salt.modules.openbsd_sysctl.`**`persist`**`(`*name*, *value*, *config='/etc/sysctl.conf'*`)`

Assign and persist a simple sysctl parameter for this minion

CLI Example:

```
salt '*' sysctl.persist net.inet.ip.forwarding 1
```

`salt.modules.openbsd_sysctl.`**`show`**`(`*config_file=False*`)`

Return a list of sysctl parameters for this minion

CLI Example:

```
salt '*' sysctl.show
```

21.9.295 salt.modules.openbsdpkg

Package support for OpenBSD

Note: The package repository is configured on each host using `/etc/installurl` from OpenBSD 6.1 onwards. Earlier releases relied on `/etc/pkg.conf`.

Changed in version 2016.3.5: Package versions on OpenBSD are not normally specified explicitly; instead packages may be available in multiple *flavors*, and *branches* which are specified by the format of the package name. This module allows you to use the same formatting as `pkg_add(1)`, and will select the empty flavor and default branch by default. Examples:

```
- rsync
- vim--no_x11
- ruby%2.3
```

`salt.modules.openbsdpkg.`**`install`**`(`*name=None, pkgs=None, sources=None, **kwargs*`)`

Install the passed package

Return a dict containing the new package names and versions:

```
{'<package>': {'old': '<old-version>',
               'new': '<new-version>'}}
```

CLI Example, Install one package:

```
salt '*' pkg.install <package name>
```

CLI Example, Install more than one package:

```
salt '*' pkg.install pkgs='["<package name>", "<package name>"]'
```

CLI Example, Install more than one package from a alternate source (e.g. salt file-server, HTTP, FTP, local filesystem):

```
salt '*' pkg.install sources='[{"<pkg name>": "salt://pkgs/<pkg filename>"}]'
```

`salt.modules.openbsdpkg.`**`latest_version`**`(`**names, **kwargs*`)`

Return the latest version of the named package available for upgrade or installation. If more than one package name is specified, a dict of name/version pairs is returned.

If the latest version of a given package is already installed, an empty string will be returned for that package.

CLI Example:

```
salt '*' pkg.latest_version <package name>
```

`salt.modules.openbsdpkg.`**`list_pkgs`**`(`*versions_as_list=False, **kwargs*`)`

List the packages currently installed as a dict:

```
{'<package_name>': '<version>'}
```

CLI Example:

```
salt '*' pkg.list_pkgs
```

`salt.modules.openbsdpkg.`**`purge`**`(`*name=None, pkgs=None, **kwargs*`)`

Remove a package and extra configuration files.

name The name of the package to be deleted.

Multiple Package Options:

pkgs A list of packages to delete. Must be passed as a python list. The `name` parameter will be ignored if this option is passed.

New in version 0.16.0.

Returns a dict containing the changes.

CLI Example:

```
salt '*' pkg.purge <package name>
salt '*' pkg.purge <package1>,<package2>,<package3>
salt '*' pkg.purge pkgs='["foo", "bar"]'
```

`salt.modules.openbsdpkg.`**`remove`**`(`*name=None, pkgs=None, purge=False, **kwargs*`)`

Remove a single package with pkg_delete

Multiple Package Options:

pkgs A list of packages to delete. Must be passed as a python list. The `name` parameter will be ignored if this option is passed.

New in version 0.16.0.

Returns a dict containing the changes.

CLI Example:

```
salt '*' pkg.remove <package name>
salt '*' pkg.remove <package1>,<package2>,<package3>
salt '*' pkg.remove pkgs='["foo", "bar"]'
```

`salt.modules.openbsdpkg.`**`upgrade`**`(`*name=None*, *pkgs=None*, ***kwargs*`)`

Run a full package upgrade (`pkg_add -u`), or upgrade a specific package if `name` or `pkgs` is provided. `name` is ignored when `pkgs` is specified.

Returns a dictionary containing the changes:

New in version 2019.2.0.

```
{'<package>': {'old': '<old-version>',
               'new': '<new-version>'}}
```

CLI Example:

```
salt '*' pkg.upgrade
salt '*' pkg.upgrade python%2.7
```

`salt.modules.openbsdpkg.`**`upgrade_available`**`(`*name*`)`

Check whether or not an upgrade is available for a given package

New in version 2019.2.0.

CLI Example:

```
salt '*' pkg.upgrade_available <package name>
```

`salt.modules.openbsdpkg.`**`version`**`(`**names*, ***kwargs*`)`

Returns a string representing the package version or an empty string if not installed. If more than one package name is specified, a dict of name/version pairs is returned.

CLI Example:

```
salt '*' pkg.version <package name>
salt '*' pkg.version <package1> <package2> <package3> ...
```

21.9.296 salt.modules.openbsdrcctl_service.py

The rcctl service module for OpenBSD

`salt.modules.openbsdrcctl_service.`**`available`**`(`*name*`)`

Return True if the named service is available.

CLI Example:

```
salt '*' service.available sshd
```

`salt.modules.openbsdrcctl_service.`**`disable`**`(`*name*, ***kwargs*`)`

Disable the named service to not start at boot.

CLI Example:

```
salt '*' service.disable <service name>
```

`salt.modules.openbsdrcctl_service.`**`disabled`**`(`*name*`)`

Return True if the named service is disabled at boot, False otherwise.

CLI Example:

```
salt '*' service.disabled <service name>
```

`salt.modules.openbsdrcctl_service.`**`enable`**`(`*name*, ***kwargs*`)`

Enable the named service to start at boot.

flags [None] Set optional flags to run the service with.

service.flags can be used to change the default flags.

CLI Example:

```
salt '*' service.enable <service name>
salt '*' service.enable <service name> flags=<flags>
```

`salt.modules.openbsdrcctl_service.`**`enabled`**`(`*name*, ***kwargs*`)`

Return True if the named service is enabled at boot and the provided flags match the configured ones (if any). Return False otherwise.

name Service name

CLI Example:

```
salt '*' service.enabled <service name>
salt '*' service.enabled <service name> flags=<flags>
```

`salt.modules.openbsdrcctl_service.`**`get_all`**`()`

Return all installed services.

CLI Example:

```
salt '*' service.get_all
```

`salt.modules.openbsdrcctl_service.`**`get_disabled`**`()`

Return what services are available but not enabled to start at boot.

CLI Example:

```
salt '*' service.get_disabled
```

`salt.modules.openbsdrcctl_service.`**`get_enabled`**`()`

Return what services are set to run on boot.

CLI Example:

```
salt '*' service.get_enabled
```

`salt.modules.openbsdrcctl_service.`**`missing`**`(`*name*`)`

The inverse of service.available. Return True if the named service is not available.

CLI Example:

```
salt '*' service.missing sshd
```

`salt.modules.openbsdrcctl_service.`**`reload_`**`(`*name*`)`

Reload the named service.

CLI Example:

```
salt '*' service.reload <service name>
```

`salt.modules.openbsdrcctl_service.`**`restart`**`(`*name*`)`

Restart the named service.

CLI Example:

```
salt '*' service.restart <service name>
```

`salt.modules.openbsdrcctl_service.`**`start`**`(`*name*`)`

Start the named service.

CLI Example:

```
salt '*' service.start <service name>
```

`salt.modules.openbsdrcctl_service.`**`status`**`(`*name, sig=None*`)`

Return the status for a service, returns a bool whether the service is running.

CLI Example:

```
salt '*' service.status <service name>
```

`salt.modules.openbsdrcctl_service.`**`stop`**`(`*name*`)`

Stop the named service.

CLI Example:

```
salt '*' service.stop <service name>
```

21.9.297 salt.modules.openbsdservice

The service module for OpenBSD

Important: If you feel that Salt should be using this module to manage services on a minion, and it is using a different module (or gives an error similar to *'service.start' is not available*), see *here*.

`salt.modules.openbsdservice.`**`available`**`(`*name*`)`

New in version 2014.7.0.

Returns `True` if the specified service is available, otherwise returns `False`.

CLI Example:

```
salt '*' service.available sshd
```

`salt.modules.openbsdservice.`**`disabled`**`(`*name*`)`

New in version 2014.7.0.

Return True if the named service is disabled, false otherwise

CLI Example:

```
salt '*' service.disabled <service name>
```

salt.modules.openbsdservice.**enabled**(*name*, ***kwargs*)

New in version 2014.7.0.

Return True if the named service is enabled, false otherwise

CLI Example:

```
salt '*' service.enabled <service name>
```

salt.modules.openbsdservice.**get_all**()

New in version 2014.7.0.

Return all available boot services

CLI Example:

```
salt '*' service.get_all
```

salt.modules.openbsdservice.**get_disabled**()

New in version 2014.7.0.

Return a set of services that are installed but disabled

CLI Example:

```
salt '*' service.get_disabled
```

salt.modules.openbsdservice.**get_enabled**()

New in version 2014.7.0.

Return a list of service that are enabled on boot

CLI Example:

```
salt '*' service.get_enabled
```

salt.modules.openbsdservice.**missing**(*name*)

New in version 2014.7.0.

The inverse of service.available. Returns `True` if the specified service is not available, otherwise returns `False`.

CLI Example:

```
salt '*' service.missing sshd
```

salt.modules.openbsdservice.**reload_**(*name*)

New in version 2014.7.0.

Reload the named service

CLI Example:

```
salt '*' service.reload <service name>
```

salt.modules.openbsdservice.**restart**(*name*)

Restart the named service

CLI Example:

```
salt '*' service.restart <service name>
```

salt.modules.openbsdservice.**start**(*name*)

Start the specified service

CLI Example:

```
salt '*' service.start <service name>
```

salt.modules.openbsdservice.**status**(*name*, *sig=None*)

Return the status for a service. If the name contains globbing, a dict mapping service name to True/False values is returned.

Changed in version 2018.3.0: The service name can now be a glob (e.g. `salt*`)

Parameters

- **name** (*str*) – The name of the service to check
- **sig** (*str*) – Signature to use to find the service via ps

Returns True if running, False otherwise dict: Maps service name to True if running, False otherwise

Return type bool

CLI Example:

```
salt '*' service.status <service name> [service signature]
```

salt.modules.openbsdservice.**stop**(*name*)

Stop the specified service

CLI Example:

```
salt '*' service.stop <service name>
```

21.9.298 salt.modules.openscap module

Module for OpenSCAP Management

salt.modules.openscap.**xccdf**(*params*)

Run `oscap xccdf` commands on minions. It uses cp.push_dir to upload the generated files to the salt master in the master's minion files cachedir (defaults to `/var/cache/salt/master/minions/minion-id/files`)

It needs `file_recv` set to `True` in the master configuration file.

CLI Example:

```
salt '*'  openscap.xccdf "eval --profile Default /usr/share/openscap/scap-
↪yast2sec-xccdf.xml"
```

21.9.299 salt.modules.openstack_config

Modify, retrieve, or delete values from OpenStack configuration files.

maintainer Jeffrey C. Ollie <jeff@ocjtech.us>

maturity new

depends

platform linux

`salt.modules.openstack_config.`**`delete`**(*filename*, *section*, *parameter*)

Delete a value from an OpenStack configuration file.

filename The full path to the configuration file

section The section from which to delete the parameter

parameter The parameter to delete

CLI Example:

```
salt-call openstack_config.delete /etc/keystone/keystone.conf sql connection
```

`salt.modules.openstack_config.`**`get`**(*filename*, *section*, *parameter*)

Get a value from an OpenStack configuration file.

filename The full path to the configuration file

section The section from which to search for the parameter

parameter The parameter to return

CLI Example:

```
salt-call openstack_config.get /etc/keystone/keystone.conf sql connection
```

`salt.modules.openstack_config.`**`set_`**(*filename*, *section*, *parameter*, *value*)

Set a value in an OpenStack configuration file.

filename The full path to the configuration file

section The section in which the parameter will be set

parameter The parameter to change

value The value to set

CLI Example:

```
salt-call openstack_config.set /etc/keystone/keystone.conf sql connection foo
```

21.9.300 salt.modules.openstack_mng module

Module for OpenStack Management

codeauthor Konrad Mosoń <mosonkonrad@gmail.com>

maturity new

depends openstack-utils

platform linux

`salt.modules.openstack_mng.`**`restart_service`**(*service_name*, *minimum_running_time=None*)

Restart OpenStack service immediately, or only if it's running longer than specified value

CLI Example:

```
salt '*' openstack_mng.restart_service neutron
salt '*' openstack_mng.restart_service neutron minimum_running_time=600
```

`salt.modules.openstack_mng.`**`start_service`**(*service_name*)

Start OpenStack service immediately

CLI Example:

```
salt '*' openstack_mng.start_service neutron
```

salt.modules.openstack_mng.**stop_service**(*service_name*)

Stop OpenStack service immediately

CLI Example:

```
salt '*' openstack_mng.stop_service neutron
```

21.9.301 salt.modules.openvswitch module

Support for Open vSwitch - module with basic Open vSwitch commands.

Suitable for setting up Openstack Neutron.

codeauthor Jiri Kotlin <jiri.kotlin@ultimum.io>

salt.modules.openvswitch.**bridge_create**(*br*, *may_exist=True*)

Creates a new bridge.

Parameters

- **br** – A string - bridge name
- **may_exist** – Bool, if False - attempting to create a bridge that exists returns False.

Returns True on success, else False.

New in version 2016.3.0.

CLI Example: .. code-block:: bash

salt '*' openvswitch.bridge_create br0

salt.modules.openvswitch.**bridge_delete**(*br*, *if_exists=True*)

Deletes bridge and all of its ports.

Parameters

- **br** – A string - bridge name
- **if_exists** – Bool, if False - attempting to delete a bridge that does not exist returns False.

Returns True on success, else False.

New in version 2016.3.0.

CLI Example: .. code-block:: bash

salt '*' openvswitch.bridge_delete br0

salt.modules.openvswitch.**bridge_exists**(*br*)

Tests whether bridge exists as a real or fake bridge.

Returns True if Bridge exists, else False.

New in version 2016.3.0.

CLI Example: .. code-block:: bash

salt '*' openvswitch.bridge_exists br0

salt.modules.openvswitch.**bridge_list**()

Lists all existing real and fake bridges.

Returns List of bridges (or empty list), False on failure.

New in version 2016.3.0.

CLI Example: .. code-block:: bash

salt '*' openvswitch.bridge_list

salt.modules.openvswitch.**interface_get_options**(*port*)

Port's interface's optional parameters.

Parameters **port** – A string - port name.

Returns String containing optional parameters of port's interface, False on failure.

New in version 2016.3.0.

CLI Example: .. code-block:: bash
salt '*' openvswitch.interface_get_options tap0

`salt.modules.openvswitch.`**`interface_get_type`**(*port*)

Type of port's interface.

Parameters **`port`** – A string - port name.

Returns String - type of interface or empty string, False on failure.

New in version 2016.3.0.

CLI Example: .. code-block:: bash
salt '*' openvswitch.interface_get_type tap0

`salt.modules.openvswitch.`**`port_add`**(*br*, *port*, *may_exist=False*, *internal=False*)

Creates on bridge a new port named port.

Returns True on success, else False.

Parameters

- **`br`** – A string - bridge name
- **`port`** – A string - port name
- **`may_exist`** – Bool, if False - attempting to create a port that exists returns False.
- **`internal`** – A boolean to create an internal interface if one does not exist.

New in version 2016.3.0.

CLI Example: .. code-block:: bash
salt '*' openvswitch.port_add br0 8080

`salt.modules.openvswitch.`**`port_create_gre`**(*br*, *port*, *id*, *remote*)

Generic Routing Encapsulation - creates GRE tunnel between endpoints.

Parameters

- **`br`** – A string - bridge name.
- **`port`** – A string - port name.
- **`id`** – An integer - unsigned 32-bit number, tunnel's key.
- **`remote`** – A string - remote endpoint's IP address.

Returns True on success, else False.

New in version 2016.3.0.

CLI Example: .. code-block:: bash
salt '*' openvswitch.port_create_gre br0 gre1 5001 192.168.1.10

`salt.modules.openvswitch.`**`port_create_vlan`**(*br*, *port*, *id*, *internal=False*)

Isolate VM traffic using VLANs.

Parameters

- **`br`** – A string - bridge name.
- **`port`** – A string - port name.
- **`id`** – An integer in the valid range 0 to 4095 (inclusive), name of VLAN.
- **`internal`** – A boolean to create an internal interface if one does not exist.

Returns True on success, else False.

New in version 2016.3.0.

CLI Example: .. code-block:: bash
salt '*' openvswitch.port_create_vlan br0 tap0 100

`salt.modules.openvswitch.`**`port_create_vxlan`**(*br*, *port*, *id*, *remote*, *dst_port=None*)

Virtual eXtensible Local Area Network - creates VXLAN tunnel between endpoints.

Parameters

- **`br`** – A string - bridge name.
- **`port`** – A string - port name.
- **`id`** – An integer - unsigned 64-bit number, tunnel's key.

- **remote** – A string - remote endpoint's IP address.
- **dst_port** – An integer - port to use when creating tunnelport in the switch.

Returns True on success, else False.

New in version 2016.3.0.

CLI Example: .. code-block:: bash
salt '*' openvswitch.port_create_vxlan br0 vx1 5001 192.168.1.10 8472

`salt.modules.openvswitch.`**`port_get_tag`**(*port*)

Lists tags of the port.

Parameters **port** – A string - port name.

Returns List of tags (or empty list), False on failure.

New in version 2016.3.0.

CLI Example: .. code-block:: bash
salt '*' openvswitch.port_get_tag tap0

`salt.modules.openvswitch.`**`port_list`**(*br*)

Lists all of the ports within bridge.

Parameters **br** – A string - bridge name.

Returns List of bridges (or empty list), False on failure.

New in version 2016.3.0.

CLI Example: .. code-block:: bash
salt '*' openvswitch.port_list br0

`salt.modules.openvswitch.`**`port_remove`**(*br*, *port*, *if_exists=True*)

Deletes port.

Parameters

- **br** – A string - bridge name (If bridge is None, port is removed from whatever bridge contains it)
- **port** – A string - port name.
- **if_exists** – Bool, if False - attempting to delete a por that does not exist returns False. (Default True)

Returns True on success, else False.

New in version 2016.3.0.

CLI Example: .. code-block:: bash
salt '*' openvswitch.port_remove br0 8080

21.9.302 salt.modules.opkg module

Support for Opkg

Important: If you feel that Salt should be using this module to manage packages on a minion, and it is using a different module (or gives an error similar to *'pkg.install' is not available*), see *here*.

Note: For version comparison support on opkg < 0.3.4, the `opkg-utils` package must be installed.

`salt.modules.opkg.`**`available_version`**(**names*, ***kwargs*)

Return the latest version of the named package available for upgrade or installation. If more than one package name is specified, a dict of name/version pairs is returned.

If the latest version of a given package is already installed, an empty string will be returned for that package.

CLI Example:

```
salt '*' pkg.latest_version <package name>
salt '*' pkg.latest_version <package name>
salt '*' pkg.latest_version <package1> <package2> <package3> ...
```

`salt.modules.opkg.`**`check_extra_requirements`**`(`*pkgname*, *pkgver*`)`
Check if the installed package already has the given requirements. There's nothing do to here for nipkg.py, therefore it will always return True.

`salt.modules.opkg.`**`del_repo`**`(`*repo*, ***kwargs*`)`
Delete a repo from `/etc/opkg/*.conf`

If the file does not contain any other repo configuration, the file itself will be deleted.

CLI Examples:

```
salt '*' pkg.del_repo repo
```

`salt.modules.opkg.`**`file_dict`**`(`**packages*, ***kwargs*`)`
List the files that belong to a package, grouped by package. Not specifying any packages will return a list of _every_ file on the system's package database (not generally recommended).

CLI Examples:

```
salt '*' pkg.file_list httpd
salt '*' pkg.file_list httpd postfix
salt '*' pkg.file_list
```

`salt.modules.opkg.`**`file_list`**`(`**packages*, ***kwargs*`)`
List the files that belong to a package. Not specifying any packages will return a list of _every_ file on the system's package database (not generally recommended).

CLI Examples:

```
salt '*' pkg.file_list httpd
salt '*' pkg.file_list httpd postfix
salt '*' pkg.file_list
```

`salt.modules.opkg.`**`get_repo`**`(`*repo*, ***kwargs*`)`
Display a repo from the `/etc/opkg/*.conf`

CLI Examples:

```
salt '*' pkg.get_repo repo
```

`salt.modules.opkg.`**`hold`**`(`*name=None*, *pkgs=None*, *sources=None*, ***kwargs*`)`
Set package in 'hold' state, meaning it will not be upgraded.

name The name of the package, e.g., 'tmux'

CLI Example:

```
salt '*' pkg.hold <package name>
```

pkgs A list of packages to hold. Must be passed as a python list.

CLI Example:

```
salt '*' pkg.hold pkgs='["foo", "bar"]'
```

`salt.modules.opkg.`**`info_installed`**`(`**names, **kwargs*`)`

Return the information of the named package(s), installed on the system.

New in version 2017.7.0.

Parameters

- **`names`** – Names of the packages to get information about. If none are specified, will return information for all installed packages.
- **`attr`** – Comma-separated package attributes. If no 'attr' is specified, all available attributes returned.

  Valid attributes are: arch, conffiles, conflicts, depends, description, filename, group, install_date_time_t, md5sum, packager, provides, recommends, replaces, size, source, suggests, url, version

CLI example:

```
salt '*' pkg.info_installed
salt '*' pkg.info_installed attr=version,packager
salt '*' pkg.info_installed <package1>
salt '*' pkg.info_installed <package1> <package2> <package3> ...
salt '*' pkg.info_installed <package1> attr=version,packager
salt '*' pkg.info_installed <package1> <package2> <package3> ... attr=version,
↪packager
```

`salt.modules.opkg.`**`install`**`(`*name=None, refresh=False, pkgs=None, sources=None, reinstall=False, **kwargs*`)`

Install the passed package, add refresh=True to update the opkg database.

name The name of the package to be installed. Note that this parameter is ignored if either "pkgs" or "sources" is passed. Additionally, please note that this option can only be used to install packages from a software repository. To install a package file manually, use the "sources" option.

CLI Example:

```
salt '*' pkg.install <package name>
```

refresh Whether or not to refresh the package database before installing.

version Install a specific version of the package, e.g. 1.2.3~0ubuntu0. Ignored if "pkgs" or "sources" is passed.

New in version 2017.7.0.

reinstall [False] Specifying reinstall=True will use `opkg install --force-reinstall` rather than simply `opkg install` for requested packages that are already installed.

If a version is specified with the requested package, then `opkg install --force-reinstall` will only be used if the installed version matches the requested version.

New in version 2017.7.0.

Multiple Package Installation Options:

pkgs A list of packages to install from a software repository. Must be passed as a python list.

CLI Example:

```
salt '*' pkg.install pkgs='["foo", "bar"]'
salt '*' pkg.install pkgs='["foo", {"bar": "1.2.3-0ubuntu0"}]'
```

sources A list of IPK packages to install. Must be passed as a list of dicts, with the keys being package names, and the values being the source URI or local path to the package. Dependencies are automatically resolved and marked as auto-installed.

CLI Example:

```
salt '*' pkg.install sources='[{"foo": "salt://foo.deb"},{"bar": "salt://bar.
↪deb"}]'
```

install_recommends Whether to install the packages marked as recommended. Default is True.

only_upgrade Only upgrade the packages (disallow downgrades), if they are already installed. Default is False.

New in version 2017.7.0.

Returns a dict containing the new package names and versions:

```
{'<package>': {'old': '<old-version>',
               'new': '<new-version>'}}
```

`salt.modules.opkg.`**`latest_version`**(**names*, ***kwargs*)

Return the latest version of the named package available for upgrade or installation. If more than one package name is specified, a dict of name/version pairs is returned.

If the latest version of a given package is already installed, an empty string will be returned for that package.

CLI Example:

```
salt '*' pkg.latest_version <package name>
salt '*' pkg.latest_version <package name>
salt '*' pkg.latest_version <package1> <package2> <package3> ...
```

`salt.modules.opkg.`**`list_pkgs`**(*versions_as_list=False*, ***kwargs*)

List the packages currently installed in a dict:

```
{'<package_name>': '<version>'}
```

CLI Example:

```
salt '*' pkg.list_pkgs
salt '*' pkg.list_pkgs versions_as_list=True
```

`salt.modules.opkg.`**`list_repos`**(***kwargs*)

Lists all repos on `/etc/opkg/*.conf`

CLI Example:

```
salt '*' pkg.list_repos
```

`salt.modules.opkg.`**`list_upgrades`**(*refresh=True*, ***kwargs*)

List all available package upgrades.

CLI Example:

```
salt '*' pkg.list_upgrades
```

`salt.modules.opkg.`**`mod_repo`**(*repo*, ***kwargs*)

Modify one or more values for a repo. If the repo does not exist, it will be created, so long as uri is defined.

The following options are available to modify a repo definition:

repo alias by which opkg refers to the repo.

uri the URI to the repo.

compressed defines (True or False) if the index file is compressed

enabled enable or disable (True or False) repository but do not remove if disabled.

refresh enable or disable (True or False) auto-refresh of the repositories

CLI Examples:

```
salt '*' pkg.mod_repo repo uri=http://new/uri
salt '*' pkg.mod_repo repo enabled=False
```

`salt.modules.opkg.`**`owner`**`(`**paths*, ***kwargs*`)`

Return the name of the package that owns the file. Multiple file paths can be passed. Like `pkg.version <salt.modules.opkg.version`, if a single path is passed, a string will be returned, and if multiple paths are passed, a dictionary of file/package name pairs will be returned.

If the file is not owned by a package, or is not present on the minion, then an empty string will be returned for that path.

CLI Example:

salt '*' pkg.owner /usr/bin/apachectl salt '*' pkg.owner /usr/bin/apachectl /usr/bin/basename

`salt.modules.opkg.`**`purge`**`(`*name=None*, *pkgs=None*, ***kwargs*`)`

Package purges are not supported by opkg, this function is identical to *`pkg.remove`*.

name The name of the package to be deleted.

Multiple Package Options:

pkgs A list of packages to delete. Must be passed as a python list. The `name` parameter will be ignored if this option is passed.

Returns a dict containing the changes.

CLI Example:

```
salt '*' pkg.purge <package name>
salt '*' pkg.purge <package1>,<package2>,<package3>
salt '*' pkg.purge pkgs='["foo", "bar"]'
```

`salt.modules.opkg.`**`refresh_db`**`(`*failhard=False*, ***kwargs*`)`

Updates the opkg database to latest packages based upon repositories

Returns a dict, with the keys being package databases and the values being the result of the update attempt. Values can be one of the following:

- `True`: Database updated successfully
- `False`: Problem updating database

failhard If False, return results of failed lines as `False` for the package database that encountered the error. If True, raise an error with a list of the package databases that encountered errors.

New in version 2018.3.0.

CLI Example:

```
salt '*' pkg.refresh_db
```

`salt.modules.opkg.`**`remove`**`(`*name=None*, *pkgs=None*, ***kwargs*`)`

Remove packages using `opkg remove`.

name The name of the package to be deleted.

Multiple Package Options:

pkgs A list of packages to delete. Must be passed as a python list. The `name` parameter will be ignored if this option is passed.

remove_dependencies Remove package and all dependencies

New in version 2019.2.0.

auto_remove_deps Remove packages that were installed automatically to satisfy dependencies

New in version 2019.2.0.

Returns a dict containing the changes.

CLI Example:

```
salt '*' pkg.remove <package name>
salt '*' pkg.remove <package1>,<package2>,<package3>
```

(continues on next page)

(continued from previous page)

```
salt '*' pkg.remove pkgs='["foo", "bar"]'
salt '*' pkg.remove pkgs='["foo", "bar"]' remove_dependencies=True auto_remove_
↪deps=True
```

`salt.modules.opkg.`**`unhold`**(*name=None*, *pkgs=None*, *sources=None*, ***kwargs*)

Set package current in 'hold' state to install state, meaning it will be upgraded.

name The name of the package, e.g., 'tmux'

CLI Example:

```
salt '*' pkg.unhold <package name>
```

pkgs A list of packages to hold. Must be passed as a python list.

CLI Example:

```
salt '*' pkg.unhold pkgs='["foo", "bar"]'
```

`salt.modules.opkg.`**`upgrade`**(*refresh=True*, ***kwargs*)

Upgrades all packages via `opkg upgrade`

Returns a dictionary containing the changes:

```
{'<package>':  {'old': '<old-version>',
                'new': '<new-version>'}}
```

CLI Example:

```
salt '*' pkg.upgrade
```

`salt.modules.opkg.`**`upgrade_available`**(*name*, ***kwargs*)

Check whether or not an upgrade is available for a given package

CLI Example:

```
salt '*' pkg.upgrade_available <package name>
```

`salt.modules.opkg.`**`version`**(**names*, ***kwargs*)

Returns a string representing the package version or an empty string if not installed. If more than one package name is specified, a dict of name/version pairs is returned.

CLI Example:

```
salt '*' pkg.version <package name>
salt '*' pkg.version <package1> <package2> <package3> ...
```

`salt.modules.opkg.`**`version_clean`**(*version*)

Clean the version string removing extra data. There's nothing do to here for nipkg.py, therefore it will always return the given version.

`salt.modules.opkg.`**`version_cmp`**(*pkg1*, *pkg2*, *ignore_epoch=False*, ***kwargs*)

Do a cmp-style comparison on two packages. Return -1 if pkg1 < pkg2, 0 if pkg1 == pkg2, and 1 if pkg1 > pkg2. Return None if there was a problem making the comparison.

ignore_epoch [False] Set to `True` to ignore the epoch when comparing versions

New in version 2016.3.4.

CLI Example:

```
salt '*' pkg.version_cmp '0.2.4-0' '0.2.4.1-0'
```

21.9.303 salt.modules.opsgenie

Module for sending data to OpsGenie

New in version 2018.3.0.

configuration This module can be used in Reactor System for posting data to OpsGenie as a remote-execution function.

For example:

```
opsgenie_event_poster:
  local.opsgenie.post_data:
    - tgt: 'salt-minion'
    - kwarg:
        name: event.reactor
        api_key: XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX
        reason: {{ data['data']['reason'] }}
        action_type: Create
```

`salt.modules.opsgenie.`**`post_data`**`(`*api_key=None, name='OpsGenie Execution Module', reason=None, action_type=None*`)`

Post data to OpsGenie. It's designed for Salt's Event Reactor.

After configuring the sls reaction file as shown above, you can trigger the module with your designated tag (og-tag in this case).

CLI Example:

```
salt-call event.send 'og-tag' '{"reason" : "Overheating CPU!"}'
```

Required parameters:
api_key It's the API Key you've copied while adding integration in OpsGenie.
reason It will be used as alert's default message in OpsGenie.
action_type OpsGenie supports the default values Create/Close for action_type. You can customize this field with OpsGenie's custom actions for other purposes like adding notes or acknowledging alerts.
Optional parameters:
name It will be used as alert's alias. If you want to use the close functionality you must provide name field for both states like in this case.

21.9.304 salt.modules.oracle

Oracle DataBase connection module

maintainer Vladimir Bormotov <bormotov@gmail.com>

maturity new

depends cx_Oracle

platform all

configuration module provide connections for multiple Oracle DB instances.

OS Environment

```
ORACLE_HOME: path to oracle product
PATH: path to Oracle Client libs need to be in PATH
```

pillar

```
oracle:
  dbs:
    <db>:
      uri: connection credentials in format:
    user/password@host[:port]/sid[ servicename as {sysdba|sysoper}]
      optional keyword servicename will determine whether it is a sid or
↪service_name
    <db>:
      uri: .....
```

`salt.modules.oracle.`**`client_version`**`()`

Oracle Client Version

CLI Example:

```
salt '*' oracle.client_version
```

`salt.modules.oracle.`**`run_query`**`(`*db*, *query*`)`

Run SQL query and return result

CLI Example:

```
salt '*' oracle.run_query my_db "select * from my_table"
```

`salt.modules.oracle.`**`show_dbs`**`(`**dbs*`)`

Show databases configuration from pillar. Filter by **args*

CLI Example:

```
salt '*' oracle.show_dbs
salt '*' oracle.show_dbs my_db
```

`salt.modules.oracle.`**`show_env`**`()`

Show Environment used by Oracle Client

CLI Example:

```
salt '*' oracle.show_env
```

Note: at first _connect() `NLS_LANG` will forced to '.AL32UTF8'

`salt.modules.oracle.`**`show_pillar`**`(`*item=None*`)`

Show Pillar segment oracle.* and subitem with notation "item:subitem"

CLI Example:

```
salt '*' oracle.show_pillar
salt '*' oracle.show_pillar dbs:my_db
```

`salt.modules.oracle.`**`version`**`(`**dbs*`)`

Server Version (select banner from v$version)

CLI Example:

```
salt '*' oracle.version
salt '*' oracle.version my_db
```

21.9.305 salt.modules.osquery

Support for OSQuery - https://osquery.io.

New in version 2015.8.0.

salt.modules.osquery.**acpi_tables**(*attrs=None*, *where=None*)
Return acpi_tables information from osquery

CLI Example:

```
salt '*' osquery.acpi_tables
```

salt.modules.osquery.**alf**(*attrs=None*, *where=None*)
Return alf information from osquery

CLI Example:

```
salt '*' osquery.alf
```

salt.modules.osquery.**alf_exceptions**(*attrs=None*, *where=None*)
Return alf_exceptions information from osquery

CLI Example:

```
salt '*' osquery.alf_exceptions
```

salt.modules.osquery.**alf_explicit_auths**(*attrs=None*, *where=None*)
Return alf_explicit_auths information from osquery

CLI Example:

```
salt '*' osquery.alf_explicit_auths
```

salt.modules.osquery.**alf_services**(*attrs=None*, *where=None*)
Return alf_services information from osquery

CLI Example:

```
salt '*' osquery.alf_services
```

salt.modules.osquery.**apps**(*attrs=None*, *where=None*)
Return apps information from osquery

CLI Example:

```
salt '*' osquery.apps
```

salt.modules.osquery.**apt_sources**(*attrs=None*, *where=None*)
Return apt_sources information from osquery

CLI Example:

```
salt '*' osquery.apt_sources
```

`salt.modules.osquery.`**`arp_cache`**(*attrs=None*, *where=None*)

Return arp_cache information from osquery

CLI Example:

```
salt '*' osquery.arp_cache
```

`salt.modules.osquery.`**`block_devices`**(*attrs=None*, *where=None*)

Return block_devices information from osquery

CLI Example:

```
salt '*' osquery.block_devices
```

`salt.modules.osquery.`**`certificates`**(*attrs=None*, *where=None*)

Return certificates information from osquery

CLI Example:

```
salt '*' osquery.certificates
```

`salt.modules.osquery.`**`chrome_extensions`**(*attrs=None*, *where=None*)

Return chrome_extensions information from osquery

CLI Example:

```
salt '*' osquery.chrome_extensions
```

`salt.modules.osquery.`**`cpuid`**(*attrs=None*, *where=None*)

Return cpuid information from osquery

CLI Example:

```
salt '*' osquery.cpuid
```

`salt.modules.osquery.`**`crontab`**(*attrs=None*, *where=None*)

Return crontab information from osquery

CLI Example:

```
salt '*' osquery.crontab
```

`salt.modules.osquery.`**`deb_packages`**(*attrs=None*, *where=None*)

Return deb_packages information from osquery

CLI Example:

```
salt '*' osquery.deb_packages
```

`salt.modules.osquery.`**`etc_hosts`**(*attrs=None*, *where=None*)

Return etc_hosts information from osquery

CLI Example:

```
salt '*' osquery.etc_hosts
```

`salt.modules.osquery.`**`etc_services`**(*attrs=None*, *where=None*)

Return etc_services information from osquery

CLI Example:

```
salt '*' osquery.etc_services
```

`salt.modules.osquery.`**`file_`**(*attrs=None*, *where=None*)
Return file information from osquery

CLI Example:

```
salt '*' osquery.file
```

`salt.modules.osquery.`**`file_changes`**(*attrs=None*, *where=None*)
Return file_changes information from osquery

CLI Example:

```
salt '*' osquery.file_changes
```

`salt.modules.osquery.`**`firefox_addons`**(*attrs=None*, *where=None*)
Return firefox_addons information from osquery

CLI Example:

```
salt '*' osquery.firefox_addons
```

`salt.modules.osquery.`**`groups`**(*attrs=None*, *where=None*)
Return groups information from osquery

CLI Example:

```
salt '*' osquery.groups
```

`salt.modules.osquery.`**`hardware_events`**(*attrs=None*, *where=None*)
Return hardware_events information from osquery

CLI Example:

```
salt '*' osquery.hardware_events
```

`salt.modules.osquery.`**`hash_`**(*attrs=None*, *where=None*)
Return hash information from osquery

CLI Example:

```
salt '*' osquery.hash
```

`salt.modules.osquery.`**`homebrew_packages`**(*attrs=None*, *where=None*)
Return homebrew_packages information from osquery

CLI Example:

```
salt '*' osquery.homebrew_packages
```

`salt.modules.osquery.`**`interface_addresses`**(*attrs=None*, *where=None*)
Return interface_addresses information from osquery

CLI Example:

```
salt '*' osquery.interface_addresses
```

`salt.modules.osquery.`**`interface_details`**`(`*attrs=None*, *where=None*`)`

Return interface_details information from osquery

CLI Example:

```
salt '*' osquery.interface_details
```

`salt.modules.osquery.`**`iokit_devicetree`**`(`*attrs=None*, *where=None*`)`

Return iokit_devicetree information from osquery

CLI Example:

```
salt '*' osquery.iokit_devicetree
```

`salt.modules.osquery.`**`iokit_registry`**`(`*attrs=None*, *where=None*`)`

Return iokit_registry information from osquery

CLI Example:

```
salt '*' osquery.iokit_registry
```

`salt.modules.osquery.`**`kernel_extensions`**`(`*attrs=None*, *where=None*`)`

Return kernel_extensions information from osquery

CLI Example:

```
salt '*' osquery.kernel_extensions
```

`salt.modules.osquery.`**`kernel_info`**`(`*attrs=None*, *where=None*`)`

Return kernel_info information from osquery

CLI Example:

```
salt '*' osquery.kernel_info
```

`salt.modules.osquery.`**`kernel_integrity`**`(`*attrs=None*, *where=None*`)`

Return kernel_integrity information from osquery

CLI Example:

```
salt '*' osquery.kernel_integrity
```

`salt.modules.osquery.`**`kernel_modules`**`(`*attrs=None*, *where=None*`)`

Return kernel_modules information from osquery

CLI Example:

```
salt '*' osquery.kernel_modules
```

`salt.modules.osquery.`**`keychain_items`**`(`*attrs=None*, *where=None*`)`

Return keychain_items information from osquery

CLI Example:

```
salt '*' osquery.keychain_items
```

`salt.modules.osquery.`**`last`**`(`*attrs=None*, *where=None*`)`

Return last information from osquery

CLI Example:

```
salt '*' osquery.last
```

`salt.modules.osquery.`**`launchd`**(*attrs=None*, *where=None*)
Return launchd information from osquery

CLI Example:

```
salt '*' osquery.launchd
```

`salt.modules.osquery.`**`listening_ports`**(*attrs=None*, *where=None*)
Return listening_ports information from osquery

CLI Example:

```
salt '*' osquery.listening_ports
```

`salt.modules.osquery.`**`logged_in_users`**(*attrs=None*, *where=None*)
Return logged_in_users information from osquery

CLI Example:

```
salt '*' osquery.logged_in_users
```

`salt.modules.osquery.`**`memory_map`**(*attrs=None*, *where=None*)
Return memory_map information from osquery

CLI Example:

```
salt '*' osquery.memory_map
```

`salt.modules.osquery.`**`mounts`**(*attrs=None*, *where=None*)
Return mounts information from osquery

CLI Example:

```
salt '*' osquery.mounts
```

`salt.modules.osquery.`**`nfs_shares`**(*attrs=None*, *where=None*)
Return nfs_shares information from osquery

CLI Example:

```
salt '*' osquery.nfs_shares
```

`salt.modules.osquery.`**`nvram`**(*attrs=None*, *where=None*)
Return nvram information from osquery

CLI Example:

```
salt '*' osquery.nvram
```

`salt.modules.osquery.`**`os_version`**(*attrs=None*, *where=None*)
Return os_version information from osquery

CLI Example:

```
salt '*' osquery.os_version
```

salt.modules.osquery.**osquery_extensions**(*attrs=None*, *where=None*)

Return osquery_extensions information from osquery

CLI Example:

```
salt '*' osquery.osquery_extensions
```

salt.modules.osquery.**osquery_flags**(*attrs=None*, *where=None*)

Return osquery_flags information from osquery

CLI Example:

```
salt '*' osquery.osquery_flags
```

salt.modules.osquery.**osquery_info**(*attrs=None*, *where=None*)

Return osquery_info information from osquery

CLI Example:

```
salt '*' osquery.osquery_info
```

salt.modules.osquery.**osquery_registry**(*attrs=None*, *where=None*)

Return osquery_registry information from osquery

CLI Example:

```
salt '*' osquery.osquery_registry
```

salt.modules.osquery.**passwd_changes**(*attrs=None*, *where=None*)

Return passwd_changes information from osquery

CLI Example:

```
salt '*' osquery.passwd_changes
```

salt.modules.osquery.**pci_devices**(*attrs=None*, *where=None*)

Return pci_devices information from osquery

CLI Example:

```
salt '*' osquery.pci_devices
```

salt.modules.osquery.**preferences**(*attrs=None*, *where=None*)

Return preferences information from osquery

CLI Example:

```
salt '*' osquery.preferences
```

salt.modules.osquery.**process_envs**(*attrs=None*, *where=None*)

Return process_envs information from osquery

CLI Example:

```
salt '*' osquery.process_envs
```

salt.modules.osquery.**process_memory_map**(*attrs=None*, *where=None*)

Return process_memory_map information from osquery

CLI Example:

```
salt '*' osquery.process_memory_map
```

salt.modules.osquery.**process_open_files**(*attrs=None*, *where=None*)
Return process_open_files information from osquery

CLI Example:

```
salt '*' osquery.process_open_files
```

salt.modules.osquery.**process_open_sockets**(*attrs=None*, *where=None*)
Return process_open_sockets information from osquery

CLI Example:

```
salt '*' osquery.process_open_sockets
```

salt.modules.osquery.**processes**(*attrs=None*, *where=None*)
Return processes information from osquery

CLI Example:

```
salt '*' osquery.processes
```

salt.modules.osquery.**quarantine**(*attrs=None*, *where=None*)
Return quarantine information from osquery

CLI Example:

```
salt '*' osquery.quarantine
```

salt.modules.osquery.**query**(*sql=None*)
Return time information from osquery

CLI Example:

```
salt '*' osquery.query "select * from users;"
```

salt.modules.osquery.**routes**(*attrs=None*, *where=None*)
Return routes information from osquery

CLI Example:

```
salt '*' osquery.routes
```

salt.modules.osquery.**rpm_packages**(*attrs=None*, *where=None*)
Return cpuid information from osquery

CLI Example:

```
salt '*' osquery.rpm_packages
```

salt.modules.osquery.**safari_extensions**(*attrs=None*, *where=None*)
Return safari_extensions information from osquery

CLI Example:

```
salt '*' osquery.safari_extensions
```

`salt.modules.osquery.`**`shared_memory`**`(`*attrs=None*, *where=None*`)`

Return shared_memory information from osquery

CLI Example:

```
salt '*' osquery.shared_memory
```

`salt.modules.osquery.`**`shell_history`**`(`*attrs=None*, *where=None*`)`

Return shell_history information from osquery

CLI Example:

```
salt '*' osquery.shell_history
```

`salt.modules.osquery.`**`smbios_tables`**`(`*attrs=None*, *where=None*`)`

Return smbios_tables information from osquery

CLI Example:

```
salt '*' osquery.smbios_tables
```

`salt.modules.osquery.`**`startup_items`**`(`*attrs=None*, *where=None*`)`

Return startup_items information from osquery

CLI Example:

```
salt '*' osquery.startup_items
```

`salt.modules.osquery.`**`suid_bin`**`(`*attrs=None*, *where=None*`)`

Return suid_bin information from osquery

CLI Example:

```
salt '*' osquery.suid_bin
```

`salt.modules.osquery.`**`system_controls`**`(`*attrs=None*, *where=None*`)`

Return system_controls information from osquery

CLI Example:

```
salt '*' osquery.system_controls
```

`salt.modules.osquery.`**`time_`**`(`*attrs=None*`)`

Return time information from osquery

CLI Example:

```
salt '*' osquery.time
```

`salt.modules.osquery.`**`usb_devices`**`(`*attrs=None*, *where=None*`)`

Return usb_devices information from osquery

CLI Example:

```
salt '*' osquery.usb_devices
```

`salt.modules.osquery.`**`users`**`(`*attrs=None*, *where=None*`)`

Return users information from osquery

CLI Example:

```
salt '*' osquery.users
```

`salt.modules.osquery.`**`version`**`()`
Return version of osquery

CLI Example:

```
salt '*' osquery.version
```

`salt.modules.osquery.`**`xattr_where_from`**`(`*attrs=None, where=None*`)`
Return xattr_where_from information from osquery

CLI Example:

```
salt '*' osquery.xattr_where_from
```

`salt.modules.osquery.`**`xprotect_entries`**`(`*attrs=None, where=None*`)`
Return xprotect_entries information from osquery

CLI Example:

```
salt '*' osquery.xprotect_entries
```

`salt.modules.osquery.`**`xprotect_reports`**`(`*attrs=None, where=None*`)`
Return xprotect_reports information from osquery

CLI Example:

```
salt '*' osquery.xprotect_reports
```

21.9.306 salt.modules.out module

Output Module

New in version 2018.3.0.

Execution module that processes JSON serializable data and returns string having the format as processed by the outputters.

Although this does not bring much value on the CLI, it turns very handy in applications that require human readable data rather than Python objects.

For example, inside a Jinja template:

```
{{ salt.out.string_format(complex_object, out='highstate') }}
```

`salt.modules.out.`**`html_format`**`(`*data, out='nested', opts=None, **kwargs*`)`
Return the formatted string as HTML.

data The JSON serializable object.

out: **`nested`** The name of the output to use to transform the data. Default: `nested`.

opts Dictionary of configuration options. Default: `__opts__`.

kwargs Arguments to sent to the outputter module.

CLI Example:

```
salt '*' out.html_format "{'key': 'value'}" out=yaml
```

`salt.modules.out.`**`out_format`**(*data*, *out='nested'*, *opts=None*, ***kwargs*)

Return the formatted outputter string for the Python object.

data The JSON serializable object.

out: `nested` The name of the output to use to transform the data. Default: `nested`.

opts Dictionary of configuration options. Default: `__opts__`.

kwargs Arguments to sent to the outputter module.

CLI Example:

```
salt '*' out.out_format "{'key': 'value'}"
```

`salt.modules.out.`**`string_format`**(*data*, *out='nested'*, *opts=None*, ***kwargs*)

Return the outputter formatted string, removing the ANSI escape sequences.

data The JSON serializable object.

out: `nested` The name of the output to use to transform the data. Default: `nested`.

opts Dictionary of configuration options. Default: `__opts__`.

kwargs Arguments to sent to the outputter module.

CLI Example:

```
salt '*' out.string_format "{'key': 'value'}" out=table
```

21.9.307 salt.modules.pacmanpkg

A module to wrap pacman calls, since Arch is the best (https://wiki.archlinux.org/index.php/Arch_is_the_best)

Important: If you feel that Salt should be using this module to manage packages on a minion, and it is using a different module (or gives an error similar to *'pkg.install' is not available*), see *here*.

`salt.modules.pacmanpkg.`**`file_dict`**(**packages*)

List the files that belong to a package, grouped by package. Not specifying any packages will return a list of _every_ file on the system's package database (not generally recommended).

CLI Examples:

```
salt '*' pkg.file_list httpd
salt '*' pkg.file_list httpd postfix
salt '*' pkg.file_list
```

`salt.modules.pacmanpkg.`**`file_list`**(**packages*)

List the files that belong to a package. Not specifying any packages will return a list of _every_ file on the system's package database (not generally recommended).

CLI Examples:

```
salt '*' pkg.file_list httpd
salt '*' pkg.file_list httpd postfix
salt '*' pkg.file_list
```

`salt.modules.pacmanpkg.`**`group_diff`**(*name*)

New in version 2016.11.0.

Lists which of a group's packages are installed and which are not installed

Compatible with yumpkg.group_diff for easy support of state.pkg.group_installed

CLI Example:

```
salt '*' pkg.group_diff 'xorg'
```

`salt.modules.pacmanpkg.`**`group_info`**`(`*name*`)`

New in version 2016.11.0.

Lists all packages in the specified group

CLI Example:

```
salt '*' pkg.group_info 'xorg'
```

`salt.modules.pacmanpkg.`**`group_list`**`()`

New in version 2016.11.0.

Lists all groups known by pacman on this system

CLI Example:

```
salt '*' pkg.group_list
```

`salt.modules.pacmanpkg.`**`install`**`(`*name=None, refresh=False, sysupgrade=None, pkgs=None, sources=None, **kwargs*`)`

Changed in version 2015.8.12,2016.3.3,2016.11.0: On minions running systemd>=205, systemd-run(1) is now used to isolate commands which modify installed packages from the `salt-minion` daemon's control group. This is done to keep systemd from killing any pacman commands spawned by Salt when the `salt-minion` service is restarted. (see `KillMode` in the systemd.kill(5) manpage for more information). If desired, usage of systemd-run(1) can be suppressed by setting a *config option* called `systemd.scope`, with a value of `False` (no quotes).

Install (`pacman -S`) the specified packag(s). Add `refresh=True` to install with `-y`, add `sysupgrade=True` to install with `-u`.

name The name of the package to be installed. Note that this parameter is ignored if either `pkgs` or `sources` is passed. Additionally, please note that this option can only be used to install packages from a software repository. To install a package file manually, use the `sources` option.

CLI Example:

```
salt '*' pkg.install <package name>
```

refresh Whether or not to refresh the package database before installing.

sysupgrade Whether or not to upgrade the system packages before installing. If refresh is set to `True` but sysupgrade is not specified, `-u` will be applied

Multiple Package Installation Options:

pkgs A list of packages to install from a software repository. Must be passed as a python list. A specific version number can be specified by using a single-element dict representing the package and its version. As with the `version` parameter above, comparison operators can be used to target a specific version of a package.

CLI Examples:

```
salt '*' pkg.install pkgs='["foo", "bar"]'
salt '*' pkg.install pkgs='["foo", {"bar": "1.2.3-4"}]'
salt '*' pkg.install pkgs='["foo", {"bar": "<1.2.3-4"}]'
```

sources A list of packages to install. Must be passed as a list of dicts, with the keys being package names, and the values being the source URI or local path to the package.

CLI Example:

```
salt '*' pkg.install                    sources='[{"foo": "salt://foo.pkg.tar.xz
↪"},                  {"bar": "salt://bar.pkg.tar.xz"}]'
```

Returns a dict containing the new package names and versions:

```
{'<package>': {'old': '<old-version>',
               'new': '<new-version>'}}
```

`salt.modules.pacmanpkg.latest_version(*names, **kwargs)`

Return the latest version of the named package available for upgrade or installation. If more than one package name is specified, a dict of name/version pairs is returned.

If the latest version of a given package is already installed, an empty string will be returned for that package.

CLI Example:

```
salt '*' pkg.latest_version <package name>
salt '*' pkg.latest_version <package1> <package2> <package3> ...
```

`salt.modules.pacmanpkg.list_pkgs(versions_as_list=False, **kwargs)`

List the packages currently installed as a dict:

```
{'<package_name>': '<version>'}
```

CLI Example:

```
salt '*' pkg.list_pkgs
```

`salt.modules.pacmanpkg.list_repo_pkgs(*args, **kwargs)`

Returns all available packages. Optionally, package names (and name globs) can be passed and the results will be filtered to packages matching those names.

This function can be helpful in discovering the version or repo to specify in a `pkg.installed` state.

The return data will be a dictionary mapping package names to a list of version numbers, ordered from newest to oldest. If `byrepo` is set to `True`, then the return dictionary will contain repository names at the top level, and each repository will map packages to lists of version numbers. For example:

```
# With byrepo=False (default)
{
    'bash': ['4.4.005-2'],
    'nginx': ['1.10.2-2']
}
# With byrepo=True
{
    'core': {
        'bash': ['4.4.005-2']
    },
    'extra': {
        'nginx': ['1.10.2-2']
    }
}
```

fromrepo [None] Only include results from the specified repo(s). Multiple repos can be specified, comma-separated.

byrepo [False] When `True`, the return data for each package will be organized by repository.

refresh [False] When `True`, the package database will be refreshed (i.e. `pacman -Sy`) before checking for available versions.

CLI Examples:

```
salt '*' pkg.list_repo_pkgs
salt '*' pkg.list_repo_pkgs foo bar baz
salt '*' pkg.list_repo_pkgs 'samba4*' fromrepo=base,updates
salt '*' pkg.list_repo_pkgs 'python2-*' byrepo=True
```

`salt.modules.pacmanpkg.`**`list_upgrades`**(*refresh=False*, *root=None*, ***kwargs*)

List all available package upgrades on this system

CLI Example:

```
salt '*' pkg.list_upgrades
```

`salt.modules.pacmanpkg.`**`owner`**(**paths*)

New in version 2014.7.0.

Return the name of the package that owns the file. Multiple file paths can be passed. Like *`pkg.version`*, if a single path is passed, a string will be returned, and if multiple paths are passed, a dictionary of file/package name pairs will be returned.

If the file is not owned by a package, or is not present on the minion, then an empty string will be returned for that path.

CLI Example:
salt '*' pkg.owner /usr/bin/apachectl salt '*' pkg.owner /usr/bin/apachectl /usr/bin/zsh

`salt.modules.pacmanpkg.`**`purge`**(*name=None*, *pkgs=None*, ***kwargs*)

Changed in version 2015.8.12,2016.3.3,2016.11.0: On minions running systemd>=205, systemd-run(1) is now used to isolate commands which modify installed packages from the `salt-minion` daemon's control group. This is done to keep systemd from killing any pacman commands spawned by Salt when the `salt-minion` service is restarted. (see `KillMode` in the systemd.kill(5) manpage for more information). If desired, usage of systemd-run(1) can be suppressed by setting a *`config option`* called `systemd.scope`, with a value of `False` (no quotes).

Recursively remove a package and all dependencies which were installed with it, this will call a `pacman -Rsc`

name The name of the package to be deleted.

Multiple Package Options:

pkgs A list of packages to delete. Must be passed as a python list. The `name` parameter will be ignored if this option is passed.

New in version 0.16.0.

Returns a dict containing the changes.

CLI Example:

```
salt '*' pkg.purge <package name>
salt '*' pkg.purge <package1>,<package2>,<package3>
salt '*' pkg.purge pkgs='["foo", "bar"]'
```

`salt.modules.pacmanpkg.`**`refresh_db`**(*root=None*)

Just run a `pacman -Sy`, return a dict:

```
{'<database name>': Bool}
```

CLI Example:

```
salt '*' pkg.refresh_db
```

`salt.modules.pacmanpkg.`**`remove`**`(`*name=None, pkgs=None, **kwargs*`)`

Changed in version 2015.8.12,2016.3.3,2016.11.0: On minions running systemd>=205, systemd-run(1) is now used to isolate commands which modify installed packages from the `salt-minion` daemon's control group. This is done to keep systemd from killing any pacman commands spawned by Salt when the `salt-minion` service is restarted. (see `KillMode` in the systemd.kill(5) manpage for more information). If desired, usage of systemd-run(1) can be suppressed by setting a *config option* called `systemd.scope`, with a value of `False` (no quotes).

Remove packages with `pacman -R`.

name The name of the package to be deleted.

Multiple Package Options:

pkgs A list of packages to delete. Must be passed as a python list. The `name` parameter will be ignored if this option is passed.

New in version 0.16.0.

Returns a dict containing the changes.

CLI Example:

```
salt '*' pkg.remove <package name>
salt '*' pkg.remove <package1>,<package2>,<package3>
salt '*' pkg.remove pkgs='["foo", "bar"]'
```

`salt.modules.pacmanpkg.`**`upgrade`**`(`*refresh=False, root=None, **kwargs*`)`

Changed in version 2015.8.12,2016.3.3,2016.11.0: On minions running systemd>=205, systemd-run(1) is now used to isolate commands which modify installed packages from the `salt-minion` daemon's control group. This is done to keep systemd from killing any pacman commands spawned by Salt when the `salt-minion` service is restarted. (see `KillMode` in the systemd.kill(5) manpage for more information). If desired, usage of systemd-run(1) can be suppressed by setting a *config option* called `systemd.scope`, with a value of `False` (no quotes).

Run a full system upgrade, a pacman -Syu

refresh Whether or not to refresh the package database before installing.

Returns a dictionary containing the changes:

```
{'<package>':  {'old': '<old-version>',
               'new': '<new-version>'}}
```

CLI Example:

```
salt '*' pkg.upgrade
```

`salt.modules.pacmanpkg.`**`upgrade_available`**`(`*name*`)`

Check whether or not an upgrade is available for a given package

CLI Example:

```
salt '*' pkg.upgrade_available <package name>
```

`salt.modules.pacmanpkg.`**`version`**`(`**names, **kwargs*`)`

Returns a string representing the package version or an empty string if not installed. If more than one package name is specified, a dict of name/version pairs is returned.

CLI Example:

```
salt '*' pkg.version <package name>
salt '*' pkg.version <package1> <package2> <package3> ...
```

21.9.308 salt.modules.pagerduty

Module for Firing Events via PagerDuty

New in version 2014.1.0.

configuration This module can be used by specifying the name of a configuration profile in the minion config, minion pillar, or master config.

For example:

```
my-pagerduty-account:
    pagerduty.api_key: F3Rbyjbve43rfFWf2214
    pagerduty.subdomain: mysubdomain
```

`salt.modules.pagerduty.`**`create_event`**(*service_key=None*, *description=None*, *details=None*, *incident_key=None*, *profile=None*)

Create an event in PagerDuty. Designed for use in states.

CLI Example:

```
salt myminion pagerduty.create_event <service_key> <description> <details>
↪ profile=my-pagerduty-account
```

The following parameters are required:
service_key This key can be found by using pagerduty.list_services.
description This is a short description of the event.
details This can be a more detailed description of the event.
profile This refers to the configuration profile to use to connect to the PagerDuty service.

`salt.modules.pagerduty.`**`list_escalation_policies`**(*profile=None*, *api_key=None*)

This function is an alias of `list_policies`.

List escalation policies belonging to this account

CLI Example:

salt myminion pagerduty.list_policies my-pagerduty-account salt myminion pagerduty.list_escalation_policies my-pagerduty-account

`salt.modules.pagerduty.`**`list_incidents`**(*profile=None*, *api_key=None*)

List incidents belonging to this account

CLI Example:

salt myminion pagerduty.list_incidents my-pagerduty-account

`salt.modules.pagerduty.`**`list_maintenance_windows`**(*profile=None*, *api_key=None*)

This function is an alias of `list_windows`.

List maintenance windows belonging to this account

CLI Example:

salt myminion pagerduty.list_windows my-pagerduty-account salt myminion pagerduty.list_maintenance_windows my-pagerduty-account

`salt.modules.pagerduty.`**`list_policies`**(*profile=None*, *api_key=None*)

List escalation policies belonging to this account

CLI Example:

salt myminion pagerduty.list_policies my-pagerduty-account salt myminion pagerduty.list_escalation_policies my-pagerduty-account

`salt.modules.pagerduty.`**`list_schedules`**(*profile=None*, *api_key=None*)

List schedules belonging to this account

CLI Example:

salt myminion pagerduty.list_schedules my-pagerduty-account

`salt.modules.pagerduty.`**`list_services`**(*profile=None*, *api_key=None*)

List services belonging to this account

CLI Example:

salt myminion pagerduty.list_services my-pagerduty-account

`salt.modules.pagerduty.`**`list_users`**(*profile=None*, *api_key=None*)

List users belonging to this account

CLI Example:

salt myminion pagerduty.list_users my-pagerduty-account

`salt.modules.pagerduty.`**`list_windows`**(*profile=None*, *api_key=None*)

List maintenance windows belonging to this account

CLI Example:

salt myminion pagerduty.list_windows my-pagerduty-account salt myminion pagerduty.list_maintenance_windows my-pagerduty-account

21.9.309 salt.modules.pagerduty_util

Module for manageing PagerDuty resource

configuration This module can be used by specifying the name of a configuration profile in the minion config, minion pillar, or master config. The default configuration profile name is 'pagerduty.'

For example:

```
pagerduty:
    pagerduty.api_key: F3Rbyjbve43rfFWf2214
    pagerduty.subdomain: mysubdomain
```

For PagerDuty API details, see https://developer.pagerduty.com/documentation/rest

`salt.modules.pagerduty_util.`**`create_or_update_resource`**(*resource_name*, *identifier_fields*, *data*, *diff=None*, *profile='pagerduty'*, *subdomain=None*, *api_key=None*)

create or update any pagerduty resource Helper method for present().

Determining if two resources are the same is different for different PD resource, so this method accepts a diff function. The diff function will be invoked as diff(state_information, object_returned_from_pagerduty), and should return a dict of data to pass to the PagerDuty update API method, or None if no update is to be performed. If no diff method is provided, the default behavior is to scan the keys in the state_information, comparing the matching values in the object_returned_from_pagerduty, and update any values that differ.

Examples

create_or_update_resource("user", ["id","name","email"]) create_or_update_resource("escalation_policies", ["id","name"], diff=my_diff_function)

`salt.modules.pagerduty_util.`**`delete_resource`**(*resource_name*, *key*, *identifier_fields*, *profile='pagerduty'*, *subdomain=None*, *api_key=None*)

delete any pagerduty resource

Helper method for absent()

Example

delete_resource("users", key, ["id","name","email"]) # delete by id or name or email

`salt.modules.pagerduty_util.`**`get_escalation_policies`**(*profile='pagerduty'*, *subdomain=None*, *api_key=None*)

List escalation_policies belonging to this account

CLI Example:

salt myminion pagerduty.get_escalation_policies

`salt.modules.pagerduty_util.`**`get_resource`**(*resource_name*, *key*, *identifier_fields*, *profile='pagerduty'*, *subdomain=None*, *api_key=None*)

Get any single pagerduty resource by key.

We allow flexible lookup by any of a list of identifier_fields. So, for example, you can look up users by email address or name by calling:

get_resource('users', key, ['name', 'email'], ...)

This method is mainly used to translate state sls into pagerduty id's for dependent objects. For example, a pagerduty escalation policy contains one or more schedules, which must be passed by their pagerduty id. We look up the schedules by name (using this method), and then translate the names into id's.

This method is implemented by getting all objects of the resource type (cached into __context__), then brute force searching through the list and trying to match any of the identifier_fields. The __context__ cache is purged after any create, update or delete to the resource.

`salt.modules.pagerduty_util.`**`get_schedules`**(*profile='pagerduty'*, *subdomain=None*, *api_key=None*)

List schedules belonging to this account

CLI Example:

salt myminion pagerduty.get_schedules

`salt.modules.pagerduty_util.`**`get_services`**(*profile='pagerduty'*, *subdomain=None*, *api_key=None*)

List services belonging to this account

CLI Example:

salt myminion pagerduty.get_services

`salt.modules.pagerduty_util.`**`get_users`**(*profile='pagerduty'*, *subdomain=None*, *api_key=None*)

List users belonging to this account

CLI Example:

salt myminion pagerduty.get_users

`salt.modules.pagerduty_util.`**`resource_absent`**(*resource*, *identifier_fields*, *profile='pagerduty'*, *subdomain=None*, *api_key=None*, ***kwargs*)

Generic resource.absent state method. Pagerduty state modules should be a thin wrapper over this method, with a custom diff function.

This method calls delete_resource() and formats the result as a salt state return value.

Example

resource_absent("users", ["id","name","email"]) `salt.modules.pagerduty_util.`**`resource_present`**(*resource*, *identifier_fields*, *diff=None*, *profile='pagerduty'*, *subdomain=None*, *api_key=None*, ***kwargs*)

Generic resource.present state method. Pagerduty state modules should be a thin wrapper over this method, with a custom diff function.

This method calls create_or_update_resource() and formats the result as a salt state return value.

Example

resource_present("users", ["id","name","email"])

21.9.310 salt.modules.pam

Support for pam

`salt.modules.pam.`**`read_file`**(*file_name*)

This is just a test function, to make sure parsing works

CLI Example:

```
salt '*' pam.read_file /etc/pam.d/login
```

21.9.311 salt.modules.panos

Module to provide Palo Alto compatibility to Salt

codeauthor `Spencer Ervin <spencer_ervin@hotmail.com>`

maturity new

depends none

platform unix

New in version 2018.3.0.

Configuration

This module accepts connection configuration details either as parameters, or as configuration settings in pillar as a Salt proxy. Options passed into opts will be ignored if options are passed into pillar.

See also:

`Palo Alto Proxy Module`

About

This execution module was designed to handle connections to a Palo Alto based firewall. This module adds support to send connections directly to the device through the XML API or through a brokered connection to Panorama.

`salt.modules.panos.`**`add_config_lock`**()

Prevent other users from changing configuration until the lock is released.

CLI Example:

```
salt '*' panos.add_config_lock
```

`salt.modules.panos.`**`check_antivirus`**`()`

Get anti-virus information from PaloAlto Networks server

CLI Example:

```
salt '*' panos.check_antivirus
```

`salt.modules.panos.`**`check_software`**`()`

Get software information from PaloAlto Networks server.

CLI Example:

```
salt '*' panos.check_software
```

`salt.modules.panos.`**`clear_commit_tasks`**`()`

Clear all commit tasks.

CLI Example:

```
salt '*' panos.clear_commit_tasks
```

`salt.modules.panos.`**`commit`**`()`

Commits the candidate configuration to the running configuration.

CLI Example:

```
salt '*' panos.commit
```

`salt.modules.panos.`**`deactivate_license`**`(`*key_name=None*`)`

Deactivates an installed license. Required version 7.0.0 or greater.

key_name(str): The file name of the license key installed.

CLI Example:

```
salt '*' panos.deactivate_license key_name=License_File_Name.key
```

`salt.modules.panos.`**`delete_license`**`(`*key_name=None*`)`

Remove license keys on disk.

key_name(str): The file name of the license key to be deleted.

CLI Example:

```
salt '*' panos.delete_license key_name=License_File_Name.key
```

`salt.modules.panos.`**`download_antivirus`**`()`

Download the most recent anti-virus package.

CLI Example:

```
salt '*' panos.download_antivirus
```

`salt.modules.panos.`**`download_software_file`**`(`*filename=None, synch=False*`)`

Download software packages by filename.

Parameters

- **filename** (*str*) – The filename of the PANOS file to download.
- **synch** (*bool*) – If true then the file will synch to the peer unit.

CLI Example:

```
salt '*' panos.download_software_file PanOS_5000-8.0.0
salt '*' panos.download_software_file PanOS_5000-8.0.0 True
```

`salt.modules.panos.`**`download_software_version`**(*version=None*, *synch=False*)

Download software packages by version number.

Parameters

- **version** (*str*) – The version of the PANOS file to download.
- **synch** (*bool*) – If true then the file will synch to the peer unit.

CLI Example:

```
salt '*' panos.download_software_version 8.0.0
salt '*' panos.download_software_version 8.0.0 True
```

`salt.modules.panos.`**`fetch_license`**(*auth_code=None*)

Get new license(s) using from the Palo Alto Network Server.

auth_code The license authorization code.

CLI Example:

```
salt '*' panos.fetch_license
salt '*' panos.fetch_license auth_code=foobar
```

`salt.modules.panos.`**`get_address`**(*address=None*, *vsys='1'*)

Get the candidate configuration for the specified get_address object. This will not return address objects that are marked as pre-defined objects.

address(str): The name of the address object.

vsys(str): The string representation of the VSYS ID.

CLI Example:

```
salt '*' panos.get_address myhost
salt '*' panos.get_address myhost 3
```

`salt.modules.panos.`**`get_address_group`**(*addressgroup=None*, *vsys='1'*)

Get the candidate configuration for the specified address group. This will not return address groups that are marked as pre-defined objects.

addressgroup(str): The name of the address group.

vsys(str): The string representation of the VSYS ID.

CLI Example:

```
salt '*' panos.get_address_group foobar
salt '*' panos.get_address_group foobar 3
```

`salt.modules.panos.`**`get_admins_active`**()

Show active administrators.

CLI Example:

```
salt '*' panos.get_admins_active
```

`salt.modules.panos.`**`get_admins_all`**()

Show all administrators.

CLI Example:

```
salt '*' panos.get_admins_all
```

salt.modules.panos.**get_antivirus_info**()

Show information about available anti-virus packages.

CLI Example:

```
salt '*' panos.get_antivirus_info
```

salt.modules.panos.**get_arp**()

Show ARP information.

CLI Example:

```
salt '*' panos.get_arp
```

salt.modules.panos.**get_cli_idle_timeout**()

Show timeout information for this administrative session.

CLI Example:

```
salt '*' panos.get_cli_idle_timeout
```

salt.modules.panos.**get_cli_permissions**()

Show cli administrative permissions.

CLI Example:

```
salt '*' panos.get_cli_permissions
```

salt.modules.panos.**get_disk_usage**()

Report filesystem disk space usage.

CLI Example:

```
salt '*' panos.get_disk_usage
```

salt.modules.panos.**get_dns_server_config**()

Get the DNS server configuration from the candidate configuration.

CLI Example:

```
salt '*' panos.get_dns_server_config
```

salt.modules.panos.**get_domain_config**()

Get the domain name configuration from the candidate configuration.

CLI Example:

```
salt '*' panos.get_domain_config
```

salt.modules.panos.**get_dos_blocks**()

Show the DoS block-ip table.

CLI Example:

```
salt '*' panos.get_dos_blocks
```

salt.modules.panos.**get_fqdn_cache**()

Print FQDNs used in rules and their IPs.

CLI Example:

```
salt '*' panos.get_fqdn_cache
```

salt.modules.panos.**get_ha_config**()

Get the high availability configuration.

CLI Example:

```
salt '*' panos.get_ha_config
```

salt.modules.panos.**get_ha_link**()

Show high-availability link-monitoring state.

CLI Example:

```
salt '*' panos.get_ha_link
```

salt.modules.panos.**get_ha_path**()

Show high-availability path-monitoring state.

CLI Example:

```
salt '*' panos.get_ha_path
```

salt.modules.panos.**get_ha_state**()

Show high-availability state information.

CLI Example:

```
salt '*' panos.get_ha_state
```

salt.modules.panos.**get_ha_transitions**()

Show high-availability transition statistic information.

CLI Example:

```
salt '*' panos.get_ha_transitions
```

salt.modules.panos.**get_hostname**()

Get the hostname of the device.

CLI Example:

```
salt '*' panos.get_hostname
```

salt.modules.panos.**get_interface_counters**(*name='all'*)

Get the counter statistics for interfaces.

Parameters **name** (*str*) – The name of the interface to view. By default, all interface statistics are viewed.

CLI Example:

```
salt '*' panos.get_interface_counters
salt '*' panos.get_interface_counters ethernet1/1
```

salt.modules.panos.**get_interfaces**(*name='all'*)

Show interface information.

Parameters **name** (*str*) – The name of the interface to view. By default, all interface statistics are viewed.

CLI Example:

```
salt '*' panos.get_interfaces
salt '*' panos.get_interfaces ethernet1/1
```

`salt.modules.panos.`**`get_job`**`(`*jid=None*`)`

List all a single job by ID.

jid The ID of the job to retrieve.

CLI Example:

```
salt '*' panos.get_job jid=15
```

`salt.modules.panos.`**`get_jobs`**`(`*state='all'*`)`

List all jobs on the device.

state The state of the jobs to display. Valid options are all, pending, or processed. Pending jobs are jobs that are currently in a running or waiting state. Processed jobs are jobs that have completed execution.

CLI Example:

```
salt '*' panos.get_jobs
salt '*' panos.get_jobs state=pending
```

`salt.modules.panos.`**`get_lacp`**`()`

Show LACP state.

CLI Example:

```
salt '*' panos.get_lacp
```

`salt.modules.panos.`**`get_license_info`**`()`

Show information about owned license(s).

CLI Example:

```
salt '*' panos.get_license_info
```

`salt.modules.panos.`**`get_license_tokens`**`()`

Show license token files for manual license deactivation.

CLI Example:

```
salt '*' panos.get_license_tokens
```

`salt.modules.panos.`**`get_lldp_config`**`()`

Show lldp config for interfaces.

CLI Example:

```
salt '*' panos.get_lldp_config
```

`salt.modules.panos.`**`get_lldp_counters`**`()`

Show lldp counters for interfaces.

CLI Example:

```
salt '*' panos.get_lldp_counters
```

salt.modules.panos.**get_lldp_local**()

Show lldp local info for interfaces.

CLI Example:

```
salt '*' panos.get_lldp_local
```

salt.modules.panos.**get_lldp_neighbors**()

Show lldp neighbors info for interfaces.

CLI Example:

```
salt '*' panos.get_lldp_neighbors
```

salt.modules.panos.**get_local_admins**()

Show all local administrator accounts.

CLI Example:

```
salt '*' panos.get_local_admins
```

salt.modules.panos.**get_logdb_quota**()

Report the logdb quotas.

CLI Example:

```
salt '*' panos.get_logdb_quota
```

salt.modules.panos.**get_master_key**()

Get the master key properties.

CLI Example:

```
salt '*' panos.get_master_key
```

salt.modules.panos.**get_ntp_config**()

Get the NTP configuration from the candidate configuration.

CLI Example:

```
salt '*' panos.get_ntp_config
```

salt.modules.panos.**get_ntp_servers**()

Get list of configured NTP servers.

CLI Example:

```
salt '*' panos.get_ntp_servers
```

salt.modules.panos.**get_operational_mode**()

Show device operational mode setting.

CLI Example:

```
salt '*' panos.get_operational_mode
```

salt.modules.panos.**get_panorama_status**()

Show panorama connection status.

CLI Example:

```
salt '*' panos.get_panorama_status
```

salt.modules.panos.**get_permitted_ips**()

Get the IP addresses that are permitted to establish management connections to the device.

CLI Example:

```
salt '*' panos.get_permitted_ips
```

salt.modules.panos.**get_platform**()

Get the platform model information and limitations.

CLI Example:

```
salt '*' panos.get_platform
```

salt.modules.panos.**get_predefined_application**(*application=None*)

Get the configuration for the specified pre-defined application object. This will only return pre-defined application objects.

application(str): The name of the pre-defined application object.

CLI Example:

```
salt '*' panos.get_predefined_application saltstack
```

salt.modules.panos.**get_security_rule**(*rulename=None, vsys='1'*)

Get the candidate configuration for the specified security rule.

rulename(str): The name of the security rule.

vsys(str): The string representation of the VSYS ID.

CLI Example:

```
salt '*' panos.get_security_rule rule01
salt '*' panos.get_security_rule rule01 3
```

salt.modules.panos.**get_service**(*service=None, vsys='1'*)

Get the candidate configuration for the specified service object. This will not return services that are marked as pre-defined objects.

service(str): The name of the service object.

vsys(str): The string representation of the VSYS ID.

CLI Example:

```
salt '*' panos.get_service tcp-443
salt '*' panos.get_service tcp-443 3
```

salt.modules.panos.**get_service_group**(*servicegroup=None, vsys='1'*)

Get the candidate configuration for the specified service group. This will not return service groups that are marked as pre-defined objects.

servicegroup(str): The name of the service group.

vsys(str): The string representation of the VSYS ID.

CLI Example:

```
salt '*' panos.get_service_group foobar
salt '*' panos.get_service_group foobar 3
```

`salt.modules.panos.`**`get_session_info`**`()`

Show device session statistics.

CLI Example:

```
salt '*' panos.get_session_info
```

`salt.modules.panos.`**`get_snmp_config`**`()`

Get the SNMP configuration from the device.

CLI Example:

```
salt '*' panos.get_snmp_config
```

`salt.modules.panos.`**`get_software_info`**`()`

Show information about available software packages.

CLI Example:

```
salt '*' panos.get_software_info
```

`salt.modules.panos.`**`get_system_date_time`**`()`

Get the system date/time.

CLI Example:

```
salt '*' panos.get_system_date_time
```

`salt.modules.panos.`**`get_system_files`**`()`

List important files in the system.

CLI Example:

```
salt '*' panos.get_system_files
```

`salt.modules.panos.`**`get_system_info`**`()`

Get the system information.

CLI Example:

```
salt '*' panos.get_system_info
```

`salt.modules.panos.`**`get_system_services`**`()`

Show system services.

CLI Example:

```
salt '*' panos.get_system_services
```

`salt.modules.panos.`**`get_system_state`**`(`*mask=None*`)`

Show the system state variables.

mask Filters by a subtree or a wildcard.

CLI Example:

```
salt '*' panos.get_system_state
salt '*' panos.get_system_state mask=cfg.ha.config.enabled
salt '*' panos.get_system_state mask=cfg.ha.*
```

`salt.modules.panos.`**`get_uncommitted_changes`**`()`

Retrieve a list of all uncommitted changes on the device. Requires PANOS version 8.0.0 or greater.

CLI Example:

```
salt '*' panos.get_uncommitted_changes
```

`salt.modules.panos.`**`get_users_config`**`()`

Get the local administrative user account configuration.

CLI Example:

```
salt '*' panos.get_users_config
```

`salt.modules.panos.`**`get_vlans`**`()`

Show all VLAN information.

CLI Example:

```
salt '*' panos.get_vlans
```

`salt.modules.panos.`**`get_xpath`**`(`*xpath=''*`)`

Retrieve a specified xpath from the candidate configuration.

xpath(str): The specified xpath in the candidate configuration.

CLI Example:

```
salt '*' panos.get_xpath /config/shared/service
```

`salt.modules.panos.`**`get_zone`**`(`*zone='', vsys='1'*`)`

Get the candidate configuration for the specified zone.

zone(str): The name of the zone.

vsys(str): The string representation of the VSYS ID.

CLI Example:

```
salt '*' panos.get_zone trust
salt '*' panos.get_zone trust 2
```

`salt.modules.panos.`**`get_zones`**`(`*vsys='1'*`)`

Get all the zones in the candidate configuration.

vsys(str): The string representation of the VSYS ID.

CLI Example:

```
salt '*' panos.get_zones
salt '*' panos.get_zones 2
```

`salt.modules.panos.`**`install_antivirus`**`(`*version=None, latest=False, synch=False, skip_commit=False*`)`

Install anti-virus packages.

Parameters

- **`version`** (`str`) – The version of the PANOS file to install.

- **latest** (*bool*) – If true, the latest anti-virus file will be installed. The specified version option will be ignored.
- **synch** (*bool*) – If true, the anti-virus will synch to the peer unit.
- **skip_commit** (*bool*) – If true, the install will skip committing to the device.

CLI Example:

```
salt '*' panos.install_antivirus 8.0.0
```

`salt.modules.panos.`**`install_license`**`()`

Install the license key(s).

CLI Example:

```
salt '*' panos.install_license
```

`salt.modules.panos.`**`install_software`**`(`*version=None*`)`

Upgrade to a software package by version.

Parameters **version** (*str*) – The version of the PANOS file to install.

CLI Example:

```
salt '*' panos.install_license 8.0.0
```

`salt.modules.panos.`**`reboot`**`()`

Reboot a running system.

CLI Example:

```
salt '*' panos.reboot
```

`salt.modules.panos.`**`refresh_fqdn_cache`**`(`*force=False*`)`

Force refreshes all FQDNs used in rules.

force Forces all fqdn refresh

CLI Example:

```
salt '*' panos.refresh_fqdn_cache
salt '*' panos.refresh_fqdn_cache force=True
```

`salt.modules.panos.`**`remove_config_lock`**`()`

Release config lock previously held.

CLI Example:

```
salt '*' panos.remove_config_lock
```

`salt.modules.panos.`**`resolve_address`**`(`*address=None, vsys=None*`)`

Resolve address to ip address. Required version 7.0.0 or greater.

address Address name you want to resolve.

vsys The vsys name.

CLI Example:

```
salt '*' panos.resolve_address foo.bar.com
salt '*' panos.resolve_address foo.bar.com vsys=2
```

`salt.modules.panos.`**`save_device_config`**`(`*filename=None*`)`

Save device configuration to a named file.

filename The filename to save the configuration to.

CLI Example:

```
salt '*' panos.save_device_config foo.xml
```

`salt.modules.panos.`**`save_device_state`**`()`

Save files needed to restore device to local disk.

CLI Example:

```
salt '*' panos.save_device_state
```

`salt.modules.panos.`**`set_authentication_profile`**`(`*profile=None, deploy=False*`)`

Set the authentication profile of the Palo Alto proxy minion. A commit will be required before this is processed.

CLI Example:

Parameters

- **profile** (*str*) – The name of the authentication profile to set.
- **deploy** (*bool*) – If true then commit the full candidate configuration, if false only set pending change.

```
salt '*' panos.set_authentication_profile foo
salt '*' panos.set_authentication_profile foo deploy=True
```

`salt.modules.panos.`**`set_hostname`**`(`*hostname=None, deploy=False*`)`

Set the hostname of the Palo Alto proxy minion. A commit will be required before this is processed.

CLI Example:

Parameters

- **hostname** (*str*) – The hostname to set
- **deploy** (*bool*) – If true then commit the full candidate configuration, if false only set pending change.

```
salt '*' panos.set_hostname newhostname
salt '*' panos.set_hostname newhostname deploy=True
```

`salt.modules.panos.`**`set_management_http`**`(`*enabled=True, deploy=False*`)`

Enables or disables the HTTP management service on the device.

CLI Example:

Parameters

- **enabled** (*bool*) – If true the service will be enabled. If false the service will be disabled.
- **deploy** (*bool*) – If true then commit the full candidate configuration, if false only set pending change.

```
salt '*' panos.set_management_http
salt '*' panos.set_management_http enabled=False deploy=True
```

`salt.modules.panos.`**`set_management_https`**`(`*enabled=True, deploy=False*`)`

Enables or disables the HTTPS management service on the device.

CLI Example:

Parameters

- **enabled** (*bool*) – If true the service will be enabled. If false the service will be disabled.
- **deploy** (*bool*) – If true then commit the full candidate configuration, if false only set pending change.

```
salt '*' panos.set_management_https
salt '*' panos.set_management_https enabled=False deploy=True
```

salt.modules.panos.**set_management_icmp**(*enabled=True*, *deploy=False*)
Enables or disables the ICMP management service on the device.

CLI Example:

Parameters

- **enabled** (*bool*) – If true the service will be enabled. If false the service will be disabled.
- **deploy** (*bool*) – If true then commit the full candidate configuration, if false only set pending change.

```
salt '*' panos.set_management_icmp
salt '*' panos.set_management_icmp enabled=False deploy=True
```

salt.modules.panos.**set_management_ocsp**(*enabled=True*, *deploy=False*)
Enables or disables the HTTP OCSP management service on the device.

CLI Example:

Parameters

- **enabled** (*bool*) – If true the service will be enabled. If false the service will be disabled.
- **deploy** (*bool*) – If true then commit the full candidate configuration, if false only set pending change.

```
salt '*' panos.set_management_ocsp
salt '*' panos.set_management_ocsp enabled=False deploy=True
```

salt.modules.panos.**set_management_snmp**(*enabled=True*, *deploy=False*)
Enables or disables the SNMP management service on the device.

CLI Example:

Parameters

- **enabled** (*bool*) – If true the service will be enabled. If false the service will be disabled.
- **deploy** (*bool*) – If true then commit the full candidate configuration, if false only set pending change.

```
salt '*' panos.set_management_snmp
salt '*' panos.set_management_snmp enabled=False deploy=True
```

salt.modules.panos.**set_management_ssh**(*enabled=True*, *deploy=False*)
Enables or disables the SSH management service on the device.

CLI Example:

Parameters

- **enabled** (*bool*) – If true the service will be enabled. If false the service will be disabled.
- **deploy** (*bool*) – If true then commit the full candidate configuration, if false only set pending change.

```
salt '*' panos.set_management_ssh
salt '*' panos.set_management_ssh enabled=False deploy=True
```

salt.modules.panos.**set_management_telnet**(*enabled=True*, *deploy=False*)
Enables or disables the Telnet management service on the device.

CLI Example:

Parameters

- **enabled** (*bool*) – If true the service will be enabled. If false the service will be disabled.
- **deploy** (*bool*) – If true then commit the full candidate configuration, if false only set pending change.

```
salt '*' panos.set_management_telnet
salt '*' panos.set_management_telnet enabled=False deploy=True
```

`salt.modules.panos.`**`set_ntp_authentication`**(*target=None, authentication_type=None, key_id=None, authentication_key=None, algorithm=None, deploy=False*)

Set the NTP authentication of the Palo Alto proxy minion. A commit will be required before this is processed.

CLI Example:

Parameters

- **target** (*str*) – Determines the target of the authentication. Valid options are primary, secondary, or both.
- **authentication_type** (*str*) – The authentication type to be used. Valid options are symmetric, autokey, and none.
- **key_id** (*int*) – The NTP authentication key ID.
- **authentication_key** (*str*) – The authentication key.
- **algorithm** (*str*) – The algorithm type to be used for a symmetric key. Valid options are md5 and sha1.
- **deploy** (*bool*) – If true then commit the full candidate configuration, if false only set pending change.

```
salt '*' ntp.set_authentication target=both authentication_type=autokey
salt '*' ntp.set_authentication target=primary authentication_type=none
salt '*' ntp.set_authentication target=both authentication_type=symmetric key_
↪id=15 authentication_key=mykey algorithm=md5
salt '*' ntp.set_authentication target=both authentication_type=symmetric key_
↪id=15 authentication_key=mykey algorithm=md5 deploy=True
```

`salt.modules.panos.`**`set_ntp_servers`**(*primary_server=None, secondary_server=None, deploy=False*)

Set the NTP servers of the Palo Alto proxy minion. A commit will be required before this is processed.

CLI Example:

Parameters

- **primary_server** (*str*) – The primary NTP server IP address or FQDN.
- **secondary_server** (*str*) – The secondary NTP server IP address or FQDN.
- **deploy** (*bool*) – If true then commit the full candidate configuration, if false only set pending change.

```
salt '*' ntp.set_servers 0.pool.ntp.org 1.pool.ntp.org
salt '*' ntp.set_servers primary_server=0.pool.ntp.org secondary_server=1.pool.
↪ntp.org
salt '*' ntp.ser_servers 0.pool.ntp.org 1.pool.ntp.org deploy=True
```

`salt.modules.panos.`**`set_permitted_ip`**(*address=None, deploy=False*)

Add an IPv4 address or network to the permitted IP list.

CLI Example:

Parameters

- **address** (*str*) – The IPv4 address or network to allow access to add to the Palo Alto device.
- **deploy** (*bool*) – If true then commit the full candidate configuration, if false only set pending change.

```
salt '*' panos.set_permitted_ip 10.0.0.1
salt '*' panos.set_permitted_ip 10.0.0.0/24
salt '*' panos.set_permitted_ip 10.0.0.1 deploy=True
```

`salt.modules.panos.`**`set_timezone`**`(`*tz=None, deploy=False*`)`

Set the timezone of the Palo Alto proxy minion. A commit will be required before this is processed.

CLI Example:

Parameters

- **tz** (*str*) – The name of the timezone to set.
- **deploy** (*bool*) – If true then commit the full candidate configuration, if false only set pending change.

```
salt '*' panos.set_timezone UTC
salt '*' panos.set_timezone UTC deploy=True
```

`salt.modules.panos.`**`shutdown`**`()`

Shutdown a running system.

CLI Example:

```
salt '*' panos.shutdown
```

`salt.modules.panos.`**`test_fib_route`**`(`*ip=None, vr='vr1'*`)`

Perform a route lookup within active route table (fib).

ip (str): The destination IP address to test.

vr (str): The name of the virtual router to test.

CLI Example:

```
salt '*' panos.test_fib_route 4.2.2.2
salt '*' panos.test_fib_route 4.2.2.2 my-vr
```

`salt.modules.panos.`**`test_security_policy`**`(`*sourcezone=None, destinationzone=None, source=None, destination=None, protocol=None, port=None, application=None, category=None, vsys='1', allrules=False*`)`

Checks which security policy as connection will match on the device.

sourcezone (str): The source zone matched against the connection.

destinationzone (str): The destination zone matched against the connection.

source (str): The source address. This must be a single IP address.

destination (str): The destination address. This must be a single IP address.

protocol (int): The protocol number for the connection. This is the numerical representation of the protocol.

port (int): The port number for the connection.

application (str): The application that should be matched.

category (str): The category that should be matched.

vsys (int): The numerical representation of the VSYS ID.

allrules (bool): Show all potential match rules until first allow rule.

CLI Example:

```
salt '*' panos.test_security_policy sourcezone=trust destinationzone=untrust␣
↪protocol=6 port=22
salt '*' panos.test_security_policy sourcezone=trust destinationzone=untrust␣
↪protocol=6 port=22 vsys=2
```

`salt.modules.panos.`**`unlock_admin`**(*username=None*)

Unlocks a locked administrator account.

username Username of the administrator.

CLI Example:

```
salt '*' panos.unlock_admin username=bob
```

21.9.312 salt.modules.parallels module

Manage Parallels Desktop VMs with `prlctl` and `prlsrvctl`. Only some of the prlctl commands implemented so far. Of those that have been implemented, not all of the options may have been provided yet. For a complete reference, see the Parallels Desktop Reference Guide.

This module requires the prlctl binary to be installed to run most functions. To run parallels.prlsrvctl, the prlsrvctl binary is required.

What has not been implemented yet can be accessed through `parallels.prlctl` and `parallels.prlsrvctl` (note the preceding double dash `--` as necessary):

```
salt '*' parallels.prlctl installtools macvm runas=macdev
salt -- '*' parallels.prlctl capture 'macvm --file macvm.display.png' runas=macdev
salt -- '*' parallels.prlsrvctl set '--mem-limit auto' runas=macdev
```

New in version 2016.3.0.

`salt.modules.parallels.`**`clone`**(*name, new_name, linked=False, template=False, runas=None*)

Clone a VM

New in version 2016.11.0.

Parameters

- **`name`** (*`str`*) – Name/ID of VM to clone
- **`new_name`** (*`str`*) – Name of the new VM
- **`linked`** (*`bool`*) – Create a linked virtual machine.
- **`template`** (*`bool`*) – Create a virtual machine template instead of a real virtual machine.
- **`runas`** (*`str`*) – The user that the prlctl command will be run as

Example:

```
salt '*' parallels.clone macvm macvm_new runas=macdev
salt '*' parallels.clone macvm macvm_templ template=True runas=macdev
```

`salt.modules.parallels.`**`delete`**(*name, runas=None*)

Delete a VM

New in version 2016.11.0.

Parameters

- **`name`** (*`str`*) – Name/ID of VM to clone
- **`runas`** (*`str`*) – The user that the prlctl command will be run as

Example:

```
salt '*' parallels.exec macvm 'find /etc/paths.d' runas=macdev
```

`salt.modules.parallels.`**`delete_snapshot`**(*name*, *snap_name*, *runas=None*, *all=False*)

Delete a snapshot

Note: Deleting a snapshot from which other snapshots are dervied will not delete the derived snapshots

Parameters

- **name** (*str*) – Name/ID of VM whose snapshot will be deleted
- **snap_name** (*str*) – Name/ID of snapshot to delete
- **runas** (*str*) – The user that the prlctl command will be run as
- **all** (*bool*) – Delete all snapshots having the name given

  New in version 2016.11.0.

Example:

```
salt '*' parallels.delete_snapshot macvm 'unneeded snapshot' runas=macdev
salt '*' parallels.delete_snapshot macvm 'Snapshot for linked clone' all=True
↪runas=macdev
```

`salt.modules.parallels.`**`exec_`**(*name*, *command*, *runas=None*)

Run a command on a VM

Parameters

- **name** (*str*) – Name/ID of VM whose exec will be returned
- **command** (*str*) – Command to run on the VM
- **runas** (*str*) – The user that the prlctl command will be run as

Example:

```
salt '*' parallels.exec macvm 'find /etc/paths.d' runas=macdev
```

`salt.modules.parallels.`**`exists`**(*name*, *runas=None*)

Query whether a VM exists

New in version 2016.11.0.

Parameters

- **name** (*str*) – Name/ID of VM
- **runas** (*str*) – The user that the prlctl command will be run as

Example:

```
salt '*' parallels.exists macvm runas=macdev
```

`salt.modules.parallels.`**`list_snapshots`**(*name*, *snap_name=None*, *tree=False*, *names=False*, *runas=None*)

List the snapshots

Parameters

- **name** (*str*) – Name/ID of VM whose snapshots will be listed
- **snap_id** (*str*) – Name/ID of snapshot to display information about. If `tree=True` is also specified, display the snapshot subtree having this snapshot as the root snapshot
- **tree** (*bool*) – List snapshots in tree format rather than tabular format
- **names** (*bool*) – List snapshots as ID, name pairs
- **runas** (*str*) – The user that the prlctl command will be run as

Example:

```
salt '*' parallels.list_snapshots macvm runas=macdev
salt '*' parallels.list_snapshots macvm tree=True runas=macdev
salt '*' parallels.list_snapshots macvm snap_name=original runas=macdev
salt '*' parallels.list_snapshots macvm names=True runas=macdev
```

`salt.modules.parallels.`**`list_vms`**(*name=None, info=False, all=False, args=None, runas=None, template=False*)

List information about the VMs

Parameters

- **name** (*str*) – Name/ID of VM to list

  Changed in version 2016.11.0: No longer implies `info=True`
- **info** (*str*) – List extra information
- **all** (*bool*) – List all non-template VMs
- **args** (*tuple*) – Additional arguments given to `prctl list`
- **runas** (*str*) – The user that the prlctl command will be run as
- **template** (*bool*) – List the available virtual machine templates. The real virtual machines will not be included in the output

  New in version 2016.11.0.

Example:

```
salt '*' parallels.list_vms runas=macdev
salt '*' parallels.list_vms name=macvm info=True runas=macdev
salt '*' parallels.list_vms info=True runas=macdev
salt '*' parallels.list_vms ' -o uuid,status' all=True runas=macdev
```

`salt.modules.parallels.`**`prlctl`**(*sub_cmd, args=None, runas=None*)

Execute a prlctl command

Parameters

- **sub_cmd** (*str*) – prlctl subcommand to execute
- **args** (*str*) – The arguments supplied to `prlctl <sub_cmd>`
- **runas** (*str*) – The user that the prlctl command will be run as

Example:

```
salt '*' parallels.prlctl user list runas=macdev
salt '*' parallels.prlctl exec 'macvm uname' runas=macdev
salt -- '*' parallels.prlctl capture 'macvm --file macvm.display.png' runas=macdev
```

`salt.modules.parallels.`**`prlsrvctl`**(*sub_cmd, args=None, runas=None*)

Execute a prlsrvctl command

New in version 2016.11.0.

Parameters

- **sub_cmd** (*str*) – prlsrvctl subcommand to execute
- **args** (*str*) – The arguments supplied to `prlsrvctl <sub_cmd>`
- **runas** (*str*) – The user that the prlsrvctl command will be run as

Example:

```
salt '*' parallels.prlsrvctl info runas=macdev
salt '*' parallels.prlsrvctl usb list runas=macdev
salt -- '*' parallels.prlsrvctl set '--mem-limit auto' runas=macdev
```

`salt.modules.parallels.`**`reset`**(*name, runas=None*)

Reset a VM by performing a hard shutdown and then a restart

Parameters

- **name** (*str*) – Name/ID of VM to reset

- **runas** (*str*) – The user that the prlctl command will be run as

Example:

```
salt '*' parallels.reset macvm runas=macdev
```

salt.modules.parallels.**restart**(*name*, *runas=None*)

Restart a VM by gracefully shutting it down and then restarting it

Parameters

- **name** (*str*) – Name/ID of VM to restart
- **runas** (*str*) – The user that the prlctl command will be run as

Example:

```
salt '*' parallels.restart macvm runas=macdev
```

salt.modules.parallels.**revert_snapshot**(*name*, *snap_name*, *runas=None*)

Revert a VM to a snapshot

Parameters

- **name** (*str*) – Name/ID of VM to revert to a snapshot
- **snap_name** (*str*) – Name/ID of snapshot to revert to
- **runas** (*str*) – The user that the prlctl command will be run as

Example:

```
salt '*' parallels.revert_snapshot macvm base-with-updates runas=macdev
```

salt.modules.parallels.**snapshot**(*name*, *snap_name=None*, *desc=None*, *runas=None*)

Create a snapshot

Parameters

- **name** (*str*) – Name/ID of VM to take a snapshot of
- **snap_name** (*str*) – Name of snapshot
- **desc** (*str*) – Description of snapshot
- **runas** (*str*) – The user that the prlctl command will be run as

Example:

```
salt '*' parallels.create_snapshot macvm snap_name=macvm-original runas=macdev
salt '*' parallels.create_snapshot macvm snap_name=macvm-updates desc='clean
↪install with updates' runas=macdev
```

salt.modules.parallels.**snapshot_id_to_name**(*name*, *snap_id*, *strict=False*, *runas=None*)

Attempt to convert a snapshot ID to a snapshot name. If the snapshot has no name or if the ID is not found or invalid, an empty string will be returned

Parameters

- **name** (*str*) – Name/ID of VM whose snapshots are inspected
- **snap_id** (*str*) – ID of the snapshot
- **strict** (*bool*) – Raise an exception if a name cannot be found for the given `snap_id`
- **runas** (*str*) – The user that the prlctl command will be run as

Example data

```
ID: {a5b8999f-5d95-4aff-82de-e515b0101b66}
Name: original
Date: 2016-03-04 10:50:34
Current: yes
State: poweroff
Description: original state
```

CLI Example:

```
salt '*' parallels.snapshot_id_to_name macvm a5b8999f-5d95-4aff-82de-e515b0101b66␣
↪runas=macdev
```

`salt.modules.parallels.`**`snapshot_name_to_id`**(*name*, *snap_name*, *strict=False*, *runas=None*)

Attempt to convert a snapshot name to a snapshot ID. If the name is not found an empty string is returned. If multiple snapshots share the same name, a list will be returned

Parameters

- **name** (*str*) – Name/ID of VM whose snapshots are inspected
- **snap_name** (*str*) – Name of the snapshot
- **strict** (*bool*) – Raise an exception if multiple snapshot IDs are found
- **runas** (*str*) – The user that the prlctl command will be run as

CLI Example:

```
salt '*' parallels.snapshot_id_to_name macvm original runas=macdev
```

`salt.modules.parallels.`**`start`**(*name*, *runas=None*)

Start a VM

Parameters

- **name** (*str*) – Name/ID of VM to start
- **runas** (*str*) – The user that the prlctl command will be run as

Example:

```
salt '*' parallels.start macvm runas=macdev
```

`salt.modules.parallels.`**`status`**(*name*, *runas=None*)

Status of a VM

Parameters

- **name** (*str*) – Name/ID of VM whose status will be returned
- **runas** (*str*) – The user that the prlctl command will be run as

Example:

```
salt '*' parallels.status macvm runas=macdev
```

`salt.modules.parallels.`**`stop`**(*name*, *kill=False*, *runas=None*)

Stop a VM

Parameters

- **name** (*str*) – Name/ID of VM to stop
- **kill** (*bool*) – Perform a hard shutdown
- **runas** (*str*) – The user that the prlctl command will be run as

Example:

```
salt '*' parallels.stop macvm runas=macdev
salt '*' parallels.stop macvm kill=True runas=macdev
```

21.9.313 salt.modules.parted_partition

Module for managing partitions on POSIX-like systems.

depends

- parted, partprobe, lsblk (usually parted and util-linux packages)

Some functions may not be available, depending on your version of parted.

Check the manpage for `parted(8)` for more information, or the online docs at:

http://www.gnu.org/software/parted/manual/html_chapter/parted_2.html

In light of parted not directly supporting partition IDs, some of this module has been written to utilize sfdisk instead. For further information, please reference the man page for `sfdisk(8)`.

`salt.modules.parted_partition.`**`align_check`**`(`*device, part_type, partition*`)`

Check if partition satisfies the alignment constraint of part_type. Type must be "minimal" or "optimal".

CLI Example:

```
salt '*' partition.align_check /dev/sda minimal 1
```

`salt.modules.parted_partition.`**`check`**`(`*device, minor*`)`

Checks if the file system on partition <minor> has any errors.

CLI Example:

```
salt '*' partition.check 1
```

`salt.modules.parted_partition.`**`cp`**`(`*device, from_minor, to_minor*`)`

Copies the file system on the partition <from-minor> to partition <to-minor>, deleting the original contents of the destination partition.

CLI Example:

```
salt '*' partition.cp /dev/sda 2 3
```

`salt.modules.parted_partition.`**`disk_set`**`(`*device, flag, state*`)`

Changes a flag on selected device.

A flag can be either "on" or "off" (make sure to use proper quoting, see *YAML Idiosyncrasies*). Some or all of these flags will be available, depending on what disk label you are using.

Valid flags are:

- cylinder_alignment
- pmbr_boot
- implicit_partition_table

CLI Example:

```
salt '*' partition.disk_set /dev/sda pmbr_boot '"on"'
```

`salt.modules.parted_partition.`**`disk_toggle`**`(`*device, flag*`)`

Toggle the state of <flag> on <device>. Valid flags are the same as the disk_set command.

CLI Example:

```
salt '*' partition.disk_toggle /dev/sda pmbr_boot
```

`salt.modules.parted_partition.`**`exists`**`(`*device=''*`)`

Check to see if the partition exists

CLI Example:

```
salt '*' partition.exists /dev/sdb1
```

`salt.modules.parted_partition.`**`get_block_device`**`()`

Retrieve a list of disk devices

New in version 2014.7.0.

CLI Example:

```
salt '*' partition.get_block_device
```

`salt.modules.parted_partition.`**`get_id`**(*device*, *minor*)

Prints the system ID for the partition. Some typical values are:

```
 b: FAT32 (vfat)
 7: HPFS/NTFS
82: Linux Swap
83: Linux
8e: Linux LVM
fd: Linux RAID Auto
```

CLI Example:

```
salt '*' partition.get_id /dev/sda 1
```

`salt.modules.parted_partition.`**`list_`**(*device*, *unit=None*)

Prints partition information of given <device>

CLI Examples:

```
salt '*' partition.list /dev/sda
salt '*' partition.list /dev/sda unit=s
salt '*' partition.list /dev/sda unit=kB
```

`salt.modules.parted_partition.`**`mkfs`**(*device*, *fs_type*)

Makes a file system <fs_type> on partition <device>, destroying all data that resides on that partition. <fs_type> must be one of ”ext2”, ”fat32”, ”fat16”, ”linux-swap” or ”reiserfs” (if libreiserfs is installed)

CLI Example:

```
salt '*' partition.mkfs /dev/sda2 fat32
```

`salt.modules.parted_partition.`**`mklabel`**(*device*, *label_type*)

Create a new disklabel (partition table) of label_type.

Type should be one of ”aix”, ”amiga”, ”bsd”, ”dvh”, ”gpt”, ”loop”, ”mac”, ”msdos”, ”pc98”, or ”sun”.

CLI Example:

```
salt '*' partition.mklabel /dev/sda msdos
```

`salt.modules.parted_partition.`**`mkpart`**(*device*, *part_type*, *fs_type=None*, *start=None*, *end=None*)

Make a part_type partition for filesystem fs_type, beginning at start and ending at end (by default in megabytes). part_type should be one of ”primary”, ”logical”, or ”extended”.

CLI Examples:

```
salt '*' partition.mkpart /dev/sda primary fs_type=fat32 start=0 end=639
salt '*' partition.mkpart /dev/sda primary start=0 end=639
```

`salt.modules.parted_partition.`**`mkpartfs`**(*device*, *part_type*, *fs_type*, *start*, *end*)

Make a <part_type> partition with a new filesystem of <fs_type>, beginning at <start> and ending at <end> (by default in megabytes).

<part_type> should be one of ”primary”, ”logical”, or ”extended”. <fs_type> must be one of ”ext2”, ”fat32”, ”fat16”, ”linux-swap” or ”reiserfs” (if libreiserfs is installed)

CLI Example:

```
salt '*' partition.mkpartfs /dev/sda logical ext2 440 670
```

`salt.modules.parted_partition.`**`name`**(*device*, *partition*, *name*)

Set the name of partition to name. This option works only on Mac, PC98, and GPT disklabels. The name can be placed in quotes, if necessary.

CLI Example:

```
salt '*' partition.name /dev/sda 1 'My Documents'
```

`salt.modules.parted_partition.`**`probe`**(**devices*)

Ask the kernel to update its local partition data. When no args are specified all block devices are tried.

Caution: Generally only works on devices with no mounted partitions and may take a long time to return if specified devices are in use.

CLI Examples:

```
salt '*' partition.probe
salt '*' partition.probe /dev/sda
salt '*' partition.probe /dev/sda /dev/sdb
```

`salt.modules.parted_partition.`**`rescue`**(*device*, *start*, *end*)

Rescue a lost partition that was located somewhere between start and end. If a partition is found, parted will ask if you want to create an entry for it in the partition table.

CLI Example:

```
salt '*' partition.rescue /dev/sda 0 8056
```

`salt.modules.parted_partition.`**`resize`**(*device*, *minor*, *start*, *end*)

Resizes the partition with number <minor>.

The partition will start <start> from the beginning of the disk, and end <end> from the beginning of the disk. resize never changes the minor number. Extended partitions can be resized, so long as the new extended partition completely contains all logical partitions.

CLI Example:

```
salt '*' partition.resize /dev/sda 3 200 850
```

`salt.modules.parted_partition.`**`rm`**(*device*, *minor*)

Removes the partition with number <minor>.

CLI Example:

```
salt '*' partition.rm /dev/sda 5
```

`salt.modules.parted_partition.`**`set_`**(*device*, *minor*, *flag*, *state*)

Changes a flag on the partition with number <minor>.

A flag can be either ”on” or ”off” (make sure to use proper quoting, see *YAML Idiosyncrasies*). Some or all of these flags will be available, depending on what disk label you are using.

Valid flags are:

- boot
- root
- swap
- hidden
- raid

- lvm
- lba
- hp-service
- palo
- prep
- msftres
- bios_grub
- atvrecv
- diag
- legacy_boot
- msftdata
- irst
- esp
- type

CLI Example:

```
salt '*' partition.set /dev/sda 1 boot '"on"'
```

`salt.modules.parted_partition.`**`set_id`**`(`*device, minor, system_id*`)`

Sets the system ID for the partition. Some typical values are:

```
 b: FAT32 (vfat)
 7: HPFS/NTFS
82: Linux Swap
83: Linux
8e: Linux LVM
fd: Linux RAID Auto
```

CLI Example:

```
salt '*' partition.set_id /dev/sda 1 83
```

`salt.modules.parted_partition.`**`system_types`**`()`

List the system types that are supported by the installed version of sfdisk

CLI Example:

```
salt '*' partition.system_types
```

`salt.modules.parted_partition.`**`toggle`**`(`*device, partition, flag*`)`

Toggle the state of <flag> on <partition>. Valid flags are the same as the set command.

CLI Example:

```
salt '*' partition.toggle /dev/sda 1 boot
```

21.9.314 salt.modules.pcs module

Configure a Pacemaker/Corosync cluster with PCS

Configure Pacemaker/Cororsync clusters with the Pacemaker/Cororsync conifguration system (PCS)

depends pcs

New in version 2016.3.0.

`salt.modules.pcs.`**`auth`**(*nodes, pcsuser='hacluster', pcspasswd='hacluster', extra_args=None*)

Authorize nodes to the cluster

nodes a list of nodes which should be authorized to the cluster

pcsuser user for communitcation with PCS (default: hacluster)

pcspasswd password for pcsuser (default: hacluster)

extra_args list of extra option for the 'pcs cluster auth' command

CLI Example:

```
salt '*' pcs.auth nodes='[ node1.example.org node2.example.org ]'
↪pcsuser=hacluster pcspasswd=hoonetorg extra_args="[ '--force' ]"
```

`salt.modules.pcs.`**`cib_create`**(*cibfile, scope='configuration', extra_args=None*)

Create a CIB-file from the current CIB of the cluster

cibfile name/path of the file containing the CIB

scope specific section of the CIB (default: configuration)

extra_args additional options for creating the CIB-file

CLI Example:

```
salt '*' pcs.cib_create cibfile='/tmp/VIP_apache_1.cib' scope=False
```

`salt.modules.pcs.`**`cib_push`**(*cibfile, scope='configuration', extra_args=None*)

Push a CIB-file as the new CIB to the cluster

cibfile name/path of the file containing the CIB

scope specific section of the CIB (default: configuration)

extra_args additional options for creating the CIB-file

CLI Example:

```
salt '*' pcs.cib_push cibfile='/tmp/VIP_apache_1.cib' scope=False
```

`salt.modules.pcs.`**`cluster_node_add`**(*node, extra_args=None*)

Add a node to the pacemaker cluster via pcs command

node node that should be added

extra_args list of extra option for the 'pcs cluster node add' command

CLI Example:

```
salt '*' pcs.cluster_node_add node=node2.example.org
```

`salt.modules.pcs.`**`cluster_setup`**(*nodes, pcsclustername='pcscluster', extra_args=None*)

Setup pacemaker cluster via pcs command

nodes a list of nodes which should be set up

pcsclustername Name of the Pacemaker cluster (default: pcscluster)

extra_args list of extra option for the 'pcs cluster setup' command

CLI Example:

```
salt '*' pcs.cluster_setup nodes='[ node1.example.org node2.example.org ]'
↪pcsclustername=pcscluster
```

`salt.modules.pcs.`**`config_show`**(*cibfile=None*)

Show config of cluster

cibfile name/path of the file containing the CIB

CLI Example:

```
salt '*' pcs.config_show cibfile='/tmp/cib_for_galera'
```

`salt.modules.pcs.`**`is_auth`**(*nodes*)

Check if nodes are already authorized

nodes a list of nodes to be checked for authorization to the cluster

CLI Example:

```
salt '*' pcs.is_auth nodes='[node1.example.org node2.example.org]'
```

salt.modules.pcs.**item_create**(*item*, *item_id*, *item_type*, *create='create'*, *extra_args=None*, *cibfile=None*)

Create an item via pcs command (mainly for use with the pcs state module)

item config, property, resource, constraint etc.

item_id id of the item

item_type item type

create create command (create or set f.e., default: create)

extra_args additional options for the pcs command

cibfile use cibfile instead of the live CIB

salt.modules.pcs.**item_show**(*item*, *item_id=None*, *item_type=None*, *show='show'*, *extra_args=None*, *cibfile=None*)

Show an item via pcs command (mainly for use with the pcs state module)

item config, property, resource, constraint etc.

item_id id of the item

item_type item type

show show command (probably None, default: show)

extra_args additional options for the pcs command

cibfile use cibfile instead of the live CIB

salt.modules.pcs.**prop_set**(*prop*, *value*, *extra_args=None*, *cibfile=None*)

Set the value of a cluster property

prop name of the property

value value of the property prop

extra_args additional options for the pcs property command

cibfile use cibfile instead of the live CIB

CLI Example:

```
salt '*' pcs.prop_set prop='no-quorum-policy' value='ignore' cibfile='/tmp/2_node_
↪cluster.cib'
```

salt.modules.pcs.**prop_show**(*prop*, *extra_args=None*, *cibfile=None*)

Show the value of a cluster property

prop name of the property

extra_args additional options for the pcs property command

cibfile use cibfile instead of the live CIB

CLI Example:

```
salt '*' pcs.prop_show cibfile='/tmp/2_node_cluster.cib' prop='no-quorum-policy'
↪cibfile='/tmp/2_node_cluster.cib'
```

salt.modules.pcs.**resource_create**(*resource_id*, *resource_type*, *resource_options=None*, *cibfile=None*)

Create a resource via pcs command

resource_id name for the resource

resource_type resource type (f.e. ocf:heartbeat:IPaddr2 or VirtualIP)

resource_options additional options for creating the resource

cibfile use cibfile instead of the live CIB for manipulation

CLI Example:

```
salt '*' pcs.resource_create resource_id='galera' resource_type=
↪'ocf:heartbeat:galera' resource_options="['wsrep_cluster_address=gcomm://node1.
↪example.org,node2.example.org,node3.example.org', '--master']" cibfile='/tmp/
↪cib_for_galera.cib'
```

salt.modules.pcs.**resource_show**(*resource_id*, *extra_args=None*, *cibfile=None*)

Show a resource via pcs command

resource_id name of the resource

extra_args additional options for the pcs command

cibfile use cibfile instead of the live CIB

CLI Example:

```
salt '*' pcs.resource_show resource_id='galera' cibfile='/tmp/cib_for_galera.cib'
```

salt.modules.pcs.**stonith_create**(*stonith_id*, *stonith_device_type*, *stonith_device_options=None*, *cibfile=None*)

Create a stonith resource via pcs command

stonith_id name for the stonith resource

stonith_device_type name of the stonith agent fence_eps, fence_xvm f.e.

stonith_device_options additional options for creating the stonith resource

cibfile use cibfile instead of the live CIB for manipulation

CLI Example:

```
salt '*' pcs.stonith_create stonith_id='eps_fence' stonith_device_type='fence_eps'
                            stonith_device_options="['pcmk_host_map=node1.example.
↪org:01;node2.example.org:02', 'ipaddr=myepsdevice.example.org', 'action=reboot',
↪ 'power_wait=5', 'verbose=1', 'debug=/var/log/pcsd/eps_fence.log', 'login=hidden
↪', 'passwd=hoonetorg']" cibfile='/tmp/cib_for_stonith.cib'
```

salt.modules.pcs.**stonith_show**(*stonith_id*, *extra_args=None*, *cibfile=None*)

Show the value of a cluster stonith

stonith_id name for the stonith resource

extra_args additional options for the pcs stonith command

cibfile use cibfile instead of the live CIB

CLI Example:

```
salt '*' pcs.stonith_show stonith_id='eps_fence' cibfile='/tmp/2_node_cluster.cib'
```

21.9.315 salt.modules.pdbedit

Manage accounts in Samba's passdb using pdbedit

maintainer Jorge Schrauwen <sjorge@blackdot.be>

maturity new

platform posix

New in version 2017.7.0.

salt.modules.pdbedit.**create**(*login*, *password*, *password_hashed=False*, *machine_account=False*)

Create user account

login [string] login name

password [string] password

password_hashed [boolean] set if password is a nt hash instead of plain text

machine_account [boolean] set to create a machine trust account instead

CLI Example:

```
salt '*' pdbedit.create zoe 9764951149F84E770889011E1DC4A927 nthash
salt '*' pdbedit.create river  1sw4ll0w3d4bug
```

`salt.modules.pdbedit.`**`delete`**`(`*login*`)`

Delete user account

login [string] login name

CLI Example:

```
salt '*' pdbedit.delete wash
```

`salt.modules.pdbedit.`**`generate_nt_hash`**`(`*password*`)`

Generate a NT HASH

CLI Example:

```
salt '*' pdbedit.generate_nt_hash my_passwd
```

`salt.modules.pdbedit.`**`get_user`**`(`*login, hashes=False*`)`

Get user account details

login [string] login name

hashes [boolean] include NTHASH and LMHASH in verbose output

CLI Example:

```
salt '*' pdbedit.get kaylee
```

`salt.modules.pdbedit.`**`list_users`**`(`*verbose=True, hashes=False*`)`

List user accounts

verbose [boolean] return all information

hashes [boolean] include NT HASH and LM HASH in verbose output

CLI Example:

```
salt '*' pdbedit.list
```

`salt.modules.pdbedit.`**`modify`**`(`*login, password=None, password_hashed=False, domain=None, profile=None, script=None, drive=None, homedir=None, fullname=None, account_desc=None, account_control=None, machine_sid=None, user_sid=None, reset_login_hours=False, reset_bad_password_count=False*`)`

Modify user account

login [string] login name

password [string] password

password_hashed [boolean] set if password is a nt hash instead of plain text

domain [string] users domain

profile [string] profile path

script [string] logon script

drive [string] home drive

homedir [string] home directory

fullname [string] full name

account_desc [string] account description

machine_sid [string] specify the machines new primary group SID or rid

user_sid [string] specify the users new primary group SID or rid

account_control [string] specify user account control properties

Note: Only the following can be set: - N: No password required - D: Account disabled - H: Home directory required - L: Automatic Locking - X: Password does not expire

reset_login_hours [boolean] reset the users allowed logon hours

reset_bad_password_count [boolean] reset the stored bad login counter

Note: if user is absent and password is provided, the user will be created

CLI Example:

```
salt '*' pdbedit.modify inara fullname='Inara Serra'
salt '*' pdbedit.modify simon password=r1v3r
salt '*' pdbedit.modify jane drive='V:' homedir='\\serenity\jane\profile'
salt '*' pdbedit.modify mal account_control=NX
```

21.9.316 salt.modules.pecl

Manage PHP pecl extensions.

`salt.modules.pecl.`**`install`**(*pecls*, *defaults=False*, *force=False*, *preferred_state='stable'*)

New in version 0.17.0.

Installs one or several pecl extensions.

pecls The pecl extensions to install.

defaults Use default answers for extensions such as pecl_http which ask questions before installation. Without this option, the pecl.installed state will hang indefinitely when trying to install these extensions.

force Whether to force the installed version or not

CLI Example:

```
salt '*' pecl.install fuse
```

`salt.modules.pecl.`**`list_`**(*channel=None*)

List installed pecl extensions.

CLI Example:

```
salt '*' pecl.list
```

`salt.modules.pecl.`**`uninstall`**(*pecls*)

Uninstall one or several pecl extensions.

pecls The pecl extensions to uninstall.

CLI Example:

```
salt '*' pecl.uninstall fuse
```

`salt.modules.pecl.`**`update`**(*pecls*)

Update one or several pecl extensions.

pecls The pecl extensions to update.

CLI Example:

```
salt '*' pecl.update fuse
```

21.9.317 salt.modules.peeringdb module

PeeringDB Module

New in version 2019.2.0.

Execution module for the basic interaction with the PeeringDB API.

While for GET operations (the functions prefixed by `get_`) the credentials are optional, there are some specific details that are visible only to authenticated users. Moreover, the credentials are required when adding or updating information. That means, the module can equally work out of the box without any further configuration with the limitations imposed by the PeeringDB API.

For complete API documentation, please refer to https://www.peeringdb.com/apidocs/.

Configuration (in the opts or Pillar):

```
peeringdb:
  username: salt
  password: 5@1t
```

`salt.modules.peeringdb.`**`get_fac`**`(`*`**kwargs`*`)`

Return the details of the facility identified using the search filters specified in the query.

Note: If no `id` or filter arguments are specified, it will return all the possible facilities registered in PeeringDB.

The available filters are documented at: https://www.peeringdb.com/apidocs/#!/netfac/netfac_list

CLI Example:

```
salt '*' peeringdb.get_fac id=1774
salt '*' peeringdb.get_fac state=UT
```

`salt.modules.peeringdb.`**`get_ix`**`(`*`**kwargs`*`)`

Return the details of an IX (Internet Exchange) using the search filters specified in the query.

Note: If no `id` or filter arguments are specified, it will return all the possible IXs registered in PeeringDB.

The available filters are documented at: https://www.peeringdb.com/apidocs/#!/ix/ix_list

CLI Example:

```
salt '*' peeringdb.get_ix id=1
salt '*' peeringdb.get_ix city='Milwaukee'
```

`salt.modules.peeringdb.`**`get_ixfac`**`(`*`**kwargs`*`)`

Return the details of an IX (Internet Exchange) facility using the search filters specified in the query.

Note: If no `id` or filter arguments are specified, it will return all the possible IX facilities registered in PeeringDB.

The available filters are documented at: https://www.peeringdb.com/apidocs/#!/ixfac/ixfac_list

CLI Example:

```
salt '*' peeringdb.get_ixfac id=1
salt '*' peeringdb.get_ixfac city='Milwaukee'
```

`salt.modules.peeringdb.`**`get_ixlan`**`(`***kwargs*`)`

Return the details of an IX (Internet Exchange) together with the networks available in this location (and their details), using the search filters specified in the query.

Note: If no `id` or filter arguments are specified, it will return all the possible IX LAN facilities registered in PeeringDB.

The available filters are documented at: https://www.peeringdb.com/apidocs/#!/ixlan/ixlan_list

CLI Example:

```
salt '*' peeringdb.get_ixlan id=780
salt '*' peeringdb.get_ixlan city='Milwaukee'
```

`salt.modules.peeringdb.`**`get_ixpfx`**`(`***kwargs*`)`

Return the details of an IX (Internet Exchange) together with the PeeringDB IDs of the networks available in this location, using the search filters specified in the query.

Note: If no `id` or filter arguments are specified, it will return all the possible IX LAN facilities registered in PeeringDB.

The available filters are documented at: https://www.peeringdb.com/apidocs/#!/ixpfx/ixpfx_list

CLI Example:

```
salt '*' peeringdb.get_ixpfx id=780
salt '*' peeringdb.get_ixpfx city='Milwaukee'
```

`salt.modules.peeringdb.`**`get_net`**`(`***kwargs*`)`

Return the details of a network identified using the search filters specified in the query.

Note: If no `id` or filter arguments are specified, it will return all the possible networks registered in PeeringDB.

The available filters are documented at: https://www.peeringdb.com/apidocs/#!/net/net_list

CLI Example:

```
salt '*' peeringdb.get_net id=4224
salt '*' peeringdb.get_net asn=13335
salt '*' peeringdb.get_net city='Salt Lake City'
salt '*' peeringdb.get_net name__startswith=GTT
```

`salt.modules.peeringdb.`**`get_netfac`**`(`***kwargs*`)`

Return the list of facilities used by a particular network, given the `id` or other filters specified in the query.

Note: If no `id` or filter arguments are specified, it will return all the possible network facilities registered in PeeringDB.

The available filters are documented at: https://www.peeringdb.com/apidocs/#!/netfac/netfac_list

CLI Example:

```
salt '*' peeringdb.get_netfac id=780
salt '*' peeringdb.get_netfac city='Milwaukee'
```

`salt.modules.peeringdb.`**`get_netixlan`**`(`***kwargs*`)`

Return the IP addresses used by a particular network at all the IXs where it is available. The network is selected either via the `id` argument or the other filters specified in the query.

Note: If no `id` or filter arguments are specified, it will return all the possible IP addresses, of all networks, at all IXs, registered in PeeringDB.

The available filters are documented at: https://www.peeringdb.com/apidocs/#!/netixlan/netixlan_list

---

CLI Example:

```
salt '*' peeringdb.get_netixlan asn=13335
salt '*' peeringdb.get_netixlan ipaddr4=185.1.114.25
```

`salt.modules.peeringdb.`**`get_org`**`(`***kwargs*`)`

Return the details of an organisation together with the networks available in this location, using the search filters specified in the query.

Note: If no `id` or filter arguments are specified, it will return all the possible organisations registered in PeeringDB.

The available filters are documented at: https://www.peeringdb.com/apidocs/#!/org/org_list

---

CLI Example:

```
salt '*' peeringdb.get_org id=2
salt '*' peeringdb.get_org city=Duesseldorf
```

`salt.modules.peeringdb.`**`get_poc`**`(`***kwargs*`)`

Return the details of a person of contact together using the search filters specified in the query.

Note: If no `id` or filter arguments are specified, it will return all the possible contacts registered in PeeringDB.

The available filters are documented at: https://www.peeringdb.com/apidocs/#!/poc/poc_list

---

CLI Example:

```
salt '*' peeringdb.get_poc id=6721
salt '*' peeringdb.get_poc email__contains='@cloudflare.com'
```

21.9.318 salt.modules.pf

Control the OpenBSD packet filter (PF).

codeauthor Jasper Lievisse Adriaanse <j@jasper.la>

New in version 2019.2.0.

`salt.modules.pf.`**`disable`**`()`

Disable the Packet Filter.

CLI example:

```
salt '*' pf.disable
```

`salt.modules.pf.`**`enable`**`()`

Enable the Packet Filter.

CLI example:

```
salt '*' pf.enable
```

`salt.modules.pf.`**`flush`**`(`*modifier*`)`

Flush the specified packet filter parameters.

modifier: Should be one of the following:

- all
- info
- osfp
- rules
- sources
- states
- tables

Please refer to the OpenBSD pfctl(8) documentation for a detailed explanation of each command.

CLI example:

```
salt '*' pf.flush states
```

`salt.modules.pf.`**`load`**`(`*file='/etc/pf.conf', noop=False*`)`

Load a ruleset from the specific file, overwriting the currently loaded ruleset.

file: Full path to the file containing the ruleset.

noop: Don't actually load the rules, just parse them.

CLI example:

```
salt '*' pf.load /etc/pf.conf.d/lockdown.conf
```

`salt.modules.pf.`**`loglevel`**`(`*level*`)`

Set the debug level which limits the severity of log messages printed by `pf(4)`.

level: Log level. Should be one of the following: emerg, alert, crit, err, warning, notice, info or debug.

CLI example:

```
salt '*' pf.loglevel emerg
```

`salt.modules.pf.`**`show`**`(`*modifier*`)`

Show filter parameters.

modifier: Modifier to apply for filtering. Only a useful subset of what pfctl supports can be used with Salt.

- rules
- states
- tables

CLI example:

```
salt '*' pf.show rules
```

`salt.modules.pf.`**`table`**`(`*command, table, **kwargs*`)`

Apply a command on the specified table.

table: Name of the table.

command: Command to apply to the table. Supported commands are:

- add

- delete
- expire
- flush
- kill
- replace
- show
- test
- zero

Please refer to the OpenBSD pfctl(8) documentation for a detailed explanation of each command.

CLI example:

```
salt '*' pf.table expire table=spam_hosts number=300
salt '*' pf.table add table=local_hosts addresses='["127.0.0.1", "::1"]'
```

21.9.319 salt.modules.philips_hue module

Philips HUE lamps module for proxy.

New in version 2015.8.3.

21.9.320 salt.modules.pillar

Extract the pillar data for this minion

salt.modules.pillar.**ext**(*external*, *pillar=None*)

Changed in version 2016.3.6,2016.11.3,2017.7.0: The supported ext_pillar types are now tunable using the *on_demand_ext_pillar* config option. Earlier releases used a hard-coded default.

Generate the pillar and apply an explicit external pillar

external A single ext_pillar to add to the ext_pillar configuration. This must be passed as a single section from the ext_pillar configuration (see CLI examples below). For more complicated `ext_pillar` configurations, it can be helpful to use the Python shell to load YAML configuration into a dictionary, and figure out

```
>>> import salt.utils.yaml
>>> ext_pillar = salt.utils.yaml.safe_load("""
... ext_pillar:
...   - git:
...     - issue38440 https://github.com/terminalmage/git_pillar:
...       - env: base
... """)
>>> ext_pillar
{'ext_pillar': [{'git': [{'mybranch https://github.com/myuser/myrepo': [{'env
↪': 'base'}]}]}]}
>>> ext_pillar['ext_pillar'][0]
{'git': [{'mybranch https://github.com/myuser/myrepo': [{'env': 'base'}]}]}
```

In the above example, the value to pass would be `{'git': [{'mybranch https://github.com/myuser/myrepo': [{'env': 'base'}]}]}`. Note that this would need to be quoted when passing on the CLI (as in the CLI examples below).

pillar [None] If specified, allows for a dictionary of pillar data to be made available to pillar and ext_pillar rendering. These pillar variables will also override any variables of the same name in pillar or ext_pillar.

New in version 2015.5.0.

CLI Examples:

```
salt '*' pillar.ext '{libvirt: _}'
salt '*' pillar.ext "{'git': ['master https://github.com/myuser/myrepo']}"
salt '*' pillar.ext "{'git': [{'mybranch https://github.com/myuser/myrepo': [{'env
↪': 'base'}]}]}"
```

`salt.modules.pillar.`**`fetch`**(*key*, *default=<class 'KeyError'>*, *merge=False*, *merge_nested_lists=None*, *delimiter=':'*, *pillarenv=None*, *saltenv=None*)

New in version 0.14.

Attempt to retrieve the named value from *in-memory pillar data*. If the pillar key is not present in the in-memory pillar, then the value specified in the `default` option (described below) will be returned.

If the merge parameter is set to `True`, the default will be recursively merged into the returned pillar data.

The value can also represent a value in a nested dict using a ":" delimiter for the dict. This means that if a dict in pillar looks like this:

```
{'pkg': {'apache': 'httpd'}}
```

To retrieve the value associated with the `apache` key in the `pkg` dict this key can be passed as:

```
pkg:apache
```

key The pillar key to get value from

default The value specified by this option will be returned if the desired pillar key does not exist.

If a default value is specified, then it will be an empty string, unless *`pillar_raise_on_missing`* is set to `True`, in which case an error will be raised.

merge `[False]` If `True`, the retrieved values will be merged into the passed default. When the default and the retrieved value are both dictionaries, the dictionaries will be recursively merged.

New in version 2014.7.0.

Changed in version 2016.3.7,2016.11.4,2017.7.0: If the default and the retrieved value are not of the same type, then merging will be skipped and the retrieved value will be returned. Earlier releases raised an error in these cases.

merge_nested_lists If set to `False`, lists nested within the retrieved pillar dictionary will *overwrite* lists in `default`. If set to `True`, nested lists will be *merged* into lists in `default`. If unspecified (the default), this option is inherited from the `pillar_merge_lists` minion config option.

Note: This option is ignored when `merge` is set to `False`.

New in version 2016.11.6.

delimiter Specify an alternate delimiter to use when traversing a nested dict. This is useful for when the desired key contains a colon. See CLI example below for usage.

New in version 2014.7.0.

pillarenv If specified, this function will query the master to generate fresh pillar data on the fly, specifically from the requested pillar environment. Note that this can produce different pillar data than executing this function without an environment, as its normal behavior is just to return a value from minion's pillar data in memory (which can be sourced from more than one pillar environment).

Using this argument will not affect the pillar data in memory. It will however be slightly slower and use more resources on the master due to the need for the master to generate and send the minion fresh pillar data. This tradeoff in performance however allows for the use case where pillar data is desired only from a single environment.

New in version 2017.7.0.

saltenv Included only for compatibility with *pillarenv_from_saltenv*, and is otherwise ignored.

New in version 2017.7.0.

CLI Example:

```
salt '*' pillar.get pkg:apache
salt '*' pillar.get abc::def|ghi delimiter='|'
```

`salt.modules.pillar.`**`file_exists`**(*path*, *saltenv=None*)

New in version 2016.3.0.

This is a master-only function. Calling from the minion is not supported.

Use the given path and search relative to the pillar environments to see if a file exists at that path.

If the `saltenv` argument is given, restrict search to that environment only.

Will only work with `pillar_roots`, not external pillars.

Returns True if the file is found, and False otherwise.
path The path to the file in question. Will be treated as a relative path
saltenv Optional argument to restrict the search to a specific saltenv
CLI Example:

```
salt '*' pillar.file_exists foo/bar.sls
```

`salt.modules.pillar.`**`filter_by`**(*lookup_dict*, *pillar*, *merge=None*, *default='default'*, *base=None*)

New in version 2017.7.0.

Look up the given pillar in a given dictionary and return the result

Parameters

- **`lookup_dict`** – A dictionary, keyed by a pillar, containing a value or values relevant to systems matching that pillar. For example, a key could be a pillar for a role and the value could the name of a package on that particular OS.

  The dictionary key can be a globbing pattern. The function will return the corresponding `lookup_dict` value where the pillar value matches the pattern. For example:

  ```
  # this will render 'got some salt' if ``role`` begins with
  ↪'salt'
  salt '*' pillar.filter_by '{salt*: got some salt, default:
  ↪salt is not here}' role
  ```

- **`pillar`** – The name of a pillar to match with the system's pillar. For example, the value of the "role" pillar could be used to pull values from the `lookup_dict` dictionary.

  The pillar value can be a list. The function will return the `lookup_dict` value for a first found item in the list matching one of the `lookup_dict` keys.
- **`merge`** – A dictionary to merge with the results of the pillar selection from `lookup_dict`. This allows another dictionary to override the values in the `lookup_dict`.
- **`default`** – default lookup_dict's key used if the pillar does not exist or if the pillar value has no match on lookup_dict. If unspecified the value is "default".
- **`base`** – A lookup_dict key to use for a base dictionary. The pillar-selected `lookup_dict` is merged over this and then finally the `merge` dictionary is merged. This allows common values for each case to be collected in the base and overridden by the pillar selection dictionary and the merge dictionary. Default is unset.

CLI Example:

```
salt '*' pillar.filter_by '{web: Serve it up, db: I query, default: x_x}' role
```

`salt.modules.pillar.`**`get`**(*key*, *default=<class 'KeyError'>*, *merge=False*, *merge_nested_lists=None*, *delimiter=':'*, *pillarenv=None*, *saltenv=None*)

New in version 0.14.

Attempt to retrieve the named value from *in-memory pillar data*. If the pillar key is not present in the in-memory pillar, then the value specified in the `default` option (described below) will be returned.

If the merge parameter is set to `True`, the default will be recursively merged into the returned pillar data.

The value can also represent a value in a nested dict using a ":" delimiter for the dict. This means that if a dict in pillar looks like this:

```
{'pkg': {'apache': 'httpd'}}
```

To retrieve the value associated with the `apache` key in the `pkg` dict this key can be passed as:

```
pkg:apache
```

key The pillar key to get value from

default The value specified by this option will be returned if the desired pillar key does not exist.

If a default value is specified, then it will be an empty string, unless *`pillar_raise_on_missing`* is set to `True`, in which case an error will be raised.

merge `[False]` If `True`, the retrieved values will be merged into the passed default. When the default and the retrieved value are both dictionaries, the dictionaries will be recursively merged.

New in version 2014.7.0.

Changed in version 2016.3.7,2016.11.4,2017.7.0: If the default and the retrieved value are not of the same type, then merging will be skipped and the retrieved value will be returned. Earlier releases raised an error in these cases.

merge_nested_lists If set to `False`, lists nested within the retrieved pillar dictionary will *overwrite* lists in `default`. If set to `True`, nested lists will be *merged* into lists in `default`. If unspecified (the default), this option is inherited from the `pillar_merge_lists` minion config option.

Note: This option is ignored when `merge` is set to `False`.

New in version 2016.11.6.

delimiter Specify an alternate delimiter to use when traversing a nested dict. This is useful for when the desired key contains a colon. See CLI example below for usage.

New in version 2014.7.0.

pillarenv If specified, this function will query the master to generate fresh pillar data on the fly, specifically from the requested pillar environment. Note that this can produce different pillar data than executing this function without an environment, as its normal behavior is just to return a value from minion's pillar data in memory (which can be sourced from more than one pillar environment).

Using this argument will not affect the pillar data in memory. It will however be slightly slower and use more resources on the master due to the need for the master to generate and send the minion fresh pillar data. This tradeoff in performance however allows for the use case where pillar data is desired only from a single environment.

New in version 2017.7.0.

saltenv Included only for compatibility with *`pillarenv_from_saltenv`*, and is otherwise ignored.

New in version 2017.7.0.

CLI Example:

```
salt '*' pillar.get pkg:apache
salt '*' pillar.get abc::def|ghi delimiter='|'
```

`salt.modules.pillar.`**`item`**`(`**args*, ***kwargs*`)`

New in version 0.16.2.

Return one or more pillar entries from the *in-memory pillar data*.

delimiter Delimiter used to traverse nested dictionaries.

> **Note:** This is different from *`pillar.get`* in that no default value can be specified. *`pillar.get`* should probably still be used in most cases to retrieve nested pillar values, as it is a bit more flexible. One reason to use this function instead of *`pillar.get`* however is when it is desirable to retrieve the values of more than one key, since *`pillar.get`* can only retrieve one key at a time.

New in version 2015.8.0.

pillarenv If specified, this function will query the master to generate fresh pillar data on the fly, specifically from the requested pillar environment. Note that this can produce different pillar data than executing this function without an environment, as its normal behavior is just to return a value from minion's pillar data in memory (which can be sourced from more than one pillar environment).

Using this argument will not affect the pillar data in memory. It will however be slightly slower and use more resources on the master due to the need for the master to generate and send the minion fresh pillar data. This tradeoff in performance however allows for the use case where pillar data is desired only from a single environment.

New in version 2017.7.6,2018.3.1.

saltenv Included only for compatibility with *`pillarenv_from_saltenv`*, and is otherwise ignored.

New in version 2017.7.6,2018.3.1.

CLI Examples:

```
salt '*' pillar.item foo
salt '*' pillar.item foo:bar
salt '*' pillar.item foo bar baz
```

`salt.modules.pillar.`**`items`**`(`**args*, ***kwargs*`)`

Calls the master for a fresh pillar and generates the pillar data on the fly

Contrast with *`raw()`* which returns the pillar data that is currently loaded into the minion.

pillar If specified, allows for a dictionary of pillar data to be made available to pillar and ext_pillar rendering. these pillar variables will also override any variables of the same name in pillar or ext_pillar.

New in version 2015.5.0.

pillar_enc If specified, the data passed in the `pillar` argument will be passed through this renderer to decrypt it.

> **Note:** This will decrypt on the minion side, so the specified renderer must be set up on the minion for this to work. Alternatively, pillar data can be decrypted master-side. For more information, see the *Pillar Encryption* documentation. Pillar data that is decrypted master-side, is not decrypted until the end of pillar compilation though, so minion-side decryption will be necessary if the encrypted pillar data must be made available in an decrypted state pillar/ext_pillar rendering.

New in version 2017.7.0.

pillarenv Pass a specific pillar environment from which to compile pillar data. If not specified, then the minion's *pillarenv* option is not used, and if that also is not specified then all configured pillar environments will be merged into a single pillar dictionary and returned.

New in version 2016.11.2.

saltenv Included only for compatibility with *pillarenv_from_saltenv*, and is otherwise ignored.

CLI Example:

```
salt '*' pillar.items
```

`salt.modules.pillar.`**`keys`**`(`*key, delimiter=':'*`)`

New in version 2015.8.0.

Attempt to retrieve a list of keys from the named value from the pillar.

The value can also represent a value in a nested dict using a ":" delimiter for the dict, similar to how pillar.get works.

delimiter Specify an alternate delimiter to use when traversing a nested dict

CLI Example:

```
salt '*' pillar.keys web:sites
```

`salt.modules.pillar.`**`ls`**`(`**args*`)`

New in version 2015.8.0.

Calls the master for a fresh pillar, generates the pillar data on the fly (same as `items()`), but only shows the available main keys.

CLI Examples:

```
salt '*' pillar.ls
```

`salt.modules.pillar.`**`obfuscate`**`(`**args*`)`

New in version 2015.8.0.

Same as `items()`, but replace pillar values with a simple type indication.

This is useful to avoid displaying sensitive information on console or flooding the console with long output, such as certificates. For many debug or control purposes, the stakes lie more in dispatching than in actual values.

In case the value is itself a collection type, obfuscation occurs within the value. For mapping types, keys are not obfuscated. Here are some examples:

- `'secret password'` becomes `'<str>'`
- `['secret', 1]` becomes `['<str>', '<int>']`
- `{'login': 'somelogin', 'pwd': 'secret'}` becomes `{'login': '<str>', 'pwd': '<str>'}`

CLI Examples:

```
salt '*' pillar.obfuscate
```

`salt.modules.pillar.`**`raw`**`(`*key=None*`)`

Return the raw pillar data that is currently loaded into the minion.

Contrast with `items()` which calls the master to fetch the most up-to-date Pillar.

CLI Example:

```
salt '*' pillar.raw
```

With the optional key argument, you can select a subtree of the pillar raw data.:

```
salt '*' pillar.raw key='roles'
```

21.9.321 salt.modules.pip

Install Python packages with pip to either the system or a virtualenv

Windows Support

New in version 2014.7.4.

Salt now uses a portable python. As a result the entire pip module is now functional on the salt installation itself. You can pip install dependencies for your custom modules. You can even upgrade salt itself using pip. For this to work properly, you must specify the Current Working Directory (`cwd`) and the Pip Binary (`bin_env`) salt should use. The variable `pip_bin` can be either a virtualenv path or the path to the pip binary itself.

For example, the following command will list all software installed using pip to your current salt environment:

```
salt <minion> pip.list cwd='C:\salt\bin\Scripts' bin_env='C:\salt\bin\Scripts\pip.exe'
```

Specifying the `cwd` and `bin_env` options ensures you're modifying the salt environment. If these are omitted, it will default to the local installation of python. If python is not installed locally it will fail saying it couldn't find pip.

State File Support

This functionality works in states as well. If you need to pip install colorama with a state, for example, the following will work:

```
install_colorama:
  pip.installed:
    - name: colorama
    - cwd: 'C:\salt\bin\scripts'
    - bin_env: 'C:\salt\bin\scripts\pip.exe'
    - upgrade: True
```

Upgrading Salt using Pip

You can now update salt using pip to any version from the 2014.7 branch forward. Previous version require recompiling some of the dependencies which is painful in windows.

To do this you just use pip with git to update to the version you want and then restart the service. Here is a sample state file that upgrades salt to the head of the 2015.5 branch:

```
install_salt:
  pip.installed:
    - cwd: 'C:\salt\bin\scripts'
    - bin_env: 'C:\salt\bin\scripts\pip.exe'
    - editable: git+https://github.com/saltstack/salt@2015.5#egg=salt
    - upgrade: True

restart_service:
```

(continues on next page)

(continued from previous page)

```
service.running:
  - name: salt-minion
  - enable: True
  - watch:
    - pip: install_salt
```

Note: If you're having problems, you might try doubling the back slashes. For example, cwd: 'C:\salt\bin\scripts'. Sometimes python thinks the single back slash is an escape character.

There is a known incompatibility between Python2 pip>=10.* and Salt <=2018.3.0. The issue is decribed here: https://github.com/saltstack/salt/issues/46163

`salt.modules.pip.`**`freeze`**(*bin_env=None, user=None, cwd=None, use_vt=False, env_vars=None, **kwargs*)

Return a list of installed packages either globally or in the specified virtualenv

bin_env Path to pip (or to a virtualenv). This can be used to specify the path to the pip to use when more than one Python release is installed (e.g. `/usr/bin/pip-2.7` or `/usr/bin/pip-2.6`. If a directory path is specified, it is assumed to be a virtualenv.

user The user under which to run pip

cwd Directory from which to run pip

Note: If the version of pip available is older than 8.0.3, the list will not include the packages `pip`, `wheel`, `setuptools`, or `distribute` even if they are installed.

CLI Example:

```
salt '*' pip.freeze bin_env=/home/code/path/to/virtualenv
```

`salt.modules.pip.`**`install`**(*pkgs=None, requirements=None, bin_env=None, use_wheel=False, no_use_wheel=False, log=None, proxy=None, timeout=None, editable=None, find_links=None, index_url=None, extra_index_url=None, no_index=False, mirrors=None, build=None, target=None, download=None, download_cache=None, source=None, upgrade=False, force_reinstall=False, ignore_installed=False, exists_action=None, no_deps=False, no_install=False, no_download=False, global_options=None, install_options=None, user=None, cwd=None, pre_releases=False, cert=None, allow_all_external=False, allow_external=None, allow_unverified=None, process_dependency_links=False, saltenv='base', env_vars=None, use_vt=False, trusted_host=None, no_cache_dir=False, extra_args=None, cache_dir=None, no_binary=None, disable_version_check=False, **kwargs*)

Install packages with pip

Install packages individually or from a pip requirements file. Install packages globally or to a virtualenv.

pkgs Comma separated list of packages to install

requirements Path to requirements

bin_env Path to pip (or to a virtualenv). This can be used to specify the path to the pip to use when more than one Python release is installed (e.g. `/usr/bin/pip-2.7` or `/usr/bin/pip-2.6`. If a directory path is specified, it is assumed to be a virtualenv.

Note: For Windows, if the pip module is being used to upgrade the pip package, bin_env should be the path to the virtualenv or to the python binary that should be used. The pip command is unable to upgrade itself in Windows.

---

use_wheel Prefer wheel archives (requires pip>=1.4)

no_use_wheel Force to not use wheel archives (requires pip>=1.4,<10.0.0)

no_binary Force to not use binary packages (requires pip >= 7.0.0) Accepts either :all: to disable all binary packages, :none: to empty the set, or one or more package names with commas between them

log Log file where a complete (maximum verbosity) record will be kept

proxy Specify a proxy in the form `user:passwd@proxy.server:port`. Note that the `user:password@` is optional and required only if you are behind an authenticated proxy. If you provide `user@proxy.server:port` then you will be prompted for a password.

Note: If the the Minion has a globaly configured proxy - it will be used even if no proxy was set here. To explicitly disable proxy for pip you should pass `False` as a value.

timeout Set the socket timeout (default 15 seconds)

editable install something editable (e.g. `git+https://github.com/worldcompany/djangoembed.git#egg=djangoembed`)

find_links URL to search for packages

index_url Base URL of Python Package Index

extra_index_url Extra URLs of package indexes to use in addition to `index_url`

no_index Ignore package index

mirrors Specific mirror URL(s) to query (automatically adds –use-mirrors)

Warning: This option has been deprecated and removed in pip version 7.0.0. Please use `index_url` and/or `extra_index_url` instead.

build Unpack packages into `build` dir

target Install packages into `target` dir

download Download packages into `download` instead of installing them

download_cache | cache_dir Cache downloaded packages in `download_cache` or `cache_dir` dir

source Check out `editable` packages into `source` dir

upgrade Upgrade all packages to the newest available version

force_reinstall When upgrading, reinstall all packages even if they are already up-to-date.

ignore_installed Ignore the installed packages (reinstalling instead)

exists_action Default action when a path already exists: (s)witch, (i)gnore, (w)ipe, (b)ackup

no_deps Ignore package dependencies

no_install Download and unpack all packages, but don't actually install them

no_download Don't download any packages, just install the ones already downloaded (completes an install run with `--no-install`)

install_options Extra arguments to be supplied to the setup.py install command (e.g. like `--install-option='--install-scripts=/usr/local/bin'`). Use multiple –install-option options to pass multiple options to setup.py install. If you are using an option with a directory path, be sure to use absolute path.

global_options Extra global options to be supplied to the setup.py call before the install command.

user The user under which to run pip

cwd Directory from which to run pip

pre_releases Include pre-releases in the available versions

cert Provide a path to an alternate CA bundle

allow_all_external Allow the installation of all externally hosted files

allow_external Allow the installation of externally hosted files (comma separated list)

allow_unverified Allow the installation of insecure and unverifiable files (comma separated list)

process_dependency_links Enable the processing of dependency links

env_vars Set environment variables that some builds will depend on. For example, a Python C-module may have a Makefile that needs INCLUDE_PATH set to pick up a header file while compiling. This must be in the form of a dictionary or a mapping.

Example:

```
salt '*' pip.install django_app env_vars="{'CUSTOM_PATH': '/opt/django_app'}"
```

trusted_host Mark this host as trusted, even though it does not have valid or any HTTPS.
use_vt Use VT terminal emulation (see output while installing)
no_cache_dir Disable the cache.
extra_args pip keyword and positional arguments not yet implemented in salt

```
salt '*' pip.install pandas extra_args="[{'--latest-pip-kwarg':'param'}, '--
↪latest-pip-arg']"
```

Warning: If unsupported options are passed here that are not supported in a minion's version of pip, a *No such option error* will be thrown.

Will be translated into the following pip command:

```
pip install pandas --latest-pip-kwarg param --latest-pip-arg
```

disable_version_check Pip may periodically check PyPI to determine whether a new version of pip is available to download. Passing True for this option disables that check.

CLI Example:

```
salt '*' pip.install <package name>,<package2 name>
salt '*' pip.install requirements=/path/to/requirements.txt
salt '*' pip.install <package name> bin_env=/path/to/virtualenv
salt '*' pip.install <package name> bin_env=/path/to/pip_bin
```

Complicated CLI example:

```
salt '*' pip.install markdown,django                editable=git+https://github.
↪com/worldcompany/djangoembed.git#egg=djangoembed upgrade=True no_deps=True
```

`salt.modules.pip.`**`is_installed`**(*pkgname=None*, *bin_env=None*, *user=None*, *cwd=None*)

New in version 2018.3.0.

Filter list of installed apps from `freeze` and return True or False if `pkgname` exists in the list of packages installed.

Note: If the version of pip available is older than 8.0.3, the packages wheel, setuptools, and distribute will not be reported by this function even if they are installed. Unlike `pip.freeze`, this function always reports the version of pip which is installed.

CLI Example:

```
salt '*' pip.is_installed salt
```

`salt.modules.pip.`**`list_`**(*prefix=None*, *bin_env=None*, *user=None*, *cwd=None*, *env_vars=None*, ***kwargs*)

Filter list of installed apps from `freeze` and check to see if `prefix` exists in the list of packages installed.

Note: If the version of pip available is older than 8.0.3, the packages `wheel`, `setuptools`, and `distribute` will not be reported by this function even if they are installed. Unlike *pip.freeze*, this function always reports the version of pip which is installed.

CLI Example:

```
salt '*' pip.list salt
```

salt.modules.pip.**list_all_versions**(*pkg*, *bin_env=None*, *include_alpha=False*, *include_beta=False*, *include_rc=False*, *user=None*, *cwd=None*, *index_url=None*, *extra_index_url=None*)

New in version 2017.7.3.

List all available versions of a pip package

pkg The package to check

bin_env Path to pip (or to a virtualenv). This can be used to specify the path to the pip to use when more than one Python release is installed (e.g. `/usr/bin/pip-2.7` or `/usr/bin/pip-2.6`. If a directory path is specified, it is assumed to be a virtualenv.

include_alpha Include alpha versions in the list

include_beta Include beta versions in the list

include_rc Include release candidates versions in the list

user The user under which to run pip

cwd Directory from which to run pip

index_url Base URL of Python Package Index .. versionadded:: 2019.2.0

extra_index_url Additional URL of Python Package Index .. versionadded:: 2019.2.0

CLI Example:

```
salt '*' pip.list_all_versions <package name>
```

salt.modules.pip.**list_upgrades**(*bin_env=None*, *user=None*, *cwd=None*)

Check whether or not an upgrade is available for all packages

CLI Example:

```
salt '*' pip.list_upgrades
```

salt.modules.pip.**uninstall**(*pkgs=None*, *requirements=None*, *bin_env=None*, *log=None*, *proxy=None*, *timeout=None*, *user=None*, *cwd=None*, *saltenv='base'*, *use_vt=False*)

Uninstall packages individually or from a pip requirements file

pkgs comma separated list of packages to install

requirements Path to requirements file

bin_env Path to pip (or to a virtualenv). This can be used to specify the path to the pip to use when more than one Python release is installed (e.g. `/usr/bin/pip-2.7` or `/usr/bin/pip-2.6`. If a directory path is specified, it is assumed to be a virtualenv.

log Log file where a complete (maximum verbosity) record will be kept

proxy Specify a proxy in the format `user:passwd@proxy.server:port`. Note that the `user:password@` is optional and required only if you are behind an authenticated proxy. If you provide `user@proxy.server:port` then you will be prompted for a password.

Note: If the the Minion has a globaly configured proxy - it will be used even if no proxy was set here. To explicitly disable proxy for pip you should pass `False` as a value.

---

timeout Set the socket timeout (default 15 seconds)

user The user under which to run pip

cwd Directory from which to run pip

use_vt Use VT terminal emulation (see output while installing)

CLI Example:

```
salt '*' pip.uninstall <package name>,<package2 name>
salt '*' pip.uninstall requirements=/path/to/requirements.txt
salt '*' pip.uninstall <package name> bin_env=/path/to/virtualenv
salt '*' pip.uninstall <package name> bin_env=/path/to/pip_bin
```

`salt.modules.pip.`**`upgrade`**(*bin_env=None, user=None, cwd=None, use_vt=False*)

New in version 2015.5.0.

Upgrades outdated pip packages.

Note: On Windows you can't update salt from pip using salt, so salt will be skipped

Returns a dict containing the changes.

{'<package>': {'old': '<old-version>', 'new': '<new-version>'}}

CLI Example:

```
salt '*' pip.upgrade
```

`salt.modules.pip.`**`upgrade_available`**(*pkg, bin_env=None, user=None, cwd=None*)

New in version 2015.5.0.

Check whether or not an upgrade is available for a given package

CLI Example:

```
salt '*' pip.upgrade_available <package name>
```

`salt.modules.pip.`**`version`**(*bin_env=None, cwd=None*)

New in version 0.17.0.

Returns the version of pip. Use `bin_env` to specify the path to a virtualenv and get the version of pip in that virtualenv.

If unable to detect the pip version, returns `None`.

CLI Example:

```
salt '*' pip.version
```

21.9.322 salt.modules.pkg_resource

Resources needed by pkg providers

`salt.modules.pkg_resource.`**`add_pkg`**(*pkgs, name, pkgver*)

Add a package to a dict of installed packages.

CLI Example:

```
salt '*' pkg_resource.add_pkg '{}' bind 9
```

`salt.modules.pkg_resource.`**`check_extra_requirements`**(*pkgname, pkgver*)

Check if the installed package already has the given requirements. This function will return the result of `pkg.check_extra_requirements` if this function exists for the minion, otherwise it will return True.

CLI Example:

```
salt '*' pkg_resource.check_extra_requirements <pkgname> <extra_requirements>
```

`salt.modules.pkg_resource.`**`format_pkg_list`**(*packages, versions_as_list, attr*)
Formats packages according to parameters for list_pkgs.

`salt.modules.pkg_resource.`**`format_version`**(*epoch, version, release*)
Formats a version string for list_pkgs.

`salt.modules.pkg_resource.`**`pack_sources`**(*sources, normalize=True*)
Accepts list of dicts (or a string representing a list of dicts) and packs the key/value pairs into a single dict.

`'[{"foo": "salt://foo.rpm"}, {"bar": "salt://bar.rpm"}]'` would become `{"foo": "salt://foo.rpm", "bar": "salt://bar.rpm"}`

normalize [True] Normalize the package name by removing the architecture, if the architecture of the package is different from the architecture of the operating system. The ability to disable this behavior is useful for poorly-created packages which include the architecture as an actual part of the name, such as kernel modules which match a specific kernel version.

New in version 2015.8.0.

CLI Example:

```
salt '*' pkg_resource.pack_sources '[{"foo": "salt://foo.rpm"}, {"bar": "salt://
↪bar.rpm"}]'
```

`salt.modules.pkg_resource.`**`parse_targets`**(*name=None, pkgs=None, sources=None, saltenv='base', normalize=True, **kwargs*)
Parses the input to pkg.install and returns back the package(s) to be installed. Returns a list of packages, as well as a string noting whether the packages are to come from a repository or a binary package.

CLI Example:

```
salt '*' pkg_resource.parse_targets
```

`salt.modules.pkg_resource.`**`sort_pkglist`**(*pkgs*)
Accepts a dict obtained from pkg.list_pkgs() and sorts in place the list of versions for any packages that have multiple versions installed, so that two package lists can be compared to one another.

CLI Example:

```
salt '*' pkg_resource.sort_pkglist '["3.45", "2.13"]'
```

`salt.modules.pkg_resource.`**`stringify`**(*pkgs*)
Takes a dict of package name/version information and joins each list of installed versions into a string.

CLI Example:

```
salt '*' pkg_resource.stringify 'vim: 7.127'
```

`salt.modules.pkg_resource.`**`version`**(**names, **kwargs*)
Common interface for obtaining the version of installed packages.

CLI Example:

```
salt '*' pkg_resource.version vim
salt '*' pkg_resource.version foo bar baz
salt '*' pkg_resource.version 'python*'
```

`salt.modules.pkg_resource.`**`version_clean`**`(`*verstr*`)`

Clean the version string removing extra data. This function will simply try to call `pkg.version_clean`.

CLI Example:

```
salt '*' pkg_resource.version_clean <version_string>
```

21.9.323 salt.modules.pkgin

Package support for pkgin based systems, inspired from freebsdpkg module

---

Important: If you feel that Salt should be using this module to manage packages on a minion, and it is using a different module (or gives an error similar to *'pkg.install' is not available*), see *here*.

`salt.modules.pkgin.`**`available_version`**`(`**names, **kwargs*`)`

This function is an alias of `latest_version`.

Return the latest version of the named package available for upgrade or installation.

If the latest version of a given package is already installed, an empty string will be returned for that package.

CLI Example:

```
salt '*' pkg.latest_version <package name>
salt '*' pkg.latest_version <package1> <package2> ...
```

`salt.modules.pkgin.`**`file_dict`**`(`**packages*`)`

List the files that belong to a package.

CLI Examples:

```
salt '*' pkg.file_dict nginx
salt '*' pkg.file_dict nginx varnish
```

`salt.modules.pkgin.`**`file_list`**`(`*package*`)`

List the files that belong to a package.

CLI Examples:

```
salt '*' pkg.file_list nginx
```

`salt.modules.pkgin.`**`install`**`(`*name=None, refresh=False, fromrepo=None, pkgs=None, sources=None, **kwargs*`)`

Install the passed package

name The name of the package to be installed.

refresh Whether or not to refresh the package database before installing.

fromrepo Specify a package repository to install from.

Multiple Package Installation Options:

pkgs A list of packages to install from a software repository. Must be passed as a python list.

CLI Example:

```
salt '*' pkg.install pkgs='["foo","bar"]'
```

sources A list of packages to install. Must be passed as a list of dicts, with the keys being package names, and the values being the source URI or local path to the package.

CLI Example:

```
salt '*' pkg.install sources='[{"foo": "salt://foo.deb"},{"bar": "salt://bar.
↪deb"}]'
```

Return a dict containing the new package names and versions:

```
{'<package>': {'old': '<old-version>',
               'new': '<new-version>'}}
```

CLI Example:

```
salt '*' pkg.install <package name>
```

salt.modules.pkgin.**latest_version**(**names, **kwargs*)

Return the latest version of the named package available for upgrade or installation.

If the latest version of a given package is already installed, an empty string will be returned for that package.

CLI Example:

```
salt '*' pkg.latest_version <package name>
salt '*' pkg.latest_version <package1> <package2> ...
```

salt.modules.pkgin.**list_pkgs**(*versions_as_list=False, **kwargs*)

List the packages currently installed as a dict:

```
{'<package_name>': '<version>'}
```

CLI Example:

```
salt '*' pkg.list_pkgs
```

salt.modules.pkgin.**list_upgrades**(*refresh=True, **kwargs*)

List all available package upgrades.

New in version 2018.3.0.

refresh Whether or not to refresh the package database before installing.

CLI Example:

```
salt '*' pkg.list_upgrades
```

salt.modules.pkgin.**purge**(*name=None, pkgs=None, **kwargs*)

Package purges are not supported, this function is identical to `remove()`.

name The name of the package to be deleted.

Multiple Package Options:

pkgs A list of packages to delete. Must be passed as a python list. The `name` parameter will be ignored if this option is passed.

New in version 0.16.0.

Returns a dict containing the changes.

CLI Example:

```
salt '*' pkg.purge <package name>
salt '*' pkg.purge <package1>,<package2>,<package3>
salt '*' pkg.purge pkgs='["foo", "bar"]'
```

salt.modules.pkgin.**refresh_db**(*force=False*)

Use pkg update to get latest pkg_summary

force Pass -f so that the cache is always refreshed.

New in version 2018.3.0.

CLI Example:

```
salt '*' pkg.refresh_db
```

`salt.modules.pkgin.`**`remove`**`(`*name=None, pkgs=None, **kwargs*`)`

name The name of the package to be deleted.

Multiple Package Options:

pkgs A list of packages to delete. Must be passed as a python list. The `name` parameter will be ignored if this option is passed.

New in version 0.16.0.

Returns a list containing the removed packages.

CLI Example:

```
salt '*' pkg.remove <package name>
salt '*' pkg.remove <package1>,<package2>,<package3>
salt '*' pkg.remove pkgs='["foo", "bar"]'
```

`salt.modules.pkgin.`**`search`**`(`*pkg_name*`)`

Searches for an exact match using pkgin ^package$

CLI Example:

```
salt '*' pkg.search 'mysql-server'
```

`salt.modules.pkgin.`**`upgrade`**`(`*refresh=True, pkgs=None, **kwargs*`)`

Run pkg upgrade, if pkgin used. Otherwise do nothing

refresh Whether or not to refresh the package database before installing.

Multiple Package Upgrade Options:

pkgs A list of packages to upgrade from a software repository. Must be passed as a python list.

CLI Example:

```
salt '*' pkg.upgrade pkgs='["foo","bar"]'
```

Returns a dictionary containing the changes:

```
{'<package>':  {'old': '<old-version>',
                'new': '<new-version>'}}
```

CLI Example:

```
salt '*' pkg.upgrade
```

`salt.modules.pkgin.`**`version`**`(`**names, **kwargs*`)`

Returns a string representing the package version or an empty string if not installed. If more than one package name is specified, a dict of name/version pairs is returned.

CLI Example:

```
salt '*' pkg.version <package name>
salt '*' pkg.version <package1> <package2> <package3> ...
```

21.9.324 salt.modules.pkgng

Support for `pkgng`, the new package manager for FreeBSD

Important: If you feel that Salt should be using this module to manage packages on a minion, and it is using a different module (or gives an error similar to *'pkg.install' is not available*), see *here*.

Warning: This module has been completely rewritten. Up to and including version 0.17.x, it was available as the `pkgng` module, (`pkgng.install`, `pkgng.delete`, etc.), but moving forward this module will no longer be available as `pkgng`, as it will behave like a normal Salt `pkg` provider. The documentation below should not be considered to apply to this module in versions <= 0.17.x. If your minion is running a 0.17.x release or older, then the documentation for this module can be viewed using the `sys.doc` function:

```
salt bsdminion sys.doc pkgng
```

This module provides an interface to `pkg(8)`. It acts as the default package provider for FreeBSD 10 and newer. For FreeBSD hosts which have been upgraded to use pkgng, you will need to override the `pkg` provider by setting the *`providers`* parameter in your Minion config file, in order to use this module to manage packages, like so:

```
providers:
  pkg: pkgng
```

`salt.modules.pkgng.`**`audit`**(*jail=None*, *chroot=None*, *root=None*)

Audits installed packages against known vulnerabilities

CLI Example:

```
salt '*' pkg.audit
```

jail Audit packages within the specified jail

CLI Example:

```
salt '*' pkg.audit jail=<jail name or id>
```

chroot Audit packages within the specified chroot (ignored if `jail` is specified)

root Audit packages within the specified root (ignored if `jail` is specified)

CLI Example:

```
salt '*' pkg.audit chroot=/path/to/chroot
```

`salt.modules.pkgng.`**`autoremove`**(*jail=None*, *chroot=None*, *root=None*, *dryrun=False*)

Delete packages which were automatically installed as dependencies and are not required anymore.

dryrun Dry-run mode. The list of changes to packages is always printed, but no changes are actually made.

CLI Example:

```
salt '*' pkg.autoremove
salt '*' pkg.autoremove jail=<jail name or id>
salt '*' pkg.autoremove dryrun=True
salt '*' pkg.autoremove jail=<jail name or id> dryrun=True
```

`salt.modules.pkgng.`**`backup`**(*file_name*, *jail=None*, *chroot=None*, *root=None*)

Export installed packages into yaml+mtree file

CLI Example:

```
salt '*' pkg.backup /tmp/pkg
```

jail Backup packages from the specified jail. Note that this will run the command within the jail, and so the path to the backup file will be relative to the root of the jail

CLI Example:

```
salt '*' pkg.backup /tmp/pkg jail=<jail name or id>
```

chroot Backup packages from the specified chroot (ignored if `jail` is specified). Note that this will run the command within the chroot, and so the path to the backup file will be relative to the root of the chroot.

root Backup packages from the specified root (ignored if `jail` is specified). Note that this will run the command within the root, and so the path to the backup file will be relative to the root of the root.

CLI Example:

```
salt '*' pkg.backup /tmp/pkg chroot=/path/to/chroot
```

`salt.modules.pkgng.`**`check`**(*jail=None, chroot=None, root=None, depends=False, recompute=False, checksum=False*)

Sanity checks installed packages

jail Perform the sanity check in the specified jail

CLI Example:

```
salt '*' pkg.check jail=<jail name or id>
```

chroot Perform the sanity check in the specified chroot (ignored if `jail` is specified)

root Perform the sanity check in the specified root (ignored if `jail` is specified)

CLI Example:

```
salt '*' pkg.check chroot=/path/to/chroot
```

Of the below, at least one must be set to `True`.

depends Check for and install missing dependencies.

CLI Example:

```
salt '*' pkg.check recompute=True
```

recompute Recompute sizes and checksums of installed packages.

CLI Example:

```
salt '*' pkg.check depends=True
```

checksum Find invalid checksums for installed packages.

CLI Example:

```
salt '*' pkg.check checksum=True
```

`salt.modules.pkgng.`**`clean`**(*jail=None, chroot=None, root=None, clean_all=False, dryrun=False*)

Cleans the local cache of fetched remote packages

CLI Example:

```
salt '*' pkg.clean
```

jail Cleans the package cache in the specified jail

CLI Example:

```
salt '*' pkg.clean jail=<jail name or id>
```

chroot Cleans the package cache in the specified chroot (ignored if `jail` is specified)

root Cleans the package cache in the specified root (ignored if `jail` is specified)

CLI Example:

```
salt '*' pkg.clean chroot=/path/to/chroot
```

clean_all Clean all packages from the local cache (not just those that have been superseded by newer versions).

CLI Example:

```
```

salt '*' pkg.clean clean_all=True

dryrun Dry-run mode. This list of changes to the local cache is always printed, but no changes are actually made.

CLI Example:

```
salt '*' pkg.clean dryrun=True
```

`salt.modules.pkgng.`**`fetch`**(*name, jail=None, chroot=None, root=None, fetch_all=False, quiet=False, fromrepo=None, glob=True, regex=False, pcre=False, local=False, depends=False*)

Fetches remote packages

CLI Example:

```
salt '*' pkg.fetch <package name>
```

jail Fetch package in the specified jail

CLI Example:

```
salt '*' pkg.fetch <package name> jail=<jail name or id>
```

chroot Fetch package in the specified chroot (ignored if `jail` is specified)

root Fetch package in the specified root (ignored if `jail` is specified)

CLI Example:

```
salt '*' pkg.fetch <package name> chroot=/path/to/chroot
```

fetch_all Fetch all packages.

CLI Example:

```
salt '*' pkg.fetch <package name> fetch_all=True
```

quiet Quiet mode. Show less output.

CLI Example:

```
salt '*' pkg.fetch <package name> quiet=True
```

fromrepo Fetches packages from the given repo if multiple repo support is enabled. See `pkg.conf(5)`.

CLI Example:

```
salt '*' pkg.fetch <package name> fromrepo=repo
```

glob Treat pkg_name as a shell glob pattern.

CLI Example:

```
salt '*' pkg.fetch <package name> glob=True
```

regex Treat pkg_name as a regular expression.

CLI Example:

```
salt '*' pkg.fetch <regular expression> regex=True
```

pcre Treat pkg_name is an extended regular expression.

CLI Example:

```
salt '*' pkg.fetch <extended regular expression> pcre=True
```

local Skip updating the repository catalogs with pkg-update(8). Use the local cache only.

CLI Example:

```
salt '*' pkg.fetch <package name> local=True
```

depends Fetch the package and its dependencies as well.

CLI Example:

```
salt '*' pkg.fetch <package name> depends=True
```

`salt.modules.pkgng.`**`hold`**(*name=None*, *pkgs=None*, ***kwargs*)

Version-lock packages

Note: This function is provided primarily for compatibilty with some parts of `states.pkg`. Consider using Consider using `pkg.lock` instead. instead.

---

name The name of the package to be held.

Multiple Package Options:

pkgs A list of packages to hold. Must be passed as a python list. The `name` parameter will be ignored if this option is passed.

Returns a dict containing the changes.

CLI Example:

```
salt '*' pkg.hold <package name>
salt '*' pkg.hold pkgs='["foo", "bar"]'
```

`salt.modules.pkgng.`**`install`**(*name=None*, *fromrepo=None*, *pkgs=None*, *sources=None*, *jail=None*, *chroot=None*, *root=None*, *orphan=False*, *force=False*, *glob=False*, *local=False*, *dryrun=False*, *quiet=False*, *reinstall_requires=False*, *regex=False*, *pcre=False*, *batch=False*, ***kwargs*)

Install package(s) from a repository

name The name of the package to install

CLI Example:

```
salt '*' pkg.install <package name>
```

jail Install the package into the specified jail

chroot Install the package into the specified chroot (ignored if `jail` is specified)

root Install the package into the specified root (ignored if `jail` is specified)

orphan Mark the installed package as orphan. Will be automatically removed if no other packages depend on them. For more information please refer to `pkg-autoremove(8)`.

CLI Example:

```
salt '*' pkg.install <package name> orphan=True
```

force Force the reinstallation of the package if already installed.

CLI Example:

```
salt '*' pkg.install <package name> force=True
```

glob Treat the package names as shell glob patterns.

CLI Example:

```
salt '*' pkg.install <package name> glob=True
```

local Do not update the repository catalogs with `pkg-update(8)`. A value of `True` here is equivalent to using the `-U` flag with `pkg install`.

CLI Example:

```
salt '*' pkg.install <package name> local=True
```

dryrun Dru-run mode. The list of changes to packages is always printed, but no changes are actually made.

CLI Example:

```
salt '*' pkg.install <package name> dryrun=True
```

quiet Force quiet output, except when dryrun is used, where pkg install will always show packages to be installed, upgraded or deleted.

CLI Example:

```
salt '*' pkg.install <package name> quiet=True
```

reinstall_requires When used with force, reinstalls any packages that require the given package.

CLI Example:

```
salt '*' pkg.install <package name> reinstall_requires=True force=True
```

Changed in version 2014.7.0: `require` kwarg renamed to `reinstall_requires`

fromrepo In multi-repo mode, override the pkg.conf ordering and only attempt to download packages from the named repository.

CLI Example:

```
salt '*' pkg.install <package name> fromrepo=repo
```

regex Treat the package names as a regular expression

CLI Example:

```
salt '*' pkg.install <regular expression> regex=True
```

pcre Treat the package names as extended regular expressions.

CLI Example:

batch Use BATCH=true for pkg install, skipping all questions. Be careful when using in production.

CLI Example:

```
salt '*' pkg.install <package name> batch=True
```

salt.modules.pkgng.**latest_version**(**names, **kwargs*)

Return the latest version of the named package available for upgrade or installation. If more than one package name is specified, a dict of name/version pairs is returned.

If the latest version of a given package is already installed, an empty string will be returned for that package.

CLI Example:

```
salt '*' pkg.latest_version <package name>
salt '*' pkg.latest_version <package name> jail=<jail name or id>
salt '*' pkg.latest_version <package name> chroot=/path/to/chroot
```

salt.modules.pkgng.**list_locked**(***kwargs*)

Query the package database those packages which are locked against reinstallation, modification or deletion.

Returns returns a list of package names with version strings

CLI Example:

```
salt '*' pkg.list_locked
```

jail List locked packages within the specified jail

CLI Example:

```
salt '*' pkg.list_locked jail=<jail name or id>
```

chroot List locked packages within the specified chroot (ignored if `jail` is specified)

CLI Example:

```
salt '*' pkg.list_locked chroot=/path/to/chroot
```

root List locked packages within the specified root (ignored if `jail` is specified)

CLI Example:

```
salt '*' pkg.list_locked root=/path/to/chroot
```

salt.modules.pkgng.**list_pkgs**(*versions_as_list=False, jail=None, chroot=None, root=None, with_origin=False, **kwargs*)

List the packages currently installed as a dict:

```
{'<package_name>': '<version>'}
```

jail List the packages in the specified jail
chroot List the packages in the specified chroot (ignored if `jail` is specified)
root List the packages in the specified root (ignored if `jail` is specified)
with_origin [False] Return a nested dictionary containing both the origin name and version for each installed package.

New in version 2014.1.0.

CLI Example:

```
salt '*' pkg.list_pkgs
salt '*' pkg.list_pkgs jail=<jail name or id>
salt '*' pkg.list_pkgs chroot=/path/to/chroot
```

`salt.modules.pkgng.`**`list_upgrades`**`(`*refresh=True*, ***kwargs*`)`

List those packages for which an upgrade is available

The `fromrepo` argument is also supported, as used in pkg states.

CLI Example:

```
salt '*' pkg.list_upgrades
```

jail List upgrades within the specified jail

CLI Example:

```
salt '*' pkg.list_upgrades jail=<jail name or id>
```

chroot List upgrades within the specified chroot (ignored if `jail` is specified)

CLI Example:

```
salt '*' pkg.list_upgrades chroot=/path/to/chroot
```

root List upgrades within the specified root (ignored if `jail` is specified)

CLI Example:

```
salt '*' pkg.list_upgrades root=/path/to/chroot
```

`salt.modules.pkgng.`**`lock`**`(`*name*, ***kwargs*`)`

Lock the named package against reinstallation, modification or deletion.

Returns True if the named package was successfully locked.

CLI Example:

```
salt '*' pkg.lock <package name>
```

jail Lock packages within the specified jail

CLI Example:

```
salt '*' pkg.lock <package name> jail=<jail name or id>
```

chroot Lock packages within the specified chroot (ignored if `jail` is specified)

CLI Example:

```
salt '*' pkg.lock <package name> chroot=/path/to/chroot
```

root Lock packages within the specified root (ignored if `jail` is specified)

CLI Example:

```
salt '*' pkg.lock <package name> root=/path/to/chroot
```

`salt.modules.pkgng.`**`locked`**`(`*name*, ***kwargs*`)`

Query the package database to determine if the named package is locked against reinstallation, modification or deletion.

Returns True if the named package is locked, False otherwise.

CLI Example:

```
salt '*' pkg.locked <package name>
```

jail Test if a package is locked within the specified jail

CLI Example:

```
salt '*' pkg.locked <package name> jail=<jail name or id>
```

chroot Test if a package is locked within the specified chroot (ignored if `jail` is specified)

CLI Example:

```
salt '*' pkg.locked <package name> chroot=/path/to/chroot
```

root Test if a package is locked within the specified root (ignored if `jail` is specified)

CLI Example:

```
salt '*' pkg.locked <package name> root=/path/to/chroot
```

`salt.modules.pkgng.`**`parse_config`**(*file_name='/usr/local/etc/pkg.conf'*)

Return dict of uncommented global variables.

CLI Example:

```
salt '*' pkg.parse_config
```

`NOTE:` not working properly right now

`salt.modules.pkgng.`**`refresh_db`**(*jail=None, chroot=None, root=None, force=False*)

Refresh PACKAGESITE contents

Note: This function can accessed using `pkg.update` in addition to `pkg.refresh_db`, to more closely match the CLI usage of `pkg(8)`.

CLI Example:

```
salt '*' pkg.refresh_db
```

jail Refresh the pkg database within the specified jail

chroot Refresh the pkg database within the specified chroot (ignored if `jail` is specified)

root Refresh the pkg database within the specified root (ignored if `jail` is specified)

force Force a full download of the repository catalog without regard to the respective ages of the local and remote copies of the catalog.

CLI Example:

```
salt '*' pkg.refresh_db force=True
```

`salt.modules.pkgng.`**`remove`**(*name=None, pkgs=None, jail=None, chroot=None, root=None, all_installed=False, force=False, glob=False, dryrun=False, recurse=False, regex=False, pcre=False, **kwargs*)

Remove a package from the database and system

Note: This function can accessed using `pkg.delete` in addition to `pkg.remove`, to more closely match the CLI usage of `pkg(8)`.

name The package to remove

CLI Example:

```
salt '*' pkg.remove <package name>
```

jail Delete the package from the specified jail

chroot Delete the package from the specified chroot (ignored if `jail` is specified)

root Delete the package from the specified root (ignored if `jail` is specified)

all_installed Deletes all installed packages from the system and empties the database. USE WITH CAUTION!

CLI Example:

```
salt '*' pkg.remove all all_installed=True force=True
```

force Forces packages to be removed despite leaving unresolved dependencies.

CLI Example:

```
salt '*' pkg.remove <package name> force=True
```

glob Treat the package names as shell glob patterns.

CLI Example:

```
salt '*' pkg.remove <package name> glob=True
```

dryrun Dry run mode. The list of packages to delete is always printed, but no packages are actually deleted.

CLI Example:

```
salt '*' pkg.remove <package name> dryrun=True
```

recurse Delete all packages that require the listed package as well.

CLI Example:

```
salt '*' pkg.remove <package name> recurse=True
```

regex Treat the package names as regular expressions.

CLI Example:

```
salt '*' pkg.remove <regular expression> regex=True
```

pcre Treat the package names as extended regular expressions.

CLI Example:

```
salt '*' pkg.remove <extended regular expression> pcre=True
```

`salt.modules.pkgng.`**`restore`**(*file_name*, *jail=None*, *chroot=None*, *root=None*)

Reads archive created by pkg backup -d and recreates the database.

CLI Example:

```
salt '*' pkg.restore /tmp/pkg
```

jail Restore database to the specified jail. Note that this will run the command within the jail, and so the path to the file from which the pkg database will be restored is relative to the root of the jail.

CLI Example:

```
salt '*' pkg.restore /tmp/pkg jail=<jail name or id>
```

chroot Restore database to the specified chroot (ignored if `jail` is specified). Note that this will run the command within the chroot, and so the path to the file from which the pkg database will be restored is relative to the root of the chroot.

root Restore database to the specified root (ignored if `jail` is specified). Note that this will run the command within the root, and so the path to the file from which the pkg database will be restored is relative to the root of the root.

CLI Example:

```
salt '*' pkg.restore /tmp/pkg chroot=/path/to/chroot
```

`salt.modules.pkgng.search`(*name*, *jail=None*, *chroot=None*, *root=None*, *exact=False*, *glob=False*, *regex=False*, *pcre=False*, *comment=False*, *desc=False*, *full=False*, *depends=False*, *size=False*, *quiet=False*, *origin=False*, *prefix=False*)

Searches in remote package repositories

CLI Example:

```
salt '*' pkg.search pattern
```

jail Perform the search using the `pkg.conf(5)` from the specified jail

CLI Example:

```
salt '*' pkg.search pattern jail=<jail name or id>
```

chroot Perform the search using the `pkg.conf(5)` from the specified chroot (ignored if `jail` is specified)

root Perform the search using the `pkg.conf(5)` from the specified root (ignored if `jail` is specified)

CLI Example:

```
salt '*' pkg.search pattern chroot=/path/to/chroot
```

exact Treat pattern as exact pattern.

CLI Example:

```
salt '*' pkg.search pattern exact=True
```

glob Treat pattern as a shell glob pattern.

CLI Example:

```
salt '*' pkg.search pattern glob=True
```

regex Treat pattern as a regular expression.

CLI Example:

```
salt '*' pkg.search pattern regex=True
```

pcre Treat pattern as an extended regular expression.

CLI Example:

```
salt '*' pkg.search pattern pcre=True
```

comment Search for pattern in the package comment one-line description.

CLI Example:

```
salt '*' pkg.search pattern comment=True
```

desc Search for pattern in the package description.

CLI Example:

```
salt '*' pkg.search pattern desc=True
```

full Displays full information about the matching packages.

CLI Example:

```
salt '*' pkg.search pattern full=True
```

depends Displays the dependencies of pattern.

CLI Example:

```
salt '*' pkg.search pattern depends=True
```

size Displays the size of the package

CLI Example:

```
salt '*' pkg.search pattern size=True
```

quiet Be quiet. Prints only the requested information without displaying many hints.

CLI Example:

```
salt '*' pkg.search pattern quiet=True
```

origin Displays pattern origin.

CLI Example:

```
salt '*' pkg.search pattern origin=True
```

prefix Displays the installation prefix for each package matching pattern.

CLI Example:

```
salt '*' pkg.search pattern prefix=True
```

`salt.modules.pkgng.`**`stats`**(*local=False, remote=False, jail=None, chroot=None, root=None*)

Return pkgng stats.

CLI Example:

```
salt '*' pkg.stats
```

local Display stats only for the local package database.

CLI Example:

```
salt '*' pkg.stats local=True
```

remote Display stats only for the remote package database(s).

CLI Example:

```
salt '*' pkg.stats remote=True
```

jail Retrieve stats from the specified jail.

CLI Example:

```
salt '*' pkg.stats jail=<jail name or id>
salt '*' pkg.stats jail=<jail name or id> local=True
salt '*' pkg.stats jail=<jail name or id> remote=True
```

chroot Retrieve stats from the specified chroot (ignored if `jail` is specified).

root Retrieve stats from the specified root (ignored if `jail` is specified).

CLI Example:

```
salt '*' pkg.stats chroot=/path/to/chroot
salt '*' pkg.stats chroot=/path/to/chroot local=True
salt '*' pkg.stats chroot=/path/to/chroot remote=True
```

`salt.modules.pkgng.`**`unhold`**(*name=None*, *pkgs=None*, ***kwargs*)

Remove version locks

Note: This function is provided primarily for compatibilty with some parts of *`states.pkg`*. Consider using *`pkg.unlock`* instead.

---

name The name of the package to be unheld

Multiple Package Options:

pkgs A list of packages to unhold. Must be passed as a python list. The `name` parameter will be ignored if this option is passed.

Returns a dict containing the changes.

CLI Example:

```
salt '*' pkg.unhold <package name>
salt '*' pkg.unhold pkgs='["foo", "bar"]'
```

`salt.modules.pkgng.`**`unlock`**(*name*, ***kwargs*)

Unlock the named package against reinstallation, modification or deletion.

Returns True if the named package was successfully unlocked.

CLI Example:

```
salt '*' pkg.unlock <package name>
```

jail Unlock packages within the specified jail

CLI Example:

```
salt '*' pkg.unlock <package name> jail=<jail name or id>
```

chroot Unlock packages within the specified chroot (ignored if `jail` is specified)

CLI Example:

```
salt '*' pkg.unlock <package name> chroot=/path/to/chroot
```

root Unlock packages within the specified root (ignored if `jail` is specified)

CLI Example:

```
salt '*' pkg.unlock <package name> root=/path/to/chroot
```

`salt.modules.pkgng.`**`update_package_site`**(*new_url*)

Updates remote package repo URL, PACKAGESITE var to be exact.

Must use `http://`, `ftp://`, or `https://` protocol

CLI Example:

```
salt '*' pkg.update_package_site http://127.0.0.1/
```

`salt.modules.pkgng.`**`updating`**`(`*name, jail=None, chroot=None, root=None, filedate=None, filename=None*`)`

' Displays UPDATING entries of software packages

CLI Example:

```
salt '*' pkg.updating foo
```

jail Perform the action in the specified jail

CLI Example:

```
salt '*' pkg.updating foo jail=<jail name or id>
```

chroot Perform the action in the specified chroot (ignored if `jail` is specified)

root Perform the action in the specified root (ignored if `jail` is specified)

CLI Example:

```
salt '*' pkg.updating foo chroot=/path/to/chroot
```

filedate Only entries newer than date are shown. Use a YYYYMMDD date format.

CLI Example:

```
salt '*' pkg.updating foo filedate=20130101
```

filename Defines an alternative location of the UPDATING file.

CLI Example:

```
salt '*' pkg.updating foo filename=/tmp/UPDATING
```

`salt.modules.pkgng.`**`upgrade`**`(`**names, **kwargs*`)`

Upgrade named or all packages (run a `pkg upgrade`). If <package name> is omitted, the operation is executed on all packages.

Returns a dictionary containing the changes:

```
{'<package>':  {'old': '<old-version>',
                'new': '<new-version>'}}
```

CLI Example:

```
salt '*' pkg.upgrade <package name>
```

jail Audit packages within the specified jail

CLI Example:

```
salt '*' pkg.upgrade <package name> jail=<jail name or id>
```

chroot Audit packages within the specified chroot (ignored if `jail` is specified)

root Audit packages within the specified root (ignored if `jail` is specified)

CLI Example:

```
salt '*' pkg.upgrade <package name> chroot=/path/to/chroot
```

Any of the below options can also be used with `jail` or `chroot`.

force Force reinstalling/upgrading the whole set of packages.

CLI Example:

```
salt '*' pkg.upgrade <package name> force=True
```

local Do not update the repository catalogs with `pkg-update(8)`. A value of `True` here is equivalent to using the `-U` flag with `pkg upgrade`.

CLI Example:

```
salt '*' pkg.upgrade <package name> local=True
```

dryrun Dry-run mode: show what packages have updates available, but do not perform any upgrades. Repository catalogs will be updated as usual unless the local option is also given.

CLI Example:

```
salt '*' pkg.upgrade <package name> dryrun=True
```

`salt.modules.pkgng.`**`version`**`(`**names, **kwargs*`)`

Returns a string representing the package version or an empty string if not installed. If more than one package name is specified, a dict of name/version pairs is returned.

Note: This function can accessed using `pkg.info` in addition to `pkg.version`, to more closely match the CLI usage of `pkg(8)`.

jail Get package version information for the specified jail

chroot Get package version information for the specified chroot (ignored if `jail` is specified)

root Get package version information for the specified root (ignored if `jail` is specified)

with_origin [False] Return a nested dictionary containing both the origin name and version for each specified package.

New in version 2014.1.0.

CLI Example:

```
salt '*' pkg.version <package name>
salt '*' pkg.version <package name> jail=<jail name or id>
salt '*' pkg.version <package1> <package2> <package3> ...
```

`salt.modules.pkgng.`**`version_cmp`**`(`*pkg1, pkg2, ignore_epoch=False*`)`

Do a cmp-style comparison on two packages. Return -1 if pkg1 < pkg2, 0 if pkg1 == pkg2, and 1 if pkg1 > pkg2. Return None if there was a problem making the comparison.

CLI Example:

```
salt '*' pkg.version_cmp '2.1.11' '2.1.12'
```

`salt.modules.pkgng.`**`which`**`(`*path, jail=None, chroot=None, root=None, origin=False, quiet=False*`)`

Displays which package installed a specific file

CLI Example:

```
salt '*' pkg.which <file name>
```

jail Perform the check in the specified jail

CLI Example:

```
salt '*' pkg.which <file name> jail=<jail name or id>
```

chroot Perform the check in the specified chroot (ignored if `jail` is specified)
root Perform the check in the specified root (ignored if `jail` is specified)

CLI Example:

```
salt '*' pkg.which <file name> chroot=/path/to/chroot
```

origin Shows the origin of the package instead of name-version.

CLI Example:

```
salt '*' pkg.which <file name> origin=True
```

quiet Quiet output.

CLI Example:

```
salt '*' pkg.which <file name> quiet=True
```

21.9.325 salt.modules.pkgutil

Pkgutil support for Solaris

Important: If you feel that Salt should be using this module to manage packages on a minion, and it is using a different module (or gives an error similar to *'pkg.install' is not available*), see *here*.

`salt.modules.pkgutil.`**`install`**(*name=None*, *refresh=False*, *version=None*, *pkgs=None*, ***kwargs*)

Install packages using the pkgutil tool.

CLI Example:

```
salt '*' pkg.install <package_name>
salt '*' pkg.install SMClgcc346
```

Multiple Package Installation Options:
pkgs A list of packages to install from OpenCSW. Must be passed as a python list.

CLI Example:

```
salt '*' pkg.install pkgs='["foo", "bar"]'
salt '*' pkg.install pkgs='["foo", {"bar": "1.2.3"}]'
```

Returns a dict containing the new package names and versions:

```
{'<package>': {'old': '<old-version>',
               'new': '<new-version>'}}
```

`salt.modules.pkgutil.`**`latest_version`**(**names*, ***kwargs*)

Return the latest version of the named package available for upgrade or installation. If more than one package name is specified, a dict of name/version pairs is returned.

If the latest version of a given package is already installed, an empty string will be returned for that package.

CLI Example:

```
salt '*' pkgutil.latest_version CSWpython
salt '*' pkgutil.latest_version <package1> <package2> <package3> ...
```

`salt.modules.pkgutil.`**`list_pkgs`**(*versions_as_list=False*, ***kwargs*)

List the packages currently installed as a dict:

```
{'<package_name>': '<version>'}
```

CLI Example:

```
salt '*' pkg.list_pkgs
salt '*' pkg.list_pkgs versions_as_list=True
```

`salt.modules.pkgutil.`**`list_upgrades`**(*refresh=True*, ***kwargs*)

List all available package upgrades on this system

CLI Example:

```
salt '*' pkgutil.list_upgrades
```

`salt.modules.pkgutil.`**`purge`**(*name=None*, *pkgs=None*, ***kwargs*)

Package purges are not supported, this function is identical to `remove()`.

name The name of the package to be deleted.

Multiple Package Options:

pkgs A list of packages to delete. Must be passed as a python list. The `name` parameter will be ignored if this option is passed.

New in version 0.16.0.

Returns a dict containing the changes.

CLI Example:

```
salt '*' pkg.purge <package name>
salt '*' pkg.purge <package1>,<package2>,<package3>
salt '*' pkg.purge pkgs='["foo", "bar"]'
```

`salt.modules.pkgutil.`**`refresh_db`**()

Updates the pkgutil repo database (pkgutil -U)

CLI Example:

```
salt '*' pkgutil.refresh_db
```

`salt.modules.pkgutil.`**`remove`**(*name=None*, *pkgs=None*, ***kwargs*)

Remove a package and all its dependencies which are not in use by other packages.

name The name of the package to be deleted.

Multiple Package Options:

pkgs A list of packages to delete. Must be passed as a python list. The `name` parameter will be ignored if this option is passed.

New in version 0.16.0.

Returns a dict containing the changes.

CLI Example:

```
salt '*' pkg.remove <package name>
salt '*' pkg.remove <package1>,<package2>,<package3>
salt '*' pkg.remove pkgs='["foo", "bar"]'
```

`salt.modules.pkgutil.`**`upgrade`**`(`*refresh=True*`)`

Upgrade all of the packages to the latest available version.

Returns a dict containing the changes:

```
{'<package>': {'old': '<old-version>',
               'new': '<new-version>'}}
```

CLI Example:

```
salt '*' pkgutil.upgrade
```

`salt.modules.pkgutil.`**`upgrade_available`**`(`*name*`)`

Check if there is an upgrade available for a certain package

CLI Example:

```
salt '*' pkgutil.upgrade_available CSWpython
```

`salt.modules.pkgutil.`**`version`**`(`**names, **kwargs*`)`

Returns a version if the package is installed, else returns an empty string

CLI Example:

```
salt '*' pkgutil.version CSWpython
```

21.9.326 salt.modules.portage_config

Configure `portage(5)`

`salt.modules.portage_config.`**`append_to_package_conf`**`(`*conf, atom=”, flags=None, string=”, overwrite=False*`)`

Append a string or a list of flags for a given package or DEPEND atom to a given configuration file.

CLI Example:

```
salt '*' portage_config.append_to_package_conf use string="app-admin/salt ldap -
↪libvirt"
salt '*' portage_config.append_to_package_conf use atom="> = app-admin/salt-0.14.1
↪" flags="['ldap', '-libvirt']"
```

`salt.modules.portage_config.`**`append_use_flags`**`(`*atom, uses=None, overwrite=False*`)`

Append a list of use flags for a given package or DEPEND atom

CLI Example:

```
salt '*' portage_config.append_use_flags "app-admin/salt[ldap, -libvirt]"
salt '*' portage_config.append_use_flags ">=app-admin/salt-0.14.1" "['ldap', '-
↪libvirt']"
```

`salt.modules.portage_config.`**`enforce_nice_config`**`()`

Enforce a nice tree structure for /etc/portage/package.* configuration files.

See also:

`salt.modules.ebuild.ex_mod_init()` for information on automatically running this when pkg is used.

CLI Example:

```
salt '*' portage_config.enforce_nice_config
```

salt.modules.portage_config.**filter_flags**(*use*, *use_expand_hidden*, *usemasked*, *useforced*)
New in version 2015.8.0.

Filter function to remove hidden or otherwise not normally visible USE flags from a list.

@type use: list @param use: the USE flag list to be filtered. @type use_expand_hidden: list @param use_expand_hidden: list of flags hidden. @type usemasked: list @param usemasked: list of masked USE flags. @type useforced: list @param useforced: the forced USE flags. @rtype: list @return the filtered USE flags.

salt.modules.portage_config.**get_all_cpv_use**(*cp*)
New in version 2015.8.0.

Uses portage to determine final USE flags and settings for an emerge.

@type cp: string @param cp: eg cat/pkg @rtype: lists @return use, use_expand_hidden, usemask, useforce

salt.modules.portage_config.**get_cleared_flags**(*cp*)
New in version 2015.8.0.

Uses portage for compare use flags which is used for installing package and use flags which now exist int /etc/portage/package.use/

@type cp: string @param cp: eg cat/pkg @rtype: tuple @rparam: tuple with two lists - list of used flags and list of flags which will be used

salt.modules.portage_config.**get_flags_from_package_conf**(*conf*, *atom*)
Get flags for a given package or DEPEND atom. Warning: This only works if the configuration files tree is in the correct format (the one enforced by enforce_nice_config)

CLI Example:

```
salt '*' portage_config.get_flags_from_package_conf license salt
```

salt.modules.portage_config.**get_installed_use**(*cp*, *use='USE'*)
New in version 2015.8.0.

Gets the installed USE flags from the VARDB.

@type: cp: string @param cp: cat/pkg @type use: string @param use: 1 of ["USE", "PKGUSE"] @rtype list @returns [] or the list of IUSE flags

salt.modules.portage_config.**get_iuse**(*cp*)
New in version 2015.8.0.

Gets the current IUSE flags from the tree.

@type: cpv: string @param cpv: cat/pkg @rtype list @returns [] or the list of IUSE flags

salt.modules.portage_config.**get_missing_flags**(*conf*, *atom*, *flags*)
Find out which of the given flags are currently not set. CLI Example:

```
salt '*' portage_config.get_missing_flags use salt "['ldap', '-libvirt', 'openssl
↪']"
```

salt.modules.portage_config.**has_flag**(*conf*, *atom*, *flag*)
Verify if the given package or DEPEND atom has the given flag. Warning: This only works if the configuration files tree is in the correct format (the one enforced by enforce_nice_config)

CLI Example:

```
salt '*' portage_config.has_flag license salt Apache-2.0
```

salt.modules.portage_config.**has_use**(*atom, use*)

Verify if the given package or DEPEND atom has the given use flag. Warning: This only works if the configuration files tree is in the correct format (the one enforced by enforce_nice_config)

CLI Example:

```
salt '*' portage_config.has_use salt libvirt
```

salt.modules.portage_config.**is_changed_uses**(*cp*)

New in version 2015.8.0.

Uses portage for determine if the use flags of installed package is compatible with use flags in portage configs.

@type cp: string @param cp: eg cat/pkg

salt.modules.portage_config.**is_present**(*conf, atom*)

Tell if a given package or DEPEND atom is present in the configuration files tree. Warning: This only works if the configuration files tree is in the correct format (the one enforced by enforce_nice_config)

CLI Example:

```
salt '*' portage_config.is_present unmask salt
```

21.9.327 salt.modules.postfix

Support for Postfix

This module is currently little more than a config file viewer and editor. It is able to read the master.cf file (which is one style) and files in the style of main.cf (which is a different style, that is used in multiple postfix configuration files).

The design of this module is such that when files are edited, a minimum of changes are made to them. Each file should look as if it has been edited by hand; order, comments and whitespace are all preserved.

salt.modules.postfix.**delete**(*queue_id*)

Delete message(s) from the mail queue

CLI Example:

```
salt '*' postfix.delete 5C33CA0DEA

salt '*' postfix.delete ALL
```

salt.modules.postfix.**hold**(*queue_id*)

Put message(s) on hold from the mail queue

CLI Example:

```
salt '*' postfix.hold 5C33CA0DEA

salt '*' postfix.hold ALL
```

salt.modules.postfix.**requeue**(*queue_id*)

Requeue message(s) in the mail queue

CLI Example:

```
salt '*' postfix.requeue 5C33CA0DEA

salt '*' postfix.requeue ALL
```

`salt.modules.postfix.`**`set_main`**`(`*key, value, path='/etc/postfix/main.cf'*`)`

Set a single config value in the main.cf file. If the value does not already exist, it will be appended to the end.

CLI Example:
salt <minion> postfix.set_main mailq_path /usr/bin/mailq

`salt.modules.postfix.`**`set_master`**`(`*service, conn_type, private='y', unpriv='y', chroot='y', wakeup='n', maxproc='100', command='', write_conf=True, path='/etc/postfix/master.cf'*`)`

Set a single config value in the master.cf file. If the value does not already exist, it will be appended to the end.

Because of shell parsing issues, '-' cannot be set as a value, as is normal in the master.cf file; either 'y', 'n' or a number should be used when calling this function from the command line. If the value used matches the default, it will internally be converted to a '-'. Calling this function from the Python API is not affected by this limitation

The settings and their default values, in order, are: service (required), conn_type (required), private (y), unpriv (y), chroot (y), wakeup (n), maxproc (100), command (required).

By default, this function will write out the changes to the master.cf file, and then returns the full contents of the file. By setting the `write_conf` option to `False`, it will skip writing the file.

CLI Example:
salt <minion> postfix.set_master smtp inet n y n n 100 smtpd

`salt.modules.postfix.`**`show_main`**`(`*path='/etc/postfix/main.cf'*`)`

Return a dict of active config values. This does not include comments, spacing or order. Bear in mind that order is functionally important in the main.cf file, since keys can be referred to as variables. This means that the data returned from this function should not be used for direct modification of the main.cf file; other functions are available for that.

CLI Examples:
salt <minion> postfix.show_main salt <minion> postfix.show_main path=/path/to/main.cf

`salt.modules.postfix.`**`show_master`**`(`*path='/etc/postfix/master.cf'*`)`

Return a dict of active config values. This does not include comments, spacing or order.

The data returned from this function should not be used for direct modification of the main.cf file; other functions are available for that.

CLI Examples:
salt <minion> postfix.show_master salt <minion> postfix.show_master path=/path/to/master.cf

`salt.modules.postfix.`**`show_queue`**`()`

Show contents of the mail queue

CLI Example:

```
salt '*' postfix.show_queue
```

`salt.modules.postfix.`**`unhold`**`(`*queue_id*`)`

Set held message(s) in the mail queue to unheld

CLI Example:

```
salt '*' postfix.unhold 5C33CA0DEA

salt '*' postfix.unhold ALL
```

21.9.328 salt.modules.postgres

Module to provide Postgres compatibility to salt.

configuration In order to connect to Postgres, certain configuration is required in /etc/salt/minion on the relevant minions. Some sample configs might look like:

```
postgres.host: 'localhost'
postgres.port: '5432'
postgres.user: 'postgres' -> db user
postgres.pass: ''
postgres.maintenance_db: 'postgres'
```

The default for the maintenance_db is 'postgres' and in most cases it can be left at the default setting. This data can also be passed into pillar. Options passed into opts will overwrite options passed into pillar

note This module uses MD5 hashing which may not be compliant with certain security audits.

note When installing postgres from the official postgres repos, on certain linux distributions, either the psql or the initdb binary is *not* automatically placed on the path. Add a configuration to the location of the postgres bin's path to the relevant minion for this module:

```
postgres.bins_dir: '/usr/pgsql-9.5/bin/'
```

`salt.modules.postgres.`**`available_extensions`**(*user=None, host=None, port=None, maintenance_db=None, password=None, runas=None*)

List available postgresql extensions

CLI Example:

```
salt '*' postgres.available_extensions
```

`salt.modules.postgres.`**`create_extension`**(*name, if_not_exists=None, schema=None, ext_version=None, from_version=None, user=None, host=None, port=None, maintenance_db=None, password=None, runas=None*)

Install a postgresql extension

CLI Example:

```
salt '*' postgres.create_extension 'adminpack'
```

`salt.modules.postgres.`**`create_metadata`**(*name, ext_version=None, schema=None, user=None, host=None, port=None, maintenance_db=None, password=None, runas=None*)

Get lifecycle information about an extension

CLI Example:

```
salt '*' postgres.create_metadata adminpack
```

`salt.modules.postgres.`**`datadir_exists`**`(`*name*`)`

New in version 2016.3.0.

Checks if postgres data directory has been initialized

CLI Example:

```
salt '*' postgres.datadir_exists '/var/lib/pgsql/data'
```

name Name of the directory to check

`salt.modules.postgres.`**`datadir_init`**`(`*name, auth='password', user=None, password=None, encoding='UTF8', locale=None, waldir=None, checksums=False, runas=None*`)`

New in version 2016.3.0.

Initializes a postgres data directory

CLI Example:

```
salt '*' postgres.datadir_init '/var/lib/pgsql/data'
```

name The name of the directory to initialize
auth The default authentication method for local connections
password The password to set for the postgres user
user The database superuser name
encoding The default encoding for new databases
locale The default locale for new databases
waldir The transaction log (WAL) directory (default is to keep WAL inside the data directory)

New in version 2019.2.0.

checksums If True, the cluster will be created with data page checksums.

> **Note:** Data page checksums are supported since PostgreSQL 9.3.

New in version 2019.2.0.

runas The system user the operation should be performed on behalf of

`salt.modules.postgres.`**`db_alter`**`(`*name, user=None, host=None, port=None, maintenance_db=None, password=None, tablespace=None, owner=None, owner_recurse=False, runas=None*`)`

Change tablespace or/and owner of database.

CLI Example:

```
salt '*' postgres.db_alter dbname owner=otheruser
```

`salt.modules.postgres.`**`db_create`**`(`*name, user=None, host=None, port=None, maintenance_db=None, password=None, tablespace=None, encoding=None, lc_collate=None, lc_ctype=None, owner=None, template=None, runas=None*`)`

Adds a databases to the Postgres server.

CLI Example:

```
salt '*' postgres.db_create 'dbname'

salt '*' postgres.db_create 'dbname' template=template_postgis
```

`salt.modules.postgres.`**`db_exists`**(*name*, *user=None*, *host=None*, *port=None*, *maintenance_db=None*, *password=None*, *runas=None*)

Checks if a database exists on the Postgres server.

CLI Example:

```
salt '*' postgres.db_exists 'dbname'
```

`salt.modules.postgres.`**`db_list`**(*user=None*, *host=None*, *port=None*, *maintenance_db=None*, *password=None*, *runas=None*)

Return dictionary with information about databases of a Postgres server.

CLI Example:

```
salt '*' postgres.db_list
```

`salt.modules.postgres.`**`db_remove`**(*name*, *user=None*, *host=None*, *port=None*, *maintenance_db=None*, *password=None*, *runas=None*)

Removes a databases from the Postgres server.

CLI Example:

```
salt '*' postgres.db_remove 'dbname'
```

`salt.modules.postgres.`**`drop_extension`**(*name*, *if_exists=None*, *restrict=None*, *cascade=None*, *user=None*, *host=None*, *port=None*, *maintenance_db=None*, *password=None*, *runas=None*)

Drop an installed postgresql extension

CLI Example:

```
salt '*' postgres.drop_extension 'adminpack'
```

`salt.modules.postgres.`**`get_available_extension`**(*name*, *user=None*, *host=None*, *port=None*, *maintenance_db=None*, *password=None*, *runas=None*)

Get info about an available postgresql extension

CLI Example:

```
salt '*' postgres.get_available_extension plpgsql
```

`salt.modules.postgres.`**`get_installed_extension`**(*name*, *user=None*, *host=None*, *port=None*, *maintenance_db=None*, *password=None*, *runas=None*)

Get info about an installed postgresql extension

CLI Example:

```
salt '*' postgres.get_installed_extension plpgsql
```

`salt.modules.postgres.`**`group_create`**(*groupname*, *user=None*, *host=None*, *port=None*, *maintenance_db=None*, *password=None*, *createdb=None*, *createroles=None*, *encrypted=None*, *login=None*, *inherit=None*, *superuser=None*, *replication=None*, *rolepassword=None*, *groups=None*, *runas=None*)

Creates a Postgres group. A group is postgres is similar to a user, but cannot login.

CLI Example:

```
salt '*' postgres.group_create 'groupname' user='user' \
        host='hostname' port='port' password='password' \
        rolepassword='rolepassword'
```

`salt.modules.postgres.`**`group_remove`**(*groupname, user=None, host=None, port=None, maintenance_db=None, password=None, runas=None*)

Removes a group from the Postgres server.

CLI Example:

```
salt '*' postgres.group_remove 'groupname'
```

`salt.modules.postgres.`**`group_update`**(*groupname, user=None, host=None, port=None, maintenance_db=None, password=None, createdb=None, createroles=None, encrypted=None, inherit=None, login=None, superuser=None, replication=None, rolepassword=None, groups=None, runas=None*)

Updates a postgres group

CLI Examples:

```
salt '*' postgres.group_update 'username' user='user' \
        host='hostname' port='port' password='password' \
        rolepassword='rolepassword'
```

`salt.modules.postgres.`**`has_privileges`**(*name, object_name, object_type, privileges=None, grant_option=None, prepend='public', maintenance_db=None, user=None, host=None, port=None, password=None, runas=None*)

New in version 2016.3.0.

Check if a role has the specified privileges on an object

CLI Example:

```
salt '*' postgres.has_privileges user_name table_name table \
SELECT,INSERT maintenance_db=db_name
```

name Name of the role whose privileges should be checked on object_type

object_name Name of the object on which the check is to be performed

object_type The object type, which can be one of the following:

- table
- sequence
- schema
- tablespace
- language
- database
- group
- function

privileges Comma separated list of privileges to check, from the list below:

- INSERT
- CREATE
- TRUNCATE
- CONNECT
- TRIGGER
- SELECT
- USAGE

- TEMPORARY
- UPDATE
- EXECUTE
- REFERENCES
- DELETE
- ALL

grant_option If grant_option is set to True, the grant option check is performed
prepend Table and Sequence object types live under a schema so this should be provided if the object is not under the default *public* schema
maintenance_db The database to connect to
user database username if different from config or default
password user password if any password for a specified user
host Database host if different from config or default
port Database port if different from config or default
runas System user all operations should be performed on behalf of

`salt.modules.postgres.`**`installed_extensions`**(*user=None*, *host=None*, *port=None*, *maintenance_db=None*, *password=None*, *runas=None*)

List installed postgresql extensions

CLI Example:

```
salt '*' postgres.installed_extensions
```

`salt.modules.postgres.`**`is_available_extension`**(*name*, *user=None*, *host=None*, *port=None*, *maintenance_db=None*, *password=None*, *runas=None*)

Test if a specific extension is available

CLI Example:

```
salt '*' postgres.is_available_extension
```

`salt.modules.postgres.`**`is_installed_extension`**(*name*, *user=None*, *host=None*, *port=None*, *maintenance_db=None*, *password=None*, *runas=None*)

Test if a specific extension is installed

CLI Example:

```
salt '*' postgres.is_installed_extension
```

`salt.modules.postgres.`**`language_create`**(*name*, *maintenance_db*, *user=None*, *host=None*, *port=None*, *password=None*, *runas=None*)

New in version 2016.3.0.

Installs a language into a database

CLI Example:

```
salt '*' postgres.language_create plpgsql dbname
```

name Language to install
maintenance_db The database to install the language in
user database username if different from config or default
password user password if any password for a specified user
host Database host if different from config or default
port Database port if different from config or default

runas System user all operations should be performed on behalf of

`salt.modules.postgres.`**`language_exists`**(*name*, *maintenance_db*, *user=None*, *host=None*, *port=None*, *password=None*, *runas=None*)

New in version 2016.3.0.

Checks if language exists in a database.

CLI Example:

```
salt '*' postgres.language_exists plpgsql dbname
```

name Language to check for
maintenance_db The database to check in
user database username if different from config or default
password user password if any password for a specified user
host Database host if different from config or default
port Database port if different from config or default
runas System user all operations should be performed on behalf of

`salt.modules.postgres.`**`language_list`**(*maintenance_db*, *user=None*, *host=None*, *port=None*, *password=None*, *runas=None*)

New in version 2016.3.0.

Return a list of languages in a database.

CLI Example:

```
salt '*' postgres.language_list dbname
```

maintenance_db The database to check
user database username if different from config or default
password user password if any password for a specified user
host Database host if different from config or default
port Database port if different from config or default
runas System user all operations should be performed on behalf of

`salt.modules.postgres.`**`language_remove`**(*name*, *maintenance_db*, *user=None*, *host=None*, *port=None*, *password=None*, *runas=None*)

New in version 2016.3.0.

Removes a language from a database

CLI Example:

```
salt '*' postgres.language_remove plpgsql dbname
```

name Language to remove
maintenance_db The database to install the language in
user database username if different from config or default
password user password if any password for a specified user
host Database host if different from config or default
port Database port if different from config or default
runas System user all operations should be performed on behalf of

`salt.modules.postgres.`**`owner_to`**(*dbname*, *ownername*, *user=None*, *host=None*, *port=None*, *password=None*, *runas=None*)

Set the owner of all schemas, functions, tables, views and sequences to the given username.

CLI Example:

```
salt '*' postgres.owner_to 'dbname' 'username'
```

salt.modules.postgres.**privileges_grant**(*name*, *object_name*, *object_type*, *privileges=None*, *grant_option=None*, *prepend='public'*, *maintenance_db=None*, *user=None*, *host=None*, *port=None*, *password=None*, *runas=None*)

New in version 2016.3.0.

Grant privileges on a postgres object

CLI Example:

```
salt '*' postgres.privileges_grant user_name table_name table \
SELECT,UPDATE maintenance_db=db_name
```

name Name of the role to which privileges should be granted
object_name Name of the object on which the grant is to be performed
object_type The object type, which can be one of the following:

- table
- sequence
- schema
- tablespace
- language
- database
- group
- function

privileges Comma separated list of privileges to grant, from the list below:

- INSERT
- CREATE
- TRUNCATE
- CONNECT
- TRIGGER
- SELECT
- USAGE
- TEMPORARY
- UPDATE
- EXECUTE
- REFERENCES
- DELETE
- ALL

grant_option If grant_option is set to True, the recipient of the privilege can in turn grant it to others
prepend Table and Sequence object types live under a schema so this should be provided if the object is not under the default *public* schema
maintenance_db The database to connect to
user database username if different from config or default
password user password if any password for a specified user
host Database host if different from config or default
port Database port if different from config or default
runas System user all operations should be performed on behalf of

salt.modules.postgres.**privileges_list**(*name*, *object_type*, *prepend='public'*, *maintenance_db=None*, *user=None*, *host=None*, *port=None*, *password=None*, *runas=None*)

New in version 2016.3.0.

Return a list of privileges for the specified object.

CLI Example:

```
salt '*' postgres.privileges_list table_name table maintenance_db=db_name
```

name Name of the object for which the permissions should be returned

object_type The object type, which can be one of the following:

- table
- sequence
- schema
- tablespace
- language
- database
- group
- function

prepend Table and Sequence object types live under a schema so this should be provided if the object is not under the default *public* schema

maintenance_db The database to connect to

user database username if different from config or default

password user password if any password for a specified user

host Database host if different from config or default

port Database port if different from config or default

runas System user all operations should be performed on behalf of

salt.modules.postgres.**privileges_revoke**(*name*, *object_name*, *object_type*, *privileges=None*, *prepend='public'*, *maintenance_db=None*, *user=None*, *host=None*, *port=None*, *password=None*, *runas=None*)

New in version 2016.3.0.

Revoke privileges on a postgres object

CLI Example:

```
salt '*' postgres.privileges_revoke user_name table_name table \
SELECT,UPDATE maintenance_db=db_name
```

name Name of the role whose privileges should be revoked

object_name Name of the object on which the revoke is to be performed

object_type The object type, which can be one of the following:

- table
- sequence
- schema
- tablespace
- language
- database
- group
- function

privileges Comma separated list of privileges to revoke, from the list below:

- INSERT
- CREATE
- TRUNCATE
- CONNECT
- TRIGGER
- SELECT
- USAGE
- TEMPORARY

- UPDATE
- EXECUTE
- REFERENCES
- DELETE
- ALL

maintenance_db The database to connect to
user database username if different from config or default
password user password if any password for a specified user
host Database host if different from config or default
port Database port if different from config or default
runas System user all operations should be performed on behalf of

`salt.modules.postgres.`**`psql_query`**(*query, user=None, host=None, port=None, maintenance_db=None, password=None, runas=None, write=False*)

Run an SQL-Query and return the results as a list. This command only supports SELECT statements. This limitation can be worked around with a query like this:

WITH updated AS (UPDATE pg_authid SET rolconnlimit = 2000 WHERE rolname = 'rolename' RETURNING rolconnlimit) SELECT * FROM updated;

query The query string.
user Database username, if different from config or default.
host Database host, if different from config or default.
port Database port, if different from the config or default.
maintenance_db The database to run the query against.
password User password, if different from the config or default.
runas User to run the command as.
write Mark query as READ WRITE transaction.
CLI Example:

```
salt '*' postgres.psql_query 'select * from pg_stat_activity'
```

`salt.modules.postgres.`**`role_get`**(*name, user=None, host=None, port=None, maintenance_db=None, password=None, runas=None, return_password=False*)

Return a dict with information about users of a Postgres server.

Set return_password to True to get password hash in the result.

CLI Example:

```
salt '*' postgres.role_get postgres
```

`salt.modules.postgres.`**`schema_create`**(*dbname, name, owner=None, user=None, db_user=None, db_password=None, db_host=None, db_port=None*)

Creates a Postgres schema.

CLI Example:

```
salt '*' postgres.schema_create dbname name owner='owner' \
        user='user' \
        db_user='user' db_password='password'
        db_host='hostname' db_port='port'
```

`salt.modules.postgres.`**`schema_exists`**(*dbname, name, user=None, db_user=None, db_password=None, db_host=None, db_port=None*)

Checks if a schema exists on the Postgres server.

CLI Example:

```
salt '*' postgres.schema_exists dbname schemaname
```

dbname Database name we query on
name Schema name we look for
user The system user the operation should be performed on behalf of
db_user database username if different from config or default
db_password user password if any password for a specified user
db_host Database host if different from config or default
db_port Database port if different from config or default

`salt.modules.postgres.`**`schema_get`**(*dbname*, *name*, *user=None*, *db_user=None*, *db_password=None*, *db_host=None*, *db_port=None*)

Return a dict with information about schemas in a database.

CLI Example:

```
salt '*' postgres.schema_get dbname name
```

dbname Database name we query on
name Schema name we look for
user The system user the operation should be performed on behalf of
db_user database username if different from config or default
db_password user password if any password for a specified user
db_host Database host if different from config or default
db_port Database port if different from config or default

`salt.modules.postgres.`**`schema_list`**(*dbname*, *user=None*, *db_user=None*, *db_password=None*, *db_host=None*, *db_port=None*)

Return a dict with information about schemas in a Postgres database.

CLI Example:

```
salt '*' postgres.schema_list dbname
```

dbname Database name we query on
user The system user the operation should be performed on behalf of
db_user database username if different from config or default
db_password user password if any password for a specified user
db_host Database host if different from config or default
db_port Database port if different from config or default

`salt.modules.postgres.`**`schema_remove`**(*dbname*, *name*, *user=None*, *db_user=None*, *db_password=None*, *db_host=None*, *db_port=None*)

Removes a schema from the Postgres server.

CLI Example:

```
salt '*' postgres.schema_remove dbname schemaname
```

dbname Database name we work on
schemaname The schema's name we'll remove
user System user all operations should be performed on behalf of
db_user database username if different from config or default
db_password user password if any password for a specified user
db_host Database host if different from config or default
db_port Database port if different from config or default

`salt.modules.postgres.`**`tablespace_alter`**(*name, user=None, host=None, port=None, maintenance_db=None, password=None, new_name=None, new_owner=None, set_option=None, reset_option=None, runas=None*)

Change tablespace name, owner, or options.

CLI Example:

```
salt '*' postgres.tablespace_alter tsname new_owner=otheruser
salt '*' postgres.tablespace_alter index_space new_name=fast_raid
salt '*' postgres.tablespace_alter test set_option="{'seq_page_cost': '1.1'}"
salt '*' postgres.tablespace_alter tsname reset_option=seq_page_cost
```

New in version 2015.8.0.

`salt.modules.postgres.`**`tablespace_create`**(*name, location, options=None, owner=None, user=None, host=None, port=None, maintenance_db=None, password=None, runas=None*)

Adds a tablespace to the Postgres server.

CLI Example:

```
salt '*' postgres.tablespace_create tablespacename '/path/datadir'
```

New in version 2015.8.0.

`salt.modules.postgres.`**`tablespace_exists`**(*name, user=None, host=None, port=None, maintenance_db=None, password=None, runas=None*)

Checks if a tablespace exists on the Postgres server.

CLI Example:

```
salt '*' postgres.tablespace_exists 'dbname'
```

New in version 2015.8.0.

`salt.modules.postgres.`**`tablespace_list`**(*user=None, host=None, port=None, maintenance_db=None, password=None, runas=None*)

Return dictionary with information about tablespaces of a Postgres server.

CLI Example:

```
salt '*' postgres.tablespace_list
```

New in version 2015.8.0.

`salt.modules.postgres.`**`tablespace_remove`**(*name, user=None, host=None, port=None, maintenance_db=None, password=None, runas=None*)

Removes a tablespace from the Postgres server.

CLI Example:

```
salt '*' postgres.tablespace_remove tsname
```

New in version 2015.8.0.

`salt.modules.postgres.`**`user_create`**(*username, user=None, host=None, port=None, maintenance_db=None, password=None, createdb=None, createroles=None, inherit=None, login=None, connlimit=None, encrypted=None, superuser=None, replication=None, rolepassword=None, valid_until=None, groups=None, runas=None*)

Creates a Postgres user.

CLI Examples:

```
salt '*' postgres.user_create 'username' user='user' \
        host='hostname' port='port' password='password' \
        rolepassword='rolepassword' valid_until='valid_until'
```

`salt.modules.postgres.`**`user_exists`**(*name, user=None, host=None, port=None, maintenance_db=None, password=None, runas=None*)

Checks if a user exists on the Postgres server.

CLI Example:

```
salt '*' postgres.user_exists 'username'
```

`salt.modules.postgres.`**`user_list`**(*user=None, host=None, port=None, maintenance_db=None, password=None, runas=None, return_password=False*)

Return a dict with information about users of a Postgres server.

Set return_password to True to get password hash in the result.

CLI Example:

```
salt '*' postgres.user_list
```

`salt.modules.postgres.`**`user_remove`**(*username, user=None, host=None, port=None, maintenance_db=None, password=None, runas=None*)

Removes a user from the Postgres server.

CLI Example:

```
salt '*' postgres.user_remove 'username'
```

`salt.modules.postgres.`**`user_update`**(*username, user=None, host=None, port=None, maintenance_db=None, password=None, createdb=None, createroles=None, encrypted=None, superuser=None, inherit=None, login=None, connlimit=None, replication=None, rolepassword=None, valid_until=None, groups=None, runas=None*)

Updates a Postgres user.

CLI Examples:

```
salt '*' postgres.user_update 'username' user='user' \
        host='hostname' port='port' password='password' \
        rolepassword='rolepassword' valid_until='valid_until'
```

`salt.modules.postgres.`**`version`**(*user=None, host=None, port=None, maintenance_db=None, password=None, runas=None*)

Return the version of a Postgres server.

CLI Example:

```
salt '*' postgres.version
```

21.9.329 salt.modules.poudriere

Support for poudriere

`salt.modules.poudriere.`**`bulk_build`**`(`*jail*, *pkg_file*, *keep=False*`)`

Run bulk build on poudriere server.

Return number of pkg builds, failures, and errors, on error dump to CLI

CLI Example:

```
salt -N buildbox_group poudriere.bulk_build 90amd64 /root/pkg_list
```

`salt.modules.poudriere.`**`create_jail`**`(`*name*, *arch*, *version='9.0-RELEASE'*`)`

Creates a new poudriere jail if one does not exist

NOTE creating a new jail will take some time the master is not hanging

CLI Example:

```
salt '*' poudriere.create_jail 90amd64 amd64
```

`salt.modules.poudriere.`**`create_ports_tree`**`()`

Not working need to run portfetch non interactive

`salt.modules.poudriere.`**`delete_jail`**`(`*name*`)`

Deletes poudriere jail with *name*

CLI Example:

```
salt '*' poudriere.delete_jail 90amd64
```

`salt.modules.poudriere.`**`is_jail`**`(`*name*`)`

Return True if jail exists False if not

CLI Example:

```
salt '*' poudriere.is_jail <jail name>
```

`salt.modules.poudriere.`**`list_jails`**`()`

Return a list of current jails managed by poudriere

CLI Example:

```
salt '*' poudriere.list_jails
```

`salt.modules.poudriere.`**`list_ports`**`()`

Return a list of current port trees managed by poudriere

CLI Example:

```
salt '*' poudriere.list_ports
```

`salt.modules.poudriere.`**`make_pkgng_aware`**`(`*jname*`)`

Make jail `jname` pkgng aware

CLI Example:

```
salt '*' poudriere.make_pkgng_aware <jail name>
```

`salt.modules.poudriere.`**`parse_config`**`(`*config_file=None*`)`

Returns a dict of poudriere main configuration definitions

CLI Example:

```
salt '*' poudriere.parse_config
```

`salt.modules.poudriere.`**`update_jail`**`(`*name*`)`

Run freebsd-update on *name* poudriere jail

CLI Example:

```
salt '*' poudriere.update_jail freebsd:10:x86:64
```

`salt.modules.poudriere.`**`update_ports_tree`**`(`*ports_tree*`)`

Updates the ports tree, either the default or the *ports_tree* specified

CLI Example:

```
salt '*' poudriere.update_ports_tree staging
```

`salt.modules.poudriere.`**`version`**`()`

Return poudriere version

CLI Example:

```
salt '*' poudriere.version
```

21.9.330 salt.modules.powerpath

powerpath support.

Assumes RedHat

`salt.modules.powerpath.`**`add_license`**`(`*key*`)`

Add a license

`salt.modules.powerpath.`**`has_powerpath`**`()`

`salt.modules.powerpath.`**`list_licenses`**`()`

returns a list of applied powerpath license keys

`salt.modules.powerpath.`**`remove_license`**`(`*key*`)`

Remove a license

21.9.331 salt.modules.proxy module

This module allows you to manage proxy settings

```
salt '*' network.get_http_proxy
```

`salt.modules.proxy.`**`get_ftp_proxy`**`(`*network_service='Ethernet'*`)`

Returns the current ftp proxy settings

network_service The network service to apply the changes to, this only necessary on macOS

CLI Example:

```
salt '*' proxy.get_ftp_proxy Ethernet
```

`salt.modules.proxy.`**`get_http_proxy`**`(`*network_service='Ethernet'*`)`

Returns the current http proxy settings

network_service The network service to apply the changes to, this only necessary on macOS

CLI Example:

```
salt '*' proxy.get_http_proxy Ethernet
```

`salt.modules.proxy.`**`get_https_proxy`**`(`*network_service='Ethernet'*`)`

Returns the current https proxy settings

network_service The network service to apply the changes to, this only necessary on macOS

CLI Example:

```
salt '*' proxy.get_https_proxy Ethernet
```

`salt.modules.proxy.`**`get_proxy_bypass`**`(`*network_service='Ethernet'*`)`

Returns the current domains that can bypass the proxy

network_service The network service to get the bypass domains from, this is only necessary on macOS

CLI Example:

```
salt '*' proxy.get_proxy_bypass
```

`salt.modules.proxy.`**`get_proxy_win`**`()`

Gets all of the proxy settings in one call, only available on Windows

CLI Example:

```
salt '*' proxy.get_proxy_win
```

`salt.modules.proxy.`**`set_ftp_proxy`**`(`*server, port, user=None, password=None, network_service='Ethernet', bypass_hosts=None*`)`

Sets the ftp proxy settings

server The proxy server to use

port The port used by the proxy server

user The username to use for the proxy server if required

password The password to use if required by the server

network_service The network service to apply the changes to, this only necessary on macOS

bypass_hosts The hosts that are allowed to by pass the proxy. Only used on Windows for other OS's use set_proxy_bypass to edit the bypass hosts.

CLI Example:

```
salt '*' proxy.set_ftp_proxy example.com 1080 user=proxy_user password=proxy_pass␣
↪network_service=Ethernet
```

`salt.modules.proxy.`**`set_http_proxy`**`(`*server, port, user=None, password=None, network_service='Ethernet', bypass_hosts=None*`)`

Sets the http proxy settings. Note: On Windows this will override any other proxy settings you have, the preferred method of updating proxies on windows is using set_proxy.

server The proxy server to use

port The port used by the proxy server

user The username to use for the proxy server if required

password The password to use if required by the server

network_service The network service to apply the changes to, this only necessary on macOS

bypass_hosts The hosts that are allowed to by pass the proxy. Only used on Windows for other OS's use set_proxy_bypass to edit the bypass hosts.

CLI Example:

```
salt '*' proxy.set_http_proxy example.com 1080 user=proxy_user password=proxy_
↪pass network_service=Ethernet
```

`salt.modules.proxy.`**`set_https_proxy`**`(`*server, port, user=None, password=None, network_service='Ethernet', bypass_hosts=None*`)`

Sets the https proxy settings. Note: On Windows this will override any other proxy settings you have, the preferred method of updating proxies on windows is using set_proxy.

server The proxy server to use

port The port used by the proxy server

user The username to use for the proxy server if required

password The password to use if required by the server

network_service The network service to apply the changes to, this only necessary on macOS

bypass_hosts The hosts that are allowed to by pass the proxy. Only used on Windows for other OS's use set_proxy_bypass to edit the bypass hosts.

CLI Example:

```
salt '*' proxy.set_https_proxy example.com 1080 user=proxy_user password=proxy_
↪pass network_service=Ethernet
```

`salt.modules.proxy.`**`set_proxy_bypass`**(*domains*, *network_service='Ethernet'*)

Sets the domains that can bypass the proxy

domains An array of domains allowed to bypass the proxy

network_service The network service to apply the changes to, this only necessary on macOS

CLI Example:

```
salt '*' proxy.set_proxy_bypass "['127.0.0.1', 'localhost']"
```

`salt.modules.proxy.`**`set_proxy_win`**(*server*, *port*, *types=None*, *bypass_hosts=None*)

Sets the http proxy settings, only works with Windows.

server The proxy server to use

password The password to use if required by the server

types The types of proxy connections should be setup with this server. Valid types are:

- `http`
- `https`
- `ftp`

bypass_hosts The hosts that are allowed to by pass the proxy.

CLI Example:

```
salt '*' proxy.set_http_proxy example.com 1080 types="['http', 'https']"
```

21.9.332 salt.modules.ps

A salt interface to psutil, a system and process library. See http://code.google.com/p/psutil.

depends

- psutil Python module, version 0.3.0 or later
- python-utmp package (optional)

`salt.modules.ps.`**`boot_time`**(*time_format=None*)

Return the boot time in number of seconds since the epoch began.

CLI Example:

time_format Optionally specify a strftime format string. Use `time_format='%c'` to get a nicely-formatted locale specific date and time (i.e. `Fri May 2 19:08:32 2014`).

New in version 2014.1.4.

```
salt '*' ps.boot_time
```

salt.modules.ps.**cpu_percent**(*interval=0.1*, *per_cpu=False*)

Return the percent of time the CPU is busy.

interval the number of seconds to sample CPU usage over

per_cpu if True return an array of CPU percent busy for each CPU, otherwise aggregate all percents into one number

CLI Example:

```
salt '*' ps.cpu_percent
```

salt.modules.ps.**cpu_times**(*per_cpu=False*)

Return the percent of time the CPU spends in each state, e.g. user, system, idle, nice, iowait, irq, softirq.

per_cpu if True return an array of percents for each CPU, otherwise aggregate all percents into one number

CLI Example:

```
salt '*' ps.cpu_times
```

salt.modules.ps.**disk_io_counters**(*device=None*)

Return disk I/O statistics.

CLI Example:

```
salt '*' ps.disk_io_counters

salt '*' ps.disk_io_counters device=sda1
```

salt.modules.ps.**disk_partition_usage**(*all=False*)

Return a list of disk partitions plus the mount point, filesystem and usage statistics.

CLI Example:

```
salt '*' ps.disk_partition_usage
```

salt.modules.ps.**disk_partitions**(*all=False*)

Return a list of disk partitions and their device, mount point, and filesystem type.

all if set to False, only return local, physical partitions (hard disk, USB, CD/DVD partitions). If True, return all filesystems.

CLI Example:

```
salt '*' ps.disk_partitions
```

salt.modules.ps.**disk_usage**(*path*)

Given a path, return a dict listing the total available space as well as the free space, and used space.

CLI Example:

```
salt '*' ps.disk_usage /home
```

salt.modules.ps.**get_pid_list**()

Return a list of process ids (PIDs) for all running processes.

CLI Example:

```
salt '*' ps.get_pid_list
```

salt.modules.ps.**get_users**()

Return logged-in users.

CLI Example:

```
salt '*' ps.get_users
```

`salt.modules.ps.`**`kill_pid`**`(`*pid, signal=15*`)`

Kill a process by PID.

```
salt 'minion' ps.kill_pid pid [signal=signal_number]
```

pid PID of process to kill.

signal Signal to send to the process. See manpage entry for kill for possible values. Default: 15 (SIGTERM).

Example:

Send SIGKILL to process with PID 2000:

```
salt 'minion' ps.kill_pid 2000 signal=9
```

`salt.modules.ps.`**`lsof`**`(`*name*`)`

Retrieve the lsof information of the given process name.

CLI Example:

```
salt '*' ps.lsof apache2
```

`salt.modules.ps.`**`netstat`**`(`*name*`)`

Retrieve the netstat information of the given process name.

CLI Example:

```
salt '*' ps.netstat apache2
```

`salt.modules.ps.`**`network_io_counters`**`(`*interface=None*`)`

Return network I/O statistics.

CLI Example:

```
salt '*' ps.network_io_counters

salt '*' ps.network_io_counters interface=eth0
```

`salt.modules.ps.`**`num_cpus`**`()`

Return the number of CPUs.

CLI Example:

```
salt '*' ps.num_cpus
```

`salt.modules.ps.`**`pgrep`**`(`*pattern, user=None, full=False*`)`

Return the pids for processes matching a pattern.

If full is true, the full command line is searched for a match, otherwise only the name of the command is searched.

```
salt '*' ps.pgrep pattern [user=username] [full=(true|false)]
```

pattern Pattern to search for in the process list.

user Limit matches to the given username. Default: All users.

full A boolean value indicating whether only the name of the command or the full command line should be matched against the pattern.

Examples:

Find all httpd processes on all 'www' minions:

```
salt 'www.*' ps.pgrep httpd
```

Find all bash processes owned by user 'tom':

```
salt '*' ps.pgrep bash user=tom
```

`salt.modules.ps.`**`pkill`**(*pattern*, *user=None*, *signal=15*, *full=False*)

Kill processes matching a pattern.

```
salt '*' ps.pkill pattern [user=username] [signal=signal_number] \
        [full=(true|false)]
```

pattern Pattern to search for in the process list.

user Limit matches to the given username. Default: All users.

signal Signal to send to the process(es). See manpage entry for kill for possible values. Default: 15 (SIGTERM).

full A boolean value indicating whether only the name of the command or the full command line should be matched against the pattern.

Examples:

Send SIGHUP to all httpd processes on all 'www' minions:

```
salt 'www.*' ps.pkill httpd signal=1
```

Send SIGKILL to all bash processes owned by user 'tom':

```
salt '*' ps.pkill bash signal=9 user=tom
```

`salt.modules.ps.`**`proc_info`**(*pid*, *attrs=None*)

Return a dictionary of information for a process id (PID).

CLI Example:

```
salt '*' ps.proc_info 2322
salt '*' ps.proc_info 2322 attrs='["pid", "name"]'
```

pid PID of process to query.

attrs Optional list of desired process attributes. The list of possible attributes can be found here: http://pythonhosted.org/psutil/#psutil.Process

`salt.modules.ps.`**`psaux`**(*name*)

Retrieve information corresponding to a "ps aux" filtered with the given pattern. It could be just a name or a regular expression (using python search from "re" module).

CLI Example:

```
salt '*' ps.psaux www-data.+apache2
```

`salt.modules.ps.`**`ss`**(*name*)

Retrieve the ss information of the given process name.

CLI Example:

```
salt '*' ps.ss apache2
```

New in version 2016.11.6.

`salt.modules.ps.`**`swap_memory`**`()`

New in version 2014.7.0.

Return a dict that describes swap memory statistics.

Note: This function is only available in psutil version 0.6.0 and above.

CLI Example:

```
salt '*' ps.swap_memory
```

`salt.modules.ps.`**`top`**`(`*num_processes=5, interval=3*`)`

Return a list of top CPU consuming processes during the interval. num_processes = return the top N CPU consuming processes interval = the number of seconds to sample CPU usage over

CLI Examples:

```
salt '*' ps.top

salt '*' ps.top 5 10
```

`salt.modules.ps.`**`total_physical_memory`**`()`

Return the total number of bytes of physical memory.

CLI Example:

```
salt '*' ps.total_physical_memory
```

`salt.modules.ps.`**`virtual_memory`**`()`

New in version 2014.7.0.

Return a dict that describes statistics about system memory usage.

Note: This function is only available in psutil version 0.6.0 and above.

CLI Example:

```
salt '*' ps.virtual_memory
```

21.9.333 salt.modules.publish

Publish a command from a minion to a target

`salt.modules.publish.`**`full_data`**`(`*tgt, fun, arg=None, tgt_type='glob', returner='', timeout=5*`)`

Return the full data about the publication, this is invoked in the same way as the publish function

CLI Example:

```
salt system.example.com publish.full_data '*' cmd.run 'ls -la /tmp'
```

Attention

If you need to pass a value to a function argument and that value contains an equal sign, you **must** include the argument name. For example:

```
salt '*' publish.full_data test.kwarg arg='cheese=spam'
```

`salt.modules.publish.`**`publish`**(*tgt*, *fun*, *arg=None*, *tgt_type='glob'*, *returner=''*, *timeout=5*, *via_master=None*)

Publish a command from the minion out to other minions.

Publications need to be enabled on the Salt master and the minion needs to have permission to publish the command. The Salt master will also prevent a recursive publication loop, this means that a minion cannot command another minion to command another minion as that would create an infinite command loop.

The `tgt_type` argument is used to pass a target other than a glob into the execution, the available options are:

- glob
- pcre
- grain
- grain_pcre
- pillar
- pillar_pcre
- ipcidr
- range
- compound

Changed in version 2017.7.0: The `expr_form` argument has been renamed to `tgt_type`, earlier releases must use `expr_form`.

Note that for pillar matches must be exact, both in the pillar matcher and the compound matcher. No globbing is supported.

The arguments sent to the minion publish function are separated with commas. This means that for a minion executing a command with multiple args it will look like this:

```
salt system.example.com publish.publish '*' user.add 'foo,1020,1020'
salt system.example.com publish.publish 'os:Fedora' network.interfaces '' grain
```

CLI Example:

```
salt system.example.com publish.publish '*' cmd.run 'ls -la /tmp'
```

Attention

If you need to pass a value to a function argument and that value contains an equal sign, you **must** include the argument name. For example:

```
salt '*' publish.publish test.kwarg arg='cheese=spam'
```

Multiple keyword arguments should be passed as a list.

```
salt '*' publish.publish test.kwarg arg="['cheese=spam','spam=cheese']"
```

When running via salt-call, the *via_master* flag may be set to specific which master the publication should be sent to. Only one master may be specified. If unset, the publication will be sent only to the first master in minion configuration.

`salt.modules.publish.`**`runner`**(*fun*, *arg=None*, *timeout=5*)

Execute a runner on the master and return the data from the runner function

CLI Example:

```
salt publish.runner manage.down
```

21.9.334 salt.modules.puppet

Execute puppet routines

`salt.modules.puppet.`**`disable`**`(`*message=None*`)`

New in version 2014.7.0.

Disable the puppet agent

message New in version 2015.5.2.

Disable message to send to puppet

CLI Example:

```
salt '*' puppet.disable
salt '*' puppet.disable 'Disabled, contact XYZ before enabling'
```

`salt.modules.puppet.`**`enable`**`()`

New in version 2014.7.0.

Enable the puppet agent

CLI Example:

```
salt '*' puppet.enable
```

`salt.modules.puppet.`**`fact`**`(`*name*, *puppet=False*`)`

Run facter for a specific fact

CLI Example:

```
salt '*' puppet.fact kernel
```

`salt.modules.puppet.`**`facts`**`(`*puppet=False*`)`

Run facter and return the results

CLI Example:

```
salt '*' puppet.facts
```

`salt.modules.puppet.`**`noop`**`(`**args*, ***kwargs*`)`

Execute a puppet noop run and return a dict with the stderr, stdout, return code, etc. Usage is the same as for puppet.run.

CLI Example:

```
salt '*' puppet.noop
salt '*' puppet.noop tags=basefiles::edit,apache::server
salt '*' puppet.noop debug
salt '*' puppet.noop apply /a/b/manifest.pp modulepath=/a/b/modules␣
↪tags=basefiles::edit,apache::server
```

`salt.modules.puppet.`**`plugin_sync`**`()`

Runs a plugin sync between the puppet master and agent

CLI Example:

```
salt '*' puppet.plugin_sync
```

`salt.modules.puppet.`**`run`**`(`**args, **kwargs*`)`

Execute a puppet run and return a dict with the stderr, stdout, return code, etc. The first positional argument given is checked as a subcommand. Following positional arguments should be ordered with arguments required by the subcommand first, followed by non-keyword arguments. Tags are specified by a tag keyword and comma separated list of values. – http://docs.puppetlabs.com/puppet/latest/reference/lang_tags.html

CLI Examples:

```
salt '*' puppet.run
salt '*' puppet.run tags=basefiles::edit,apache::server
salt '*' puppet.run agent onetime no-daemonize no-usecacheonfailure no-splay␣
↪ignorecache
salt '*' puppet.run debug
salt '*' puppet.run apply /a/b/manifest.pp modulepath=/a/b/modules␣
↪tags=basefiles::edit,apache::server
```

`salt.modules.puppet.`**`status`**`()`

New in version 2014.7.0.

Display puppet agent status

CLI Example:

```
salt '*' puppet.status
```

`salt.modules.puppet.`**`summary`**`()`

New in version 2014.7.0.

Show a summary of the last puppet agent run

CLI Example:

```
salt '*' puppet.summary
```

21.9.335 salt.modules.purefa

Management of Pure Storage FlashArray

Installation Prerequisites

- You will need the `purestorage` python package in your python installation path that is running salt.

```
pip install purestorage
```

- Configure Pure Storage FlashArray authentication. Use one of the following three methods.

  1) From the minion config

```
pure_tags:
  fa:
    san_ip: management vip or hostname for the FlashArray
    api_token: A valid api token for the FlashArray being managed
```

  2) From environment (PUREFA_IP and PUREFA_API)

3) From the pillar (PUREFA_IP and PUREFA_API)

maintainer Simon Dodsley (simon@purestorage.com)

maturity new

requires purestorage

platform all

New in version 2018.3.0.

`salt.modules.purefa.`**`hg_create`**`(`*name*, *host=None*, *volume=None*`)`

Create a hostgroup on a Pure Storage FlashArray.

Will return False if hostgroup already exists, or if named host or volume do not exist.

New in version 2018.3.0.
name [string] name of hostgroup (truncated to 63 characters)
host [string] name of host to add to hostgroup
volume [string] name of volume to add to hostgroup
CLI Example:

```
salt '*' purefa.hg_create foo host=bar volume=vol
```

`salt.modules.purefa.`**`hg_delete`**`(`*name*`)`

Delete a hostgroup on a Pure Storage FlashArray (removes all volumes and hosts).

Will return False is hostgroup is already in a deleted state.

New in version 2018.3.0.
name [string] name of hostgroup
CLI Example:

```
salt '*' purefa.hg_delete foo
```

`salt.modules.purefa.`**`hg_remove`**`(`*name*, *volume=None*, *host=None*`)`

Remove a host and/or volume from a hostgroup on a Pure Storage FlashArray.

Will return False is hostgroup does not exist, or named host or volume are not in the hostgroup.

New in version 2018.3.0.
name [string] name of hostgroup
volume [string] name of volume to remove from hostgroup
host [string] name of host to remove from hostgroup
CLI Example:

```
salt '*' purefa.hg_remove foo volume=test host=bar
```

`salt.modules.purefa.`**`hg_update`**`(`*name*, *host=None*, *volume=None*`)`

Adds entries to a hostgroup on a Pure Storage FlashArray.

Will return False is hostgroup doesn't exist, or host or volume do not exist.

New in version 2018.3.0.
name [string] name of hostgroup
host [string] name of host to add to hostgroup
volume [string] name of volume to add to hostgroup
CLI Example:

```
salt '*' purefa.hg_update foo host=bar volume=vol
```

`salt.modules.purefa.`**`host_create`**`(`*name*, *iqn=None*, *wwn=None*`)`

Add a host on a Pure Storage FlashArray.

Will return False if host already exists, or the iSCSI or Fibre Channel parameters are not in a valid format. See Pure Storage FlashArray documentation.

New in version 2018.3.0.
name [string] name of host (truncated to 63 characters)
iqn [string] iSCSI IQN of host
wwn [string] Fibre Channel WWN of host
CLI Example:

```
salt '*' purefa.host_create foo iqn='<Valid iSCSI IQN>' wwn='<Valid WWN>'
```

`salt.modules.purefa.`**`host_delete`**`(`*name*`)`

Delete a host on a Pure Storage FlashArray (detaches all volumes).

Will return False if the host doesn't exist.

New in version 2018.3.0.
name [string] name of host
CLI Example:

```
salt '*' purefa.host_delete foo
```

`salt.modules.purefa.`**`host_update`**`(`*name*, *iqn=None*, *wwn=None*`)`

Update a hosts port definitions on a Pure Storage FlashArray.

Will return False if new port definitions are already in use by another host, or are not in a valid format. See Pure Storage FlashArray documentation.

New in version 2018.3.0.
name [string] name of host
iqn [string] Additional iSCSI IQN of host
wwn [string] Additional Fibre Channel WWN of host
CLI Example:

```
salt '*' purefa.host_update foo iqn='<Valid iSCSI IQN>' wwn='<Valid WWN>'
```

`salt.modules.purefa.`**`pg_create`**`(`*name*, *hostgroup=None*, *host=None*, *volume=None*, *enabled=True*`)`

Create a protection group on a Pure Storage FlashArray.
Will return False is the following cases:

- Protection Grop already exists
- Protection Group in a deleted state
- More than one type is specified - protection groups are for only hostgroups, hosts or volumes
- Named type for protection group does not exist

New in version 2018.3.0.
name [string] name of protection group
hostgroup [string] name of hostgroup to add to protection group
host [string] name of host to add to protection group
volume [string] name of volume to add to protection group
CLI Example:

```
salt '*' purefa.pg_create foo [hostgroup=foo | host=bar | volume=vol]␣
↪enabled=[true | false]
```

`salt.modules.purefa.`**`pg_delete`**`(`*name*, *eradicate=False*`)`

Delete a protecton group on a Pure Storage FlashArray.

Will return False if protection group is already in a deleted state.

New in version 2018.3.0.
name [string] name of protection group
CLI Example:

```
salt '*' purefa.pg_delete foo
```

`salt.modules.purefa.`**`pg_eradicate`**`(`*name*`)`
Eradicate a deleted protecton group on a Pure Storage FlashArray.

Will return False if protection group is not in a deleted state.

New in version 2018.3.0.
name [string] name of protection group
CLI Example:

```
salt '*' purefa.pg_eradicate foo
```

`salt.modules.purefa.`**`pg_remove`**`(`*name, hostgroup=None, host=None, volume=None*`)`
Remove a hostgroup, host or volume from a protection group on a Pure Storage FlashArray.
Will return False in the following cases:

- Protection group does not exist
- Specified type is not currently associated with the protection group

New in version 2018.3.0.
name [string] name of hostgroup
hostgroup [string] name of hostgroup to remove from protection group
host [string] name of host to remove from hostgroup
volume [string] name of volume to remove from hostgroup
CLI Example:

```
salt '*' purefa.pg_remove foo [hostgroup=bar | host=test | volume=bar]
```

`salt.modules.purefa.`**`pg_update`**`(`*name, hostgroup=None, host=None, volume=None*`)`
Update a protection group on a Pure Storage FlashArray.
Will return False in the following cases:

- Protection group does not exist
- Incorrect type selected for current protection group type
- Specified type does not exist

New in version 2018.3.0.
name [string] name of protection group
hostgroup [string] name of hostgroup to add to protection group
host [string] name of host to add to protection group
volume [string] name of volume to add to protection group
CLI Example:

```
salt '*' purefa.pg_update foo [hostgroup=foo | host=bar | volume=vol]
```

`salt.modules.purefa.`**`snap_create`**`(`*name, suffix=None*`)`
Create a volume snapshot on a Pure Storage FlashArray.

Will return False is volume selected to snap does not exist.

New in version 2018.3.0.
name [string] name of volume to snapshot
suffix [string] if specificed forces snapshot name suffix. If not specified defaults to timestamp.
CLI Example:

```
salt '*' purefa.snap_create foo
salt '*' purefa.snap_create foo suffix=bar
```

`salt.modules.purefa.`**`snap_delete`**(*name*, *suffix=None*, *eradicate=False*)

Delete a volume snapshot on a Pure Storage FlashArray.

Will return False if selected snapshot does not exist.

New in version 2018.3.0.
name [string] name of volume
suffix [string] name of snapshot
eradicate [boolean] Eradicate snapshot after deletion if True. Default is False
CLI Example:

```
salt '*' purefa.snap_delete foo suffix=snap eradicate=True
```

`salt.modules.purefa.`**`snap_eradicate`**(*name*, *suffix=None*)

Eradicate a deleted volume snapshot on a Pure Storage FlashArray.

Will return False if snapshot is not in a deleted state.

New in version 2018.3.0.
name [string] name of volume
suffix [string] name of snapshot
CLI Example:

```
salt '*' purefa.snap_eradicate foo suffix=snap
```

`salt.modules.purefa.`**`snap_volume_create`**(*name*, *target*, *overwrite=False*)

Create R/W volume from snapshot on a Pure Storage FlashArray.

Will return False if target volume already exists and overwrite is not specified, or selected snapshot doesn't exist.

New in version 2018.3.0.
name [string] name of volume snapshot
target [string] name of clone volume
overwrite [boolean] overwrite clone if already exists (default: False)
CLI Example:

```
salt '*' purefa.snap_volume_create foo.bar clone overwrite=True
```

`salt.modules.purefa.`**`volume_attach`**(*name*, *host*)

Attach a volume to a host on a Pure Storage FlashArray.

Host and volume must exist or else will return False.

New in version 2018.3.0.
name [string] name of volume
host [string] name of host
CLI Example:

```
salt '*' purefa.volume_attach foo bar
```

`salt.modules.purefa.`**`volume_clone`**(*name*, *target*, *overwrite=False*)

Clone an existing volume on a Pure Storage FlashArray.

Will return False if source volume doesn't exist, or target volume already exists and overwrite not specified.

New in version 2018.3.0.

name [string] name of volume
target [string] name of clone volume
overwrite [boolean] overwrite clone if already exists (default: False)
CLI Example:

```
salt '*' purefa.volume_clone foo bar overwrite=True
```

`salt.modules.purefa.`**`volume_create`**`(`*name*, *size=None*`)`
Create a volume on a Pure Storage FlashArray.

Will return False if volume already exists.

New in version 2018.3.0.
name [string] name of volume (truncated to 63 characters)
size [string] if specificed capacity of volume. If not specified default to 1G. Refer to Pure Storage documentation for formatting rules.
CLI Example:

```
salt '*' purefa.volume_create foo
salt '*' purefa.volume_create foo size=10T
```

`salt.modules.purefa.`**`volume_delete`**`(`*name*, *eradicate=False*`)`
Delete a volume on a Pure Storage FlashArray.

Will return False if volume doesn't exist is already in a deleted state.

New in version 2018.3.0.
name [string] name of volume
eradicate [boolean] Eradicate volume after deletion if True. Default is False
CLI Example:

```
salt '*' purefa.volume_delete foo eradicate=True
```

`salt.modules.purefa.`**`volume_detach`**`(`*name*, *host*`)`
Detach a volume from a host on a Pure Storage FlashArray.

Will return False if either host or volume do not exist, or if selected volume isn't already connected to the host.

New in version 2018.3.0.
name [string] name of volume
host [string] name of host
CLI Example:

```
salt '*' purefa.volume_detach foo bar
```

`salt.modules.purefa.`**`volume_eradicate`**`(`*name*`)`
Eradicate a deleted volume on a Pure Storage FlashArray.

Will return False is volume is not in a deleted state.

New in version 2018.3.0.
name [string] name of volume
CLI Example:

```
salt '*' purefa.volume_eradicate foo
```

`salt.modules.purefa.`**`volume_extend`**`(`*name*, *size*`)`
Extend an existing volume on a Pure Storage FlashArray.

Will return False if new size is less than or equal to existing size.

New in version 2018.3.0.
name [string] name of volume
size [string] New capacity of volume. Refer to Pure Storage documentation for formatting rules.
CLI Example:

```
salt '*' purefa.volume_extend foo 10T
```

21.9.336 salt.modules.purefb

Management of Pure Storage FlashBlade

Installation Prerequisites

- You will need the `purity_fb` python package in your python installation path that is running salt.

```
pip install purity_fb
```

- Configure Pure Storage FlashBlade authentication. Use one of the following three methods.

  1) From the minion config

```
pure_tags:
  fb:
    san_ip: management vip or hostname for the FlashBlade
    api_token: A valid api token for the FlashBlade being managed
```

  2) From environment (PUREFB_IP and PUREFB_API)

  3) From the pillar (PUREFB_IP and PUREFB_API)

maintainer Simon Dodsley (simon@purestorage.com)

maturity new

requires purity_fb

platform all

New in version 2019.2.0.

`salt.modules.purefb.`**`fs_create`**(*name, size=None, proto='NFS', nfs_rules='*(rw, no_root_squash)', snapshot=False*)

Create a filesystem on a Pure Storage FlashBlade.

Will return False if filesystem already exists.

New in version 2019.2.0.
name [string] name of filesystem (truncated to 63 characters)
proto [string] (Optional) Sharing protocol (NFS, CIFS or HTTP). If not specified default is NFS
snapshot: boolean (Optional) Are snapshots enabled on the filesystem. Default is False
nfs_rules [string] (Optional) export rules for NFS. If not specified default is `*(rw,no_root_squash)`. Refer to Pure Storage documentation for formatting rules.
size [string] if specified capacity of filesystem. If not specified default to 32G. Refer to Pure Storage documentation for formatting rules.
CLI Example:

```
salt '*' purefb.fs_create foo proto=CIFS
salt '*' purefb.fs_create foo size=10T
```

`salt.modules.purefb.`**`fs_delete`**`(`*name, eradicate=False*`)`

Delete a share on a Pure Storage FlashBlade.

Will return False if filesystem doesn't exist or is already in a deleted state.

New in version 2019.2.0.
name [string] name of filesystem
eradicate [boolean] (Optional) Eradicate filesystem after deletion if True. Default is False
CLI Example:

```
salt '*' purefb.fs_delete foo eradicate=True
```

`salt.modules.purefb.`**`fs_eradicate`**`(`*name*`)`

Eradicate a deleted filesystem on a Pure Storage FlashBlade.

Will return False is filesystem is not in a deleted state.

New in version 2019.2.0.
name [string] name of filesystem
CLI Example:

```
salt '*' purefb.fs_eradicate foo
```

`salt.modules.purefb.`**`fs_extend`**`(`*name, size*`)`

Resize an existing filesystem on a Pure Storage FlashBlade.

Will return False if new size is less than or equal to existing size.

New in version 2019.2.0.
name [string] name of filesystem
size [string] New capacity of filesystem. Refer to Pure Storage documentation for formatting rules.
CLI Example:

```
salt '*' purefb.fs_extend foo 10T
```

`salt.modules.purefb.`**`fs_update`**`(`*name, rules, snapshot=False*`)`

Update filesystem on a Pure Storage FlashBlade.

Allows for change of NFS export rules and enabling/disabled of snapshotting capability.

New in version 2019.2.0.
name [string] name of filesystem
rules [string] NFS export rules for filesystem Refer to Pure Storage documentation for formatting rules.
snapshot: boolean (Optional) Enable/Disable snapshots on the filesystem. Default is False
CLI Example:

```
salt '*' purefb.fs_nfs_update foo rules='10.234.112.23(ro), 10.234.112.24(rw)'
↪snapshot=True
```

`salt.modules.purefb.`**`snap_create`**`(`*name, suffix=None*`)`

Create a filesystem snapshot on a Pure Storage FlashBlade.

Will return False if filesystem selected to snap does not exist.

New in version 2019.2.0.
name [string] name of filesystem to snapshot
suffix [string] if specificed forces snapshot name suffix. If not specified defaults to timestamp.
CLI Example:

```
salt '*' purefb.snap_create foo
salt '*' purefb.snap_create foo suffix=bar
```

`salt.modules.purefb.`**`snap_delete`**`(`*name, suffix=None, eradicate=False*`)`

Delete a filesystem snapshot on a Pure Storage FlashBlade.

Will return False if selected snapshot does not exist.

New in version 2019.2.0.

name [string] name of filesystem
suffix [string] name of snapshot
eradicate [boolean] Eradicate snapshot after deletion if True. Default is False
CLI Example:

```
salt '*' purefb.snap_delete foo suffix=snap eradicate=True
```

`salt.modules.purefb.`**`snap_eradicate`**`(`*name, suffix=None*`)`

Eradicate a deleted filesystem snapshot on a Pure Storage FlashBlade.

Will return False if snapshot is not in a deleted state.

New in version 2019.2.0.

name [string] name of filesystem
suffix [string] name of snapshot
CLI Example:

```
salt '*' purefb.snap_eradicate foo suffix=snap
```

21.9.337 salt.modules.pushbullet module

Module for sending messages to Pushbullet (https://www.pushbullet.com)

New in version 2015.8.0.

Requires an `api_key` in `/etc/salt/minion`:

```
pushbullet:
  api_key: 'ABC123abc123ABC123abc123ABC123ab'
```

For example:

```
pushbullet:
  device: "Chrome"
  title: "Example push message"
  body: "Message body."
```

`salt.modules.pushbullet.`**`push_note`**`(`*device=None, title=None, body=None*`)`

Pushing a text note.

Parameters

- **`device`** – Pushbullet target device
- **`title`** – Note title
- **`body`** – Note body

Returns Boolean if message was sent successfully.

CLI Example:

```
salt "*" pushbullet.push_note device="Chrome" title="Example title" body="Example
↪body."
```

21.9.338 salt.modules.pushover_notify

Module for sending messages to Pushover (https://www.pushover.net)

New in version 2016.3.0.

configuration This module can be used by either passing an api key and version directly or by specifying both in a configuration profile in the salt master/minion config.

For example:

```
pushover:
  token: abAHuZyCLtdH8P4zhmFZmgUHUsv1ei8
```

`salt.modules.pushover_notify.`**`post_message`**(*user=None, device=None, message=None, title=None, priority=None, expire=None, retry=None, sound=None, api_version=1, token=None*)

Send a message to a Pushover user or group.

Parameters

- **`user`** – The user or group to send to, must be key of user or group not email address.
- **`message`** – The message to send to the PushOver user or group.
- **`title`** – Specify who the message is from.
- **`priority`** – The priority of the message, defaults to 0.
- **`expire`** – The message should expire after N number of seconds.
- **`retry`** – The number of times the message should be retried.
- **`sound`** – The sound to associate with the message.
- **`api_version`** – The PushOver API version, if not specified in the configuration.
- **`token`** – The PushOver token, if not specified in the configuration.

Returns Boolean if message was sent successfully.

CLI Example:

```
salt '*' pushover.post_message user='xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx' title=
↪'Message from Salt' message='Build is done'

salt '*' pushover.post_message user='xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx' title=
↪'Message from Salt' message='Build is done' priority='2' expire='720' retry='5'
```

21.9.339 salt.modules.pw_group

Manage groups on FreeBSD

Important: If you feel that Salt should be using this module to manage groups on a minion, and it is using a different module (or gives an error similar to *'group.info' is not available*), see *here*.

`salt.modules.pw_group.`**`add`**(*name, gid=None, **kwargs*)

Add the specified group

CLI Example:

```
salt '*' group.add foo 3456
```

`salt.modules.pw_group.`**`adduser`**(*name, username*)

Add a user in the group.

CLI Example:

```
salt '*' group.adduser foo bar
```

Verifies if a valid username 'bar' as a member of an existing group 'foo', if not then adds it.

`salt.modules.pw_group.`**`chgid`**`(`*name, gid*`)`

Change the gid for a named group

CLI Example:

```
salt '*' group.chgid foo 4376
```

`salt.modules.pw_group.`**`delete`**`(`*name*`)`

Remove the named group

CLI Example:

```
salt '*' group.delete foo
```

`salt.modules.pw_group.`**`deluser`**`(`*name, username*`)`

Remove a user from the group.

CLI Example:

```
salt '*' group.deluser foo bar
```

Removes a member user 'bar' from a group 'foo'. If group is not present then returns True.

`salt.modules.pw_group.`**`getent`**`(`*refresh=False*`)`

Return info on all groups

CLI Example:

```
salt '*' group.getent
```

`salt.modules.pw_group.`**`info`**`(`*name*`)`

Return information about a group

CLI Example:

```
salt '*' group.info foo
```

`salt.modules.pw_group.`**`members`**`(`*name, members_list*`)`

Replaces members of the group with a provided list.

New in version 2015.5.4.

CLI Example:
salt '*' group.members foo 'user1,user2,user3,...'

Replaces a membership list for a local group 'foo'. foo:x:1234:user1,user2,user3,...

21.9.340 salt.modules.pw_user

Manage users with the pw command

Important: If you feel that Salt should be using this module to manage users on a minion, and it is using a different module (or gives an error similar to *'user.info' is not available*), see *here*.

`salt.modules.pw_user.`**`add`**(*name, uid=None, gid=None, groups=None, home=None, shell=None, unique=True, fullname='', roomnumber='', workphone='', homephone='', createhome=True, loginclass=None, **kwargs*)

Add a user to the minion

CLI Example:

```
salt '*' user.add name <uid> <gid> <groups> <home> <shell>
```

`salt.modules.pw_user.`**`chfullname`**(*name, fullname*)

Change the user's Full Name

CLI Example:

```
salt '*' user.chfullname foo "Foo Bar"
```

`salt.modules.pw_user.`**`chgid`**(*name, gid*)

Change the default group of the user

CLI Example:

```
salt '*' user.chgid foo 4376
```

`salt.modules.pw_user.`**`chgroups`**(*name, groups, append=False*)

Change the groups to which a user belongs

name Username to modify

groups List of groups to set for the user. Can be passed as a comma-separated list or a Python list.

append [False] Set to `True` to append these groups to the user's existing list of groups. Otherwise, the specified groups will replace any existing groups for the user.

CLI Example:

```
salt '*' user.chgroups foo wheel,root True
```

`salt.modules.pw_user.`**`chhome`**(*name, home, persist=False*)

Set a new home directory for an existing user

name Username to modify

home New home directory to set

persist [False] Set to `True` to prevent configuration files in the new home directory from being overwritten by the files from the skeleton directory.

CLI Example:

```
salt '*' user.chhome foo /home/users/foo True
```

`salt.modules.pw_user.`**`chhomephone`**(*name, homephone*)

Change the user's Home Phone

CLI Example:

```
salt '*' user.chhomephone foo "7735551234"
```

`salt.modules.pw_user.`**`chloginclass`**(*name, loginclass, root=None*)

Change the default login class of the user

New in version 2016.3.5.

CLI Example:

```
salt '*' user.chloginclass foo staff
```

`salt.modules.pw_user.`**`chroomnumber`**`(`*name, roomnumber*`)`

Change the user's Room Number

CLI Example:

```
salt '*' user.chroomnumber foo 123
```

`salt.modules.pw_user.`**`chshell`**`(`*name, shell*`)`

Change the default shell of the user

CLI Example:

```
salt '*' user.chshell foo /bin/zsh
```

`salt.modules.pw_user.`**`chuid`**`(`*name, uid*`)`

Change the uid for a named user

CLI Example:

```
salt '*' user.chuid foo 4376
```

`salt.modules.pw_user.`**`chworkphone`**`(`*name, workphone*`)`

Change the user's Work Phone

CLI Example:

```
salt '*' user.chworkphone foo "7735550123"
```

`salt.modules.pw_user.`**`delete`**`(`*name, remove=False, force=False*`)`

Remove a user from the minion

CLI Example:

```
salt '*' user.delete name remove=True force=True
```

`salt.modules.pw_user.`**`get_loginclass`**`(`*name*`)`

Get the login class of the user

New in version 2016.3.0.

CLI Example:

```
salt '*' user.get_loginclass foo
```

`salt.modules.pw_user.`**`getent`**`(`*refresh=False*`)`

Return the list of all info for all users

CLI Example:

```
salt '*' user.getent
```

`salt.modules.pw_user.`**`info`**`(`*name*`)`

Return user information

CLI Example:

```
salt '*' user.info root
```

`salt.modules.pw_user.`**`list_groups`**`(`*name*`)`

Return a list of groups the named user belongs to

CLI Example:

```
salt '*' user.list_groups foo
```

salt.modules.pw_user.**list_users**()
Return a list of all users

CLI Example:

```
salt '*' user.list_users
```

salt.modules.pw_user.**rename**(*name*, *new_name*)
Change the username for a named user

CLI Example:

```
salt '*' user.rename name new_name
```

21.9.341 salt.modules.pyenv

Manage python installations with pyenv.

Note: Git needs to be installed and available via PATH if pyenv is to be installed automatically by the module.

New in version v2014.04.

salt.modules.pyenv.**default**(*python=None*, *runas=None*)
Returns or sets the currently defined default python.

python=None The version to set as the default. Should match one of the versions listed by `pyenv.versions`. Leave blank to return the current default.

CLI Example:

```
salt '*' pyenv.default
salt '*' pyenv.default 2.0.0-p0
```

salt.modules.pyenv.**do**(*cmdline=None*, *runas=None*)
Execute a python command with pyenv's shims from the user or the system.

CLI Example:

```
salt '*' pyenv.do 'gem list bundler'
salt '*' pyenv.do 'gem list bundler' deploy
```

salt.modules.pyenv.**do_with_python**(*python*, *cmdline*, *runas=None*)
Execute a python command with pyenv's shims using a specific python version.

CLI Example:

```
salt '*' pyenv.do_with_python 2.0.0-p0 'gem list bundler'
salt '*' pyenv.do_with_python 2.0.0-p0 'gem list bundler' deploy
```

salt.modules.pyenv.**install**(*runas=None*, *path=None*)
Install pyenv systemwide

CLI Example:

```
salt '*' pyenv.install
```

`salt.modules.pyenv.`**`install_python`**(*python*, *runas=None*)

Install a python implementation.

python The version of python to install, should match one of the versions listed by pyenv.list

CLI Example:

```
salt '*' pyenv.install_python 2.0.0-p0
```

`salt.modules.pyenv.`**`is_installed`**(*runas=None*)

Check if pyenv is installed.

CLI Example:

```
salt '*' pyenv.is_installed
```

`salt.modules.pyenv.`**`list_`**(*runas=None*)

List the installable versions of python.

CLI Example:

```
salt '*' pyenv.list
```

`salt.modules.pyenv.`**`rehash`**(*runas=None*)

Run pyenv rehash to update the installed shims.

CLI Example:

```
salt '*' pyenv.rehash
```

`salt.modules.pyenv.`**`uninstall_python`**(*python*, *runas=None*)

Uninstall a python implementation.

python The version of python to uninstall. Should match one of the versions listed by `pyenv.versions`

CLI Example:

```
salt '*' pyenv.uninstall_python 2.0.0-p0
```

`salt.modules.pyenv.`**`update`**(*runas=None*, *path=None*)

Updates the current versions of pyenv and python-Build

CLI Example:

```
salt '*' pyenv.update
```

`salt.modules.pyenv.`**`versions`**(*runas=None*)

List the installed versions of python.

CLI Example:

```
salt '*' pyenv.versions
```

21.9.342 salt.modules.qemu_img

Qemu-img Command Wrapper

The qemu img command is wrapped for specific functions

depends qemu-img

`salt.modules.qemu_img.`**`convert`**`(`*orig, dest, fmt*`)`

Convert an existing disk image to another format using qemu-img

CLI Example:

```
salt '*' qemu_img.convert /path/to/original.img /path/to/new.img qcow2
```

`salt.modules.qemu_img.`**`make_image`**`(`*location, size, fmt*`)`

Create a blank virtual machine image file of the specified size in megabytes. The image can be created in any format supported by qemu

CLI Example:

```
salt '*' qemu_img.make_image /tmp/image.qcow 2048 qcow2
salt '*' qemu_img.make_image /tmp/image.raw 10240 raw
```

21.9.343 salt.modules.qemu_nbd

Qemu Command Wrapper

The qemu system comes with powerful tools, such as qemu-img and qemu-nbd which are used here to build up kvm images.

`salt.modules.qemu_nbd.`**`clear`**`(`*mnt*`)`

Pass in the mnt dict returned from nbd_mount to unmount and disconnect the image from nbd. If all of the partitions are unmounted return an empty dict, otherwise return a dict containing the still mounted partitions

CLI Example:

```
salt '*' qemu_nbd.clear '{"/mnt/foo": "/dev/nbd0p1"}'
```

`salt.modules.qemu_nbd.`**`connect`**`(`*image*`)`

Activate nbd for an image file.

CLI Example:

```
salt '*' qemu_nbd.connect /tmp/image.raw
```

`salt.modules.qemu_nbd.`**`init`**`(`*image, root=None*`)`

Mount the named image via qemu-nbd and return the mounted roots

CLI Example:

```
salt '*' qemu_nbd.init /srv/image.qcow2
```

`salt.modules.qemu_nbd.`**`mount`**`(`*nbd, root=None*`)`

Pass in the nbd connection device location, mount all partitions and return a dict of mount points

CLI Example:

```
salt '*' qemu_nbd.mount /dev/nbd0
```

21.9.344 salt.modules.quota

Module for managing quotas on POSIX-like systems.

`salt.modules.quota.`**`get_mode`**`(`*device*`)`

Report whether the quota system for this device is on or off

CLI Example:

```
salt '*' quota.get_mode
```

`salt.modules.quota.`**`off`**`(`*device*`)`

Turns off the quota system

CLI Example:

```
salt '*' quota.off
```

`salt.modules.quota.`**`on`**`(`*device*`)`

Turns on the quota system

CLI Example:

```
salt '*' quota.on
```

`salt.modules.quota.`**`report`**`(`*mount*`)`

Report on quotas for a specific volume

CLI Example:

```
salt '*' quota.report /media/data
```

`salt.modules.quota.`**`set_`**`(`*device, **kwargs*`)`

Calls out to setquota, for a specific user or group

CLI Example:

```
salt '*' quota.set /media/data user=larry block-soft-limit=1048576
salt '*' quota.set /media/data group=painters file-hard-limit=1000
```

`salt.modules.quota.`**`stats`**`()`

Runs the quotastats command, and returns the parsed output

CLI Example:

```
salt '*' quota.stats
```

`salt.modules.quota.`**`warn`**`()`

Runs the warnquota command, to send warning emails to users who are over their quota limit.

CLI Example:

```
salt '*' quota.warn
```

21.9.345 salt.modules.rabbitmq

Module to provide RabbitMQ compatibility to Salt. Todo: A lot, need to add cluster support, logging, and minion configuration data.

`salt.modules.rabbitmq.`**`add_user`**`(`*name, password=None, runas=None*`)`

Add a rabbitMQ user via rabbitmqctl user_add <user> <password>

CLI Example:

```
salt '*' rabbitmq.add_user rabbit_user password
```

`salt.modules.rabbitmq.`**`add_vhost`**`(`*vhost*, *runas=None*`)`

Adds a vhost via rabbitmqctl add_vhost.

CLI Example:

```
salt '*' rabbitmq add_vhost '<vhost_name>'
```

`salt.modules.rabbitmq.`**`change_password`**`(`*name*, *password*, *runas=None*`)`

Changes a user's password.

CLI Example:

```
salt '*' rabbitmq.change_password rabbit_user password
```

`salt.modules.rabbitmq.`**`check_password`**`(`*name*, *password*, *runas=None*`)`

New in version 2016.3.0.

Checks if a user's password is valid.

CLI Example:

```
salt '*' rabbitmq.check_password rabbit_user password
```

`salt.modules.rabbitmq.`**`clear_password`**`(`*name*, *runas=None*`)`

Removes a user's password.

CLI Example:

```
salt '*' rabbitmq.clear_password rabbit_user
```

`salt.modules.rabbitmq.`**`cluster_status`**`(`*runas=None*`)`

return rabbitmq cluster_status

CLI Example:

```
salt '*' rabbitmq.cluster_status
```

`salt.modules.rabbitmq.`**`delete_policy`**`(`*vhost*, *name*, *runas=None*`)`

Delete a policy based on rabbitmqctl clear_policy.

Reference: http://www.rabbitmq.com/ha.html

CLI Example:

```
salt '*' rabbitmq.delete_policy / HA
```

`salt.modules.rabbitmq.`**`delete_upstream`**`(`*name*, *runas=None*`)`

Deletes an upstream via rabbitmqctl clear_parameter.

Parameters

- **name** (*str*) – The name of the upstream to delete.
- **runas** (*str*) – The name of the user to run the command as.

CLI Example:

```
salt '*' rabbitmq.delete_upstream upstream_name
```

New in version 3000.

`salt.modules.rabbitmq.`**`delete_user`**(*name*, *runas=None*)

Deletes a user via rabbitmqctl delete_user.

CLI Example:

```
salt '*' rabbitmq.delete_user rabbit_user
```

`salt.modules.rabbitmq.`**`delete_vhost`**(*vhost*, *runas=None*)

Deletes a vhost rabbitmqctl delete_vhost.

CLI Example:

```
salt '*' rabbitmq.delete_vhost '<vhost_name>'
```

`salt.modules.rabbitmq.`**`disable_plugin`**(*name*, *runas=None*)

Disable a RabbitMQ plugin via the rabbitmq-plugins command.

CLI Example:

```
salt '*' rabbitmq.disable_plugin foo
```

`salt.modules.rabbitmq.`**`enable_plugin`**(*name*, *runas=None*)

Enable a RabbitMQ plugin via the rabbitmq-plugins command.

CLI Example:

```
salt '*' rabbitmq.enable_plugin foo
```

`salt.modules.rabbitmq.`**`force_reset`**(*runas=None*)

Forcefully Return a RabbitMQ node to its virgin state

CLI Example:

```
salt '*' rabbitmq.force_reset
```

`salt.modules.rabbitmq.`**`join_cluster`**(*host*, *user='rabbit'*, *ram_node=None*, *runas=None*)

Join a rabbit cluster

CLI Example:

```
salt '*' rabbitmq.join_cluster rabbit.example.com rabbit
```

`salt.modules.rabbitmq.`**`list_available_plugins`**(*runas=None*)

Returns a list of the names of all available plugins (enabled and disabled).

CLI Example:

```
salt '*' rabbitmq.list_available_plugins
```

`salt.modules.rabbitmq.`**`list_enabled_plugins`**(*runas=None*)

Returns a list of the names of the enabled plugins.

CLI Example:

```
salt '*' rabbitmq.list_enabled_plugins
```

`salt.modules.rabbitmq.`**`list_permissions`**(*vhost*, *runas=None*)

Lists permissions for vhost via rabbitmqctl list_permissions

CLI Example:

```
salt '*' rabbitmq.list_permissions /myvhost
```

`salt.modules.rabbitmq.`**`list_policies`**`(`*vhost='/', runas=None*`)`

Return a dictionary of policies nested by vhost and name based on the data returned from rabbitmqctl list_policies.

Reference: http://www.rabbitmq.com/ha.html

CLI Example:

```
salt '*' rabbitmq.list_policies
```

`salt.modules.rabbitmq.`**`list_queues`**`(`*runas=None, *args*`)`

Returns queue details of the / virtual host

CLI Example:

```
salt '*' rabbitmq.list_queues messages consumers
```

`salt.modules.rabbitmq.`**`list_queues_vhost`**`(`*vhost, runas=None, *args*`)`

Returns queue details of specified virtual host. This command will consider first parameter as the vhost name and rest will be treated as queueinfoitem. For getting details on vhost `/`, use `list_queues` instead).

CLI Example:

```
salt '*' rabbitmq.list_queues messages consumers
```

`salt.modules.rabbitmq.`**`list_upstreams`**`(`*runas=None*`)`

Returns a dict of upstreams based on rabbitmqctl list_parameters.

Parameters **`runas`** (`str`) – The name of the user to run this command as.

CLI Example:

```
salt '*' rabbitmq.list_upstreams
```

New in version 3000.

`salt.modules.rabbitmq.`**`list_user_permissions`**`(`*name, runas=None*`)`

List permissions for a user via rabbitmqctl list_user_permissions

CLI Example:

```
salt '*' rabbitmq.list_user_permissions user
```

`salt.modules.rabbitmq.`**`list_users`**`(`*runas=None*`)`

Return a list of users based off of rabbitmqctl user_list.

CLI Example:

```
salt '*' rabbitmq.list_users
```

`salt.modules.rabbitmq.`**`list_vhosts`**`(`*runas=None*`)`

Return a list of vhost based on rabbitmqctl list_vhosts.

CLI Example:

```
salt '*' rabbitmq.list_vhosts
```

`salt.modules.rabbitmq.`**`plugin_is_enabled`**`(`*name, runas=None*`)`

Return whether the plugin is enabled.

CLI Example:

```
salt '*' rabbitmq.plugin_is_enabled rabbitmq_plugin_name
```

`salt.modules.rabbitmq.`**`policy_exists`**(*vhost*, *name*, *runas=None*)

Return whether the policy exists based on rabbitmqctl list_policies.

Reference: http://www.rabbitmq.com/ha.html

CLI Example:

```
salt '*' rabbitmq.policy_exists / HA
```

`salt.modules.rabbitmq.`**`reset`**(*runas=None*)

Return a RabbitMQ node to its virgin state

CLI Example:

```
salt '*' rabbitmq.reset
```

`salt.modules.rabbitmq.`**`set_permissions`**(*vhost*, *user*, *conf='.*'*, *write='.*'*, *read='.*'*, *runas=None*)

Sets permissions for vhost via rabbitmqctl set_permissions

CLI Example:

```
salt '*' rabbitmq.set_permissions myvhost myuser
```

`salt.modules.rabbitmq.`**`set_policy`**(*vhost*, *name*, *pattern*, *definition*, *priority=None*, *runas=None*, *apply_to=None*)

Set a policy based on rabbitmqctl set_policy.

Reference: http://www.rabbitmq.com/ha.html

CLI Example:

```
salt '*' rabbitmq.set_policy / HA '.*' '{"ha-mode":"all"}'
```

`salt.modules.rabbitmq.`**`set_upstream`**(*name*, *uri*, *prefetch_count=None*, *reconnect_delay=None*, *ack_mode=None*, *trust_user_id=None*, *exchange=None*, *max_hops=None*, *expires=None*, *message_ttl=None*, *ha_policy=None*, *queue=None*, *runas=None*)

Configures an upstream via rabbitmqctl set_parameter. This can be an exchange-upstream, a queue-upstream or both.

Parameters **`name`** (*`str`*) – The name of the upstream to configure.

The following parameters apply to federated exchanges and federated queues:

Parameters

- **`uri`** (*`str`*) – The AMQP URI(s) for the upstream.
- **`prefetch_count`** (*`int`*) – The maximum number of unacknowledged messages copied over a link at any one time. Default: 1000
- **`reconnect_delay`** (*`int`*) – The duration (in seconds) to wait before reconnecting to the broker after being disconnected. Default: 1
- **`ack_mode`** (*`str`*) – Determines how the link should acknowledge messages. If set to `on-confirm` (the default), messages are acknowledged to the upstream broker after they have been confirmed downstream. This handles network errors and broker failures without losing messages, and is the slowest option. If set to `on-publish`, messages are acknowledged to the upstream broker after they have been published downstream. This handles network errors without losing messages, but may lose messages in the event of broker failures. If set to `no-ack`, message acknowledgements

are not used. This is the fastest option, but may lose messages in the event of network or broker failures.
- **trust_user_id** (*bool*) – Determines how federation should interact with the validated user-id feature. If set to true, federation will pass through any validated user-id from the upstream, even though it cannot validate it itself. If set to false or not set, it will clear any validated user-id it encounters. You should only set this to true if you trust the upstream server (and by extension, all its upstreams) not to forge user-ids.

The following parameters apply to federated exchanges only:

Parameters

- **exchange** (*str*) – The name of the upstream exchange. Default is to use the same name as the federated exchange.
- **max_hops** (*int*) – The maximum number of federation links that a message published to a federated exchange can traverse before it is discarded. Default is 1. Note that even if max-hops is set to a value greater than 1, messages will never visit the same node twice due to travelling in a loop. However, messages may still be duplicated if it is possible for them to travel from the source to the destination via multiple routes.
- **expires** (*int*) – The expiry time (in milliseconds) after which an upstream queue for a federated exchange may be deleted, if a connection to the upstream broker is lost. The default is 'none', meaning the queue should never expire. This setting controls how long the upstream queue will last before it is eligible for deletion if the connection is lost. This value is used to set the "x-expires" argument for the upstream queue.
- **message_ttl** (*int*) – The expiry time for messages in the upstream queue for a federated exchange (see expires), in milliseconds. Default is `None`, meaning messages should never expire. This does not apply to federated queues. This value is used to set the "x-message-ttl" argument for the upstream queue.
- **ha_policy** (*str*) – Determines the "x-ha-policy" argument for the upstream queue for a federated exchange (see expires). This is only of interest when connecting to old brokers which determine queue HA mode using this argument. Default is `None`, meaning the queue is not HA.

The following parameter applies to federated queues only:

Parameters

- **queue** (*str*) – The name of the upstream queue. Default is to use the same name as the federated queue.
- **runas** (*str*) – The name of the user to run the command as.

CLI Example:

```
salt '*' rabbitmq.set_upstream upstream_name ack_mode=on-confirm max_hops=1
↪      trust_user_id=True uri=amqp://hostname
```

New in version 3000.

`salt.modules.rabbitmq.`**`set_user_tags`**(*name, tags, runas=None*)

Add user tags via rabbitmqctl set_user_tags

CLI Example:

```
salt '*' rabbitmq.set_user_tags myadmin administrator
```

`salt.modules.rabbitmq.`**`start_app`**(*runas=None*)

Start the RabbitMQ application.

CLI Example:

```
salt '*' rabbitmq.start_app
```

`salt.modules.rabbitmq.`**`status`**(*runas=None*)

return rabbitmq status

CLI Example:

```
salt '*' rabbitmq.status
```

`salt.modules.rabbitmq.`**`stop_app`**(*runas=None*)

Stops the RabbitMQ application, leaving the Erlang node running.

CLI Example:

```
salt '*' rabbitmq.stop_app
```

`salt.modules.rabbitmq.`**`upstream_exists`**(*name, runas=None*)

Return whether the upstreamexists based on rabbitmqctl list_parameters.

Parameters

- **name** (*str*) – The name of the upstream to check for.
- **runas** (*str*) – The name of the user to run the command as.

CLI Example:

```
salt '*' rabbitmq.upstream_exists rabbit_upstream
```

New in version 3000.

`salt.modules.rabbitmq.`**`user_exists`**(*name, runas=None*)

Return whether the user exists based on rabbitmqctl list_users.

CLI Example:

```
salt '*' rabbitmq.user_exists rabbit_user
```

`salt.modules.rabbitmq.`**`vhost_exists`**(*name, runas=None*)

Return whether the vhost exists based on rabbitmqctl list_vhosts.

CLI Example:

```
salt '*' rabbitmq.vhost_exists rabbit_host
```

21.9.346 salt.modules.rallydev

Support for RallyDev

New in version 2015.8.0.

Requires a `username` and a `password` in `/etc/salt/minion`:

```
rallydev:
  username: myuser@example.com
  password: 123pass
```

`salt.modules.rallydev.`**`list_items`**(*name*)

List items of a particular type

CLI Examples:

```
salt myminion rallydev.list_<item name>s
salt myminion rallydev.list_users
salt myminion rallydev.list_artifacts
```

`salt.modules.rallydev.`**`list_users`**`()`

List the users

CLI Example:

```
salt myminion rallydev.list_users
```

`salt.modules.rallydev.`**`query_item`**`(`*name, query_string, order='Rank'*`)`

Query a type of record for one or more items. Requires a valid query string. See https://rally1.rallydev.com/slm/doc/webservice/introduction.jsp for information on query syntax.

CLI Example:

```
salt myminion rallydev.query_<item name> <query string> [<order>]
salt myminion rallydev.query_task '(Name contains github)'
salt myminion rallydev.query_task '(Name contains reactor)' Rank
```

`salt.modules.rallydev.`**`query_user`**`(`*query_string, order='UserName'*`)`

Update a user

CLI Example:

```
salt myminion rallydev.query_user '(Name contains Jo)'
```

`salt.modules.rallydev.`**`show_artifact`**`(`*id_*`)`

Show an artifact

CLI Example:

```
salt myminion rallydev.show_artifact <artifact id>
```

`salt.modules.rallydev.`**`show_item`**`(`*name, id_*`)`

Show an item

CLI Example:

```
salt myminion rallydev.show_<item name> <item id>
```

`salt.modules.rallydev.`**`show_user`**`(`*id_*`)`

Show a user

CLI Example:

```
salt myminion rallydev.show_user <user id>
```

`salt.modules.rallydev.`**`update_item`**`(`*name, id_, field=None, value=None, postdata=None*`)`

Update an item. Either a field and a value, or a chunk of POST data, may be used, but not both.

CLI Example:

```
salt myminion rallydev.update_<item name> <item id> field=<field> value=<value>
salt myminion rallydev.update_<item name> <item id> postdata=<post data>
```

`salt.modules.rallydev.`**`update_user`**`(`*id_, field, value*`)`

Update a user

CLI Example:

```
salt myminion rallydev.update_user <user id> <field> <new value>
```

21.9.347 salt.modules.random_org

Module for retrieving random information from Random.org

New in version 2015.5.0.

configuration This module can be used by either passing an api key and version directly or by specifying both in a configuration profile in the salt master/minion config.

For example:

```
random_org:
  api_key: 7be1402d-5719-5bd3-a306-3def9f135da5
  api_version: 1
```

`salt.modules.random_org.`**`generateBlobs`**(*api_key=None*, *api_version=None*, ***kwargs*)

List all Slack users.

Parameters

- **`api_key`** – The Random.org api key.
- **`api_version`** – The Random.org api version.
- **`format`** – Specifies the format in which the blobs will be returned. Values allowed are base64 and hex.

Returns The user list.

CLI Example:

```
salt '*' get_integers number=5 min=1 max=6

salt '*' get_integers number=5 min=1 max=6
```

`salt.modules.random_org.`**`generateDecimalFractions`**(*api_key=None*, *api_version=None*, ***kwargs*)

Generates true random decimal fractions

Parameters

- **`api_key`** – The Random.org api key.
- **`api_version`** – The Random.org api version.
- **`number`** – How many random decimal fractions you need. Must be within the [1,1e4] range.
- **`decimalPlaces`** – The number of decimal places to use. Must be within the [1,20] range.
- **`replacement`** – Specifies whether the random numbers should be picked with replacement. The default (true) will cause the numbers to be picked with replacement, i.e., the resulting numbers may contain duplicate values (like a series of dice rolls). If you want the numbers picked to be unique (like raffle tickets drawn from a container), set this value to false.

Returns A list of decimal fraction

CLI Example:

```
salt '*' random_org.generateDecimalFractions number=10 decimalPlaces=4

salt '*' random_org.generateDecimalFractions number=10 decimalPlaces=4␣
↪replacement=True
```

`salt.modules.random_org.`**`generateGaussians`**(*api_key=None*, *api_version=None*, ***kwargs*)

This method generates true random numbers from a Gaussian distribution (also known as a normal distribution).

Parameters

- **`api_key`** – The Random.org api key.

- **api_version** – The Random.org api version.
- **number** – How many random numbers you need. Must be within the [1,1e4] range.
- **mean** – The distribution's mean. Must be within the [-1e6,1e6] range.
- **standardDeviation** – The distribution's standard deviation. Must be within the [-1e6,1e6] range.
- **significantDigits** – The number of significant digits to use. Must be within the [2,20] range.

Returns The user list.

CLI Example:

```
salt '*' random_org.generateGaussians number=10 mean=0.0 standardDeviation=1.0␣
↪significantDigits=8
```

salt.modules.random_org.**generateIntegers**(*api_key=None*, *api_version=None*, ***kwargs*)

Generate random integers

Parameters

- **api_key** – The Random.org api key.
- **api_version** – The Random.org api version.
- **number** – The number of integers to generate
- **minimum** – The lower boundary for the range from which the random numbers will be picked. Must be within the [-1e9,1e9] range.
- **maximum** – The upper boundary for the range from which the random numbers will be picked. Must be within the [-1e9,1e9] range.
- **replacement** – Specifies whether the random numbers should be picked with replacement. The default (true) will cause the numbers to be picked with replacement, i.e., the resulting numbers may contain duplicate values (like a series of dice rolls). If you want the numbers picked to be unique (like raffle tickets drawn from a container), set this value to false.
- **base** – Specifies the base that will be used to display the numbers. Values allowed are 2, 8, 10 and 16. This affects the JSON types and formatting of the resulting data as discussed below.

Returns A list of integers.

CLI Example:

```
salt '*' random_org.generateIntegers number=5 minimum=1 maximum=6

salt '*' random_org.generateIntegers number=5 minimum=2 maximum=255 base=2
```

salt.modules.random_org.**generateStrings**(*api_key=None*, *api_version=None*, ***kwargs*)

Generate random strings.

Parameters

- **api_key** – The Random.org api key.
- **api_version** – The Random.org api version.
- **number** – The number of strings to generate.
- **length** – The length of each string. Must be within the [1,20] range. All strings will be of the same length
- **characters** – A string that contains the set of characters that are allowed to occur in the random strings. The maximum number of characters is 80.
- **replacement** – Specifies whether the random strings should be picked with replacement. The default (true) will cause the strings to be picked with replacement, i.e., the resulting list of strings may contain duplicates (like a series of dice rolls). If you want the strings to be unique (like raffle tickets drawn from a container), set this value to false.

Returns A list of strings.

CLI Example:

```
salt '*' random_org.generateStrings number=5 length=8 characters=
↪'abcdefghijklmnopqrstuvwxyz'

salt '*' random_org.generateStrings number=10 length=16 characters
↪'abcdefghijklmnopqrstuvwxyz'
```

`salt.modules.random_org.`**`generateUUIDs`**(*api_key=None*, *api_version=None*, ***kwargs*)

Generate a list of random UUIDs

Parameters

- **`api_key`** – The Random.org api key.
- **`api_version`** – The Random.org api version.
- **`number`** – How many random UUIDs you need. Must be within the [1,1e3] range.

Returns A list of UUIDs

CLI Example:

```
salt '*' random_org.generateUUIDs number=5
```

`salt.modules.random_org.`**`getUsage`**(*api_key=None*, *api_version=None*)

Show current usages statistics

Parameters

- **`api_key`** – The Random.org api key.
- **`api_version`** – The Random.org api version.

Returns The current usage statistics.

CLI Example:

```
salt '*' random_org.getUsage

salt '*' random_org.getUsage api_key=peWcBiMOS9HrZG15peWcBiMOS9HrZG15 api_
↪version=1
```

21.9.348 salt.modules.rbac_solaris

Module for Solaris' Role-Based Access Control

`salt.modules.rbac_solaris.`**`auth_add`**(*user*, *auth*)

Add authorization to user

user [string] username

auth [string] authorization name

CLI Example:

```
salt '*' rbac.auth_add martine solaris.zone.manage
salt '*' rbac.auth_add martine solaris.zone.manage,solaris.mail.mailq
```

`salt.modules.rbac_solaris.`**`auth_get`**(*user*, *computed=True*)

List authorization for user

user [string] username

computed [boolean] merge results from *auths* command into data from user_attr

CLI Example:

```
salt '*' rbac.auth_get leo
```

`salt.modules.rbac_solaris.`**`auth_list`**()

List all available authorization

CLI Example:

```
salt '*' rbac.auth_list
```

salt.modules.rbac_solaris.**auth_rm**(*user, auth*)

Remove authorization from user

user [string] username

auth [string] authorization name

CLI Example:

```
salt '*' rbac.auth_rm jorge solaris.zone.manage
salt '*' rbac.auth_rm jorge solaris.zone.manage,solaris.mail.mailq
```

salt.modules.rbac_solaris.**profile_add**(*user, profile*)

Add profile to user

user [string] username

profile [string] profile name

CLI Example:

```
salt '*' rbac.profile_add martine 'Primary Administrator'
salt '*' rbac.profile_add martine 'User Management,User Security'
```

salt.modules.rbac_solaris.**profile_get**(*user, default_hidden=True*)

List profiles for user

user [string] username

default_hidden [boolean] hide default profiles

CLI Example:

```
salt '*' rbac.profile_get leo
salt '*' rbac.profile_get leo default_hidden=False
```

salt.modules.rbac_solaris.**profile_list**(*default_only=False*)

List all available profiles

default_only [boolean] return only default profile

CLI Example:

```
salt '*' rbac.profile_list
```

salt.modules.rbac_solaris.**profile_rm**(*user, profile*)

Remove profile from user

user [string] username

profile [string] profile name

CLI Example:

```
salt '*' rbac.profile_rm jorge 'Primary Administrator'
salt '*' rbac.profile_rm jorge 'User Management,User Security'
```

salt.modules.rbac_solaris.**role_add**(*user, role*)

Add role to user

user [string] username

role [string] role name

CLI Example:

```
salt '*' rbac.role_add martine netcfg
salt '*' rbac.role_add martine netcfg,zfssnap
```

salt.modules.rbac_solaris.**role_get**(*user*)

List roles for user

user [string] username

CLI Example:

```
salt '*' rbac.role_get leo
```

salt.modules.rbac_solaris.**role_list**()

List all available roles

CLI Example:

```
salt '*' rbac.role_list
```

salt.modules.rbac_solaris.**role_rm**(*user*, *role*)

Remove role from user

user [string] username

role [string] role name

CLI Example:

```
salt '*' rbac.role_rm jorge netcfg
salt '*' rbac.role_rm jorge netcfg,zfssnap
```

21.9.349 salt.modules.rbenv

Manage ruby installations with rbenv. rbenv is supported on Linux and macOS. rbenv doesn't work on Windows (and isn't really necessary on Windows as there is no system Ruby on Windows). On Windows, the RubyInstaller and/or Pik are both good alternatives to work with multiple versions of Ruby on the same box.

http://misheska.com/blog/2013/06/15/using-rbenv-to-manage-multiple-versions-of-ruby/

New in version 0.16.0.

salt.modules.rbenv.**default**(*ruby=None*, *runas=None*)

Returns or sets the currently defined default ruby

ruby The version to set as the default. Should match one of the versions listed by `rbenv.versions`. Leave blank to return the current default.

CLI Example:

```
salt '*' rbenv.default
salt '*' rbenv.default 2.0.0-p0
```

salt.modules.rbenv.**do**(*cmdline*, *runas=None*, *env=None*)

Execute a ruby command with rbenv's shims from the user or the system

CLI Example:

```
salt '*' rbenv.do 'gem list bundler'
salt '*' rbenv.do 'gem list bundler' deploy
```

salt.modules.rbenv.**do_with_ruby**(*ruby*, *cmdline*, *runas=None*)

Execute a ruby command with rbenv's shims using a specific ruby version

CLI Example:

```
salt '*' rbenv.do_with_ruby 2.0.0-p0 'gem list bundler'
salt '*' rbenv.do_with_ruby 2.0.0-p0 'gem list bundler' runas=deploy
```

salt.modules.rbenv.**install**(*runas=None*, *path=None*)

Install rbenv systemwide

CLI Example:

```
salt '*' rbenv.install
```

`salt.modules.rbenv.`**`install_ruby`**(*ruby*, *runas=None*)

Install a ruby implementation.

ruby The version of Ruby to install, should match one of the versions listed by `rbenv.list`

runas The user under which to run rbenv. If not specified, then rbenv will be run as the user under which Salt is running.

Additional environment variables can be configured in pillar / grains / master:

```
rbenv:
  build_env: 'CONFIGURE_OPTS="--no-tcmalloc" CFLAGS="-fno-tree-dce"'
```

CLI Example:

```
salt '*' rbenv.install_ruby 2.0.0-p0
```

`salt.modules.rbenv.`**`is_installed`**(*runas=None*)

Check if rbenv is installed

CLI Example:

```
salt '*' rbenv.is_installed
```

`salt.modules.rbenv.`**`list_`**(*runas=None*)

List the installable versions of ruby

runas The user under which to run rbenv. If not specified, then rbenv will be run as the user under which Salt is running.

CLI Example:

```
salt '*' rbenv.list
```

`salt.modules.rbenv.`**`rehash`**(*runas=None*)

Run `rbenv rehash` to update the installed shims

runas The user under which to run rbenv. If not specified, then rbenv will be run as the user under which Salt is running.

CLI Example:

```
salt '*' rbenv.rehash
```

`salt.modules.rbenv.`**`uninstall_ruby`**(*ruby*, *runas=None*)

Uninstall a ruby implementation.

ruby The version of ruby to uninstall. Should match one of the versions listed by *`rbenv.versions`*.

runas The user under which to run rbenv. If not specified, then rbenv will be run as the user under which Salt is running.

CLI Example:

```
salt '*' rbenv.uninstall_ruby 2.0.0-p0
```

`salt.modules.rbenv.`**`update`**(*runas=None*, *path=None*)

Updates the current versions of rbenv and ruby-build

runas The user under which to run rbenv. If not specified, then rbenv will be run as the user under which Salt is running.

CLI Example:

```
salt '*' rbenv.update
```

salt.modules.rbenv.**versions**(*runas=None*)

List the installed versions of ruby

CLI Example:

```
salt '*' rbenv.versions
```

21.9.350 salt.modules.rdp

Manage RDP Service on Windows servers

salt.modules.rdp.**disable**()

Disable RDP the service on the server

CLI Example:

```
salt '*' rdp.disable
```

salt.modules.rdp.**disconnect_session**(*session_id*)

Disconnect a session.

New in version 2016.11.0.

Parameters **session_id** – The numeric Id of the session.

Returns A boolean representing whether the disconnect succeeded.

CLI Example:

```
salt '*' rdp.disconnect_session session_id

salt '*' rdp.disconnect_session 99
```

salt.modules.rdp.**enable**()

Enable RDP the service on the server

CLI Example:

```
salt '*' rdp.enable
```

salt.modules.rdp.**get_session**(*session_id*)

Get information about a session.

New in version 2016.11.0.

Parameters **session_id** – The numeric Id of the session.

Returns A dictionary of session information.

CLI Example:

```
salt '*' rdp.get_session session_id

salt '*' rdp.get_session 99
```

salt.modules.rdp.**list_sessions**(*logged_in_users_only=False*)

List information about the sessions.

New in version 2016.11.0.

Parameters **logged_in_users_only** – If True, only return sessions with users logged in.

Returns A list containing dictionaries of session information.

CLI Example:

```
salt '*' rdp.list_sessions
```

salt.modules.rdp.**logoff_session**(*session_id*)

Initiate the logoff of a session.

New in version 2016.11.0.

Parameters **session_id** – The numeric Id of the session.

Returns A boolean representing whether the logoff succeeded.

CLI Example:

```
salt '*' rdp.logoff_session session_id

salt '*' rdp.logoff_session 99
```

salt.modules.rdp.**status**()

Show if rdp is enabled on the server

CLI Example:

```
salt '*' rdp.status
```

21.9.351 salt.modules.redis

Module to provide redis functionality to Salt

New in version 2014.7.0.

configuration This module requires the redis python module and uses the following defaults which may be overridden in the minion configuration:

```
redis.host: 'salt'
redis.port: 6379
redis.db: 0
redis.password: None
```

salt.modules.redismod.**bgrewriteaof**(*host=None*, *port=None*, *db=None*, *password=None*)

Asynchronously rewrite the append-only file

CLI Example:

```
salt '*' redis.bgrewriteaof
```

salt.modules.redismod.**bgsave**(*host=None*, *port=None*, *db=None*, *password=None*)

Asynchronously save the dataset to disk

CLI Example:

```
salt '*' redis.bgsave
```

salt.modules.redismod.**config_get**(*pattern='*'*, *host=None*, *port=None*, *db=None*, *password=None*)

Get redis server configuration values

CLI Example:

```
salt '*' redis.config_get
salt '*' redis.config_get port
```

`salt.modules.redismod.`**`config_set`**(*name, value, host=None, port=None, db=None, password=None*)

Set redis server configuration values

CLI Example:

```
salt '*' redis.config_set masterauth luv_kittens
```

`salt.modules.redismod.`**`dbsize`**(*host=None, port=None, db=None, password=None*)

Return the number of keys in the selected database

CLI Example:

```
salt '*' redis.dbsize
```

`salt.modules.redismod.`**`delete`**(**keys, **connection_args*)

Deletes the keys from redis, returns number of keys deleted

CLI Example:

```
salt '*' redis.delete foo
```

`salt.modules.redismod.`**`exists`**(*key, host=None, port=None, db=None, password=None*)

Return true if the key exists in redis

CLI Example:

```
salt '*' redis.exists foo
```

`salt.modules.redismod.`**`expire`**(*key, seconds, host=None, port=None, db=None, password=None*)

Set a keys time to live in seconds

CLI Example:

```
salt '*' redis.expire foo 300
```

`salt.modules.redismod.`**`expireat`**(*key, timestamp, host=None, port=None, db=None, password=None*)

Set a keys expire at given UNIX time

CLI Example:

```
salt '*' redis.expireat foo 1400000000
```

`salt.modules.redismod.`**`flushall`**(*host=None, port=None, db=None, password=None*)

Remove all keys from all databases

CLI Example:

```
salt '*' redis.flushall
```

`salt.modules.redismod.`**`flushdb`**(*host=None, port=None, db=None, password=None*)

Remove all keys from the selected database

CLI Example:

```
salt '*' redis.flushdb
```

`salt.modules.redismod.`**`get_key`**(*key, host=None, port=None, db=None, password=None*)

Get redis key value

CLI Example:

```
salt '*' redis.get_key foo
```

`salt.modules.redismod.`**`get_master_ip`**`(`*host=None, port=None, password=None*`)`

Get host information about slave

CLI Example:

```
salt '*' redis.get_master_ip
```

`salt.modules.redismod.`**`hdel`**`(`*key, *fields, **options*`)`

Delete one of more hash fields.

New in version 2017.7.0.

CLI Example:

```
salt '*' redis.hdel foo_hash bar_field1 bar_field2
```

`salt.modules.redismod.`**`hexists`**`(`*key, field, host=None, port=None, db=None, password=None*`)`

Determine if a hash fields exists.

New in version 2017.7.0.

CLI Example:

```
salt '*' redis.hexists foo_hash bar_field
```

`salt.modules.redismod.`**`hget`**`(`*key, field, host=None, port=None, db=None, password=None*`)`

Get specific field value from a redis hash, returns dict

CLI Example:

```
salt '*' redis.hget foo_hash bar_field
```

`salt.modules.redismod.`**`hgetall`**`(`*key, host=None, port=None, db=None, password=None*`)`

Get all fields and values from a redis hash, returns dict

CLI Example:

```
salt '*' redis.hgetall foo_hash
```

`salt.modules.redismod.`**`hincrby`**`(`*key, field, increment=1, host=None, port=None, db=None, password=None*`)`

Increment the integer value of a hash field by the given number.

New in version 2017.7.0.

CLI Example:

```
salt '*' redis.hincrby foo_hash bar_field 5
```

`salt.modules.redismod.`**`hincrbyfloat`**`(`*key, field, increment=1.0, host=None, port=None, db=None, password=None*`)`

Increment the float value of a hash field by the given number.

New in version 2017.7.0.

CLI Example:

```
salt '*' redis.hincrbyfloat foo_hash bar_field 5.17
```

`salt.modules.redismod.`**`hlen`**`(`*key, host=None, port=None, db=None, password=None*`)`

Returns number of fields of a hash.

New in version 2017.7.0.

CLI Example:

```
salt '*' redis.hlen foo_hash
```

`salt.modules.redismod.`**`hmget`**`(`*key, *fields, **options*`)`

Returns the values of all the given hash fields.

New in version 2017.7.0.

CLI Example:

```
salt '*' redis.hmget foo_hash bar_field1 bar_field2
```

`salt.modules.redismod.`**`hmset`**`(`*key, **fieldsvals*`)`

Sets multiple hash fields to multiple values.

New in version 2017.7.0.

CLI Example:

```
salt '*' redis.hmset foo_hash bar_field1=bar_value1 bar_field2=bar_value2
```

`salt.modules.redismod.`**`hscan`**`(`*key, cursor=0, match=None, count=None, host=None, port=None, db=None, password=None*`)`

Incrementally iterate hash fields and associated values.

New in version 2017.7.0.

CLI Example:

```
salt '*' redis.hscan foo_hash match='field_prefix_*' count=1
```

`salt.modules.redismod.`**`hset`**`(`*key, field, value, host=None, port=None, db=None, password=None*`)`

Set the value of a hash field.

New in version 2017.7.0.

CLI Example:

```
salt '*' redis.hset foo_hash bar_field bar_value
```

`salt.modules.redismod.`**`hsetnx`**`(`*key, field, value, host=None, port=None, db=None, password=None*`)`

Set the value of a hash field only if the field does not exist.

New in version 2017.7.0.

CLI Example:

```
salt '*' redis.hsetnx foo_hash bar_field bar_value
```

`salt.modules.redismod.`**`hvals`**`(`*key, host=None, port=None, db=None, password=None*`)`

Return all the values in a hash.

New in version 2017.7.0.

CLI Example:

```
salt '*' redis.hvals foo_hash bar_field1 bar_value1
```

`salt.modules.redismod.`**`info`**`(`*host=None, port=None, db=None, password=None*`)`

Get information and statistics about the server

CLI Example:

```
salt '*' redis.info
```

`salt.modules.redismod.`**`key_type`**`(`*key, host=None, port=None, db=None, password=None*`)`

Get redis key type

CLI Example:

```
salt '*' redis.type foo
```

`salt.modules.redismod.`**`keys`**`(`*pattern='*', host=None, port=None, db=None, password=None*`)`

Get redis keys, supports glob style patterns

CLI Example:

```
salt '*' redis.keys
salt '*' redis.keys test*
```

`salt.modules.redismod.`**`lastsave`**`(`*host=None, port=None, db=None, password=None*`)`

Get the UNIX time in seconds of the last successful save to disk

CLI Example:

```
salt '*' redis.lastsave
```

`salt.modules.redismod.`**`llen`**`(`*key, host=None, port=None, db=None, password=None*`)`

Get the length of a list in Redis

CLI Example:

```
salt '*' redis.llen foo_list
```

`salt.modules.redismod.`**`lrange`**`(`*key, start, stop, host=None, port=None, db=None, password=None*`)`

Get a range of values from a list in Redis

CLI Example:

```
salt '*' redis.lrange foo_list 0 10
```

`salt.modules.redismod.`**`ping`**`(`*host=None, port=None, db=None, password=None*`)`

Ping the server, returns False on connection errors

CLI Example:

```
salt '*' redis.ping
```

`salt.modules.redismod.`**`save`**`(`*host=None, port=None, db=None, password=None*`)`

Synchronously save the dataset to disk

CLI Example:

```
salt '*' redis.save
```

salt.modules.redismod.**sentinel_get_master_ip**(*master*, *host=None*, *port=None*, *password=None*)

Get ip for sentinel master

CLI Example:

```
salt '*' redis.sentinel_get_master_ip 'mymaster'
```

salt.modules.redismod.**set_key**(*key*, *value*, *host=None*, *port=None*, *db=None*, *password=None*)

Set redis key value

CLI Example:

```
salt '*' redis.set_key foo bar
```

salt.modules.redismod.**shutdown**(*host=None*, *port=None*, *db=None*, *password=None*)

Synchronously save the dataset to disk and then shut down the server

CLI Example:

```
salt '*' redis.shutdown
```

salt.modules.redismod.**slaveof**(*master_host=None*, *master_port=None*, *host=None*, *port=None*, *db=None*, *password=None*)

Make the server a slave of another instance, or promote it as master

CLI Example:

```
# Become slave of redis-n01.example.com:6379
salt '*' redis.slaveof redis-n01.example.com 6379
salt '*' redis.slaveof redis-n01.example.com
# Become master
salt '*' redis.slaveof
```

salt.modules.redismod.**smembers**(*key*, *host=None*, *port=None*, *db=None*, *password=None*)

Get members in a Redis set

CLI Example:

```
salt '*' redis.smembers foo_set
```

salt.modules.redismod.**time**(*host=None*, *port=None*, *db=None*, *password=None*)

Return the current server UNIX time in seconds

CLI Example:

```
salt '*' redis.time
```

salt.modules.redismod.**zcard**(*key*, *host=None*, *port=None*, *db=None*, *password=None*)

Get the length of a sorted set in Redis

CLI Example:

```
salt '*' redis.zcard foo_sorted
```

salt.modules.redismod.**zrange**(*key*, *start*, *stop*, *host=None*, *port=None*, *db=None*, *password=None*)

Get a range of values from a sorted set in Redis by index

CLI Example:

```
salt '*' redis.zrange foo_sorted 0 10
```

21.9.352 salt.modules.reg

Manage the Windows registry

Hives

Hives are the main sections of the registry and all begin with the word HKEY.

- HKEY_LOCAL_MACHINE
- HKEY_CURRENT_USER
- HKEY_USER

Keys

Keys are the folders in the registry. Keys can have many nested subkeys. Keys can have a value assigned to them under the (Default)

When passing a key on the CLI it must be quoted correctly depending on the backslashes being used (\ vs \\). The following are valid methods of passing the the key on the CLI:

Using single backslashes: `"SOFTWARE\Python"` `'SOFTWARE\Python'` (will not work on a Windows Master)

Using double backslashes: `SOFTWARE\\Python`

Values or Entries

Values or Entries are the name/data pairs beneath the keys and subkeys. All keys have a default name/data pair. The name is `(Default)` with a displayed value of `(value not set)`. The actual value is Null.

Example

The following example is an export from the Windows startup portion of the registry:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run]
"RTHDVCPL"="\"C:\\Program Files\\Realtek\\Audio\\HDA\\RtkNGUI64.exe\" -s"
"NvBackend"="\"C:\\Program Files (x86)\\NVIDIA Corporation\\Update Core\\NvBackend.
↪exe\""
"BTMTrayAgent"="rundll32.exe \"C:\\Program Files (x86)\\Intel\\Bluetooth\\btmshellex.
↪dll\",TrayApp"
```

In this example these are the values for each:

Hive: `HKEY_LOCAL_MACHINE`

Key and subkeys: `SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run`

Value:

- **There are 3 value names:**
  - *RTHDVCPL*

  - *NvBackend*
  - *BTMTrayAgent*
- Each value name has a corresponding value

depends

- salt.utils.win_reg

`salt.modules.reg.`**`broadcast_change`**`()`

Refresh the windows environment.

Note: This will only effect new processes and windows. Services will not see the change until the system restarts.

Returns True if successful, otherwise False

Return type bool

CLI Example:

```
salt '*' reg.broadcast_change
```

`salt.modules.reg.`**`delete_key_recursive`**`(`*hive, key, use_32bit_registry=False*`)`

New in version 2015.5.4.

Delete a registry key to include all subkeys and value/data pairs.

Parameters **`hive`** (*`str`*) – The name of the hive. Can be one of the following

- HKEY_LOCAL_MACHINE or HKLM
- HKEY_CURRENT_USER or HKCU
- HKEY_USER or HKU
- HKEY_CLASSES_ROOT or HKCR
- HKEY_CURRENT_CONFIG or HKCC

key (str): The key to remove (looks like a path)

use_32bit_registry (bool): Deletes the 32bit portion of the registry on 64bit installations. On 32bit machines this is ignored.

Returns

A dictionary listing the keys that deleted successfully as well as those that failed to delete.

Return type dict

CLI Example:

The following example will remove `delete_me` and all its subkeys from the `SOFTWARE` key in `HKEY_LOCAL_MACHINE`:

```
salt '*' reg.delete_key_recursive HKLM SOFTWARE\\delete_me
```

`salt.modules.reg.`**`delete_value`**`(`*hive, key, vname=None, use_32bit_registry=False*`)`

Delete a registry value entry or the default value for a key.

Parameters

- **`hive`** (*`str`*) – The name of the hive. Can be one of the following
  - HKEY_LOCAL_MACHINE or HKLM
  - HKEY_CURRENT_USER or HKCU
  - HKEY_USER or HKU
  - HKEY_CLASSES_ROOT or HKCR
  - HKEY_CURRENT_CONFIG or HKCC
- **`key`** (*`str`*) – The key (looks like a path) to the value name.

- **vname** (*str*) – The value name. These are the individual name/data pairs under the key. If not passed, the key (Default) value will be deleted.
- **use_32bit_registry** (*bool*) – Deletes the 32bit portion of the registry on 64bit installations. On 32bit machines this is ignored.

Returns True if successful, otherwise False

Return type bool

CLI Example:

```
salt '*' reg.delete_value HKEY_CURRENT_USER 'SOFTWARE\\Salt' 'version'
```

`salt.modules.reg.`**`import_file`**(*source*, *use_32bit_registry=False*)

Import registry settings from a Windows `REG` file by invoking `REG.EXE`.

New in version 2018.3.0.

Parameters

- **source** (*str*) – The full path of the `REG` file. This can be either a local file path or a URL type supported by salt (e.g. `salt://salt_master_path`)
- **use_32bit_registry** (*bool*) – If the value of this parameter is `True` then the `REG` file will be imported into the Windows 32 bit registry. Otherwise the Windows 64 bit registry will be used.

Returns True if successful, otherwise an error is raised

Return type bool

Raises

- **ValueError** – If the value of `source` is an invalid path or otherwise causes `cp.cache_file` to return `False`
- ***CommandExecutionError*** – If `reg.exe` exits with a non-0 exit code

CLI Example:

```
salt machine1 reg.import_file salt://win/printer_config/110_Canon/
↪postinstall_config.reg
```

`salt.modules.reg.`**`key_exists`**(*hive*, *key*, *use_32bit_registry=False*)

Check that the key is found in the registry. This refers to keys and not value/data pairs.

Parameters

- **hive** (*str*) – The hive to connect to
- **key** (*str*) – The key to check
- **use_32bit_registry** (*bool*) – Look in the 32bit portion of the registry

Returns True if exists, otherwise False

Return type bool

CLI Example:

```
salt '*' reg.key_exists HKLM SOFTWARE\Microsoft
```

`salt.modules.reg.`**`list_keys`**(*hive*, *key=None*, *use_32bit_registry=False*)

Enumerates the subkeys in a registry key or hive.

Parameters

- **hive** (*str*) – The name of the hive. Can be one of the following:
  - HKEY_LOCAL_MACHINE or HKLM
  - HKEY_CURRENT_USER or HKCU
  - HKEY_USER or HKU
  - HKEY_CLASSES_ROOT or HKCR
  - HKEY_CURRENT_CONFIG or HKCC
- **key** (*str*) – The key (looks like a path) to the value name. If a key is not passed, the keys under the hive will be returned.
- **use_32bit_registry** (*bool*) – Accesses the 32bit portion of the registry on 64 bit installations. On 32bit machines this is ignored.

Returns A list of keys/subkeys under the hive or key.

Return type list

CLI Example:

```
salt '*' reg.list_keys HKLM 'SOFTWARE'
```

`salt.modules.reg.`**`list_values`**(*hive*, *key=None*, *use_32bit_registry=False*)

Enumerates the values in a registry key or hive.

Note: The `(Default)` value will only be returned if it is set, otherwise it will not be returned in the list of values.

Parameters

- **`hive`** (*str*) – The name of the hive. Can be one of the following:
  - HKEY_LOCAL_MACHINE or HKLM
  - HKEY_CURRENT_USER or HKCU
  - HKEY_USER or HKU
  - HKEY_CLASSES_ROOT or HKCR
  - HKEY_CURRENT_CONFIG or HKCC
- **`key`** (*str*) – The key (looks like a path) to the value name. If a key is not passed, the values under the hive will be returned.
- **`use_32bit_registry`** (*bool*) – Accesses the 32bit portion of the registry on 64 bit installations. On 32bit machines this is ignored.

Returns A list of values under the hive or key.

Return type list

CLI Example:

```
salt '*' reg.list_values HKLM 'SYSTEM\\CurrentControlSet\\Services\\Tcpip
↪'
```

`salt.modules.reg.`**`read_value`**(*hive*, *key*, *vname=None*, *use_32bit_registry=False*)

Reads a registry value entry or the default value for a key. To read the default value, don't pass `vname`

Parameters

- **`hive`** (*str*) – The name of the hive. Can be one of the following:
  - HKEY_LOCAL_MACHINE or HKLM
  - HKEY_CURRENT_USER or HKCU
  - HKEY_USER or HKU
  - HKEY_CLASSES_ROOT or HKCR
  - HKEY_CURRENT_CONFIG or HKCC
- **`key`** (*str*) – The key (looks like a path) to the value name.
- **`vname`** (*str*) – The value name. These are the individual name/data pairs under the key. If not passed, the key (Default) value will be returned.
- **`use_32bit_registry`** (*bool*) – Accesses the 32bit portion of the registry on 64bit installations. On 32bit machines this is ignored.

Returns

A dictionary containing the passed settings as well as the value_data if successful. If unsuccessful, sets success to False.

bool: Returns False if the key is not found

If vname is not passed:

- Returns the first unnamed value (Default) as a string.
- Returns none if first unnamed value is empty.

Return type dict

CLI Example:

The following will get the value of the `version` value name in the `HKEY_LOCAL_MACHINE\\SOFTWARE\\Salt` key

```
salt '*' reg.read_value HKEY_LOCAL_MACHINE 'SOFTWARE\Salt' 'version'
```

CLI Example:

The following will get the default value of the `HKEY_LOCAL_MACHINE\\SOFTWARE\\Salt` key

```
salt '*' reg.read_value HKEY_LOCAL_MACHINE 'SOFTWARE\Salt'
```

`salt.modules.reg.`**`set_value`**`(`*hive, key, vname=None, vdata=None, vtype='REG_SZ', use_32bit_registry=False, volatile=False*`)`

Sets a value in the registry. If `vname` is passed, it will be the value for that value name, otherwise it will be the default value for the specified key

Parameters

- **`hive`** (*`str`*) – The name of the hive. Can be one of the following
  - HKEY_LOCAL_MACHINE or HKLM
  - HKEY_CURRENT_USER or HKCU
  - HKEY_USER or HKU
  - HKEY_CLASSES_ROOT or HKCR
  - HKEY_CURRENT_CONFIG or HKCC
- **`key`** (*`str`*) – The key (looks like a path) to the value name.
- **`vname`** (*`str`*) – The value name. These are the individual name/data pairs under the key. If not passed, the key (Default) value will be set.
- **`vdata`** (*`str, int, list, bytes`*) – The value you'd like to set. If a value name (vname) is passed, this will be the data for that value name. If not, this will be the (Default) value for the key.

  The type of data this parameter expects is determined by the value type specified in `vtype`. The correspondence is as follows:
  - REG_BINARY: Binary data (str in Py2, bytes in Py3)
  - REG_DWORD: int
  - REG_EXPAND_SZ: str
  - REG_MULTI_SZ: list of str
  - REG_QWORD: int
  - REG_SZ: str

  Note: When setting REG_BINARY, string data will be converted to binary.

  Note: The type for the (Default) value is always REG_SZ and cannot be changed.

  Note: This parameter is optional. If `vdata` is not passed, the Key will be created with no associated item/value pairs.
- **`vtype`** (*`str`*) – The value type. The possible values of the vtype parameter are indicated above in the description of the vdata parameter.
- **`use_32bit_registry`** (*`bool`*) – Sets the 32bit portion of the registry on 64bit installations. On 32bit machines this is ignored.
- **`volatile`** (*`bool`*) – When this parameter has a value of True, the registry key will be made volatile (i.e. it will not persist beyond a system reset or shutdown). This

parameter only has an effect when a key is being created and at no other time.

Returns True if successful, otherwise False

Return type bool

CLI Example:

This will set the version value to 2015.5.2 in the SOFTWARESalt key in the HKEY_LOCAL_MACHINE hive

```
salt '*' reg.set_value HKEY_LOCAL_MACHINE 'SOFTWARE\Salt' 'version'
↪'2015.5.2'
```

CLI Example:

This function is strict about the type of vdata. For instance this example will fail because vtype has a value of REG_SZ and vdata has a type of int (as opposed to str as expected).

```
salt '*' reg.set_value HKEY_LOCAL_MACHINE 'SOFTWARE\Salt' 'str_data' 1.2
```

CLI Example:

In this next example vdata is properly quoted and should succeed.

```
salt '*' reg.set_value HKEY_LOCAL_MACHINE 'SOFTWARE\Salt' 'str_data'
↪vtype=REG_SZ vdata="'1.2'"
```

CLI Example:

This is an example of using vtype REG_BINARY.

```
salt '*' reg.set_value HKEY_LOCAL_MACHINE 'SOFTWARE\Salt' 'bin_data'
↪vtype=REG_BINARY vdata='Salty Data'
```

CLI Example:

An example of using vtype REG_MULTI_SZ is as follows:

```
salt '*' reg.set_value HKEY_LOCAL_MACHINE 'SOFTWARE\Salt' 'list_data'
↪vtype=REG_MULTI_SZ vdata='["Salt", "is", "great"]'
```

`salt.modules.reg.`**`value_exists`**(*hive, key, vname, use_32bit_registry=False*)

Check that the value/data pair is found in the registry.

New in version 3000.

Parameters

- **hive** (*str*) – The hive to connect to
- **key** (*str*) – The key to check in
- **vname** (*str*) – The name of the value/data pair you're checking
- **use_32bit_registry** (*bool*) – Look in the 32bit portion of the registry

Returns True if exists, otherwise False

Return type bool

CLI Example:

```
salt '*' reg.value_exists HKLM SOFTWARE\Microsoft\Windows\CurrentVersion
↪CommonFilesDir
```

21.9.353 salt.modules.rest_pkg

Package support for the REST example

`salt.modules.rest_pkg.`**`install`**(*name=None, refresh=False, fromrepo=None, pkgs=None, sources=None, **kwargs*)

`salt.modules.rest_pkg.`**`installed`**(*name, version=None, refresh=False, fromrepo=None, skip_verify=False, pkgs=None, sources=None, **kwargs*) `salt.modules.rest_pkg.`**`list_pkgs`**`(`*versions_as_list=False, **kwargs*`)`

`salt.modules.rest_pkg.`**`remove`**`(`*name=None, pkgs=None, **kwargs*`)`

`salt.modules.rest_pkg.`**`upgrade`**`(`*refresh=True, skip_verify=True, **kwargs*`)`

`salt.modules.rest_pkg.`**`version`**`(`**names, **kwargs*`)`

Returns a string representing the package version or an empty string if not installed. If more than one package name is specified, a dict of name/version pairs is returned.

CLI Example:

```
salt '*' pkg.version <package name>
salt '*' pkg.version <package1> <package2> <package3> ...
```

21.9.354 salt.modules.rest_sample_utils module

Utility functions for the rest_sample

`salt.modules.rest_sample_utils.`**`fix_outage`**`()`

"Fix" the outage

CLI Example:

```
salt 'rest-sample-proxy' rest_sample.fix_outage
```

`salt.modules.rest_sample_utils.`**`get_test_string`**`()`

Helper function to test cross-calling to the __proxy__ dunder.

CLI Example:

```
salt 'rest-sample-proxy' rest_sample.get_test_string
```

21.9.355 salt.modules.rest_service

Provide the service module for the proxy-minion REST sample

`salt.modules.rest_service.`**`enabled`**`(`*name, sig=None*`)`

Only the 'redbull' service is 'enabled' in the test

New in version 2015.8.1.

`salt.modules.rest_service.`**`get_all`**`()`

Return a list of all available services

New in version 2015.8.0.

CLI Example:

```
salt '*' service.get_all
```

`salt.modules.rest_service.`**`list_`**`()`

Return a list of all available services.

CLI Example:

```
salt '*' service.list
```

`salt.modules.rest_service.`**`restart`**(*name*, *sig=None*)

Restart the specified service with rest_sample

New in version 2015.8.0.

CLI Example:

```
salt '*' service.restart <service name>
```

`salt.modules.rest_service.`**`running`**(*name*, *sig=None*)

Return whether this service is running.

New in version 2015.8.0.

`salt.modules.rest_service.`**`start`**(*name*, *sig=None*)

Start the specified service on the rest_sample

New in version 2015.8.0.

CLI Example:

```
salt '*' service.start <service name>
```

`salt.modules.rest_service.`**`status`**(*name*, *sig=None*)

Return the status for a service via rest_sample. If the name contains globbing, a dict mapping service name to True/False values is returned.

New in version 2015.8.0.

Changed in version 2018.3.0: The service name can now be a glob (e.g. `salt*`)

Parameters

- **`name`** (`str`) – The name of the service to check
- **`sig`** (`str`) – Not implemented

Returns True if running, False otherwise dict: Maps service name to True if running, False otherwise

Return type bool

CLI Example:

```
salt '*' service.status <service name>
```

`salt.modules.rest_service.`**`stop`**(*name*, *sig=None*)

Stop the specified service on the rest_sample

New in version 2015.8.0.

CLI Example:

```
salt '*' service.stop <service name>
```

21.9.356 salt.modules.restartcheck module

checkrestart functionality for Debian and Red Hat Based systems

Identifies services (processes) that are linked against deleted files (for example after downloading an updated binary of a shared library).

Based on checkrestart script from debian-goodies (written by Matt Zimmerman for the Debian GNU/Linux distribution, https://packages.debian.org/debian-goodies) and psdel by Sam Morris.

codeauthor Jiri Kotlin <jiri.kotlin@ultimum.io>

`salt.modules.restartcheck.`**`restartcheck`**(*ignorelist=None*, *blacklist=None*, *excludepid=None*, ***kwargs*)

Analyzes files openeded by running processes and seeks for packages which need to be restarted.

Parameters

- **`ignorelist`** – string or list of packages to be ignored
- **`blacklist`** – string or list of file paths to be ignored
- **`excludepid`** – string or list of process IDs to be ignored
- **`verbose`** – boolean, enables extensive output
- **`timeout`** – int, timeout in minute

Returns { 'result': False, 'comment': '<reason>' } String with checkrestart output if some package seems to need to be restarted or if no packages need restarting.

Return type Dict on error

New in version 2015.8.3.

CLI Example: .. code-block:: bash

salt '*' restartcheck.restartcheck

21.9.357 salt.modules.ret

Module to integrate with the returner system and retrieve data sent to a salt returner

`salt.modules.ret.`**`get_fun`**(*returner*, *fun*)

Return info about last time fun was called on each minion

CLI Example:

```
salt '*' ret.get_fun mysql network.interfaces
```

`salt.modules.ret.`**`get_jid`**(*returner*, *jid*)

Return the information for a specified job id

CLI Example:

```
salt '*' ret.get_jid redis 20421104181954700505
```

`salt.modules.ret.`**`get_jids`**(*returner*)

Return a list of all job ids

CLI Example:

```
salt '*' ret.get_jids mysql
```

`salt.modules.ret.`**`get_minions`**(*returner*)

Return a list of all minions

CLI Example:

```
salt '*' ret.get_minions mysql
```

21.9.358 salt.modules.rh_ip

The networking module for RHEL/Fedora based distros

`salt.modules.rh_ip.`**`apply_network_settings`**(***settings*)

Apply global network configuration.

CLI Example:

```
salt '*' ip.apply_network_settings
```

salt.modules.rh_ip.**build_bond**(*iface*, ***settings*)
Create a bond script in /etc/modprobe.d with the passed settings and load the bonding kernel module.

CLI Example:

```
salt '*' ip.build_bond bond0 mode=balance-alb
```

salt.modules.rh_ip.**build_interface**(*iface*, *iface_type*, *enabled*, ***settings*)
Build an interface script for a network interface.

CLI Example:

```
salt '*' ip.build_interface eth0 eth <settings>
```

salt.modules.rh_ip.**build_network_settings**(***settings*)
Build the global network script.

CLI Example:

```
salt '*' ip.build_network_settings <settings>
```

salt.modules.rh_ip.**build_routes**(*iface*, ***settings*)
Build a route script for a network interface.

CLI Example:

```
salt '*' ip.build_routes eth0 <settings>
```

salt.modules.rh_ip.**down**(*iface*, *iface_type*)
Shutdown a network interface

CLI Example:

```
salt '*' ip.down eth0
```

salt.modules.rh_ip.**get_bond**(*iface*)
Return the content of a bond script

CLI Example:

```
salt '*' ip.get_bond bond0
```

salt.modules.rh_ip.**get_interface**(*iface*)
Return the contents of an interface script

CLI Example:

```
salt '*' ip.get_interface eth0
```

salt.modules.rh_ip.**get_network_settings**()
Return the contents of the global network script.

CLI Example:

```
salt '*' ip.get_network_settings
```

`salt.modules.rh_ip.`**`get_routes`**`(`*iface*`)`

Return the contents of the interface routes script.

CLI Example:

```
salt '*' ip.get_routes eth0
```

`salt.modules.rh_ip.`**`up`**`(`*iface, iface_type*`)`

Start up a network interface

CLI Example:

```
salt '*' ip.up eth0
```

21.9.359 salt.modules.rh_service

Service support for RHEL-based systems, including support for both upstart and sysvinit

---

Important: If you feel that Salt should be using this module to manage services on a minion, and it is using a different module (or gives an error similar to *'service.start' is not available*), see *here*.

`salt.modules.rh_service.`**`available`**`(`*name, limit=''*`)`

Return True if the named service is available. Use the `limit` param to restrict results to services of that type.

CLI Examples:

```
salt '*' service.available sshd
salt '*' service.available sshd limit=upstart
salt '*' service.available sshd limit=sysvinit
```

`salt.modules.rh_service.`**`delete`**`(`*name, **kwargs*`)`

Delete the named service

New in version 2016.3.

CLI Example:

```
salt '*' service.delete <service name>
```

`salt.modules.rh_service.`**`disable`**`(`*name, **kwargs*`)`

Disable the named service to start at boot

CLI Example:

```
salt '*' service.disable <service name>
```

`salt.modules.rh_service.`**`disabled`**`(`*name*`)`

Check to see if the named service is disabled to start on boot

CLI Example:

```
salt '*' service.disabled <service name>
```

`salt.modules.rh_service.`**`enable`**`(`*name, **kwargs*`)`

Enable the named service to start at boot

CLI Example:

```
salt '*' service.enable <service name>
```

`salt.modules.rh_service.`**`enabled`**`(`*name*, ***kwargs*`)`
Check to see if the named service is enabled to start on boot

CLI Example:

```
salt '*' service.enabled <service name>
```

`salt.modules.rh_service.`**`get_all`**`(`*limit=''*`)`
Return all installed services. Use the `limit` param to restrict results to services of that type.

CLI Example:

```
salt '*' service.get_all
salt '*' service.get_all limit=upstart
salt '*' service.get_all limit=sysvinit
```

`salt.modules.rh_service.`**`get_disabled`**`(`*limit=''*`)`
Return the disabled services. Use the `limit` param to restrict results to services of that type.

CLI Example:

```
salt '*' service.get_disabled
salt '*' service.get_disabled limit=upstart
salt '*' service.get_disabled limit=sysvinit
```

`salt.modules.rh_service.`**`get_enabled`**`(`*limit=''*`)`
Return the enabled services. Use the `limit` param to restrict results to services of that type.

CLI Examples:

```
salt '*' service.get_enabled
salt '*' service.get_enabled limit=upstart
salt '*' service.get_enabled limit=sysvinit
```

`salt.modules.rh_service.`**`missing`**`(`*name*, *limit=''*`)`
The inverse of service.available. Return True if the named service is not available. Use the `limit` param to restrict results to services of that type.

CLI Examples:

```
salt '*' service.missing sshd
salt '*' service.missing sshd limit=upstart
salt '*' service.missing sshd limit=sysvinit
```

`salt.modules.rh_service.`**`reload_`**`(`*name*`)`
Reload the named service

CLI Example:

```
salt '*' service.reload <service name>
```

`salt.modules.rh_service.`**`restart`**`(`*name*`)`
Restart the named service

CLI Example:

```
salt '*' service.restart <service name>
```

`salt.modules.rh_service.`**`start`**`(`*name*`)`

Start the specified service

CLI Example:

```
salt '*' service.start <service name>
```

`salt.modules.rh_service.`**`status`**`(`*name, sig=None*`)`

Return the status for a service. If the name contains globbing, a dict mapping service name to True/False values is returned.

Changed in version 2018.3.0: The service name can now be a glob (e.g. `salt*`)

Parameters

- **name** (`str`) – The name of the service to check
- **sig** (`str`) – Signature to use to find the service via ps

Returns True if running, False otherwise dict: Maps service name to True if running, False otherwise

Return type bool

CLI Example:

```
salt '*' service.status <service name> [service signature]
```

`salt.modules.rh_service.`**`stop`**`(`*name*`)`

Stop the specified service

CLI Example:

```
salt '*' service.stop <service name>
```

21.9.360 salt.modules.riak

Riak Salt Module

`salt.modules.riak.`**`cluster_commit`**`()`

Commit Cluster Changes

Changed in version 2015.8.0.

CLI Example:

```
salt '*' riak.cluster_commit
```

`salt.modules.riak.`**`cluster_join`**`(`*username, hostname*`)`

Join a Riak cluster

Changed in version 2015.8.0.

CLI Example:

```
salt '*' riak.cluster_join <user> <host>
```

username - The riak username to join the cluster hostname - The riak hostname you are connecting to

`salt.modules.riak.`**`cluster_leave`**`(`*username, hostname*`)`

Leave a Riak cluster

New in version 2015.8.0.

CLI Example:

```
salt '*' riak.cluster_leave <username> <host>
```

username - The riak username to join the cluster hostname - The riak hostname you are connecting to

`salt.modules.riak.`**`cluster_plan`**`()`

Review Cluster Plan

Changed in version 2015.8.0.

CLI Example:

```
salt '*' riak.cluster_plan
```

`salt.modules.riak.`**`member_status`**`()`

Get cluster member status

Changed in version 2015.8.0.

CLI Example:

```
salt '*' riak.member_status
```

`salt.modules.riak.`**`services`**`()`

List available services on a node

New in version 2015.8.0.

CLI Example:

```
salt '*' riak.services
```

`salt.modules.riak.`**`start`**`()`

Start Riak

CLI Example:

```
salt '*' riak.start
```

`salt.modules.riak.`**`status`**`()`

Current node status

New in version 2015.8.0.

CLI Example:

```
salt '*' riak.status
```

`salt.modules.riak.`**`stop`**`()`

Stop Riak

Changed in version 2015.8.0.

CLI Example:

```
salt '*' riak.stop
```

`salt.modules.riak.`**`test`**`()`

Runs a test of a few standard Riak operations

New in version 2015.8.0.

CLI Example:

```
salt '*' riak.test
```

21.9.361 salt.modules.rpm_lowpkg

Support for rpm

`salt.modules.rpm_lowpkg.`**`bin_pkg_info`**`(`*path, saltenv='base'*`)`

New in version 2015.8.0.

Parses RPM metadata and returns a dictionary of information about the package (name, version, etc.).

path Path to the file. Can either be an absolute path to a file on the minion, or a salt fileserver URL (e.g. `salt://path/to/file.rpm`). If a salt fileserver URL is passed, the file will be cached to the minion so that it can be examined.

saltenv [base] Salt fileserver envrionment from which to retrieve the package. Ignored if `path` is a local file path on the minion.

CLI Example:

```
salt '*' lowpkg.bin_pkg_info /root/salt-2015.5.1-2.el7.noarch.rpm
salt '*' lowpkg.bin_pkg_info salt://salt-2015.5.1-2.el7.noarch.rpm
```

`salt.modules.rpm_lowpkg.`**`checksum`**`(`**paths*`)`

Return if the signature of a RPM file is valid.

CLI Example:

```
salt '*' lowpkg.checksum /path/to/package1.rpm
salt '*' lowpkg.checksum /path/to/package1.rpm /path/to/package2.rpm
```

`salt.modules.rpm_lowpkg.`**`diff`**`(`*package_path, path*`)`

Return a formatted diff between current file and original in a package. NOTE: this function includes all files (configuration and not), but does not work on binary content.

Parameters

- **package** – Full pack of the RPM file
- **path** – Full path to the installed file

Returns Difference or empty string. For binary files only a notification.

CLI example:

```
salt '*' lowpkg.diff /path/to/apache2.rpm /etc/apache2/httpd.conf
```

`salt.modules.rpm_lowpkg.`**`file_dict`**`(`**packages*`)`

List the files that belong to a package, sorted by group. Not specifying any packages will return a list of _every_ file on the system's rpm database (not generally recommended).

CLI Examples:

```
salt '*' lowpkg.file_dict httpd
salt '*' lowpkg.file_dict httpd postfix
salt '*' lowpkg.file_dict
```

`salt.modules.rpm_lowpkg.`**`file_list`**`(`**packages*`)`

List the files that belong to a package. Not specifying any packages will return a list of _every_ file on the system's rpm database (not generally recommended).

CLI Examples:

```
salt '*' lowpkg.file_list httpd
salt '*' lowpkg.file_list httpd postfix
salt '*' lowpkg.file_list
```

`salt.modules.rpm_lowpkg.`**`info`**`(`**packages, **kwargs*`)`

Return a detailed package(s) summary information. If no packages specified, all packages will be returned.

Parameters

- **`packages`** –
- **`attr`** – Comma-separated package attributes. If no 'attr' is specified, all available attributes returned.

  Valid attributes are: version, vendor, release, build_date, build_date_time_t, install_date, install_date_time_t, build_host, group, source_rpm, arch, epoch, size, license, signature, packager, url, summary, description.
- **`all_versions`** – Return information for all installed versions of the packages

Returns

CLI example:

```
salt '*' lowpkg.info apache2 bash
salt '*' lowpkg.info apache2 bash attr=version
salt '*' lowpkg.info apache2 bash attr=version,build_date_iso,size
salt '*' lowpkg.info apache2 bash attr=version,build_date_iso,size all_
↪versions=True
```

`salt.modules.rpm_lowpkg.`**`list_pkgs`**`(`**packages*`)`

List the packages currently installed in a dict:

```
{'<package_name>': '<version>'}
```

CLI Example:

```
salt '*' lowpkg.list_pkgs
```

`salt.modules.rpm_lowpkg.`**`modified`**`(`**packages, **flags*`)`

List the modified files that belong to a package. Not specifying any packages will return a list of _all_ modified files on the system's RPM database.

New in version 2015.5.0.

CLI examples:

```
salt '*' lowpkg.modified httpd
salt '*' lowpkg.modified httpd postfix
salt '*' lowpkg.modified
```

`salt.modules.rpm_lowpkg.`**`owner`**`(`**paths*`)`

Return the name of the package that owns the file. Multiple file paths can be passed. If a single path is passed, a string will be returned, and if multiple paths are passed, a dictionary of file/package name pairs will be returned.

If the file is not owned by a package, or is not present on the minion, then an empty string will be returned for that path.

CLI Examples:

```
salt '*' lowpkg.owner /usr/bin/apachectl
salt '*' lowpkg.owner /usr/bin/apachectl /etc/httpd/conf/httpd.conf
```

`salt.modules.rpm_lowpkg.`**`verify`**`(`**packages, **kwargs*`)`

Runs an rpm -Va on a system, and returns the results in a dict

Files with an attribute of config, doc, ghost, license or readme in the package header can be ignored using the `ignore_types` keyword argument

CLI Example:

```
salt '*' lowpkg.verify
salt '*' lowpkg.verify httpd
salt '*' lowpkg.verify httpd postfix
salt '*' lowpkg.verify httpd postfix ignore_types=['config','doc']
```

`salt.modules.rpm_lowpkg.`**`version_cmp`**`(`*ver1, ver2, ignore_epoch=False*`)`

New in version 2015.8.9.

Do a cmp-style comparison on two packages. Return -1 if ver1 < ver2, 0 if ver1 == ver2, and 1 if ver1 > ver2. Return None if there was a problem making the comparison.

ignore_epoch [False] Set to `True` to ignore the epoch when comparing versions

New in version 2015.8.10,2016.3.2.

CLI Example:

```
salt '*' pkg.version_cmp '0.2-001' '0.2.0.1-002'
```

21.9.362 salt.modules.rpmbuild_pkgbuild

RPM Package builder system

New in version 2015.8.0.

This system allows for all of the components to build rpms safely in chrooted environments. This also provides a function to generate yum repositories

This module implements the pkgbuild interface

`salt.modules.rpmbuild_pkgbuild.`**`build`**`(`*runas, tgt, dest_dir, spec, sources, deps, env, template, saltenv='base', log_dir='/var/log/salt/pkgbuild'*`)`

Given the package destination directory, the spec file source and package sources, use mock to safely build the rpm defined in the spec file

CLI Example:

```
salt '*' pkgbuild.build mock epel-7-x86_64 /var/www/html
            https://raw.githubusercontent.com/saltstack/libnacl/master/pkg/rpm/
↪python-libnacl.spec
            https://pypi.python.org/packages/source/l/libnacl/libnacl-1.3.5.tar.gz
```

This example command should build the libnacl package for rhel 7 using user mock and place it in /var/www/html/ on the minion

`salt.modules.rpmbuild_pkgbuild.`**`make_repo`**`(`*repodir, keyid=None, env=None, use_passphrase=False, gnupghome='/etc/salt/gpgkeys', runas='root', timeout=15.0*`)`

Make a package repository and optionally sign packages present

Given the repodir, create a `yum` repository out of the rpms therein and optionally sign it and packages present, the name is directory to turn into a repo. This state is best used with onchanges linked to your package building states.

repodir The directory to find packages that will be in the repository.

keyid Changed in version 2016.3.0.

Optional Key ID to use in signing packages and repository. Utilizes Public and Private keys associated with keyid which have been loaded into the minion's Pillar data.

For example, contents from a Pillar data file with named Public and Private keys as follows:

```
gpg_pkg_priv_key: |
  -----BEGIN PGP PRIVATE KEY BLOCK-----
  Version: GnuPG v1

  lQO+BFciIfQBCADAPCtzx7I5Rl32escCMZsPzaEKWe7bIX1em4KCKkBoX47IG54b
  w82PCE8Y1jF/9Uk2m3RKVWp3YcLlc7Ap3gj6VO4ysvVz28UbnhPxsIkOlf2cq8qc
  .
  .
  Ebe+8JCQTwqSXPRTzXmy/b5WXDeM79CkLWvuGpXFor76D+ECMRPv/rawukEcNptn
  R5OmgHqvydEnO4pWbn8JzQO9YX/Us0SMHBVzLC8eIi5ZIopzalvX
  =JvW8
  -----END PGP PRIVATE KEY BLOCK-----

gpg_pkg_priv_keyname: gpg_pkg_key.pem

gpg_pkg_pub_key: |
  -----BEGIN PGP PUBLIC KEY BLOCK-----
  Version: GnuPG v1

  mQENBFciIfQBCADAPCtzx7I5Rl32escCMZsPzaEKWe7bIX1em4KCKkBoX47IG54b
  w82PCE8Y1jF/9Uk2m3RKVWp3YcLlc7Ap3gj6VO4ysvVz28UbnhPxsIkOlf2cq8qc
  .
  .
  bYP7t5iwJmQzRMyFInYRt77wkJBPCpJc9FPNebL9vlZcN4zv0KQta+4alcWivvoP
  4QIxE+/+trC6QRw2m2dHk6aAeq/J0Sc7ilZufwnNA71hf9SzRIwcFXMsLx4iLlki
  inNqW9c=
  =s1CX
  -----END PGP PUBLIC KEY BLOCK-----

gpg_pkg_pub_keyname: gpg_pkg_key.pub
```

env Changed in version 2016.3.0.

A dictionary of environment variables to be utilized in creating the repository.

Note: This parameter is not used for making `yum` repositories.

use_passphrase [False] New in version 2016.3.0.

Use a passphrase with the signing key presented in `keyid`. Passphrase is received from Pillar data which could be passed on the command line with `pillar` parameter. For example:

```
pillar='{ "gpg_passphrase" : "my_passphrase" }'
```

gnupghome [/etc/salt/gpgkeys] New in version 2016.3.0.

Location where GPG related files are stored, used with `keyid`.

runas [root] New in version 2016.3.0.

User to create the repository as, and optionally sign packages.

Note: Ensure the user has correct permissions to any files and directories which are to be utilized.

timeout [15.0] New in version 2016.3.4.

Timeout in seconds to wait for the prompt for inputting the passphrase.

CLI Example:

```
salt '*' pkgbuild.make_repo /var/www/html/
```

salt.modules.rpmbuild_pkgbuild.**make_src_pkg**(*dest_dir*, *spec*, *sources*, *env=None*, *template=None*, *saltenv='base'*, *runas='root'*)

Create a source rpm from the given spec file and sources

CLI Example:

```
salt '*' pkgbuild.make_src_pkg /var/www/html/
        https://raw.githubusercontent.com/saltstack/libnacl/master/pkg/rpm/python-
↪libnacl.spec
        https://pypi.python.org/packages/source/l/libnacl/libnacl-1.3.5.tar.gz
```

This example command should build the libnacl SOURCE package and place it in /var/www/html/ on the minion

Changed in version 2017.7.0.

dest_dir The directory on the minion to place the built package(s)

spec The location of the spec file (used for rpms)

sources The list of package sources

env A dictionary of environment variables to be set prior to execution.

template Run the spec file through a templating engine Optional arguement, allows for no templating engine used to be if none is desired.

saltenv The saltenv to use for files downloaded from the salt filesever

runas The user to run the build process as

New in version 2018.3.3.

Note: using SHA256 as digest and minimum level dist el6

21.9.363 salt.modules.rsync

Wrapper for rsync

New in version 2014.1.0.

This data can also be passed into *pillar*. Options passed into opts will overwrite options passed into pillar.

salt.modules.rsync.**config**(*conf_path='/etc/rsyncd.conf'*)

Changed in version 2016.3.0: Return data now contains just the contents of the rsyncd.conf as a string, instead of a dictionary as returned from `cmd.run_all`.

Returns the contents of the rsync config file

conf_path [/etc/rsyncd.conf] Path to the config file

CLI Example:

```
salt '*' rsync.config
```

`salt.modules.rsync.`**`rsync`**`(`*src, dst, delete=False, force=False, update=False, passwordfile=None, exclude=None, excludefrom=None, dryrun=False, rsh=None, additional_opts=None, saltenv='base'*`)`

Changed in version 2016.3.0: Return data now contains just the output of the rsync command, instead of a dictionary as returned from `cmd.run_all`.

Rsync files from src to dst

src The source location where files will be rsynced from.

dst The destination location where files will be rsynced to.

delete [False] Whether to enable the rsync –*delete* flag, which will delete extraneous files from dest dirs

force [False] Whether to enable the rsync –*force* flag, which will force deletion of dirs even if not empty.

update [False] Whether to enable the rsync –*update* flag, which forces rsync to skip any files which exist on the destination and have a modified time that is newer than the source file.

passwordfile A file that contains a password for accessing an rsync daemon. The file should contain just the password.

exclude Whether to enable the rsync –*exclude* flag, which will exclude files matching a PATTERN.

excludefrom Whether to enable the rsync –*excludefrom* flag, which will read exclude patterns from a file.

dryrun [False] Whether to enable the rsync –*dry-run* flag, which will perform a trial run with no changes made.

rsh Whether to enable the rsync –*rsh* flag, to specify the remote shell to use.

additional_opts Any additional rsync options, should be specified as a list.

saltenv

Specify a salt fileserver environment to be used.

CLI Example:

```
salt '*' rsync.rsync /path/to/src /path/to/dest delete=True update=True↩
↪passwordfile=/etc/pass.crt exclude=exclude/dir
salt '*' rsync.rsync /path/to/src delete=True excludefrom=/xx.ini
salt '*' rsync.rsync /path/to/src delete=True exclude='[exclude1/dir,
↪exclude2/dir]' additional_opts='["--partial", "--bwlimit=5000"]'
```

`salt.modules.rsync.`**`version`**`()`

Changed in version 2016.3.0: Return data now contains just the version number as a string, instead of a dictionary as returned from `cmd.run_all`.

Returns rsync version

CLI Example:

```
salt '*' rsync.version
```

21.9.364 salt.modules.runit

runit service module (http://smarden.org/runit)

This module is compatible with the `service` states, so it can be used to maintain services using the `provider` argument:

```
myservice:
  service:
    - running
    - provider: runit
```

Provides virtual *service* module on systems using runit as init.

Service management rules (*sv* command):

> service $n is ENABLED if file SERVICE_DIR/$n/run exists service $n is AVAILABLE if ENABLED or if file AVAIL_SVR_DIR/$n/run exists service $n is DISABLED if AVAILABLE but not ENABLED
>
> SERVICE_DIR/$n is normally a symlink to a AVAIL_SVR_DIR/$n folder

Service auto-start/stop mechanism:

> *sv* (auto)starts/stops service as soon as SERVICE_DIR/<service> is created/deleted, both on service creation or a boot time.
>
> autostart feature is disabled if file SERVICE_DIR/<n>/down exists. This does not affect the current's service status (if already running) nor manual service management.

Service's alias:

> Service *sva* is an alias of service *svc* when *AVAIL_SVR_DIR/sva* symlinks to folder *AVAIL_SVR_DIR/svc*. *svc* can't be enabled if it is already enabled through an alias already enabled, since *sv* files are stored in folder *SERVICE_DIR/svc/*.
>
> XBPS package management uses a service's alias to provides service alternative(s), such as chrony and openntpd both aliased to ntpd.

`salt.modules.runit.`**`add_svc_avail_path`**`(`*path*`)`

Add a path that may contain available services. Return `True` if added (or already present), `False` on error.

path directory to add to AVAIL_SVR_DIRS

`salt.modules.runit.`**`available`**`(`*name*`)`

Returns `True` if the specified service is available, otherwise returns `False`.

name the service's name

CLI Example:

```
salt '*' runit.available <service name>
```

`salt.modules.runit.`**`disable`**`(`*name, stop=False, **kwargs*`)`

Don't start service `name` at boot Returns `True` if operation is successful

name the service's name

stop if True, also stops the service

CLI Example:

```
salt '*' service.disable <name> [stop=True]
```

`salt.modules.runit.`**`disabled`**`(`*name*`)`

Return `True` if the named service is disabled, `False` otherwise

name the service's name

CLI Example:

```
salt '*' service.disabled <service name>
```

`salt.modules.runit.`**`enable`**`(`*name, start=False, **kwargs*`)`

Start service `name` at boot. Returns `True` if operation is successful

name the service's name

start [False] If `True`, start the service once enabled.

CLI Example:

```
salt '*' service.enable <name> [start=True]
```

`salt.modules.runit.`**`enabled`**`(`*name*`)`
Return `True` if the named service is enabled, `False` otherwise
name the service's name
CLI Example:

```
salt '*' service.enabled <service name>
```

`salt.modules.runit.`**`full_restart`**`(`*name*`)`
Calls runit.restart()
name the service's name
CLI Example:

```
salt '*' runit.full_restart <service name>
```

`salt.modules.runit.`**`get_all`**`()`
Return a list of all available services

CLI Example:

```
salt '*' runit.get_all
```

`salt.modules.runit.`**`get_disabled`**`()`
Return a list of all disabled services

CLI Example:

```
salt '*' service.get_disabled
```

`salt.modules.runit.`**`get_enabled`**`()`
Return a list of all enabled services

CLI Example:

```
salt '*' service.get_enabled
```

`salt.modules.runit.`**`get_svc_alias`**`()`
Returns the list of service's name that are aliased and their alias path(s)

`salt.modules.runit.`**`get_svc_avail_path`**`()`
Return list of paths that may contain available services

`salt.modules.runit.`**`get_svc_broken_path`**`(`*name='*'*`)`
Return list of broken path(s) in SERVICE_DIR that match `name`

A path is broken if it is a broken symlink or can not be a runit service
name a glob for service name. default is '*'
CLI Example:

```
salt '*' runit.get_svc_broken_path <service name>
```

`salt.modules.runit.`**`missing`**`(`*name*`)`
The inverse of runit.available. Returns `True` if the specified service is not available, otherwise returns `False`.
name the service's name
CLI Example:

```
salt '*' runit.missing <service name>
```

`salt.modules.runit.`**`reload_`**`(`*name*`)`
Reload service

name the service's name
CLI Example:

```
salt '*' runit.reload <service name>
```

`salt.modules.runit.`**`remove`**`(`*name*`)`
Remove the service <name> from system. Returns `True` if operation is successful. The service will be also stopped.
name the service's name
CLI Example:

```
salt '*' service.remove <name>
```

`salt.modules.runit.`**`restart`**`(`*name*`)`
Restart service
name the service's name
CLI Example:

```
salt '*' runit.restart <service name>
```

`salt.modules.runit.`**`show`**`(`*name*`)`
Show properties of one or more units/jobs or the manager
name the service's name
CLI Example:
salt '*' service.show <service name>

`salt.modules.runit.`**`start`**`(`*name*`)`
Start service
name the service's name
CLI Example:

```
salt '*' runit.start <service name>
```

`salt.modules.runit.`**`status`**`(`*name, sig=None*`)`
Return `True` if service is running
name the service's name
sig signature to identify with ps
CLI Example:

```
salt '*' runit.status <service name>
```

`salt.modules.runit.`**`status_autostart`**`(`*name*`)`
Return `True` if service <name> is autostarted by sv (file $service_folder/down does not exist) NB: return `False` if the service is not enabled.
name the service's name
CLI Example:

```
salt '*' runit.status_autostart <service name>
```

`salt.modules.runit.`**`stop`**`(`*name*`)`
Stop service
name the service's name
CLI Example:

```
salt '*' runit.stop <service name>
```

21.9.365 salt.modules.rvm

Manage ruby installations and gemsets with RVM, the Ruby Version Manager.

`salt.modules.rvm.`**`do`**(*ruby*, *command*, *runas=None*, *cwd=None*, *env=None*)

Execute a command in an RVM controlled environment.

ruby Which ruby to use

command The rvm command to execute

runas The user under which to run rvm. If not specified, then rvm will be run as the user under which Salt is running.

cwd The directory from which to run the rvm command. Defaults to the user's home directory.

CLI Example:

```
salt '*' rvm.do 2.0.0 <command>
```

`salt.modules.rvm.`**`gemset_copy`**(*source*, *destination*, *runas=None*)

Copy all gems from one gemset to another.

source The name of the gemset to copy, complete with ruby version

destination The destination gemset

runas The user under which to run rvm. If not specified, then rvm will be run as the user under which Salt is running.

CLI Example:

```
salt '*' rvm.gemset_copy foobar bazquo
```

`salt.modules.rvm.`**`gemset_create`**(*ruby*, *gemset*, *runas=None*)

Creates a gemset.

ruby The ruby version for which to create the gemset

gemset The name of the gemset to create

runas The user under which to run rvm. If not specified, then rvm will be run as the user under which Salt is running.

CLI Example:

```
salt '*' rvm.gemset_create 2.0.0 foobar
```

`salt.modules.rvm.`**`gemset_delete`**(*ruby*, *gemset*, *runas=None*)

Delete a gemset

ruby The ruby version to which the gemset belongs

gemset The gemset to delete

runas The user under which to run rvm. If not specified, then rvm will be run as the user under which Salt is running.

CLI Example:

```
salt '*' rvm.gemset_delete 2.0.0 foobar
```

`salt.modules.rvm.`**`gemset_empty`**(*ruby*, *gemset*, *runas=None*)

Remove all gems from a gemset.

ruby The ruby version to which the gemset belongs

gemset The gemset to empty

runas The user under which to run rvm. If not specified, then rvm will be run as the user under which Salt is running.

CLI Example:

```
salt '*' rvm.gemset_empty 2.0.0 foobar
```

`salt.modules.rvm.`**`gemset_list`**(*ruby='default'*, *runas=None*)

List all gemsets for the given ruby.

ruby [default] The ruby version for which to list the gemsets

runas The user under which to run rvm. If not specified, then rvm will be run as the user under which Salt is running.

CLI Example:

```
salt '*' rvm.gemset_list
```

`salt.modules.rvm.`**`gemset_list_all`**(*runas=None*)

List all gemsets for all installed rubies.

Note that you must have set a default ruby before this can work.

runas The user under which to run rvm. If not specified, then rvm will be run as the user under which Salt is running.

CLI Example:

```
salt '*' rvm.gemset_list_all
```

`salt.modules.rvm.`**`get`**(*version='stable'*, *runas=None*)

Update RVM

version [stable] Which version of RVM to install, (e.g. stable or head)

CLI Example:

```
salt '*' rvm.get
```

`salt.modules.rvm.`**`install`**(*runas=None*)

Install RVM system-wide

runas The user under which to run the rvm installer script. If not specified, then it be run as the user under which Salt is running.

CLI Example:

```
salt '*' rvm.install
```

`salt.modules.rvm.`**`install_ruby`**(*ruby*, *runas=None*, *opts=None*, *env=None*)

Install a ruby implementation.

ruby The version of ruby to install

runas The user under which to run rvm. If not specified, then rvm will be run as the user under which Salt is running.

env Environment to set for the install command. Useful for exporting compilation flags such as RUBY_CONFIGURE_OPTS

opts List of options to pass to the RVM installer (ie -C, –patch, etc)

CLI Example:

```
salt '*' rvm.install_ruby 1.9.3-p385
```

`salt.modules.rvm.`**`is_installed`**(*runas=None*)

Check if RVM is installed.

CLI Example:

```
salt '*' rvm.is_installed
```

`salt.modules.rvm.`**`list_`**(*runas=None*)

List all rvm-installed rubies

runas The user under which to run rvm. If not specified, then rvm will be run as the user under which Salt is running.

CLI Example:

```
salt '*' rvm.list
```

`salt.modules.rvm.`**`reinstall_ruby`**(*ruby*, *runas=None*, *env=None*)

Reinstall a ruby implementation

ruby The version of ruby to reinstall

runas The user under which to run rvm. If not specified, then rvm will be run as the user under which Salt is running.

CLI Example:

```
salt '*' rvm.reinstall_ruby 1.9.3-p385
```

`salt.modules.rvm.`**`rubygems`**(*ruby*, *version*, *runas=None*)

Installs a specific rubygems version in the given ruby

ruby The ruby for which to install rubygems

version The version of rubygems to install, or 'remove' to use the version that ships with 1.9

runas The user under which to run rvm. If not specified, then rvm will be run as the user under which Salt is running.

CLI Example:

```
salt '*' rvm.rubygems 2.0.0 1.8.24
```

`salt.modules.rvm.`**`set_default`**(*ruby*, *runas=None*)

Set the default ruby

ruby The version of ruby to make the default

runas The user under which to run rvm. If not specified, then rvm will be run as the user under which Salt is running.

CLI Example:

```
salt '*' rvm.set_default 2.0.0
```

`salt.modules.rvm.`**`wrapper`**(*ruby_string*, *wrapper_prefix*, *runas=None*, **binaries*)

Install RVM wrapper scripts

ruby_string Ruby/gemset to install wrappers for

wrapper_prefix What to prepend to the name of the generated wrapper binaries

runas The user under which to run rvm. If not specified, then rvm will be run as the user under which Salt is running.

binaries [None] The names of the binaries to create wrappers for. When nothing is given, wrappers for ruby, gem, rake, irb, rdoc, ri and testrb are generated.

CLI Example:

```
salt '*' rvm.wrapper <ruby_string> <wrapper_prefix>
```

21.9.366 salt.modules.s3

Connection module for Amazon S3

configuration This module accepts explicit s3 credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More Information available at:

> http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-ec2.html

If IAM roles are not used you need to specify them either in a pillar or in the minion's config file:

```
s3.keyid: GKTADJGHEIQSXMKKRBJ08H
s3.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

Warning: This is literally the pillar key `s3.keyid` or the config option `s3.keyid`, not:

```
s3:
  keyid: blah
```

A `service_url` may also be specified in the configuration:

```
s3.service_url: s3.amazonaws.com
```

A `role_arn` may also be specified in the configuration:

```
s3.role_arn: arn:aws:iam::111111111111:role/my-role-to-assume
```

If a `service_url` is not specified, the default is `s3.amazonaws.com`. This may appear in various documentation as an ”endpoint”. A comprehensive list for Amazon S3 may be found at:

> http://docs.aws.amazon.com/general/latest/gr/rande.html#s3_region

The `service_url` will form the basis for the final endpoint that is used to query the service.

Path style can be enabled:

```
s3.path_style: True
```

This can be useful if you need to use Salt with a proxy for a S3 compatible storage.

You can use either HTTPS protocol or HTTP protocol:

```
s3.https_enable: True
```

SSL verification may also be turned off in the configuration:

```
s3.verify_ssl: False
```

This is required if using S3 bucket names that contain a period, as these will not match Amazon's S3 wildcard certificates. Certificate verification is enabled by default.

AWS region may be specified in the configuration:

```
s3.location: eu-central-1
```

Default is `us-east-1`.

This module should be usable to query other S3-like services, such as Eucalyptus.

depends requests

`salt.modules.s3.`**`delete`**(*bucket*, *path=None*, *action=None*, *key=None*, *keyid=None*, *service_url=None*, *verify_ssl=None*, *kms_keyid=None*, *location=None*, *role_arn=None*, *path_style=None*, *https_enable=None*)

Delete a bucket, or delete an object from a bucket.

CLI Example to delete a bucket:

```
salt myminion s3.delete mybucket
```

CLI Example to delete an object from a bucket:

```
salt myminion s3.delete mybucket remoteobject
```

`salt.modules.s3.`**`get`**`(`*bucket='', path='', return_bin=False, action=None, local_file=None, key=None, keyid=None, service_url=None, verify_ssl=None, kms_keyid=None, location=None, role_arn=None, path_style=None, https_enable=None*`)`

List the contents of a bucket, or return an object from a bucket. Set return_bin to True in order to retrieve an object wholesale. Otherwise, Salt will attempt to parse an XML response.

CLI Example to list buckets:
    salt myminion s3.get
CLI Example to list the contents of a bucket:
    salt myminion s3.get mybucket
CLI Example to return the binary contents of an object:
    salt myminion s3.get mybucket myfile.png return_bin=True
CLI Example to save the binary contents of an object to a local file:
    salt myminion s3.get mybucket myfile.png local_file=/tmp/myfile.png
It is also possible to perform an action on a bucket. Currently, S3 supports the following actions:

```
acl
cors
lifecycle
policy
location
logging
notification
tagging
versions
requestPayment
versioning
website
```

To perform an action on a bucket:
    salt myminion s3.get mybucket myfile.png action=acl

`salt.modules.s3.`**`head`**`(`*bucket, path='', key=None, keyid=None, service_url=None, verify_ssl=None, kms_keyid=None, location=None, role_arn=None, path_style=None, https_enable=None*`)`

Return the metadata for a bucket, or an object in a bucket.

CLI Examples:
    salt myminion s3.head mybucket salt myminion s3.head mybucket myfile.png

`salt.modules.s3.`**`put`**`(`*bucket, path=None, return_bin=False, action=None, local_file=None, key=None, keyid=None, service_url=None, verify_ssl=None, kms_keyid=None, location=None, role_arn=None, path_style=None, https_enable=None*`)`

Create a new bucket, or upload an object to a bucket.

CLI Example to create a bucket:
    salt myminion s3.put mybucket
CLI Example to upload an object to a bucket:
    salt myminion s3.put mybucket remotepath local_file=/path/to/file

21.9.367 salt.modules.s6 module

s6 service module

This module is compatible with the `service` states, so it can be used to maintain services using the `provider` argument:

```
myservice:
  service:
    - running
    - provider: s6
```

Note that the `enabled` argument is not available with this provider.

codeauthor Marek Skrobacki <skrobul@skrobul.com>

`salt.modules.s6.`**`available`**`(`*name*`)`
Returns `True` if the specified service is available, otherwise returns `False`.

CLI Example:

```
salt '*' s6.available foo
```

`salt.modules.s6.`**`full_restart`**`(`*name*`)`
Calls s6.restart() function

CLI Example:

```
salt '*' s6.full_restart <service name>
```

`salt.modules.s6.`**`get_all`**`()`
Return a list of all available services

CLI Example:

```
salt '*' s6.get_all
```

`salt.modules.s6.`**`missing`**`(`*name*`)`
The inverse of s6.available. Returns `True` if the specified service is not available, otherwise returns `False`.

CLI Example:

```
salt '*' s6.missing foo
```

`salt.modules.s6.`**`reload_`**`(`*name*`)`
Send a HUP to service via s6

CLI Example:

```
salt '*' s6.reload <service name>
```

`salt.modules.s6.`**`restart`**`(`*name*`)`
Restart service via s6. This will stop/start service

CLI Example:

```
salt '*' s6.restart <service name>
```

`salt.modules.s6.`**`start`**`(`*name*`)`
Starts service via s6

CLI Example:

```
salt '*' s6.start <service name>
```

`salt.modules.s6.`**`status`**(*name*, *sig=None*)

Return the status for a service via s6, return pid if running

CLI Example:

```
salt '*' s6.status <service name>
```

`salt.modules.s6.`**`stop`**(*name*)

Stops service via s6

CLI Example:

```
salt '*' s6.stop <service name>
```

`salt.modules.s6.`**`term`**(*name*)

Send a TERM to service via s6

CLI Example:

```
salt '*' s6.term <service name>
```

21.9.368 salt.modules.salt_proxy module

Salt proxy module

New in version 2015.8.3.

Module to deploy and manage salt-proxy processes on a minion.

`salt.modules.salt_proxy.`**`configure_proxy`**(*proxyname*, *start=True*)

Create the salt proxy file and start the proxy process if required

Parameters

- **`proxyname`** – Name to be used for this proxy (should match entries in pillar)
- **`start`** – Boolean indicating if the process should be started default = True

CLI Example:

```
salt deviceminion salt_proxy.configure_proxy p8000
```

`salt.modules.salt_proxy.`**`is_running`**(*proxyname*)

Check if the salt-proxy process associated with this proxy (name) is running.

Returns True if the process is running False otherwise

Parameters **`proxyname`** – String name of the proxy (p8000 for example)

CLI Example:

```
salt deviceminion salt_proxy.is_running p8000
```

21.9.369 salt.modules.salt_version

Access Salt's elemental release code-names.

New in version 3000.

Salt's feature release schedule is based on the Periodic Table, as described in the Version Numbers documentation.

When a feature was added (or removed) in a specific release, it can be difficult to build out future-proof functionality that is dependent on a naming scheme that moves.

For example, a state syntax needs to change to support an option that will be removed in the future, but there are many Minion versions in use across an infrastructure. It would be handy to use some Jinja syntax to check for these instances to perform one state syntax over another.

A simple example might be something like the following:

```
{# a boolean check #}
{% set option_deprecated = salt['salt_version.less_than']("Sodium") %}

{% if option_deprecated %}
  <use old syntax>
{% else %}
  <use new syntax>
{% endif %}
```

`salt.modules.salt_version.`**`equal`**`(`*name*`)`

Returns a boolean (True) if the minion's current version code name matches the named version.

name The release code name to check the version against.

CLI Example:

```
salt '*' salt_version.equal 'Oxygen'
```

`salt.modules.salt_version.`**`get_release_number`**`(`*name*`)`

Returns the release number of a given release code name in a `MAJOR.PATCH` format.

If the release name has not been given an assigned release number, the function returns a string. If the release cannot be found, it returns `None`.

name The release code name for which to find a release number.

CLI Example:

```
salt '*' salt_version.get_release_number 'Oxygen'
```

`salt.modules.salt_version.`**`greater_than`**`(`*name*`)`

Returns a boolean (True) if the minion's current version code name is greater than the named version.

name The release code name to check the version against.

CLI Example:

```
salt '*' salt_version.greater_than 'Sodium'
```

`salt.modules.salt_version.`**`less_than`**`(`*name*`)`

Returns a boolean (True) if the minion's current version code name is less than the named version.

name The release code name to check the version against.

CLI Example:

```
salt '*' salt_version.less_than 'Sodium'
```

21.9.370 salt.modules.saltcheck

A module for testing the logic of states and highstates

codeauthor William Cannon <william.cannon@gmail.com>

maturity new

Saltcheck provides unittest like functionality requiring only the knowledge of salt module execution and yaml. Saltcheck uses salt modules to return data, then runs an assertion against that return. This allows for testing with all the features included in salt modules.

In order to run state and highstate saltcheck tests, a sub-folder in the state directory must be created and named `saltcheck-tests`. Tests for a state should be created in files ending in `*.tst` and placed in the `saltcheck-tests` folder. `tst` files are run through the salt rendering system, enabling tests to be written in yaml (or renderer of choice), and include jinja, as well as the usual grain and pillar information. Like states, multiple tests can be specified in a `tst` file. Multiple `tst` files can be created in the `saltcheck-tests` folder, and should be named the same as the associated state. The `id` of a test works in the same manner as in salt state files and should be unique and descriptive.

New in version 3000: The `saltcheck-tests` folder can be customized using the `saltcheck_test_location` minion configuration setting. This setting is a relative path from the formula's `salt://` path to the test files.

Usage

Example Default file system layout:

```
/srv/salt/apache/
    init.sls
    config.sls
    saltcheck-tests/
        init.tst
        config.tst
        deployment_validation.tst
```

Alternative example file system layout with custom saltcheck_test_location:

Minion configuration:

```
saltcheck_test_location: tests/integration/saltcheck
```

Filesystem layout:

```
/srv/salt/apache/
    init.sls
    config.sls
    tests/integration/saltcheck/
        init.tst
        config.tst
        deployment_validation.tst
```

Tests can be run for each state by name, for all `apache/saltcheck/*.tst` files, or for all states assigned to the minion in top.sls. Tests may also be created with no associated state. These tests will be run through the use of `saltcheck.run_state_tests`, but will not be automatically run by `saltcheck.run_highstate_tests`.

```
salt '*' saltcheck.run_state_tests apache,apache.config
salt '*' saltcheck.run_state_tests apache check_all=True
salt '*' saltcheck.run_highstate_tests
salt '*' saltcheck.run_state_tests apache.deployment_validation
```

Saltcheck Keywords

module_and_function: (str) This is the salt module which will be run locally, the same as `salt-call --local <module>`. The `saltcheck.state_apply` module name is special as it bypasses the local option in order to resolve state names when run in a master/minion environment.

args: (list) Optional arguments passed to the salt module

kwargs: (dict) Optional keyword arguments to be passed to the salt module

assertion: (str) One of the supported assertions and required except for `saltcheck.state_apply` Tests which fail the assertion and expected_return, cause saltcheck to exit which a non-zero exit code.

expected_return: (str) Required except by `assertEmpty`, `assertNotEmpty`, `assertTrue`, `assertFalse`. The return of module_and_function is compared to this value in the assertion.

assertion_section: (str) Optional keyword used to parse the module_and_function return. If a salt module returns a dictionary as a result, the `assertion_section` value is used to lookup a specific value in that return for the assertion comparison.

assertion_section_delimiter: (str) Optional delimiter to use when splitting a nested structure. Defaults to ':'

print_result: (bool) Optional keyword to show results in the `assertEqual`, `assertNotEqual`, `assertIn`, and `assertNotIn` output. Defaults to True.

output_details: (bool) Optional keyword to display `module_and_function`, `args`, `assertion_section`, and assertion results text in the output. If print_result is False, assertion results will be hidden. This is a per test setting, but can be set globally for all tests by adding `saltcheck_output_details: True` in the minion configuration file. Defaults to False

pillar_data: (dict) Optional keyword for passing in pillar data. Intended for use in potential test setup or teardown with the `saltcheck.state_apply` function.

skip: (bool) Optional keyword to skip running the individual test

Sample Cases/Examples

Basic Example

```
echo_test_hello:
  module_and_function: test.echo
  args:
    - "hello"
  kwargs:
  assertion: assertEqual
  expected_return:  'hello'
```

Example with jinja

```
{% for package in ["apache2", "openssh"] %}
{# or another example #}
{# for package in salt['pillar.get']("packages") #}
test_{{ package }}_latest:
  module_and_function: pkg.upgrade_available
  args:
    - {{ package }}
```

(continues on next page)

(continued from previous page)

```
  assertion: assertFalse
{% endfor %}
```

Example with setup state including pillar

```
setup_test_environment:
  module_and_function: saltcheck.state_apply
  args:
    - common
  pillar_data:
    data: value

verify_vim:
  module_and_function: pkg.version
  args:
    - vim
  assertion: assertNotEmpty
```

Example with jinja

```
{% for package in ["apache2", "openssh"] %}
{# or another example #}
{# for package in salt['pillar.get']("packages") #}
test_{{ package }}_latest:
  module_and_function: pkg.upgrade_available
  args:
    - {{ package }}
  assertion: assertFalse
{% endfor %}
```

Example with setup state including pillar

```
setup_test_environment:
  module_and_function: saltcheck.state_apply
  args:
    - common
  pillar-data:
    data: value

verify_vim:
  module_and_function: pkg.version
  args:
    - vim
  assertion: assertNotEmpty
```

Example with skip

```
package_latest:
  module_and_function: pkg.upgrade_available
  args:
    - apache2
  assertion: assertFalse
  skip: True
```

Example with assertion_section

```
validate_shell:
  module_and_function: user.info
  args:
    - root
  assertion: assertEqual
  expected_return: /bin/bash
  assertion_section: shell
```

Example with a nested assertion_section

```
validate_smb_signing:
  module_and_function: lgpo.get
  args:
    - 'Machine'
  kwargs:
    return_full_policy_names: True
  assertion: assertEqual
  expected_return: Enabled
  assertion_section: 'Computer Configuration|Microsoft network client: Digitally sign
↪communications (always)'
  assertion_section_delimiter: '|'
```

Example suppressing print results

```
validate_env_nameNode:
  module_and_function: hadoop.dfs
  args:
    - text
    - /oozie/common/env.properties
  expected_return: nameNode = hdfs://nameservice2
  assertion: assertNotIn
  print_result: False
```

Supported assertions

- assertEqual
- assertNotEqual
- assertTrue
- assertFalse

- assertIn
- assertNotIn
- assertGreater
- assertGreaterEqual
- assertLess
- assertLessEqual
- assertEmpty
- assertNotEmpty

> **Warning:** The saltcheck.state_apply function is an alias for `state.apply`. If using the *ACL system* `saltcheck.*` might provide more capability than intended if only `saltcheck.run_state_tests` and `saltcheck.run_highstate_tests` are needed.

`salt.modules.saltcheck.`**`report_highstate_tests`**`(`*saltenv=None*`)`

Report on tests for states assigned to the minion through highstate. Quits with the exit code for the number of missing tests.

CLI Example:

```
salt '*' saltcheck.report_highstate_tests
```

New in version 3000.

`salt.modules.saltcheck.`**`run_highstate_tests`**`(`*saltenv=None*`)`

Execute all tests for states assigned to the minion through highstate and return results

CLI Example:

```
salt '*' saltcheck.run_highstate_tests
```

`salt.modules.saltcheck.`**`run_state_tests`**`(`*state, saltenv=None, check_all=False*`)`

Execute tests for a salt state and return results Nested states will also be tested

Parameters

- **`state`** (*`str`*) – state name for which to run associated .tst test files
- **`saltenv`** (*`str`*) – optional saltenv. Defaults to base
- **`check_all`** (*`bool`*) – boolean to run all tests in state/saltcheck-tests directory

CLI Example:

```
salt '*' saltcheck.run_state_tests postfix,common
```

`salt.modules.saltcheck.`**`run_state_tests_ssh`**`(`*state, saltenv=None, check_all=False*`)`

This function is an alias of `run_state_tests`.

Execute tests for a salt state and return results Nested states will also be tested

param str state state name for which to run associated .tst test files

param str saltenv optional saltenv. Defaults to base

param bool check_all boolean to run all tests in state/saltcheck-tests directory

CLI Example:

```
salt '*' saltcheck.run_state_tests postfix,common
```

`salt.modules.saltcheck.`**`run_test`**`(`***kwargs*`)`

Execute one saltcheck test and return result

Parameters **arg test** (*keyword*) –

CLI Example:

```
salt '*' saltcheck.run_test
    test='{"module_and_function": "test.echo",
           "assertion": "assertEqual",
           "expected_return": "This works!",
           "args":["This works!"] }'
```

`salt.modules.saltcheck.`**`state_apply`**(*state_name*, ***kwargs*)

Runs `state.apply` with given options to set up test data. Intended to be used for optional test setup or teardown

Reference the `state.apply` module documentation for arguments and usage options

CLI Example:

```
salt '*' saltcheck.state_apply postfix
```

21.9.371 salt.modules.saltcloudmod

Control a salt cloud system

`salt.modules.saltcloudmod.`**`create`**(*name*, *profile*)

Create the named vm

CLI Example:

```
salt <minion-id> saltcloud.create webserver rackspace_centos_512
```

21.9.372 salt.modules.saltutil

The Saltutil module is used to manage the state of the salt minion itself. It is used to manage minion modules as well as automate updates to the salt minion.

depends

- esky Python module for update functionality

`salt.modules.saltutil.`**`clear_cache`**()

Forcibly removes all caches on a minion.

New in version 2014.7.0.

WARNING: The safest way to clear a minion cache is by first stopping the minion and then deleting the cache files before restarting it.

CLI Example:

```
salt '*' saltutil.clear_cache
```

`salt.modules.saltutil.`**`clear_job_cache`**(*hours=24*)

Forcibly removes job cache folders and files on a minion.

New in version 2018.3.0.

WARNING: The safest way to clear a minion cache is by first stopping the minion and then deleting the cache files before restarting it.

CLI Example:

```
salt '*' saltutil.clear_job_cache hours=12
```

`salt.modules.saltutil.`**`cmd`**(*tgt, fun, arg=(), timeout=None, tgt_type='glob', ret='', kwarg=None, ssh=False, **kwargs*)

Changed in version 2017.7.0: The `expr_form` argument has been renamed to `tgt_type`, earlier releases must use `expr_form`.

Assuming this minion is a master, execute a salt command

CLI Example:

```
salt '*' saltutil.cmd
```

`salt.modules.saltutil.`**`cmd_iter`**(*tgt, fun, arg=(), timeout=None, tgt_type='glob', ret='', kwarg=None, ssh=False, **kwargs*)

Changed in version 2017.7.0: The `expr_form` argument has been renamed to `tgt_type`, earlier releases must use `expr_form`.

Assuming this minion is a master, execute a salt command

CLI Example:

```
salt '*' saltutil.cmd_iter
```

`salt.modules.saltutil.`**`find_cached_job`**(*jid*)

Return the data for a specific cached job id. Note this only works if cache_jobs has previously been set to True on the minion.

CLI Example:

```
salt '*' saltutil.find_cached_job <job id>
```

`salt.modules.saltutil.`**`find_job`**(*jid*)

Return the data for a specific job id that is currently running.

jid The job id to search for and return data.

CLI Example:

```
salt '*' saltutil.find_job <job id>
```

Note that the find_job function only returns job information when the job is still running. If the job is currently running, the output looks something like this:

```
# salt my-minion saltutil.find_job 20160503150049487736
my-minion:
    ----------
    arg:
        - 30
    fun:
        test.sleep
    jid:
        20160503150049487736
    pid:
        9601
    ret:
    tgt:
        my-minion
    tgt_type:
```

(continues on next page)

(continued from previous page)

```
        glob
    user:
        root
```

If the job has already completed, the job cannot be found and therefore the function returns an empty dictionary, which looks like this on the CLI:

```
# salt my-minion saltutil.find_job 20160503150049487736
my-minion:
    ----------
```

`salt.modules.saltutil.`**`is_running`**`(`*fun*`)`

If the named function is running return the data associated with it/them. The argument can be a glob

CLI Example:

```
salt '*' saltutil.is_running state.highstate
```

`salt.modules.saltutil.`**`kill_all_jobs`**`()`

Sends a kill signal (SIGKILL 9) to all currently running jobs

CLI Example:

```
salt '*' saltutil.kill_all_jobs
```

`salt.modules.saltutil.`**`kill_job`**`(`*jid*`)`

Sends a kill signal (SIGKILL 9) to the named salt job's process

CLI Example:

```
salt '*' saltutil.kill_job <job id>
```

`salt.modules.saltutil.`**`list_extmods`**`()`

New in version 2017.7.0.

List Salt modules which have been synced externally

CLI Examples:

```
salt '*' saltutil.list_extmods
```

`salt.modules.saltutil.`**`mmodule`**`(`*saltenv, fun, *args, **kwargs*`)`

Loads minion modules from an environment so that they can be used in pillars for that environment

CLI Example:

```
salt '*' saltutil.mmodule base test.ping
```

`salt.modules.saltutil.`**`refresh_beacons`**`()`

Signal the minion to refresh the beacons.

CLI Example:

```
salt '*' saltutil.refresh_beacons
```

`salt.modules.saltutil.`**`refresh_grains`**`(`***kwargs*`)`

New in version 2016.3.6,2016.11.4,2017.7.0.

Refresh the minion's grains without syncing custom grains modules from `salt://_grains`.

Note: The available execution modules will be reloaded as part of this proceess, as grains can affect which modules are available.

---

refresh_pillar [True] Set to `False` to keep pillar data from being refreshed.

CLI Examples:

```
salt '*' saltutil.refresh_grains
```

`salt.modules.saltutil.`**`refresh_matchers`**`()`

Signal the minion to refresh its matchers.

CLI Example:

```
salt '*' saltutil.refresh_matchers
```

`salt.modules.saltutil.`**`refresh_modules`**`(`***kwargs*`)`

Signal the minion to refresh the module and grain data

The default is to refresh module asynchronously. To block until the module refresh is complete, set the 'async' flag to False.

CLI Example:

```
salt '*' saltutil.refresh_modules
```

`salt.modules.saltutil.`**`refresh_pillar`**`(`*wait=False, timeout=30*`)`

Signal the minion to refresh the pillar data.

Parameters

- **wait** (*bool, optional*) – Wait for pillar refresh to complete, defaults to False.
- **timeout** (*int, optional*) – How long to wait in seconds, only used when wait is True, defaults to 30.

Returns Boolean status, True when the pillar_refresh event was fired successfully.

CLI Example:

```
salt '*' saltutil.refresh_pillar
```

`salt.modules.saltutil.`**`regen_keys`**`()`

Used to regenerate the minion keys.

CLI Example:

```
salt '*' saltutil.regen_keys
```

`salt.modules.saltutil.`**`revoke_auth`**`(`*preserve_minion_cache=False*`)`

The minion sends a request to the master to revoke its own key. Note that the minion session will be revoked and the minion may not be able to return the result of this command back to the master.

If the 'preserve_minion_cache' flag is set to True, the master cache for this minion will not be removed.

CLI Example:

```
salt '*' saltutil.revoke_auth
```

`salt.modules.saltutil.`**`runner`**`(`*name, arg=None, kwarg=None, full_return=False, saltenv='base', jid=None, **kwargs*`)`

Execute a runner function. This function must be run on the master, either by targeting a minion running on a master or by using salt-call on a master.

New in version 2014.7.0.

name The name of the function to run

kwargs Any keyword arguments to pass to the runner function

CLI Example:

In this example, assume that *master_minion* is a minion running on a master.

```
salt master_minion saltutil.runner jobs.list_jobs
salt master_minion saltutil.runner test.arg arg="['baz']" kwarg="{'foo': 'bar'}"
```

`salt.modules.saltutil.`**`running`**`()`

Return the data on all running salt processes on the minion

CLI Example:

```
salt '*' saltutil.running
```

`salt.modules.saltutil.`**`signal_job`**`(`*jid*, *sig*`)`

Sends a signal to the named salt job's process

CLI Example:

```
salt '*' saltutil.signal_job <job id> 15
```

`salt.modules.saltutil.`**`sync_all`**`(`*saltenv=None*, *refresh=True*, *extmod_whitelist=None*, *extmod_blacklist=None*`)`

Changed in version 2015.8.11,2016.3.2: On masterless minions, pillar modules are now synced, and refreshed when `refresh` is set to `True`.

Sync down all of the dynamic modules from the file server for a specific environment. This function synchronizes custom modules, states, beacons, grains, returners, output modules, renderers, and utils.

refresh [True] Also refresh the execution modules and recompile pillar data available to the minion. This refresh will be performed even if no new dynamic modules are synced. Set to `False` to prevent this refresh.

Important: If this function is executed using a *`module.run`* state, the SLS file will not have access to newly synced execution modules unless a `refresh` argument is added to the state, like so:

```
load_my_custom_module:
  module.run:
    - name: saltutil.sync_all
    - refresh: True
```

See *here* for a more detailed explanation of why this is necessary.

---

extmod_whitelist [None] dictionary of modules to sync based on type

extmod_blacklist [None] dictionary of modules to blacklist based on type

CLI Examples:

```
salt '*' saltutil.sync_all
salt '*' saltutil.sync_all saltenv=dev
salt '*' saltutil.sync_all saltenv=base,dev
salt '*' saltutil.sync_all extmod_whitelist={'modules': ['custom_module']}
```

`salt.modules.saltutil.`**`sync_beacons`**`(`*saltenv=None*, *refresh=True*, *extmod_whitelist=None*, *extmod_blacklist=None*`)`

New in version 2015.5.1.

Sync beacons from `salt://_beacons` to the minion

saltenv The fileserver environment from which to sync. To sync from more than one environment, pass a comma-separated list.

If not passed, then all environments configured in the *top files* will be checked for beacons to sync. If no top files are found, then the `base` environment will be synced.

refresh [True] If `True`, refresh the available beacons on the minion. This refresh will be performed even if no new beacons are synced. Set to `False` to prevent this refresh.

extmod_whitelist [None] comma-seperated list of modules to sync

extmod_blacklist [None] comma-seperated list of modules to blacklist based on type

CLI Example:

```
salt '*' saltutil.sync_beacons
salt '*' saltutil.sync_beacons saltenv=dev
salt '*' saltutil.sync_beacons saltenv=base,dev
```

`salt.modules.saltutil.`**`sync_clouds`**`(`*saltenv=None, refresh=True, extmod_whitelist=None, extmod_blacklist=None*`)`

New in version 2017.7.0.

Sync cloud modules from `salt://_cloud` to the minion

saltenv [base] The fileserver environment from which to sync. To sync from more than one environment, pass a comma-separated list.

refresh [True] If `True`, refresh the available execution modules on the minion. This refresh will be performed even if no new utility modules are synced. Set to `False` to prevent this refresh.

extmod_whitelist [None] comma-seperated list of modules to sync

extmod_blacklist [None] comma-seperated list of modules to blacklist based on type

CLI Examples:

```
salt '*' saltutil.sync_clouds
salt '*' saltutil.sync_clouds saltenv=dev
salt '*' saltutil.sync_clouds saltenv=base,dev
```

`salt.modules.saltutil.`**`sync_engines`**`(`*saltenv=None, refresh=False, extmod_whitelist=None, extmod_blacklist=None*`)`

New in version 2016.3.0.

Sync engine modules from `salt://_engines` to the minion

saltenv The fileserver environment from which to sync. To sync from more than one environment, pass a comma-separated list.

If not passed, then all environments configured in the *top files* will be checked for engines to sync. If no top files are found, then the `base` environment will be synced.

refresh [True] If `True`, refresh the available execution modules on the minion. This refresh will be performed even if no new engine modules are synced. Set to `False` to prevent this refresh.

extmod_whitelist [None] comma-seperated list of modules to sync

extmod_blacklist [None] comma-seperated list of modules to blacklist based on type

CLI Examples:

```
salt '*' saltutil.sync_engines
salt '*' saltutil.sync_engines saltenv=base,dev
```

`salt.modules.saltutil.`**`sync_executors`**`(`*saltenv=None, refresh=True, extmod_whitelist=None, extmod_blacklist=None*`)`

New in version 3000.

Sync executors from `salt://_executors` to the minion

saltenv The fileserver environment from which to sync. To sync from more than one environment, pass a comma-separated list.

If not passed, then all environments configured in the *top files* will be checked for log handlers to sync. If no top files are found, then the `base` environment will be synced.

refresh [True] If `True`, refresh the available execution modules on the minion. This refresh will be performed even if no new log handlers are synced. Set to `False` to prevent this refresh.

extmod_whitelist [None] comma-seperated list of modules to sync

extmod_blacklist [None] comma-seperated list of modules to blacklist based on type

CLI Examples:

```
salt '*' saltutil.sync_executors
salt '*' saltutil.sync_executors saltenv=dev
salt '*' saltutil.sync_executors saltenv=base,dev
```

`salt.modules.saltutil.`**`sync_grains`**(*saltenv=None, refresh=True, extmod_whitelist=None, extmod_blacklist=None*)

New in version 0.10.0.

Sync grains modules from `salt://_grains` to the minion

saltenv The fileserver environment from which to sync. To sync from more than one environment, pass a comma-separated list.

If not passed, then all environments configured in the *top files* will be checked for grains modules to sync. If no top files are found, then the `base` environment will be synced.

refresh [True] If `True`, refresh the available execution modules and recompile pillar data for the minion. This refresh will be performed even if no new grains modules are synced. Set to `False` to prevent this refresh.

extmod_whitelist [None] comma-seperated list of modules to sync

extmod_blacklist [None] comma-seperated list of modules to blacklist based on type

CLI Examples:

```
salt '*' saltutil.sync_grains
salt '*' saltutil.sync_grains saltenv=dev
salt '*' saltutil.sync_grains saltenv=base,dev
```

`salt.modules.saltutil.`**`sync_log_handlers`**(*saltenv=None, refresh=True, extmod_whitelist=None, extmod_blacklist=None*)

New in version 2015.8.0.

Sync log handlers from `salt://_log_handlers` to the minion

saltenv The fileserver environment from which to sync. To sync from more than one environment, pass a comma-separated list.

If not passed, then all environments configured in the *top files* will be checked for log handlers to sync. If no top files are found, then the `base` environment will be synced.

refresh [True] If `True`, refresh the available execution modules on the minion. This refresh will be performed even if no new log handlers are synced. Set to `False` to prevent this refresh.

extmod_whitelist [None] comma-seperated list of modules to sync

extmod_blacklist [None] comma-seperated list of modules to blacklist based on type

CLI Examples:

```
salt '*' saltutil.sync_log_handlers
salt '*' saltutil.sync_log_handlers saltenv=dev
salt '*' saltutil.sync_log_handlers saltenv=base,dev
```

`salt.modules.saltutil.`**`sync_matchers`**(*saltenv=None, refresh=False, extmod_whitelist=None, extmod_blacklist=None*)

New in version 2019.2.0.

Sync engine modules from `salt://_matchers` to the minion

saltenv The fileserver environment from which to sync. To sync from more than one environment, pass a comma-separated list.

If not passed, then all environments configured in the *top files* will be checked for engines to sync. If no top files are found, then the `base` environment will be synced.

refresh [True] If `True`, refresh the available execution modules on the minion. This refresh will be performed even if no new matcher modules are synced. Set to `False` to prevent this refresh.

extmod_whitelist [None] comma-separated list of modules to sync

extmod_blacklist [None] comma-separated list of modules to blacklist based on type

CLI Examples:

```
salt '*' saltutil.sync_matchers
salt '*' saltutil.sync_matchers saltenv=base,dev
```

`salt.modules.saltutil.`**`sync_modules`**(*saltenv=None, refresh=True, extmod_whitelist=None, extmod_blacklist=None*)

New in version 0.10.0.

Sync execution modules from `salt://_modules` to the minion

saltenv The fileserver environment from which to sync. To sync from more than one environment, pass a comma-separated list.

If not passed, then all environments configured in the *top files* will be checked for execution modules to sync. If no top files are found, then the `base` environment will be synced.

refresh [True] If `True`, refresh the available execution modules on the minion. This refresh will be performed even if no new execution modules are synced. Set to `False` to prevent this refresh.

Important: If this function is executed using a *`module.run`* state, the SLS file will not have access to newly synced execution modules unless a `refresh` argument is added to the state, like so:

```
load_my_custom_module:
  module.run:
    - name: saltutil.sync_modules
    - refresh: True
```

See *here* for a more detailed explanation of why this is necessary.

---

extmod_whitelist [None] comma-seperated list of modules to sync

extmod_blacklist [None] comma-seperated list of modules to blacklist based on type

CLI Example:

```
salt '*' saltutil.sync_modules
salt '*' saltutil.sync_modules saltenv=dev
salt '*' saltutil.sync_modules saltenv=base,dev
```

`salt.modules.saltutil.`**`sync_output`**(*saltenv=None, refresh=True, extmod_whitelist=None, extmod_blacklist=None*)

Sync outputters from `salt://_output` to the minion

saltenv The fileserver environment from which to sync. To sync from more than one environment, pass a comma-separated list.

If not passed, then all environments configured in the *top files* will be checked for outputters to sync. If no top files are found, then the `base` environment will be synced.

refresh [True] If `True`, refresh the available execution modules on the minion. This refresh will be performed even if no new outputters are synced. Set to `False` to prevent this refresh.

extmod_whitelist [None] comma-seperated list of modules to sync

extmod_blacklist [None] comma-seperated list of modules to blacklist based on type

CLI Examples:

```
salt '*' saltutil.sync_output
salt '*' saltutil.sync_output saltenv=dev
salt '*' saltutil.sync_output saltenv=base,dev
```

`salt.modules.saltutil.`**`sync_outputters`**`(`*saltenv=None, refresh=True, extmod_whitelist=None, extmod_blacklist=None*`)`

This function is an alias of `sync_output`.

Sync outputters from `salt://_output` to the minion

saltenv The fileserver environment from which to sync. To sync from more than one environment, pass a comma-separated list.

If not passed, then all environments configured in the *top files* will be checked for outputters to sync. If no top files are found, then the `base` environment will be synced.

refresh [True] If `True`, refresh the available execution modules on the minion. This refresh will be performed even if no new outputters are synced. Set to `False` to prevent this refresh.

extmod_whitelist [None] comma-seperated list of modules to sync

extmod_blacklist [None] comma-seperated list of modules to blacklist based on type

CLI Examples:

```
salt '*' saltutil.sync_output
salt '*' saltutil.sync_output saltenv=dev
salt '*' saltutil.sync_output saltenv=base,dev
```

`salt.modules.saltutil.`**`sync_pillar`**`(`*saltenv=None, refresh=True, extmod_whitelist=None, extmod_blacklist=None*`)`

New in version 2015.8.11,2016.3.2.

Sync pillar modules from the `salt://_pillar` directory on the Salt fileserver. This function is environment-aware, pass the desired environment to grab the contents of the `_pillar` directory from that environment. The default environment, if none is specified, is `base`.

refresh [True] Also refresh the execution modules available to the minion, and refresh pillar data.

extmod_whitelist [None] comma-seperated list of modules to sync

extmod_blacklist [None] comma-seperated list of modules to blacklist based on type

Note: This function will raise an error if executed on a traditional (i.e. not masterless) minion

CLI Examples:

```
salt '*' saltutil.sync_pillar
salt '*' saltutil.sync_pillar saltenv=dev
```

`salt.modules.saltutil.`**`sync_proxymodules`**`(`*saltenv=None, refresh=False, extmod_whitelist=None, extmod_blacklist=None*`)`

New in version 2015.8.2.

Sync proxy modules from `salt://_proxy` to the minion

saltenv The fileserver environment from which to sync. To sync from more than one environment, pass a comma-separated list.

If not passed, then all environments configured in the *top files* will be checked for proxy modules to sync. If no top files are found, then the `base` environment will be synced.

refresh [True] If `True`, refresh the available execution modules on the minion. This refresh will be performed even if no new proxy modules are synced. Set to `False` to prevent this refresh.

extmod_whitelist [None] comma-seperated list of modules to sync

extmod_blacklist [None] comma-seperated list of modules to blacklist based on type

CLI Examples:

```
salt '*' saltutil.sync_proxymodules
salt '*' saltutil.sync_proxymodules saltenv=dev
salt '*' saltutil.sync_proxymodules saltenv=base,dev
```

`salt.modules.saltutil.`**`sync_renderers`**`(`*saltenv=None, refresh=True, extmod_whitelist=None, extmod_blacklist=None*`)`

New in version 0.10.0.

Sync renderers from `salt://_renderers` to the minion

saltenv The fileserver environment from which to sync. To sync from more than one environment, pass a comma-separated list.

If not passed, then all environments configured in the *top files* will be checked for renderers to sync. If no top files are found, then the `base` environment will be synced.

refresh [True] If `True`, refresh the available execution modules on the minion. This refresh will be performed even if no new renderers are synced. Set to `False` to prevent this refresh. Set to `False` to prevent this refresh.

extmod_whitelist [None] comma-seperated list of modules to sync

extmod_blacklist [None] comma-seperated list of modules to blacklist based on type

CLI Examples:

```
salt '*' saltutil.sync_renderers
salt '*' saltutil.sync_renderers saltenv=dev
salt '*' saltutil.sync_renderers saltenv=base,dev
```

`salt.modules.saltutil.`**`sync_returners`**`(`*saltenv=None, refresh=True, extmod_whitelist=None, extmod_blacklist=None*`)`

New in version 0.10.0.

Sync returners from `salt://_returners` to the minion

saltenv The fileserver environment from which to sync. To sync from more than one environment, pass a comma-separated list.

If not passed, then all environments configured in the *top files* will be checked for returners to sync. If no top files are found, then the `base` environment will be synced.

refresh [True] If `True`, refresh the available execution modules on the minion. This refresh will be performed even if no new returners are synced. Set to `False` to prevent this refresh.

extmod_whitelist [None] comma-seperated list of modules to sync

extmod_blacklist [None] comma-seperated list of modules to blacklist based on type

CLI Examples:

```
salt '*' saltutil.sync_returners
salt '*' saltutil.sync_returners saltenv=dev
```

`salt.modules.saltutil.`**`sync_sdb`**`(`*saltenv=None, extmod_whitelist=None, extmod_blacklist=None*`)`

New in version 2015.5.8,2015.8.3.

Sync sdb modules from `salt://_sdb` to the minion

saltenv The fileserver environment from which to sync. To sync from more than one environment, pass a comma-separated list.

If not passed, then all environments configured in the *top files* will be checked for sdb modules to sync. If no top files are found, then the `base` environment will be synced.

refresh [False] This argument has no affect and is included for consistency with the other sync functions.

extmod_whitelist [None] comma-seperated list of modules to sync

extmod_blacklist [None] comma-seperated list of modules to blacklist based on type

CLI Example:

```
salt '*' saltutil.sync_sdb
salt '*' saltutil.sync_sdb saltenv=dev
salt '*' saltutil.sync_sdb saltenv=base,dev
```

`salt.modules.saltutil.`**`sync_serializers`**(*saltenv=None*, *refresh=True*, *extmod_whitelist=None*, *extmod_blacklist=None*)

New in version 2019.2.0.

Sync serializers from `salt://_serializers` to the minion

saltenv The fileserver environment from which to sync. To sync from more than one environment, pass a comma-separated list.

If not passed, then all environments configured in the *top files* will be checked for serializer modules to sync. If no top files are found, then the `base` environment will be synced.

refresh [True] If `True`, refresh the available execution modules on the minion. This refresh will be performed even if no new serializer modules are synced. Set to `False` to prevent this refresh.

extmod_whitelist [None] comma-seperated list of modules to sync

extmod_blacklist [None] comma-seperated list of modules to blacklist based on type

CLI Examples:

```
salt '*' saltutil.sync_serializers
salt '*' saltutil.sync_serializers saltenv=dev
salt '*' saltutil.sync_serializers saltenv=base,dev
```

`salt.modules.saltutil.`**`sync_states`**(*saltenv=None*, *refresh=True*, *extmod_whitelist=None*, *extmod_blacklist=None*)

New in version 0.10.0.

Sync state modules from `salt://_states` to the minion

saltenv The fileserver environment from which to sync. To sync from more than one environment, pass a comma-separated list.

If not passed, then all environments configured in the *top files* will be checked for state modules to sync. If no top files are found, then the `base` environment will be synced.

refresh [True] If `True`, refresh the available states on the minion. This refresh will be performed even if no new state modules are synced. Set to `False` to prevent this refresh.

extmod_whitelist [None] comma-seperated list of modules to sync

extmod_blacklist [None] comma-seperated list of modules to blacklist based on type

CLI Examples:

```
salt '*' saltutil.sync_states
salt '*' saltutil.sync_states saltenv=dev
salt '*' saltutil.sync_states saltenv=base,dev
```

`salt.modules.saltutil.`**`sync_thorium`**(*saltenv=None*, *refresh=False*, *extmod_whitelist=None*, *extmod_blacklist=None*)

New in version 2018.3.0.

Sync Thorium modules from `salt://_thorium` to the minion

saltenv The fileserver environment from which to sync. To sync from more than one environment, pass a comma-separated list.

If not passed, then all environments configured in the *top files* will be checked for engines to sync. If no top files are found, then the `base` environment will be synced.

refresh: **`True`** If `True`, refresh the available execution modules on the minion. This refresh will be performed even if no new Thorium modules are synced. Set to `False` to prevent this refresh.

extmod_whitelist comma-seperated list of modules to sync

extmod_blacklist comma-seperated list of modules to blacklist based on type

CLI Examples:

```
salt '*' saltutil.sync_thorium
salt '*' saltutil.sync_thorium saltenv=base,dev
```

`salt.modules.saltutil.`**`sync_utils`**(*saltenv=None*, *refresh=True*, *extmod_whitelist=None*, *extmod_blacklist=None*)

New in version 2014.7.0.

Sync utility modules from `salt://_utils` to the minion

saltenv The fileserver environment from which to sync. To sync from more than one environment, pass a comma-separated list.

If not passed, then all environments configured in the *top files* will be checked for utility modules to sync. If no top files are found, then the `base` environment will be synced.

refresh [True] If `True`, refresh the available execution modules on the minion. This refresh will be performed even if no new utility modules are synced. Set to `False` to prevent this refresh.

extmod_whitelist [None] comma-seperated list of modules to sync

extmod_blacklist [None] comma-seperated list of modules to blacklist based on type

CLI Examples:

```
salt '*' saltutil.sync_utils
salt '*' saltutil.sync_utils saltenv=dev
salt '*' saltutil.sync_utils saltenv=base,dev
```

`salt.modules.saltutil.`**`term_all_jobs`**()

Sends a termination signal (SIGTERM 15) to all currently running jobs

CLI Example:

```
salt '*' saltutil.term_all_jobs
```

`salt.modules.saltutil.`**`term_job`**(*jid*)

Sends a termination signal (SIGTERM 15) to the named salt job's process

CLI Example:

```
salt '*' saltutil.term_job <job id>
```

`salt.modules.saltutil.`**`update`**(*version=None*)

Update the salt minion from the URL defined in opts['update_url'] SaltStack, Inc provides the latest builds here: update_url: https://repo.saltstack.com/windows/

Be aware that as of 2014-8-11 there's a bug in esky such that only the latest version available in the update_url can be downloaded and installed.

This feature requires the minion to be running a bdist_esky build.

The version number is optional and will default to the most recent version available at opts['update_url'].

Returns details about the transaction upon completion.

CLI Examples:

```
salt '*' saltutil.update
salt '*' saltutil.update 0.10.3
```

`salt.modules.saltutil.`**`wheel`**(*name*, **args*, ***kwargs*)

Execute a wheel module and function. This function must be run against a minion that is local to the master.

New in version 2014.7.0.

name The name of the function to run

args Any positional arguments to pass to the wheel function. A common example of this would be the `match` arg needed for key functions.

New in version v2015.8.11.

kwargs Any keyword arguments to pass to the wheel function

CLI Example:

```
salt my-local-minion saltutil.wheel key.accept jerry
salt my-local-minion saltutil.wheel minions.connected
```

Note: Since this function must be run against a minion that is running locally on the master in order to get accurate returns, if this function is run against minions that are not local to the master, "empty" returns are expected. The remote minion does not have access to wheel functions and their return data.

21.9.373 salt.modules.schedule

Module for managing the Salt schedule on a minion

New in version 2014.7.0.

`salt.modules.schedule.`**`add`**(*name*, ***kwargs*)

Add a job to the schedule

CLI Example:

```
salt '*' schedule.add job1 function='test.ping' seconds=3600
# If function have some arguments, use job_args
salt '*' schedule.add job2 function='cmd.run' job_args="['date >> /tmp/date.log']
↪" seconds=60
```

`salt.modules.schedule.`**`build_schedule_item`**(*name*, ***kwargs*)

Build a schedule job

CLI Example:

```
salt '*' schedule.build_schedule_item job1 function='test.ping' seconds=3600
```

`salt.modules.schedule.`**`copy`**(*name*, *target*, ***kwargs*)

Copy scheduled job to another minion or minions.

CLI Example:

```
salt '*' schedule.copy jobname target
```

`salt.modules.schedule.`**`delete`**(*name*, ***kwargs*)

Delete a job from the minion's schedule

CLI Example:

```
salt '*' schedule.delete job1
```

`salt.modules.schedule.`**`disable`**(***kwargs*)

Disable all scheduled jobs on the minion

CLI Example:

```
salt '*' schedule.disable
```

salt.modules.schedule.**disable_job**(*name*, ***kwargs*)

Disable a job in the minion's schedule

CLI Example:

```
salt '*' schedule.disable_job job1
```

salt.modules.schedule.**enable**(***kwargs*)

Enable all scheduled jobs on the minion

CLI Example:

```
salt '*' schedule.enable
```

salt.modules.schedule.**enable_job**(*name*, ***kwargs*)

Enable a job in the minion's schedule

CLI Example:

```
salt '*' schedule.enable_job job1
```

salt.modules.schedule.**is_enabled**(*name*)

List a Job only if its enabled

New in version 2015.5.3.

CLI Example:

```
salt '*' schedule.is_enabled name=job_name
```

salt.modules.schedule.**list_**(*show_all=False*, *show_disabled=True*, *where=None*, *return_yaml=True*)

List the jobs currently scheduled on the minion

CLI Example:

```
salt '*' schedule.list

# Show all jobs including hidden internal jobs
salt '*' schedule.list show_all=True

# Hide disabled jobs from list of jobs
salt '*' schedule.list show_disabled=False
```

salt.modules.schedule.**modify**(*name*, ***kwargs*)

Modify an existing job in the schedule

CLI Example:

```
salt '*' schedule.modify job1 function='test.ping' seconds=3600
```

salt.modules.schedule.**move**(*name*, *target*, ***kwargs*)

Move scheduled job to another minion or minions.

CLI Example:

```
salt '*' schedule.move jobname target
```

`salt.modules.schedule.`**`postpone_job`**`(`*name, current_time, new_time, **kwargs*`)`

Postpone a job in the minion's schedule

Current time and new time should be in date string format, default value is %Y-%m-%dT%H:%M:%S.

New in version 2018.3.0.

CLI Example:

```
salt '*' schedule.postpone_job job current_time new_time

salt '*' schedule.postpone_job job current_time new_time time_fmt='%Y-%m-%dT%H:%M:
↪%S'
```

`salt.modules.schedule.`**`purge`**`(`***kwargs*`)`

Purge all the jobs currently scheduled on the minion

CLI Example:

```
salt '*' schedule.purge
```

`salt.modules.schedule.`**`reload_`**`()`

Reload saved scheduled jobs on the minion

CLI Example:

```
salt '*' schedule.reload
```

`salt.modules.schedule.`**`run_job`**`(`*name, force=False*`)`

Run a scheduled job on the minion immediately

CLI Example:

```
salt '*' schedule.run_job job1

salt '*' schedule.run_job job1 force=True
Force the job to run even if it is disabled.
```

`salt.modules.schedule.`**`save`**`(`***kwargs*`)`

Save all scheduled jobs on the minion

CLI Example:

```
salt '*' schedule.save
```

`salt.modules.schedule.`**`show_next_fire_time`**`(`*name, **kwargs*`)`

Show the next fire time for scheduled job

New in version 2018.3.0.

CLI Example:

```
salt '*' schedule.show_next_fire_time job_name
```

`salt.modules.schedule.`**`skip_job`**`(`*name, current_time, **kwargs*`)`

Skip a job in the minion's schedule at specified time.

Time to skip should be specified as date string format, default value is %Y-%m-%dT%H:%M:%S.

New in version 2018.3.0.

CLI Example:

```
salt '*' schedule.skip_job job time
```

21.9.374 salt.modules.scp module

SCP Module

New in version 2019.2.0.

Module to copy files via SCP

`salt.modules.scp_mod.`**`get`**`(`*remote_path, local_path='', recursive=False, preserve_times=False, **kwargs*`)`

Transfer files and directories from remote host to the localhost of the Minion.

remote_path Path to retrieve from remote host. Since this is evaluated by scp on the remote host, shell wildcards and environment variables may be used.

recursive: `False` Transfer files and directories recursively.

preserve_times: `False` Preserve `mtime` and `atime` of transferred files and directories.

hostname The hostname of the remote device.

port: `22` The port of the remote device.

username The username required for SSH authentication on the device.

password Used for password authentication. It is also used for private key decryption if `passphrase` is not given.

passphrase Used for decrypting private keys.

pkey An optional private key to use for authentication.

key_filename The filename, or list of filenames, of optional private key(s) and/or certificates to try for authentication.

timeout An optional timeout (in seconds) for the TCP connect.

socket_timeout: `10` The channel socket timeout in seconds.

buff_size: `16384` The size of the SCP send buffer.

allow_agent: `True` Set to `False` to disable connecting to the SSH agent.

look_for_keys: `True` Set to `False` to disable searching for discoverable private key files in `~/.ssh/`

banner_timeout An optional timeout (in seconds) to wait for the SSH banner to be presented.

auth_timeout An optional timeout (in seconds) to wait for an authentication response.

auto_add_policy: `False` Automatically add the host to the `known_hosts`.

CLI Example:

```
salt '*' scp.get /var/tmp/file /tmp/file hostname=10.10.10.1 auto_add_policy=True
```

`salt.modules.scp_mod.`**`put`**`(`*files, remote_path=None, recursive=False, preserve_times=False, saltenv='base', **kwargs*`)`

Transfer files and directories to remote host.

files A single path or a list of paths to be transferred.

remote_path The path on the remote device where to store the files.

recursive: `True` Transfer files and directories recursively.

preserve_times: `False` Preserve `mtime` and `atime` of transferred files and directories.

hostname The hostname of the remote device.

port: `22` The port of the remote device.

username The username required for SSH authentication on the device.

password Used for password authentication. It is also used for private key decryption if `passphrase` is not given.

passphrase Used for decrypting private keys.

pkey An optional private key to use for authentication.

key_filename The filename, or list of filenames, of optional private key(s) and/or certificates to try for authentication.

timeout An optional timeout (in seconds) for the TCP connect.
socket_timeout: **`10`** The channel socket timeout in seconds.
buff_size: **`16384`** The size of the SCP send buffer.
allow_agent: **`True`** Set to `False` to disable connecting to the SSH agent.
look_for_keys: **`True`** Set to `False` to disable searching for discoverable private key files in `~/.ssh/`
banner_timeout An optional timeout (in seconds) to wait for the SSH banner to be presented.
auth_timeout An optional timeout (in seconds) to wait for an authentication response.
auto_add_policy: **`False`** Automatically add the host to the `known_hosts`.
CLI Example:

```
salt '*' scp.put /path/to/file /var/tmp/file hostname=server1 auto_add_policy=True
```

21.9.375 salt.modules.scsi

SCSI administration module

`salt.modules.scsi.`**`ls_`**(*get_size=True*)
List SCSI devices, with details

CLI Examples:

```
salt '*' scsi.ls
salt '*' scsi.ls get_size=False
```

get_size [True] Get the size information for scsi devices. This option should be set to False for older OS distributions (RHEL6 and older) due to lack of support for the '-s' option in lsscsi.

New in version 2015.5.10.

`salt.modules.scsi.`**`rescan_all`**(*host*)
List scsi devices

CLI Example:

```
salt '*' scsi.rescan_all 0
```

21.9.376 salt.modules.sdb

Module for Manipulating Data via the Salt DB API

`salt.modules.sdb.`**`delete`**(*uri*)
Delete a value from a db, using a uri in the form of `sdb://<profile>/<key>`. If the uri provided does not start with `sdb://` or the value is not successfully deleted, return `False`.

CLI Example:

```
salt '*' sdb.delete sdb://mymemcached/foo
```

`salt.modules.sdb.`**`get`**(*uri*)
Get a value from a db, using a uri in the form of sdb://<profile>/<key>. If the uri provided does not start with sdb://, then it will be returned as-is.

CLI Example:

```
salt '*' sdb.get sdb://mymemcached/foo
```

`salt.modules.sdb.`**`get_or_set_hash`**(*uri, length=8, chars='abcdefghijklmnopqrstuvwxyz0123456789!@#$%^&*(-_=+)'*)

Perform a one-time generation of a hash and write it to sdb. If that value has already been set return the value instead.

This is useful for generating passwords or keys that are specific to multiple minions that need to be stored somewhere centrally.

State Example:

```
some_mysql_user:
  mysql_user:
    - present
    - host: localhost
    - password: '{{ salt['sdb.get_or_set_hash']('some_mysql_user_pass') }}'
```

CLI Example:

```
salt '*' sdb.get_or_set_hash 'SECRET_KEY' 50
```

> **Warning:** This function could return strings which may contain characters which are reserved as directives by the YAML parser, such as strings beginning with %. To avoid issues when using the output of this function in an SLS file containing YAML+Jinja, surround the call with single quotes.

`salt.modules.sdb.`**`set_`**(*uri, value*)

Set a value in a db, using a uri in the form of `sdb://<profile>/<key>`. If the uri provided does not start with `sdb://` or the value is not successfully set, return `False`.

CLI Example:

```
salt '*' sdb.set sdb://mymemcached/foo bar
```

21.9.377 salt.modules.seed

Virtual machine image management tools

`salt.modules.seed.`**`apply_`**(*path, id_=None, config=None, approve_key=True, install=True, prep_install=False, pub_key=None, priv_key=None, mount_point=None*)

Seed a location (disk image, directory, or block device) with the minion config, approve the minion's key, and/or install salt-minion.

CLI Example:

```
salt 'minion' seed.apply path id [config=config_data] \
        [gen_key=(true|false)] [approve_key=(true|false)] \
        [install=(true|false)]
```

path Full path to the directory, device, or disk image on the target minion's file system.

id Minion id with which to seed the path.

config Minion configuration options. By default, the 'master' option is set to the target host's 'master'.

approve_key Request a pre-approval of the generated minion key. Requires that the salt-master be configured to either auto-accept all keys or expect a signing request from the target host. Default: true.

install Install salt-minion, if absent. Default: true.

prep_install Prepare the bootstrap script, but don't run it. Default: false `salt.modules.seed.`**`mkconfig`**(*config=None*, *tmp=None*, *id_=None*, *approve_key=True*, *pub_key=None*, *priv_key=None*)

Generate keys and config and put them in a tmp directory.

pub_key absolute path or file content of an optional preseeded salt key

priv_key absolute path or file content of an optional preseeded salt key

CLI Example:

```
salt 'minion' seed.mkconfig [config=config_data] [tmp=tmp_dir] \
        [id_=minion_id] [approve_key=(true|false)]
```

`salt.modules.seed.`**`prep_bootstrap`**(*mpt*)

Update and get the random script to a random place

CLI Example:

```
salt '*' seed.prep_bootstrap /tmp
```

21.9.378 salt.modules.selinux

Execute calls on selinux

Note: This module requires the `semanage`, `setsebool`, and `semodule` commands to be available on the minion. On RHEL-based distributions, ensure that the `policycoreutils` and `policycoreutils-python` packages are installed. If not on a Fedora or RHEL-based distribution, consult the selinux documentation for your distribution to ensure that the proper packages are installed.

`salt.modules.selinux.`**`fcontext_add_or_delete_policy`**(*action*, *name*, *filetype=None*, *sel_type=None*, *sel_user=None*, *sel_level=None*)

New in version 2017.7.0.

Adds or deletes the SELinux policy for a given filespec and other optional parameters.

Returns the result of the call to semanage.

Note that you don't have to remove an entry before setting a new one for a given filespec and filetype, as adding one with semanage automatically overwrites a previously configured SELinux context.

Warning: Use *`selinux.fcontext_add_policy()`*, or *`selinux.fcontext_delete_policy()`*.

Deprecated since version 2019.2.0.

action The action to perform. Either `add` or `delete`.

name filespec of the file or directory. Regex syntax is allowed.

file_type The SELinux filetype specification. Use one of [a, f, d, c, b, s, l, p]. See also `man semanage-fcontext`. Defaults to 'a' (all files).

sel_type SELinux context type. There are many.

sel_user SELinux user. Use `semanage login -l` to determine which ones are available to you.

sel_level The MLS range of the SELinux context.

CLI Example:

```
salt '*' selinux.fcontext_add_or_delete_policy add my-policy
```

`salt.modules.selinux.`**`fcontext_add_policy`**(*name, filetype=None, sel_type=None, sel_user=None, sel_level=None*)

New in version 2019.2.0.

Adds the SELinux policy for a given filespec and other optional parameters.

Returns the result of the call to semanage.

Note that you don't have to remove an entry before setting a new one for a given filespec and filetype, as adding one with semanage automatically overwrites a previously configured SELinux context.

name filespec of the file or directory. Regex syntax is allowed.

file_type The SELinux filetype specification. Use one of [a, f, d, c, b, s, l, p]. See also `man semanage-fcontext`. Defaults to 'a' (all files).

sel_type SELinux context type. There are many.

sel_user SELinux user. Use `semanage login -l` to determine which ones are available to you.

sel_level The MLS range of the SELinux context.

CLI Example:

```
salt '*' selinux.fcontext_add_policy my-policy
```

`salt.modules.selinux.`**`fcontext_apply_policy`**(*name, recursive=False*)

New in version 2017.7.0.

Applies SElinux policies to filespec using *restorecon [-R] filespec*. Returns dict with changes if successful, the output of the restorecon command otherwise.

name filespec of the file or directory. Regex syntax is allowed.

recursive Recursively apply SELinux policies.

CLI Example:

```
salt '*' selinux.fcontext_apply_policy my-policy
```

`salt.modules.selinux.`**`fcontext_delete_policy`**(*name, filetype=None, sel_type=None, sel_user=None, sel_level=None*)

New in version 2019.2.0.

Deletes the SELinux policy for a given filespec and other optional parameters.

Returns the result of the call to semanage.

Note that you don't have to remove an entry before setting a new one for a given filespec and filetype, as adding one with semanage automatically overwrites a previously configured SELinux context.

name filespec of the file or directory. Regex syntax is allowed.

file_type The SELinux filetype specification. Use one of [a, f, d, c, b, s, l, p]. See also `man semanage-fcontext`. Defaults to 'a' (all files).

sel_type SELinux context type. There are many.

sel_user SELinux user. Use `semanage login -l` to determine which ones are available to you.

sel_level The MLS range of the SELinux context.

CLI Example:

```
salt '*' selinux.fcontext_delete_policy my-policy
```

`salt.modules.selinux.`**`fcontext_get_policy`**(*name, filetype=None, sel_type=None, sel_user=None, sel_level=None*)

New in version 2017.7.0.

Returns the current entry in the SELinux policy list as a dictionary. Returns None if no exact match was found.

Returned keys are:

- filespec (the name supplied and matched)
- filetype (the descriptive name of the filetype supplied)

- sel_user, sel_role, sel_type, sel_level (the selinux context)

For a more in-depth explanation of the selinux context, go to https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Security-Enhanced_Linux/chap-Security-Enhanced_Linux-SELinux_Contexts.html

name filespec of the file or directory. Regex syntax is allowed.

filetype The SELinux filetype specification. Use one of [a, f, d, c, b, s, l, p]. See also *man semanage-fcontext*. Defaults to 'a' (all files).

CLI Example:

```
salt '*' selinux.fcontext_get_policy my-policy
```

`salt.modules.selinux.`**`fcontext_policy_is_applied`**`(`*name*, *recursive=False*`)`

New in version 2017.7.0.

Returns an empty string if the SELinux policy for a given filespec is applied, returns string with differences in policy and actual situation otherwise.

name filespec of the file or directory. Regex syntax is allowed.

CLI Example:

```
salt '*' selinux.fcontext_policy_is_applied my-policy
```

`salt.modules.selinux.`**`filetype_id_to_string`**`(`*filetype='a'*`)`

New in version 2017.7.0.

Translates SELinux filetype single-letter representation to a more human-readable version (which is also used in *semanage fcontext -l*).

`salt.modules.selinux.`**`getconfig`**`()`

Return the selinux mode from the config file

CLI Example:

```
salt '*' selinux.getconfig
```

`salt.modules.selinux.`**`getenforce`**`()`

Return the mode selinux is running in

CLI Example:

```
salt '*' selinux.getenforce
```

`salt.modules.selinux.`**`getsebool`**`(`*boolean*`)`

Return the information on a specific selinux boolean

CLI Example:

```
salt '*' selinux.getsebool virt_use_usb
```

`salt.modules.selinux.`**`getsemod`**`(`*module*`)`

Return the information on a specific selinux module

CLI Example:

```
salt '*' selinux.getsemod mysql
```

New in version 2016.3.0.

`salt.modules.selinux.`**`install_semod`**`(`*module_path*`)`

Install custom SELinux module from file

CLI Example:

```
salt '*' selinux.install_semod [salt://]path/to/module.pp
```

New in version 2016.11.6.

`salt.modules.selinux.`**`list_sebool`**`()`

Return a structure listing all of the selinux booleans on the system and what state they are in

CLI Example:

```
salt '*' selinux.list_sebool
```

`salt.modules.selinux.`**`list_semod`**`()`

Return a structure listing all of the selinux modules on the system and what state they are in

CLI Example:

```
salt '*' selinux.list_semod
```

New in version 2016.3.0.

`salt.modules.selinux.`**`port_add_policy`**`(`*name, sel_type=None, protocol=None, port=None, sel_range=None*`)`

New in version 2019.2.0.

Adds the SELinux policy for a given protocol and port.

Returns the result of the call to semanage.

name The protocol and port spec. Can be formatted as `(tcp|udp)/(port|port-range)`.

sel_type The SELinux Type. Required.

protocol The protocol for the port, `tcp` or `udp`. Required if name is not formatted.

port The port or port range. Required if name is not formatted.

sel_range The SELinux MLS/MCS Security Range.

CLI Example:

```
salt '*' selinux.port_add_policy add tcp/8080 http_port_t
salt '*' selinux.port_add_policy add foobar http_port_t protocol=tcp port=8091
```

`salt.modules.selinux.`**`port_delete_policy`**`(`*name, protocol=None, port=None*`)`

New in version 2019.2.0.

Deletes the SELinux policy for a given protocol and port.

Returns the result of the call to semanage.

name The protocol and port spec. Can be formatted as `(tcp|udp)/(port|port-range)`.

protocol The protocol for the port, `tcp` or `udp`. Required if name is not formatted.

port The port or port range. Required if name is not formatted.

CLI Example:

```
salt '*' selinux.port_delete_policy tcp/8080
salt '*' selinux.port_delete_policy foobar protocol=tcp port=8091
```

`salt.modules.selinux.`**`port_get_policy`**`(`*name, sel_type=None, protocol=None, port=None*`)`

New in version 2019.2.0.

Returns the current entry in the SELinux policy list as a dictionary. Returns None if no exact match was found.

Returned keys are:

- sel_type (the selinux type)
- proto (the protocol)

- port (the port(s) and/or port range(s))

name The protocol and port spec. Can be formatted as `(tcp|udp)/(port|port-range)`.
sel_type The SELinux Type.
protocol The protocol for the port, `tcp` or `udp`. Required if name is not formatted.
port The port or port range. Required if name is not formatted.

CLI Example:

```
salt '*' selinux.port_get_policy tcp/80
salt '*' selinux.port_get_policy foobar protocol=tcp port=80
```

`salt.modules.selinux.`**`remove_semod`**`(`*module*`)`

Remove SELinux module

CLI Example:

```
salt '*' selinux.remove_semod module_name
```

New in version 2016.11.6.

`salt.modules.selinux.`**`selinux_fs_path`**`()`

Return the location of the SELinux VFS directory

CLI Example:

```
salt '*' selinux.selinux_fs_path
```

`salt.modules.selinux.`**`setenforce`**`(`*mode*`)`

Set the SELinux enforcing mode

CLI Example:

```
salt '*' selinux.setenforce enforcing
```

`salt.modules.selinux.`**`setsebool`**`(`*boolean, value, persist=False*`)`

Set the value for a boolean

CLI Example:

```
salt '*' selinux.setsebool virt_use_usb off
```

`salt.modules.selinux.`**`setsebools`**`(`*pairs, persist=False*`)`

Set the value of multiple booleans

CLI Example:

```
salt '*' selinux.setsebools '{virt_use_usb: on, squid_use_tproxy: off}'
```

`salt.modules.selinux.`**`setsemod`**`(`*module, state*`)`

Enable or disable an SELinux module.

CLI Example:

```
salt '*' selinux.setsemod nagios Enabled
```

New in version 2016.3.0.

21.9.379 salt.modules.sensehat

Module for controlling the LED matrix or reading environment data on the SenseHat of a Raspberry Pi.

New in version 2017.7.0.

maintainer Benedikt Werner <1benediktwerner@gmail.com>, Joachim Werner <joe@suse.com>

maturity new

depends sense_hat Python module

The rotation of the Pi can be specified in a pillar. This is useful if the Pi is used upside down or sideways to correct the orientation of the image being shown.

Example:

```
sensehat:
    rotation: 90
```

`salt.modules.sensehat.`**`clear`**`(`*color=None*`)`

Sets the LED matrix to a single color or turns all LEDs off.

CLI Example:

```
salt 'raspberry' sensehat.clear
salt 'raspberry' sensehat.clear '[255, 0, 0]'
```

`salt.modules.sensehat.`**`get_humidity`**`()`

Get the percentage of relative humidity from the humidity sensor.

`salt.modules.sensehat.`**`get_pixel`**`(`*x, y*`)`

Returns the color of a single pixel on the LED matrix.
x The x coordinate of the pixel. Ranges from 0 on the left to 7 on the right.
y The y coordinate of the pixel. Ranges from 0 at the top to 7 at the bottom.

Note: Please read the note for `get_pixels`

`salt.modules.sensehat.`**`get_pixels`**`()`

Returns a list of 64 smaller lists of `[R, G, B]` pixels representing the the currently displayed image on the LED matrix.

Note: When using `set_pixels` the pixel values can sometimes change when you read them again using `get_pixels`. This is because we specify each pixel element as 8 bit numbers (0 to 255) but when they're passed into the Linux frame buffer for the LED matrix the numbers are bit shifted down to fit into RGB 565. 5 bits for red, 6 bits for green and 5 bits for blue. The loss of binary precision when performing this conversion (3 bits lost for red, 2 for green and 3 for blue) accounts for the discrepancies you see.

The `get_pixels` method provides an accurate representation of how the pixels end up in frame buffer memory after you have called `set_pixels`.

`salt.modules.sensehat.`**`get_pressure`**`()`

Gets the current pressure in Millibars from the pressure sensor.

`salt.modules.sensehat.`**`get_temperature`**`()`

Gets the temperature in degrees Celsius from the humidity sensor. Equivalent to calling `get_temperature_from_humidity`.

If you get strange results try using `get_temperature_from_pressure`.

`salt.modules.sensehat.`**`get_temperature_from_humidity`**`()`

Gets the temperature in degrees Celsius from the humidity sensor.

`salt.modules.sensehat.`**`get_temperature_from_pressure`**`()`

Gets the temperature in degrees Celsius from the pressure sensor.

`salt.modules.sensehat.`**`low_light`**`(`*low_light=True*`)`

Sets the LED matrix to low light mode. Useful in a dark environment.

CLI Example:

```
salt 'raspberry' sensehat.low_light
salt 'raspberry' sensehat.low_light False
```

`salt.modules.sensehat.`**`set_pixel`**`(`*x, y, color*`)`

Sets a single pixel on the LED matrix to a specified color.

x The x coordinate of the pixel. Ranges from 0 on the left to 7 on the right.

y The y coordinate of the pixel. Ranges from 0 at the top to 7 at the bottom.

color The new color of the pixel as a list of `[R, G, B]` values.

CLI Example:

```
salt 'raspberry' sensehat.set_pixel 0 0 '[255, 0, 0]'
```

`salt.modules.sensehat.`**`set_pixels`**`(`*pixels*`)`

Sets the entire LED matrix based on a list of 64 pixel values

pixels A list of 64 `[R, G, B]` color values.

`salt.modules.sensehat.`**`show_image`**`(`*image*`)`

Displays an 8 x 8 image on the LED matrix.

image The path to the image to display. The image must be 8 x 8 pixels in size.

CLI Example:

```
salt 'raspberry' sensehat.show_image /tmp/my_image.png
```

`salt.modules.sensehat.`**`show_letter`**`(`*letter, text_color=None, back_color=None*`)`

Displays a single letter on the LED matrix.

letter The letter to display

text_color The color in which the letter is shown. Defaults to '[255, 255, 255]' (white).

back_color The background color of the display. Defaults to '[0, 0, 0]' (black).

CLI Example:

```
salt 'raspberry' sensehat.show_letter O
salt 'raspberry' sensehat.show_letter X '[255, 0, 0]'
salt 'raspberry' sensehat.show_letter B '[0, 0, 255]' '[255, 255, 0]'
```

`salt.modules.sensehat.`**`show_message`**`(`*message, msg_type=None, text_color=None, back_color=None, scroll_speed=0.1*`)`

Displays a message on the LED matrix.

message The message to display

msg_type The type of the message. Changes the appearance of the message.

Available types are:

```
error:      red text
warning:    orange text
success:    green text
info:       blue text
```

scroll_speed The speed at which the message moves over the LED matrix. This value represents the time paused for between shifting the text to the left by one column of pixels. Defaults to '0.1'.

text_color The color in which the message is shown. Defaults to '[255, 255, 255]' (white).

back_color The background color of the display. Defaults to '[0, 0, 0]' (black).

CLI Example:

```
salt 'raspberry' sensehat.show_message 'Status ok'
salt 'raspberry' sensehat.show_message 'Something went wrong' error
salt 'raspberry' sensehat.show_message 'Red' text_color='[255, 0, 0]'
salt 'raspberry' sensehat.show_message 'Hello world' None '[0, 0, 255]' '[255,
↪255, 0]' 0.2
```

21.9.380 salt.modules.sensors

Read lm-sensors

New in version 2014.1.3.

`salt.modules.sensors.`**`sense`**(*chip*, *fahrenheit=False*)

Gather lm-sensors data from a given chip

To determine the chip to query, use the 'sensors' command and see the leading line in the block.

Example:

/usr/bin/sensors

coretemp-isa-0000 Adapter: ISA adapter Physical id 0: +56.0°C (high = +87.0°C, crit = +105.0°C) Core 0: +52.0°C (high = +87.0°C, crit = +105.0°C) Core 1: +50.0°C (high = +87.0°C, crit = +105.0°C) Core 2: +56.0°C (high = +87.0°C, crit = +105.0°C) Core 3: +53.0°C (high = +87.0°C, crit = +105.0°C)

Given the above, the chip is 'coretemp-isa-0000'.

21.9.381 salt.modules.serverdensity_device

Wrapper around Server Density API

New in version 2014.7.0.

`salt.modules.serverdensity_device.`**`create`**(*name*, ***params*)

Function to create device in Server Density. For more info, see the API docs.

CLI Example:

```
salt '*' serverdensity_device.create lama
salt '*' serverdensity_device.create rich_lama group=lama_band installedRAM=32768
```

`salt.modules.serverdensity_device.`**`delete`**(*device_id*)

Delete a device from Server Density. For more information, see the API docs.

CLI Example:

```
salt '*' serverdensity_device.delete 51f7eafcdba4bb235e000ae4
```

`salt.modules.serverdensity_device.`**`get_sd_auth`**(*val*, *sd_auth_pillar_name='serverdensity'*)

Returns requested Server Density authentication value from pillar.

CLI Example:

```
salt '*' serverdensity_device.get_sd_auth <val>
```

salt.modules.serverdensity_device.**install_agent**(*agent_key, agent_version=1*)

Function downloads Server Density installation agent, and installs sd-agent with agent_key. Optionally the agent_version would select the series to use (defaults on the v1 one).

CLI Example:

```
salt '*' serverdensity_device.install_agent c2bbdd6689ff46282bdaa07555641498
salt '*' serverdensity_device.install_agent c2bbdd6689ff46282bdaa07555641498 2
```

salt.modules.serverdensity_device.**ls**(***params*)

List devices in Server Density

Results will be filtered by any params passed to this function. For more information, see the API docs on listing and searching.

CLI Example:

```
salt '*' serverdensity_device.ls
salt '*' serverdensity_device.ls name=lama
salt '*' serverdensity_device.ls name=lama group=lama_band installedRAM=32768
```

salt.modules.serverdensity_device.**update**(*device_id, **params*)

Updates device information in Server Density. For more information see the API docs.

CLI Example:

```
salt '*' serverdensity_device.update 51f7eafcdba4bb235e000ae4 name=lama␣
↪group=lama_band
salt '*' serverdensity_device.update 51f7eafcdba4bb235e000ae4 name=better_lama␣
↪group=rock_lamas swapSpace=512
```

21.9.382 salt.modules.servicenow module

Module for execution of ServiceNow CI (configuration items)

New in version 2016.11.0.

depends servicenow_rest python module

configuration Configure this module by specifying the name of a configuration profile in the minion config, minion pillar, or master config. The module will use the 'servicenow' key by default, if defined.

For example:

```
servicenow:
  instance_name: ''
  username: ''
  password: ''
```

salt.modules.servicenow.**delete_record**(*table, sys_id*)

Delete an existing record

Parameters

- **table** (str) – The table name, e.g. sys_user
- **sys_id** (str) – The unique ID of the record

CLI Example:

```
salt myminion servicenow.delete_record sys_computer 2134566
```

`salt.modules.servicenow.`**`non_structured_query`**(*table*, *query=None*, ***kwargs*)

Run a non-structed (not a dict) query on a servicenow table. See http://wiki.servicenow.com/index.php?title=Encoded_Query_Strings#gsc.tab=0 for help on constructing a non-structured query string.

Parameters

- **`table`** (`str`) – The table name, e.g. sys_user
- **`query`** (`str`) – The query to run (or use keyword arguments to filter data)

CLI Example:

```
salt myminion servicenow.non_structured_query sys_computer 'role=web'
salt myminion servicenow.non_structured_query sys_computer role=web type=computer
```

`salt.modules.servicenow.`**`set_change_request_state`**(*change_id*, *state='approved'*)

Set the approval state of a change request/record

Parameters

- **`change_id`** (`str`) – The ID of the change request, e.g. CHG123545
- **`state`** (`str`) – The target state, e.g. approved

CLI Example:

```
salt myminion servicenow.set_change_request_state CHG000123 declined
salt myminion servicenow.set_change_request_state CHG000123 approved
```

`salt.modules.servicenow.`**`update_record_field`**(*table*, *sys_id*, *field*, *value*)

Update the value of a record's field in a servicenow table

Parameters

- **`table`** (`str`) – The table name, e.g. sys_user
- **`sys_id`** (`str`) – The unique ID of the record
- **`field`** (`str`) – The new value
- **`value`** (`str`) – The new value

CLI Example:

```
salt myminion servicenow.update_record_field sys_user 2348234 first_name jimmy
```

21.9.383 salt.modules.slack_notify

Module for sending messages to Slack

New in version 2015.5.0.

configuration This module can be used by either passing an api key and version directly or by specifying both in a configuration profile in the salt master/minion config.

For example:

```
slack:
  api_key: peWcBiMOS9HrZG15peWcBiMOS9HrZG15
```

`salt.modules.slack_notify.`**`call_hook`**(*message*, *attachment=None*, *color='good'*, *short=False*, *identifier=None*, *channel=None*, *username=None*, *icon_emoji=None*)

Send message to Slack incoming webhook.

Parameters

- **`message`** – The topic of message.
- **`attachment`** – The message to send to the Slacke WebHook.

- **color** – The color of border of left side
- **short** – An optional flag indicating whether the value is short enough to be displayed side-by-side with other values.
- **identifier** – The identifier of WebHook.
- **channel** – The channel to use instead of the WebHook default.
- **username** – Username to use instead of WebHook default.
- **icon_emoji** – Icon to use instead of WebHook default.

Returns Boolean if message was sent successfully.

CLI Example:

```
salt '*' slack.call_hook message='Hello, from SaltStack'
```

`salt.modules.slack_notify.`**`find_room`**`(`*name*, *api_key=None*`)`

Find a room by name and return it.

Parameters

- **name** – The room name.
- **api_key** – The Slack admin api key.

Returns The room object.

CLI Example:

```
salt '*' slack.find_room name="random"

salt '*' slack.find_room name="random" api_key=peWcBiMOS9HrZG15peWcBiMOS9HrZG15
```

`salt.modules.slack_notify.`**`find_user`**`(`*name*, *api_key=None*`)`

Find a user by name and return it.

Parameters

- **name** – The user name.
- **api_key** – The Slack admin api key.

Returns The user object.

CLI Example:

```
salt '*' slack.find_user name="ThomasHatch"

salt '*' slack.find_user name="ThomasHatch" api_
↪key=peWcBiMOS9HrZG15peWcBiMOS9HrZG15
```

`salt.modules.slack_notify.`**`list_rooms`**`(`*api_key=None*`)`

List all Slack rooms.

Parameters **api_key** – The Slack admin api key.

Returns The room list.

CLI Example:

```
salt '*' slack.list_rooms

salt '*' slack.list_rooms api_key=peWcBiMOS9HrZG15peWcBiMOS9HrZG15
```

`salt.modules.slack_notify.`**`list_users`**`(`*api_key=None*`)`

List all Slack users.

Parameters **api_key** – The Slack admin api key.

Returns The user list.

CLI Example:

```
salt '*' slack.list_users

salt '*' slack.list_users api_key=peWcBiMOS9HrZG15peWcBiMOS9HrZG15
```

`salt.modules.slack_notify.`**`post_message`**(*channel, message, from_name, api_key=None, icon=None*)

Send a message to a Slack channel.

Parameters

- **`channel`** – The channel name, either will work.
- **`message`** – The message to send to the Slack channel.
- **`from_name`** – Specify who the message is from.
- **`api_key`** – The Slack api key, if not specified in the configuration.
- **`icon`** – URL to an image to use as the icon for this message

Returns Boolean if message was sent successfully.

CLI Example:

```
salt '*' slack.post_message channel="Development Room" message="Build is done"
↪from_name="Build Server"
```

21.9.384 salt.modules.slsutil

Utility functions for use with or in SLS files

`salt.modules.slsutil.`**`deserialize`**(*serializer, stream_or_string, **mod_kwargs*)

Deserialize a Python object using a `serializer module`

CLI Example:

```
salt '*' slsutil.deserialize 'json' '{"foo": "Foo!"}'
salt '*' --no-parse=stream_or_string slsutil.deserialize 'json' \
    stream_or_string='{"foo": "Foo!"}'
```

Jinja Example:

```
{% set python_object = salt.slsutil.deserialize('json',
    '{"foo": "Foo!"}') %}
```

`salt.modules.slsutil.`**`merge`**(*obj_a, obj_b, strategy='smart', renderer='yaml', merge_lists=False*)

Merge a data structure into another by choosing a merge strategy

Strategies:

- aggregate
- list
- overwrite
- recurse
- smart

CLI Example:

```
salt '*' slsutil.merge '{foo: Foo}' '{bar: Bar}'
```

`salt.modules.slsutil.`**`merge_all`**(*lst, strategy='smart', renderer='yaml', merge_lists=False*)

New in version 2019.2.0.

Merge a list of objects into each other in order

Parameters

- **`lst`** (*`Iterable`*) – List of objects to be merged.
- **`strategy`** (*`String`*) – Merge strategy. See utils.dictupdate.
- **`renderer`** (*`String`*) – Renderer type. Used to determine strategy when strategy is 'smart'.
- **`merge_lists`** (*`Bool`*) – Defines whether to merge embedded object lists.

CLI Example:

```
$ salt-call --output=txt slsutil.merge_all '[{foo: Foo}, {foo: Bar}]'
local: {u'foo': u'Bar'}
```

`salt.modules.slsutil.`**`renderer`**(*path=None*, *string=None*, *default_renderer='jinja|yaml'*, ***kwargs*)

Parse a string or file through Salt's renderer system

Changed in version 2018.3.0: Add support for Salt fileserver URIs.

This is an open-ended function and can be used for a variety of tasks. It makes use of Salt's "renderer pipes" system to run a string or file through a pipe of any of the loaded renderer modules.

Parameters

- **`path`** – The path to a file on Salt's fileserver (any URIs supported by `cp.get_url`) or on the local file system.
- **`string`** – An inline string to be used as the file to send through the renderer system. Note, not all renderer modules can work with strings; the 'py' renderer requires a file, for example.
- **`default_renderer`** – The renderer pipe to send the file through; this is overridden by a "she-bang" at the top of the file.
- **`kwargs`** – Keyword args to pass to Salt's compile_template() function.

Keep in mind the goal of each renderer when choosing a render-pipe; for example, the Jinja renderer processes a text file and produces a string, however the YAML renderer processes a text file and produces a data structure.

One possible use is to allow writing "map files", as are commonly seen in Salt formulas, but without tying the renderer of the map file to the renderer used in the other sls files. In other words, a map file could use the Python renderer and still be included and used by an sls file that uses the default 'jinja|yaml' renderer.

For example, the two following map files produce identical results but one is written using the normal 'jinja|yaml' and the other is using 'py':

```
#!jinja|yaml
{% set apache = salt.grains.filter_by({
    ...normal jinja map file here...
}, merge=salt.pillar.get('apache:lookup')) %}
{{ apache | yaml() }}
```

```
#!py
def run():
    apache = __salt__.grains.filter_by({
        ...normal map here but as a python dict...
    }, merge=__salt__.pillar.get('apache:lookup'))
    return apache
```

Regardless of which of the above map files is used, it can be accessed from any other sls file by calling this function. The following is a usage example in Jinja:

```
{% set apache = salt.slsutil.renderer('map.sls') %}
```

CLI Example:

```
salt '*' slsutil.renderer salt://path/to/file
salt '*' slsutil.renderer /path/to/file
salt '*' slsutil.renderer /path/to/file.jinja 'jinja'
salt '*' slsutil.renderer /path/to/file.sls 'jinja|yaml'
salt '*' slsutil.renderer string='Inline template! {{ saltenv }}'
salt '*' slsutil.renderer string='Hello, {{ name }}.' name='world'
```

`salt.modules.slsutil.`**`serialize`**(*serializer*, *obj*, ***mod_kwargs*)
Serialize a Python object using a `serializer module`

CLI Example:

```
salt '*' --no-parse=obj slsutil.serialize 'json' obj="{'foo': 'Foo!'}
```

Jinja Example:

```
{% set json_string = salt.slsutil.serialize('json',
    {'foo': 'Foo!'}) %}
```

`salt.modules.slsutil.`**`update`**(*dest*, *upd*, *recursive_update=True*, *merge_lists=False*)
Merge `upd` recursively into `dest`

If `merge_lists=True`, will aggregate list object types instead of replacing. This behavior is only activated when `recursive_update=True`.

CLI Example:

```
salt '*' slsutil.update '{foo: Foo}' '{bar: Bar}'
```

21.9.385 salt.modules.smartos_imgadm

Module for running imgadm command on SmartOS

`salt.modules.smartos_imgadm.`**`avail`**(*search=None*, *verbose=False*)
Return a list of available images
search [string] search keyword
verbose [boolean (False)] toggle verbose output
CLI Example:

```
salt '*' imgadm.avail [percona]
salt '*' imgadm.avail verbose=True
```

`salt.modules.smartos_imgadm.`**`delete`**(*uuid*)
Remove an installed image
uuid [string] Specifies uuid to import
CLI Example:

```
salt '*' imgadm.delete e42f8c84-bbea-11e2-b920-078fab2aab1f
```

`salt.modules.smartos_imgadm.`**`docker_to_uuid`**(*uuid*)
Get the image uuid from an imported docker image

New in version 2019.2.0.

`salt.modules.smartos_imgadm.`**`get`**(*uuid*)
Return info on an installed image
uuid [string] uuid of image
CLI Example:

```
salt '*' imgadm.get e42f8c84-bbea-11e2-b920-078fab2aab1f
salt '*' imgadm.get plexinc/pms-docker:plexpass
```

`salt.modules.smartos_imgadm.`**`import_image`**(*uuid*, *verbose=False*)
Import an image from the repository
uuid [string] uuid to import

verbose [boolean (False)] toggle verbose output
CLI Example:

```
salt '*' imgadm.import e42f8c84-bbea-11e2-b920-078fab2aab1f [verbose=True]
```

`salt.modules.smartos_imgadm.`**`list_installed`**(*verbose=False*)
Return a list of installed images
verbose [boolean (False)] toggle verbose output
Changed in version 2019.2.0: Docker images are now also listed

CLI Example:

```
salt '*' imgadm.list
salt '*' imgadm.list docker=True
salt '*' imgadm.list verbose=True
```

`salt.modules.smartos_imgadm.`**`show`**(*uuid*)
Show manifest of a given image
uuid [string] uuid of image
CLI Example:

```
salt '*' imgadm.show e42f8c84-bbea-11e2-b920-078fab2aab1f
salt '*' imgadm.show plexinc/pms-docker:plexpass
```

`salt.modules.smartos_imgadm.`**`source_add`**(*source, source_type='imgapi'*)
Add a new source
source [string] source url to add
source_trype [string (imgapi)] source type, either imgapi or docker
New in version 2019.2.0.

CLI Example:

```
salt '*' imgadm.source_add https://updates.joyent.com
salt '*' imgadm.source_add https://docker.io docker
```

`salt.modules.smartos_imgadm.`**`source_delete`**(*source*)
Delete a source
source [string] source url to delete
New in version 2019.2.0.

CLI Example:

```
salt '*' imgadm.source_delete https://updates.joyent.com
```

`salt.modules.smartos_imgadm.`**`sources`**(*verbose=False*)
Return a list of available sources
verbose [boolean (False)] toggle verbose output
New in version 2019.2.0.

CLI Example:

```
salt '*' imgadm.sources
```

`salt.modules.smartos_imgadm.`**`update_installed`**(*uuid="*)
Gather info on unknown image(s) (locally installed)
uuid [string] optional uuid of image
CLI Example:

```
salt '*' imgadm.update [uuid]
```

salt.modules.smartos_imgadm.**vacuum**(*verbose=False*)

Remove unused images

verbose [boolean (False)] toggle verbose output

CLI Example:

```
salt '*' imgadm.vacuum [verbose=True]
```

salt.modules.smartos_imgadm.**version**()

Return imgadm version

CLI Example:

```
salt '*' imgadm.version
```

21.9.386 salt.modules.smartos_nictagadm module

Module for running nictagadm command on SmartOS :maintainer: Jorge Schrauwen <sjorge@blackdot.be> :maturity: new :depends: nictagadm binary, dladm binary :platform: smartos

..versionadded:: 2016.11.0

salt.modules.smartos_nictagadm.**add**(*name, mac, mtu=1500*)

Add a new nictag

name [string] name of new nictag

mac [string] mac of parent interface or 'etherstub' to create a ether stub

mtu [int] MTU (ignored for etherstubs)

CLI Example:

```
salt '*' nictagadm.add storage0 etherstub
salt '*' nictagadm.add trunk0 'DE:AD:OO:OO:BE:EF' 9000
```

salt.modules.smartos_nictagadm.**delete**(*name, force=False*)

Delete nictag

name [string] nictag to delete

force [boolean] force delete even if vms attached

CLI Example:

```
salt '*' nictagadm.exists admin
```

salt.modules.smartos_nictagadm.**exists**(**nictag, **kwargs*)

Check if nictags exists

nictag [string] one or more nictags to check

verbose [boolean] return list of nictags

CLI Example:

```
salt '*' nictagadm.exists admin
```

salt.modules.smartos_nictagadm.**list_nictags**(*include_etherstubs=True*)

List all nictags

include_etherstubs [boolean] toggle include of etherstubs

CLI Example:

```
salt '*' nictagadm.list
```

`salt.modules.smartos_nictagadm.`**`update`**`(`*name, mac=None, mtu=None*`)`

Update a nictag

name [string] name of nictag

mac [string] optional new mac for nictag

mtu [int] optional new MTU for nictag

CLI Example:

```
salt '*' nictagadm.update trunk mtu=9000
```

`salt.modules.smartos_nictagadm.`**`vms`**`(`*nictag*`)`

List all vms connect to nictag

nictag [string] name of nictag

CLI Example:

```
salt '*' nictagadm.vms admin
```

21.9.387 salt.modules.smartos_virt

virst compatibility module for managing VMs on SmartOS

`salt.modules.smartos_virt.`**`get_macs`**`(`*domain*`)`

Return a list off MAC addresses from the named VM

CLI Example:

```
salt '*' virt.get_macs <domain>
```

`salt.modules.smartos_virt.`**`init`**`(`***kwargs*`)`

Initialize a new VM

CLI Example:

```
salt '*' virt.init image_uuid='...' alias='...' [...]
```

`salt.modules.smartos_virt.`**`list_active_vms`**`()`

Return a list of uuids for active virtual machine on the minion

CLI Example:

```
salt '*' virt.list_active_vms
```

`salt.modules.smartos_virt.`**`list_domains`**`()`

Return a list of virtual machine names on the minion

CLI Example:

```
salt '*' virt.list_domains
```

`salt.modules.smartos_virt.`**`list_inactive_vms`**`()`

Return a list of uuids for inactive virtual machine on the minion

CLI Example:

```
salt '*' virt.list_inactive_vms
```

`salt.modules.smartos_virt.`**`reboot`**(*domain*)

Reboot a domain via ACPI request

CLI Example:

```
salt '*' virt.reboot <domain>
```

`salt.modules.smartos_virt.`**`setmem`**(*domain*, *memory*)

Change the amount of memory allocated to VM. <memory> is to be specified in MB.

Note for KVM : this would require a restart of the VM.

CLI Example:

```
salt '*' virt.setmem <domain> 512
```

`salt.modules.smartos_virt.`**`shutdown`**(*domain*)

Send a soft shutdown signal to the named vm

CLI Example:

```
salt '*' virt.shutdown <domain>
```

`salt.modules.smartos_virt.`**`start`**(*domain*)

Start a defined domain

CLI Example:

```
salt '*' virt.start <domain>
```

`salt.modules.smartos_virt.`**`stop`**(*domain*)

Hard power down the virtual machine, this is equivalent to powering off the hardware.

CLI Example:

```
salt '*' virt.destroy <domain>
```

`salt.modules.smartos_virt.`**`vm_info`**(*domain*)

Return a dict with information about the specified VM on this CN

CLI Example:

```
salt '*' virt.vm_info <domain>
```

`salt.modules.smartos_virt.`**`vm_virt_type`**(*domain*)

Return VM virtualization type : OS or KVM

CLI Example:

```
salt '*' virt.vm_virt_type <domain>
```

21.9.388 salt.modules.smartos_vmadm

Module for running vmadm command on SmartOS

`salt.modules.smartos_vmadm.`**`create`**(*from_file=None*, ***kwargs*)

Create a new vm

from_file [string] json file to create the vm from – if present, all other options will be ignored

kwargs [string|int|...] options to set for the vm

CLI Example:

```
salt '*' vmadm.create from_file=/tmp/new_vm.json
salt '*' vmadm.create image_uuid='...' alias='...' nics='[{ "nic_tag": "admin",
↪"ip": "198.51.100.123", ...}, {...}]' [...]
```

`salt.modules.smartos_vmadm.`**`create_snapshot`**(*vm*, *name*, *key='uuid'*)

Create snapshot of a vm

vm [string] vm to be targeted

name [string]

snapshot name The snapname must be 64 characters or less and must only contain alphanumeric characters and characters in the set [-_.:%] to comply with ZFS restrictions.

key [string [uuid|alias|hostname]] value type of 'vm' parameter

CLI Example:

```
salt '*' vmadm.create_snapshot 186da9ab-7392-4f55-91a5-b8f1fe770543 baseline
salt '*' vmadm.create_snapshot nacl baseline key=alias
```

`salt.modules.smartos_vmadm.`**`delete`**(*vm*, *key='uuid'*)

Delete a vm

vm [string] vm to be deleted

key [string [uuid|alias|hostname]] value type of 'vm' parameter

CLI Example:

```
salt '*' vmadm.delete 186da9ab-7392-4f55-91a5-b8f1fe770543
salt '*' vmadm.delete nacl key=alias
```

`salt.modules.smartos_vmadm.`**`delete_snapshot`**(*vm*, *name*, *key='uuid'*)

Delete snapshot of a vm

vm [string] vm to be targeted

name [string]

snapshot name The snapname must be 64 characters or less and must only contain alphanumeric characters and characters in the set [-_.:%] to comply with ZFS restrictions.

key [string [uuid|alias|hostname]] value type of 'vm' parameter

CLI Example:

```
salt '*' vmadm.delete_snapshot 186da9ab-7392-4f55-91a5-b8f1fe770543 baseline
salt '*' vmadm.delete_snapshot nacl baseline key=alias
```

`salt.modules.smartos_vmadm.`**`get`**(*vm*, *key='uuid'*)

Output the JSON object describing a VM

vm [string] vm to be targeted

key [string [uuid|alias|hostname]] value type of 'vm' parameter

CLI Example:

```
salt '*' vmadm.get 186da9ab-7392-4f55-91a5-b8f1fe770543
salt '*' vmadm.get nacl key=alias
```

`salt.modules.smartos_vmadm.`**`info`**(*vm*, *info_type='all'*, *key='uuid'*)

Lookup info on running kvm

vm [string] vm to be targeted

info_type [string [all|block|blockstats|chardev|cpus|kvm|pci|spice|version|vnc]] info type to return

key [string [uuid|alias|hostname]] value type of 'vm' parameter

CLI Example:

```
salt '*' vmadm.info 186da9ab-7392-4f55-91a5-b8f1fe770543
salt '*' vmadm.info 186da9ab-7392-4f55-91a5-b8f1fe770543 vnc
salt '*' vmadm.info nacl key=alias
salt '*' vmadm.info nacl vnc key=alias
```

`salt.modules.smartos_vmadm.`**`list_vms`**(*search=None, sort=None, order='uuid, type, ram, state, alias', keyed=True*)

Return a list of VMs

search [string] vmadm filter property

sort [string] vmadm sort (-s) property

order [string] vmadm order (-o) property – Default: uuid,type,ram,state,alias

keyed [boolean]

specified if the output should be an array (False) or dict (True) For a dict the key is the first item from the order parameter. Note: If key is not unique last vm wins.

CLI Example:

```
salt '*' vmadm.list
salt '*' vmadm.list order=alias,ram,cpu_cap sort=-ram,-cpu_cap
salt '*' vmadm.list search='type=KVM'
```

`salt.modules.smartos_vmadm.`**`lookup`**(*search=None, order=None, one=False*)

Return a list of VMs using lookup

search [string] vmadm filter property

order [string] vmadm order (-o) property – Default: uuid,type,ram,state,alias

one [boolean] return only one result (vmadm's -1)

CLI Example:

```
salt '*' vmadm.lookup search='state=running'
salt '*' vmadm.lookup search='state=running' order=uuid,alias,hostname
salt '*' vmadm.lookup search='alias=nacl' one=True
```

`salt.modules.smartos_vmadm.`**`reboot`**(*vm, force=False, key='uuid'*)

Reboot a vm

vm [string] vm to be rebooted

force [boolean] force reboot of vm if true

key [string [uuid|alias|hostname]] value type of 'vm' parameter

CLI Example:

```
salt '*' vmadm.reboot 186da9ab-7392-4f55-91a5-b8f1fe770543
salt '*' vmadm.reboot 186da9ab-7392-4f55-91a5-b8f1fe770543 True
salt '*' vmadm.reboot vm=nacl key=alias
salt '*' vmadm.reboot vm=nina.example.org key=hostname
```

`salt.modules.smartos_vmadm.`**`receive`**(*uuid, source*)

Receive a vm from a directory

uuid [string] uuid of vm to be received

source [string] source directory

CLI Example:

```
salt '*' vmadm.receive 186da9ab-7392-4f55-91a5-b8f1fe770543 /opt/backups
```

`salt.modules.smartos_vmadm.`**`reprovision`**(*vm, image, key='uuid'*)

Reprovision a vm

vm [string] vm to be reprovisioned

image [string] uuid of new image

key [string [uuid|alias|hostname]] value type of 'vm' parameter

CLI Example:

```
salt '*' vmadm.reprovision 186da9ab-7392-4f55-91a5-b8f1fe770543 c02a2044-c1bd-
↪11e4-bd8c-dfc1db8b0182
salt '*' vmadm.reprovision nacl c02a2044-c1bd-11e4-bd8c-dfc1db8b0182 key=alias
```

`salt.modules.smartos_vmadm.`**`rollback_snapshot`**(*vm*, *name*, *key='uuid'*)

Rollback snapshot of a vm

vm [string] vm to be targeted

name [string]

snapshot name The snapname must be 64 characters or less and must only contain alphanumeric characters and characters in the set [-_.:%] to comply with ZFS restrictions.

key [string [uuid|alias|hostname]] value type of 'vm' parameter

CLI Example:

```
salt '*' vmadm.rollback_snapshot 186da9ab-7392-4f55-91a5-b8f1fe770543 baseline
salt '*' vmadm.rollback_snapshot nacl baseline key=alias
```

`salt.modules.smartos_vmadm.`**`send`**(*vm*, *target*, *key='uuid'*)

Send a vm to a directory

vm [string] vm to be sent

target [string] target directory

key [string [uuid|alias|hostname]] value type of 'vm' parameter

CLI Example:

```
salt '*' vmadm.send 186da9ab-7392-4f55-91a5-b8f1fe770543 /opt/backups
salt '*' vmadm.send vm=nacl target=/opt/backups key=alias
```

`salt.modules.smartos_vmadm.`**`start`**(*vm*, *options=None*, *key='uuid'*)

Start a vm

vm [string] vm to be started

options [string] optional additional options

key [string [uuid|alias|hostname]] value type of 'vm' parameter

CLI Example:

```
salt '*' vmadm.start 186da9ab-7392-4f55-91a5-b8f1fe770543
salt '*' vmadm.start 186da9ab-7392-4f55-91a5-b8f1fe770543 'order=c,once=d cdrom=/
↪path/to/image.iso,ide'
salt '*' vmadm.start vm=nacl key=alias
salt '*' vmadm.start vm=nina.example.org key=hostname
```

`salt.modules.smartos_vmadm.`**`stop`**(*vm*, *force=False*, *key='uuid'*)

Stop a vm

vm [string] vm to be stopped

force [boolean] force stop of vm if true

key [string [uuid|alias|hostname]] value type of 'vm' parameter

CLI Example:

```
salt '*' vmadm.stop 186da9ab-7392-4f55-91a5-b8f1fe770543
salt '*' vmadm.stop 186da9ab-7392-4f55-91a5-b8f1fe770543 True
salt '*' vmadm.stop vm=nacl key=alias
salt '*' vmadm.stop vm=nina.example.org key=hostname
```

`salt.modules.smartos_vmadm.`**`sysrq`**(*vm*, *action='nmi'*, *key='uuid'*)

Send non-maskable interrupt to vm or capture a screenshot

vm [string] vm to be targeted

action [string] nmi or screenshot – Default: nmi
key [string [uuid|alias|hostname]] value type of 'vm' parameter
CLI Example:

```
salt '*' vmadm.sysrq 186da9ab-7392-4f55-91a5-b8f1fe770543 nmi
salt '*' vmadm.sysrq 186da9ab-7392-4f55-91a5-b8f1fe770543 screenshot
salt '*' vmadm.sysrq nacl nmi key=alias
```

`salt.modules.smartos_vmadm.`**`update`**(*vm, from_file=None, key='uuid', **kwargs*)
Update a new vm
vm [string] vm to be updated
from_file [string] json file to update the vm with – if present, all other options will be ignored
key [string [uuid|alias|hostname]] value type of 'vm' parameter
kwargs [string|int|...] options to update for the vm
CLI Example:

```
salt '*' vmadm.update vm=186da9ab-7392-4f55-91a5-b8f1fe770543 from_file=/tmp/new_
↪vm.json
salt '*' vmadm.update vm=nacl key=alias from_file=/tmp/new_vm.json
salt '*' vmadm.update vm=186da9ab-7392-4f55-91a5-b8f1fe770543 max_physical_
↪memory=1024
```

21.9.389 salt.modules.smbios

Interface to SMBIOS/DMI

(Parsing through dmidecode)

External References

Desktop Management Interface (DMI)
System Management BIOS
DMIdecode

`salt.modules.smbios.`**`get`**(*string, clean=True*)
Get an individual DMI string from SMBIOS info
string
The string to fetch. DMIdecode supports:

- `bios-vendor`
- `bios-version`
- `bios-release-date`
- `system-manufacturer`
- `system-product-name`
- `system-version`
- `system-serial-number`
- `system-uuid`
- `baseboard-manufacturer`
- `baseboard-product-name`
- `baseboard-version`
- `baseboard-serial-number`
- `baseboard-asset-tag`
- `chassis-manufacturer`

  - `chassis-type`
  - `chassis-version`
  - `chassis-serial-number`
  - `chassis-asset-tag`
  - `processor-family`
  - `processor-manufacturer`
  - `processor-version`
  - `processor-frequency`

clean

Don't return well-known false information

(invalid UUID's, serial 000000000's, etcetera)

Defaults to `True`

CLI Example:

```
salt '*' smbios.get system-uuid clean=False
```

`salt.modules.smbios.`**`records`**(*rec_type=None*, *fields=None*, *clean=True*)

Return DMI records from SMBIOS

type Return only records of type(s) The SMBIOS specification defines the following DMI types:

Type	Information
0	BIOS
1	System
2	Baseboard
3	Chassis
4	Processor
5	Memory Controller
6	Memory Module
7	Cache
8	Port Connector
9	System Slots
10	On Board Devices
11	OEM Strings
12	System Configuration Options
13	BIOS Language
14	Group Associations
15	System Event Log
16	Physical Memory Array
17	Memory Device
18	32-bit Memory Error
19	Memory Array Mapped Address
20	Memory Device Mapped Address
21	Built-in Pointing Device
22	Portable Battery
23	System Reset
24	Hardware Security
25	System Power Controls
26	Voltage Probe
27	Cooling Device
28	Temperature Probe
29	Electrical Current Probe

Continued on next page

Table 10 – continued from previous page

Type	Information
30	Out-of-band Remote Access
31	Boot Integrity Services
32	System Boot
33	64-bit Memory Error
34	Management Device
35	Management Device Component
36	Management Device Threshold Data
37	Memory Channel
38	IPMI Device
39	Power Supply
40	Additional Information
41	Onboard Devices Extended Information
42	Management Controller Host Interface

clean

Don't return well-known false information

(invalid UUID's, serial 000000000's, etcetera)

Defaults to `True`

CLI Example:

```
salt '*' smbios.records clean=False
salt '*' smbios.records 14
salt '*' smbios.records 4 core_count,thread_count,current_speed
```

21.9.390 salt.modules.smf_service

Service support for Solaris 10 and 11, should work with other systems that use SMF also. (e.g. SmartOS)

Important: If you feel that Salt should be using this module to manage services on a minion, and it is using a different module (or gives an error similar to *'service.start' is not available*), see here.

`salt.modules.smf_service.`**`available`**`(`*name*`)`

Returns `True` if the specified service is available, otherwise returns `False`.

We look up the name with the svcs command to get back the FMRI This allows users to use simpler service names

CLI Example:

```
salt '*' service.available net-snmp
```

`salt.modules.smf_service.`**`disable`**`(`*name*, ***kwargs*`)`

Disable the named service to start at boot

CLI Example:

```
salt '*' service.disable <service name>
```

`salt.modules.smf_service.`**`disabled`**`(`*name*`)`

Check to see if the named service is disabled to start on boot

CLI Example:

```
salt '*' service.disabled <service name>
```

`salt.modules.smf_service.`**`enable`**`(`*name*, ***kwargs*`)`

Enable the named service to start at boot

CLI Example:

```
salt '*' service.enable <service name>
```

`salt.modules.smf_service.`**`enabled`**`(`*name*, ***kwargs*`)`

Check to see if the named service is enabled to start on boot

CLI Example:

```
salt '*' service.enabled <service name>
```

`salt.modules.smf_service.`**`get_all`**`()`

Return all installed services

CLI Example:

```
salt '*' service.get_all
```

`salt.modules.smf_service.`**`get_disabled`**`()`

Return the disabled services

CLI Example:

```
salt '*' service.get_disabled
```

`salt.modules.smf_service.`**`get_enabled`**`()`

Return the enabled services

CLI Example:

```
salt '*' service.get_enabled
```

`salt.modules.smf_service.`**`get_running`**`()`

Return the running services

CLI Example:

```
salt '*' service.get_running
```

`salt.modules.smf_service.`**`get_stopped`**`()`

Return the stopped services

CLI Example:

```
salt '*' service.get_stopped
```

`salt.modules.smf_service.`**`missing`**`(`*name*`)`

The inverse of service.available. Returns `True` if the specified service is not available, otherwise returns `False`.

CLI Example:

```
salt '*' service.missing net-snmp
```

`salt.modules.smf_service.`**`reload_`**`(`*name*`)`

Reload the named service

CLI Example:

```
salt '*' service.reload <service name>
```

`salt.modules.smf_service.`**`restart`**`(`*name*`)`

Restart the named service

CLI Example:

```
salt '*' service.restart <service name>
```

`salt.modules.smf_service.`**`start`**`(`*name*`)`

Start the specified service

CLI Example:

```
salt '*' service.start <service name>
```

`salt.modules.smf_service.`**`status`**`(`*name, sig=None*`)`

Return the status for a service. If the name contains globbing, a dict mapping service name to True/False values is returned.

Changed in version 2018.3.0: The service name can now be a glob (e.g. `salt*`)

Parameters

- **name** (*str*) – The name of the service to check
- **sig** (*str*) – Not implemented

Returns True if running, False otherwise dict: Maps service name to True if running, False otherwise

Return type bool

CLI Example:

```
salt '*' service.status <service name>
```

`salt.modules.smf_service.`**`stop`**`(`*name*`)`

Stop the specified service

CLI Example:

```
salt '*' service.stop <service name>
```

21.9.391 salt.modules.smtp

Module for Sending Messages via SMTP

New in version 2014.7.0.

depends

- smtplib python module

configuration This module can be used by either passing a jid and password directly to send_message, or by specifying the name of a configuration profile in the minion config, minion pillar, or master config.

For example:

```
my-smtp-login:
    smtp.server: smtp.domain.com
    smtp.tls: True
    smtp.sender: admin@domain.com
    smtp.username: myuser
    smtp.password: verybadpass
```

The resourcename refers to the resource that is using this account. It is user-definable, and optional. The following configurations are both valid:

```
my-smtp-login:
    smtp.server: smtp.domain.com
    smtp.tls: True
    smtp.sender: admin@domain.com
    smtp.username: myuser
    smtp.password: verybadpass

another-smtp-login:
    smtp.server: smtp.domain.com
    smtp.tls: True
    smtp.sender: admin@domain.com
    smtp.username: myuser
    smtp.password: verybadpass
```

`salt.modules.smtp.`**`send_msg`**(*recipient, message, subject='Message from Salt', sender=None, server=None, use_ssl='True', username=None, password=None, profile=None, attachments=None*)

Send a message to an SMTP recipient. Designed for use in states.

CLI Examples:

```
salt '*' smtp.send_msg 'admin@example.com' 'This is a salt module test' profile=
↪'my-smtp-account'
salt '*' smtp.send_msg 'admin@example.com' 'This is a salt module test' username=
↪'myuser' password='verybadpass' sender='admin@example.com' server='smtp.domain.
↪com'
salt '*' smtp.send_msg 'admin@example.com' 'This is a salt module test' username=
↪'myuser' password='verybadpass' sender='admin@example.com' server='smtp.domain.
↪com' attachments="['/var/log/messages']"
```

21.9.392 salt.modules.solaris_fmadm

Module for running fmadm and fmdump on Solaris

maintainer Jorge Schrauwen <sjorge@blackdot.be>

maturity new

platform solaris,illumos

New in version 2016.3.0.

`salt.modules.solaris_fmadm.`**`acquit`**(*fmri*)

Acquit resource or acquit case
fmri: string fmri or uuid
CLI Example:

```
salt '*' fmadm.acquit fmri | uuid
```

salt.modules.solaris_fmadm.**config**()

Display fault manager configuration

CLI Example:

```
salt '*' fmadm.config
```

salt.modules.solaris_fmadm.**faulty**()

Display list of faulty resources

CLI Example:

```
salt '*' fmadm.faulty
```

salt.modules.solaris_fmadm.**flush**(*fmri*)

Flush cached state for resource

fmri: **string** fmri

CLI Example:

```
salt '*' fmadm.flush fmri
```

salt.modules.solaris_fmadm.**healthy**()

Return whether fmadm is reporting faults

CLI Example:

```
salt '*' fmadm.healthy
```

salt.modules.solaris_fmadm.**list_records**(*after=None*, *before=None*)

Display fault management logs

after [string] filter events after time, see man fmdump for format

before [string] filter events before time, see man fmdump for format

CLI Example:

```
salt '*' fmadm.list
```

salt.modules.solaris_fmadm.**load**(*path*)

Load specified fault manager module

path: **string** path of fault manager module

CLI Example:

```
salt '*' fmadm.load /module/path
```

salt.modules.solaris_fmadm.**repaired**(*fmri*)

Notify fault manager that resource has been repaired

fmri: **string** fmri

CLI Example:

```
salt '*' fmadm.repaired fmri
```

salt.modules.solaris_fmadm.**replaced**(*fmri*)

Notify fault manager that resource has been replaced

fmri: **string** fmri

CLI Example:

```
salt '*' fmadm.repaired fmri
```

`salt.modules.solaris_fmadm.`**`reset`**(*module*, *serd=None*)

Reset module or sub-component

module: **string** module to unload

serd [string] serd sub module

CLI Example:

```
salt '*' fmadm.reset software-response
```

`salt.modules.solaris_fmadm.`**`show`**(*uuid*)

Display log details

uuid: **string** uuid of fault

CLI Example:

```
salt '*' fmadm.show 11b4070f-4358-62fa-9e1e-998f485977e1
```

`salt.modules.solaris_fmadm.`**`unload`**(*module*)

Unload specified fault manager module

module: **string** module to unload

CLI Example:

```
salt '*' fmadm.unload software-response
```

21.9.393 salt.modules.solaris_group

Manage groups on Solaris

Important: If you feel that Salt should be using this module to manage groups on a minion, and it is using a different module (or gives an error similar to *'group.info' is not available*), see here.

`salt.modules.solaris_group.`**`add`**(*name*, *gid=None*, ***kwargs*)

Add the specified group

CLI Example:

```
salt '*' group.add foo 3456
```

`salt.modules.solaris_group.`**`chgid`**(*name*, *gid*)

Change the gid for a named group

CLI Example:

```
salt '*' group.chgid foo 4376
```

`salt.modules.solaris_group.`**`delete`**(*name*)

Remove the named group

CLI Example:

```
salt '*' group.delete foo
```

`salt.modules.solaris_group.`**`getent`**(*refresh=False*)

Return info on all groups

CLI Example:

```
salt '*' group.getent
```

`salt.modules.solaris_group.`**`info`**`(`*name*`)`

Return information about a group

CLI Example:

```
salt '*' group.info foo
```

21.9.394 salt.modules.solaris_shadow

Manage the password database on Solaris systems

Important: If you feel that Salt should be using this module to manage passwords on a minion, and it is using a different module (or gives an error similar to *'shadow.info' is not available*), see *here*.

`salt.modules.solaris_shadow.`**`default_hash`**`()`

Returns the default hash used for unset passwords

CLI Example:

```
salt '*' shadow.default_hash
```

`salt.modules.solaris_shadow.`**`del_password`**`(`*name*`)`

New in version 2015.8.8.

Delete the password from name user

CLI Example:

```
salt '*' shadow.del_password username
```

`salt.modules.solaris_shadow.`**`gen_password`**`(`*password, crypt_salt=None, algorithm='sha512'*`)`

New in version 2015.8.8.

Generate hashed password

Note: When called this function is called directly via remote-execution, the password argument may be displayed in the system's process list. This may be a security risk on certain systems.

---

password Plaintext password to be hashed.

crypt_salt Crpytographic salt. If not given, a random 8-character salt will be generated.

algorithm The following hash algorithms are supported:

- md5
- blowfish (not in mainline glibc, only available in distros that add it)
- sha256
- sha512 (default)

CLI Example:

```
salt '*' shadow.gen_password 'I_am_password'
salt '*' shadow.gen_password 'I_am_password' crypt_salt='I_am_salt'␣
↪algorithm=sha256
```

`salt.modules.solaris_shadow.`**`info`**`(`*name*`)`

Return information for the specified user

CLI Example:

```
salt '*' shadow.info root
```

`salt.modules.solaris_shadow.`**`set_maxdays`**`(`*name*, *maxdays*`)`

Set the maximum number of days during which a password is valid. See man passwd.

CLI Example:

```
salt '*' shadow.set_maxdays username 90
```

`salt.modules.solaris_shadow.`**`set_mindays`**`(`*name*, *mindays*`)`

Set the minimum number of days between password changes. See man passwd.

CLI Example:

```
salt '*' shadow.set_mindays username 7
```

`salt.modules.solaris_shadow.`**`set_password`**`(`*name*, *password*`)`

Set the password for a named user. The password must be a properly defined hash, the password hash can be generated with this command: `openssl passwd -1 <plaintext password>`

CLI Example:

```
salt '*' shadow.set_password root $1$UYCIxa628.9qXjpQCjM4a..
```

`salt.modules.solaris_shadow.`**`set_warndays`**`(`*name*, *warndays*`)`

Set the number of days of warning before a password change is required. See man passwd.

CLI Example:

```
salt '*' shadow.set_warndays username 7
```

21.9.395 salt.modules.solaris_system

Support for reboot, shutdown, etc

This module is assumes we are using solaris-like shutdown

New in version 2016.3.0.

`salt.modules.solaris_system.`**`halt`**`()`

Halt a running system

CLI Example:

```
salt '*' system.halt
```

`salt.modules.solaris_system.`**`init`**`(`*state*`)`

Change the system runlevel on sysV compatible systems

CLI Example:

state [string] Init state

```
salt '*' system.init 3
```

`salt.modules.solaris_system.`**`poweroff`**`()`

Poweroff a running system

CLI Example:

```
salt '*' system.poweroff
```

`salt.modules.solaris_system.`**`reboot`**`(`*delay=0, message=None*`)`

Reboot the system

delay [int] Optional wait time in seconds before the system will be rebooted.

message [string] Optional message to broadcast before rebooting.

CLI Example:

```
salt '*' system.reboot
salt '*' system.reboot 60 "=== system upgraded ==="
```

`salt.modules.solaris_system.`**`shutdown`**`(`*delay=0, message=None*`)`

Shutdown a running system

delay [int] Optional wait time in seconds before the system will be shutdown.

message [string] Optional message to broadcast before rebooting.

CLI Example:

```
salt '*' system.shutdown
salt '*' system.shutdown 60 "=== disk replacement ==="
```

21.9.396 salt.modules.solaris_user

Manage users with the useradd command

Important: If you feel that Salt should be using this module to manage users on a minion, and it is using a different module (or gives an error similar to *'user.info' is not available*), see *here*.

`salt.modules.solaris_user.`**`add`**`(`*name, uid=None, gid=None, groups=None, home=None, shell=None, unique=True, fullname='', roomnumber='', workphone='', homephone='', createhome=True, **kwargs*`)`

Add a user to the minion

CLI Example:

```
salt '*' user.add name <uid> <gid> <groups> <home> <shell>
```

`salt.modules.solaris_user.`**`chfullname`**`(`*name, fullname*`)`

Change the user's Full Name

CLI Example:

```
salt '*' user.chfullname foo "Foo Bar"
```

`salt.modules.solaris_user.`**`chgid`**`(`*name, gid*`)`

Change the default group of the user

CLI Example:

```
salt '*' user.chgid foo 4376
```

`salt.modules.solaris_user.`**`chgroups`**(*name*, *groups*, *append=False*)
Change the groups to which a user belongs

name Username to modify

groups List of groups to set for the user. Can be passed as a comma-separated list or a Python list.

append [False] Set to `True` to append these groups to the user's existing list of groups. Otherwise, the specified groups will replace any existing groups for the user.

CLI Example:

```
salt '*' user.chgroups foo wheel,root True
```

`salt.modules.solaris_user.`**`chhome`**(*name*, *home*, *persist=False*)
Set a new home directory for an existing user

name Username to modify

home New home directory to set

persist [False] Set to `True` to prevent configuration files in the new home directory from being overwritten by the files from the skeleton directory.

CLI Example:

```
salt '*' user.chhome foo /home/users/foo True
```

`salt.modules.solaris_user.`**`chhomephone`**(*name*, *homephone*)
Change the user's Home Phone

CLI Example:

```
salt '*' user.chhomephone foo "7735551234"
```

`salt.modules.solaris_user.`**`chroomnumber`**(*name*, *roomnumber*)
Change the user's Room Number

CLI Example:

```
salt '*' user.chroomnumber foo 123
```

`salt.modules.solaris_user.`**`chshell`**(*name*, *shell*)
Change the default shell of the user

CLI Example:

```
salt '*' user.chshell foo /bin/zsh
```

`salt.modules.solaris_user.`**`chuid`**(*name*, *uid*)
Change the uid for a named user

CLI Example:

```
salt '*' user.chuid foo 4376
```

`salt.modules.solaris_user.`**`chworkphone`**(*name*, *workphone*)
Change the user's Work Phone

CLI Example:

```
salt '*' user.chworkphone foo "7735550123"
```

`salt.modules.solaris_user.`**`delete`**(*name*, *remove=False*, *force=False*)
Remove a user from the minion

CLI Example:

```
salt '*' user.delete name remove=True force=True
```

`salt.modules.solaris_user.`**`getent`**(*refresh=False*)

Return the list of all info for all users

CLI Example:

```
salt '*' user.getent
```

`salt.modules.solaris_user.`**`info`**(*name*)

Return user information

CLI Example:

```
salt '*' user.info root
```

`salt.modules.solaris_user.`**`list_groups`**(*name*)

Return a list of groups the named user belongs to

CLI Example:

```
salt '*' user.list_groups foo
```

`salt.modules.solaris_user.`**`list_users`**()

Return a list of all users

CLI Example:

```
salt '*' user.list_users
```

`salt.modules.solaris_user.`**`rename`**(*name, new_name*)

Change the username for a named user

CLI Example:

```
salt '*' user.rename name new_name
```

21.9.397 salt.modules.solarisipspkg

IPS pkg support for Solaris

Important: If you feel that Salt should be using this module to manage packages on a minion, and it is using a different module (or gives an error similar to *'pkg.install' is not available*), see *here*.

This module provides support for Solaris 11 new package management - IPS (Image Packaging System). This is the default pkg module for Solaris 11 (and later).

If you want to use also other packaging module (e.g. pkgutil) together with IPS, you need to override the `pkg` provider in sls for each package:

```
mypackage:
  pkg.installed:
    - provider: pkgutil
```

Or you can override it globally by setting the *`providers`* parameter in your Minion config file like this:

```
providers:
  pkg: pkgutil
```

Or you can override it globally by setting the `providers` parameter in your Minion config file like this:

```
providers:
  pkg: pkgutil
```

`salt.modules.solarisipspkg.`**`available_version`**`(`**names*, ***kwargs*`)`

This function is an alias of `latest_version`.

The available version of packages in the repository. Accepts full or partial FMRI. Partial FMRI is returned if the full FMRI could not be resolved.

If the latest version of a given package is already installed, an empty string will be returned for that package.

Please use pkg.latest_version as pkg.available_version is being deprecated.

Changed in version 2019.2.0: Support for multiple package names added.

CLI Example:

```
salt '*' pkg.latest_version bash
salt '*' pkg.latest_version pkg://solaris/entire
salt '*' pkg.latest_version postfix sendmail
```

`salt.modules.solarisipspkg.`**`get_fmri`**`(`*name*, ***kwargs*`)`

Returns FMRI from partial name. Returns empty string (") if not found. In case of multiple match, the function returns list of all matched packages.

CLI Example:

```
salt '*' pkg.get_fmri bash
```

`salt.modules.solarisipspkg.`**`install`**`(`*name=None*, *refresh=False*, *pkgs=None*, *version=None*, *test=False*, ***kwargs*`)`

Install the named package using the IPS pkg command. Accepts full or partial FMRI.

Returns a dict containing the new package names and versions:

```
{'<package>': {'old': '<old-version>',
               'new': '<new-version>'}}
```

Multiple Package Installation Options:

pkgs A list of packages to install. Must be passed as a python list.

CLI Example:

```
salt '*' pkg.install vim
salt '*' pkg.install pkg://solaris/editor/vim
salt '*' pkg.install pkg://solaris/editor/vim refresh=True
salt '*' pkg.install pkgs='["foo", "bar"]'
```

`salt.modules.solarisipspkg.`**`is_installed`**`(`*name*, ***kwargs*`)`

Returns True if the package is installed. Otherwise returns False. Name can be full or partial FMRI. In case of multiple match from partial FMRI name, it returns True.

CLI Example:

```
salt '*' pkg.is_installed bash
```

`salt.modules.solarisipspkg.`**`latest_version`**`(`**names, **kwargs*`)`

The available version of packages in the repository. Accepts full or partial FMRI. Partial FMRI is returned if the full FMRI could not be resolved.

If the latest version of a given package is already installed, an empty string will be returned for that package.

Please use pkg.latest_version as pkg.available_version is being deprecated.

Changed in version 2019.2.0: Support for multiple package names added.

CLI Example:

```
salt '*' pkg.latest_version bash
salt '*' pkg.latest_version pkg://solaris/entire
salt '*' pkg.latest_version postfix sendmail
```

`salt.modules.solarisipspkg.`**`list_pkgs`**`(`*versions_as_list=False, **kwargs*`)`

List the currently installed packages as a dict:

```
{'<package_name>': '<version>'}
```

CLI Example:

```
salt '*' pkg.list_pkgs
```

`salt.modules.solarisipspkg.`**`list_upgrades`**`(`*refresh=True, **kwargs*`)`

Lists all packages available for update.

When run in global zone, it reports only upgradable packages for the global zone.

When run in non-global zone, it can report more upgradable packages than `pkg update -vn`, because `pkg update` hides packages that require newer version of `pkg://solaris/entire` (which means that they can be upgraded only from the global zone). If `pkg://solaris/entire` is found in the list of upgrades, then the global zone should be updated to get all possible updates. Use `refresh=True` to refresh the package database.

refresh [True] Runs a full package database refresh before listing. Set to `False` to disable running the refresh.

> Changed in version 2017.7.0.
>
> In previous versions of Salt, `refresh` defaulted to `False`. This was changed to default to `True` in the 2017.7.0 release to make the behavior more consistent with the other package modules, which all default to `True`.

CLI Example:

```
salt '*' pkg.list_upgrades
salt '*' pkg.list_upgrades refresh=False
```

`salt.modules.solarisipspkg.`**`normalize_name`**`(`*name, **kwargs*`)`

Internal function. Normalizes pkg name to full FMRI before running pkg.install. In case of multiple matches or no match, it returns the name without modifications.

CLI Example:

```
salt '*' pkg.normalize_name vim
```

`salt.modules.solarisipspkg.`**`purge`**`(`*name, **kwargs*`)`

Remove specified package. Accepts full or partial FMRI.

Returns a list containing the removed packages.

CLI Example:

```
salt '*' pkg.purge <package name>
```

`salt.modules.solarisipspkg.`**`refresh_db`**`(`*full=False*`)`

Updates the remote repos database.

full : False
Set to `True` to force a refresh of the pkg DB from all publishers, regardless of the last refresh time.

CLI Example:

```
salt '*' pkg.refresh_db
salt '*' pkg.refresh_db full=True
```

`salt.modules.solarisipspkg.`**`remove`**`(`*name=None, pkgs=None, **kwargs*`)`

Remove specified package. Accepts full or partial FMRI. In case of multiple match, the command fails and won't modify the OS.

name The name of the package to be deleted.

Multiple Package Options:

pkgs A list of packages to delete. Must be passed as a python list. The `name` parameter will be ignored if this option is passed.

Returns a list containing the removed packages.

CLI Example:

```
salt '*' pkg.remove <package name>
salt '*' pkg.remove tcsh
salt '*' pkg.remove pkg://solaris/shell/tcsh
salt '*' pkg.remove pkgs='["foo", "bar"]'
```

`salt.modules.solarisipspkg.`**`search`**`(`*name, versions_as_list=False, **kwargs*`)`

Searches the repository for given pkg name. The name can be full or partial FMRI. All matches are printed. Globs are also supported.

CLI Example:

```
salt '*' pkg.search bash
```

`salt.modules.solarisipspkg.`**`upgrade`**`(`*refresh=False, **kwargs*`)`

Upgrade all packages to the latest possible version. When run in global zone, it updates also all non-global zones. In non-global zones upgrade is limited by dependency constrains linked to the version of pkg://solaris/entire.

Returns a dictionary containing the changes:

```
{'<package>':  {'old': '<old-version>',
                'new': '<new-version>'}}
```

When there is a failure, an explanation is also included in the error message, based on the return code of the `pkg update` command.

CLI Example:

```
salt '*' pkg.upgrade
```

`salt.modules.solarisipspkg.`**`upgrade_available`**(*name*)

Check if there is an upgrade available for a certain package Accepts full or partial FMRI. Returns all matches found.

CLI Example:

```
salt '*' pkg.upgrade_available apache-22
```

`salt.modules.solarisipspkg.`**`version`**(**names*, ***kwargs*)

Common interface for obtaining the version of installed packages. Accepts full or partial FMRI. If called using pkg_resource, full FMRI is required. Partial FMRI is returned if the package is not installed.

CLI Example:

```
salt '*' pkg.version vim
salt '*' pkg.version foo bar baz
salt '*' pkg_resource.version pkg://solaris/entire
```

21.9.398 salt.modules.solarispkg

Package support for Solaris

Important: If you feel that Salt should be using this module to manage packages on a minion, and it is using a different module (or gives an error similar to *'pkg.install' is not available*), see *here*.

`salt.modules.solarispkg.`**`install`**(*name=None*, *sources=None*, *saltenv='base'*, ***kwargs*)

Install the passed package. Can install packages from the following sources:

- Locally (package already exists on the minion
- HTTP/HTTPS server
- FTP server
- Salt master

Returns a dict containing the new package names and versions:

```
{'<package>': {'old': '<old-version>',
               'new': '<new-version>'}}
```

CLI Examples:

```
# Installing a data stream pkg that already exists on the minion

salt '*' pkg.install sources='[{"<pkg name>": "/dir/on/minion/<pkg filename>"}]'
salt '*' pkg.install sources='[{"SMClgcc346": "/var/spool/pkg/gcc-3.4.6-sol10-
↪sparc-local.pkg"}]'

# Installing a data stream pkg that exists on the salt master

salt '*' pkg.install sources='[{"<pkg name>": "salt://pkgs/<pkg filename>"}]'
salt '*' pkg.install sources='[{"SMClgcc346": "salt://pkgs/gcc-3.4.6-sol10-sparc-
↪local.pkg"}]'
```

CLI Example:

```
# Installing a data stream pkg that exists on a HTTP server
salt '*' pkg.install sources='[{"<pkg name>": "http://packages.server.com/<pkg␣
↪filename>"}]'
salt '*' pkg.install sources='[{"SMClgcc346": "http://packages.server.com/gcc-3.4.
↪6-sol10-sparc-local.pkg"}]'
```

If working with solaris zones and you want to install a package only in the global zone you can pass 'current_zone_only=True' to salt to have the package only installed in the global zone. (Behind the scenes this is passing '-G' to the pkgadd command.) Solaris default when installing a package in the global zone is to install it in all zones. This overrides that and installs the package only in the global.

CLI Example:

```
# Installing a data stream package only in the global zone:
salt 'global_zone' pkg.install sources='[{"SMClgcc346": "/var/spool/pkg/gcc-3.4.6-
↪sol10-sparc-local.pkg"}]' current_zone_only=True
```

By default salt automatically provides an adminfile, to automate package installation, with these options set:

```
email=
instance=quit
partial=nocheck
runlevel=nocheck
idepend=nocheck
rdepend=nocheck
space=nocheck
setuid=nocheck
conflict=nocheck
action=nocheck
basedir=default
```

You can override any of these options in two ways. First you can optionally pass any of the options as a kwarg to the module/state to override the default value or you can optionally pass the 'admin_source' option providing your own adminfile to the minions.

Note: You can find all of the possible options to provide to the adminfile by reading the admin man page:

```
man -s 4 admin
```

CLI Example:

```
# Overriding the 'instance' adminfile option when calling the module directly
salt '*' pkg.install sources='[{"<pkg name>": "salt://pkgs/<pkg filename>"}]'␣
↪instance="overwrite"
```

SLS Example:

```
# Overriding the 'instance' adminfile option when used in a state

SMClgcc346:
  pkg.installed:
    - sources:
      - SMClgcc346: salt://srv/salt/pkgs/gcc-3.4.6-sol10-sparc-local.pkg
    - instance: overwrite
```

Note: The ID declaration is ignored, as the package name is read from the `sources` parameter.

CLI Example:

```
# Providing your own adminfile when calling the module directly

salt '*' pkg.install sources='[{"<pkg name>": "salt://pkgs/<pkg filename>"}]'
↪admin_source='salt://pkgs/<adminfile filename>'

# Providing your own adminfile when using states

<pkg name>:
  pkg.installed:
    - sources:
      - <pkg name>: salt://pkgs/<pkg filename>
    - admin_source: salt://pkgs/<adminfile filename>
```

Note: The ID declaration is ignored, as the package name is read from the `sources` parameter.

`salt.modules.solarispkg.`**`latest_version`**(*names, **kwargs)

Return the latest version of the named package available for upgrade or installation. If more than one package name is specified, a dict of name/version pairs is returned.

If the latest version of a given package is already installed, an empty string will be returned for that package.

CLI Example:

```
salt '*' pkg.latest_version <package name>
salt '*' pkg.latest_version <package1> <package2> <package3> ...
```

NOTE: As package repositories are not presently supported for Solaris pkgadd, this function will always return an empty string for a given package.

`salt.modules.solarispkg.`**`list_pkgs`**(*versions_as_list=False, **kwargs*)

List the packages currently installed as a dict:

```
{'<package_name>': '<version>'}
```

CLI Example:

```
salt '*' pkg.list_pkgs
```

`salt.modules.solarispkg.`**`purge`**(*name=None, pkgs=None, **kwargs*)

Package purges are not supported, this function is identical to `remove()`.

name The name of the package to be deleted

Multiple Package Options:

pkgs A list of packages to delete. Must be passed as a python list. The `name` parameter will be ignored if this option is passed.

New in version 0.16.0.

Returns a dict containing the changes.

CLI Example:

```
salt '*' pkg.purge <package name>
salt '*' pkg.purge <package1>,<package2>,<package3>
salt '*' pkg.purge pkgs='["foo", "bar"]'
```

`salt.modules.solarispkg.`**`remove`**(*name=None, pkgs=None, saltenv='base', **kwargs*)

Remove packages with pkgrm

name The name of the package to be deleted

By default salt automatically provides an adminfile, to automate package removal, with these options set:

```
email=
instance=quit
partial=nocheck
runlevel=nocheck
idepend=nocheck
rdepend=nocheck
space=nocheck
setuid=nocheck
conflict=nocheck
action=nocheck
basedir=default
```

You can override any of these options in two ways. First you can optionally pass any of the options as a kwarg to the module/state to override the default value or you can optionally pass the 'admin_source' option providing your own adminfile to the minions.

Note: You can find all of the possible options to provide to the adminfile by reading the admin man page:

```
man -s 4 admin
```

Multiple Package Options:

pkgs A list of packages to delete. Must be passed as a python list. The `name` parameter will be ignored if this option is passed.

New in version 0.16.0.

Returns a dict containing the changes.

CLI Example:

```
salt '*' pkg.remove <package name>
salt '*' pkg.remove SUNWgit
salt '*' pkg.remove <package1>,<package2>,<package3>
salt '*' pkg.remove pkgs='["foo", "bar"]'
```

`salt.modules.solarispkg.`**`upgrade_available`**`(`*name*`)`

Check whether or not an upgrade is available for a given package

CLI Example:

```
salt '*' pkg.upgrade_available <package name>
```

`salt.modules.solarispkg.`**`version`**`(`**names, **kwargs*`)`

Returns a string representing the package version or an empty string if not installed. If more than one package name is specified, a dict of name/version pairs is returned.

CLI Example:

```
salt '*' pkg.version <package name>
salt '*' pkg.version <package1> <package2> <package3> ...
```

21.9.399 salt.modules.solr

Apache Solr Salt Module

Author: Jed Glazner Version: 0.2.1 Modified: 12/09/2011

This module uses HTTP requests to talk to the apache solr request handlers to gather information and report errors. Because of this the minion doesn't necessarily need to reside on the actual slave. However if you want to use the signal function the minion must reside on the physical solr host.

This module supports multi-core and standard setups. Certain methods are master/slave specific. Make sure you set the solr.type. If you have questions or want a feature request please ask.

Coming Features in 0.3

1. Add command for checking for replication failures on slaves
2. Improve match_index_versions since it's pointless on busy solr masters
3. Add additional local fs checks for backups to make sure they succeeded

Override these in the minion config

solr.cores A list of core names e.g. ['core1','core2']. An empty list indicates non-multicore setup.

solr.baseurl The root level URL to access solr via HTTP

solr.request_timeout The number of seconds before timing out an HTTP/HTTPS/FTP request. If nothing is specified then the python global timeout setting is used.

solr.type Possible values are 'master' or 'slave'

solr.backup_path The path to store your backups. If you are using cores and you can specify to append the core name to the path in the backup method.

solr.num_backups For versions of solr >= 3.5. Indicates the number of backups to keep. This option is ignored if your version is less.

solr.init_script The full path to your init script with start/stop options

solr.dih.options A list of options to pass to the DIH.

Required Options for DIH

clean [False] Clear the index before importing

commit [True] Commit the documents to the index upon completion

optimize [True] Optimize the index after commit is complete

verbose [True] Get verbose output

`salt.modules.solr.`**`abort_import`**(*handler*, *host=None*, *core_name=None*, *verbose=False*)

MASTER ONLY Aborts an existing import command to the specified handler. This command can only be run if the minion is configured with solr.type=master

handler [str] The name of the data import handler.
host [str (None)] The solr host to query. __opts__['host'] is default.
core [str (None)] The core the handler belongs to.
verbose [boolean (False)] Run the command with verbose output.
Return : dict<str,obj>:

```
{'success':boolean, 'data':dict, 'errors':list, 'warnings':list}
```

CLI Example:

```
salt '*' solr.abort_import dataimport None music {'clean':True}
```

salt.modules.solr.**backup**(*host=None*, *core_name=None*, *append_core_to_path=False*)

Tell solr make a backup. This method can be mis-leading since it uses the backup API. If an error happens during the backup you are not notified. The status: 'OK' in the response simply means that solr received the request successfully.

host [str (None)] The solr host to query. __opts__['host'] is default.

core_name [str (None)] The name of the solr core if using cores. Leave this blank if you are not using cores or if you want to check all cores.

append_core_to_path [boolean (False)] If True add the name of the core to the backup path. Assumes that minion backup path is not None.

Return : dict<str,obj>:

```
{'success':boolean, 'data':dict, 'errors':list, 'warnings':list}
```

CLI Example:

```
salt '*' solr.backup music
```

salt.modules.solr.**core_status**(*host=None*, *core_name=None*)

MULTI-CORE HOSTS ONLY Get the status for a given core or all cores if no core is specified

host [str (None)] The solr host to query. __opts__['host'] is default.

core_name [str] The name of the core to reload

Return : dict<str,obj>:

```
{'success':boolean, 'data':dict, 'errors':list, 'warnings':list}
```

CLI Example:

```
salt '*' solr.core_status None music
```

salt.modules.solr.**delta_import**(*handler*, *host=None*, *core_name=None*, *options=None*, *extra=None*)

Submits an import command to the specified handler using specified options. This command can only be run if the minion is configured with solr.type=master

handler [str] The name of the data import handler.

host [str (None)] The solr host to query. __opts__['host'] is default.

core [str (None)] The core the handler belongs to.

options [dict (__opts__)] A list of options such as clean, optimize commit, verbose, and pause_replication. leave blank to use __opts__ defaults. options will be merged with __opts__

extra [dict ([])] Extra name value pairs to pass to the handler. e.g. ["name=value"]

Return : dict<str,obj>:

```
{'success':boolean, 'data':dict, 'errors':list, 'warnings':list}
```

CLI Example:

```
salt '*' solr.delta_import dataimport None music {'clean':True}
```

salt.modules.solr.**full_import**(*handler*, *host=None*, *core_name=None*, *options=None*, *extra=None*)

MASTER ONLY Submits an import command to the specified handler using specified options. This command can only be run if the minion is configured with solr.type=master

handler [str] The name of the data import handler.

host [str (None)] The solr host to query. __opts__['host'] is default.

core [str (None)] The core the handler belongs to.

options [dict (__opts__)] A list of options such as clean, optimize commit, verbose, and pause_replication. leave blank to use __opts__ defaults. options will be merged with __opts__

extra [dict ([])] Extra name value pairs to pass to the handler. e.g. ["name=value"]

Return : dict<str,obj>:

```
{'success':boolean, 'data':dict, 'errors':list, 'warnings':list}
```

CLI Example:

```
salt '*' solr.full_import dataimport None music {'clean':True}
```

salt.modules.solr.**import_status**(*handler, host=None, core_name=None, verbose=False*)

Submits an import command to the specified handler using specified options. This command can only be run if the minion is configured with solr.type: 'master'

handler [str] The name of the data import handler.

host [str (None)] The solr host to query. __opts__['host'] is default.

core [str (None)] The core the handler belongs to.

verbose [boolean (False)] Specifies verbose output

Return : dict<str,obj>:

```
{'success':boolean, 'data':dict, 'errors':list, 'warnings':list}
```

CLI Example:

```
salt '*' solr.import_status dataimport None music False
```

salt.modules.solr.**is_replication_enabled**(*host=None, core_name=None*)

SLAVE CALL Check for errors, and determine if a slave is replicating or not.

host [str (None)] The solr host to query. __opts__['host'] is default.

core_name [str (None)] The name of the solr core if using cores. Leave this blank if you are not using cores or if you want to check all cores.

Return : dict<str,obj>:

```
{'success':boolean, 'data':dict, 'errors':list, 'warnings':list}
```

CLI Example:

```
salt '*' solr.is_replication_enabled music
```

salt.modules.solr.**lucene_version**(*core_name=None*)

Gets the lucene version that solr is using. If you are running a multi-core setup you should specify a core name since all the cores run under the same servlet container, they will all have the same version.

core_name [str (None)] The name of the solr core if using cores. Leave this blank if you are not using cores or if you want to check all cores.

Return: dict<str,obj>:

```
{'success':boolean, 'data':dict, 'errors':list, 'warnings':list}
```

CLI Example:

```
salt '*' solr.lucene_version
```

salt.modules.solr.**match_index_versions**(*host=None, core_name=None*)

SLAVE CALL Verifies that the master and the slave versions are in sync by comparing the index version. If you are constantly pushing updates the index the master and slave versions will seldom match. A solution to this is pause indexing every so often to allow the slave to replicate and then call this method before allowing indexing to resume.

host [str (None)] The solr host to query. __opts__['host'] is default.

core_name [str (None)] The name of the solr core if using cores. Leave this blank if you are not using cores or if you want to check all cores.

Return : dict<str,obj>:

```
{'success':boolean, 'data':dict, 'errors':list, 'warnings':list}
```

CLI Example:

```
salt '*' solr.match_index_versions music
```

`salt.modules.solr.`**`optimize`**(*host=None, core_name=None*)

Search queries fast, but it is a very expensive operation. The ideal process is to run this with a master/slave configuration. Then you can optimize the master, and push the optimized index to the slaves. If you are running a single solr instance, or if you are going to run this on a slave be aware than search performance will be horrible while this command is being run. Additionally it can take a LONG time to run and your HTTP request may timeout. If that happens adjust your timeout settings.

host [str (None)] The solr host to query. __opts__['host'] is default.

core_name [str (None)] The name of the solr core if using cores. Leave this blank if you are not using cores or if you want to check all cores.

Return : dict<str,obj>:

```
{'success':boolean, 'data':dict, 'errors':list, 'warnings':list}
```

CLI Example:

```
salt '*' solr.optimize music
```

`salt.modules.solr.`**`ping`**(*host=None, core_name=None*)

Does a health check on solr, makes sure solr can talk to the indexes.

host [str (None)] The solr host to query. __opts__['host'] is default.

core_name [str (None)] The name of the solr core if using cores. Leave this blank if you are not using cores or if you want to check all cores.

Return : dict<str,obj>:

```
{'success':boolean, 'data':dict, 'errors':list, 'warnings':list}
```

CLI Example:

```
salt '*' solr.ping music
```

`salt.modules.solr.`**`reload_core`**(*host=None, core_name=None*)

MULTI-CORE HOSTS ONLY Load a new core from the same configuration as an existing registered core. While the "new" core is initializing, the "old" one will continue to accept requests. Once it has finished, all new request will go to the "new" core, and the "old" core will be unloaded.

host [str (None)] The solr host to query. __opts__['host'] is default.

core_name [str] The name of the core to reload

Return : dict<str,obj>:

```
{'success':boolean, 'data':dict, 'errors':list, 'warnings':list}
```

CLI Example:

```
salt '*' solr.reload_core None music
```

Return data is in the following format:

```
{'success':bool, 'data':dict, 'errors':list, 'warnings':list}
```

`salt.modules.solr.`**`reload_import_config`**(*handler*, *host=None*, *core_name=None*, *verbose=False*)

MASTER ONLY re-loads the handler config XML file. This command can only be run if the minion is a 'master' type

handler [str] The name of the data import handler.

host [str (None)] The solr host to query. __opts__['host'] is default.

core [str (None)] The core the handler belongs to.

verbose [boolean (False)] Run the command with verbose output.

Return : dict<str,obj>:

```
{'success':boolean, 'data':dict, 'errors':list, 'warnings':list}
```

CLI Example:

```
salt '*' solr.reload_import_config dataimport None music {'clean':True}
```

`salt.modules.solr.`**`replication_details`**(*host=None*, *core_name=None*)

Get the full replication details.

host [str (None)] The solr host to query. __opts__['host'] is default.

core_name [str (None)] The name of the solr core if using cores. Leave this blank if you are not using cores or if you want to check all cores.

Return : dict<str,obj>:

```
{'success':boolean, 'data':dict, 'errors':list, 'warnings':list}
```

CLI Example:

```
salt '*' solr.replication_details music
```

`salt.modules.solr.`**`set_is_polling`**(*polling*, *host=None*, *core_name=None*)

SLAVE CALL Prevent the slaves from polling the master for updates.

polling [boolean] True will enable polling. False will disable it.

host [str (None)] The solr host to query. __opts__['host'] is default.

core_name [str (None)] The name of the solr core if using cores. Leave this blank if you are not using cores or if you want to check all cores.

Return : dict<str,obj>:

```
{'success':boolean, 'data':dict, 'errors':list, 'warnings':list}
```

CLI Example:

```
salt '*' solr.set_is_polling False
```

`salt.modules.solr.`**`set_replication_enabled`**(*status*, *host=None*, *core_name=None*)

MASTER ONLY Sets the master to ignore poll requests from the slaves. Useful when you don't want the slaves replicating during indexing or when clearing the index.

status [boolean] Sets the replication status to the specified state.

host [str (None)] The solr host to query. __opts__['host'] is default.

core_name [str (None)] The name of the solr core if using cores. Leave this blank if you are not using cores or if you want to set the status on all cores.

Return : dict<str,obj>:

```
{'success':boolean, 'data':dict, 'errors':list, 'warnings':list}
```

CLI Example:

```
salt '*' solr.set_replication_enabled false, None, music
```

`salt.modules.solr.`**`signal`**(*signal=None*)

Signals Apache Solr to start, stop, or restart. Obviously this is only going to work if the minion resides on the solr host. Additionally Solr doesn't ship with an init script so one must be created.

signal [str (None)] The command to pass to the apache solr init valid values are 'start', 'stop', and 'restart'

CLI Example:

```
salt '*' solr.signal restart
```

`salt.modules.solr.`**`version`**(*core_name=None*)

Gets the solr version for the core specified. You should specify a core here as all the cores will run under the same servlet container and so will all have the same version.

core_name [str (None)] The name of the solr core if using cores. Leave this blank if you are not using cores or if you want to check all cores.

Return : dict<str,obj>:

```
{'success':boolean, 'data':dict, 'errors':list, 'warnings':list}
```

CLI Example:

```
salt '*' solr.version
```

21.9.400 salt.modules.solrcloud module

Module for solrcloud configuration

New in version 2017.7.0.

For now, module is limited to http-exposed API. It doesn't implement config upload via Solr zkCli

`salt.modules.solrcloud.`**`BOOL_PROPS_LIST = ['transient', 'loadOnStartup']`**

Collections options type definition Reference: https://cwiki.apache.org/confluence/display/solr/Collections+API#CollectionsAPI-api1

`salt.modules.solrcloud.`**`DICT_OPTIONS_LIST = ['properties']`**

Collection unmodifiable options Reference: https://cwiki.apache.org/confluence/display/solr/Collections+API#CollectionsAPI-modifycoll

`salt.modules.solrcloud.`**`alias_exists`**(*alias_name*, ***kwargs*)

Check alias existence

Additional parameters (kwargs) may be passed, they will be proxied to http.query

CLI Example:

```
salt '*' solrcloud.alias_exists my_alias
```

`salt.modules.solrcloud.`**`alias_get_collections`**(*alias_name*, ***kwargs*)

Get collection list for an alias

Additional parameters (kwargs) may be passed, they will be proxied to http.query

CLI Example:

```
salt '*' solrcloud.alias_get my_alias
```

`salt.modules.solrcloud.`**`alias_set_collections`**(*alias_name*, *collections=None*, ***kwargs*)

Define an alias

Additional parameters (kwargs) may be passed, they will be proxied to http.query

CLI Example:

```
salt '*' solrcloud.alias_set my_alias collections=[collection1, colletion2]
```

`salt.modules.solrcloud.`**`cluster_status`**(***kwargs*)

Get cluster status

Additional parameters (kwargs) may be passed, they will be proxied to http.query

CLI Example:

```
salt '*' solrcloud.cluster_status
```

`salt.modules.solrcloud.`**`collection_backup`**(*collection_name*, *location*, *backup_name=None*, ***kwargs*)

Create a backup for a collection.

Additional parameters (kwargs) may be passed, they will be proxied to http.query

CLI Example:

```
salt '*' solrcloud.core_backup collection_name /mnt/nfs_backup
```

`salt.modules.solrcloud.`**`collection_backup_all`**(*location*, *backup_name=None*, ***kwargs*)

Create a backup for all collection present on the server.

Additional parameters (kwargs) may be passed, they will be proxied to http.query

CLI Example:

```
salt '*' solrcloud.core_backup /mnt/nfs_backup
```

`salt.modules.solrcloud.`**`collection_check_options`**(*options*)

Check collections options

CLI Example:

```
salt '*' solrcloud.collection_check_options '{"replicationFactor":4}'
```

`salt.modules.solrcloud.`**`collection_create`**(*collection_name*, *options=None*, ***kwargs*)

Create a collection,

Additional parameters (kwargs) may be passed, they will be proxied to http.query

CLI Example:

```
salt '*' solrcloud.collection_create collection_name
```

Collection creation options may be passed using the "options" parameter. Do not include option "name" since it already specified by the mandatory parameter "collection_name"

```
salt '*' solrcloud.collection_create collection_name options={"replicationFactor
↪":2, "numShards":3}
```

Cores options may be passed using the ”properties” key in options. Do not include property ”name”

```
salt '*' solrcloud.collection_create collection_name options={"replicationFactor
↪":2, "numShards":3,            "properties":{"dataDir":"/srv/solr/
↪hugePartitionSollection"}}
```

salt.modules.solrcloud.**collection_creation_options**()

Get collection option list that can only be defined at creation

CLI Example:

```
salt '*' solrcloud.collection_creation_options
```

salt.modules.solrcloud.**collection_exists**(*collection_name*, ***kwargs*)

Check if a collection exists

Additional parameters (kwargs) may be passed, they will be proxied to http.query

CLI Example:

```
salt '*' solrcloud.collection_exists collection_name
```

salt.modules.solrcloud.**collection_get_options**(*collection_name*, ***kwargs*)

Get collection options

Additional parameters (kwargs) may be passed, they will be proxied to http.query

CLI Example:

```
salt '*' solrcloud.collection_get_options collection_name
```

salt.modules.solrcloud.**collection_list**(***kwargs*)

List all collections

Additional parameters (kwargs) may be passed, they will be proxied to http.query

CLI Example:

```
salt '*' solrcloud.collection_list
```

salt.modules.solrcloud.**collection_reload**(*collection*, ***kwargs*)

Check if a collection exists

Additional parameters (kwargs) may be passed, they will be proxied to http.query

CLI Example:

```
salt '*' solrcloud.collection_reload collection_name
```

salt.modules.solrcloud.**collection_set_options**(*collection_name*, *options*, ***kwargs*)

Change collection options

Additional parameters (kwargs) may be passed, they will be proxied to http.query

Note that not every parameter can be changed after collection creation

CLI Example:

```
salt '*' solrcloud.collection_set_options collection_name options={
↪"replicationFactor":4}
```

salt.modules.solrcloud.**log = <SaltLoggingClass salt.modules.solrcloud (GARBAGE)>**
Core properties type definition. Reference: https://cwiki.apache.org/confluence/display/solr/Defining+core.properties

21.9.401 salt.modules.splunk

Module for interop with the Splunk API

New in version 2016.3.0..

depends

- splunk-sdk python module

configuration Configure this module by specifying the name of a configuration profile in the minion config, minion pillar, or master config. The module will use the 'splunk' key by default, if defined.

For example:

```
splunk:
    username: alice
    password: abc123
    host: example.splunkcloud.com
    port: 8080
```

salt.modules.splunk.**create_user**(*email, profile='splunk', **kwargs*)
create a splunk user by name/email

CLI Example:
salt myminion splunk.create_user user@example.com roles=['user'] realname="Test User" name=testuser

salt.modules.splunk.**delete_user**(*email, profile='splunk'*)
Delete a splunk user by email

CLI Example:
salt myminion splunk_user.delete 'user@example.com'

salt.modules.splunk.**get_user**(*email, profile='splunk', **kwargs*)
Get a splunk user by name/email

CLI Example:
salt myminion splunk.get_user 'user@example.com' user_details=false salt myminion splunk.get_user 'user@example.com' user_details=true

salt.modules.splunk.**list_users**(*profile='splunk'*)
List all users in the splunk DB

CLI Example:
salt myminion splunk.list_users

salt.modules.splunk.**update_user**(*email, profile='splunk', **kwargs*)
Create a splunk user by email

CLI Example:
salt myminion splunk.update_user example@domain.com roles=['user'] realname="Test User"

21.9.402 salt.modules.splunk_search

Module for interop with the Splunk API

New in version 2015.5.0.

depends

- splunk-sdk python module

configuration Configure this module by specifying the name of a configuration profile in the minion config, minion pillar, or master config. The module will use the 'splunk' key by default, if defined.

For example:

```
splunk:
    username: alice
    password: abc123
    host: example.splunkcloud.com
    port: 8080
```

`salt.modules.splunk_search.`**`create`**(*name*, *profile='splunk'*, ***kwargs*)

Create a splunk search

CLI Example:

splunk_search.create 'my search name' search='error msg'

`salt.modules.splunk_search.`**`delete`**(*name*, *profile='splunk'*)

Delete a splunk search

CLI Example:

splunk_search.delete 'my search name'

`salt.modules.splunk_search.`**`get`**(*name*, *profile='splunk'*)

Get a splunk search

CLI Example:

splunk_search.get 'my search name'

`salt.modules.splunk_search.`**`list_`**(*profile='splunk'*)

List splunk searches (names only)

CLI Example: splunk_search.list

`salt.modules.splunk_search.`**`list_all`**(*prefix=None*, *app=None*, *owner=None*, *description_contains=None*, *name_not_contains=None*, *profile='splunk'*)

Get all splunk search details. Produces results that can be used to create an sls file.

if app or owner are specified, results will be limited to matching saved searches.

if description_contains is specified, results will be limited to those where "description_contains in description" is true if name_not_contains is specified, results will be limited to those where "name_not_contains not in name" is true.

If prefix parameter is given, alarm names in the output will be prepended with the prefix; alarms that have the prefix will be skipped. This can be used to convert existing alarms to be managed by salt, as follows:

CLI example:

1. **Make a "backup" of all existing searches** $ salt-call splunk_search.list_all –out=txt | sed "s/local: //" > legacy_searches.sls
2. **Get all searches with new prefixed names** $ salt-call splunk_search.list_all "prefix=**MANAGED BY SALT** " –out=txt | sed "s/local: //" > managed_searches.sls
3. **Insert the managed searches into splunk** $ salt-call state.sls managed_searches.sls
4. Manually verify that the new searches look right
5. Delete the original searches $ sed s/present/absent/ legacy_searches.sls > remove_legacy_searches.sls $ salt-call state.sls remove_legacy_searches.sls

6. Get all searches again, verify no changes $ salt-call splunk_search.list_all –out=txt | sed ”s/local: //” > final_searches.sls $ diff final_searches.sls managed_searches.sls

`salt.modules.splunk_search.`**`update`**`(`*name*, *profile='splunk'*, ***kwargs*`)`

Update a splunk search

CLI Example:

splunk_search.update 'my search name' sharing=app

21.9.403 salt.modules.sqlite3

Support for SQLite3

`salt.modules.sqlite3.`**`fetch`**`(`*db=None*, *sql=None*`)`

Retrieve data from an sqlite3 db (returns all rows, be careful!)

CLI Example:

```
salt '*' sqlite3.fetch /root/test.db 'SELECT * FROM test;'
```

`salt.modules.sqlite3.`**`indexes`**`(`*db=None*`)`

Show all indices in the database, for people with poor spelling skills

CLI Example:

```
salt '*' sqlite3.indexes /root/test.db
```

`salt.modules.sqlite3.`**`indices`**`(`*db=None*`)`

Show all indices in the database

CLI Example:

```
salt '*' sqlite3.indices /root/test.db
```

`salt.modules.sqlite3.`**`modify`**`(`*db=None*, *sql=None*`)`

Issue an SQL query to sqlite3 (with no return data), usually used to modify the database in some way (insert, delete, create, etc)

CLI Example:

```
salt '*' sqlite3.modify /root/test.db 'CREATE TABLE test(id INT, testdata TEXT);'
```

`salt.modules.sqlite3.`**`sqlite_version`**`()`

Return version of sqlite

CLI Example:

```
salt '*' sqlite3.sqlite_version
```

`salt.modules.sqlite3.`**`tables`**`(`*db=None*`)`

Show all tables in the database

CLI Example:

```
salt '*' sqlite3.tables /root/test.db
```

`salt.modules.sqlite3.`**`version`**`()`

Return version of pysqlite

CLI Example:

```
salt '*' sqlite3.version
```

21.9.404 salt.modules.ssh

Manage client ssh components

Note: This module requires the use of MD5 hashing. Certain security audits may not permit the use of MD5. For those cases, this module should be disabled or removed.

`salt.modules.ssh.`**`auth_keys`**(*user=None, config='.ssh/authorized_keys', fingerprint_hash_type=None*)

Return the authorized keys for users

CLI Example:

```
salt '*' ssh.auth_keys
salt '*' ssh.auth_keys root
salt '*' ssh.auth_keys user=root
salt '*' ssh.auth_keys user="[user1, user2]"
```

`salt.modules.ssh.`**`check_key`**(*user, key, enc, comment, options, config='.ssh/authorized_keys', cache_keys=None, fingerprint_hash_type=None*)

Check to see if a key needs updating, returns "update", "add" or "exists"

CLI Example:

```
salt '*' ssh.check_key <user> <key> <enc> <comment> <options>
```

`salt.modules.ssh.`**`check_key_file`**(*user, source, config='.ssh/authorized_keys', saltenv='base', fingerprint_hash_type=None*)

Check a keyfile from a source destination against the local keys and return the keys to change

CLI Example:

```
salt '*' ssh.check_key_file root salt://ssh/keyfile
```

`salt.modules.ssh.`**`check_known_host`**(*user=None, hostname=None, key=None, fingerprint=None, config=None, port=None, fingerprint_hash_type=None*)

Check the record in known_hosts file, either by its value or by fingerprint (it's enough to set up either key or fingerprint, you don't need to set up both).

If provided key or fingerprint doesn't match with stored value, return "update", if no value is found for a given host, return "add", otherwise return "exists".

If neither key, nor fingerprint is defined, then additional validation is not performed.

CLI Example:

```
salt '*' ssh.check_known_host <user> <hostname> key='AAAA...FAaQ=='
```

`salt.modules.ssh.`**`get_known_host_entries`**(*user, hostname, config=None, port=None, fingerprint_hash_type=None*)

New in version 2018.3.0.

Return information about known host entries from the configfile, if any. If there are no entries for a matching hostname, return None.

CLI Example:

```
salt '*' ssh.get_known_host_entries <user> <hostname>
```

`salt.modules.ssh.`**`hash_known_hosts`**`(`*user=None, config=None*`)`

Hash all the hostnames in the known hosts file.

New in version 2014.7.0.

user hash known hosts of this user

config path to known hosts file: can be absolute or relative to user's home directory

CLI Example:

```
salt '*' ssh.hash_known_hosts
```

`salt.modules.ssh.`**`host_keys`**`(`*keydir=None, private=True, certs=True*`)`

Return the minion's host keys

CLI Example:

```
salt '*' ssh.host_keys
salt '*' ssh.host_keys keydir=/etc/ssh
salt '*' ssh.host_keys keydir=/etc/ssh private=False
salt '*' ssh.host_keys keydir=/etc/ssh certs=False
```

`salt.modules.ssh.`**`key_is_encrypted`**`(`*key*`)`

New in version 2015.8.7.

Function to determine whether or not a private key is encrypted with a passphrase.

Checks key for a `Proc-Type` header with `ENCRYPTED` in the value. If found, returns `True`, otherwise returns `False`.

CLI Example:

```
salt '*' ssh.key_is_encrypted /root/id_rsa
```

`salt.modules.ssh.`**`recv_known_host_entries`**`(`*hostname, enc=None, port=None, hash_known_hosts=True, timeout=5, fingerprint_hash_type=None*`)`

New in version 2018.3.0.

Retrieve information about host public keys from remote server

hostname The name of the remote host (e.g. "github.com")

enc Defines what type of key is being used, can be ed25519, ecdsa ssh-rsa or ssh-dss

port Optional parameter, denoting the port of the remote host on which an SSH daemon is running. By default the port 22 is used.

hash_known_hosts [True] Hash all hostnames and addresses in the known hosts file.

timeout [int] Set the timeout for connection attempts. If `timeout` seconds have elapsed since a connection was initiated to a host or since the last time anything was read from that host, then the connection is closed and the host in question considered unavailable. Default is 5 seconds.

fingerprint_hash_type The fingerprint hash type that the public key fingerprints were originally hashed with. This defaults to `sha256` if not specified.

New in version 2016.11.4.

Changed in version 2017.7.0:: default changed from `md5` to `sha256`

CLI Example:

```
salt '*' ssh.recv_known_host_entries <hostname> enc=<enc> port=<port>
```

`salt.modules.ssh.`**`rm_auth_key`**(*user*, *key*, *config='.ssh/authorized_keys'*, *fingerprint_hash_type=None*)

Remove an authorized key from the specified user's authorized key file

CLI Example:

```
salt '*' ssh.rm_auth_key <user> <key>
```

`salt.modules.ssh.`**`rm_auth_key_from_file`**(*user*, *source*, *config='.ssh/authorized_keys'*, *saltenv='base'*, *fingerprint_hash_type=None*)

Remove an authorized key from the specified user's authorized key file, using a file as source

CLI Example:

```
salt '*' ssh.rm_auth_key_from_file <user> salt://ssh_keys/<user>.id_rsa.pub
```

`salt.modules.ssh.`**`rm_known_host`**(*user=None*, *hostname=None*, *config=None*, *port=None*)

Remove all keys belonging to hostname from a known_hosts file.

CLI Example:

```
salt '*' ssh.rm_known_host <user> <hostname>
```

`salt.modules.ssh.`**`set_auth_key`**(*user*, *key*, *enc='ssh-rsa'*, *comment=''*, *options=None*, *config='.ssh/authorized_keys'*, *cache_keys=None*, *fingerprint_hash_type=None*)

Add a key to the authorized_keys file. The "key" parameter must only be the string of text that is the encoded key. If the key begins with "ssh-rsa" or ends with user@host, remove those from the key before passing it to this function.

CLI Example:

```
salt '*' ssh.set_auth_key <user> '<key>' enc='dsa'
```

`salt.modules.ssh.`**`set_auth_key_from_file`**(*user*, *source*, *config='.ssh/authorized_keys'*, *saltenv='base'*, *fingerprint_hash_type=None*)

Add a key to the authorized_keys file, using a file as the source.

CLI Example:

```
salt '*' ssh.set_auth_key_from_file <user> salt://ssh_keys/<user>.id_rsa.pub
```

`salt.modules.ssh.`**`set_known_host`**(*user=None*, *hostname=None*, *fingerprint=None*, *key=None*, *port=None*, *enc=None*, *config=None*, *hash_known_hosts=True*, *timeout=5*, *fingerprint_hash_type=None*)

Download SSH public key from remote host "hostname", optionally validate its fingerprint against "fingerprint" variable and save the record in the known_hosts file.

If such a record does already exists in there, do nothing.

user The user who owns the ssh authorized keys file to modify

hostname The name of the remote host (e.g. "github.com")

fingerprint The fingerprint of the key which must be present in the known_hosts file (optional if key specified)

key The public key which must be presented in the known_hosts file (optional if fingerprint specified)

port optional parameter, denoting the port of the remote host, which will be used in case, if the public key will be requested from it. By default the port 22 is used.

enc Defines what type of key is being used, can be ed25519, ecdsa ssh-rsa or ssh-dss

config The location of the authorized keys file relative to the user's home directory, defaults to ".ssh/known_hosts". If no user is specified, defaults to "/etc/ssh/ssh_known_hosts". If present, must be an absolute path when a user is not specified.

hash_known_hosts [True] Hash all hostnames and addresses in the known hosts file.

timeout [int] Set the timeout for connection attempts. If `timeout` seconds have elapsed since a connection was initiated to a host or since the last time anything was read from that host, then the connection is closed and the host in question considered unavailable. Default is 5 seconds.

New in version 2016.3.0.

fingerprint_hash_type The public key fingerprint hash type that the public key fingerprint was originally hashed with. This defaults to `sha256` if not specified.

New in version 2016.11.4.

Changed in version 2017.7.0:: default changed from `md5` to `sha256`

CLI Example:

```
salt '*' ssh.set_known_host <user> fingerprint='xx:xx:..:xx' enc='ssh-rsa' config=
↪'.ssh/known_hosts'
```

`salt.modules.ssh.`**`user_keys`**(*user=None*, *pubfile=None*, *prvfile=None*)

Return the user's ssh keys on the minion

New in version 2014.7.0.

CLI Example:

```
salt '*' ssh.user_keys
salt '*' ssh.user_keys user=user1
salt '*' ssh.user_keys user=user1 pubfile=/home/user1/.ssh/id_rsa.pub prvfile=/
↪home/user1/.ssh/id_rsa
salt '*' ssh.user_keys user=user1 prvfile=False
salt '*' ssh.user_keys user="['user1','user2'] pubfile=id_rsa.pub prvfile=id_rsa
```

As you can see you can tell Salt not to read from the user's private (or public) key file by setting the file path to `False`. This can be useful to prevent Salt from publishing private data via Salt Mine or others.

21.9.405 salt.modules.ssh_pkg

Service support for the REST example

`salt.modules.ssh_pkg.`**`install`**(*name=None*, *refresh=False*, *fromrepo=None*, *pkgs=None*, *sources=None*, ***kwargs*)

`salt.modules.ssh_pkg.`**`list_pkgs`**(*versions_as_list=False*, ***kwargs*)

`salt.modules.ssh_pkg.`**`remove`**(*name=None*, *pkgs=None*, ***kwargs*)

21.9.406 salt.modules.ssh_service module

Provide the service module for the proxy-minion SSH sample .. versionadded:: 2015.8.2

`salt.modules.ssh_service.`**`enabled`**(*name*, *sig=None*)

Only the 'redbull' service is 'enabled' in the test

`salt.modules.ssh_service.`**`get_all`**()

Return a list of all available services

CLI Example:

```
salt '*' service.get_all
```

`salt.modules.ssh_service.`**`list_`**`()`

Return a list of all available services.

CLI Example:

```
salt '*' service.list
```

`salt.modules.ssh_service.`**`restart`**`(`*name*, *sig=None*`)`

Restart the specified service with rest_sample CLI Example:

```
salt '*' service.restart <service name>
```

`salt.modules.ssh_service.`**`running`**`(`*name*, *sig=None*`)`

Return whether this service is running.

`salt.modules.ssh_service.`**`start`**`(`*name*, *sig=None*`)`

Start the specified service on the ssh_sample

CLI Example:

```
salt '*' service.start <service name>
```

`salt.modules.ssh_service.`**`status`**`(`*name*, *sig=None*`)`

Return the status for a service via ssh_sample. If the name contains globbing, a dict mapping service name to True/False values is returned.

Changed in version 2018.3.0: The service name can now be a glob (e.g. `salt*`)

Parameters

- **`name`** (`str`) – The name of the service to check
- **`sig`** (`str`) – Not implemented

Returns True if running, False otherwise dict: Maps service name to True if running, False otherwise

Return type bool

CLI Example:

```
salt '*' service.status <service name>
```

`salt.modules.ssh_service.`**`stop`**`(`*name*, *sig=None*`)`

Stop the specified service on the rest_sample

CLI Example:

```
salt '*' service.stop <service name>
```

21.9.407 salt.modules.snapper module

Module to manage filesystem snapshots with snapper

New in version 2016.11.0.

codeauthor Duncan Mac-Vicar P. <dmacvicar@suse.de>

codeauthor Pablo Suárez Hernández <psuarezhernandez@suse.de>

depends `dbus` Python module.

depends `snapper` http://snapper.io, available in most distros

maturity new

platform Linux

`salt.modules.snapper.`**`changed_files`**(*config='root', num_pre=None, num_post=None*)

Returns the files changed between two snapshots

config Configuration name.

num_pre first snapshot ID to compare. Default is last snapshot

num_post last snapshot ID to compare. Default is 0 (current state)

CLI example:

```
salt '*' snapper.changed_files
salt '*' snapper.changed_files num_pre=19 num_post=20
```

`salt.modules.snapper.`**`create_baseline`**(*tag='baseline', config='root'*)

Creates a snapshot marked as baseline

tag Tag name for the baseline

config Configuration name.

CLI Example:

```
salt '*' snapper.create_baseline
salt '*' snapper.create_baseline my_custom_baseline
```

`salt.modules.snapper.`**`create_config`**(*name=None, subvolume=None, fstype=None, template=None, extra_opts=None*)

Creates a new Snapper configuration

name Name of the new Snapper configuration.

subvolume Path to the related subvolume.

fstype Filesystem type of the subvolume.

template Configuration template to use. (Default: default)

extra_opts Extra Snapper configuration opts dictionary. It will override the values provided by the given template (if any).

CLI example:

```
salt '*' snapper.create_config name=myconfig subvolume=/foo/bar/ fstype=btrfs
salt '*' snapper.create_config name=myconfig subvolume=/foo/bar/ fstype=btrfs␣
↪template="default"
salt '*' snapper.create_config name=myconfig subvolume=/foo/bar/ fstype=btrfs␣
↪extra_opts='{"NUMBER_CLEANUP": False}'
```

`salt.modules.snapper.`**`create_snapshot`**(*config='root', snapshot_type='single', pre_number=None, description=None, cleanup_algorithm='number', userdata=None, **kwargs*)

Creates an snapshot

config Configuration name.

snapshot_type Specifies the type of the new snapshot. Possible values are single, pre and post.

pre_number For post snapshots the number of the pre snapshot must be provided.

description Description for the snapshot. If not given, the salt job will be used.

cleanup_algorithm Set the cleanup algorithm for the snapshot.

number Deletes old snapshots when a certain number of snapshots is reached.

timeline Deletes old snapshots but keeps a number of hourly, daily, weekly, monthly and yearly snapshots.

empty-pre-post Deletes pre/post snapshot pairs with empty diffs.

userdata Set userdata for the snapshot (key-value pairs).

Returns the number of the created snapshot.

CLI example:

```
salt '*' snapper.create_snapshot
```

`salt.modules.snapper.`**`delete_snapshot`**(*snapshots_ids=None, config='root'*)

Deletes an snapshot

config Configuration name. (Default: root)

snapshots_ids List of the snapshots IDs to be deleted.

CLI example:

```
salt '*' snapper.delete_snapshot 54
salt '*' snapper.delete_snapshot config=root 54
salt '*' snapper.delete_snapshot config=root snapshots_ids=[54,55,56]
```

`salt.modules.snapper.`**`diff`**(*config='root', filename=None, num_pre=None, num_post=None*)

Returns the differences between two snapshots

config Configuration name.

filename if not provided the showing differences between snapshots for all "text" files

num_pre first snapshot ID to compare. Default is last snapshot

num_post last snapshot ID to compare. Default is 0 (current state)

CLI Example:

```
salt '*' snapper.diff
salt '*' snapper.diff filename=/var/log/snapper.log num_pre=19 num_post=20
```

`salt.modules.snapper.`**`diff_jid`**(*jid, config='root'*)

Returns the changes applied by a *jid*

jid The job id to lookup

config Configuration name.

CLI Example:

```
salt '*' snapper.diff_jid jid=20160607130930720112
```

`salt.modules.snapper.`**`get_config`**(*name='root'*)

Retrieves all values from a given configuration

CLI example:

```
salt '*' snapper.get_config
```

`salt.modules.snapper.`**`get_snapshot`**(*number=0, config='root'*)

Get detailed information about a given snapshot

CLI example:

```
salt '*' snapper.get_snapshot 1
```

`salt.modules.snapper.`**`list_configs`**()

List all available configs

CLI example:

```
salt '*' snapper.list_configs
```

`salt.modules.snapper.`**`list_snapshots`**(*config='root'*)

List available snapshots

CLI example:

```
salt '*' snapper.list_snapshots config=myconfig
```

`salt.modules.snapper.`**`modify_snapshot`**(*snapshot_id=None*, *description=None*, *userdata=None*, *cleanup=None*, *config='root'*)

Modify attributes of an existing snapshot.

config Configuration name. (Default: root)

snapshot_id ID of the snapshot to be modified.

cleanup Change the cleanup method of the snapshot. (str)

description Change the description of the snapshot. (str)

userdata Change the userdata dictionary of the snapshot. (dict)

CLI example:

```
salt '*' snapper.modify_snapshot 54 description="my snapshot description"
salt '*' snapper.modify_snapshot 54 description="my snapshot description"
salt '*' snapper.modify_snapshot 54 userdata='{"foo": "bar"}'
salt '*' snapper.modify_snapshot snapshot_id=54 cleanup="number"
```

`salt.modules.snapper.`**`run`**(*function*, **args*, ***kwargs*)

Runs a function from an execution module creating pre and post snapshots and associating the salt job id with those snapshots for easy undo and cleanup.

function Salt function to call.

config Configuration name. (default: "root")

description A description for the snapshots. (default: None)

userdata Data to include in the snapshot metadata. (default: None)

cleanup_algorithm Snapper cleanup algorithm. (default: "number")

****args*** args for the function to call. (default: None)

*****kwargs*** kwargs for the function to call (default: None)

This would run append text to /etc/motd using the file.append module, and will create two snapshots, pre and post with the associated metadata. The jid will be available as salt_jid in the userdata of the snapshot.

You can immediately see the changes

CLI Example:

```
salt '*' snapper.run file.append args='["/etc/motd", "some text"]'
```

`salt.modules.snapper.`**`set_config`**(*name='root'*, ***kwargs*)

Set configuration values

CLI example:

```
salt '*' snapper.set_config SYNC_ACL=True
```

Keys are case insensitive as they will be always uppercased to snapper convention. The above example is equivalent to:

```
salt '*' snapper.set_config sync_acl=True
```

`salt.modules.snapper.`**`status`**(*config='root'*, *num_pre=None*, *num_post=None*)

Returns a comparison between two snapshots

config Configuration name.

num_pre first snapshot ID to compare. Default is last snapshot

num_post last snapshot ID to compare. Default is 0 (current state)

CLI example:

```
salt '*' snapper.status
salt '*' snapper.status num_pre=19 num_post=20
```

`salt.modules.snapper.`**`status_to_string`**(*dbus_status*)

Converts a numeric dbus snapper status into a string

CLI Example:

```
salt '*' snapper.status_to_string <dbus_status>
```

`salt.modules.snapper.`**`undo`**(*config='root', files=None, num_pre=None, num_post=None*)

Undo all file changes that happened between num_pre and num_post, leaving the files into the state of num_pre.

> **Warning:** If one of the files has changes after num_post, they will be overwritten The snapshots are used to determine the file list, but the current version of the files will be overwritten by the versions in num_pre.
>
> You to undo changes between num_pre and the current version of the files use num_post=0.

CLI Example:

```
salt '*' snapper.undo
```

`salt.modules.snapper.`**`undo_jid`**(*jid, config='root'*)

Undo the changes applied by a salt job

jid The job id to lookup

config Configuration name.

CLI Example:

```
salt '*' snapper.undo_jid jid=20160607130930720112
```

21.9.408 salt.modules.state

Control the state system on the minion.

State Caching

When a highstate is called, the minion automatically caches a copy of the last high data. If you then run a highstate with cache=True it will use that cached highdata and won't hit the fileserver except for `salt://` links in the states themselves.

`salt.modules.state.`**`apply_`**(*mods=None, **kwargs*)

New in version 2015.5.0.

This function will call *`state.highstate`* or *`state.sls`* based on the arguments passed to this function. It exists as a more intuitive way of applying states.

APPLYING ALL STATES CONFIGURED IN TOP.SLS (A.K.A. HIGHSTATE)

To apply all configured states, simply run `state.apply`:

```
salt '*' state.apply
```

The following additional arguments are also accepted when applying all states configured in top.sls:

test Run states in test-only (dry-run) mode

mock The mock option allows for the state run to execute without actually calling any states. This then returns a mocked return which will show the requisite ordering as well as fully validate the state run.

New in version 2015.8.4.

pillar Custom Pillar values, passed as a dictionary of key-value pairs

```
salt '*' state.apply stuff pillar='{"foo": "bar"}'
```

Note: Values passed this way will override Pillar values set via `pillar_roots` or an external Pillar source.

---

exclude Exclude specific states from execution. Accepts a list of sls names, a comma-separated string of sls names, or a list of dictionaries containing `sls` or `id` keys. Glob-patterns may be used to match multiple states.

```
salt '*' state.apply exclude=bar,baz
salt '*' state.apply exclude=foo*
salt '*' state.apply exclude="[{'id': 'id_to_exclude'}, {'sls': 'sls_to_
↪exclude'}]"
```

queue [False] Instead of failing immediately when another state run is in progress, queue the new state run to begin running once the other has finished.

This option starts a new thread for each queued state run, so use this option sparingly.

localconfig Optionally, instead of using the minion config, load minion opts from the file specified by this argument, and then merge them with the options from the minion config. This functionality allows for specific states to be run with their own custom minion configuration, including different pillars, file_roots, etc.

```
salt '*' state.apply localconfig=/path/to/minion.yml
```

APPLYING INDIVIDUAL SLS FILES (A.K.A. `STATE.SLS`)

To apply individual SLS files, pass them as a comma-separated list:

```
# Run the states configured in salt://stuff.sls (or salt://stuff/init.sls)
salt '*' state.apply stuff

# Run the states configured in salt://stuff.sls (or salt://stuff/init.sls)
# and salt://pkgs.sls (or salt://pkgs/init.sls).
salt '*' state.apply stuff,pkgs

# Run the states configured in a more deeply nested directory such as salt://my/
↪organized/stuff.sls (or salt://my/organized/stuff/init.sls)
salt '*' state.apply my.organized.stuff
```

The following additional arguments are also accepted when applying individual SLS files:

test Run states in test-only (dry-run) mode

mock The mock option allows for the state run to execute without actually calling any states. This then returns a mocked return which will show the requisite ordering as well as fully validate the state run.

New in version 2015.8.4.

pillar Custom Pillar values, passed as a dictionary of key-value pairs

```
salt '*' state.apply stuff pillar='{"foo": "bar"}'
```

Note: Values passed this way will override Pillar values set via `pillar_roots` or an external Pillar source.

queue [False] Instead of failing immediately when another state run is in progress, queue the new state run to begin running once the other has finished.

This option starts a new thread for each queued state run, so use this option sparingly.

concurrent [False] Execute state runs concurrently instead of serially

Warning: This flag is potentially dangerous. It is designed for use when multiple state runs can safely be run at the same time. Do *not* use this flag for performance optimization.

saltenv Specify a salt fileserver environment to be used when applying states

Changed in version 0.17.0: Argument name changed from `env` to `saltenv`

Changed in version 2014.7.0: If no saltenv is specified, the minion config will be checked for an `environment` parameter and if found, it will be used. If none is found, `base` will be used. In prior releases, the minion config was not checked and `base` would always be assumed when the saltenv was not explicitly set.

pillarenv Specify a Pillar environment to be used when applying states. This can also be set in the minion config file using the *`pillarenv`* option. When neither the *`pillarenv`* minion config option nor this CLI argument is used, all Pillar environments will be merged together.

localconfig Optionally, instead of using the minion config, load minion opts from the file specified by this argument, and then merge them with the options from the minion config. This functionality allows for specific states to be run with their own custom minion configuration, including different pillars, file_roots, etc.

```
salt '*' state.apply stuff localconfig=/path/to/minion.yml
```

sync_mods If specified, the desired custom module types will be synced prior to running the SLS files:

```
salt '*' state.apply stuff sync_mods=states,modules
salt '*' state.apply stuff sync_mods=all
```

Note: This option is ignored when no SLS files are specified, as a *highstate* automatically syncs all custom module types.

New in version 2017.7.8,2018.3.3,2019.2.0.

`salt.modules.state.`**`check_request`**`(`*name=None*`)`

New in version 2015.5.0.

Return the state request information, if any

CLI Example:

```
salt '*' state.check_request
```

`salt.modules.state.`**`clear_cache`**`()`

Clear out cached state files, forcing even cache runs to refresh the cache on the next state execution.

Remember that the state cache is completely disabled by default, this execution only applies if cache=True is used in states

CLI Example:

```
salt '*' state.clear_cache
```

`salt.modules.state.`**`clear_request`**`(`*name=None*`)`

New in version 2015.5.0.

Clear out the state execution request without executing it

CLI Example:

```
salt '*' state.clear_request
```

`salt.modules.state.`**`disable`**`(`*states*`)`

Disable state runs.

CLI Example:

```
salt '*' state.disable highstate

salt '*' state.disable highstate,test.succeed_without_changes
```

Note: To disable a state file from running provide the same name that would be passed in a state.sls call.

salt '*' state.disable bind.config

`salt.modules.state.`**`enable`**`(`*states*`)`

Enable state function or sls run

CLI Example:

```
salt '*' state.enable highstate

salt '*' state.enable test.succeed_without_changes
```

Note: To enable a state file from running provide the same name that would be passed in a state.sls call.

salt '*' state.disable bind.config

`salt.modules.state.`**`event`**`(`*tagmatch='*', count=-1, quiet=False, sock_dir=None, pretty=False, node='minion'*`)`

Watch Salt's event bus and block until the given tag is matched

New in version 2016.3.0.

Changed in version 2019.2.0: `tagmatch` can now be either a glob or regular expression.

This is useful for utilizing Salt's event bus from shell scripts or for taking simple actions directly from the CLI.

Enable debug logging to see ignored events.

Parameters

- **`tagmatch`** – the event is written to stdout for each tag that matches this glob or regular expression.
- **`count`** – this number is decremented for each event that matches the `tagmatch` parameter; pass `-1` to listen forever.
- **`quiet`** – do not print to stdout; just block
- **`sock_dir`** – path to the Salt master's event socket file.
- **`pretty`** – Output the JSON all on a single line if `False` (useful for shell tools); pretty-print the JSON output if `True`.

- **node** – Watch the minion-side or master-side event bus.

CLI Example:

```
salt-call --local state.event pretty=True
```

`salt.modules.state.`**`get_pauses`**(*jid=None*)

Get a report on all of the currently paused state runs and pause run settings. Optionally send in a jid if you only desire to see a single pause data set.

`salt.modules.state.`**`high`**(*data, test=None, queue=False, **kwargs*)

Execute the compound calls stored in a single set of high data

This function is mostly intended for testing the state system and is not likely to be needed in everyday usage.

CLI Example:

```
salt '*' state.high '{"vim": {"pkg": ["installed"]}}'
```

`salt.modules.state.`**`highstate`**(*test=None, queue=False, **kwargs*)

Retrieve the state data from the salt master for this minion and execute it

test Run states in test-only (dry-run) mode

pillar Custom Pillar values, passed as a dictionary of key-value pairs

```
salt '*' state.highstate stuff pillar='{"foo": "bar"}'
```

Note: Values passed this way will override Pillar values set via `pillar_roots` or an external Pillar source.

Changed in version 2016.3.0: GPG-encrypted CLI Pillar data is now supported via the GPG renderer. See *here* for details.

pillar_enc Specify which renderer to use to decrypt encrypted data located within the `pillar` value. Currently, only `gpg` is supported.

New in version 2016.3.0.

exclude Exclude specific states from execution. Accepts a list of sls names, a comma-separated string of sls names, or a list of dictionaries containing `sls` or `id` keys. Glob-patterns may be used to match multiple states.

```
salt '*' state.highstate exclude=bar,baz
salt '*' state.highstate exclude=foo*
salt '*' state.highstate exclude="[{'id': 'id_to_exclude'}, {'sls': 'sls_to_
↪exclude'}]"
```

saltenv Specify a salt fileserver environment to be used when applying states

Changed in version 0.17.0: Argument name changed from `env` to `saltenv`.

Changed in version 2014.7.0: If no saltenv is specified, the minion config will be checked for a `saltenv` parameter and if found, it will be used. If none is found, `base` will be used. In prior releases, the minion config was not checked and `base` would always be assumed when the saltenv was not explicitly set.

pillarenv Specify a Pillar environment to be used when applying states. This can also be set in the minion config file using the *`pillarenv`* option. When neither the *`pillarenv`* minion config option nor this CLI argument is used, all Pillar environments will be merged together.

queue [False] Instead of failing immediately when another state run is in progress, queue the new state run to begin running once the other has finished.

This option starts a new thread for each queued state run, so use this option sparingly.

localconfig Optionally, instead of using the minion config, load minion opts from the file specified by this argument, and then merge them with the options from the minion config. This functionality allows for specific states to be run with their own custom minion configuration, including different pillars, file_roots, etc.

mock The mock option allows for the state run to execute without actually calling any states. This then returns a mocked return which will show the requisite ordering as well as fully validate the state run.

New in version 2015.8.4.

CLI Examples:

```
salt '*' state.highstate

salt '*' state.highstate whitelist=sls1_to_run,sls2_to_run
salt '*' state.highstate exclude=sls_to_exclude
salt '*' state.highstate exclude="[{'id': 'id_to_exclude'}, {'sls': 'sls_to_
↪exclude'}]"

salt '*' state.highstate pillar="{foo: 'Foo!', bar: 'Bar!'}"
```

`salt.modules.state.`**`id_exists`**(*ids*, *mods*, *test=None*, *queue=False*, ***kwargs*)

Tests for the existence of a specific ID or list of IDs within the specified SLS file(s). Similar to *`state.sls_exists`*, returns True or False. The default environment is base``, use `saltenv` to specify a different environment.

New in version 2019.2.0.

saltenv Specify a salt fileserver environment from which to look for the SLS files specified in the `mods` argument

CLI Example:

```
salt '*' state.id_exists create_myfile,update_template filestate saltenv=dev
```

`salt.modules.state.`**`list_disabled`**()

List the states which are currently disabled

CLI Example:

```
salt '*' state.list_disabled
```

`salt.modules.state.`**`low`**(*data*, *queue=False*, ***kwargs*)

Execute a single low data call

This function is mostly intended for testing the state system and is not likely to be needed in everyday usage.

CLI Example:

```
salt '*' state.low '{"state": "pkg", "fun": "installed", "name": "vi"}'
```

`salt.modules.state.`**`orchestrate`**(*mods*, *saltenv='base'*, *test=None*, *exclude=None*, *pillar=None*, *pillarenv=None*)

New in version 2016.11.0.

Execute the orchestrate runner from a masterless minion.

See also:

More Orchestrate documentation

- *Full Orchestrate Tutorial*
- `Docs for the ``salt`` state module <salt.states.saltmod>`

CLI Examples:

```
salt-call --local state.orchestrate webserver
salt-call --local state.orchestrate webserver saltenv=dev test=True
salt-call --local state.orchestrate webserver saltenv=dev pillarenv=aws
```

`salt.modules.state.`**`pause`**(*jid*, *state_id=None*, *duration=None*)

Set up a state id pause, this instructs a running state to pause at a given state id. This needs to pass in the jid of the running state and can optionally pass in a duration in seconds. If a state_id is not passed then the jid referenced will be paused at the beginning of the next state run.

The given state id is the id got a given state execution, so given a state that looks like this:

```
vim:
  pkg.installed: []
```

The state_id to pass to *pause* is *vim*

CLI Examples:

```
salt '*' state.pause 20171130110407769519
salt '*' state.pause 20171130110407769519 vim
salt '*' state.pause 20171130110407769519 vim 20
```

`salt.modules.state.`**`pkg`**(*pkg_path*, *pkg_sum*, *hash_type*, *test=None*, ***kwargs*)

Execute a packaged state run, the packaged state run will exist in a tarball available locally. This packaged state can be generated using salt-ssh.

CLI Example:

```
salt '*' state.pkg /tmp/salt_state.tgz 760a9353810e36f6d81416366fc426dc md5
```

`salt.modules.state.`**`request`**(*mods=None*, ***kwargs*)

New in version 2015.5.0.

Request that the local admin execute a state run via *salt-call state.run_request*. All arguments match those of state.apply.

CLI Example:

```
salt '*' state.request
salt '*' state.request stuff
salt '*' state.request stuff,pkgs
```

`salt.modules.state.`**`resume`**(*jid*, *state_id=None*)

Remove a pause from a jid, allowing it to continue. If the state_id is not specified then the a general pause will be resumed.

The given state_id is the id got a given state execution, so given a state that looks like this:

```
vim:
  pkg.installed: []
```

The state_id to pass to *rm_pause* is *vim*

CLI Examples:

```
salt '*' state.resume 20171130110407769519
salt '*' state.resume 20171130110407769519 vim
```

`salt.modules.state.`**`run_request`**(*name='default'*, ***kwargs*)

New in version 2015.5.0.

Execute the pending state request

CLI Example:

```
salt '*' state.run_request
```

`salt.modules.state.`**`running`**(*concurrent=False*)

Return a list of strings that contain state return data if a state function is already running. This function is used to prevent multiple state calls from being run at the same time.

CLI Example:

```
salt '*' state.running
```

`salt.modules.state.`**`show_highstate`**(*queue=False*, ***kwargs*)

Retrieve the highstate data from the salt master and display it

Custom Pillar data can be passed with the `pillar` kwarg.

CLI Example:

```
salt '*' state.show_highstate
```

`salt.modules.state.`**`show_low_sls`**(*mods*, *test=None*, *queue=False*, ***kwargs*)

Display the low data from a specific sls. The default environment is `base`, use `saltenv` to specify a different environment.

saltenv Specify a salt fileserver environment to be used when applying states

pillar Custom Pillar values, passed as a dictionary of key-value pairs

```
salt '*' state.show_low_sls stuff pillar='{"foo": "bar"}'
```

Note: Values passed this way will override Pillar values set via `pillar_roots` or an external Pillar source.

---

pillarenv Specify a Pillar environment to be used when applying states. This can also be set in the minion config file using the `pillarenv` option. When neither the `pillarenv` minion config option nor this CLI argument is used, all Pillar environments will be merged together.

CLI Example:

```
salt '*' state.show_low_sls foo
salt '*' state.show_low_sls foo saltenv=dev
```

`salt.modules.state.`**`show_lowstate`**(*queue=False*, ***kwargs*)

List out the low data that will be applied to this minion

CLI Example:

```
salt '*' state.show_lowstate
```

`salt.modules.state.`**`show_sls`**(*mods*, *test=None*, *queue=False*, ***kwargs*)

Display the state data from a specific sls or list of sls files on the master. The default environment is `base`, use `saltenv` to specify a different environment.

This function does not support topfiles. For `top.sls` please use `show_top` instead.

Custom Pillar data can be passed with the `pillar` kwarg.

saltenv Specify a salt fileserver environment to be used when applying states

pillarenv Specify a Pillar environment to be used when applying states. This can also be set in the minion config file using the *pillarenv* option. When neither the *pillarenv* minion config option nor this CLI argument is used, all Pillar environments will be merged together.

CLI Example:

```
salt '*' state.show_sls core,edit.vim saltenv=dev
```

`salt.modules.state.`**`show_state_usage`**(*queue=False*, ***kwargs*)

Retrieve the highstate data from the salt master to analyse used and unused states

Custom Pillar data can be passed with the `pillar` kwarg.

CLI Example:

```
salt '*' state.show_state_usage
```

`salt.modules.state.`**`show_states`**(*queue=False*, ***kwargs*)

Returns the list of states that will be applied on highstate.

CLI Example:

```
salt '*' state.show_states
```

New in version 2019.2.0.

`salt.modules.state.`**`show_top`**(*queue=False*, ***kwargs*)

Return the top data that the minion will use for a highstate

CLI Example:

```
salt '*' state.show_top
```

`salt.modules.state.`**`single`**(*fun*, *name*, *test=None*, *queue=False*, ***kwargs*)

Execute a single state function with the named kwargs, returns False if insufficient data is sent to the command

By default, the values of the kwargs will be parsed as YAML. So, you can specify lists values, or lists of single entry key-value maps, as you would in a YAML salt file. Alternatively, JSON format of keyword values is also supported.

CLI Example:

```
salt '*' state.single pkg.installed name=vim
```

`salt.modules.state.`**`sls`**(*mods*, *test=None*, *exclude=None*, *queue=False*, *sync_mods=None*, ***kwargs*)

Execute the states in one or more SLS files

test Run states in test-only (dry-run) mode

pillar Custom Pillar values, passed as a dictionary of key-value pairs

```
salt '*' state.sls stuff pillar='{"foo": "bar"}'
```

Note: Values passed this way will override existing Pillar values set via `pillar_roots` or an external Pillar source. Pillar values that are not included in the kwarg will not be overwritten.

Changed in version 2016.3.0: GPG-encrypted CLI Pillar data is now supported via the GPG renderer. See *here* for details.

pillar_enc Specify which renderer to use to decrypt encrypted data located within the `pillar` value. Currently, only `gpg` is supported.

New in version 2016.3.0.

exclude Exclude specific states from execution. Accepts a list of sls names, a comma-separated string of sls names, or a list of dictionaries containing `sls` or `id` keys. Glob-patterns may be used to match multiple states.

```
salt '*' state.sls foo,bar,baz exclude=bar,baz
salt '*' state.sls foo,bar,baz exclude=ba*
salt '*' state.sls foo,bar,baz exclude="[{'id': 'id_to_exclude'}, {'sls':
↪'sls_to_exclude'}]"
```

queue [False] Instead of failing immediately when another state run is in progress, queue the new state run to begin running once the other has finished.

This option starts a new thread for each queued state run, so use this option sparingly.

concurrent [False] Execute state runs concurrently instead of serially

Warning: This flag is potentially dangerous. It is designed for use when multiple state runs can safely be run at the same time. Do *not* use this flag for performance optimization.

saltenv Specify a salt fileserver environment to be used when applying states

Changed in version 0.17.0: Argument name changed from `env` to `saltenv`.

Changed in version 2014.7.0: If no saltenv is specified, the minion config will be checked for an `environment` parameter and if found, it will be used. If none is found, `base` will be used. In prior releases, the minion config was not checked and `base` would always be assumed when the saltenv was not explicitly set.

pillarenv Specify a Pillar environment to be used when applying states. This can also be set in the minion config file using the *`pillarenv`* option. When neither the *`pillarenv`* minion config option nor this CLI argument is used, all Pillar environments will be merged together.

localconfig Optionally, instead of using the minion config, load minion opts from the file specified by this argument, and then merge them with the options from the minion config. This functionality allows for specific states to be run with their own custom minion configuration, including different pillars, file_roots, etc.

mock The mock option allows for the state run to execute without actually calling any states. This then returns a mocked return which will show the requisite ordering as well as fully validate the state run.

New in version 2015.8.4.

sync_mods If specified, the desired custom module types will be synced prior to running the SLS files:

```
salt '*' state.sls stuff sync_mods=states,modules
salt '*' state.sls stuff sync_mods=all
```

New in version 2017.7.8,2018.3.3,2019.2.0.

CLI Example:

```
# Run the states configured in salt://example.sls (or salt://example/init.sls)
salt '*' state.apply example

# Run the states configured in salt://core.sls (or salt://core/init.sls)
# and salt://edit/vim.sls (or salt://edit/vim/init.sls)
salt '*' state.sls core,edit.vim

# Run the states configured in a more deeply nested directory such as salt://my/
↪nested/state.sls (or salt://my/nested/state/init.sls)
```

(continues on next page)

(continued from previous page)

```
salt '*' state.sls my.nested.state

salt '*' state.sls core exclude="[{'id': 'id_to_exclude'}, {'sls': 'sls_to_exclude
↪'}]"
salt '*' state.sls myslsfile pillar="{foo: 'Foo!', bar: 'Bar!'}"
```

`salt.modules.state.`**`sls_exists`**(*mods*, *test=None*, *queue=False*, ***kwargs*)

Tests for the existance the of a specific SLS or list of SLS files on the master. Similar to *`state.show_sls`*, rather than returning state details, returns True or False. The default environment is `base`, use `saltenv` to specify a different environment.

New in version 2019.2.0.

saltenv Specify a salt fileserver environment from which to look for the SLS files specified in the `mods` argument

CLI Example:

```
salt '*' state.sls_exists core,edit.vim saltenv=dev
```

`salt.modules.state.`**`sls_id`**(*id_*, *mods*, *test=None*, *queue=False*, ***kwargs*)

Call a single ID from the named module(s) and handle all requisites

The state ID comes *before* the module ID(s) on the command line.

id ID to call

mods Comma-delimited list of modules to search for given id and its requisites

New in version 2014.7.0.

saltenv [base] Specify a salt fileserver environment to be used when applying states

pillarenv Specify a Pillar environment to be used when applying states. This can also be set in the minion config file using the *`pillarenv`* option. When neither the *`pillarenv`* minion config option nor this CLI argument is used, all Pillar environments will be merged together.

pillar Custom Pillar values, passed as a dictionary of key-value pairs

```
salt '*' state.sls_id my_state my_module pillar='{"foo": "bar"}'
```

Note: Values passed this way will override existing Pillar values set via `pillar_roots` or an external Pillar source. Pillar values that are not included in the kwarg will not be overwritten.

New in version 2018.3.0.

CLI Example:

```
salt '*' state.sls_id my_state my_module

salt '*' state.sls_id my_state my_module,a_common_module
```

`salt.modules.state.`**`soft_kill`**(*jid*, *state_id=None*)

Set up a state run to die before executing the given state id, this instructs a running state to safely exit at a given state id. This needs to pass in the jid of the running state. If a state_id is not passed then the jid referenced will be safely exited at the beginning of the next state run.

The given state id is the id got a given state execution, so given a state that looks like this:

```
vim:
  pkg.installed: []
```

The state_id to pass to *soft_kill* is *vim*

CLI Examples:

```
salt '*' state.soft_kill 20171130110407769519
salt '*' state.soft_kill 20171130110407769519 vim
```

`salt.modules.state.`**`template`**(*tem*, *queue=False*, ***kwargs*)

Execute the information stored in a template file on the minion.

This function does not ask a master for a SLS file to render but instead directly processes the file at the provided path on the minion.

CLI Example:

```
salt '*' state.template '<Path to template on the minion>'
```

`salt.modules.state.`**`template_str`**(*tem*, *queue=False*, ***kwargs*)

Execute the information stored in a string from an sls template

CLI Example:

```
salt '*' state.template_str '<Template String>'
```

`salt.modules.state.`**`top`**(*topfn*, *test=None*, *queue=False*, ***kwargs*)

Execute a specific top file instead of the default. This is useful to apply configurations from a different environment (for example, dev or prod), without modifying the default top file.

queue [False] Instead of failing immediately when another state run is in progress, queue the new state run to begin running once the other has finished.

This option starts a new thread for each queued state run, so use this option sparingly.

saltenv Specify a salt fileserver environment to be used when applying states

pillarenv Specify a Pillar environment to be used when applying states. This can also be set in the minion config file using the *`pillarenv`* option. When neither the *`pillarenv`* minion config option nor this CLI argument is used, all Pillar environments will be merged together.

New in version 2017.7.0.

CLI Example:

```
salt '*' state.top reverse_top.sls
salt '*' state.top prod_top.sls exclude=sls_to_exclude
salt '*' state.top dev_top.sls exclude="[{'id': 'id_to_exclude'}, {'sls': 'sls_to_
↪exclude'}]"
```

21.9.409 salt.modules.status

Module for returning various status data about a minion. These data can be useful for compiling into stats later.

`salt.modules.status.`**`all_status`**()

Return a composite of all status data and info for this minion. Warning: There is a LOT here!

CLI Example:

```
salt '*' status.all_status
```

`salt.modules.status.`**`cpuinfo`**()

Changed in version 2016.3.2: Return the CPU info for this minion

Changed in version 2016.11.4: Added support for AIX

Changed in version 2018.3.0: Added support for NetBSD and OpenBSD

CLI Example:

```
salt '*' status.cpuinfo
```

salt.modules.status.**cpustats**()

Return the CPU stats for this minion

Changed in version 2016.11.4: Added support for AIX

Changed in version 2018.3.0: Added support for OpenBSD

CLI Example:

```
salt '*' status.cpustats
```

salt.modules.status.**custom**()

Return a custom composite of status data and info for this minion, based on the minion config file. An example config like might be:

```
status.cpustats.custom: [ 'cpu', 'ctxt', 'btime', 'processes' ]
```

Where status refers to status.py, cpustats is the function where we get our data, and custom is this function It is followed by a list of keys that we want returned.

This function is meant to replace all_status(), which returns anything and everything, which we probably don't want.

By default, nothing is returned. Warning: Depending on what you include, there can be a LOT here!

CLI Example:

```
salt '*' status.custom
```

salt.modules.status.**diskstats**()

Changed in version 2016.3.2: Return the disk stats for this minion

Changed in version 2016.11.4: Added support for AIX

CLI Example:

```
salt '*' status.diskstats
```

salt.modules.status.**diskusage**(**args*)

Return the disk usage for this minion

Usage:

```
salt '*' status.diskusage [paths and/or filesystem types]
```

CLI Example:

```
salt '*' status.diskusage         # usage for all filesystems
salt '*' status.diskusage / /tmp  # usage for / and /tmp
salt '*' status.diskusage ext?    # usage for ext[234] filesystems
salt '*' status.diskusage / ext?  # usage for / and all ext filesystems
```

salt.modules.status.**loadavg**()

Return the load averages for this minion

Changed in version 2016.11.4: Added support for AIX

CLI Example:

```
salt '*' status.loadavg

:raises CommandExecutionError: If the system cannot report loadaverages to Python
```

salt.modules.status.**master**(*master=None*, *connected=True*)

New in version 2014.7.0.

Return the connection status with master. Fire an event if the connection to master is not as expected. This function is meant to be run via a scheduled job from the minion. If master_ip is an FQDN/Hostname, it must be resolvable to a valid IPv4 address.

Changed in version 2016.11.4: Added support for AIX

CLI Example:

```
salt '*' status.master
```

salt.modules.status.**meminfo**()

Return the memory info for this minion

Changed in version 2016.11.4: Added support for AIX

Changed in version 2018.3.0: Added support for OpenBSD

CLI Example:

```
salt '*' status.meminfo
```

salt.modules.status.**netdev**()

Changed in version 2016.3.2: Return the network device stats for this minion

Changed in version 2016.11.4: Added support for AIX

CLI Example:

```
salt '*' status.netdev
```

salt.modules.status.**netstats**()

Return the network stats for this minion

Changed in version 2016.11.4: Added support for AIX

Changed in version 2018.3.0: Added support for OpenBSD

CLI Example:

```
salt '*' status.netstats
```

salt.modules.status.**nproc**()

Return the number of processing units available on this system

Changed in version 2016.11.4: Added support for AIX

Changed in version 2018.3.0: Added support for Darwin, FreeBSD and OpenBSD

CLI Example:

```
salt '*' status.nproc
```

salt.modules.status.**pid**(*sig*)

Return the PID or an empty string if the process is running or not. Pass a signature to use to find the process via ps. Note you can pass a Python-compatible regular expression to return all pids of processes matching the regexp.

Changed in version 2016.11.4: Added support for AIX

CLI Example:

```
salt '*' status.pid <sig>
```

`salt.modules.status.`**`ping_master`**`(`*master*`)`

New in version 2016.3.0.

Sends ping request to the given master. Fires '__master_failback' event on success. Returns bool result.

CLI Example:

```
salt '*' status.ping_master localhost
```

`salt.modules.status.`**`procs`**`()`

Return the process data

Changed in version 2016.11.4: Added support for AIX

CLI Example:

```
salt '*' status.procs
```

`salt.modules.status.`**`proxy_reconnect`**`(`*proxy_name, opts=None*`)`

Forces proxy minion reconnection when not alive.

proxy_name The virtual name of the proxy module.

opts: None Opts dictionary. Not intended for CLI usage.

CLI Example:

salt '*' status.proxy_reconnect rest_sample

`salt.modules.status.`**`time_`**`(`*format='%A, %d. %B %Y %I:%M%p'*`)`

New in version 2016.3.0.

Return the current time on the minion, formatted based on the format parameter.

Default date format: Monday, 27. July 2015 07:55AM

CLI Example:

```
salt '*' status.time

salt '*' status.time '%s'
```

`salt.modules.status.`**`uptime`**`()`

Return the uptime for this system.

Changed in version 2015.8.9: The uptime function was changed to return a dictionary of easy-to-read key/value pairs containing uptime information, instead of the output from a `cmd.run` call.

Changed in version 2016.11.0: Support for OpenBSD, FreeBSD, NetBSD, MacOS, and Solaris

Changed in version 2016.11.4: Added support for AIX

CLI Example:

```
salt '*' status.uptime
```

`salt.modules.status.`**`version`**`()`

Return the system version for this minion

Changed in version 2016.11.4: Added support for AIX

Changed in version 2018.3.0: Added support for OpenBSD

CLI Example:

```
salt '*' status.version
```

`salt.modules.status.`**`vmstats`**`()`

Changed in version 2016.3.2: Return the virtual memory stats for this minion

Changed in version 2016.11.4: Added support for AIX

CLI Example:

```
salt '*' status.vmstats
```

`salt.modules.status.`**`w`**`()`

Return a list of logged in users for this minion, using the w command

CLI Example:

```
salt '*' status.w
```

21.9.410 salt.modules.statuspage

StatusPage

Handle requests for the StatusPage API.

In the minion configuration file, the following block is required:

```
statuspage:
  api_key: <API_KEY>
  page_id: <PAGE_ID>
```

New in version 2017.7.0.

`salt.modules.statuspage.`**`create`**`(`*endpoint='incidents', api_url=None, page_id=None, api_key=None, api_version=None, **kwargs*`)`

Insert a new entry under a specific endpoint.

endpoint: **incidents** Insert under this specific endpoint.

page_id Page ID. Can also be specified in the config file.

api_key API key. Can also be specified in the config file.

api_version: **1** API version. Can also be specified in the config file.

api_url Custom API URL in case the user has a StatusPage service running in a custom environment.

CLI Example:

```
salt 'minion' statuspage.create endpoint='components' name='my component' group_
↪id='993vgplshj12'
```

Example output:

```
minion:
    ----------
    comment:
    out:
        ----------
        created_at:
            2017-01-05T19:35:27.135Z
        description:
            None
        group_id:
            993vgplshj12
        id:
            mjkmtt5lhdgc
        name:
            my component
        page_id:
            ksdhgfyiuhaa
        position:
            7
        status:
            operational
        updated_at:
            2017-01-05T19:35:27.135Z
    result:
        True
```

`salt.modules.statuspage.`**`delete`**(*endpoint='incidents', id=None, api_url=None, page_id=None, api_key=None, api_version=None*)

Remove an entry from an endpoint.

endpoint: incidents Request a specific endpoint.

page_id Page ID. Can also be specified in the config file.

api_key API key. Can also be specified in the config file.

api_version: 1 API version. Can also be specified in the config file.

api_url Custom API URL in case the user has a StatusPage service running in a custom environment.

CLI Example:

```
salt 'minion' statuspage.delete endpoint='components' id='ftgks51sfs2d'
```

Example output:

```
minion:
    ----------
    comment:
    out:
        None
    result:
        True
```

`salt.modules.statuspage.`**`retrieve`**(*endpoint='incidents', api_url=None, page_id=None, api_key=None, api_version=None*)

Retrieve a specific endpoint from the Statuspage API.

endpoint: incidents Request a specific endpoint.

page_id Page ID. Can also be specified in the config file.

api_key API key. Can also be specified in the config file.

api_version: 1 API version. Can also be specified in the config file.

api_url Custom API URL in case the user has a StatusPage service running in a custom environment.

CLI Example:

```
salt 'minion' statuspage.retrieve components
```

Example output:

```
minion:
    ----------
    comment:
    out:
        |_
          ----------
          backfilled:
              False
          created_at:
              2015-01-26T20:25:02.702Z
          id:
              kh2qwjbheqdc36
          impact:
              major
          impact_override:
              None
          incident_updates:
              |_
                ----------
                affected_components:
                    None
                body:
                    We are currently investigating this issue.
                created_at:
                    2015-01-26T20:25:02.849Z
                display_at:
                    2015-01-26T20:25:02.849Z
                id:
                    zvx7xz2z5skr
                incident_id:
                    kh2qwjbheqdc36
                status:
                    investigating
                twitter_updated_at:
                    None
                updated_at:
                    2015-01-26T20:25:02.849Z
                wants_twitter_update:
                    False
          monitoring_at:
              None
          name:
              just testing some stuff
          page_id:
              ksdhgfyiuhaa
          postmortem_body:
              None
          postmortem_body_last_updated_at:
              None
          postmortem_ignored:
              False
          postmortem_notified_subscribers:
              False
```

(continues on next page)

(continued from previous page)

```
            postmortem_notified_twitter:
                False
            postmortem_published_at:
                None
            resolved_at:
                None
            scheduled_auto_completed:
                False
            scheduled_auto_in_progress:
                False
            scheduled_for:
                None
            scheduled_remind_prior:
                False
            scheduled_reminded_at:
                None
            scheduled_until:
                None
            shortlink:
                http://stspg.io/voY
            status:
                investigating
            updated_at:
                2015-01-26T20:25:13.379Z
    result:
        True
```

salt.modules.statuspage.**update**(*endpoint='incidents'*, *id=None*, *api_url=None*, *page_id=None*, *api_key=None*, *api_version=None*, ***kwargs*)

Update attribute(s) of a specific endpoint.

id The unique ID of the enpoint entry.

endpoint: incidents Endpoint name.

page_id Page ID. Can also be specified in the config file.

api_key API key. Can also be specified in the config file.

api_version: 1 API version. Can also be specified in the config file.

api_url Custom API URL in case the user has a StatusPage service running in a custom environment.

CLI Example:

```
salt 'minion' statuspage.update id=dz959yz2nd4l status=resolved
```

Example output:

```
minion:
    ----------
    comment:
    out:
        ----------
        created_at:
            2017-01-03T15:25:30.718Z
        description:
            None
        group_id:
            993vgplshj12
        id:
            dz959yz2nd4l
        name:
```

(continues on next page)

(continued from previous page)

```
            Management Portal
        page_id:
            xzwjjdw87vpf
        position:
            11
        status:
            resolved
        updated_at:
            2017-01-05T15:34:27.676Z
    result:
        True
```

21.9.411 salt.modules.supervisord

Provide the service module for system supervisord or supervisord in a virtualenv

`salt.modules.supervisord.`**`add`**`(`*name*, *user=None*, *conf_file=None*, *bin_env=None*`)`

Activates any updates in config for process/group.

user user to run supervisorctl as

conf_file path to supervisord config file

bin_env path to supervisorctl bin or path to virtualenv with supervisor installed

CLI Example:

```
salt '*' supervisord.add <name>
```

`salt.modules.supervisord.`**`custom`**`(`*command*, *user=None*, *conf_file=None*, *bin_env=None*`)`

Run any custom supervisord command

user user to run supervisorctl as

conf_file path to supervisord config file

bin_env path to supervisorctl bin or path to virtualenv with supervisor installed

CLI Example:

```
salt '*' supervisord.custom "mstop '*gunicorn*'"
```

`salt.modules.supervisord.`**`options`**`(`*name*, *conf_file=None*`)`

New in version 2014.1.0.

Read the config file and return the config options for a given process

name Name of the configured process

conf_file path to supervisord config file

CLI Example:

```
salt '*' supervisord.options foo
```

`salt.modules.supervisord.`**`remove`**`(`*name*, *user=None*, *conf_file=None*, *bin_env=None*`)`

Removes process/group from active config

user user to run supervisorctl as

conf_file path to supervisord config file

bin_env path to supervisorctl bin or path to virtualenv with supervisor installed

CLI Example:

```
salt '*' supervisord.remove <name>
```

`salt.modules.supervisord.`**`reread`**`(`*user=None*, *conf_file=None*, *bin_env=None*`)`

Reload the daemon's configuration files

user user to run supervisorctl as

conf_file path to supervisord config file

bin_env path to supervisorctl bin or path to virtualenv with supervisor installed

CLI Example:

```
salt '*' supervisord.reread
```

`salt.modules.supervisord.`**`restart`**(*name='all', user=None, conf_file=None, bin_env=None*)

Restart the named service. Process group names should not include a trailing asterisk.

user user to run supervisorctl as

conf_file path to supervisord config file

bin_env path to supervisorctl bin or path to virtualenv with supervisor installed

CLI Example:

```
salt '*' supervisord.restart <service>
salt '*' supervisord.restart <group>:
```

`salt.modules.supervisord.`**`start`**(*name='all', user=None, conf_file=None, bin_env=None*)

Start the named service. Process group names should not include a trailing asterisk.

user user to run supervisorctl as

conf_file path to supervisord config file

bin_env path to supervisorctl bin or path to virtualenv with supervisor installed

CLI Example:

```
salt '*' supervisord.start <service>
salt '*' supervisord.start <group>:
```

`salt.modules.supervisord.`**`status`**(*name=None, user=None, conf_file=None, bin_env=None*)

List programs and its state

user user to run supervisorctl as

conf_file path to supervisord config file

bin_env path to supervisorctl bin or path to virtualenv with supervisor installed

CLI Example:

```
salt '*' supervisord.status
```

`salt.modules.supervisord.`**`status_raw`**(*name=None, user=None, conf_file=None, bin_env=None*)

Display the raw output of status

user user to run supervisorctl as

conf_file path to supervisord config file

bin_env path to supervisorctl bin or path to virtualenv with supervisor installed

CLI Example:

```
salt '*' supervisord.status_raw
```

`salt.modules.supervisord.`**`stop`**(*name='all', user=None, conf_file=None, bin_env=None*)

Stop the named service. Process group names should not include a trailing asterisk.

user user to run supervisorctl as

conf_file path to supervisord config file

bin_env path to supervisorctl bin or path to virtualenv with supervisor installed

CLI Example:

```
salt '*' supervisord.stop <service>
salt '*' supervisord.stop <group>:
```

`salt.modules.supervisord.`**`update`**`(`*user=None*, *conf_file=None*, *bin_env=None*, *name=None*`)`

Reload config and add/remove/update as necessary

user user to run supervisorctl as

conf_file path to supervisord config file

bin_env path to supervisorctl bin or path to virtualenv with supervisor installed

name name of the process group to update. if none then update any process group that has changes

CLI Example:

```
salt '*' supervisord.update
```

21.9.412 salt.modules.suse_apache

Support for Apache

Please note: The functions in here are SUSE-specific. Placing them in this separate file will allow them to load only on SUSE systems, while still loading under the `apache` namespace.

`salt.modules.suse_apache.`**`a2dismod`**`(`*mod*`)`

Runs a2dismod for the given mod.

CLI Example:

```
salt '*' apache.a2dismod vhost_alias
```

`salt.modules.suse_apache.`**`a2enmod`**`(`*mod*`)`

Runs a2enmod for the given mod.

CLI Example:

```
salt '*' apache.a2enmod vhost_alias
```

`salt.modules.suse_apache.`**`check_mod_enabled`**`(`*mod*`)`

Checks to see if the specific apache mod is enabled.

This will only be functional on operating systems that support *a2enmod -l* to list the enabled mods.

CLI Example:

```
salt '*' apache.check_mod_enabled status
```

21.9.413 salt.modules.svn

Subversion SCM

`salt.modules.svn.`**`add`**`(`*cwd*, *targets*, *user=None*, *username=None*, *password=None*, **opts*`)`

Add files to be tracked by the Subversion working-copy checkout

cwd The path to the Subversion repository

targets [None] files and directories to pass to the command as arguments

user [None] Run svn as a user other than what the minion runs as

username [None] Connect to the Subversion server as another user

password [None] Connect to the Subversion server with this password

New in version 0.17.0.

CLI Example:

```
salt '*' svn.add /path/to/repo /path/to/new/file
```

`salt.modules.svn.`**`checkout`**`(`*cwd*, *remote*, *target=None*, *user=None*, *username=None*, *password=None*, **opts*`)`

Download a working copy of the remote Subversion repository directory or file

cwd The path to the Subversion repository

remote [None] URL to checkout

target [None] The name to give the file or directory working copy Default: svn uses the remote basename

user [None] Run svn as a user other than what the minion runs as

username [None] Connect to the Subversion server as another user

password [None] Connect to the Subversion server with this password

New in version 0.17.0.

CLI Example:

```
salt '*' svn.checkout /path/to/repo svn://remote/repo
```

`salt.modules.svn.`**`commit`**`(`*cwd*, *targets=None*, *msg=None*, *user=None*, *username=None*, *password=None*, **opts*`)`

Commit the current directory, files, or directories to the remote Subversion repository

cwd The path to the Subversion repository

targets [None] files and directories to pass to the command as arguments Default: svn uses '.'

msg [None] Message to attach to the commit log

user [None] Run svn as a user other than what the minion runs as

username [None] Connect to the Subversion server as another user

password [None] Connect to the Subversion server with this password

New in version 0.17.0.

CLI Example:

```
salt '*' svn.commit /path/to/repo
```

`salt.modules.svn.`**`diff`**`(`*cwd*, *targets=None*, *user=None*, *username=None*, *password=None*, **opts*`)`

Return the diff of the current directory, files, or directories from the remote Subversion repository

cwd The path to the Subversion repository

targets [None] files and directories to pass to the command as arguments Default: svn uses '.'

user [None] Run svn as a user other than what the minion runs as

username [None] Connect to the Subversion server as another user

password [None] Connect to the Subversion server with this password

New in version 0.17.0.

CLI Example:

```
salt '*' svn.diff /path/to/repo
```

`salt.modules.svn.`**`export`**`(`*cwd*, *remote*, *target=None*, *user=None*, *username=None*, *password=None*, *revision='HEAD'*, **opts*`)`

Create an unversioned copy of a tree.

cwd The path to the Subversion repository

remote [None] URL and path to file or directory checkout

target [None] The name to give the file or directory working copy Default: svn uses the remote basename

user [None] Run svn as a user other than what the minion runs as

username [None] Connect to the Subversion server as another user

password [None] Connect to the Subversion server with this password

New in version 0.17.0.

CLI Example:

```
salt '*' svn.export /path/to/repo svn://remote/repo
```

salt.modules.svn.**info**(*cwd*, *targets=None*, *user=None*, *username=None*, *password=None*, *fmt='str'*)

Display the Subversion information from the checkout.

cwd The path to the Subversion repository

targets [None] files, directories, and URLs to pass to the command as arguments svn uses '.' by default

user [None] Run svn as a user other than what the minion runs as

username [None] Connect to the Subversion server as another user

password [None] Connect to the Subversion server with this password

New in version 0.17.0.

fmt [str] How to fmt the output from info. (str, xml, list, dict)

CLI Example:

```
salt '*' svn.info /path/to/svn/repo
```

salt.modules.svn.**remove**(*cwd*, *targets*, *msg=None*, *user=None*, *username=None*, *password=None*, **opts*)

Remove files and directories from the Subversion repository

cwd The path to the Subversion repository

targets [None] files, directories, and URLs to pass to the command as arguments

msg [None] Message to attach to the commit log

user [None] Run svn as a user other than what the minion runs as

username [None] Connect to the Subversion server as another user

password [None] Connect to the Subversion server with this password

New in version 0.17.0.

CLI Example:

```
salt '*' svn.remove /path/to/repo /path/to/repo/remove
```

salt.modules.svn.**status**(*cwd*, *targets=None*, *user=None*, *username=None*, *password=None*, **opts*)

Display the status of the current directory, files, or directories in the Subversion repository

cwd The path to the Subversion repository

targets [None] files, directories, and URLs to pass to the command as arguments Default: svn uses '.'

user [None] Run svn as a user other than what the minion runs as

username [None] Connect to the Subversion server as another user

password [None] Connect to the Subversion server with this password

New in version 0.17.0.

CLI Example:

```
salt '*' svn.status /path/to/repo
```

salt.modules.svn.**switch**(*cwd*, *remote*, *target=None*, *user=None*, *username=None*, *password=None*, **opts*)

New in version 2014.1.0.

Switch a working copy of a remote Subversion repository directory

cwd The path to the Subversion repository

remote [None] URL to switch

target [None] The name to give the file or directory working copy Default: svn uses the remote basename

user [None] Run svn as a user other than what the minion runs as

username [None] Connect to the Subversion server as another user

password [None] Connect to the Subversion server with this password

CLI Example:

```
salt '*' svn.switch /path/to/repo svn://remote/repo
```

`salt.modules.svn.`**`update`**(*cwd, targets=None, user=None, username=None, password=None, *opts*)

Update the current directory, files, or directories from the remote Subversion repository

cwd The path to the Subversion repository

targets [None] files and directories to pass to the command as arguments Default: svn uses '.'

user [None] Run svn as a user other than what the minion runs as

password [None] Connect to the Subversion server with this password

New in version 0.17.0.

username [None] Connect to the Subversion server as another user

CLI Example:

```
salt '*' svn.update /path/to/repo
```

21.9.414 salt.modules.swarm

Docker Swarm Module using Docker's Python SDK

codeauthor Tyler Jones <jonestyler806@gmail.com>

New in version 2018.3.0.

The Docker Swarm Module is used to manage and create Docker Swarms.

Dependencies

- Docker installed on the host
- Docker python sdk >= 2.5.1

Docker Python SDK

```
pip install -U docker
```

More information: https://docker-py.readthedocs.io/en/stable/

`salt.modules.swarm.`**`joinswarm`**(*remote_addr=<class 'int'>, listen_addr=<class 'int'>, token=<class 'str'>*)

Join a Swarm Worker to the cluster

remote_addr The manager node you want to connect to for the swarm

listen_addr Listen address used for inter-manager communication if the node gets promoted to manager, as well as determining the networking interface used for the VXLAN Tunnel Endpoint (VTEP)

token Either the manager join token or the worker join token. You can get the worker or manager token via `salt '*' swarm.swarm_tokens`

CLI Example:

```
salt '*' swarm.joinswarm remote_addr=192.168.50.10 listen_addr='0.0.0.0'
↪    token='SWMTKN-1-64tux2g0701r84ofq93zppcih0pe081akq45owe9ts61f30x4t-
↪06trjugdu7x2z47j938s54il'
```

`salt.modules.swarm.`**`leave_swarm`**(*force=<class 'bool'>*)

Force the minion to leave the swarm

force Will force the minion/worker/manager to leave the swarm

CLI Example:

```
salt '*' swarm.leave_swarm force=False
```

`salt.modules.swarm.`**`node_ls`**(*server=<class ’str’>*)

Displays Information about Swarm Nodes with passing in the server

server The minion/server name

CLI Example:

```
salt '*' swarm.node_ls server=minion1
```

`salt.modules.swarm.`**`remove_node`**(*node_id=<class ’str’>, force=<class ’bool’>*)

Remove a node from a swarm and the target needs to be a swarm manager

node_id The node id from the return of swarm.node_ls

force Forcefully remove the node/minion from the service

CLI Example:

```
salt '*' swarm.remove_node node_id=z4gjbe9rwmqahc2a91snvolm5 force=false
```

`salt.modules.swarm.`**`remove_service`**(*service=<class ’str’>*)

Remove Swarm Service

service The name of the service

CLI Example:

```
salt '*' swarm.remove_service service=Test_Service
```

`salt.modules.swarm.`**`service_create`**(*image=<class ’str’>, name=<class ’str’>, command=<class ’str’>, hostname=<class ’str’>, replicas=<class ’int’>, target_port=<class ’int’>, published_port=<class ’int’>*)

Create Docker Swarm Service Create

image The docker image

name Is the service name

command The docker command to run in the container at launch

hostname The hostname of the containers

replicas How many replicas you want running in the swarm

target_port The target port on the container

published_port port thats published on the host/os

CLI Example:

```
salt '*' swarm.service_create image=httpd name=Test_Service
↪command=None hostname=salthttpd replicas=6 target_port=80 published_port=80
```

`salt.modules.swarm.`**`swarm_init`**(*advertise_addr=<class ’str’>, listen_addr=<class ’int’>, force_new_cluster=<class ’bool’>*)

Initalize Docker on Minion as a Swarm Manager

advertise_addr The ip of the manager

listen_addr Listen address used for inter-manager communication, as well as determining the networking interface used for the VXLAN Tunnel Endpoint (VTEP). This can either be an address/port combination in the form 192.168.1.1:4567, or an interface followed by a port number, like eth0:4567

force_new_cluster Force a new cluster if True is passed

CLI Example:

```
salt '*' swarm.swarm_init advertise_addr='192.168.50.10' listen_addr='0.0.0.0'
↪force_new_cluster=False
```

`salt.modules.swarm.`**`swarm_service_info`**`(`*service_name=<class 'str'>*`)`

Swarm Service Information

service_name The name of the service that you want information on about the service

CLI Example:

```
salt '*' swarm.swarm_service_info service_name=Test_Service
```

`salt.modules.swarm.`**`swarm_tokens`**`()`

Get the Docker Swarm Manager or Worker join tokens

CLI Example:

```
salt '*' swarm.swarm_tokens
```

`salt.modules.swarm.`**`update_node`**`(`*availability=<class 'str'>, node_name=<class 'str'>, role=<class 'str'>, node_id=<class 'str'>, version=<class 'int'>*`)`

Updates docker swarm nodes/needs to target a manager node/minion

availability Drain or Active

node_name minion/node

role role of manager or worker

node_id The Id and that can be obtained via swarm.node_ls

version Is obtained by swarm.node_ls

CLI Example:

```
salt '*' docker_util.update_node availability=drain node_name=minion2 ␣
↪role=worker node_id=3k9x7t8m4pel9c0nqr3iajnzp version=19
```

21.9.415 salt.modules.swift

Module for handling OpenStack Swift calls Author: Anthony Stanton <anthony.stanton@gmail.com>

Inspired by the S3 and Nova modules

depends

- swiftclient Python module

configuration This module is not usable until the user, tenant, auth URL, and password or auth_key are specified either in a pillar or in the minion's config file. For example:

```
keystone.user: admin
keystone.tenant: admin
keystone.auth_url: 'http://127.0.0.1:5000/v2.0/'
keystone.password: verybadpass
# or
keystone.auth_key: 203802934809284k2j34lkj2l3kj43k
```

If configuration for multiple OpenStack accounts is required, they can be set up as different configuration profiles: For example:

```
openstack1:
  keystone.user: admin
  keystone.tenant: admin
  keystone.auth_url: 'http://127.0.0.1:5000/v2.0/'
  keystone.password: verybadpass
  # or
  keystone.auth_key: 203802934809284k2j34lkj2l3kj43k
```

(continues on next page)

(continued from previous page)

```
openstack2:
  keystone.user: admin
  keystone.tenant: admin
  keystone.auth_url: 'http://127.0.0.2:5000/v2.0/'
  keystone.password: verybadpass
  # or
  keystone.auth_key: 303802934809284k2j34lkj2l3kj43k
```

With this configuration in place, any of the swift functions can make use of a configuration profile by declaring it explicitly. For example:

```
salt '*' swift.get mycontainer myfile /tmp/file profile=openstack1
```

NOTE: For Rackspace cloud files setting keystone.auth_version = 1 is recommended.

`salt.modules.swift.`**`delete`**`(`*cont, path=None, profile=None*`)`

Delete a container, or delete an object from a container.

CLI Example to delete a container:

```
salt myminion swift.delete mycontainer
```

CLI Example to delete an object from a container:

```
salt myminion swift.delete mycontainer remoteobject
```

`salt.modules.swift.`**`get`**`(`*cont=None, path=None, local_file=None, return_bin=False, profile=None*`)`

List the contents of a container, or return an object from a container. Set return_bin to True in order to retrieve an object wholesale. Otherwise, Salt will attempt to parse an XML response.

CLI Example to list containers:

```
salt myminion swift.get
```

CLI Example to list the contents of a container:

```
salt myminion swift.get mycontainer
```

CLI Example to return the binary contents of an object:

```
salt myminion swift.get mycontainer myfile.png return_bin=True
```

CLI Example to save the binary contents of an object to a local file:

```
salt myminion swift.get mycontainer myfile.png local_file=/tmp/myfile.png
```

`salt.modules.swift.`**`head`**`()`

`salt.modules.swift.`**`put`**`(`*cont, path=None, local_file=None, profile=None*`)`

Create a new container, or upload an object to a container.

CLI Example to create a container:

```
salt myminion swift.put mycontainer
```

CLI Example to upload an object to a container:

```
salt myminion swift.put mycontainer remotepath local_file=/path/to/file
```

21.9.416 salt.modules.sysbench

The 'sysbench' module is used to analyze the performance of the minions, right from the master! It measures various system parameters such as CPU, Memory, File I/O, Threads and Mutex.

salt.modules.sysbench.**cpu**()

Tests for the CPU performance of minions.

CLI Examples:

```
salt '*' sysbench.cpu
```

salt.modules.sysbench.**fileio**()

This tests for the file read and write operations Various modes of operations are

- sequential write
- sequential rewrite
- sequential read
- random read
- random write
- random read and write

The test works with 32 files with each file being 1Gb in size The test consumes a lot of time. Be patient!

CLI Examples:

```
salt '*' sysbench.fileio
```

salt.modules.sysbench.**memory**()

This tests the memory for read and write operations.

CLI Examples:

```
salt '*' sysbench.memory
```

salt.modules.sysbench.**mutex**()

Tests the implementation of mutex

CLI Examples:

```
salt '*' sysbench.mutex
```

salt.modules.sysbench.**ping**()

salt.modules.sysbench.**threads**()

This tests the performance of the processor's scheduler

CLI Example:

```
salt '*' sysbench.threads
```

21.9.417 salt.modules.sysfs module

Module for interfacing with SysFS

See also:

https://www.kernel.org/doc/Documentation/filesystems/sysfs.txt

New in version 2016.3.0.

`salt.modules.sysfs.`**`attr`**`(`*key*, *value=None*`)`

Access/write a SysFS attribute. If the attribute is a symlink, it's destination is returned

Returns value or bool

CLI example:

```
salt '*' sysfs.attr block/sda/queue/logical_block_size
```

`salt.modules.sysfs.`**`interfaces`**`(`*root*`)`

Generate a dictionary with all available interfaces relative to root. Symlinks are not followed.

CLI example:

```
salt '*' sysfs.interfaces block/bcache0/bcache
```

Output example:

```
{
   "r": [
     "state",
     "partial_stripes_expensive",
     "writeback_rate_debug",
     "stripe_size",
     "dirty_data",
     "stats_total/cache_hits",
     "stats_total/cache_bypass_misses",
     "stats_total/bypassed",
     "stats_total/cache_readaheads",
     "stats_total/cache_hit_ratio",
     "stats_total/cache_miss_collisions",
     "stats_total/cache_misses",
     "stats_total/cache_bypass_hits",
   ],
   "rw": [
     "writeback_rate",
     "writeback_rate_update_seconds",
     "cache_mode",
     "writeback_delay",
     "label",
     "writeback_running",
     "writeback_metadata",
     "running",
     "writeback_rate_p_term_inverse",
     "sequential_cutoff",
     "writeback_percent",
     "writeback_rate_d_term",
     "readahead"
   ],
   "w": [
     "stop",
     "clear_stats",
     "attach",
     "detach"
   ]
}
```

Note:

- 'r' interfaces are read-only
- 'w' interfaces are write-only (e.g. actions)
- 'rw' are interfaces that can both be read or written

`salt.modules.sysfs.`**`read`**`(`*key, root=''*`)`

Read from SysFS

Parameters **`key`** – file or path in SysFS; if key is a list then root will be prefixed on each key

Returns the full (tree of) SysFS attributes under key

CLI example:

```
salt '*' sysfs.read class/net/em1/statistics
```

`salt.modules.sysfs.`**`target`**`(`*key, full=True*`)`

Return the basename of a SysFS key path

Parameters

- **`key`** – the location to resolve within SysFS
- **`full`** – full path instead of basename

Returns fullpath or basename of path

CLI example:

```
salt '*' sysfs.read class/ttyS0
```

`salt.modules.sysfs.`**`write`**`(`*key, value*`)`

Write a SysFS attribute/action

CLI example:

```
salt '*' sysfs.write devices/system/cpu/cpu0/cpufreq/scaling_governor
↪'performance'
```

21.9.418 salt.modules.syslog_ng

Module for getting information about syslog-ng

maintainer Tibor Benke <btibi@sch.bme.hu>

maturity new

depends cmd

platform all

This module is capable of managing syslog-ng instances which were installed via a package manager or from source. Users can use a directory as a parameter in the case of most functions, which contains the syslog-ng and syslog-ng-ctl binaries.

Syslog-ng can be installed via a package manager or from source. In the latter case, the syslog-ng and syslog-ng-ctl binaries are not available from the PATH, so users should set location of the sbin directory with *`syslog_ng.set_binary_path`*.

Similarly, users can specify the location of the configuration file with *`syslog_ng.set_config_file`*, then the module will use it. If it is not set, syslog-ng uses the default configuration file.

`class` `salt.modules.syslog_ng.`**`Argument`**`(`*value=''*`)`

A TypedParameterValue has one or more Arguments. For example this can be the value of key_file.

Does not need examples.

build()

class salt.modules.syslog_ng.**Buildable**(*iterable*, *join_body_on=''*, *append_extra_newline=True*)

Base class of most classes, which have a build method.

It contains a common build function.

Does not need examples.

build()

Builds the textual representation of the whole configuration object with it's children.

build_body()

Builds the body of a syslog-ng configuration object.

build_header()

Builds the header of a syslog-ng configuration object.

build_tail()

Builds the tail of a syslog-ng configuration object.

class salt.modules.syslog_ng.**GivenStatement**(*value*, *add_newline=True*)

This statement returns a string without modification. It can be used to use existing configuration snippets.

Does not need examples.

build()

Builds the textual representation of the whole configuration object with it's children.

class salt.modules.syslog_ng.**NamedStatement**(*type*, *id=''*, *options=None*)

It represents a configuration statement, which has a name, e.g. a source.

Does not need examples.

class salt.modules.syslog_ng.**Option**(*type=''*, *params=None*)

A Statement class contains Option instances.

An instance of Option can represent a file(), tcp(), udp(), etc. option.

Does not need examples.

add_parameter(*param*)

build()

Builds the textual representation of the whole configuration object with it's children.

class salt.modules.syslog_ng.**Parameter**(*iterable=None*, *join_body_on=''*)

An Option has one or more Parameter instances.

Does not need examples.

class salt.modules.syslog_ng.**ParameterValue**(*iterable=None*, *join_body_on=''*)

A TypedParameter can have one or more values.

Does not need examples.

class salt.modules.syslog_ng.**SimpleParameter**(*value=''*)

A Parameter is a SimpleParameter, if it's just a simple type, like a string.

For example:

```
destination d_file {
    file(
        '/var/log/messages'
```

(continues on next page)

(continued from previous page)

```
    );
};
```

`/var/log/messages` is a SimpleParameter.

Does not need examples.

build()
Builds the textual representation of the whole configuration object with it's children.

class salt.modules.syslog_ng.**SimpleParameterValue**(*value=''*)
A ParameterValuem which holds a simple type, like a string or a number.

For example in ip(127.0.0.1) 127.0.0.1 is a SimpleParameterValue.

Does not need examples.

build()
Builds the textual representation of the whole configuration object with it's children.

class salt.modules.syslog_ng.**Statement**(*type*, *id=''*, *options=None*, *has_name=True*)
It represents a syslog-ng configuration statement, e.g. source, destination, filter.

Does not need examples.

add_child(*option*)

build_header()
Builds the header of a syslog-ng configuration object.

build_tail()
Builds the tail of a syslog-ng configuration object.

exception salt.modules.syslog_ng.**SyslogNgError**

class salt.modules.syslog_ng.**TypedParameter**(*type=''*, *values=None*)
A Parameter, which has a type:

```
destination d_tcp {
    tcp(
        ip(127.0.0.1)
    );
};
```

`ip(127.0.0.1)` is a TypedParameter.

Does not need examples.

add_value(*value*)

build()
Builds the textual representation of the whole configuration object with it's children.

class salt.modules.syslog_ng.**TypedParameterValue**(*type=''*, *arguments=None*)
We have to go deeper…

A TypedParameter can have a 'parameter', which also have a type. For example key_file and cert_file:

```
source demo_tls_source {
    tcp(
        ip(0.0.0.0)
        port(1999)
```

(continues on next page)

(continued from previous page)

```
        tls(
            key_file('/opt/syslog-ng/etc/syslog-ng/key.d/syslog-ng.key')
            cert_file('/opt/syslog-ng/etc/syslog-ng/cert.d/syslog-ng.cert')
        )
    );
};
```

Does not need examples.

add_argument(*arg*)

build()

Builds the textual representation of the whole configuration object with it's children.

class salt.modules.syslog_ng.**UnnamedStatement**(*type*, *options=None*)

It represents a configuration statement, which doesn't have a name, e.g. a log path.

Does not need examples.

salt.modules.syslog_ng.**config**(*name*, *config*, *write=True*)

Builds syslog-ng configuration. This function is intended to be used from the state module, users should not use it directly!

name : the id of the Salt document or it is the format of <statement name>.id config : the parsed YAML code write : if True, it writes the config into the configuration file, otherwise just returns it

CLI Example:

```
salt '*' syslog_ng.config name='s_local' config="[{'tcp':[{'ip':'127.0.0.1'},{
↪'port':1233}]}]"
```

salt.modules.syslog_ng.**config_test**(*syslog_ng_sbin_dir=None*, *cfgfile=None*)

Runs syntax check against cfgfile. If syslog_ng_sbin_dir is specified, it is added to the PATH during the test.

CLI Example:

```
salt '*' syslog_ng.config_test
salt '*' syslog_ng.config_test /home/user/install/syslog-ng/sbin
salt '*' syslog_ng.config_test /home/user/install/syslog-ng/sbin /etc/syslog-ng/
↪syslog-ng.conf
```

salt.modules.syslog_ng.**get_config_file**()

Returns the configuration directory, which contains syslog-ng.conf.

CLI Example:

```
salt '*' syslog_ng.get_config_file
```

salt.modules.syslog_ng.**modules**(*syslog_ng_sbin_dir=None*)

Returns the available modules. If syslog_ng_sbin_dir is specified, it is added to the PATH during the execution of the command syslog-ng.

CLI Example:

```
salt '*' syslog_ng.modules
salt '*' syslog_ng.modules /home/user/install/syslog-ng/sbin
```

salt.modules.syslog_ng.**reload_**(*name*)

Reloads syslog-ng. This function is intended to be used from states.

If *syslog_ng.set_config_file*, is called before, this function will use the set binary path.

CLI Example:

```
salt '*' syslog_ng.reload
```

salt.modules.syslog_ng.**set_binary_path**(*name*)

Sets the path, where the syslog-ng binary can be found. This function is intended to be used from states.

If syslog-ng is installed via a package manager, users don't need to use this function.

CLI Example:

```
salt '*' syslog_ng.set_binary_path name=/usr/sbin
```

salt.modules.syslog_ng.**set_config_file**(*name*)

Sets the configuration's name. This function is intended to be used from states.

CLI Example:

```
salt '*' syslog_ng.set_config_file name=/etc/syslog-ng
```

salt.modules.syslog_ng.**set_parameters**(*version=None*, *binary_path=None*, *config_file=None*, **args*, ***kwargs*)

Sets variables.

CLI Example:

```
salt '*' syslog_ng.set_parameters version='3.6'
salt '*' syslog_ng.set_parameters  binary_path=/home/user/install/syslog-ng/sbin
↪config_file=/home/user/install/syslog-ng/etc/syslog-ng.conf
```

salt.modules.syslog_ng.**start**(*name=None*, *user=None*, *group=None*, *chroot=None*, *caps=None*, *no_caps=False*, *pidfile=None*, *enable_core=False*, *fd_limit=None*, *verbose=False*, *debug=False*, *trace=False*, *yydebug=False*, *persist_file=None*, *control=None*, *worker_threads=None*)

Ensures, that syslog-ng is started via the given parameters. This function is intended to be used from the state module.

Users shouldn't use this function, if the service module is available on their system. If *syslog_ng.set_config_file*, is called before, this function will use the set binary path.

CLI Example:

```
salt '*' syslog_ng.start
```

salt.modules.syslog_ng.**stats**(*syslog_ng_sbin_dir=None*)

Returns statistics from the running syslog-ng instance. If syslog_ng_sbin_dir is specified, it is added to the PATH during the execution of the command syslog-ng-ctl.

CLI Example:

```
salt '*' syslog_ng.stats
salt '*' syslog_ng.stats /home/user/install/syslog-ng/sbin
```

salt.modules.syslog_ng.**stop**(*name=None*)

Kills syslog-ng. This function is intended to be used from the state module.

Users shouldn't use this function, if the service module is available on their system. If *syslog_ng.set_config_file* is called before, this function will use the set binary path.

CLI Example:

```
salt '*' syslog_ng.stop
```

`salt.modules.syslog_ng.`**`version`**(*syslog_ng_sbin_dir=None*)

Returns the version of the installed syslog-ng. If syslog_ng_sbin_dir is specified, it is added to the PATH during the execution of the command syslog-ng.

CLI Example:

```
salt '*' syslog_ng.version
salt '*' syslog_ng.version /home/user/install/syslog-ng/sbin
```

`salt.modules.syslog_ng.`**`write_config`**(*config*, *newlines=2*)

Writes the given parameter config into the config file. This function is intended to be used from states.

If `syslog_ng.set_config_file`, is called before, this function will use the set config file.

CLI Example:

```
salt '*' syslog_ng.write_config config='# comment'
```

`salt.modules.syslog_ng.`**`write_version`**(*name*)

Removes the previous configuration file, then creates a new one and writes the name line. This function is intended to be used from states.

If `syslog_ng.set_config_file`, is called before, this function will use the set config file.

CLI Example:

```
salt '*' syslog_ng.write_version name="3.6"
```

21.9.419 salt.modules.sysmod

The sys module provides information about the available functions on the minion

`salt.modules.sysmod.`**`argspec`**(*module="*)

Return the argument specification of functions in Salt execution modules.

CLI Example:

```
salt '*' sys.argspec pkg.install
salt '*' sys.argspec sys
salt '*' sys.argspec
```

Module names can be specified as globs.

New in version 2015.5.0.

```
salt '*' sys.argspec 'pkg.*'
```

`salt.modules.sysmod.`**`doc`**(**args*)

Return the docstrings for all modules. Optionally, specify a module or a function to narrow the selection.

The strings are aggregated into a single document on the master for easy reading.

Multiple modules/functions can be specified.

CLI Example:

```
salt '*' sys.doc
salt '*' sys.doc sys
salt '*' sys.doc sys.doc
salt '*' sys.doc network.traceroute user.info
```

Modules can be specified as globs.

New in version 2015.5.0.

```
salt '*' sys.doc 'sys.*'
salt '*' sys.doc 'sys.list_*'
```

`salt.modules.sysmod.`**`list_functions`**`(`**args*, ***kwargs*`)`

List the functions for all modules. Optionally, specify a module or modules from which to list.

CLI Example:

```
salt '*' sys.list_functions
salt '*' sys.list_functions sys
salt '*' sys.list_functions sys user
```

Function names can be specified as globs.

New in version 2015.5.0.

```
salt '*' sys.list_functions 'sys.list_*'
```

New in version ?.

```
salt '*' sys.list_functions 'module.specific_function'
```

`salt.modules.sysmod.`**`list_modules`**`(`**args*`)`

List the modules loaded on the minion

New in version 2015.5.0.

CLI Example:

```
salt '*' sys.list_modules
```

Module names can be specified as globs.

```
salt '*' sys.list_modules 's*'
```

`salt.modules.sysmod.`**`list_renderers`**`(`**args*`)`

List the renderers loaded on the minion

New in version 2015.5.0.

CLI Example:

```
salt '*' sys.list_renderers
```

Render names can be specified as globs.

```
salt '*' sys.list_renderers 'yaml*'
```

`salt.modules.sysmod.`**`list_returner_functions`**`(`**args*, ***kwargs*`)`

List the functions for all returner modules. Optionally, specify a returner module or modules from which to list.

New in version 2014.7.0.

CLI Example:

```
salt '*' sys.list_returner_functions
salt '*' sys.list_returner_functions mysql
salt '*' sys.list_returner_functions mysql etcd
```

Returner names can be specified as globs.

New in version 2015.5.0.

```
salt '*' sys.list_returner_functions 'sqlite3.get_*'
```

`salt.modules.sysmod.`**`list_returners`**`(`**args*`)`

List the returners loaded on the minion

New in version 2014.7.0.

CLI Example:

```
salt '*' sys.list_returners
```

Returner names can be specified as globs.

New in version 2015.5.0.

```
salt '*' sys.list_returners 's*'
```

`salt.modules.sysmod.`**`list_runner_functions`**`(`**args, **kwargs*`)`

List the functions for all runner modules. Optionally, specify a runner module or modules from which to list.

New in version 2014.7.0.

CLI Example:

```
salt '*' sys.list_runner_functions
salt '*' sys.list_runner_functions state
salt '*' sys.list_runner_functions state virt
```

Runner function names can be specified as globs.

New in version 2015.5.0.

```
salt '*' sys.list_runner_functions 'state.*' 'virt.*'
```

`salt.modules.sysmod.`**`list_runners`**`(`**args*`)`

List the runners loaded on the minion

New in version 2014.7.0.

CLI Example:

```
salt '*' sys.list_runners
```

Runner names can be specified as globs.

New in version 2015.5.0.

```
salt '*' sys.list_runners 'm*'
```

salt.modules.sysmod.**list_state_functions**(**args*, ***kwargs*)

List the functions for all state modules. Optionally, specify a state module or modules from which to list.

New in version 2014.7.0.

CLI Example:

```
salt '*' sys.list_state_functions
salt '*' sys.list_state_functions file
salt '*' sys.list_state_functions pkg user
```

State function names can be specified as globs.

New in version 2015.5.0.

```
salt '*' sys.list_state_functions 'file.*'
salt '*' sys.list_state_functions 'file.s*'
```

New in version ?.

```
salt '*' sys.list_state_functions 'module.specific_function'
```

salt.modules.sysmod.**list_state_modules**(**args*)

List the modules loaded on the minion

New in version 2014.7.0.

CLI Example:

```
salt '*' sys.list_state_modules
```

State module names can be specified as globs.

New in version 2015.5.0.

```
salt '*' sys.list_state_modules 'mysql_*'
```

salt.modules.sysmod.**reload_modules**()

Tell the minion to reload the execution modules

CLI Example:

```
salt '*' sys.reload_modules
```

salt.modules.sysmod.**renderer_doc**(**args*)

Return the docstrings for all renderers. Optionally, specify a renderer or a function to narrow the selection.

The strings are aggregated into a single document on the master for easy reading.

Multiple renderers can be specified.

New in version 2015.5.0.

CLI Example:

```
salt '*' sys.renderer_doc
salt '*' sys.renderer_doc cheetah
salt '*' sys.renderer_doc jinja json
```

Renderer names can be specified as globs.

```
salt '*' sys.renderer_doc 'c*' 'j*'
```

`salt.modules.sysmod.`**`returner_argspec`**`(`*module=''*`)`

Return the argument specification of functions in Salt returner modules.

New in version 2015.5.0.

CLI Example:

```
salt '*' sys.returner_argspec xmpp
salt '*' sys.returner_argspec xmpp smtp
salt '*' sys.returner_argspec
```

Returner names can be specified as globs.

```
salt '*' sys.returner_argspec 'sqlite3.*'
```

`salt.modules.sysmod.`**`returner_doc`**`(`**args*`)`

Return the docstrings for all returners. Optionally, specify a returner or a function to narrow the selection.

The strings are aggregated into a single document on the master for easy reading.

Multiple returners/functions can be specified.

New in version 2014.7.0.

CLI Example:

```
salt '*' sys.returner_doc
salt '*' sys.returner_doc sqlite3
salt '*' sys.returner_doc sqlite3.get_fun
salt '*' sys.returner_doc sqlite3.get_fun etcd.get_fun
```

Returner names can be specified as globs.

New in version 2015.5.0.

```
salt '*' sys.returner_doc 'sqlite3.get_*'
```

`salt.modules.sysmod.`**`runner_argspec`**`(`*module=''*`)`

Return the argument specification of functions in Salt runner modules.

New in version 2015.5.0.

CLI Example:

```
salt '*' sys.runner_argspec state
salt '*' sys.runner_argspec http
salt '*' sys.runner_argspec
```

Runner names can be specified as globs.

```
salt '*' sys.runner_argspec 'winrepo.*'
```

`salt.modules.sysmod.`**`runner_doc`**`(`**args*`)`

Return the docstrings for all runners. Optionally, specify a runner or a function to narrow the selection.

The strings are aggregated into a single document on the master for easy reading.

Multiple runners/functions can be specified.

New in version 2014.7.0.

CLI Example:

```
salt '*' sys.runner_doc
salt '*' sys.runner_doc cache
salt '*' sys.runner_doc cache.grains
salt '*' sys.runner_doc cache.grains mine.get
```

Runner names can be specified as globs.

New in version 2015.5.0.

```
salt '*' sys.runner_doc 'cache.clear_*'
```

salt.modules.sysmod.**state_argspec**(*module=''*)

Return the argument specification of functions in Salt state modules.

New in version 2015.5.0.

CLI Example:

```
salt '*' sys.state_argspec pkg.installed
salt '*' sys.state_argspec file
salt '*' sys.state_argspec
```

State names can be specified as globs.

```
salt '*' sys.state_argspec 'pkg.*'
```

salt.modules.sysmod.**state_doc**(**args*)

Return the docstrings for all states. Optionally, specify a state or a function to narrow the selection.

The strings are aggregated into a single document on the master for easy reading.

Multiple states/functions can be specified.

New in version 2014.7.0.

CLI Example:

```
salt '*' sys.state_doc
salt '*' sys.state_doc service
salt '*' sys.state_doc service.running
salt '*' sys.state_doc service.running ipables.append
```

State names can be specified as globs.

New in version 2015.5.0.

```
salt '*' sys.state_doc 'service.*' 'iptables.*'
```

salt.modules.sysmod.**state_schema**(*module=''*)

Return a JSON Schema for the given state function(s)

New in version 2016.3.0.

CLI Example:

```
salt '*' sys.state_schema
salt '*' sys.state_schema pkg.installed
```

21.9.420 salt.modules.sysrc

sysrc module for FreeBSD

`salt.modules.sysrc.`**`get`**`(`*`**kwargs`*`)`

Return system rc configuration variables

CLI Example:

```
salt '*' sysrc.get includeDefaults=True
```

`salt.modules.sysrc.`**`remove`**`(`*`name, **kwargs`*`)`

Remove system rc configuration variables

CLI Example:

```
salt '*' sysrc.remove name=sshd_enable
```

`salt.modules.sysrc.`**`set_`**`(`*`name, value, **kwargs`*`)`

Set system rc configuration variables

CLI Example:

```
salt '*' sysrc.set name=sshd_flags value="-p 2222"
```

21.9.421 salt.modules.system

Support for reboot, shutdown, etc on POSIX-like systems.

Note: If you have configured a wrapper such as `molly-guard` to intercept *interactive* shutdown commands, be aware that calling `system.halt`, `system.poweroff`, `system.reboot`, and `system.shutdown` with `salt-call` will hang indefinitely while the wrapper script waits for user input. Calling them with `salt` will work as expected.

`salt.modules.system.`**`get_computer_desc`**`()`

Get PRETTY_HOSTNAME value stored in /etc/machine-info If this file doesn't exist or the variable doesn't exist return False.

Returns Value of PRETTY_HOSTNAME if this does not exist False.

Return type str

CLI Example:

```
salt '*' system.get_computer_desc
```

`salt.modules.system.`**`get_computer_name`**`()`

Get hostname.

CLI Example:

```
salt '*' network.get_hostname
```

`salt.modules.system.`**`get_reboot_required_witnessed`**`()`

Determine if at any time during the current boot session the salt minion witnessed an event indicating that a reboot is required.

Returns `True` if the a reboot request was witnessed, `False` otherwise

Return type bool

CLI Example:

```
salt '*' system.get_reboot_required_witnessed
```

`salt.modules.system.`**`get_system_date`**`(`*utc_offset=None*`)`

Get the system date

Parameters **`utc_offset`** (*`str`*) – The utc offset in 4 digit (+0600) format with an optional sign (+/-). Will default to None which will use the local timezone. To set the time based off of UTC use "'+0000'". Note: if being passed through the command line will need to be quoted twice to allow negative offsets.

Returns Returns the system date.

Return type str

CLI Example:

```
salt '*' system.get_system_date
```

`salt.modules.system.`**`get_system_date_time`**`(`*utc_offset=None*`)`

Get the system date/time.

Parameters **`utc_offset`** (*`str`*) – The utc offset in 4 digit (+0600) format with an optional sign (+/-). Will default to None which will use the local timezone. To set the time based off of UTC use "'+0000'". Note: if being passed through the command line will need to be quoted twice to allow negative offsets.

Returns Returns the system time in YYYY-MM-DD hh:mm:ss format.

Return type str

CLI Example:

```
salt '*' system.get_system_date_time "'-0500'"
```

`salt.modules.system.`**`get_system_time`**`(`*utc_offset=None*`)`

Get the system time.

Parameters **`utc_offset`** (*`str`*) – The utc offset in 4 digit (+0600) format with an optional sign (+/-). Will default to None which will use the local timezone. To set the time based off of UTC use "'+0000'". Note: if being passed through the command line will need to be quoted twice to allow negative offsets.

Returns Returns the system time in HH:MM:SS AM/PM format.

Return type str

CLI Example:

```
salt '*' system.get_system_time
```

`salt.modules.system.`**`halt`**`()`

Halt a running system

CLI Example:

```
salt '*' system.halt
```

`salt.modules.system.`**`has_settable_hwclock`**`()`

Returns True if the system has a hardware clock capable of being set from software.

CLI Example:

salt '*' system.has_settable_hwclock

`salt.modules.system.`**`init`**`(`*runlevel*`)`

Change the system runlevel on sysV compatible systems

CLI Example:

```
salt '*' system.init 3
```

`salt.modules.system.`**`poweroff`**`()`

Poweroff a running system

CLI Example:

```
salt '*' system.poweroff
```

`salt.modules.system.`**`reboot`**`(`*at_time=None*`)`

Reboot the system

at_time The wait time in minutes before the system will be rebooted.

CLI Example:

```
salt '*' system.reboot
```

`salt.modules.system.`**`set_computer_desc`**`(`*desc*`)`

Set PRETTY_HOSTNAME value stored in /etc/machine-info This will create the file if it does not exist. If it is unable to create or modify this file returns False.

Parameters **desc** (*str*) – The computer description

Returns False on failure. True if successful.

CLI Example:

```
salt '*' system.set_computer_desc "Michael's laptop"
```

`salt.modules.system.`**`set_computer_name`**`(`*hostname*`)`

Modify hostname.

CLI Example:

```
salt '*' system.set_computer_name master.saltstack.com
```

`salt.modules.system.`**`set_reboot_required_witnessed`**`()`

This function is used to remember that an event indicating that a reboot is required was witnessed. This function writes to a temporary filesystem so the event gets cleared upon reboot.

Returns `True` if successful, otherwise `False`

Return type bool

```
salt '*' system.set_reboot_required_witnessed
```

`salt.modules.system.`**`set_system_date`**`(`*newdate, utc_offset=None*`)`

Set the system date. Use <mm-dd-yy> format for the date.

Parameters **newdate** (*str*) – The date to set. Can be any of the following formats:

- YYYY-MM-DD
- MM-DD-YYYY
- MM-DD-YY
- MM/DD/YYYY
- MM/DD/YY
- YYYY/MM/DD

CLI Example:

```
salt '*' system.set_system_date '03-28-13'
```

`salt.modules.system.`**`set_system_date_time`**`(`*years=None, months=None, days=None, hours=None, minutes=None, seconds=None, utc_offset=None*`)`

Set the system date and time. Each argument is an element of the date, but not required. If an element is not passed, the current system value for that element will be used. For example, if you don't pass the year, the current system year will be used. (Used by set_system_date and set_system_time)

Updates hardware clock, if present, in addition to software (kernel) clock.

Parameters

- **years** (*int*) – Years digit, ie: 2015
- **months** (*int*) – Months digit: 1 - 12
- **days** (*int*) – Days digit: 1 - 31
- **hours** (*int*) – Hours digit: 0 - 23
- **minutes** (*int*) – Minutes digit: 0 - 59
- **seconds** (*int*) – Seconds digit: 0 - 59
- **utc_offset** (*str*) – The utc offset in 4 digit (+0600) format with an optional sign (+/-). Will default to None which will use the local timezone. To set the time based off of UTC use "'+0000'". Note: if being passed through the command line will need to be quoted twice to allow negative offsets.

Returns True if successful. Otherwise False.

Return type bool

CLI Example:

```
salt '*' system.set_system_date_time 2015 5 12 11 37 53 "'-0500'"
```

salt.modules.system.**set_system_time**(*newtime*, *utc_offset=None*)

Set the system time.

Parameters

- **newtime** (*str*) – The time to set. Can be any of the following formats. - HH:MM:SS AM/PM - HH:MM AM/PM - HH:MM:SS (24 hour) - HH:MM (24 hour)

  Note that the salt command line parser parses the date/time before we obtain the argument (preventing us from doing utc) Therefore the argument must be passed in as a string. Meaning you may have to quote the text twice from the command line.
- **utc_offset** (*str*) – The utc offset in 4 digit (+0600) format with an optional sign (+/-). Will default to None which will use the local timezone. To set the time based off of UTC use "'+0000'". Note: if being passed through the command line will need to be quoted twice to allow negative offsets.

Returns Returns True if successful. Otherwise False.

Return type bool

CLI Example:

```
salt '*' system.set_system_time "'11:20'"
```

salt.modules.system.**shutdown**(*at_time=None*)

Shutdown a running system

at_time The wait time in minutes before the system will be shutdown.

CLI Example:

```
salt '*' system.shutdown 5
```

21.9.422 salt.modules.system_profiler

System Profiler Module

Interface with macOS's command-line System Profiler utility to get information about package receipts and installed applications.

New in version 2015.5.0.

`salt.modules.system_profiler.`**`applications`**`()`

Return the results of a call to `system_profiler -xml -detail full SPApplicationsDataType` as a dictionary. Top-level keys of the dictionary are the names of each set of install receipts, since there can be multiple receipts with the same name. Contents of each key are a list of dictionaries.

Note that this can take a long time depending on how many applications are installed on the target Mac.

CLI Example:

```
salt '*' systemprofiler.applications
```

`salt.modules.system_profiler.`**`receipts`**`()`

Return the results of a call to `system_profiler -xml -detail full SPInstallHistoryDataType` as a dictionary. Top-level keys of the dictionary are the names of each set of install receipts, since there can be multiple receipts with the same name. Contents of each key are a list of dictionaries.

CLI Example:

```
salt '*' systemprofiler.receipts
```

21.9.423 salt.modules.systemd_service

Provides the service module for systemd

New in version 0.10.0.

Important: If you feel that Salt should be using this module to manage services on a minion, and it is using a different module (or gives an error similar to *'service.start' is not available*), see *here*.

`salt.modules.systemd_service.`**`available`**`(`*name*`)`

New in version 0.10.4.

Check that the given service is available taking into account template units.

CLI Example:

```
salt '*' service.available sshd
```

`salt.modules.systemd_service.`**`disable`**`(`*name, no_block=False, root=None, **kwargs*`)`

Changed in version 2015.8.12,2016.3.3,2016.11.0: On minions running systemd>=205, systemd-run(1) is now used to isolate commands run by this function from the `salt-minion` daemon's control group. This is done to avoid a race condition in cases where the `salt-minion` service is restarted while a service is being modified. If desired, usage of systemd-run(1) can be suppressed by setting a *`config option`* called `systemd.scope`, with a value of `False` (no quotes).

Disable the named service to not start when the system boots

no_block [False] Set to `True` to start the service using `--no-block`.

New in version 2017.7.0.

root Enable/disable/mask unit files in the specified root directory

CLI Example:

```
salt '*' service.disable <service name>
```

`salt.modules.systemd_service.`**`disabled`**`(`*name*, *root=None*`)`

Return if the named service is disabled from starting on boot

root Enable/disable/mask unit files in the specified root directory

CLI Example:

```
salt '*' service.disabled <service name>
```

`salt.modules.systemd_service.`**`enable`**`(`*name*, *no_block=False*, *unmask=False*, *unmask_runtime=False*, *root=None*, ***kwargs*`)`

Changed in version 2015.8.12,2016.3.3,2016.11.0: On minions running systemd>=205, systemd-run(1) is now used to isolate commands run by this function from the `salt-minion` daemon's control group. This is done to avoid a race condition in cases where the `salt-minion` service is restarted while a service is being modified. If desired, usage of systemd-run(1) can be suppressed by setting a *`config option`* called `systemd.scope`, with a value of `False` (no quotes).

Enable the named service to start when the system boots

no_block [False] Set to `True` to start the service using `--no-block`.

New in version 2017.7.0.

unmask [False] Set to `True` to remove an indefinite mask before attempting to enable the service.

New in version 2017.7.0: In previous releases, Salt would simply unmask a service before enabling. This behavior is no longer the default.

unmask_runtime [False] Set to `True` to remove a runtime mask before attempting to enable the service.

New in version 2017.7.0: In previous releases, Salt would simply unmask a service before enabling. This behavior is no longer the default.

root Enable/disable/mask unit files in the specified root directory

CLI Example:

```
salt '*' service.enable <service name>
```

`salt.modules.systemd_service.`**`enabled`**`(`*name*, *root=None*, ***kwargs*`)`

Return if the named service is enabled to start on boot

root Enable/disable/mask unit files in the specified root directory

CLI Example:

```
salt '*' service.enabled <service name>
```

`salt.modules.systemd_service.`**`execs`**`(`*root=None*`)`

New in version 2014.7.0.

Return a list of all files specified as `ExecStart` for all services.

root Enable/disable/mask unit files in the specified root directory

CLI Example:

salt '*' service.execs

`salt.modules.systemd_service.`**`force_reload`**`(`*name*, *no_block=True*, *unmask=False*, *unmask_runtime=False*`)`

Changed in version 2015.8.12,2016.3.3,2016.11.0: On minions running systemd>=205, systemd-run(1) is now used to isolate commands run by this function from the `salt-minion` daemon's control group. This is done to avoid a race condition in cases where the `salt-minion` service is restarted while a service is being modified. If desired, usage of systemd-run(1) can be suppressed by setting a *`config option`* called `systemd.scope`, with a value of `False` (no quotes).

New in version 0.12.0.

Force-reload the specified service with systemd

no_block [False] Set to `True` to start the service using `--no-block`.

New in version 2017.7.0.

unmask [False] Set to `True` to remove an indefinite mask before attempting to force-reload the service.

New in version 2017.7.0: In previous releases, Salt would simply unmask a service before force-reloading. This behavior is no longer the default.

unmask_runtime [False] Set to `True` to remove a runtime mask before attempting to force-reload the service.

New in version 2017.7.0: In previous releases, Salt would simply unmask a service before force-reloading. This behavior is no longer the default.

CLI Example:

```
salt '*' service.force_reload <service name>
```

`salt.modules.systemd_service.`**`get_all`**`(`*root=None*`)`

Return a list of all available services

root Enable/disable/mask unit files in the specified root directory

CLI Example:

```
salt '*' service.get_all
```

`salt.modules.systemd_service.`**`get_disabled`**`(`*root=None*`)`

Return a list of all disabled services

root Enable/disable/mask unit files in the specified root directory

CLI Example:

```
salt '*' service.get_disabled
```

`salt.modules.systemd_service.`**`get_enabled`**`(`*root=None*`)`

Return a list of all enabled services

root Enable/disable/mask unit files in the specified root directory

CLI Example:

```
salt '*' service.get_enabled
```

`salt.modules.systemd_service.`**`get_running`**`()`

Return a list of all running services, so far as systemd is concerned

CLI Example:

```
salt '*' service.get_running
```

`salt.modules.systemd_service.`**`get_static`**`(`*root=None*`)`

New in version 2015.8.5.

Return a list of all static services

root Enable/disable/mask unit files in the specified root directory

CLI Example:

```
salt '*' service.get_static
```

`salt.modules.systemd_service.`**`mask`**`(`*name*, *runtime=False*, *root=None*`)`

New in version 2015.5.0.

Changed in version 2015.8.12,2016.3.3,2016.11.0: On minions running systemd>=205, systemd-run(1) is now used to isolate commands run by this function from the `salt-minion` daemon's control group. This is done to avoid a race condition in cases where the `salt-minion` service is restarted while a service is being modified. If desired, usage of systemd-run(1) can be suppressed by setting a *config option* called `systemd.scope`, with a value of `False` (no quotes).

Mask the specified service with systemd

runtime [False] Set to `True` to mask this service only until the next reboot

New in version 2015.8.5.

root Enable/disable/mask unit files in the specified root directory

CLI Example:

```
salt '*' service.mask foo
salt '*' service.mask foo runtime=True
```

`salt.modules.systemd_service.`**`masked`**`(`*name, runtime=False, root=None*`)`

New in version 2015.8.0.

Changed in version 2015.8.5: The return data for this function has changed. If the service is masked, the return value will now be the output of the `systemctl is-enabled` command (so that a persistent mask can be distinguished from a runtime mask). If the service is not masked, then `False` will be returned.

Changed in version 2017.7.0: This function now returns a boolean telling the user whether a mask specified by the new `runtime` argument is set. If `runtime` is `False`, this function will return `True` if an indefinite mask is set for the named service (otherwise `False` will be returned). If `runtime` is `False`, this function will return `True` if a runtime mask is set, otherwise `False`.

Check whether or not a service is masked

runtime [False] Set to `True` to check for a runtime mask

New in version 2017.7.0: In previous versions, this function would simply return the output of `systemctl is-enabled` when the service was found to be masked. However, since it is possible to both have both indefinite and runtime masks on a service simultaneously, this function now only checks for runtime masks if this argument is set to `True`. Otherwise, it will check for an indefinite mask.

root Enable/disable/mask unit files in the specified root directory

CLI Examples:

```
salt '*' service.masked foo
salt '*' service.masked foo runtime=True
```

`salt.modules.systemd_service.`**`missing`**`(`*name*`)`

New in version 2014.1.0.

The inverse of `service.available`. Returns `True` if the specified service is not available, otherwise returns `False`.

CLI Example:

```
salt '*' service.missing sshd
```

`salt.modules.systemd_service.`**`reload_`**`(`*name, no_block=False, unmask=False, unmask_runtime=False*`)`

Changed in version 2015.8.12,2016.3.3,2016.11.0: On minions running systemd>=205, systemd-run(1) is now used to isolate commands run by this function from the `salt-minion` daemon's control group. This is done to avoid a race condition in cases where the `salt-minion` service is restarted while a service is being modified. If desired, usage of systemd-run(1) can be suppressed by setting a *config option* called `systemd.scope`, with a value of `False` (no quotes).

Reload the specified service with systemd

no_block [False] Set to `True` to reload the service using `--no-block`.

New in version 2017.7.0.

unmask [False] Set to `True` to remove an indefinite mask before attempting to reload the service.

New in version 2017.7.0: In previous releases, Salt would simply unmask a service before reloading. This behavior is no longer the default.

unmask_runtime [False] Set to `True` to remove a runtime mask before attempting to reload the service.

New in version 2017.7.0: In previous releases, Salt would simply unmask a service before reloading. This behavior is no longer the default.

CLI Example:

```
salt '*' service.reload <service name>
```

`salt.modules.systemd_service.`**`restart`**(*name*, *no_block=False*, *unmask=False*, *unmask_runtime=False*)

Changed in version 2015.8.12,2016.3.3,2016.11.0: On minions running systemd>=205, systemd-run(1) is now used to isolate commands run by this function from the `salt-minion` daemon's control group. This is done to avoid a race condition in cases where the `salt-minion` service is restarted while a service is being modified. If desired, usage of systemd-run(1) can be suppressed by setting a *`config option`* called `systemd.scope`, with a value of `False` (no quotes).

Restart the specified service with systemd

no_block [False] Set to `True` to start the service using `--no-block`.

New in version 2017.7.0.

unmask [False] Set to `True` to remove an indefinite mask before attempting to restart the service.

New in version 2017.7.0: In previous releases, Salt would simply unmask a service before restarting. This behavior is no longer the default.

unmask_runtime [False] Set to `True` to remove a runtime mask before attempting to restart the service.

New in version 2017.7.0: In previous releases, Salt would simply unmask a service before restarting. This behavior is no longer the default.

CLI Example:

```
salt '*' service.restart <service name>
```

`salt.modules.systemd_service.`**`show`**(*name*, *root=None*)

New in version 2014.7.0.

Show properties of one or more units/jobs or the manager

root Enable/disable/mask unit files in the specified root directory

CLI Example:

salt '*' service.show <service name>

`salt.modules.systemd_service.`**`start`**(*name*, *no_block=False*, *unmask=False*, *unmask_runtime=False*)

Changed in version 2015.8.12,2016.3.3,2016.11.0: On minions running systemd>=205, systemd-run(1) is now used to isolate commands run by this function from the `salt-minion` daemon's control group. This is done to avoid a race condition in cases where the `salt-minion` service is restarted while a service is being modified. If desired, usage of systemd-run(1) can be suppressed by setting a *`config option`* called `systemd.scope`, with a value of `False` (no quotes).

Start the specified service with systemd

no_block [False] Set to `True` to start the service using `--no-block`.

New in version 2017.7.0.

unmask [False] Set to `True` to remove an indefinite mask before attempting to start the service.

New in version 2017.7.0: In previous releases, Salt would simply unmask a service before starting. This behavior is no longer the default.

unmask_runtime [False] Set to `True` to remove a runtime mask before attempting to start the service.

New in version 2017.7.0: In previous releases, Salt would simply unmask a service before starting. This behavior is no longer the default.

CLI Example:

```
salt '*' service.start <service name>
```

`salt.modules.systemd_service.`**`status`**(*name*, *sig=None*, *wait=3*)

Check whether or not a service is active. If the name contains globbing, a dict mapping service names to True/False values is returned.

Changed in version 2018.3.0: The service name can now be a glob (e.g. `salt*`)

name The name of the service to check

sig Not implemented, but required to be accepted as it is passed by service states

wait [3] If the service is in the process of changing states (i.e. it is in either the `activating` or `deactivating` state), wait up to this amount of seconds (checking again periodically) before determining whether the service is active.

New in version 2019.2.3.

CLI Example:

```
salt '*' service.status <service name> [service signature]
```

`salt.modules.systemd_service.`**`stop`**(*name*, *no_block=False*)

Changed in version 2015.8.12,2016.3.3,2016.11.0: On minions running systemd>=205, systemd-run(1) is now used to isolate commands run by this function from the `salt-minion` daemon's control group. This is done to avoid a race condition in cases where the `salt-minion` service is restarted while a service is being modified. If desired, usage of systemd-run(1) can be suppressed by setting a *`config option`* called `systemd.scope`, with a value of `False` (no quotes).

Stop the specified service with systemd

no_block [False] Set to `True` to start the service using `--no-block`.

New in version 2017.7.0.

CLI Example:

```
salt '*' service.stop <service name>
```

`salt.modules.systemd_service.`**`systemctl_reload`**()

New in version 0.15.0.

Reloads systemctl, an action needed whenever unit files are updated.

CLI Example:

```
salt '*' service.systemctl_reload
```

`salt.modules.systemd_service.`**`unmask_`**(*name*, *runtime=False*, *root=None*)

New in version 2015.5.0.

Changed in version 2015.8.12,2016.3.3,2016.11.0: On minions running systemd>=205, systemd-run(1) is now used to isolate commands run by this function from the `salt-minion` daemon's control group. This is done to avoid a race condition in cases where the `salt-minion` service is restarted while a service is being modified. If desired, usage of systemd-run(1) can be suppressed by setting a *`config option`* called `systemd.scope`, with a value of `False` (no quotes).

Unmask the specified service with systemd

runtime [False] Set to `True` to unmask this service only until the next reboot

New in version 2017.7.0: In previous versions, this function would remove whichever mask was identified by running `systemctl is-enabled` on the service. However, since it is possible to both have both indefinite and runtime masks on a service simultaneously, this function now removes a runtime mask only when this argument is set to `True`, and otherwise removes an indefinite mask.

root Enable/disable/mask unit files in the specified root directory

CLI Example:

```
salt '*' service.unmask foo
salt '*' service.unmask foo runtime=True
```

21.9.424 salt.modules.telegram

Module for sending messages via Telegram.

configuration In order to send a message via the Telegram, certain configuration is required in /etc/salt/minion on the relevant minions or in the pillar. Some sample configs might look like:

```
telegram.chat_id: '123456789'
telegram.token: '00000000:xxxxxxxxxxxxxxxxxxxxxxxx'
```

`salt.modules.telegram.`**`post_message`**(*message*, *chat_id=None*, *token=None*)

Send a message to a Telegram chat.

Parameters

- **`message`** – The message to send to the Telegram chat.
- **`chat_id`** – (optional) The Telegram chat id.
- **`token`** – (optional) The Telegram API token.

Returns Boolean if message was sent successfully.

CLI Example:

```
salt '*' telegram.post_message message="Hello Telegram!"
```

21.9.425 salt.modules.telemetry

Connection module for Telemetry

New in version 2016.3.0.

https://github.com/mongolab/mongolab-telemetry-api-docs/blob/master/alerts.md

configuration This module accepts explicit telemetry credentials or can also read api key credentials from a pillar. More Information available here.

In the minion's config file:

```
telemetry.telemetry_api_keys:
  - abc123  # Key 1
  - efg321  # Backup Key 1
telemetry_api_base_url: https://telemetry-api.mongolab.com/v0
```

depends requests

`salt.modules.telemetry.`**`create_alarm`**(*deployment_id*, *metric_name*, *data*, *api_key=None*, *profile='telemetry'*)

create an telemetry alarms.

data is a dict of alert configuration data.

Returns (bool success, str message) tuple.

CLI Example:

salt myminion telemetry.create_alarm rs-ds033197 {} profile=telemetry

`salt.modules.telemetry.`**`delete_alarms`**(*deployment_id*, *alert_id=None*, *metric_name=None*, *api_key=None*, *profile='telemetry'*)

delete an alert specified by alert_id or if not specified blows away all the alerts in the current deployment.

Returns (bool success, str message) tuple.

CLI Example:

salt myminion telemetry.delete_alarms rs-ds033197 profile=telemetry

`salt.modules.telemetry.`**`get_alarms`**(*deployment_id*, *profile='telemetry'*)

get all the alarms set up against the current deployment

Returns dictionary of alarm information

CLI Example:

salt myminion telemetry.get_alarms rs-ds033197 profile=telemetry

`salt.modules.telemetry.`**`get_alert_config`**(*deployment_id*, *metric_name=None*, *api_key=None*, *profile='telemetry'*)

Get all alert definitions associated with a given deployment or if metric_name is specified, obtain the specific alert config

Returns dictionary or list of dictionaries.

CLI Example:

salt myminion telemetry.get_alert_config rs-ds033197 currentConnections profile=telemetry salt myminion telemetry.get_alert_config rs-ds033197 profile=telemetry

`salt.modules.telemetry.`**`get_notification_channel_id`**(*notify_channel*, *profile='telemetry'*)

Given an email address, creates a notification-channels if one is not found and also returns the corresponding notification channel id.

notify_channel Email escalation policy

profile A dict of telemetry config information.

CLI Example:

salt myminion telemetry.get_notification_channel_id userx@company.com profile=telemetry

`salt.modules.telemetry.`**`update_alarm`**(*deployment_id*, *metric_name*, *data*, *api_key=None*, *profile='telemetry'*)

update an telemetry alarms. data is a dict of alert configuration data.

Returns (bool success, str message) tuple.

CLI Example:

salt myminion telemetry.update_alarm rs-ds033197 {} profile=telemetry

21.9.426 salt.modules.temp

Simple module for creating temporary directories and files

This is a thin wrapper around Pythons tempfile module

New in version 2015.8.0.

`salt.modules.temp.`**`dir`**(*suffix=''*, *prefix='tmp'*, *parent=None*)

Create a temporary directory

CLI Example:

```
salt '*' temp.dir
salt '*' temp.dir prefix='mytemp-' parent='/var/run/'
```

`salt.modules.temp.`**`file`**(*suffix='', prefix='tmp', parent=None*)

Create a temporary file

CLI Example:

```
salt '*' temp.file
salt '*' temp.file prefix='mytemp-' parent='/var/run/'
```

21.9.427 salt.modules.test

Module for running arbitrary tests

`salt.modules.test.`**`arg`**(**args, **kwargs*)

Print out the data passed into the function `*args` and ``kwargs``, this is used to both test the publication data and cli argument passing, but also to display the information available within the publication data. Returns {"args": args, "kwargs": kwargs}.

CLI Example:

```
salt '*' test.arg 1 "two" 3.1 txt="hello" wow='{a: 1, b: "hello"}'
```

`salt.modules.test.`**`arg_clean`**(**args, **kwargs*)

Like test.arg but cleans kwargs of the __pub* items CLI Example:

```
salt '*' test.arg_clean 1 "two" 3.1 txt="hello" wow='{a: 1, b: "hello"}'
```

`salt.modules.test.`**`arg_repr`**(**args, **kwargs*)

Print out the data passed into the function `*args` and ``kwargs``, this is used to both test the publication data and cli argument passing, but also to display the information available within the publication data. Returns {"args": repr(args), "kwargs": repr(kwargs)}.

CLI Example:

```
salt '*' test.arg_repr 1 "two" 3.1 txt="hello" wow='{a: 1, b: "hello"}'
```

`salt.modules.test.`**`arg_type`**(**args, **kwargs*)

Print out the types of the args and kwargs. This is used to test the types of the args and kwargs passed down to the minion

CLI Example:

```
salt '*' test.arg_type 1 'int'
```

`salt.modules.test.`**`assertion`**(*assertion*)

Assert the given argument

CLI Example:

```
salt '*' test.assertion False
```

`salt.modules.test.`**`attr_call`**()

Call grains.items via the attribute

CLI Example:

```
salt '*' test.attr_call
```

salt.modules.test.collatz(*start*)

Execute the collatz conjecture from the passed starting number, returns the sequence and the time it took to compute. Used for performance tests.

CLI Example:

```
salt '*' test.collatz 3
```

salt.modules.test.conf_test()

Return the value for test.foo in the minion configuration file, or return the default value

CLI Example:

```
salt '*' test.conf_test
```

salt.modules.test.cross_test(*func, args=None*)

Execute a minion function via the __salt__ object in the test module, used to verify that the minion functions can be called via the __salt__ module.

CLI Example:

```
salt '*' test.cross_test file.gid_to_group 0
```

salt.modules.test.echo(*text*)

Return a string - used for testing the connection

CLI Example:

```
salt '*' test.echo 'foo bar baz quo qux'
```

salt.modules.test.exception(*message='Test Exception'*)

Raise an exception

Optionally provide an error message or output the full stack.

CLI Example:

```
salt '*' test.exception 'Oh noes!'
```

salt.modules.test.false_()

Always return False

CLI Example:

```
salt '*' test.false
```

salt.modules.test.fib(*num*)

Return the num-th Fibonacci number, and the time it took to compute in seconds. Used for performance tests.

This function is designed to have terrible performance.

CLI Example:

```
salt '*' test.fib 3
```

salt.modules.test.get_opts()

Return the configuration options passed to this minion

CLI Example:

```
salt '*' test.get_opts
```

`salt.modules.test.`**`kwarg`**`(`*`**kwargs`*`)`

Print out the data passed into the function `**kwargs`, this is used to both test the publication data and cli kwarg passing, but also to display the information available within the publication data.

CLI Example:

```
salt '*' test.kwarg num=1 txt="two" env='{a: 1, b: "hello"}'
```

`salt.modules.test.`**`missing_func`**`()`

`salt.modules.test.`**`module_report`**`()`

Return a dict containing all of the execution modules with a report on the overall availability via different references

CLI Example:

```
salt '*' test.module_report
```

`salt.modules.test.`**`not_loaded`**`()`

List the modules that were not loaded by the salt loader system

CLI Example:

```
salt '*' test.not_loaded
```

`salt.modules.test.`**`opts_pkg`**`()`

Return an opts package with the grains and opts for this minion. This is primarily used to create the options used for master side state compiling routines

CLI Example:

```
salt '*' test.opts_pkg
```

`salt.modules.test.`**`outputter`**`(`*`data`*`)`

Test the outputter, pass in data to return

CLI Example:

```
salt '*' test.outputter foobar
```

`salt.modules.test.`**`ping`**`()`

Used to make sure the minion is up and responding. Not an ICMP ping.

Returns `True`.

CLI Example:

```
salt '*' test.ping
```

`salt.modules.test.`**`provider`**`(`*`module`*`)`

Pass in a function name to discover what provider is being used

CLI Example:

```
salt '*' test.provider service
```

`salt.modules.test.`**`providers`**`()`

Return a dict of the provider names and the files that provided them

CLI Example:

```
salt '*' test.providers
```

`salt.modules.test.`**`raise_exception`**`(`*name*, **args*, ***kwargs*`)`

Raise an exception. Built-in exceptions and those in `salt.exceptions` can be raised by this test function. If no matching exception is found, then no exception will be raised and this function will return `False`.

This function is designed to test Salt's exception and return code handling.

CLI Example:

```
salt '*' test.raise_exception TypeError "An integer is required"
salt '*' test.raise_exception salt.exceptions.CommandExecutionError "Something
↪went wrong"
```

`salt.modules.test.`**`rand_sleep`**`(`*max=60*`)`

Sleep for a random number of seconds, used to test long-running commands and minions returning at differing intervals

CLI Example:

```
salt '*' test.rand_sleep 60
```

`salt.modules.test.`**`random_hash`**`(`*size=9999999999*, *hash_type=None*`)`

New in version 2015.5.2.

Changed in version 2018.3.0: Function has been renamed from `test.rand_str` to `test.random_hash`

Generates a random number between 1 and `size`, then returns a hash of that number. If no `hash_type` is passed, the hash_type specified by the minion's *`hash_type`* config option is used.

CLI Example:

```
salt '*' test.random_hash
salt '*' test.random_hash hash_type=sha512
```

`salt.modules.test.`**`retcode`**`(`*code=42*`)`

Test that the returncode system is functioning correctly

CLI Example:

```
salt '*' test.retcode 42
```

`salt.modules.test.`**`sleep`**`(`*length*`)`

Instruct the minion to initiate a process that will sleep for a given period of time.

CLI Example:

```
salt '*' test.sleep 20
```

`salt.modules.test.`**`stack`**`()`

Return the current stack trace

CLI Example:

```
salt '*' test.stack
```

`salt.modules.test.`**`true_`**`()`

Always return True

CLI Example:

```
salt '*' test.true
```

`salt.modules.test.`**`try_`**`(`*module*, *return_try_exception=False*, ***kwargs*`)`

Try to run a module command. On an exception return None. If *return_try_exception* is set True return the exception. This can be helpful in templates where running a module might fail as expected.

CLI Example:

```
<pre>
{% for i in range(0,230) %}
    {{ salt['test.try'](module='ipmi.get_users', bmc_host='172.2.2.
↪'+i)|yaml(False) }}
{% endfor %}
</pre>
```

`salt.modules.test.`**`tty`**`(`**args*, ***kwargs*`)`

Deprecated! Moved to cmdmod.

CLI Example:

```
salt '*' test.tty tty0 'This is a test'
salt '*' test.tty pts3 'This is a test'
```

`salt.modules.test.`**`version`**`()`

Return the version of salt on the minion

CLI Example:

```
salt '*' test.version
```

`salt.modules.test.`**`versions`**`()`

This function is an alias of `versions_report`.

> Returns versions of components used by salt
>
> CLI Example:
>
> ```
> salt '*' test.versions_report
> ```

`salt.modules.test.`**`versions_information`**`()`

Report the versions of dependent and system software

CLI Example:

```
salt '*' test.versions_information
```

`salt.modules.test.`**`versions_report`**`()`

Returns versions of components used by salt

CLI Example:

```
salt '*' test.versions_report
```

21.9.428 salt.modules.testinframod module

This module exposes the functionality of the TestInfra library for use with SaltStack in order to verify the state of your minions. In order to allow for the addition of new resource types in TestInfra this module dynamically generates wrappers for the various resources by iterating over the values in the `__all__` variable exposed by the testinfra.modules namespace.

21.9.429 salt.modules.test_virtual

Module for running arbitrary tests with a __virtual__ function

`salt.modules.test_virtual.`**`ping`**`()`

21.9.430 salt.modules.textfsm_mod module

TextFSM

New in version 2018.3.0.

Execution module that processes plain text and extracts data using TextFSM templates. The output is presented in JSON serializable data, and can be easily re-used in other modules, or directly inside the renderer (Jinja, Mako, Genshi, etc.).

depends

- textfsm Python library

Note: For Python 2/3 compatibility, it is more recommended to install the `jtextfsm` library: `pip install jtextfsm`.

`salt.modules.textfsm_mod.`**`extract`**`(`*template_path*, *raw_text=None*, *raw_text_file=None*, *saltenv='base'*`)`

Extracts the data entities from the unstructured raw text sent as input and returns the data mapping, processing using the TextFSM template.

template_path The path to the TextFSM template. This can be specified using the absolute path to the file, or using one of the following URL schemes:

- `salt://`, to fetch the template from the Salt fileserver.
- `http://` or `https://`
- `ftp://`
- `s3://`
- `swift://`

raw_text: `None` The unstructured text to be parsed.

raw_text_file: `None` Text file to read, having the raw text to be parsed using the TextFSM template. Supports the same URL schemes as the `template_path` argument.

saltenv: `base` Salt fileserver envrionment from which to retrieve the file. Ignored if `template_path` is not a `salt://` URL.

CLI Example:

```
salt '*' textfsm.extract salt://textfsm/juniper_version_template raw_text_
↪file=s3://junos_ver.txt
salt '*' textfsm.extract http://some-server/textfsm/juniper_version_template raw_
↪text='Hostname: router.abc ... snip ...'
```

Jinja template example:

```
{%- set raw_text = 'Hostname: router.abc ... snip ...' -%}
{%- set textfsm_extract = salt.textfsm.extract('https://some-server/textfsm/
↪juniper_version_template', raw_text) -%}
```

Raw text example:

```
Hostname: router.abc
Model: mx960
JUNOS Base OS boot [9.1S3.5]
JUNOS Base OS Software Suite [9.1S3.5]
JUNOS Kernel Software Suite [9.1S3.5]
JUNOS Crypto Software Suite [9.1S3.5]
JUNOS Packet Forwarding Engine Support (M/T Common) [9.1S3.5]
JUNOS Packet Forwarding Engine Support (MX Common) [9.1S3.5]
JUNOS Online Documentation [9.1S3.5]
JUNOS Routing Software Suite [9.1S3.5]
```

TextFSM Example:

```
Value Chassis (\S+)
Value Required Model (\S+)
Value Boot (.*)
Value Base (.*)
Value Kernel (.*)
Value Crypto (.*)
Value Documentation (.*)
Value Routing (.*)

Start
# Support multiple chassis systems.
  ^\S+:$$ -> Continue.Record
  ^${Chassis}:$$
  ^Model: ${Model}
  ^JUNOS Base OS boot \[${Boot}\]
  ^JUNOS Software Release \[${Base}\]
  ^JUNOS Base OS Software Suite \[${Base}\]
  ^JUNOS Kernel Software Suite \[${Kernel}\]
  ^JUNOS Crypto Software Suite \[${Crypto}\]
  ^JUNOS Online Documentation \[${Documentation}\]
  ^JUNOS Routing Software Suite \[${Routing}\]
```

Output example:

```
{
    "comment": "",
    "result": true,
    "out": [
        {
            "kernel": "9.1S3.5",
            "documentation": "9.1S3.5",
            "boot": "9.1S3.5",
            "crypto": "9.1S3.5",
            "chassis": "",
            "routing": "9.1S3.5",
            "base": "9.1S3.5",
            "model": "mx960"
        }
    ]
}
```

`salt.modules.textfsm_mod.`**`index`**(*command*, *platform=None*, *platform_grain_name=None*, *platform_column_name=None*, *output=None*, *output_file=None*, *textfsm_path=None*, *index_file=None*, *saltenv='base'*, *include_empty=False*, *include_pat=None*, *exclude_pat=None*)

Dynamically identify the template required to extract the information from the unstructured raw text.

The output has the same structure as the `extract` execution function, the difference being that `index` is capable to identify what template to use, based on the platform details and the `command`.

command The command executed on the device, to get the output.

platform The platform name, as defined in the TextFSM index file.

> **Note:** For ease of use, it is recommended to define the TextFSM indexfile with values that can be matches using the grains.

platform_grain_name The name of the grain used to identify the platform name in the TextFSM index file.

> **Note:** This option can be also specified in the minion configuration file or pillar as `textfsm_platform_grain`.

> **Note:** This option is ignored when `platform` is specified.

platform_column_name: **`Platform`** The column name used to identify the platform, exactly as specified in the TextFSM index file. Default: `Platform`.

> **Note:** This is field is case sensitive, make sure to assign the correct value to this option, exactly as defined in the index file.

> **Note:** This option can be also specified in the minion configuration file or pillar as `textfsm_platform_column_name`.

output The raw output from the device, to be parsed and extract the structured data.

output_file The path to a file that contains the raw output from the device, used to extract the structured data. This option supports the usual Salt-specific schemes: `file://`, `salt://`, `http://`, `https://`, `ftp://`, `s3://`, `swift://`.

textfsm_path The path where the TextFSM templates can be found. This can be either absolute path on the server, either specified using the following URL schemes: `file://`, `salt://`, `http://`, `https://`, `ftp://`, `s3://`, `swift://`.

> **Note:** This needs to be a directory with a flat structure, having an index file (whose name can be specified using the `index_file` option) and a number of TextFSM templates.

> **Note:** This option can be also specified in the minion configuration file or pillar as `textfsm_path`.

index_file: **`index`** The name of the TextFSM index file, under the `textfsm_path`. Default: `index`.

> **Note:** This option can be also specified in the minion configuration file or pillar as `textfsm_index_file`.

saltenv: **base** Salt fileserver envrionment from which to retrieve the file. Ignored if `textfsm_path` is not a `salt://` URL.

include_empty: **False** Include empty files under the `textfsm_path`.

include_pat Glob or regex to narrow down the files cached from the given path. If matching with a regex, the regex must be prefixed with `E@`, otherwise the expression will be interpreted as a glob.

exclude_pat Glob or regex to exclude certain files from being cached from the given path. If matching with a regex, the regex must be prefixed with `E@`, otherwise the expression will be interpreted as a glob.

> **Note:** If used with `include_pat`, files matching this pattern will be excluded from the subset of files defined by `include_pat`.

CLI Example:

```
salt '*' textfsm.index 'sh ver' platform=Juniper output_file=salt://textfsm/
↪juniper_version_example textfsm_path=salt://textfsm/
salt '*' textfsm.index 'sh ver' output_file=salt://textfsm/juniper_version_
↪example textfsm_path=ftp://textfsm/ platform_column_name=Vendor
salt '*' textfsm.index 'sh ver' output_file=salt://textfsm/juniper_version_
↪example textfsm_path=https://some-server/textfsm/ platform_column_name=Vendor␣
↪platform_grain_name=vendor
```

TextFSM index file example:

`salt://textfsm/index`

```
Template, Hostname, Vendor, Command
juniper_version_template, .*, Juniper, sh[[ow]] ve[[rsion]]
```

The usage can be simplified, by defining (some of) the following options: `textfsm_platform_grain`, `textfsm_path`, `textfsm_platform_column_name`, or `textfsm_index_file`, in the (proxy) minion configuration file or pillar.

Configuration example:

```
textfsm_platform_grain: vendor
textfsm_path: salt://textfsm/
textfsm_platform_column_name: Vendor
```

And the CLI usage becomes as simple as:

```
salt '*' textfsm.index 'sh ver' output_file=salt://textfsm/juniper_version_example
```

Usgae inside a Jinja template:

```
{%- set command = 'sh ver' -%}
{%- set output = salt.net.cli(command) -%}
{%- set textfsm_extract = salt.textfsm.index(command, output=output) -%}
```

21.9.431 salt.modules.timezone

Module for managing timezone on POSIX-like systems.

`salt.modules.timezone.`**`get_hwclock`**`()`

Get current hardware clock setting (UTC or localtime)

CLI Example:

```
salt '*' timezone.get_hwclock
```

salt.modules.timezone.**get_offset**()

Get current numeric timezone offset from UCT (i.e. -0700)

CLI Example:

```
salt '*' timezone.get_offset
```

salt.modules.timezone.**get_zone**()

Get current timezone (i.e. America/Denver)

Changed in version 2016.11.4.

Note: On AIX operating systems, Posix values can also be returned 'CST6CDT,M3.2.0/2:00:00,M11.1.0/2:00:00'

CLI Example:

```
salt '*' timezone.get_zone
```

salt.modules.timezone.**get_zonecode**()

Get current timezone (i.e. PST, MDT, etc)

CLI Example:

```
salt '*' timezone.get_zonecode
```

salt.modules.timezone.**set_hwclock**(*clock*)

Sets the hardware clock to be either UTC or localtime

CLI Example:

```
salt '*' timezone.set_hwclock UTC
```

salt.modules.timezone.**set_zone**(*timezone*)

Unlinks, then symlinks /etc/localtime to the set timezone.

The timezone is crucial to several system processes, each of which SHOULD be restarted (for instance, whatever you system uses as its cron and syslog daemons). This will not be automagically done and must be done manually!

CLI Example:

```
salt '*' timezone.set_zone 'America/Denver'
```

Changed in version 2016.11.4.

Note: On AIX operating systems, Posix values are also allowed, see below

---

```
salt '*' timezone.set_zone 'CST6CDT,M3.2.0/2:00:00,M11.1.0/2:00:00'
```

salt.modules.timezone.**zone_compare**(*timezone*)

Compares the given timezone name with the system timezone name. Checks the hash sum between the given timezone, and the one set in /etc/localtime. Returns True if names and hash sums match, and False if not. Mostly useful for running state checks.

Changed in version 2016.3.0.

Note: On Solaris-link operating systems only a string comparison is done.

Changed in version 2016.11.4.

Note: On AIX operating systems only a string comparison is done.

CLI Example:

```
salt '*' timezone.zone_compare 'America/Denver'
```

21.9.432 salt.modules.tls

A salt module for SSL/TLS. Can create a Certificate Authority (CA) or use Self-Signed certificates.

depends PyOpenSSL Python module (0.10 or later, 0.14 or later for X509 extension support)

configuration Add the following values in /etc/salt/minion for the CA module to function properly:

```
ca.cert_base_path: '/etc/pki'
```

CLI Example #1: Creating a CA, a server request and its signed certificate:

```
# salt-call tls.create_ca my_little \
days=5 \
CN='My Little CA' \
C=US \
ST=Utah \
L=Salt Lake City \
O=Saltstack \
emailAddress=pleasedontemail@example.com

Created Private Key: "/etc/pki/my_little/my_little_ca_cert.key"
Created CA "my_little_ca": "/etc/pki/my_little_ca/my_little_ca_cert.crt"

# salt-call tls.create_csr my_little CN=www.example.com
Created Private Key: "/etc/pki/my_little/certs/www.example.com.key
Created CSR for "www.example.com": "/etc/pki/my_little/certs/www.example.com.csr"

# salt-call tls.create_ca_signed_cert my_little CN=www.example.com
Created Certificate for "www.example.com": /etc/pki/my_little/certs/www.example.com.
↪crt"
```

CLI Example #2: Creating a client request and its signed certificate

```
# salt-call tls.create_csr my_little CN=DBReplica_No.1 cert_type=client
Created Private Key: "/etc/pki/my_little/certs//DBReplica_No.1.key."
Created CSR for "DBReplica_No.1": "/etc/pki/my_little/certs/DBReplica_No.1.csr."

# salt-call tls.create_ca_signed_cert my_little CN=DBReplica_No.1
Created Certificate for "DBReplica_No.1": "/etc/pki/my_little/certs/DBReplica_No.1.crt
↪"
```

CLI Example #3: Creating both a server and client req + cert for the same CN

```
# salt-call tls.create_csr my_little CN=MasterDBReplica_No.2  \
    cert_type=client
Created Private Key: "/etc/pki/my_little/certs/MasterDBReplica_No.2.key."
Created CSR for "DBReplica_No.1": "/etc/pki/my_little/certs/MasterDBReplica_No.2.csr."

# salt-call tls.create_ca_signed_cert my_little CN=MasterDBReplica_No.2
Created Certificate for "DBReplica_No.1": "/etc/pki/my_little/certs/DBReplica_No.1.crt
↪"

# salt-call tls.create_csr my_little CN=MasterDBReplica_No.2 \
    cert_type=server
Certificate "MasterDBReplica_No.2" already exists

(doh!)

# salt-call tls.create_csr my_little CN=MasterDBReplica_No.2 \
    cert_type=server type_ext=True
Created Private Key: "/etc/pki/my_little/certs/DBReplica_No.1_client.key."
Created CSR for "DBReplica_No.1": "/etc/pki/my_little/certs/DBReplica_No.1_client.csr.
↪"

# salt-call tls.create_ca_signed_cert my_little CN=MasterDBReplica_No.2
Certificate "MasterDBReplica_No.2" already exists

(DOH!)

# salt-call tls.create_ca_signed_cert my_little CN=MasterDBReplica_No.2 \
    cert_type=server type_ext=True
Created Certificate for "MasterDBReplica_No.2": "/etc/pki/my_little/certs/
↪MasterDBReplica_No.2_server.crt"
```

CLI Example #4: Create a server req + cert with non-CN filename for the cert

```
# salt-call tls.create_csr my_little CN=www.anothersometh.ing \
    cert_type=server type_ext=True
Created Private Key: "/etc/pki/my_little/certs/www.anothersometh.ing_server.key."
Created CSR for "DBReplica_No.1": "/etc/pki/my_little/certs/www.anothersometh.ing_
↪server.csr."

# salt-call tls_create_ca_signed_cert my_little CN=www.anothersometh.ing \
    cert_type=server cert_filename="something_completely_different"
Created Certificate for "www.anothersometh.ing": /etc/pki/my_little/certs/something_
↪completely_different.crt
```

salt.modules.tls.**ca_exists**(*ca_name*, *cacert_path=None*, *ca_filename=None*)

Verify whether a Certificate Authority (CA) already exists

ca_name name of the CA

cacert_path absolute path to ca certificates root directory

ca_filename alternative filename for the CA

New in version 2015.5.3.

CLI Example:

```
salt '*' tls.ca_exists test_ca /etc/certs
```

salt.modules.tls.**cert_base_path**(*cacert_path=None*)

Return the base path for certs from CLI or from options

cacert_path absolute path to ca certificates root directory

CLI Example:

```
salt '*' tls.cert_base_path
```

`salt.modules.tls.`**`cert_info`**(*cert, digest='sha256'*)

Return information for a particular certificate

cert path to the certifiate PEM file or string

Changed in version 2018.3.4.

digest what digest to use for fingerprinting

CLI Example:

```
salt '*' tls.cert_info /dir/for/certs/cert.pem
```

`salt.modules.tls.`**`create_ca`**(*ca_name, bits=2048, days=365, CN='localhost', C='US', ST='Utah', L='Salt Lake City', O='SaltStack', OU=None, emailAddress=None, fixmode=False, cacert_path=None, ca_filename=None, digest='sha256', onlyif=None, unless=None, replace=False*)

Create a Certificate Authority (CA)

ca_name name of the CA

bits number of RSA key bits, default is 2048

days number of days the CA will be valid, default is 365

CN common name in the request, default is "localhost"

C country, default is "US"

ST state, default is "Utah"

L locality, default is "Centerville", the city where SaltStack originated

O organization, default is "SaltStack"

OU organizational unit, default is None

emailAddress email address for the CA owner, default is None

cacert_path absolute path to ca certificates root directory

ca_filename alternative filename for the CA

New in version 2015.5.3.

digest The message digest algorithm. Must be a string describing a digest algorithm supported by OpenSSL (by EVP_get_digestbyname, specifically). For example, "md5" or "sha1". Default: 'sha256'

replace Replace this certificate even if it exists

New in version 2015.5.1.

Writes out a CA certificate based upon defined config values. If the file already exists, the function just returns assuming the CA certificate already exists.

If the following values were set:

```
ca.cert_base_path='/etc/pki'
ca_name='koji'
```

the resulting CA, and corresponding key, would be written in the following location:

```
/etc/pki/koji/koji_ca_cert.crt
/etc/pki/koji/koji_ca_cert.key
```

CLI Example:

```
salt '*' tls.create_ca test_ca
```

`salt.modules.tls.`**`create_ca_signed_cert`**`(`*ca_name, CN, days=365, cacert_path=None, ca_filename=None, cert_path=None, cert_filename=None, digest='sha256', cert_type=None, type_ext=False, replace=False*`)`

Create a Certificate (CERT) signed by a named Certificate Authority (CA)

If the certificate file already exists, the function just returns assuming the CERT already exists.

The CN *must* match an existing CSR generated by create_csr. If it does not, this method does nothing.

ca_name name of the CA

CN common name matching the certificate signing request

days number of days certificate is valid, default is 365 (1 year)

cacert_path absolute path to ca certificates root directory

ca_filename alternative filename for the CA

New in version 2015.5.3.

cert_path full path to the certificates directory

cert_filename alternative filename for the certificate, useful when using special characters in the CN. If this option is set it will override the certificate filename output effects of `cert_type`. `type_ext` will be completely overridden.

New in version 2015.5.3.

digest The message digest algorithm. Must be a string describing a digest algorithm supported by OpenSSL (by EVP_get_digestbyname, specifically). For example, "md5" or "sha1". Default: 'sha256'

replace Replace this certificate even if it exists

New in version 2015.5.1.

cert_type string. Either 'server' or 'client' (see create_csr() for details).

If create_csr(type_ext=True) this function **must** be called with the same cert_type so it can find the CSR file.

Note: create_csr() defaults to cert_type='server'; therefore, if it was also called with type_ext, cert_type becomes a required argument for create_ca_signed_cert()

type_ext bool. If set True, use `cert_type` as an extension to the CN when formatting the filename.

e.g.: some_subject_CN_server.crt or some_subject_CN_client.crt

This facilitates the context where both types are required for the same subject

If `cert_filename` is *not None*, setting `type_ext` has no effect

If the following values were set:

```
ca.cert_base_path='/etc/pki'
ca_name='koji'
CN='test.egavas.org'
```

the resulting signed certificate would be written in the following location:

```
/etc/pki/koji/certs/test.egavas.org.crt
```

CLI Example:

```
salt '*' tls.create_ca_signed_cert test localhost
```

`salt.modules.tls.`**`create_csr`**(*ca_name, bits=2048, CN='localhost', C='US', ST='Utah', L='Salt Lake City', O='SaltStack', OU=None, emailAddress=None, subjectAltName=None, cacert_path=None, ca_filename=None, csr_path=None, csr_filename=None, digest='sha256', type_ext=False, cert_type='server', replace=False*)

Create a Certificate Signing Request (CSR) for a particular Certificate Authority (CA)

ca_name name of the CA

bits number of RSA key bits, default is 2048

CN common name in the request, default is "localhost"

C country, default is "US"

ST state, default is "Utah"

L locality, default is "Centerville", the city where SaltStack originated

O organization, default is "SaltStack" NOTE: Must the same as CA certificate or an error will be raised

OU organizational unit, default is None

emailAddress email address for the request, default is None

subjectAltName valid subjectAltNames in full form, e.g. to add DNS entry you would call this function with this value:

examples: ['DNS:somednsname.com', 'DNS:1.2.3.4', 'IP:1.2.3.4', 'IP:2001:4801:7821:77:be76:4eff:fe11:e51', 'email:me@i.like.pie.com']

Note: some libraries do not properly query IP: prefixes, instead looking for the given req. source with a DNS: prefix. To be thorough, you may want to include both DNS: and IP: entries if you are using subjectAltNames for destinations for your TLS connections. e.g.: requests to https://1.2.3.4 will fail from python's requests library w/out the second entry in the above list

---

New in version 2015.8.0.

cert_type Specify the general certificate type. Can be either *server* or *client*. Indicates the set of common extensions added to the CSR.

```
server: {
   'basicConstraints': 'CA:FALSE',
   'extendedKeyUsage': 'serverAuth',
   'keyUsage': 'digitalSignature, keyEncipherment'
}

client: {
   'basicConstraints': 'CA:FALSE',
   'extendedKeyUsage': 'clientAuth',
   'keyUsage': 'nonRepudiation, digitalSignature, keyEncipherment'
}
```

type_ext boolean. Whether or not to extend the filename with CN_[cert_type] This can be useful if a server and client certificate are needed for the same CN. Defaults to False to avoid introducing an unexpected file naming pattern

The files normally named some_subject_CN.csr and some_subject_CN.key will then be saved

replace Replace this signing request even if it exists

New in version 2015.5.1.

Writes out a Certificate Signing Request (CSR) If the file already exists, the function just returns assuming the CSR already exists.

If the following values were set:

```
ca.cert_base_path='/etc/pki'
ca_name='koji'
CN='test.egavas.org'
```

the resulting CSR, and corresponding key, would be written in the following location:

```
/etc/pki/koji/certs/test.egavas.org.csr
/etc/pki/koji/certs/test.egavas.org.key
```

CLI Example:

```
salt '*' tls.create_csr test
```

`salt.modules.tls.`**`create_empty_crl`**(*ca_name, cacert_path=None, ca_filename=None, crl_file=None*)

Create an empty Certificate Revocation List.

New in version 2015.8.0.

ca_name name of the CA

cacert_path absolute path to ca certificates root directory

ca_filename alternative filename for the CA

New in version 2015.5.3.

crl_file full path to the CRL file

CLI Example:

```
salt '*' tls.create_empty_crl ca_name='koji'                 ca_filename='ca'     ⮒
↪             crl_file='/etc/openvpn/team1/crl.pem'
```

`salt.modules.tls.`**`create_pkcs12`**(*ca_name, CN, passphrase='', cacert_path=None, replace=False*)

Create a PKCS#12 browser certificate for a particular Certificate (CN)

ca_name name of the CA

CN common name matching the certificate signing request

passphrase used to unlock the PKCS#12 certificate when loaded into the browser

cacert_path absolute path to ca certificates root directory

replace Replace this certificate even if it exists

New in version 2015.5.1.

If the following values were set:

```
ca.cert_base_path='/etc/pki'
ca_name='koji'
CN='test.egavas.org'
```

the resulting signed certificate would be written in the following location:

```
/etc/pki/koji/certs/test.egavas.org.p12
```

CLI Example:

```
salt '*' tls.create_pkcs12 test localhost
```

`salt.modules.tls.`**`create_self_signed_cert`**(*tls_dir='tls', bits=2048, days=365, CN='localhost', C='US', ST='Utah', L='Salt Lake City', O='SaltStack', OU=None, emailAddress=None, cacert_path=None, cert_filename=None, digest='sha256', replace=False*)

Create a Self-Signed Certificate (CERT)

tls_dir location appended to the ca.cert_base_path, default is 'tls'

bits number of RSA key bits, default is 2048

CN common name in the request, default is "localhost"

C country, default is "US"

ST state, default is ”Utah”
L locality, default is ”Centerville”, the city where SaltStack originated
O organization, default is ”SaltStack” NOTE: Must the same as CA certificate or an error will be raised
OU organizational unit, default is None
emailAddress email address for the request, default is None
cacert_path absolute path to ca certificates root directory
digest The message digest algorithm. Must be a string describing a digest algorithm supported by OpenSSL (by EVP_get_digestbyname, specifically). For example, ”md5” or ”sha1”. Default: ’sha256’
replace Replace this certificate even if it exists

New in version 2015.5.1.

Writes out a Self-Signed Certificate (CERT). If the file already exists, the function just returns.

If the following values were set:

```
ca.cert_base_path='/etc/pki'
tls_dir='koji'
CN='test.egavas.org'
```

the resulting CERT, and corresponding key, would be written in the following location:

```
/etc/pki/koji/certs/test.egavas.org.crt
/etc/pki/koji/certs/test.egavas.org.key
```

CLI Example:

```
salt '*' tls.create_self_signed_cert
```

Passing options from the command line:

```
salt 'minion' tls.create_self_signed_cert CN='test.mysite.org'
```

`salt.modules.tls.`**`get_ca`**`(`*ca_name, as_text=False, cacert_path=None*`)`

Get the certificate path or content
ca_name name of the CA
as_text if true, return the certificate content instead of the path
cacert_path absolute path to ca certificates root directory
CLI Example:

```
salt '*' tls.get_ca test_ca as_text=False cacert_path=/etc/certs
```

`salt.modules.tls.`**`get_ca_signed_cert`**`(`*ca_name, CN='localhost', as_text=False, cacert_path=None, cert_filename=None*`)`

Get the certificate path or content
ca_name name of the CA
CN common name of the certificate
as_text if true, return the certificate content instead of the path
cacert_path absolute path to certificates root directory
cert_filename alternative filename for the certificate, useful when using special characters in the CN

New in version 2015.5.3.

CLI Example:

```
salt '*' tls.get_ca_signed_cert test_ca CN=localhost as_text=False cacert_path=/
↪etc/certs
```

`salt.modules.tls.`**`get_ca_signed_key`**(*ca_name, CN='localhost', as_text=False, cacert_path=None, key_filename=None*)

Get the certificate path or content

ca_name name of the CA

CN common name of the certificate

as_text if true, return the certificate content instead of the path

cacert_path absolute path to certificates root directory

key_filename alternative filename for the key, useful when using special characters

New in version 2015.5.3.

in the CN

CLI Example:

```
salt '*' tls.get_ca_signed_key                 test_ca CN=localhost            ⊠
↪  as_text=False                     cacert_path=/etc/certs
```

`salt.modules.tls.`**`get_expiration_date`**(*cert, date_format='%Y-%m-%d'*)

New in version 2019.2.0.

Get a certificate's expiration date

cert Full path to the certificate

date_format By default this will return the expiration date in YYYY-MM-DD format, use this to specify a different strftime format string. Note that the expiration time will be in UTC.

CLI Examples:

```
salt '*' tls.get_expiration_date /path/to/foo.crt
salt '*' tls.get_expiration_date /path/to/foo.crt date_format='%d/%m/%Y'
```

`salt.modules.tls.`**`get_extensions`**(*cert_type*)

Fetch X509 and CSR extension definitions from tls:extensions: (common|server|client) or set them to standard defaults.

New in version 2015.8.0.

cert_type: The type of certificate such as `server` or `client`.

CLI Example:

```
salt '*' tls.get_extensions client
```

`salt.modules.tls.`**`maybe_fix_ssl_version`**(*ca_name, cacert_path=None, ca_filename=None*)

Check that the X509 version is correct (was incorrectly set in previous salt versions). This will fix the version if needed.

ca_name ca authority name

cacert_path absolute path to ca certificates root directory

ca_filename alternative filename for the CA

New in version 2015.5.3.

CLI Example:

```
salt '*' tls.maybe_fix_ssl_version test_ca /etc/certs
```

`salt.modules.tls.`**`revoke_cert`**(*ca_name, CN, cacert_path=None, ca_filename=None, cert_path=None, cert_filename=None, crl_file=None*)

Revoke a certificate.

New in version 2015.8.0.

ca_name Name of the CA.

CN Common name matching the certificate signing request.

cacert_path Absolute path to ca certificates root directory.

ca_filename Alternative filename for the CA.
cert_path Path to the cert file.
cert_filename Alternative filename for the certificate, useful when using special characters in the CN.
crl_file Full path to the CRL file.
CLI Example:

```
salt '*' tls.revoke_cert ca_name='koji'              ca_filename='ca'           
↪       crl_file='/etc/openvpn/team1/crl.pem'
```

`salt.modules.tls.`**`set_ca_path`**`(`*cacert_path*`)`
If wanted, store the aforementioned cacert_path in context to be used as the basepath for further operations

CLI Example:

```
salt '*' tls.set_ca_path /etc/certs
```

21.9.433 salt.modules.tomcat

Support for Tomcat

This module uses the manager webapp to manage Apache tomcat webapps. If the manager webapp is not configured some of the functions won't work.

configuration

- Java bin path should be in default path
- If ipv6 is enabled make sure you permit manager access to ipv6 interface "0:0:0:0:0:0:0:1"
- If you are using tomcat.tar.gz it has to be installed or symlinked under `/opt`, preferably using name tomcat
- "tomcat.signal start/stop" works but it does not use the startup scripts

The following grains/pillar should be set:

```
tomcat-manager:
  user: <username>
  passwd: <password>
```

or the old format:

```
tomcat-manager.user: <username>
tomcat-manager.passwd: <password>
```

Also configure a user in the conf/tomcat-users.xml file:

```
<?xml version='1.0' encoding='utf-8'?>
<tomcat-users>
    <role rolename="manager-script"/>
    <user username="tomcat" password="tomcat" roles="manager-script"/>
</tomcat-users>
```

Note:

- More information about tomcat manager: http://tomcat.apache.org/tomcat-7.0-doc/manager-howto.html

- if you use only this module for deployments you've might want to strict access to the manager only from localhost for more info: http://tomcat.apache.org/tomcat-7.0-doc/manager-howto.html#Configuring_Manager_Application_Access
- Tested on:

  JVM Vendor: Sun Microsystems Inc.

  JVM Version: 1.6.0_43-b01

  OS Architecture: amd64

  OS Name: Linux

  OS Version: 2.6.32-358.el6.x86_64

  Tomcat Version: Apache Tomcat/7.0.37

`salt.modules.tomcat.`**`deploy_war`**(*war, context, force='no', url='http://localhost:8080/manager', saltenv='base', timeout=180, temp_war_location=None, version=True*)

Deploy a WAR file

war absolute path to WAR file (should be accessible by the user running tomcat) or a path supported by the salt.modules.cp.get_file function

context the context path to deploy

force [False] set True to deploy the webapp even one is deployed in the context

url [http://localhost:8080/manager] the URL of the server manager webapp

saltenv [base] the environment for WAR file in used by salt.modules.cp.get_url function

timeout [180] timeout for HTTP request

temp_war_location [None] use another location to temporarily copy to war file by default the system's temp directory is used

version ["] Specify the war version. If this argument is provided, it overrides the version encoded in the war file name, if one is present.

Examples:

```
salt '*' tomcat.deploy_war salt://salt-2015.8.6.war version=2015.08.r6
```

New in version 2015.8.6.

CLI Examples:

cp module

```
salt '*' tomcat.deploy_war salt://application.war /api
salt '*' tomcat.deploy_war salt://application.war /api no
salt '*' tomcat.deploy_war salt://application.war /api yes http://localhost:8080/
↪manager
```

minion local file system

```
salt '*' tomcat.deploy_war /tmp/application.war /api
salt '*' tomcat.deploy_war /tmp/application.war /api no
salt '*' tomcat.deploy_war /tmp/application.war /api yes http://localhost:8080/
↪manager
```

`salt.modules.tomcat.`**`extract_war_version`**(*war*)

Extract the version from the war file name. There does not seem to be a standard for encoding the version into the war file name

Examples:

```
/path/salt-2015.8.6.war -> 2015.8.6
/path/V6R2013xD5.war -> None
```

`salt.modules.tomcat.`**`fullversion`**`()`

Return all server information from catalina.sh version

CLI Example:

```
salt '*' tomcat.fullversion
```

`salt.modules.tomcat.`**`leaks`**`(`*url='http://localhost:8080/manager', timeout=180*`)`

Find memory leaks in tomcat
url [http://localhost:8080/manager] the URL of the server manager webapp
timeout [180] timeout for HTTP request
CLI Examples:

```
salt '*' tomcat.leaks
```

`salt.modules.tomcat.`**`ls`**`(`*url='http://localhost:8080/manager', timeout=180*`)`

list all the deployed webapps
url [http://localhost:8080/manager] the URL of the server manager webapp
timeout [180] timeout for HTTP request
CLI Examples:

```
salt '*' tomcat.ls
salt '*' tomcat.ls http://localhost:8080/manager
```

`salt.modules.tomcat.`**`passwd`**`(`*passwd, user='', alg='sha1', realm=None*`)`

This function replaces the $CATALINA_HOME/bin/digest.sh script convert a clear-text password to the $CATALINA_BASE/conf/tomcat-users.xml format

CLI Examples:

```
salt '*' tomcat.passwd secret
salt '*' tomcat.passwd secret tomcat sha1
salt '*' tomcat.passwd secret tomcat sha1 'Protected Realm'
```

`salt.modules.tomcat.`**`reload_`**`(`*app, url='http://localhost:8080/manager', timeout=180*`)`

Reload the webapp
app the webapp context path
url [http://localhost:8080/manager] the URL of the server manager webapp
timeout [180] timeout for HTTP request
CLI Examples:

```
salt '*' tomcat.reload /jenkins
salt '*' tomcat.reload /jenkins http://localhost:8080/manager
```

`salt.modules.tomcat.`**`serverinfo`**`(`*url='http://localhost:8080/manager', timeout=180*`)`

return details about the server
url [http://localhost:8080/manager] the URL of the server manager webapp
timeout [180] timeout for HTTP request
CLI Examples:

```
salt '*' tomcat.serverinfo
salt '*' tomcat.serverinfo http://localhost:8080/manager
```

`salt.modules.tomcat.`**`sessions`**`(`*app*, *url='http://localhost:8080/manager'*, *timeout=180*`)`
return the status of the webapp sessions
app the webapp context path
url [http://localhost:8080/manager] the URL of the server manager webapp
timeout [180] timeout for HTTP request
CLI Examples:

```
salt '*' tomcat.sessions /jenkins
salt '*' tomcat.sessions /jenkins http://localhost:8080/manager
```

`salt.modules.tomcat.`**`signal`**`(`*signal=None*`)`
Signals catalina to start, stop, securestart, forcestop.

CLI Example:

```
salt '*' tomcat.signal start
```

`salt.modules.tomcat.`**`start`**`(`*app*, *url='http://localhost:8080/manager'*, *timeout=180*`)`
Start the webapp
app the webapp context path
url [http://localhost:8080/manager] the URL of the server manager webapp
timeout timeout for HTTP request
CLI Examples:

```
salt '*' tomcat.start /jenkins
salt '*' tomcat.start /jenkins http://localhost:8080/manager
```

`salt.modules.tomcat.`**`status`**`(`*url='http://localhost:8080/manager'*, *timeout=180*`)`
Used to test if the tomcat manager is up
url [http://localhost:8080/manager] the URL of the server manager webapp
timeout [180] timeout for HTTP request
CLI Examples:

```
salt '*' tomcat.status
salt '*' tomcat.status http://localhost:8080/manager
```

`salt.modules.tomcat.`**`status_webapp`**`(`*app*, *url='http://localhost:8080/manager'*, *timeout=180*`)`
return the status of the webapp (stopped | running | missing)
app the webapp context path
url [http://localhost:8080/manager] the URL of the server manager webapp
timeout [180] timeout for HTTP request
CLI Examples:

```
salt '*' tomcat.status_webapp /jenkins
salt '*' tomcat.status_webapp /jenkins http://localhost:8080/manager
```

`salt.modules.tomcat.`**`stop`**`(`*app*, *url='http://localhost:8080/manager'*, *timeout=180*`)`
Stop the webapp
app the webapp context path
url [http://localhost:8080/manager] the URL of the server manager webapp
timeout [180] timeout for HTTP request
CLI Examples:

```
salt '*' tomcat.stop /jenkins
salt '*' tomcat.stop /jenkins http://localhost:8080/manager
```

`salt.modules.tomcat.`**`undeploy`**`(`*app*, *url='http://localhost:8080/manager'*, *timeout=180*`)`

Undeploy a webapp

app the webapp context path

url [http://localhost:8080/manager] the URL of the server manager webapp

timeout [180] timeout for HTTP request

CLI Examples:

```
salt '*' tomcat.undeploy /jenkins
salt '*' tomcat.undeploy /jenkins http://localhost:8080/manager
```

`salt.modules.tomcat.`**`version`**`()`

Return server version from catalina.sh version

CLI Example:

```
salt '*' tomcat.version
```

21.9.434 salt.modules.trafficserver

Apache Traffic Server execution module.

New in version 2015.8.0.

`traffic_ctl` is used to execute individual Traffic Server commands and to script multiple commands in a shell.

`salt.modules.trafficserver.`**`alarms`**`()`

List all alarm events that have not been acknowledged (cleared).

```
salt '*' trafficserver.alarms
```

`salt.modules.trafficserver.`**`bounce_cluster`**`()`

Bounce all Traffic Server nodes in the cluster. Bouncing Traffic Server shuts down and immediately restarts Traffic Server, node-by-node.

```
salt '*' trafficserver.bounce_cluster
```

`salt.modules.trafficserver.`**`bounce_local`**`(`*drain=False*`)`

Bounce Traffic Server on the local node. Bouncing Traffic Server shuts down and immediately restarts the Traffic Server node.

drain This option modifies the restart behavior such that traffic_server is not shut down until the number of active client connections drops to the number given by the proxy.config.restart.active_client_threshold configuration variable.

```
salt '*' trafficserver.bounce_local
salt '*' trafficserver.bounce_local drain=True
```

`salt.modules.trafficserver.`**`clear_alarms`**`(`*alarm*`)`

Clear (acknowledge) an alarm event. The arguments are "all" for all current alarms, a specific alarm number (e.g. ''1''), or an alarm string identifier (e.g. ''MGMT_ALARM_PROXY_CONFIG_ERROR'').

```
salt '*' trafficserver.clear_alarms [all | #event | name]
```

`salt.modules.trafficserver.`**`clear_cluster`**`()`

Clears accumulated statistics on all nodes in the cluster.

```
salt '*' trafficserver.clear_cluster
```

`salt.modules.trafficserver.`**`clear_node`**`()`

Clears accumulated statistics on the local node.

```
salt '*' trafficserver.clear_node
```

`salt.modules.trafficserver.`**`match_config`**`(`*regex*`)`

Display the current values of all configuration variables whose names match the given regular expression.

New in version 2016.11.0.

```
salt '*' trafficserver.match_config regex
```

`salt.modules.trafficserver.`**`match_metric`**`(`*regex*`)`

Display the current values of all metrics whose names match the given regular expression.

New in version 2016.11.0.

```
salt '*' trafficserver.match_metric regex
```

`salt.modules.trafficserver.`**`offline`**`(`*path*`)`

Mark a cache storage device as offline. The storage is identified by a path which must match exactly a path specified in storage.config. This removes the storage from the cache and redirects requests that would have used this storage to other storage. This has exactly the same effect as a disk failure for that storage. This does not persist across restarts of the traffic_server process.

```
salt '*' trafficserver.offline /path/to/cache
```

`salt.modules.trafficserver.`**`read_config`**`(`**args*`)`

Read Traffic Server configuration variable definitions.

New in version 2016.11.0.

```
salt '*' trafficserver.read_config proxy.config.http.keep_alive_post_out
```

`salt.modules.trafficserver.`**`read_metric`**`(`**args*`)`

Read Traffic Server one or more metrics.

New in version 2016.11.0.

```
salt '*' trafficserver.read_metric proxy.process.http.tcp_hit_count_stat
```

`salt.modules.trafficserver.`**`refresh`**`()`

Initiate a Traffic Server configuration file reread. Use this command to update the running configuration after any configuration file modification.

The timestamp of the last reconfiguration event (in seconds since epoch) is published in the proxy.node.config.reconfigure_time metric.

```
salt '*' trafficserver.refresh
```

`salt.modules.trafficserver.`**`restart_cluster`**`()`

Restart the traffic_manager process and the traffic_server process on all the nodes in a cluster.

```
salt '*' trafficserver.restart_cluster
```

`salt.modules.trafficserver.`**`restart_local`**`(`*drain=False*`)`

Restart the traffic_manager and traffic_server processes on the local node.

drain This option modifies the restart behavior such that `traffic_server` is not shut down until the number of active client connections drops to the number given by the `proxy.config.restart.active_client_threshold` configuration variable.

```
salt '*' trafficserver.restart_local
salt '*' trafficserver.restart_local drain=True
```

`salt.modules.trafficserver.`**`set_config`**(*variable*, *value*)

Set the value of a Traffic Server configuration variable.

variable Name of a Traffic Server configuration variable.

value The new value to set.

New in version 2016.11.0.

```
salt '*' trafficserver.set_config proxy.config.http.keep_alive_post_out 0
```

`salt.modules.trafficserver.`**`shutdown`**()

Shut down Traffic Server on the local node.

```
salt '*' trafficserver.shutdown
```

`salt.modules.trafficserver.`**`startup`**()

Start Traffic Server on the local node.

```
salt '*' trafficserver.start
```

`salt.modules.trafficserver.`**`status`**()

Show the current proxy server status, indicating if we're running or not.

```
salt '*' trafficserver.status
```

`salt.modules.trafficserver.`**`zero_cluster`**()

Reset performance statistics to zero across the cluster.

```
salt '*' trafficserver.zero_cluster
```

`salt.modules.trafficserver.`**`zero_node`**()

Reset performance statistics to zero on the local node.

```
salt '*' trafficserver.zero_cluster
```

21.9.435 salt.modules.travisci

Commands for working with travisci.

depends pyOpenSSL >= 16.0.0

`salt.modules.travisci.`**`verify_webhook`**(*signature*, *body*)

Verify the webhook signature from travisci

signature The signature header from the webhook header

body The full payload body from the webhook post

Note: The body needs to be the urlencoded version of the body.

CLI Example:

```
salt '*' travisci.verify_webhook 'M6NucCX5722bxisQs7e...' 'payload=%7B%22id%22
↪%3A183791261%2C%22repository...'
```

21.9.436 salt.modules.tuned

Interface to Red Hat tuned-adm module

maintainer Syed Ali <alicsyed@gmail.com>

maturity new

depends tuned-adm

platform Linux

`salt.modules.tuned.`**`active`**`()`

Return current active profile

CLI Example:

```
salt '*' tuned.active
```

`salt.modules.tuned.`**`list_`**`()`

List the profiles available

CLI Example:

```
salt '*' tuned.list
```

`salt.modules.tuned.`**`off`**`()`

Turn off all profiles

CLI Example:

```
salt '*' tuned.off
```

`salt.modules.tuned.`**`profile`**`(`*profile_name*`)`

Activate specified profile

CLI Example:

```
salt '*' tuned.profile virtual-guest
```

21.9.437 salt.modules.twilio_notify

Module for notifications via Twilio

New in version 2014.7.0.

depends

- twilio python module

configuration Configure this module by specifying the name of a configuration profile in the minion config, minion pillar, or master config (with `pillar_opts` set to True).

For example:

```
my-twilio-account:
    twilio.account_sid: AC32a3c83990934481addd5ce1659f04d2
    twilio.auth_token: mytoken
```

salt.modules.twilio_notify.**send_sms**(*profile*, *body*, *to*, *from_*)

Send an sms

CLI Example:

twilio.send_sms my-twilio-account 'Test sms' '+18019999999' '+18011111111'

21.9.438 salt.modules.udev

Manage and query udev info

New in version 2015.8.0.

salt.modules.udev.**env**(*dev*)

Return all environment variables udev has for dev

CLI Example:

```
salt '*' udev.env /dev/sda
salt '*' udev.env /sys/class/net/eth0
```

salt.modules.udev.**exportdb**()

Return all the udev database

CLI Example:

```
salt '*' udev.exportdb
```

salt.modules.udev.**info**(*dev*)

Extract all info delivered by udevadm

CLI Example:

```
salt '*' udev.info /dev/sda
salt '*' udev.info /sys/class/net/eth0
```

salt.modules.udev.**links**(*dev*)

Return all udev-created device symlinks

CLI Example:

```
salt '*' udev.links /dev/sda
salt '*' udev.links /sys/class/net/eth0
```

salt.modules.udev.**name**(*dev*)

Return the actual dev name(s?) according to udev for dev

CLI Example:

```
salt '*' udev.dev /dev/sda
salt '*' udev.dev /sys/class/net/eth0
```

salt.modules.udev.**path**(*dev*)

Return the physical device path(s?) according to udev for dev

CLI Example:

```
salt '*' udev.path /dev/sda
salt '*' udev.path /sys/class/net/eth0
```

21.9.439 salt.modules.upstart_service

Module for the management of upstart systems. The Upstart system only supports service starting, stopping and restarting.

Important: If you feel that Salt should be using this module to manage services on a minion, and it is using a different module (or gives an error similar to *'service.start' is not available*), see here.

Currently (as of Ubuntu 12.04) there is no tool available to disable Upstart services (like update-rc.d). This[1] is the recommended way to disable an Upstart service. So we assume that all Upstart services that have not been disabled in this manner are enabled.

But this is broken because we do not check to see that the dependent services are enabled. Otherwise we would have to do something like parse the output of "initctl show-config" to determine if all service dependencies are enabled to start on boot. For example, see the "start on" condition for the lightdm service below[2]. And this would be too hard. So we wait until the upstart developers have solved this problem. :) This is to say that an Upstart service that is enabled may not really be enabled.

Also, when an Upstart service is enabled, should the dependent services be enabled too? Probably not. But there should be a notice about this, at least.

[1] http://upstart.ubuntu.com/cookbook/#disabling-a-job-from-automatically-starting

[2] example upstart configuration file:

```
lightdm
emits login-session-start
emits desktop-session-start
emits desktop-shutdown
start on (((filesystem and runlevel [!06]) and started dbus) and (drm-device-added
↪card0 PRIMARY_DEVICE_FOR_DISPLAY=1 or stopped udev-fallback-graphics)) or runlevel
↪PREVLEVEL=S)
stop on runlevel [016]
```

Warning: This module should not be used on Red Hat systems. For these, the `rh_service` module should be used, as it supports the hybrid upstart/sysvinit system used in RHEL/CentOS 6.

`salt.modules.upstart_service.`**`available`**`(`*name*`)`

Returns `True` if the specified service is available, otherwise returns `False`.

CLI Example:

```
salt '*' service.available sshd
```

`salt.modules.upstart_service.`**`disable`**`(`*name*, ***kwargs*`)`

Disable the named service from starting on boot

CLI Example:

```
salt '*' service.disable <service name>
```

`salt.modules.upstart_service.`**`disabled`**`(`*name*`)`

Check to see if the named service is disabled to start on boot

CLI Example:

```
salt '*' service.disabled <service name>
```

`salt.modules.upstart_service.`**`enable`**`(`*name*, ***kwargs*`)`

Enable the named service to start at boot

CLI Example:

```
salt '*' service.enable <service name>
```

`salt.modules.upstart_service.`**`enabled`**`(`*name*, ***kwargs*`)`

Check to see if the named service is enabled to start on boot

CLI Example:

```
salt '*' service.enabled <service name>
```

`salt.modules.upstart_service.`**`force_reload`**`(`*name*`)`

Force-reload the named service

CLI Example:

```
salt '*' service.force_reload <service name>
```

`salt.modules.upstart_service.`**`full_restart`**`(`*name*`)`

Do a full restart (stop/start) of the named service

CLI Example:

```
salt '*' service.full_restart <service name>
```

`salt.modules.upstart_service.`**`get_all`**`()`

Return all installed services

CLI Example:

```
salt '*' service.get_all
```

`salt.modules.upstart_service.`**`get_disabled`**`()`

Return the disabled services

CLI Example:

```
salt '*' service.get_disabled
```

`salt.modules.upstart_service.`**`get_enabled`**`()`

Return the enabled services

CLI Example:

```
salt '*' service.get_enabled
```

`salt.modules.upstart_service.`**`missing`**`(`*name*`)`

The inverse of service.available. Returns `True` if the specified service is not available, otherwise returns `False`.

CLI Example:

```
salt '*' service.missing sshd
```

salt.modules.upstart_service.**reload_**(*name*)
Reload the named service

CLI Example:

```
salt '*' service.reload <service name>
```

salt.modules.upstart_service.**restart**(*name*)
Restart the named service

CLI Example:

```
salt '*' service.restart <service name>
```

salt.modules.upstart_service.**start**(*name*)
Start the specified service

CLI Example:

```
salt '*' service.start <service name>
```

salt.modules.upstart_service.**status**(*name*, *sig=None*)
Return the status for a service. If the name contains globbing, a dict mapping service name to True/False values is returned.

Changed in version 2018.3.0: The service name can now be a glob (e.g. `salt*`)

Parameters

- **name** (*str*) – The name of the service to check
- **sig** (*str*) – Signature to use to find the service via ps

Returns True if running, False otherwise dict: Maps service name to True if running, False otherwise

Return type bool

CLI Example:

```
salt '*' service.status <service name> [service signature]
```

salt.modules.upstart_service.**stop**(*name*)
Stop the specified service

CLI Example:

```
salt '*' service.stop <service name>
```

21.9.440 salt.modules.uptime

Wrapper around uptime API

salt.modules.uptime.**check_exists**(*name*)
Check if a given URL is in being monitored by uptime

CLI Example:

```
salt '*' uptime.check_exists http://example.org
```

`salt.modules.uptime.`**`checks_list`**`()`

List URL checked by uptime

CLI Example:

```
salt '*' uptime.checks_list
```

`salt.modules.uptime.`**`create`**`(`*name*, ***params*`)`

Create a check on a given URL.

Additional parameters can be used and are passed to API (for example interval, maxTime, etc). See the documentation https://github.com/fzaninotto/uptime for a full list of the parameters.

CLI Example:

```
salt '*' uptime.create http://example.org
```

`salt.modules.uptime.`**`delete`**`(`*name*`)`

Delete a check on a given URL

CLI Example:

```
salt '*' uptime.delete http://example.org
```

21.9.441 salt.modules.useradd

Manage users with the useradd command

Important: If you feel that Salt should be using this module to manage users on a minion, and it is using a different module (or gives an error similar to *'user.info' is not available*), see *here*.

`salt.modules.useradd.`**`add`**`(`*name*, *uid=None*, *gid=None*, *groups=None*, *home=None*, *shell=None*, *unique=True*, *system=False*, *fullname=''*, *roomnumber=''*, *workphone=''*, *homephone=''*, *other=''*, *createhome=True*, *loginclass=None*, *nologinit=False*, *root=None*`)`

Add a user to the minion

name Username LOGIN to add

uid User ID of the new account

gid Name or ID of the primary group of the new accoun

groups List of supplementary groups of the new account

home Home directory of the new account

shell Login shell of the new account

unique Allow to create users with duplicate

system Create a system account

fullname GECOS field for the full name

roomnumber GECOS field for the room number

workphone GECOS field for the work phone

homephone GECOS field for the home phone

other GECOS field for other information

createhome Create the user's home directory

loginclass Login class for the new account (OpenBSD)

nologinit Do not add the user to the lastlog and faillog databases

root Directory to chroot into

CLI Example:

```
salt '*' user.add name <uid> <gid> <groups> <home> <shell>
```

`salt.modules.useradd.`**`chfullname`**(*name*, *fullname*, *root=None*)

Change the user's Full Name

name User to modify

fullname GECOS field for the full name

root Directory to chroot into

CLI Example:

```
salt '*' user.chfullname foo "Foo Bar"
```

`salt.modules.useradd.`**`chgid`**(*name*, *gid*, *root=None*)

Change the default group of the user

name User to modify

gid Force use GID as new primary group

root Directory to chroot into

CLI Example:

```
salt '*' user.chgid foo 4376
```

`salt.modules.useradd.`**`chgroups`**(*name*, *groups*, *append=False*, *root=None*)

Change the groups to which this user belongs

name User to modify

groups Groups to set for the user

append [False] If `True`, append the specified group(s). Otherwise, this function will replace the user's groups with the specified group(s).

root Directory to chroot into

CLI Examples:

```
salt '*' user.chgroups foo wheel,root
salt '*' user.chgroups foo wheel,root append=True
```

`salt.modules.useradd.`**`chhome`**(*name*, *home*, *persist=False*, *root=None*)

Change the home directory of the user, pass True for persist to move files to the new home directory if the old home directory exist.

name User to modify

home New home directory for the user account

presist Move contents of the home directory to the new location

root Directory to chroot into

CLI Example:

```
salt '*' user.chhome foo /home/users/foo True
```

`salt.modules.useradd.`**`chhomephone`**(*name*, *homephone*, *root=None*)

Change the user's Home Phone

name User to modify

homephone GECOS field for the home phone

root Directory to chroot into

CLI Example:

```
salt '*' user.chhomephone foo 7735551234
```

`salt.modules.useradd.`**`chloginclass`**(*name*, *loginclass*, *root=None*)

Change the default login class of the user

name User to modify

loginclass Login class for the new account
root Directory to chroot into

Note: This function only applies to OpenBSD systems.

CLI Example:

```
salt '*' user.chloginclass foo staff
```

`salt.modules.useradd.`**`chother`**(*name*, *other*, *root=None*)
Change the user's other GECOS attribute
name User to modify
other GECOS field for other information
root Directory to chroot into
CLI Example:

```
salt '*' user.chother foobar
```

`salt.modules.useradd.`**`chroomnumber`**(*name*, *roomnumber*, *root=None*)
Change the user's Room Number

CLI Example:

```
salt '*' user.chroomnumber foo 123
```

`salt.modules.useradd.`**`chshell`**(*name*, *shell*, *root=None*)
Change the default shell of the user
name User to modify
shell New login shell for the user account
root Directory to chroot into
CLI Example:

```
salt '*' user.chshell foo /bin/zsh
```

`salt.modules.useradd.`**`chuid`**(*name*, *uid*, *root=None*)
Change the uid for a named user
name User to modify
uid New UID for the user account
root Directory to chroot into
CLI Example:

```
salt '*' user.chuid foo 4376
```

`salt.modules.useradd.`**`chworkphone`**(*name*, *workphone*, *root=None*)
Change the user's Work Phone
name User to modify
workphone GECOS field for the work phone
root Directory to chroot into
CLI Example:

```
salt '*' user.chworkphone foo 7735550123
```

`salt.modules.useradd.`**`delete`**(*name*, *remove=False*, *force=False*, *root=None*)
Remove a user from the minion
name Username to delete
remove Remove home directory and mail spool

force Force some actions that would fail otherwise
root Directory to chroot into
CLI Example:

```
salt '*' user.delete name remove=True force=True
```

`salt.modules.useradd.`**`get_loginclass`**`(`*name*`)`
Get the login class of the user
name User to get the information

Note: This function only applies to OpenBSD systems.

CLI Example:

```
salt '*' user.get_loginclass foo
```

`salt.modules.useradd.`**`getent`**`(`*refresh=False, root=None*`)`
Return the list of all info for all users
refresh Force a refresh of user information
root Directory to chroot into
CLI Example:

```
salt '*' user.getent
```

`salt.modules.useradd.`**`info`**`(`*name, root=None*`)`
Return user information
name User to get the information
root Directory to chroot into
CLI Example:

```
salt '*' user.info root
```

`salt.modules.useradd.`**`list_groups`**`(`*name*`)`
Return a list of groups the named user belongs to
name User to get the information
CLI Example:

```
salt '*' user.list_groups foo
```

`salt.modules.useradd.`**`list_users`**`(`*root=None*`)`
Return a list of all users
root Directory to chroot into
CLI Example:

```
salt '*' user.list_users
```

`salt.modules.useradd.`**`primary_group`**`(`*name*`)`
Return the primary group of the named user

New in version 2016.3.0.
name User to get the information
CLI Example:

```
salt '*' user.primary_group saltadmin
```

`salt.modules.useradd.`**`rename`**`(`*name*, *new_name*, *root=None*`)`

Change the username for a named user

name User to modify

new_name New value of the login name

root Directory to chroot into

CLI Example:

```
salt '*' user.rename name new_name
```

21.9.442 salt.modules.uwsgi

uWSGI stats server https://uwsgi-docs.readthedocs.io/en/latest/StatsServer.html

maintainer Peter Baumgartner <pete@lincolnloop.com>

maturity new

platform all

`salt.modules.uwsgi.`**`stats`**`(`*socket*`)`

Return the data from *uwsgi –connect-and-read* as a dictionary.

socket The socket the uWSGI stats server is listening on

CLI Example:

```
salt '*' uwsgi.stats /var/run/mystatsserver.sock

salt '*' uwsgi.stats 127.0.0.1:5050
```

21.9.443 salt.modules.vagrant

Work with virtual machines managed by Vagrant.

New in version 2018.3.0.

Mapping between a Salt node id and the Vagrant machine name (and the path to the Vagrantfile where it is defined) is stored in a Salt sdb database on the Vagrant host (minion) machine. In order to use this module, sdb must be configured. An SQLite database is the recommended storage method. The URI used for the sdb lookup is "sdb://vagrant_sdb_data".

requirements:

- the VM host machine must have salt-minion, Vagrant and a vm provider installed.
- the VM host must have a valid definition for *sdb://vagrant_sdb_data*

  Configuration example:

  ```
  # file /etc/salt/minion.d/vagrant_sdb.conf
  vagrant_sdb_data:
    driver: sqlite3
    database: /var/cache/salt/vagrant.sqlite
    table: sdb
    create_table: True
  ```

`salt.modules.vagrant.`**`destroy`**`(`*name*`)`

Destroy and delete a virtual machine. (vagrant destroy -f)

This also removes the salt_id name defined by vagrant.init.

CLI Example:

```
salt <host> vagrant.destroy <salt_id>
```

`salt.modules.vagrant.`**`get_machine_id`**`(`*machine*, *cwd*`)`

returns the salt_id name of the Vagrant VM

Parameters

- **machine** – the Vagrant machine name
- **cwd** – the path to Vagrantfile

Returns salt_id name

`salt.modules.vagrant.`**`get_ssh_config`**`(`*name*, *network_mask=''*, *get_private_key=False*`)`

Retrieve hints of how you might connect to a Vagrant VM.

Parameters

- **name** – the salt_id of the machine
- **network_mask** – a CIDR mask to search for the VM's address
- **get_private_key** – (default: False) return the key used for ssh login

Returns a dict of ssh login information for the VM

CLI Example:

```
salt <host> vagrant.get_ssh_config <salt_id>
salt my_laptop vagrant.get_ssh_config quail1 network_mask=10.0.0.0/8 get_private_
↪key=True
```

The returned dictionary contains:

- key_filename: the name of the private key file on the VM host computer
- ssh_username: the username to be used to log in to the VM
- ssh_host: the IP address used to log in to the VM. (This will usually be *127.0.0.1*)
- ssh_port: the TCP port used to log in to the VM. (This will often be *2222*)
- [ip_address:] (if *network_mask* is defined. see below)
- [private_key:] (if *get_private_key* is True) the private key for ssh_username

About *network_mask*:

Vagrant usually uses a redirected TCP port on its host computer to log in to a VM using ssh. This redirected port and its IP address are "ssh_port" and "ssh_host". The ssh_host is usually the localhost (127.0.0.1). This makes it impossible for a third machine (such as a salt-cloud master) to contact the VM unless the VM has another network interface defined. You will usually want a bridged network defined by having a *config.vm.network "public_network"* statement in your *Vagrantfile*.

The IP address of the bridged adapter will typically be assigned by DHCP and unknown to you, but you should be able to determine what IP network the address will be chosen from. If you enter a CIDR network mask, Salt will attempt to find the VM's address for you. The host machine will send an "ifconfig" command to the VM (using ssh to *ssh_host*:*ssh_port*) and return the IP address of the first interface it can find which matches your mask.

`salt.modules.vagrant.`**`get_vm_info`**`(`*name*`)`

get the information for a VM.

Parameters **name** – salt_id name

Returns dictionary of {'machine': x, 'cwd': y, ...}.

`salt.modules.vagrant.`**`init`**`(`*name*, *cwd=None*, *machine=''*, *runas=None*, *start=False*, *vagrant_provider=''*, *vm=None*`)`

Initialize a new Vagrant VM.

This inputs all the information needed to start a Vagrant VM. These settings are stored in a Salt sdb database on the Vagrant host minion and used to start, control, and query the guest VMs. The salt_id assigned here is the key field for that database and must be unique.

Parameters

- **name** – The salt_id name you will use to control this VM
- **cwd** – The path to the directory where the Vagrantfile is located
- **machine** – The machine name in the Vagrantfile. If blank, the primary machine will be used.
- **runas** – The username on the host who owns the Vagrant work files.
- **start** – (default: False) Start the virtual machine now.
- **vagrant_provider** – The name of a Vagrant VM provider (if not the default).
- **vm** – Optionally, all the above information may be supplied in this dictionary.

Returns A string indicating success, or False.

CLI Example:

```
salt <host> vagrant.init <salt_id> /path/to/Vagrantfile
salt my_laptop vagrant.init x1 /projects/bevy_master machine=quail1
```

`salt.modules.vagrant.`**`list_active_vms`**`(`*cwd=None*`)`

Return a list of machine names for active virtual machine on the host, which are defined in the Vagrantfile at the indicated path.

CLI Example:

```
salt '*' vagrant.list_active_vms  cwd=/projects/project_1
```

`salt.modules.vagrant.`**`list_domains`**`()`

Return a list of the salt_id names of all available Vagrant VMs on this host without regard to the path where they are defined.

CLI Example:

```
salt '*' vagrant.list_domains --log-level=info
```

The log shows information about all known Vagrant environments on this machine. This data is cached and may not be completely up-to-date.

`salt.modules.vagrant.`**`list_inactive_vms`**`(`*cwd=None*`)`

Return a list of machine names for inactive virtual machine on the host, which are defined in the Vagrantfile at the indicated path.

CLI Example:

```
salt '*' virt.list_inactive_vms cwd=/projects/project_1
```

`salt.modules.vagrant.`**`pause`**`(`*name*`)`

Pause (vagrant suspend) the named VM.

CLI Example:

```
salt <host> vagrant.pause <salt_id>
```

`salt.modules.vagrant.`**`reboot`**`(`*name, provision=False*`)`

Reboot a VM. (vagrant reload)

CLI Example:

```
salt <host> vagrant.reboot <salt_id> provision=True
```

Parameters

- **name** – The salt_id name you will use to control this VM
- **provision** – (False) also re-run the Vagrant provisioning scripts.

`salt.modules.vagrant.`**`shutdown`**`(`*name*`)`

Send a soft shutdown (vagrant halt) signal to the named vm.

This does the same thing as vagrant.stop. Other-VM control modules use "stop" and "shutdown" to differentiate between hard and soft shutdowns.

CLI Example:

```
salt <host> vagrant.shutdown <salt_id>
```

`salt.modules.vagrant.`**`start`**`(`*name*`)`

Start (vagrant up) a virtual machine defined by salt_id name. The machine must have been previously defined using "vagrant.init".

CLI Example:

```
salt <host> vagrant.start <salt_id>
```

`salt.modules.vagrant.`**`stop`**`(`*name*`)`

Hard shutdown the virtual machine. (vagrant halt)

CLI Example:

```
salt <host> vagrant.stop <salt_id>
```

`salt.modules.vagrant.`**`version`**`()`

Return the version of Vagrant on the minion

CLI Example:

```
salt '*' vagrant.version
```

`salt.modules.vagrant.`**`vm_state`**`(`*name=", cwd=None*`)`

Return list of information for all the vms indicating their state.

If you pass a VM name in as an argument then it will return info for just the named VM, otherwise it will return all VMs defined by the Vagrantfile in the *cwd* directory.

CLI Example:

```
salt '*' vagrant.vm_state <name>  cwd=/projects/project_1
```

returns a list of dictionaries with machine name, state, provider, and salt_id name.

```
datum = {'machine': _, # Vagrant machine name,
         'state': _, # string indicating machine state, like 'running'
         'provider': _, # the Vagrant VM provider
         'name': _} # salt_id name
```

Known bug: if there are multiple machines in your Vagrantfile, and you request the status of the `primary` machine, which you defined by leaving the `machine` parameter blank, then you may receive the status of all of them. Please specify the actual machine name for each VM if there are more than one.

21.9.444 salt.modules.varnish

Support for Varnish

New in version 2014.7.0.

Note: These functions are designed to work with all implementations of Varnish from 3.x onwards

---

`salt.modules.varnish.`**`ban`**`(`*ban_expression*`)`

Add ban to the varnish cache

CLI Example:

```
salt '*' varnish.ban ban_expression
```

`salt.modules.varnish.`**`ban_list`**`()`

List varnish cache current bans

CLI Example:

```
salt '*' varnish.ban_list
```

`salt.modules.varnish.`**`param_set`**`(`*param, value*`)`

Set a param in varnish cache

CLI Example:

```
salt '*' varnish.param_set param value
```

`salt.modules.varnish.`**`param_show`**`(`*param=None*`)`

Show params of varnish cache

CLI Example:

```
salt '*' varnish.param_show param
```

`salt.modules.varnish.`**`purge`**`()`

Purge the varnish cache

CLI Example:

```
salt '*' varnish.purge
```

`salt.modules.varnish.`**`version`**`()`

Return server version from varnishd -V

CLI Example:

```
salt '*' varnish.version
```

21.9.445 salt.modules.vault module

Functions to interact with Hashicorp Vault.

maintainer SaltStack

maturity new

platform all

note If you see the following error, you'll need to upgrade `requests` to atleast 2.4.2

```
<timestamp> [salt.pillar][CRITICAL][14337] Pillar render error: Failed to load ext_
↪pillar vault: {'error': "request() got an unexpected keyword argument 'json'"}
```

configuration The salt-master must be configured to allow peer-runner configuration, as well as configuration for the module.

Add this segment to the master configuration file, or /etc/salt/master.d/vault.conf:

```
vault:
    url: https://vault.service.domain:8200
    verify: /etc/ssl/certs/ca-certificates.crt
    role_name: minion_role
    auth:
        method: approle
        role_id: 11111111-2222-3333-4444-1111111111111
        secret_id: 11111111-1111-1111-1111-1111111111111
    policies:
        - saltstack/minions
        - saltstack/minion/{minion}
        .. more policies
    keys:
        - n63/TbrQuL3xaIW7ZZpuXj/tIfnK1/MbVxO4vT3wYD2A
        - S9OwCvMRhErEA4NVVELYBs6w/Me6+urgUr24xGK44Uy3
        - F1j4b7JKq850NS6Kboiy5laJ0xY8dWJvB3fcwA+SraYl
        - 1cYtvjKJNDVam9c7HNqJUfINk4PYyAXIpjkpN/sIuzPv
        - 3pPK5X6vGtwLhNOFv1U2elahECz3HpRUfNXJFYLw6lid
```

url Url to your Vault installation. Required.

verify For details please see http://docs.python-requests.org/en/master/user/advanced/#ssl-cert-verification

New in version 2018.3.0.

role_name Role name for minion tokens created. If omitted, minion tokens will be created without any role, thus being able to inherit any master token policy (including token creation capabilities). Optional.

For details please see: https://www.vaultproject.io/api/auth/token/index.html#create-token Example configuration: https://www.nomadproject.io/docs/vault-integration/index.html#vault-token-role-configuration

auth Currently only token and approle auth types are supported. Required.

Approle is the preferred way to authenticate with Vault as it provide some advanced options to control authentication process. Please visit Vault documentation for more info: https://www.vaultproject.io/docs/auth/approle.html

The token must be able to create tokens with the policies that should be assigned to minions.

You can still use the token auth via a OS environment variable via this config example:

```
vault:
  url: https://vault.service.domain:8200
  auth:
    method: token
    token: sdb://osenv/VAULT_TOKEN
osenv:
  driver: env
```

And then export the VAULT_TOKEN variable in your OS:

```
export VAULT_TOKEN=11111111-1111-1111-1111-1111111111111
```

policies Policies that are assigned to minions when requesting a token. These can either be static, eg saltstack/minions, or templated, eg `saltstack/minion/{minion}`. `{minion}` is shorthand for grains[id]. Grains are also available, for example like this: `my-policies/{grains[os]}`

If a template contains a grain which evaluates to a list, it will be expanded into multiple policies. For example, given the template `saltstack/by-role/{grains[roles]}`, and a minion having these grains:

```
grains:
    roles:
        - web
        - database
```

The minion will have the policies `saltstack/by-role/web` and `saltstack/by-role/database`. Note however that list members which do not have simple string representations, such as dictionaries or objects, do not work and will throw an exception. Strings and numbers are examples of types which work well.

Optional. If policies is not configured, `saltstack/minions` and `saltstack/{minion}` are used as defaults.

keys List of keys to use to unseal vault server with the vault.unseal runner.

Add this segment to the master configuration file, or /etc/salt/master.d/peer_run.conf:

```
peer_run:
    .*:
        - vault.generate_token
```

`salt.modules.vault.`**`delete_secret`**`(`*path*`)`

Delete secret at the path in vault. The vault policy used must allow this.

CLI Example:

```
salt '*' vault.delete_secret "secret/my/secret"
```

`salt.modules.vault.`**`list_secrets`**`(`*path*`)`

List secret keys at the path in vault. The vault policy used must allow this. The path should end with a trailing slash.

CLI Example:

```
salt '*' vault.list_secrets "secret/my/"
```

`salt.modules.vault.`**`read_secret`**`(`*path*, *key=None*`)`

Return the value of key at path in vault, or entire secret

Jinja Example:

```
my-secret: {{ salt['vault'].read_secret('secret/my/secret', 'some-key') }}
```

```
{% set supersecret = salt['vault'].read_secret('secret/my/secret') %}
secrets:
    first: {{ supersecret.first }}
    second: {{ supersecret.second }}
```

`salt.modules.vault.`**`write_raw`**`(`*path*, *raw*`)`

Set raw data at the path in vault. The vault policy used must allow this.

CLI Example:

```
salt '*' vault.write_raw "secret/my/secret" '{"user":"foo","password": "bar"}'
```

`salt.modules.vault.`**`write_secret`**`(`*path*, ***kwargs*`)`

Set secret at the path in vault. The vault policy used must allow this.

CLI Example:

```
salt '*' vault.write_secret "secret/my/secret" user="foo" password="bar"
```

21.9.446 salt.modules.vbox_guest

VirtualBox Guest Additions installer

`salt.modules.vbox_guest.`**`additions_install`**`(`***kwargs*`)`

Install VirtualBox Guest Additions. Uses the CD, connected by VirtualBox.

To connect VirtualBox Guest Additions via VirtualBox graphical interface press 'Host+D' ('Host' is usually 'Right Ctrl').

See https://www.virtualbox.org/manual/ch04.html#idp52733088 for more details.

CLI Example:

```
salt '*' vbox_guest.additions_install
salt '*' vbox_guest.additions_install reboot=True
salt '*' vbox_guest.additions_install upgrade_os=True
```

Parameters

- **`reboot`** (*bool*) – reboot computer to complete installation
- **`upgrade_os`** (*bool*) – upgrade OS (to ensure the latests version of kernel and developer tools are installed)

Returns version of VirtualBox Guest Additions or string with error

`salt.modules.vbox_guest.`**`additions_mount`**`()`

Mount VirtualBox Guest Additions CD to the temp directory.

To connect VirtualBox Guest Additions via VirtualBox graphical interface press 'Host+D' ('Host' is usually 'Right Ctrl').

CLI Example:

```
salt '*' vbox_guest.additions_mount
```

Returns True or OSError exception

`salt.modules.vbox_guest.`**`additions_remove`**`(`***kwargs*`)`

Remove VirtualBox Guest Additions.

Firstly it tries to uninstall itself by executing '/opt/VBoxGuestAdditions-VERSION/uninstall.run uninstall'. It uses the CD, connected by VirtualBox if it failes.

CLI Example:

```
salt '*' vbox_guest.additions_remove
salt '*' vbox_guest.additions_remove force=True
```

Parameters **`force`** (*bool*) – force VirtualBox Guest Additions removing

Returns True if VirtualBox Guest Additions were removed successfully else False `salt.modules.vbox_guest.`**`additions_umount`**`(`*mount_point*`)`

Unmount VirtualBox Guest Additions CD from the temp directory.

CLI Example:

```
salt '*' vbox_guest.additions_umount
```

Parameters **`mount_point`** – directory VirtualBox Guest Additions is mounted to

Returns True or an string with error

`salt.modules.vbox_guest.`**`additions_version`**`()`

Check VirtualBox Guest Additions version.

CLI Example:

```
salt '*' vbox_guest.additions_version
```

Returns version of VirtualBox Guest Additions or False if they are not installed

`salt.modules.vbox_guest.`**`grant_access_to_shared_folders_to`**`(`*name*, *users=None*`)`

Grant access to auto-mounted shared folders to the users.

User is specified by it's name. To grant access for several users use argument *users*. Access will be denied to the users not listed in *users* argument.

See https://www.virtualbox.org/manual/ch04.html#sf_mount_auto for more details.

CLI Example:

```
salt '*' vbox_guest.grant_access_to_shared_folders_to fred
salt '*' vbox_guest.grant_access_to_shared_folders_to users ['fred', 'roman']
```

Parameters

- **`name`** (*`str`*) – name of the user to grant access to auto-mounted shared folders to
- **`users`** (*`list of str`*) – list of names of users to grant access to auto-mounted shared folders to (if specified, *name* will not be taken into account)

Returns list of users who have access to auto-mounted shared folders

`salt.modules.vbox_guest.`**`list_shared_folders_users`**`()`

List users who have access to auto-mounted shared folders.

See https://www.virtualbox.org/manual/ch04.html#sf_mount_auto for more details.

CLI Example:

```
salt '*' vbox_guest.list_shared_folders_users
```

Returns list of users who have access to auto-mounted shared folders

21.9.447 salt.modules.vboxmanage module

Support for VirtualBox using the VBoxManage command

New in version 2016.3.0.

If the `vboxdrv` kernel module is not loaded, this module can automatically load it by configuring `autoload_vboxdrv` in `/etc/salt/minion`:

```
autoload_vboxdrv: True
```

The default for this setting is `False`.

depends virtualbox

`salt.modules.vboxmanage.`**`clonemedium`**`(`*medium, uuid_in=None, file_in=None, uuid_out=None, file_out=None, mformat=None, variant=None, existing=False, **kwargs*`)`

Clone a new VM from an existing VM

CLI Example:

```
salt 'hypervisor' vboxmanage.clonemedium <name> <new_name>
```

`salt.modules.vboxmanage.`**`clonevm`**`(`*name=None, uuid=None, new_name=None, snapshot_uuid=None, snapshot_name=None, mode='machine', options=None, basefolder=None, new_uuid=None, register=False, groups=None, **kwargs*`)`

Clone a new VM from an existing VM

CLI Example:

```
salt 'hypervisor' vboxmanage.clonevm <name> <new_name>
```

`salt.modules.vboxmanage.`**`create`**`(`*name, groups=None, ostype=None, register=True, basefolder=None, new_uuid=None, **kwargs*`)`

Create a new VM

CLI Example:

```
salt 'hypervisor' vboxmanage.create <name>
```

`salt.modules.vboxmanage.`**`destroy`**`(`*name*`)`

Unregister and destroy a VM

CLI Example:

```
salt '*' vboxmanage.destroy my_vm
```

`salt.modules.vboxmanage.`**`list_items`**`(`*item, details=False, group_by='UUID'*`)`

Return a list of a specific type of item. The following items are available:

> vms runningvms ostypes hostdvds hostfloppies intnets bridgedifs hostonlyifs natnets dhcpservers hostinfo hostcpuids hddbackends hdds dvds floppies usbhost usbfilters systemproperties extpacks groups webcams screenshotformats

CLI Example:

```
salt 'hypervisor' vboxmanage.items <item>
salt 'hypervisor' vboxmanage.items <item> details=True
salt 'hypervisor' vboxmanage.items <item> details=True group_by=Name
```

Some items do not display well, or at all, unless `details` is set to `True`. By default, items are grouped by the `UUID` field, but not all items contain that field. In those cases, another field must be specified.

`salt.modules.vboxmanage.`**`list_nodes`**`()`

Return a list of registered VMs

CLI Example:

```
salt '*' vboxmanage.list_nodes
```

`salt.modules.vboxmanage.`**`list_nodes_full`**`()`

Return a list of registered VMs, with detailed information

CLI Example:

```
salt '*' vboxmanage.list_nodes_full
```

salt.modules.vboxmanage.**list_nodes_min**()

Return a list of registered VMs, with minimal information

CLI Example:

```
salt '*' vboxmanage.list_nodes_min
```

salt.modules.vboxmanage.**list_ostypes**()

List the available OS Types

CLI Example:

```
salt '*' vboxmanage.list_ostypes
```

salt.modules.vboxmanage.**register**(*filename*)

Register a VM

CLI Example:

```
salt '*' vboxmanage.register my_vm_filename
```

salt.modules.vboxmanage.**start**(*name*)

Start a VM

CLI Example:

```
salt '*' vboxmanage.start my_vm
```

salt.modules.vboxmanage.**stop**(*name*)

Stop a VM

CLI Example:

```
salt '*' vboxmanage.stop my_vm
```

salt.modules.vboxmanage.**unregister**(*name*, *delete=False*)

Unregister a VM

CLI Example:

```
salt '*' vboxmanage.unregister my_vm_filename
```

salt.modules.vboxmanage.**vboxcmd**()

Return the location of the VBoxManage command

CLI Example:

```
salt '*' vboxmanage.vboxcmd
```

21.9.448 salt.modules.victorops

Support for VictorOps

New in version 2015.8.0.

Requires an `api_key` in `/etc/salt/minion`:

```
victorops:
  api_key: '280d4699-a817-4719-ba6f-ca56e573e44f'
```

`salt.modules.victorops.`**`create_event`**(*message_type=None*, *routing_key='everybody'*, ***kwargs*)

Create an event in VictorOps. Designed for use in states.

The following parameters are required:

Parameters **`message_type`** – One of the following values: INFO, WARNING, ACKNOWLEDGEMENT, CRITICAL, RECOVERY.

The following parameters are optional:

Parameters

- **`routing_key`** – The key for where messages should be routed. By default, sent to 'everyone' route.
- **`entity_id`** – The name of alerting entity. If not provided, a random name will be assigned.
- **`timestamp`** – Timestamp of the alert in seconds since epoch. Defaults to the time the alert is received at VictorOps.

:param timestamp_fmt The date format for the timestamp parameter.

Parameters

- **`state_start_time`** – The time this entity entered its current state (seconds since epoch). Defaults to the time alert is received.
- **`state_start_time_fmt`** – The date format for the timestamp parameter.
- **`state_message`** – Any additional status information from the alert item.
- **`entity_is_host`** – Used within VictorOps to select the appropriate display format for the incident.
- **`entity_display_name`** – Used within VictorOps to display a human-readable name for the entity.
- **`ack_message`** – A user entered comment for the acknowledgment.
- **`ack_author`** – The user that acknowledged the incident.

Returns A dictionary with result, entity_id, and message if result was failure.

CLI Example:

```
salt myminion victorops.create_event message_type='CRITICAL' routing_key='everyone
↪'                 entity_id='hostname/diskspace'

salt myminion victorops.create_event message_type='ACKNOWLEDGEMENT' routing_key=
↪'everyone'                  entity_id='hostname/diskspace' ack_message=
↪'Acknowledged' ack_author='username'

salt myminion victorops.create_event message_type='RECOVERY' routing_key='everyone
↪'                 entity_id='hostname/diskspace'
```

The following parameters are required: message_type

21.9.449 salt.modules.virt

Work with virtual machines managed by libvirt

depends libvirt Python module

Connection

The connection to the virtualization host can be either setup in the minion configuration, pillar data or overridden for each individual call.

By default, the libvirt connection URL will be guessed: the first available libvirt hypervisor driver will be used. This can be overridden like this:

```
virt:
  connection:
    uri: lxc:///
```

If the connection requires an authentication like for ESXi, this can be defined in the minion pillar data like this:

```
virt:
  connection:
    uri: esx://10.1.1.101/?no_verify=1&auto_answer=1
    auth:
      username: user
      password: secret
```

Connecting with SSH protocol

Libvirt can connect to remote hosts using SSH using one of the `ssh`, `libssh` and `libssh2` transports. Note that `libssh2` is likely to fail as it doesn't read the `known_hosts` file. Libvirt may also have been built without `libssh` or `libssh2` support.

To use the SSH transport, on the minion setup an SSH agent with a key authorized on the remote libvirt machine.

Per call connection setup

New in version 2019.2.0.

All the calls requiring the libvirt connection configuration as mentioned above can override this configuration using `connection`, `username` and `password` parameters.

This means that the following will list the domains on the local LXC libvirt driver, whatever the `virt:connection` is.

```
salt 'hypervisor' virt.list_domains connection=lxc:///
```

The calls not using the libvirt connection setup are:

- `seed_non_shared_migrate`
- `virt_type`
- `is_*hyper`
- all migration functions
- libvirt ESX URI format
- libvirt URI format
- libvirt authentication configuration

`salt.modules.virt.`**`capabilities`**`(`*`**kwargs`*`)`
Return the hypervisor connection capabilities.

Parameters

- **`connection`** – libvirt connection URI, overriding defaults
- **`username`** – username to connect with, overriding defaults
- **`password`** – password to connect with, overriding defaults

New in version 2019.2.0.

CLI Example:

```
salt '*' virt.capabilities
```

`salt.modules.virt.`**`cpu_baseline`**(*full=False*, *migratable=False*, *out='libvirt'*, ***kwargs*)

Return the optimal 'custom' CPU baseline config for VM's on this minion

New in version 2016.3.0.

Parameters

- **`full`** – Return all CPU features rather than the ones on top of the closest CPU model
- **`migratable`** – Exclude CPU features that are unmigratable (libvirt 2.13+)
- **`out`** – 'libvirt' (default) for usable libvirt XML definition, 'salt' for nice dict
- **`connection`** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.
- **`username`** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **`password`** – password to connect with, overriding defaults

  New in version 2019.2.0.

CLI Example:

```
salt '*' virt.cpu_baseline
```

`salt.modules.virt.`**`create_xml_path`**(*path*, ***kwargs*)

Start a transient domain based on the XML-file path passed to the function

Parameters

- **`path`** – path to a file containing the libvirt XML definition of the domain
- **`connection`** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.
- **`username`** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **`password`** – password to connect with, overriding defaults

  New in version 2019.2.0.

CLI Example:

```
salt '*' virt.create_xml_path <path to XML file on the node>
```

`salt.modules.virt.`**`create_xml_str`**(*xml*, ***kwargs*)

Start a transient domain based on the XML passed to the function

Parameters

- **`xml`** – libvirt XML definition of the domain
- **`connection`** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.
- **`username`** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **`password`** – password to connect with, overriding defaults

New in version 2019.2.0.

CLI Example:

```
salt '*' virt.create_xml_str <XML in string format>
```

salt.modules.virt.**ctrl_alt_del**(*vm_*, ***kwargs*)

Sends CTRL+ALT+DEL to a VM

Parameters

- **vm** – domain name
- **connection** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.
- **username** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **password** – password to connect with, overriding defaults

  New in version 2019.2.0.

CLI Example:

```
salt '*' virt.ctrl_alt_del <domain>
```

salt.modules.virt.**define_vol_xml_path**(*path*, ***kwargs*)

Define a volume based on the XML-file path passed to the function

Parameters

- **path** – path to a file containing the libvirt XML definition of the volume
- **connection** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.
- **username** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **password** – password to connect with, overriding defaults

  New in version 2019.2.0.

CLI Example:

```
salt '*' virt.define_vol_xml_path <path to XML file on the node>
```

salt.modules.virt.**define_vol_xml_str**(*xml*, ***kwargs*)

Define a volume based on the XML passed to the function

Parameters

- **xml** – libvirt XML definition of the storage volume
- **connection** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.
- **username** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **password** – password to connect with, overriding defaults

  New in version 2019.2.0.

CLI Example:

```
salt '*' virt.define_vol_xml_str <XML in string format>
```

The storage pool where the disk image will be defined is `default` unless changed with a configuration like this:

```
virt:
    storagepool: mine
```

`salt.modules.virt.`**`define_xml_path`**(*path*, ***kwargs*)

Define a persistent domain based on the XML-file path passed to the function

Parameters

- **`path`** – path to a file containing the libvirt XML definition of the domain
- **`connection`** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.
- **`username`** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **`password`** – password to connect with, overriding defaults

  New in version 2019.2.0.

CLI Example:

```
salt '*' virt.define_xml_path <path to XML file on the node>
```

`salt.modules.virt.`**`define_xml_str`**(*xml*, ***kwargs*)

Define a persistent domain based on the XML passed to the function

Parameters

- **`xml`** – libvirt XML definition of the domain
- **`connection`** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.
- **`username`** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **`password`** – password to connect with, overriding defaults

  New in version 2019.2.0.

CLI Example:

```
salt '*' virt.define_xml_str <XML in string format>
```

`salt.modules.virt.`**`delete_snapshots`**(*name*, **names*, ***kwargs*)

Delete one or more snapshots of the given VM.

Parameters

- **`name`** – domain name
- **`names`** – names of the snapshots to remove
- **`connection`** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.
- **`username`** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **`password`** – password to connect with, overriding defaults

  New in version 2019.2.0.

New in version 2016.3.0.

CLI Example:

```
salt '*' virt.delete_snapshots <domain> all=True
salt '*' virt.delete_snapshots <domain> <snapshot>
salt '*' virt.delete_snapshots <domain> <snapshot1> <snapshot2> ...
```

`salt.modules.virt.`**`domain_capabilities`**`(`*emulator=None, arch=None, machine=None, domain=None, **kwargs*`)`

Return the domain capabilities given an emulator, architecture, machine or virtualization type.

New in version 2019.2.0.

Parameters

- **`emulator`** – return the capabilities for the given emulator binary
- **`arch`** – return the capabilities for the given CPU architecture
- **`machine`** – return the capabilities for the given emulated machine type
- **`domain`** – return the capabilities for the given virtualization type.
- **`connection`** – libvirt connection URI, overriding defaults
- **`username`** – username to connect with, overriding defaults
- **`password`** – password to connect with, overriding defaults

The list of the possible emulator, arch, machine and domain can be found in the host capabilities output.

If none of the parameters is provided the libvirt default domain capabilities will be returned.

CLI Example:

```
salt '*' virt.domain_capabilities arch='x86_64' domain='kvm'
```

`salt.modules.virt.`**`freecpu`**`(`***kwargs*`)`

Return an int representing the number of unallocated cpus on this hypervisor

Parameters

- **`connection`** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.
- **`username`** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **`password`** – password to connect with, overriding defaults

  New in version 2019.2.0.

CLI Example:

```
salt '*' virt.freecpu
```

`salt.modules.virt.`**`freemem`**`(`***kwargs*`)`

Return an int representing the amount of memory (in MB) that has not been given to virtual machines on this node

Parameters

- **`connection`** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.
- **`username`** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **`password`** – password to connect with, overriding defaults

  New in version 2019.2.0.

CLI Example:

```
salt '*' virt.freemem
```

`salt.modules.virt.`**`full_info`**`(`***kwargs*`)`

Return the node_info, vm_info and freemem

Parameters

- **`connection`** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.

- **username** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **password** – password to connect with, overriding defaults

  New in version 2019.2.0.

CLI Example:

```
salt '*' virt.full_info
```

`salt.modules.virt.`**`get_disks`**`(`*vm_*, ***kwargs*`)`

Return the disks of a named vm

Parameters

- **vm** – name of the domain
- **connection** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.
- **username** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **password** – password to connect with, overriding defaults

  New in version 2019.2.0.

CLI Example:

```
salt '*' virt.get_disks <domain>
```

`salt.modules.virt.`**`get_graphics`**`(`*vm_*, ***kwargs*`)`

Returns the information on vnc for a given vm

Parameters

- **vm** – name of the domain
- **connection** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.
- **username** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **password** – password to connect with, overriding defaults

  New in version 2019.2.0.

CLI Example:

```
salt '*' virt.get_graphics <domain>
```

`salt.modules.virt.`**`get_hypervisor`**`()`

Returns the name of the hypervisor running on this node or `None`.

Detected hypervisors:

- kvm
- xen

CLI Example:

```
salt '*' virt.get_hypervisor
```

New in version 2019.2.0: the function and the `kvm` and `xen` hypervisors support

`salt.modules.virt.`**`get_macs`**`(`*vm_*, ***kwargs*`)`

Return a list off MAC addresses from the named vm

Parameters

- **vm** – name of the domain

- **`connection`** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.
- **`username`** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **`password`** – password to connect with, overriding defaults

  New in version 2019.2.0.

CLI Example:

```
salt '*' virt.get_macs <domain>
```

`salt.modules.virt.`**`get_nics`**`(`*vm_*, ***kwargs*`)`

Return info about the network interfaces of a named vm

Parameters

- **`vm`** – name of the domain
- **`connection`** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.
- **`username`** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **`password`** – password to connect with, overriding defaults

  New in version 2019.2.0.

CLI Example:

```
salt '*' virt.get_nics <domain>
```

`salt.modules.virt.`**`get_profiles`**`(`*hypervisor=None*, ***kwargs*`)`

Return the virt profiles for hypervisor.

Currently there are profiles for:

- nic
- disk

Parameters

- **`hypervisor`** – override the default machine type.
- **`connection`** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.
- **`username`** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **`password`** – password to connect with, overriding defaults

  New in version 2019.2.0.

CLI Example:

```
salt '*' virt.get_profiles
salt '*' virt.get_profiles hypervisor=esxi
```

`salt.modules.virt.`**`get_xml`**`(`*vm_*, ***kwargs*`)`

Returns the XML for a given vm

Parameters

- **`vm`** – domain name
- **`connection`** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.

- **username** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **password** – password to connect with, overriding defaults

  New in version 2019.2.0.

CLI Example:

```
salt '*' virt.get_xml <domain>
```

`salt.modules.virt.init`(*name, cpu, mem, image=None, nic='default', interfaces=None, hypervisor=None, start=True, disk='default', disks=None, saltenv='base', seed=True, install=True, pub_key=None, priv_key=None, seed_cmd='seed.apply', enable_vnc=False, enable_qcow=False, graphics=None, os_type=None, arch=None, boot=None, **kwargs*)

Initialize a new vm

Parameters

- **name** – name of the virtual machine to create
- **cpu** – Number of virtual CPUs to assign to the virtual machine
- **mem** – Amount of memory to allocate to the virtual machine in MiB.
- **image** – Path to a disk image to use as the first disk (Default: `None`). Deprecated in favor of the `disks` parameter. To set (or change) the image of a disk, add the following to the disks definitions:

```
{
    'name': 'name_of_disk_to_change',
    'image': '/path/to/the/image'
}
```

- **nic** – NIC profile to use (Default: `'default'`). The profile interfaces can be customized / extended with the interfaces parameter. If set to `None`, no profile will be used.
- **interfaces** – List of dictionaries providing details on the network interfaces to create. These data are merged with the ones from the nic profile. The structure of each dictionary is documented in *Network Interfaces Definitions*.

  New in version 2019.2.0.
- **hypervisor** – the virtual machine type. By default the value will be computed according to the virtual host capabilities.
- **start** – `True` to start the virtual machine after having defined it (Default: `True`)
- **disk** – Disk profile to use (Default: `'default'`). If set to `None`, no profile will be used.
- **disks** – List of dictionaries providing details on the disk devices to create. These data are merged with the ones from the disk profile. The structure of each dictionary is documented in *Disks Definitions*.

  New in version 2019.2.0.
- **saltenv** – Fileserver environment (Default: `'base'`). See *`cp module for more details`*
- **seed** – `True` to seed the disk image. Only used when the `image` parameter is provided. (Default: `True`)
- **install** – install salt minion if absent (Default: `True`)
- **pub_key** – public key to seed with (Default: `None`)
- **priv_key** – public key to seed with (Default: `None`)
- **seed_cmd** – Salt command to execute to seed the image. (Default: `'seed.apply'`)
- **enable_vnc** – `True` to setup a vnc display for the VM (Default: `False`)

Deprecated in favor of the `graphics` parameter. Could be replaced with the following:

```
graphics={'type': 'vnc'}
```

Deprecated since version 2019.2.0.

- **graphics** – Dictionary providing details on the graphics device to create. (Default: `None`) See *Graphics Definition* for more details on the possible values.

  New in version 2019.2.0.
- **os_type** – type of virtualization as found in the `//os/type` element of the libvirt definition. The default value is taken from the host capabilities, with a preference for `hvm`.

  New in version 2019.2.0.
- **arch** – architecture of the virtual machine. The default value is taken from the host capabilities, but `x86_64` is prefed over `i686`.

  New in version 2019.2.0.
- **enable_qcow** – `True` to create a QCOW2 overlay image, rather than copying the image (Default: `False`).

  Deprecated in favor of `disks` parameter. Add the following to the disks definitions to create an overlay image of a template disk image with an image set:

  ```
  {
      'name': 'name_of_disk_to_change',
      'overlay_image': True
  }
  ```

  Deprecated since version 2019.2.0.
- **pool** – Path of the folder where the image files are located for vmware/esx hypervisors.

  Deprecated in favor of `disks` parameter. Add the following to the disks definitions to set the vmware datastore of a disk image:

  ```
  {
      'name': 'name_of_disk_to_change',
      'pool': 'mydatastore'
  }
  ```

  Deprecated since version Flurorine.
- **dmac** – Default MAC address to use for the network interfaces. By default MAC addresses are automatically generated.

  Deprecated in favor of `interfaces` parameter. Add the following to the interfaces definitions to force the mac address of a NIC:

  ```
  {
      'name': 'name_of_nic_to_change',
      'mac': 'MY:MA:CC:ADD:RE:SS'
  }
  ```

  Deprecated since version 2019.2.0.
- **config** – minion configuration to use when seeding. See *`seed module for more details`*
- **boot_dev** – String of space-separated devices to boot from (Default: `'hd'`)
- **serial_type** – Serial device type. One of `'pty'`, `'tcp'` (Default: `None`)

- **`telnet_port`** – Telnet port to use for serial device of type `tcp`.
- **`console`** – `True` to add a console device along with serial one (Default: `True`)
- **`connection`** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.
- **`username`** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **`password`** – password to connect with, overriding defaults

  New in version 2019.2.0.
- **`boot`** – Specifies kernel for the virtual machine, as well as boot parameters for the virtual machine. This is an optionl parameter, and all of the keys are optional within the dictionary. If a remote path is provided to kernel or initrd, salt will handle the downloading of the specified remote fild, and will modify the XML accordingly.

  ```
  {
      'kernel': '/root/f8-i386-vmlinuz',
      'initrd': '/root/f8-i386-initrd',
      'cmdline': 'console=ttyS0 ks=http://example.com/f8-i386/os/
  ↪'
  }
  ```

  New in version 3000.

Network Interfaces Definitions

Network interfaces dictionaries can contain the following properties:

name Name of the network interface. This is only used as a key to merge with the profile data

type Network type. One of `'bridge'`, `'network'`

source The network source, typically the bridge or network name

mac The desired mac address, computed if `None` (Default: `None`).

model The network card model (Default: depends on the hypervisor)

Disks Definitions

Disk dictionaries can contain the following properties:

name Name of the disk. This is mostly used in the name of the disk image and as a key to merge with the profile data.

format Format of the disk image, like `'qcow2'`, `'raw'`, `'vmdk'`. (Default: depends on the hypervisor)

size Disk size in MiB

pool Path to the folder or name of the pool where disks should be created. (Default: depends on hypervisor)

model One of the disk busses allowed by libvirt (Default: depends on hypervisor)

See the libvirt disk element documentation for the allowed bus types.

image Path to the image to use for the disk. If no image is provided, an empty disk will be created (Default: `None`)

overlay_image `True` to create a QCOW2 disk image with `image` as backing file. If `False` the file pointed to by the `image` property will simply be copied. (Default: `False`)

source_file Absolute path to the disk image to use. Not to be confused with `image` parameter. This parameter is useful to use disk images that are created outside of this module. Can also be `None` for devices that have no associated image like cdroms.

device Type of device of the disk. Can be one of 'disk', 'cdrom', 'floppy' or 'lun'. (Default: `'disk'`)

Graphics Definition

The graphics dictionnary can have the following properties:

type Graphics type. The possible values are `none`, `'spice'`, `'vnc'` and other values allowed as a libvirt graphics type (Default: `None`)

See the libvirt graphics element documentation for more details on the possible types.

port Port to export the graphics on for `vnc`, `spice` and `rdp` types.

tls_port Port to export the graphics over a secured connection for `spice` type.

listen Dictionary defining on what address to listen on for `vnc`, `spice` and `rdp`. It has a `type` property with `address` and `None` as possible values, and an `address` property holding the IP or hostname to listen on.

By default, not setting the `listen` part of the dictionary will default to listen on all addresses.

CLI Example

```
salt 'hypervisor' virt.init vm_name 4 512 salt://path/to/image.raw
salt 'hypervisor' virt.init vm_name 4 512 /var/lib/libvirt/images/img.raw
salt 'hypervisor' virt.init vm_name 4 512 nic=profile disk=profile
```

The disk images will be created in an image folder within the directory defined by the `virt:images` option. Its default value is `/srv/salt-images/` but this can changed with such a configuration:

```
virt:
    images: /data/my/vm/images/
```

`salt.modules.virt.`**`is_hyper`**`()`

Returns a bool whether or not this node is a hypervisor of any kind

CLI Example:

```
salt '*' virt.is_hyper
```

`salt.modules.virt.`**`is_kvm_hyper`**`()`

Returns a bool whether or not this node is a KVM hypervisor

CLI Example:

```
salt '*' virt.is_kvm_hyper
```

Deprecated since version 2019.2.0.

`salt.modules.virt.`**`is_xen_hyper`**`()`

Returns a bool whether or not this node is a XEN hypervisor

CLI Example:

```
salt '*' virt.is_xen_hyper
```

Deprecated since version 2019.2.0.

`salt.modules.virt.`**`list_active_vms`**`(`***kwargs*`)`

Return a list of names for active virtual machine on the minion

Parameters

- **`connection`** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.

- **`username`** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **`password`** – password to connect with, overriding defaults

  New in version 2019.2.0.

CLI Example:

```
salt '*' virt.list_active_vms
```

`salt.modules.virt.`**`list_domains`**`(`***kwargs*`)`

Return a list of available domains.

Parameters

- **`connection`** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.
- **`username`** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **`password`** – password to connect with, overriding defaults

  New in version 2019.2.0.

CLI Example:

```
salt '*' virt.list_domains
```

`salt.modules.virt.`**`list_inactive_vms`**`(`***kwargs*`)`

Return a list of names for inactive virtual machine on the minion

Parameters

- **`connection`** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.
- **`username`** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **`password`** – password to connect with, overriding defaults

  New in version 2019.2.0.

CLI Example:

```
salt '*' virt.list_inactive_vms
```

`salt.modules.virt.`**`list_networks`**`(`***kwargs*`)`

List all virtual networks.

Parameters

- **`connection`** – libvirt connection URI, overriding defaults
- **`username`** – username to connect with, overriding defaults
- **`password`** – password to connect with, overriding defaults

New in version 2019.2.0.

CLI Example:

```
salt '*' virt.list_networks
```

`salt.modules.virt.`**`list_pools`**`(`***kwargs*`)`

List all storage pools.

Parameters

- **`connection`** – libvirt connection URI, overriding defaults
- **`username`** – username to connect with, overriding defaults
- **`password`** – password to connect with, overriding defaults

New in version 2019.2.0.

CLI Example:

```
salt '*' virt.list_pools
```

`salt.modules.virt.`**`list_snapshots`**`(`*domain=None, **kwargs*`)`

List available snapshots for certain vm or for all.

Parameters

- **`domain`** – domain name
- **`connection`** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.
- **`username`** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **`password`** – password to connect with, overriding defaults

  New in version 2019.2.0.

New in version 2016.3.0.

CLI Example:

```
salt '*' virt.list_snapshots
salt '*' virt.list_snapshots <domain>
```

`salt.modules.virt.`**`migrate`**`(`*vm_, target, ssh=False*`)`

Shared storage migration

Parameters

- **`vm`** – domain name
- **`target`** – target libvirt host name
- **`ssh`** – True to connect over ssh

CLI Example:

```
salt '*' virt.migrate <domain> <target hypervisor>
```

A tunnel data migration can be performed by setting this in the configuration:

```
virt:
    tunnel: True
```

For more details on tunnelled data migrations, report to https://libvirt.org/migration.html#transporttunnel

`salt.modules.virt.`**`migrate_non_shared`**`(`*vm_, target, ssh=False*`)`

Attempt to execute non-shared storage ”all” migration

Parameters

- **`vm`** – domain name
- **`target`** – target libvirt host name
- **`ssh`** – True to connect over ssh

CLI Example:

```
salt '*' virt.migrate_non_shared <vm name> <target hypervisor>
```

A tunnel data migration can be performed by setting this in the configuration:

```
virt:
    tunnel: True
```

For more details on tunnelled data migrations, report to https://libvirt.org/migration.html#transporttunnel `salt.modules.virt.`**`migrate_non_shared_inc`**(*vm_, target, ssh=False*)

Attempt to execute non-shared storage "all" migration

Parameters

- **`vm`** – domain name
- **`target`** – target libvirt host name
- **`ssh`** – True to connect over ssh

CLI Example:

```
salt '*' virt.migrate_non_shared_inc <vm name> <target hypervisor>
```

A tunnel data migration can be performed by setting this in the configuration:

```
virt:
    tunnel: True
```

For more details on tunnelled data migrations, report to https://libvirt.org/migration.html#transporttunnel

`salt.modules.virt.`**`network_define`**(*name, bridge, forward, ipv4_config=None, ipv6_config=None, **kwargs*)

Create libvirt network.

Parameters

- **`name`** – Network name
- **`bridge`** – Bridge name
- **`forward`** – Forward mode(bridge, router, nat)
- **`vport`** – Virtualport type
- **`tag`** – Vlan tag
- **`autostart`** – Network autostart (default True)
- **`start`** – Network start (default True)
- **`ipv4_config`** (`dict or None`) – IP v4 configuration Dictionary describing the IP v4 setup like IP range and a possible DHCP configuration. The structure is documented in *net-define-ip*.

  New in version 3000.
- **`ipv6_config`** (`dict or None`) – IP v6 configuration Dictionary describing the IP v6 setup like IP range and a possible DHCP configuration. The structure is documented in *net-define-ip*.

  New in version 3000.
- **`connection`** – libvirt connection URI, overriding defaults
- **`username`** – username to connect with, overriding defaults
- **`password`** – password to connect with, overriding defaults

** IP configuration definition

Both the IPv4 and IPv6 configuration dictionaries can contain the following properties:

cidr CIDR notation for the network. For example '192.168.124.0/24'

dhcp_ranges A list of dictionary with `'start'` and `'end'` properties.

CLI Example:

```
salt '*' virt.network_define network main bridge openvswitch
```

New in version 2019.2.0.

`salt.modules.virt.`**`network_get_xml`**(*name, **kwargs*)

Return the XML definition of a virtual network

Parameters

- **`name`** – libvirt network name
- **`connection`** – libvirt connection URI, overriding defaults
- **`username`** – username to connect with, overriding defaults

- **password** – password to connect with, overriding defaults

New in version 3000.

CLI Example:

```
salt '*' virt.network_get_xml default
```

salt.modules.virt.**network_info**(*name=None*, ***kwargs*)

Return informations on a virtual network provided its name.

Parameters

- **name** – virtual network name
- **connection** – libvirt connection URI, overriding defaults
- **username** – username to connect with, overriding defaults
- **password** – password to connect with, overriding defaults

If no name is provided, return the infos for all defined virtual networks.

New in version 2019.2.0.

CLI Example:

```
salt '*' virt.network_info default
```

salt.modules.virt.**network_set_autostart**(*name*, *state='on'*, ***kwargs*)

Set the autostart flag on a virtual network so that the network will start with the host system on reboot.

Parameters

- **name** – virtual network name
- **state** – 'on' to auto start the network, anything else to mark the virtual network not to be started when the host boots
- **connection** – libvirt connection URI, overriding defaults
- **username** – username to connect with, overriding defaults
- **password** – password to connect with, overriding defaults

New in version 2019.2.0.

CLI Example:

```
salt "*" virt.network_set_autostart <pool> <on | off>
```

salt.modules.virt.**network_start**(*name*, ***kwargs*)

Start a defined virtual network.

Parameters

- **name** – virtual network name
- **connection** – libvirt connection URI, overriding defaults
- **username** – username to connect with, overriding defaults
- **password** – password to connect with, overriding defaults

New in version 2019.2.0.

CLI Example:

```
salt '*' virt.network_start default
```

salt.modules.virt.**network_stop**(*name*, ***kwargs*)

Stop a defined virtual network.

Parameters

- **name** – virtual network name
- **connection** – libvirt connection URI, overriding defaults
- **username** – username to connect with, overriding defaults
- **password** – password to connect with, overriding defaults

New in version 2019.2.0.

CLI Example:

```
salt '*' virt.network_stop default
```

salt.modules.virt.**network_undefine**(*name*, ***kwargs*)

Remove a defined virtual network. This does not stop the virtual network.

Parameters

- **name** – virtual network name
- **connection** – libvirt connection URI, overriding defaults
- **username** – username to connect with, overriding defaults
- **password** – password to connect with, overriding defaults

New in version 2019.2.0.

CLI Example:

```
salt '*' virt.network_undefine default
```

salt.modules.virt.**node_info**(***kwargs*)

Return a dict with information about this node

Parameters

- **connection** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.
- **username** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **password** – password to connect with, overriding defaults

  New in version 2019.2.0.

CLI Example:

```
salt '*' virt.node_info
```

salt.modules.virt.**pause**(*vm_*, ***kwargs*)

Pause the named vm

Parameters

- **vm** – domain name
- **connection** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.
- **username** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **password** – password to connect with, overriding defaults

  New in version 2019.2.0.

CLI Example:

```
salt '*' virt.pause <domain>
```

salt.modules.virt.**pool_build**(*name*, ***kwargs*)

Build a defined libvirt storage pool.

Parameters

- **name** – libvirt storage pool name
- **connection** – libvirt connection URI, overriding defaults
- **username** – username to connect with, overriding defaults
- **password** – password to connect with, overriding defaults

New in version 2019.2.0.

CLI Example:

```
salt '*' virt.pool_build default
```

`salt.modules.virt.`**`pool_capabilities`**`(`*`**kwargs`*`)`

Return the hypervisor connection storage pool capabilities.

The returned data are either directly extracted from libvirt or computed. In the latter case some pool types could be listed as supported while they are not. To distinguish between the two cases, check the value of the `computed` property.

Parameters

- **`connection`** – libvirt connection URI, overriding defaults
- **`username`** – username to connect with, overriding defaults
- **`password`** – password to connect with, overriding defaults

New in version 3000.

CLI Example:

```
salt '*' virt.pool_capabilities
```

`salt.modules.virt.`**`pool_define`**`(`*name, ptype, target=None, permissions=None, source_devices=None, source_dir=None, source_initiator=None, source_adapter=None, source_hosts=None, source_auth=None, source_name=None, source_format=None, transient=False, start=True, **kwargs*`)`

Create libvirt pool.

Parameters

- **`name`** – Pool name
- **`ptype`** – Pool type. See libvirt documentation for the possible values.
- **`target`** – Pool full path target
- **`permissions`** – Permissions to set on the target folder. This is mostly used for filesystem-based pool types. See *Permissions definition* for more details on this structure.
- **`source_devices`** – List of source devices for pools backed by physical devices. (Default: `None`)

  Each item in the list is a dictionary with `path` and optionally `part_separator` keys. The path is the qualified name for iSCSI devices.

  Report to this libvirt page for more informations on the use of `part_separator`
- **`source_dir`** – Path to the source directory for pools of type `dir`, `netfs` or `gluster`. (Default: `None`)
- **`source_initiator`** – Initiator IQN for libiscsi-direct pool types. (Default: `None`)

  New in version 3000.
- **`source_adapter`** – SCSI source definition. The value is a dictionary with `type`, `name`, `parent`, `managed`, `parent_wwnn`, `parent_wwpn`, `parent_fabric_wwn`, `wwnn`, `wwpn` and `parent_address` keys.

  The `parent_address` value is a dictionary with `unique_id` and `address` keys. The address represents a PCI address and is itself a dictionary with `domain`, `bus`, `slot` and `function` properties. Report to this libvirt page for the meaning and possible values of these properties.
- **`source_hosts`** – List of source for pools backed by storage from remote servers. Each item is the hostname optionally followed by the port separated by a colon. (Default: `None`)

- **`source_auth`** – Source authentication details. (Default: `None`)

  The value is a dictionary with `type`, `username` and `secret` keys. The type can be one of `ceph` for Ceph RBD or `chap` for iSCSI sources.

  The `secret` value links to a libvirt secret object. It is a dictionary with `type` and `value` keys. The type value can be either `uuid` or `usage`.

  Examples:

```
source_auth={
    'type': 'ceph',
    'username': 'admin',
    'secret': {
        'type': 'uuid',
        'value': '2ec115d7-3a88-3ceb-bc12-0ac909a6fd87'
    }
}
```

```
source_auth={
    'type': 'chap',
    'username': 'myname',
    'secret': {
        'type': 'usage',
        'value': 'mycluster_myname'
    }
}
```

  Since 3000, instead the source authentication can only contain `username` and `password` properties. In this case the libvirt secret will be defined and used. For Ceph authentications a base64 encoded key is expected.
- **`source_name`** – Identifier of name-based sources.
- **`source_format`** – String representing the source format. The possible values are depending on the source type. See libvirt documentation for the possible values.
- **`start`** – Pool start (default True)
- **`transient`** – When `True`, the pool will be automatically undefined after being stopped. Note that a transient pool will force `start` to `True`. (Default: `False`)
- **`connection`** – libvirt connection URI, overriding defaults
- **`username`** – username to connect with, overriding defaults
- **`password`** – password to connect with, overriding defaults

Permissions definition

The permissions are described by a dictionary containing the following keys:
mode The octal representation of the permissions. (Default: *0711*)
owner the numeric user ID of the owner. (Default: from the parent folder)
group the numeric ID of the group. (Default: from the parent folder)
label the SELinux label. (Default: *None*)

CLI Example:

Local folder pool:

```
salt '*' virt.pool_define somepool dir target=/srv/mypool
↪         permissions="{'mode': '0744' 'ower': 107, 'group': 107 }"
```

CIFS backed pool:

```
salt '*' virt.pool_define myshare netfs source_format=cifs
↪        source_dir=samba_share source_hosts="['example.com']" target=/mnt/
↪cifs
```

New in version 2019.2.0.

`salt.modules.virt.`**`pool_delete`**`(`*name, **kwargs*`)`

Delete the resources of a defined libvirt storage pool.

Parameters

- **`name`** – libvirt storage pool name
- **`connection`** – libvirt connection URI, overriding defaults
- **`username`** – username to connect with, overriding defaults
- **`password`** – password to connect with, overriding defaults

New in version 2019.2.0.

CLI Example:

```
salt '*' virt.pool_delete default
```

`salt.modules.virt.`**`pool_get_xml`**`(`*name, **kwargs*`)`

Return the XML definition of a virtual storage pool

Parameters

- **`name`** – libvirt storage pool name
- **`connection`** – libvirt connection URI, overriding defaults
- **`username`** – username to connect with, overriding defaults
- **`password`** – password to connect with, overriding defaults

New in version 3000.

CLI Example:

```
salt '*' virt.pool_get_xml default
```

`salt.modules.virt.`**`pool_info`**`(`*name=None, **kwargs*`)`

Return informations on a storage pool provided its name.

Parameters

- **`name`** – libvirt storage pool name
- **`connection`** – libvirt connection URI, overriding defaults
- **`username`** – username to connect with, overriding defaults
- **`password`** – password to connect with, overriding defaults

If no name is provided, return the infos for all defined storage pools.

New in version 2019.2.0.

CLI Example:

```
salt '*' virt.pool_info default
```

`salt.modules.virt.`**`pool_list_volumes`**`(`*name, **kwargs*`)`

List the volumes contained in a defined libvirt storage pool.

Parameters

- **`name`** – libvirt storage pool name
- **`connection`** – libvirt connection URI, overriding defaults
- **`username`** – username to connect with, overriding defaults
- **`password`** – password to connect with, overriding defaults

New in version 2019.2.0.

CLI Example:

```
salt "*" virt.pool_list_volumes <pool>
```

`salt.modules.virt.`**`pool_refresh`**`(`*name*, ***kwargs*`)`

Refresh a defined libvirt storage pool.

Parameters

- **`name`** – libvirt storage pool name
- **`connection`** – libvirt connection URI, overriding defaults
- **`username`** – username to connect with, overriding defaults
- **`password`** – password to connect with, overriding defaults

New in version 2019.2.0.

CLI Example:

```
salt '*' virt.pool_refresh default
```

`salt.modules.virt.`**`pool_set_autostart`**`(`*name*, *state='on'*, ***kwargs*`)`

Set the autostart flag on a libvirt storage pool so that the storage pool will start with the host system on reboot.

Parameters

- **`name`** – libvirt storage pool name
- **`state`** – 'on' to auto start the pool, anything else to mark the pool not to be started when the host boots
- **`connection`** – libvirt connection URI, overriding defaults
- **`username`** – username to connect with, overriding defaults
- **`password`** – password to connect with, overriding defaults

New in version 2019.2.0.

CLI Example:

```
salt "*" virt.pool_set_autostart <pool> <on | off>
```

`salt.modules.virt.`**`pool_start`**`(`*name*, ***kwargs*`)`

Start a defined libvirt storage pool.

Parameters

- **`name`** – libvirt storage pool name
- **`connection`** – libvirt connection URI, overriding defaults
- **`username`** – username to connect with, overriding defaults
- **`password`** – password to connect with, overriding defaults

New in version 2019.2.0.

CLI Example:

```
salt '*' virt.pool_start default
```

`salt.modules.virt.`**`pool_stop`**`(`*name*, ***kwargs*`)`

Stop a defined libvirt storage pool.

Parameters

- **`name`** – libvirt storage pool name
- **`connection`** – libvirt connection URI, overriding defaults
- **`username`** – username to connect with, overriding defaults
- **`password`** – password to connect with, overriding defaults

New in version 2019.2.0.

CLI Example:

```
salt '*' virt.pool_stop default
```

salt.modules.virt.**pool_undefine**(*name*, ***kwargs*)

Remove a defined libvirt storage pool. The pool needs to be stopped before calling.

Parameters

- **name** – libvirt storage pool name
- **connection** – libvirt connection URI, overriding defaults
- **username** – username to connect with, overriding defaults
- **password** – password to connect with, overriding defaults

New in version 2019.2.0.

CLI Example:

```
salt '*' virt.pool_undefine default
```

salt.modules.virt.**pool_update**(*name*, *ptype*, *target=None*, *permissions=None*, *source_devices=None*, *source_dir=None*, *source_initiator=None*, *source_adapter=None*, *source_hosts=None*, *source_auth=None*, *source_name=None*, *source_format=None*, *test=False*, ***kwargs*)

Update a libvirt storage pool if needed. If called with test=True, this is also reporting whether an update would be performed.

Parameters

- **name** – Pool name
- **ptype** – Pool type. See libvirt documentation for the possible values.
- **target** – Pool full path target
- **permissions** – Permissions to set on the target folder. This is mostly used for filesystem-based pool types. See *Permissions definition* for more details on this structure.
- **source_devices** – List of source devices for pools backed by physical devices. (Default: `None`)

  Each item in the list is a dictionary with `path` and optionally `part_separator` keys. The path is the qualified name for iSCSI devices.

  Report to this libvirt page for more informations on the use of `part_separator`
- **source_dir** – Path to the source directory for pools of type `dir`, `netfs` or `gluster`. (Default: `None`)
- **source_initiator** – Initiator IQN for libiscsi-direct pool types. (Default: `None`)

  New in version 3000.
- **source_adapter** – SCSI source definition. The value is a dictionary with `type`, `name`, `parent`, `managed`, `parent_wwnn`, `parent_wwpn`, `parent_fabric_wwn`, `wwnn`, `wwpn` and `parent_address` keys.

  The `parent_address` value is a dictionary with `unique_id` and `address` keys. The address represents a PCI address and is itself a dictionary with `domain`, `bus`, `slot` and `function` properties. Report to this libvirt page for the meaning and possible values of these properties.
- **source_hosts** – List of source for pools backed by storage from remote servers. Each item is the hostname optionally followed by the port separated by a colon. (Default: `None`)
- **source_auth** – Source authentication details. (Default: `None`)

  The value is a dictionary with `type`, `username` and `secret` keys. The type can be one of `ceph` for Ceph RBD or `chap` for iSCSI sources.

  The `secret` value links to a libvirt secret object. It is a dictionary with `type` and `value` keys. The type value can be either `uuid` or `usage`.

Examples:

```
source_auth={
    'type': 'ceph',
    'username': 'admin',
    'secret': {
        'type': 'uuid',
        'uuid': '2ec115d7-3a88-3ceb-bc12-0ac909a6fd87'
    }
}
```

```
source_auth={
    'type': 'chap',
    'username': 'myname',
    'secret': {
        'type': 'usage',
        'uuid': 'mycluster_myname'
    }
}
```

Since 3000, instead the source authentication can only contain `username` and `password` properties. In this case the libvirt secret will be defined and used. For Ceph authentications a base64 encoded key is expected.

- **`source_name`** – Identifier of name-based sources.
- **`source_format`** – String representing the source format. The possible values are depending on the source type. See libvirt documentation for the possible values.
- **`test`** – run in dry-run mode if set to True
- **`connection`** – libvirt connection URI, overriding defaults
- **`username`** – username to connect with, overriding defaults
- **`password`** – password to connect with, overriding defaults

Example:

Local folder pool:

```
salt '*' virt.pool_update somepool dir target=/srv/mypool
↪         permissions="{'mode': '0744' 'ower': 107, 'group': 107 }"
```

CIFS backed pool:

```
salt '*' virt.pool_update myshare netfs source_format=cifs
↪          source_dir=samba_share source_hosts="['example.com']" target=/mnt/
↪cifs
```

New in version 3000.

`salt.modules.virt.`**`purge`**(*vm_*, *dirs=False*, *removables=None*, ***kwargs*)

Recursively destroy and delete a persistent virtual machine, pass True for dir's to also delete the directories containing the virtual machine disk images - USE WITH EXTREME CAUTION!

Pass removables=False to avoid deleting cdrom and floppy images. To avoid disruption, the default but dangerous value is True. This will be changed to the safer False default value in Sodium.

Parameters

- **`vm`** – domain name
- **`dirs`** – pass True to remove containing directories

- **removables** – pass True to remove removable devices

  New in version 2019.2.0.
- **connection** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.
- **username** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **password** – password to connect with, overriding defaults

  New in version 2019.2.0.

CLI Example:

```
salt '*' virt.purge <domain> removables=False
```

`salt.modules.virt.`**`reboot`**(*name*, ***kwargs*)

Reboot a domain via ACPI request

Parameters

- **vm** – domain name
- **connection** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.
- **username** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **password** – password to connect with, overriding defaults

  New in version 2019.2.0.

CLI Example:

```
salt '*' virt.reboot <domain>
```

`salt.modules.virt.`**`reset`**(*vm_*, ***kwargs*)

Reset a VM by emulating the reset button on a physical machine

Parameters

- **vm** – domain name
- **connection** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.
- **username** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **password** – password to connect with, overriding defaults

  New in version 2019.2.0.

CLI Example:

```
salt '*' virt.reset <domain>
```

`salt.modules.virt.`**`resume`**(*vm_*, ***kwargs*)

Resume the named vm

Parameters

- **vm** – domain name
- **connection** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.
- **username** – username to connect with, overriding defaults

  New in version 2019.2.0.

- **password** – password to connect with, overriding defaults

  New in version 2019.2.0.

CLI Example:

```
salt '*' virt.resume <domain>
```

`salt.modules.virt.`**`revert_snapshot`**`(`*name*, *vm_snapshot=None*, *cleanup=False*, ***kwargs*`)`

Revert snapshot to the previous from current (if available) or to the specific.

Parameters

- **name** – domain name
- **vm_snapshot** – name of the snapshot to revert
- **cleanup** – Remove all newer than reverted snapshots. Values: True or False (default False).
- **connection** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.
- **username** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **password** – password to connect with, overriding defaults

  New in version 2019.2.0.

New in version 2016.3.0.

CLI Example:

```
salt '*' virt.revert <domain>
salt '*' virt.revert <domain> <snapshot>
```

`salt.modules.virt.`**`seed_non_shared_migrate`**`(`*disks*, *force=False*`)`

Non shared migration requires that the disks be present on the migration destination, pass the disks information via this function, to the migration destination before executing the migration.

Parameters

- **disks** – the list of disk data as provided by virt.get_disks
- **force** – skip checking the compatibility of source and target disk images if True. (default: False)

CLI Example:

```
salt '*' virt.seed_non_shared_migrate <disks>
```

`salt.modules.virt.`**`set_autostart`**`(`*vm_*, *state='on'*, ***kwargs*`)`

Set the autostart flag on a VM so that the VM will start with the host system on reboot.

Parameters

- **vm** – domain name
- **state** – 'on' to auto start the pool, anything else to mark the pool not to be started when the host boots
- **connection** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.
- **username** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **password** – password to connect with, overriding defaults

  New in version 2019.2.0.

CLI Example:

```
salt "*" virt.set_autostart <domain> <on | off>
```

`salt.modules.virt.`**`setmem`**`(`*vm_, memory, config=False, **kwargs*`)`

Changes the amount of memory allocated to VM. The VM must be shutdown for this to work.

Parameters

- **`vm`** – name of the domain
- **`memory`** – memory amount to set in MB
- **`config`** – if True then libvirt will be asked to modify the config as well
- **`connection`** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.
- **`username`** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **`password`** – password to connect with, overriding defaults

  New in version 2019.2.0.

CLI Example:

```
salt '*' virt.setmem <domain> <size>
salt '*' virt.setmem my_domain 768
```

`salt.modules.virt.`**`setvcpus`**`(`*vm_, vcpus, config=False, **kwargs*`)`

Changes the amount of vcpus allocated to VM. The VM must be shutdown for this to work.

If config is True then we ask libvirt to modify the config as well

Parameters

- **`vm`** – name of the domain
- **`vcpus`** – integer representing the number of CPUs to be assigned
- **`config`** – if True then libvirt will be asked to modify the config as well
- **`connection`** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.
- **`username`** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **`password`** – password to connect with, overriding defaults

  New in version 2019.2.0.

CLI Example:

```
salt '*' virt.setvcpus <domain> <amount>
salt '*' virt.setvcpus my_domain 4
```

`salt.modules.virt.`**`shutdown`**`(`*vm_, **kwargs*`)`

Send a soft shutdown signal to the named vm

Parameters

- **`vm`** – domain name
- **`connection`** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.
- **`username`** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **`password`** – password to connect with, overriding defaults

  New in version 2019.2.0.

CLI Example:

```
salt '*' virt.shutdown <domain>
```

`salt.modules.virt.`**`snapshot`**`(`*domain*, *name=None*, *suffix=None*, ***kwargs*`)`

Create a snapshot of a VM.

Parameters

- **`domain`** – domain name
- **`name`** – Name of the snapshot. If the name is omitted, then will be used original domain name with ISO 8601 time as a suffix.
- **`suffix`** – Add suffix for the new name. Useful in states, where such snapshots can be distinguished from manually created.
- **`connection`** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.
- **`username`** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **`password`** – password to connect with, overriding defaults

  New in version 2019.2.0.

New in version 2016.3.0.

CLI Example:

```
salt '*' virt.snapshot <domain>
```

`salt.modules.virt.`**`start`**`(`*name*, ***kwargs*`)`

Start a defined domain

Parameters

- **`vm`** – domain name
- **`connection`** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.
- **`username`** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **`password`** – password to connect with, overriding defaults

  New in version 2019.2.0.

CLI Example:

```
salt '*' virt.start <domain>
```

`salt.modules.virt.`**`stop`**`(`*name*, ***kwargs*`)`

Hard power down the virtual machine, this is equivalent to pulling the power.

Parameters

- **`vm`** – domain name
- **`connection`** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.
- **`username`** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **`password`** – password to connect with, overriding defaults

  New in version 2019.2.0.

CLI Example:

```
salt '*' virt.stop <domain>
```

`salt.modules.virt.`**`undefine`**`(`*vm_*, ***kwargs*`)`

Remove a defined vm, this does not purge the virtual machine image, and this only works if the vm is powered down

Parameters

- **`vm`** – domain name
- **`connection`** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.
- **`username`** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **`password`** – password to connect with, overriding defaults

  New in version 2019.2.0.

CLI Example:

```
salt '*' virt.undefine <domain>
```

`salt.modules.virt.`**`update`**`(`*name*, *cpu=0*, *mem=0*, *disk_profile=None*, *disks=None*, *nic_profile=None*, *interfaces=None*, *graphics=None*, *live=True*, *boot=None*, ***kwargs*`)`

Update the definition of an existing domain.

Parameters

- **`name`** – Name of the domain to update
- **`cpu`** – Number of virtual CPUs to assign to the virtual machine
- **`mem`** – Amount of memory to allocate to the virtual machine in MiB.
- **`disk_profile`** – disk profile to use
- **`disks`** – Disk definitions as documented in the *`init()`* function. If neither the profile nor this parameter are defined, the disk devices will not be changed. However to clear disks set this parameter to empty list.
- **`nic_profile`** – network interfaces profile to use
- **`interfaces`** – Network interface definitions as documented in the *`init()`* function. If neither the profile nor this parameter are defined, the interface devices will not be changed. However to clear network interfaces set this parameter to empty list.
- **`graphics`** – The new graphics definition as defined in *Graphics Definition*. If not set, the graphics will not be changed. To remove a graphics device, set this parameter to `{'type': 'none'}`.
- **`live`** – `False` to avoid trying to live update the definition. In such a case, the new definition is applied at the next start of the virtual machine. If `True`, not all aspects of the definition can be live updated, but as much as possible will be attempted. (Default: `True`)
- **`connection`** – libvirt connection URI, overriding defaults
- **`username`** – username to connect with, overriding defaults
- **`password`** – password to connect with, overriding defaults
- **`boot`** – Specifies kernel for the virtual machine, as well as boot parameters for the virtual machine. This is an optionl parameter, and all of the keys are optional within the dictionary. If a remote path is provided to kernel or initrd, salt will handle the downloading of the specified remote fild, and will modify the XML accordingly.

  ```
  {
      'kernel': '/root/f8-i386-vmlinuz',
      'initrd': '/root/f8-i386-initrd',
      'cmdline': 'console=ttyS0 ks=http://example.com/f8-i386/os/
  ↪'
  }
  ```

  New in version 3000.

Returns

Returns a dictionary indicating the status of what has been done. It is structured in the following way:

```
{
  'definition': True,
  'cpu': True,
  'mem': True,
  'disks': {'attached': [list of actually attached disks],
            'detached': [list of actually detached disks]},
  'nics': {'attached': [list of actually attached nics],
           'detached': [list of actually detached nics]},
  'errors': ['error messages for failures']
}
```

New in version 2019.2.0.

CLI Example:

```
salt '*' virt.update domain cpu=2 mem=1024
```

`salt.modules.virt.`**`virt_type`**`()`

Returns the virtual machine type as a string

CLI Example:

```
salt '*' virt.virt_type
```

`salt.modules.virt.`**`vm_cputime`**`(`*vm_=None, **kwargs*`)`

Return cputime used by the vms on this hyper in a list of dicts:

Parameters

- **vm** – domain name
- **connection** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.
- **username** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **password** – password to connect with, overriding defaults

  New in version 2019.2.0.

```
[
    'your-vm': {
        'cputime' <int>
        'cputime_percent' <int>
        },
    ...
    ]
```

If you pass a VM name in as an argument then it will return info for just the named VM, otherwise it will return all VMs.

CLI Example:

```
salt '*' virt.vm_cputime
```

`salt.modules.virt.`**`vm_diskstats`**`(`*vm_=None, **kwargs*`)`

Return disk usage counters used by the vms on this hyper in a list of dicts:

Parameters

- **vm** – domain name

- **connection** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.
- **username** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **password** – password to connect with, overriding defaults

  New in version 2019.2.0.

```
[
    'your-vm': {
        'rd_req'   : 0,
        'rd_bytes' : 0,
        'wr_req'   : 0,
        'wr_bytes' : 0,
        'errs'     : 0
        },
    ...
    ]
```

If you pass a VM name in as an argument then it will return info for just the named VM, otherwise it will return all VMs.

CLI Example:

```
salt '*' virt.vm_blockstats
```

`salt.modules.virt.`**`vm_info`**(*vm_=None*, ***kwargs*)

Return detailed information about the vms on this hyper in a list of dicts:

Parameters

- **vm** – name of the domain
- **connection** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.
- **username** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **password** – password to connect with, overriding defaults

  New in version 2019.2.0.

```
[
    'your-vm': {
        'cpu': <int>,
        'maxMem': <int>,
        'mem': <int>,
        'state': '<state>',
        'cputime' <int>
        },
    ...
    ]
```

If you pass a VM name in as an argument then it will return info for just the named VM, otherwise it will return all VMs.

CLI Example:

```
salt '*' virt.vm_info
```

`salt.modules.virt.`**`vm_netstats`**`(`*vm_=None*, ***kwargs*`)`

Return combined network counters used by the vms on this hyper in a list of dicts:

Parameters

- **`vm`** – domain name
- **`connection`** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.
- **`username`** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **`password`** – password to connect with, overriding defaults

  New in version 2019.2.0.

```
[
    'your-vm': {
        'rx_bytes'   : 0,
        'rx_packets' : 0,
        'rx_errs'    : 0,
        'rx_drop'    : 0,
        'tx_bytes'   : 0,
        'tx_packets' : 0,
        'tx_errs'    : 0,
        'tx_drop'    : 0
        },
    ...
    ]
```

If you pass a VM name in as an argument then it will return info for just the named VM, otherwise it will return all VMs.

CLI Example:

```
salt '*' virt.vm_netstats
```

`salt.modules.virt.`**`vm_state`**`(`*vm_=None*, ***kwargs*`)`

Return list of all the vms and their state.

If you pass a VM name in as an argument then it will return info for just the named VM, otherwise it will return all VMs.

Parameters

- **`vm`** – name of the domain
- **`connection`** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.
- **`username`** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **`password`** – password to connect with, overriding defaults

  New in version 2019.2.0.

CLI Example:

```
salt '*' virt.vm_state <domain>
```

`salt.modules.virt.`**`volume_delete`**`(`*pool*, *volume*, ***kwargs*`)`

Delete a libvirt managed volume.

Parameters

- **`pool`** – libvirt storage pool name
- **`volume`** – name of the volume to delete

- **connection** – libvirt connection URI, overriding defaults
- **username** – username to connect with, overriding defaults
- **password** – password to connect with, overriding defaults

New in version 3000.

CLI Example:

```
salt "*" virt.volume_delete <pool> <volume>
```

`salt.modules.virt.`**`volume_infos`**`(`*pool=None*, *volume=None*, ***kwargs*`)`

Provide details on a storage volume. If no volume name is provided, the infos all the volumes contained in the pool are provided. If no pool is provided, the infos of the volumes of all pools are output.

Parameters

- **pool** – libvirt storage pool name (default: `None`)
- **volume** – name of the volume to get infos from (default: `None`)
- **connection** – libvirt connection URI, overriding defaults
- **username** – username to connect with, overriding defaults
- **password** – password to connect with, overriding defaults

New in version 3000.

CLI Example:

```
salt "*" virt.volume_infos <pool> <volume>
```

21.9.450 salt.modules.virtualenv

Create virtualenv environments.

New in version 0.17.0.

`salt.modules.virtualenv_mod.`**`create`**`(`*path*, *venv_bin=None*, *system_site_packages=False*, *distribute=False*, *clear=False*, *python=None*, *extra_search_dir=None*, *never_download=None*, *prompt=None*, *pip=False*, *symlinks=None*, *upgrade=None*, *user=None*, *use_vt=False*, *saltenv='base'*, ***kwargs*`)`

Create a virtualenv

path The path to the virtualenv to be created

venv_bin The name (and optionally path) of the virtualenv command. This can also be set globally in the minion config file as `virtualenv.venv_bin`. Defaults to `virtualenv`.

system_site_packages [False] Passthrough argument given to virtualenv or pyvenv

distribute [False] Passthrough argument given to virtualenv

pip [False] Install pip after creating a virtual environment. Implies `distribute=True`

clear [False] Passthrough argument given to virtualenv or pyvenv

python [None (default)] Passthrough argument given to virtualenv

extra_search_dir [None (default)] Passthrough argument given to virtualenv

never_download [None (default)] Passthrough argument given to virtualenv if True

prompt [None (default)] Passthrough argument given to virtualenv if not None

symlinks [None] Passthrough argument given to pyvenv if True

upgrade [None] Passthrough argument given to pyvenv if True

user [None] Set ownership for the virtualenv

Note: On Windows you must also pass a `password` parameter. Additionally, the user must have permissions to the location where the virtual environment is being created

runas [None] Set ownership for the virtualenv

Deprecated since version 2014.1.0: `user` should be used instead

use_vt [False] Use VT terminal emulation (see output while installing)

New in version 2015.5.0.

saltenv ['base'] Specify a different environment. The default environment is `base`.

New in version 2014.1.0.

Note: The `runas` argument is deprecated as of 2014.1.0. `user` should be used instead.

CLI Example:

```
   salt '*' virtualenv.create /path/to/new/virtualenv

Example of using --always-copy environment variable (in case your fs doesn't
↪support symlinks).
This will copy files into the virtualenv instead of symlinking them.

.. code-block:: yaml

    - env:
      - VIRTUALENV_ALWAYS_COPY: 1
```

`salt.modules.virtualenv_mod.`**`get_distribution_path`**(*venv*, *distribution*)

Return the path to a distribution installed inside a virtualenv

New in version 2016.3.0.

venv Path to the virtualenv.

distribution Name of the distribution. Note, all non-alphanumeric characters will be converted to dashes.

CLI Example:

```
salt '*' virtualenv.get_distribution_path /path/to/my/venv my_distribution
```

`salt.modules.virtualenv_mod.`**`get_resource_content`**(*venv*, *package=None*, *resource=None*)

Return the content of a package resource installed inside a virtualenv

New in version 2015.5.0.

venv Path to the virtualenv

package Name of the package in which the resource resides

New in version 2016.3.0.

resource Name of the resource of which the content is to be returned

New in version 2016.3.0.

CLI Example:

```
salt '*' virtualenv.get_resource_content /path/to/my/venv my_package my/resource.
↪xml
```

`salt.modules.virtualenv_mod.`**`get_resource_path`**(*venv*, *package=None*, *resource=None*)

Return the path to a package resource installed inside a virtualenv

New in version 2015.5.0.

venv Path to the virtualenv

package Name of the package in which the resource resides

New in version 2016.3.0.

resource Name of the resource of which the path is to be returned

New in version 2016.3.0.

CLI Example:

```
salt '*' virtualenv.get_resource_path /path/to/my/venv my_package my/resource.xml
```

`salt.modules.virtualenv_mod.`**`get_site_packages`**`(`*venv*`)`

Return the path to the site-packages directory of a virtualenv

venv Path to the virtualenv.

CLI Example:

```
salt '*' virtualenv.get_site_packages /path/to/my/venv
```

`salt.modules.virtualenv_mod.`**`virtualenv_ver`**`(`*venv_bin, user=None, **kwargs*`)`

return virtualenv version if exists

21.9.451 salt.modules.vmctl

Manage vms running on the OpenBSD VMM hypervisor using vmctl(8).

New in version 2019.2.0.

codeauthor `Jasper Lievisse Adriaanse <jasper@openbsd.org>`

Note: This module requires the *vmd* service to be running on the OpenBSD target machine.

`salt.modules.vmctl.`**`create_disk`**`(`*name, size*`)`

Create a VMM disk with the specified *name* and *size*.

size: Size in megabytes, or use a specifier such as M, G, T.

CLI Example:

```
salt '*' vmctl.create_disk /path/to/disk.img size=10G
```

`salt.modules.vmctl.`**`load`**`(`*path*`)`

Load additional configuration from the specified file.

path Path to the configuration file.

CLI Example:

```
salt '*' vmctl.load path=/etc/vm.switches.conf
```

`salt.modules.vmctl.`**`reload`**`()`

Remove all stopped VMs and reload configuration from the default configuration file.

CLI Example:

```
salt '*' vmctl.reload
```

`salt.modules.vmctl.`**`reset`**`(`*all=False, vms=False, switches=False*`)`

Reset the running state of VMM or a subsystem.

all: Reset the running state.

switches: Reset the configured switches.

vms: Reset and terminate all VMs.

CLI Example:

```
salt '*' vmctl.reset all=True
```

`salt.modules.vmctl.`**`start`**`(`*name=None, id=None, bootpath=None, disk=None, disks=None, local_iface=False, memory=None, nics=0, switch=None*`)`

Starts a VM defined by the specified parameters. When both a name and id are provided, the id is ignored.

name: Name of the defined VM.

id: VM id.

bootpath: Path to a kernel or BIOS image to load.

disk: Path to a single disk to use.

disks: List of multiple disks to use.

local_iface: Whether to add a local network interface. See "LOCAL INTERFACES" in the vmctl(8) manual page for more information.

memory: Memory size of the VM specified in megabytes.

switch: Add a network interface that is attached to the specified virtual switch on the host.

CLI Example:

```
salt '*' vmctl.start 2   # start VM with id 2
salt '*' vmctl.start name=web1 bootpath='/bsd.rd' nics=2 memory=512M disk='/disk.
↪img'
```

`salt.modules.vmctl.`**`status`**`(`*name=None, id=None*`)`

List VMs running on the host, or only the VM specified by `id`. When both a name and id are provided, the id is ignored.

name: Name of the defined VM.

id: VM id.

CLI Example:

```
salt '*' vmctl.status           # to list all VMs
salt '*' vmctl.status name=web1 # to get a single VM
```

`salt.modules.vmctl.`**`stop`**`(`*name=None, id=None*`)`

Stop (terminate) the VM identified by the given id or name. When both a name and id are provided, the id is ignored.

name: Name of the defined VM.

id: VM id.

CLI Example:

```
salt '*' vmctl.stop name=alpine
```

21.9.452 salt.modules.vsphere

Manage VMware vCenter servers and ESXi hosts.

New in version 2015.8.4.

codeauthor Alexandru Bleotu <alexandru.bleotu@morganstaley.com>

Dependencies

- pyVmomi Python Module
- ESXCLI

pyVmomi

PyVmomi can be installed via pip:

```
pip install pyVmomi
```

Note: Version 6.0 of pyVmomi has some problems with SSL error handling on certain versions of Python. If using version 6.0 of pyVmomi, Python 2.7.9, or newer must be present. This is due to an upstream dependency in pyVmomi 6.0 that is not supported in Python versions 2.7 to 2.7.8. If the version of Python is not in the supported range, you will need to install an earlier version of pyVmomi. See Issue #29537 for more information.

Based on the note above, to install an earlier version of pyVmomi than the version currently listed in PyPi, run the following:

```
pip install pyVmomi==5.5.0.2014.1.1
```

The 5.5.0.2014.1.1 is a known stable version that this original vSphere Execution Module was developed against.

vSphere Automation SDK

vSphere Automation SDK can be installed via pip:

```
pip install --upgrade pip setuptools
pip install --upgrade git+https://github.com/vmware/vsphere-automation-sdk-python.git
```

Note: The SDK also requires OpenSSL 1.0.1+ if you want to connect to vSphere 6.5+ in order to support TLS1.1 & 1.2.

In order to use the tagging functions in this module, vSphere Automation SDK is necessary to install.

The module is currently in version 1.0.3 (as of 8/26/2019)

ESXCLI

Currently, about a third of the functions used in the vSphere Execution Module require the ESXCLI package be installed on the machine running the Proxy Minion process.

The ESXCLI package is also referred to as the VMware vSphere CLI, or vCLI. VMware provides vCLI package installation instructions for vSphere 5.5 and vSphere 6.0.

Once all of the required dependencies are in place and the vCLI package is installed, you can check to see if you can connect to your ESXi host or vCenter server by running the following command:

```
esxcli -s <host-location> -u <username> -p <password> system syslog config get
```

If the connection was successful, ESXCLI was successfully installed on your system. You should see output related to the ESXi host's syslog configuration.

Note: Be aware that some functionality in this execution module may depend on the type of license attached to a vCenter Server or ESXi host(s).

For example, certain services are only available to manipulate service state or policies with a VMware vSphere Enterprise or Enterprise Plus license, while others are available with a Standard license. The `ntpd` service is restricted to an Enterprise Plus license, while `ssh` is available via the Standard license.

Please see the vSphere Comparison page for more information.

About

This execution module was designed to be able to handle connections both to a vCenter Server, as well as to an ESXi host. It utilizes the pyVmomi Python library and the ESXCLI package to run remote execution functions against either the defined vCenter server or the ESXi host.

Whether or not the function runs against a vCenter Server or an ESXi host depends entirely upon the arguments passed into the function. Each function requires a `host` location, `username`, and `password`. If the credentials provided apply to a vCenter Server, then the function will be run against the vCenter Server. For example, when listing hosts using vCenter credentials, you'll get a list of hosts associated with that vCenter Server:

```
# salt my-minion vsphere.list_hosts <vcenter-ip> <vcenter-user> <vcenter-password>
my-minion:
- esxi-1.example.com
- esxi-2.example.com
```

However, some functions should be used against ESXi hosts, not vCenter Servers. Functionality such as getting a host's coredump network configuration should be performed against a host and not a vCenter server. If the authentication information you're using is against a vCenter server and not an ESXi host, you can provide the host name that is associated with the vCenter server in the command, as a list, using the `host_names` or `esxi_host` kwarg. For example:

```
# salt my-minion vsphere.get_coredump_network_config <vcenter-ip> <vcenter-user>
↪   <vcenter-password> esxi_hosts='[esxi-1.example.com, esxi-2.example.com]'
my-minion:
----------
    esxi-1.example.com:
        ----------
        Coredump Config:
            ----------
            enabled:
                False
    esxi-2.example.com:
        ----------
        Coredump Config:
            ----------
            enabled:
                True
            host_vnic:
                vmk0
            ip:
                coredump-location.example.com
            port:
                6500
```

You can also use these functions against an ESXi host directly by establishing a connection to an ESXi host using the host's location, username, and password. If ESXi connection credentials are used instead of vCenter credentials, the `host_names` and `esxi_hosts` arguments are not needed.

```
# salt my-minion vsphere.get_coredump_network_config esxi-1.example.com root <host-
↪password>
local:
----------
    10.4.28.150:
        ----------
        Coredump Config:
            ----------
            enabled:
                True
            host_vnic:
                vmk0
            ip:
                coredump-location.example.com
            port:
                6500
```

`salt.modules.vsphere.`**`add_capacity_to_diskgroup`**(*cache_disk_id*, *capacity_disk_ids*, *safety_checks=True*, *service_instance=None*)

Adds capacity disks to the disk group with the specified cache disk.

cache_disk_id The canonical name of the cache disk.

capacity_disk_ids A list containing canonical names of the capacity disks to add.

safety_checks Specify whether to perform safety check or to skip the checks and try performing the required task. Default value is True.

service_instance Service instance (vim.ServiceInstance) of the vCenter/ESXi host. Default is None.

```
salt '*' vsphere.add_capacity_to_diskgroup
    cache_disk_id='naa.000000000000001'
    capacity_disk_ids='[naa.000000000000002, naa.000000000000003]'
```

`salt.modules.vsphere.`**`add_host_to_dvs`**(*host*, *username*, *password*, *vmknic_name*, *vmnic_name*, *dvs_name*, *target_portgroup_name*, *uplink_portgroup_name*, *protocol=None*, *port=None*, *host_names=None*)

Adds an ESXi host to a vSphere Distributed Virtual Switch and migrates the desired adapters to the DVS from the standard switch.

host The location of the vCenter server.

username The username used to login to the vCenter server.

password The password used to login to the vCenter server.

vmknic_name The name of the virtual NIC to migrate.

vmnic_name The name of the physical NIC to migrate.

dvs_name The name of the Distributed Virtual Switch.

target_portgroup_name The name of the distributed portgroup in which to migrate the virtual NIC.

uplink_portgroup_name The name of the uplink portgroup in which to migrate the physical NIC.

protocol Optionally set to alternate protocol if the vCenter server or ESX/ESXi host is not using the default protocol. Default protocol is `https`.

port Optionally set to alternate port if the vCenter server or ESX/ESXi host is not using the default port. Default port is `443`.

host_names: An array of VMware host names to migrate

CLI Example:

```
salt some_host vsphere.add_host_to_dvs host='vsphere.corp.com'
    username='administrator@vsphere.corp.com' password='vsphere_password'
    vmknic_name='vmk0' vmnic_name='vnmic0' dvs_name='DSwitch'
```

(continues on next page)

(continued from previous page)

```
    target_portgroup_name='DPortGroup' uplink_portgroup_name='DSwitch1-DVUplinks-
↪181'
    protocol='https' port='443', host_names="['esxi1.corp.com','esxi2.corp.com',
↪'esxi3.corp.com']"
```

Return Example:

```
somehost:
    ----------
    esxi1.corp.com:
        ----------
        dvs:
            DSwitch
        portgroup:
            DPortGroup
        status:
            True
        uplink:
            DSwitch-DVUplinks-181
        vmknic:
            vmk0
        vmnic:
            vmnic0
    esxi2.corp.com:
        ----------
        dvs:
            DSwitch
        portgroup:
            DPortGroup
        status:
            True
        uplink:
            DSwitch-DVUplinks-181
        vmknic:
            vmk0
        vmnic:
            vmnic0
    esxi3.corp.com:
        ----------
        dvs:
            DSwitch
        portgroup:
            DPortGroup
        status:
            True
        uplink:
            DSwitch-DVUplinks-181
        vmknic:
            vmk0
        vmnic:
            vmnic0
    message:
    success:
        True
```

This was very difficult to figure out. VMware's PyVmomi documentation at

https://github.com/vmware/pyvmomi/blob/master/docs/vim/DistributedVirtualSwitch.rst (which is a copy of the official documentation here: https://www.vmware.com/support/developer/converter-sdk/conv60_apireference/vim.DistributedVirtualSwitch.html)

says to create the DVS, create distributed portgroups, and then add the host to the DVS specifying which physical NIC to use as the port backing. However, if the physical NIC is in use as the only link from the host to vSphere, this will fail with an unhelpful ”busy” error.

There is, however, a Powershell PowerCLI cmdlet called Add-VDSwitchPhysicalNetworkAdapter that does what we want. I used Onyx (https://labs.vmware.com/flings/onyx) to sniff the SOAP stream from Powershell to our vSphere server and got this snippet out:

```
<UpdateNetworkConfig xmlns="urn:vim25">
  <_this type="HostNetworkSystem">networkSystem-187</_this>
  <config>
    <vswitch>
      <changeOperation>edit</changeOperation>
      <name>vSwitch0</name>
      <spec>
        <numPorts>7812</numPorts>
      </spec>
    </vswitch>
    <proxySwitch>
        <changeOperation>edit</changeOperation>
        <uuid>73 a4 05 50 b0 d2 7e b9-38 80 5d 24 65 8f da 70</uuid>
        <spec>
        <backing xsi:type="DistributedVirtualSwitchHostMemberPnicBacking">
            <pnicSpec><pnicDevice>vmnic0</pnicDevice></pnicSpec>
        </backing>
        </spec>
    </proxySwitch>
    <portgroup>
      <changeOperation>remove</changeOperation>
      <spec>
        <name>Management Network</name><vlanId>-1</vlanId><vswitchName /><policy /
↪>
      </spec>
    </portgroup>
    <vnic>
      <changeOperation>edit</changeOperation>
      <device>vmk0</device>
      <portgroup />
      <spec>
        <distributedVirtualPort>
          <switchUuid>73 a4 05 50 b0 d2 7e b9-38 80 5d 24 65 8f da 70</switchUuid>
          <portgroupKey>dvportgroup-191</portgroupKey>
        </distributedVirtualPort>
      </spec>
    </vnic>
  </config>
  <changeMode>modify</changeMode>
</UpdateNetworkConfig>
```

The SOAP API maps closely to PyVmomi, so from there it was (relatively) easy to figure out what Python to write.

salt.modules.vsphere.**add_license**(*key, description, safety_checks=True, service_instance=None*)

Adds a license to the vCenter or ESXi host

key License key.

description License description added in as a label.

safety_checks Specify whether to perform safety check or to skip the checks and try performing the required task

service_instance Service instance (vim.ServiceInstance) of the vCenter/ESXi host. Default is None.

```
salt '*' vsphere.add_license key=<license_key> desc='License desc'
```

`salt.modules.vsphere.`**`assign_default_storage_policy_to_datastore`**(*policy, datastore, service_instance=None*)

Assigns a storage policy as the default policy to a datastore.

policy Name of the policy to assign.

datastore Name of the datastore to assign. The datastore needs to be visible to the VMware entity the proxy points to.

service_instance Service instance (vim.ServiceInstance) of the vCenter. Default is None.

```
salt '*' vsphere.assign_storage_policy_to_datastore
    policy='policy name' datastore=ds1
```

`salt.modules.vsphere.`**`assign_license`**(*license_key, license_name, entity, entity_display_name, safety_checks=True, service_instance=None*)

Assigns a license to an entity

license_key Key of the license to assign See `_get_entity` docstrings for format.

license_name Display name of license

entity Dictionary representation of an entity

entity_display_name Entity name used in logging

safety_checks Specify whether to perform safety check or to skip the checks and try performing the required task. Default is False.

service_instance Service instance (vim.ServiceInstance) of the vCenter/ESXi host. Default is None.

```
salt '*' vsphere.assign_license license_key=00000:00000
    license name=test entity={type:cluster,datacenter:dc,cluster:cl}
```

`salt.modules.vsphere.`**`attach_tag`**(*object_id, tag_id, managed_obj='ClusterComputeResource', server=None, username=None, password=None, service_instance=None*)

Attach an existing tag to an input object.

The tag needs to meet the cardinality (*CategoryModel.cardinality*) and associability (*CategoryModel.associable_types*) criteria in order to be eligible for attachment. If the tag is already attached to the object, then this method is a no-op and an error will not be thrown. To invoke this method, you need the attach tag privilege on the tag and the read privilege on the object.

CLI Example:

```
salt vm_minion vsphere.attach_tag domain-c2283                ⊠
↪urn:vmomi:InventoryServiceTag:b55ecc77-f4a5-49f8-ab52-38865467cfbe:GLOBAL
```

Parameters

- **`object_id`** (*`str`*) – The identifier of the input object.
- **`tag_id`** (*`str`*) – The identifier of the tag object.
- **`managed_obj`** (*`str`*) – Classes that contain methods for creating and deleting resources typically contain a class attribute specifying the resource type for the resources being created and deleted.
- **`server`** (*`basestring`*) – Target DNS or IP of vCenter center.
- **`username`** (*`basestring`*) – Username associated with the vCenter center.
- **`password`** (*`basestring`*) – Password associated with the vCenter center.

Returns The list of all tag identifiers that correspond to the tags attached to the given object.

Return type list of tags
Raise Unauthorized if you do not have the privilege to read the object.
Raise Unauthenticated if the user can not be authenticated.

`salt.modules.vsphere.`**`compare_vm_configs`**(*new_config*, *current_config*)

Compares virtual machine current and new configuration, the current is the one which is deployed now, and the new is the target config. Returns the differences between the objects in a dictionary, the keys are the configuration parameter keys and the values are differences objects: either list or recursive difference

new_config: New config dictionary with every available parameter

current_config Currently deployed configuration

`salt.modules.vsphere.`**`configure_host_cache`**(*enabled*, *datastore=None*, *swap_size_MiB=None*, *service_instance=None*)

Configures the host cache on the selected host.

enabled Boolean flag specifying whether the host cache is enabled.

datastore Name of the datastore that contains the host cache. Must be set if enabled is `true`.

swap_size_MiB Swap size in Mibibytes. Needs to be set if enabled is `true`. Must be smaller than the datastore size.

service_instance Service instance (vim.ServiceInstance) of the vCenter/ESXi host. Default is None.

```
salt '*' vsphere.configure_host_cache enabled=False

salt '*' vsphere.configure_host_cache enabled=True datastore=ds1
    swap_size_MiB=1024
```

`salt.modules.vsphere.`**`coredump_network_enable`**(*host*, *username*, *password*, *enabled*, *protocol=None*, *port=None*, *esxi_hosts=None*, *credstore=None*)

Enable or disable ESXi core dump collection. Returns `True` if coredump is enabled and returns `False` if core dump is not enabled. If there was an error, the error will be the value printed in the `Error` key dictionary for the given host.

host The location of the host.

username The username used to login to the host, such as `root`.

password The password used to login to the host.

enabled Python True or False to enable or disable coredumps.

protocol Optionally set to alternate protocol if the host is not using the default protocol. Default protocol is `https`.

port Optionally set to alternate port if the host is not using the default port. Default port is `443`.

esxi_hosts If `host` is a vCenter host, then use esxi_hosts to execute this function on a list of one or more ESXi machines.

credstore Optionally set to path to the credential store file.

CLI Example:

```
# Used for ESXi host connection information
salt '*' vsphere.coredump_network_enable my.esxi.host root bad-password True

# Used for connecting to a vCenter Server
salt '*' vsphere.coredump_network_enable my.vcenter.location root bad-password
↪True               esxi_hosts='[esxi-1.host.com, esxi-2.host.com]'
```

`salt.modules.vsphere.`**`create_cluster`**(*cluster_dict*, *datacenter=None*, *cluster=None*, *service_instance=None*)

Creates a cluster.

Note: cluster_dict['name'] will be overridden by the cluster param value

config_dict Dictionary with the config values of the new cluster.

datacenter Name of datacenter containing the cluster. Ignored if already contained by proxy details. Default value is None.

cluster Name of cluster. Ignored if already contained by proxy details. Default value is None.

service_instance Service instance (vim.ServiceInstance) of the vCenter. Default is None.

```
# esxdatacenter proxy
salt '*' vsphere.create_cluster cluster_dict=$cluster_dict cluster=cl1

# esxcluster proxy
salt '*' vsphere.create_cluster cluster_dict=$cluster_dict
```

`salt.modules.vsphere.`**`create_datacenter`**(*datacenter_name*, *service_instance=None*)

Creates a datacenter.

Supported proxies: esxdatacenter

datacenter_name The datacenter name

service_instance Service instance (vim.ServiceInstance) of the vCenter. Default is None.

```
salt '*' vsphere.create_datacenter dc1
```

`salt.modules.vsphere.`**`create_diskgroup`**(*cache_disk_id*, *capacity_disk_ids*, *safety_checks=True*, *service_instance=None*)

Creates disk group on an ESXi host with the specified cache and capacity disks.

cache_disk_id The canonical name of the disk to be used as a cache. The disk must be ssd.

capacity_disk_ids A list containing canonical names of the capacity disks. Must contain at least one id. Default is True.

safety_checks Specify whether to perform safety check or to skip the checks and try performing the required task. Default value is True.

service_instance Service instance (vim.ServiceInstance) of the vCenter/ESXi host. Default is None.

```
salt '*' vsphere.create_diskgroup cache_disk_id='naa.000000000000001'
    capacity_disk_ids='[naa.000000000000002, naa.000000000000003]'
```

`salt.modules.vsphere.`**`create_dvportgroup`**(*portgroup_dict*, *portgroup_name*, *dvs*, *service_instance=None*)

Creates a distributed virtual portgroup.

Note: The `portgroup_name` param will override any name already set in `portgroup_dict`.

portgroup_dict Dictionary with the config values the portgroup should be created with (example in salt.states.dvs).

portgroup_name Name of the portgroup to be created.

dvs Name of the DVS that will contain the portgroup.

service_instance Service instance (vim.ServiceInstance) of the vCenter. Default is None.

```
salt '*' vsphere.create_dvportgroup portgroup_dict=<dict>
    portgroup_name=pg1 dvs=dvs1
```

`salt.modules.vsphere.`**`create_dvs`**(*dvs_dict*, *dvs_name*, *service_instance=None*)

Creates a distributed virtual switch (DVS).

Note: The `dvs_name` param will override any name set in `dvs_dict`.

dvs_dict Dict representation of the new DVS (example in salt.states.dvs)

dvs_name Name of the DVS to be created.

service_instance Service instance (vim.ServiceInstance) of the vCenter. Default is None.

```
salt '*' vsphere.create_dvs dvs dict=$dvs_dict dvs_name=dvs_name
```

`salt.modules.vsphere.`**`create_storage_policy`**(*policy_name*, *policy_dict*, *service_instance=None*)

Creates a storage policy.

Supported capability types: scalar, set, range.

policy_name Name of the policy to create. The value of the argument will override any existing name in `policy_dict`.

policy_dict Dictionary containing the changes to apply to the policy. (example in salt.states.pbm)

service_instance Service instance (vim.ServiceInstance) of the vCenter. Default is None.

```
salt '*' vsphere.create_storage_policy policy_name='policy name'
    policy_dict="$policy_dict"
```

`salt.modules.vsphere.`**`create_tag`**(*name*, *description*, *category_id*, *server=None*, *username=None*, *password=None*, *service_instance=None*)

Create a tag under a category with given description.

CLI Example:

```
salt vm_minion vsphere.create_tag
```

Parameters
- **`server`** (*`basestring`*) – Target DNS or IP of vCenter client.
- **`username`** (*`basestring`*) – Username associated with the vCenter client.
- **`password`** (*`basestring`*) – Password associated with the vCenter client.
- **`name`** (*`str`*) – Name of tag category to create (ex. Machine, OS, Availability, etc.)
- **`description`** (*`str`*) – Given description of tag category.
- **`category_id`** (*`str`*) – Value of category_id representative of the category created previously.

Returns The identifier of the created tag.

Return type str

Raise AlreadyExists if the name provided in the create_spec is the name of an already existing tag in the input category.

Raise InvalidArgument if any of the input information in the create_spec is invalid.

Raise NotFound if the category for in the given create_spec does not exist in the system.

Raise Unauthorized if you do not have the privilege to create tag.

`salt.modules.vsphere.`**`create_tag_category`**(*name*, *description*, *cardinality*, *server=None*, *username=None*, *password=None*, *service_instance=None*)

Create a category with given cardinality.

CLI Example:

```
salt vm_minion vsphere.create_tag_category
```

Parameters
- **`name`** (*`str`*) – Name of tag category to create (ex. Machine, OS, Availability, etc.)
- **`description`** (*`str`*) – Given description of tag category.
- **`cardinality`** (*`str`*) – The associated cardinality (SINGLE, MULTIPLE) of the category.
- **`server`** (*`basestring`*) – Target DNS or IP of vCenter center.
- **`username`** (*`basestring`*) – Username associated with the vCenter center.
- **`password`** (*`basestring`*) – Password associated with the vCenter center.

Returns Identifier of the created category.

Return type str

Raise AlreadyExists if the name‘ provided in the create_spec is the name of an already existing category.

Raise InvalidArgument if any of the information in the create_spec is invalid.

Raise Unauthorized if you do not have the privilege to create a category.

`salt.modules.vsphere.`**`create_vm`**(*vm_name, cpu, memory, image, version, datacenter, datastore, placement, interfaces, disks, scsi_devices, serial_ports=None, ide_controllers=None, sata_controllers=None, cd_drives=None, advanced_configs=None, service_instance=None*)

Creates a virtual machine container.

CLI Example:

```
salt vm_minion vsphere.create_vm vm_name=vmname cpu='{count: 2, nested: True}' ...
```

vm_name Name of the virtual machine
cpu Properties of CPUs for freshly created machines
memory Memory size for freshly created machines
image Virtual machine guest OS version identifier VirtualMachineGuestOsIdentifier
version Virtual machine container hardware version
datacenter Datacenter where the virtual machine will be deployed (mandatory)
datastore Datastore where the virtual machine files will be placed
placement Resource pool or cluster or host or folder where the virtual machine will be deployed
devices interfaces

```
interfaces:
  adapter: 'Network adapter 1'
  name: vlan100
  switch_type: distributed or standard
  adapter_type: vmxnet3 or vmxnet, vmxnet2, vmxnet3, e1000, e1000e
  mac: '00:11:22:33:44:55'
  connectable:
    allow_guest_control: True
    connected: True
    start_connected: True
```

disks

```
disks:
  adapter: 'Hard disk 1'
  size: 16
  unit: GB
  address: '0:0'
  controller: 'SCSI controller 0'
  thin_provision: False
  eagerly_scrub: False
  datastore: 'myshare'
  filename: 'vm/mydisk.vmdk'
```

scsi_devices

```
scsi_devices:
  controller: 'SCSI controller 0'
  type: paravirtual
  bus_sharing: no_sharing
```

serial_ports

```
serial_ports:
  adapter: 'Serial port 1'
  type: network
```

(continues on next page)

(continued from previous page)

```
  backing:
    uri: 'telnet://something:port'
    direction: <client|server>
    filename: 'service_uri'
  connectable:
    allow_guest_control: True
    connected: True
    start_connected: True
  yield: False
```

cd_drives

```
cd_drives:
  adapter: 'CD/DVD drive 0'
  controller: 'IDE 0'
  device_type: datastore_iso_file
  datastore_iso_file:
    path: path_to_iso
  connectable:
    allow_guest_control: True
    connected: True
    start_connected: True
```

advanced_config Advanced config parameters to be set for the virtual machine

`salt.modules.vsphere.`**`create_vmfs_datastore`**(*datastore_name, disk_id, vmfs_major_version, safety_checks=True, service_instance=None*)

Creates a ESXi host disk group with the specified cache and capacity disks.

datastore_name The name of the datastore to be created.

disk_id The disk id (canonical name) on which the datastore is created.

vmfs_major_version The VMFS major version.

safety_checks Specify whether to perform safety check or to skip the checks and try performing the required task. Default is True.

service_instance Service instance (vim.ServiceInstance) of the vCenter/ESXi host. Default is None.

```
salt '*' vsphere.create_vmfs_datastore datastore_name=ds1 disk_id=
    vmfs_major_version=5
```

`salt.modules.vsphere.`**`delete_advanced_configs`**(*vm_name, datacenter, advanced_configs, service_instance=None*)

Removes extra config parameters from a virtual machine

vm_name Virtual machine name

datacenter Datacenter name where the virtual machine is available

advanced_configs List of advanced config values to be removed

service_instance vCenter service instance for connection and configuration

`salt.modules.vsphere.`**`delete_tag`**(*tag_id, server=None, username=None, password=None, service_instance=None*)

Delete a tag.

CLI Example:

```
salt vm_minion vsphere.delete_tag
```

Parameters

- **`tag_id`** (*`str`*) – The identifier of tag to be deleted. The parameter must be an identifier for the resource type: `com.vmware.cis.tagging.Tag`.
- **`server`** (*`basestring`*) – Target DNS or IP of vCenter center.

- **username** (*basestring*) – Username associated with the vCenter center.
- **password** (*basestring*) – Password associated with the vCenter center.

Raise AlreadyExists if the name provided in the create_spec is the name of an already existing category.
Raise InvalidArgument if any of the information in the create_spec is invalid.
Raise Unauthorized if you do not have the privilege to create a category.

`salt.modules.vsphere.`**`delete_tag_category`**(*category_id*, *server=None*, *username=None*, *password=None*, *service_instance=None*)

Delete a category.

CLI Example:

```
salt vm_minion vsphere.delete_tag_category
```

Parameters
- **category_id** (*str*) – The identifier of category to be deleted. The parameter must be an identifier for the resource type: `com.vmware.cis.tagging.Category`.
- **server** (*basestring*) – Target DNS or IP of vCenter center.
- **username** (*basestring*) – Username associated with the vCenter center.
- **password** (*basestring*) – Password associated with the vCenter center.

Raise NotFound if the tag for the given tag_id does not exist in the system.
Raise Unauthorized if you do not have the privilege to delete the tag.
Raise Unauthenticated if the user can not be authenticated.

`salt.modules.vsphere.`**`delete_vm`**(*name*, *datacenter*, *placement=None*, *power_off=False*, *service_instance=None*)

Deletes a virtual machine defined by name and placement
name Name of the virtual machine
datacenter Datacenter of the virtual machine
placement Placement information of the virtual machine
service_instance vCenter service instance for connection and configuration

```
salt '*' vsphere.delete_vm name=my_vm datacenter=my_datacenter
```

`salt.modules.vsphere.`**`disconnect`**(*service_instance*)

Disconnects from a vCenter or ESXi host

Note: Should be used by state functions, not invoked directly.

service_instance Service instance (vim.ServiceInstance)

CLI Example:
See note above.

`salt.modules.vsphere.`**`enable_firewall_ruleset`**(*host*, *username*, *password*, *ruleset_enable*, *ruleset_name*, *protocol=None*, *port=None*, *esxi_hosts=None*, *credstore=None*)

Enable or disable an ESXi firewall rule set.
host The location of the host.
username The username used to login to the host, such as `root`.
password The password used to login to the host.
ruleset_enable True to enable the ruleset, false to disable.
ruleset_name Name of ruleset to target.
protocol Optionally set to alternate protocol if the host is not using the default protocol. Default protocol is `https`.
port Optionally set to alternate port if the host is not using the default port. Default port is `443`.

esxi_hosts If `host` is a vCenter host, then use esxi_hosts to execute this function on a list of one or more ESXi machines.

credstore Optionally set to path to the credential store file.

Returns A standard cmd.run_all dictionary, per host.

CLI Example:

```
# Used for ESXi host connection information
salt '*' vsphere.enable_firewall_ruleset my.esxi.host root bad-password True
↪'syslog'

# Used for connecting to a vCenter Server
salt '*' vsphere.enable_firewall_ruleset my.vcenter.location root bad-password
↪True 'syslog'             esxi_hosts='[esxi-1.host.com, esxi-2.host.com]'
```

`salt.modules.vsphere.`**`erase_disk_partitions`**(*disk_id=None*, *scsi_address=None*, *service_instance=None*)

Erases the partitions on a disk. The disk can be specified either by the canonical name, or by the scsi_address.

disk_id Canonical name of the disk. Either `disk_id` or `scsi_address` needs to be specified (`disk_id` supersedes `scsi_address`.

scsi_address Scsi address of the disk. `disk_id` or `scsi_address` needs to be specified (`disk_id` supersedes `scsi_address`.

service_instance Service instance (vim.ServiceInstance) of the vCenter/ESXi host. Default is None.

```
salt '*' vsphere.erase_disk_partitions scsi_address='vmhaba0:C0:T0:L0'

salt '*' vsphere.erase_disk_partitions disk_id='naa.000000000000001'
```

`salt.modules.vsphere.`**`esxcli_cmd`**(*cmd_str*, *host=None*, *username=None*, *password=None*, *protocol=None*, *port=None*, *esxi_hosts=None*, *credstore=None*)

Run an ESXCLI command directly on the host or list of hosts.

host The location of the host.

username The username used to login to the host, such as `root`.

password The password used to login to the host.

cmd_str The ESXCLI command to run. Note: This should not include the `-s`, `-u`, `-p`, `-h`, `--protocol`, or `--portnumber` arguments that are frequently passed when using a bare ESXCLI command from the command line. Those arguments are handled by this function via the other args and kwargs.

protocol Optionally set to alternate protocol if the host is not using the default protocol. Default protocol is `https`.

port Optionally set to alternate port if the host is not using the default port. Default port is `443`.

esxi_hosts If `host` is a vCenter host, then use esxi_hosts to execute this function on a list of one or more ESXi machines.

credstore Optionally set to path to the credential store file.

CLI Example:

```
# Used for ESXi host connection information
salt '*' vsphere.esxcli_cmd my.esxi.host root bad-password             'system
↪coredump network get'

# Used for connecting to a vCenter Server
salt '*' vsphere.esxcli_cmd my.vcenter.location root bad-password
↪'system coredump network get' esxi_hosts='[esxi-1.host.com, esxi-2.host.com]'
```

`salt.modules.vsphere.`**`get_advanced_configs`**(*vm_name*, *datacenter*, *service_instance=None*)

Returns extra config parameters from a virtual machine advanced config list

vm_name Virtual machine name

datacenter Datacenter name where the virtual machine is available
service_instance vCenter service instance for connection and configuration

`salt.modules.vsphere.`**`get_coredump_network_config`**(*host, username, password, protocol=None, port=None, esxi_hosts=None, credstore=None*)

Retrieve information on ESXi or vCenter network dump collection and format it into a dictionary.

host The location of the host.
username The username used to login to the host, such as `root`.
password The password used to login to the host.
protocol Optionally set to alternate protocol if the host is not using the default protocol. Default protocol is `https`.
port Optionally set to alternate port if the host is not using the default port. Default port is `443`.
esxi_hosts If `host` is a vCenter host, then use esxi_hosts to execute this function on a list of one or more ESXi machines.
credstore Optionally set to path to the credential store file.

Returns A dictionary with the network configuration, or, if getting the network config failed, a an error message retrieved from the standard cmd.run_all dictionary, per host.

CLI Example:

```
# Used for ESXi host connection information
salt '*' vsphere.get_coredump_network_config my.esxi.host root bad-password

# Used for connecting to a vCenter Server
salt '*' vsphere.get_coredump_network_config my.vcenter.location root bad-
↪password            esxi_hosts='[esxi-1.host.com, esxi-2.host.com]'
```

`salt.modules.vsphere.`**`get_firewall_status`**(*host, username, password, protocol=None, port=None, esxi_hosts=None, credstore=None*)

Show status of all firewall rule sets.

host The location of the host.
username The username used to login to the host, such as `root`.
password The password used to login to the host.
protocol Optionally set to alternate protocol if the host is not using the default protocol. Default protocol is `https`.
port Optionally set to alternate port if the host is not using the default port. Default port is `443`.
esxi_hosts If `host` is a vCenter host, then use esxi_hosts to execute this function on a list of one or more ESXi machines.
credstore Optionally set to path to the credential store file.

Returns Nested dictionary with two toplevel keys `rulesets` and `success` `success` will be True or False depending on query success `rulesets` will list the rulesets and their statuses if `success` was true, per host.

CLI Example:

```
# Used for ESXi host connection information
salt '*' vsphere.get_firewall_status my.esxi.host root bad-password

# Used for connecting to a vCenter Server
salt '*' vsphere.get_firewall_status my.vcenter.location root bad-password    ↩
↪    esxi_hosts='[esxi-1.host.com, esxi-2.host.com]'
```

`salt.modules.vsphere.`**`get_host_cache`**(*service_instance=None*)

Returns the host cache configuration on the proxy host.

service_instance Service instance (vim.ServiceInstance) of the vCenter/ESXi host. Default is None.

```
salt '*' vsphere.get_host_cache
```

`salt.modules.vsphere.`**`get_host_datetime`**(*host, username, password, protocol=None, port=None, host_names=None*)

Get the date/time information for a given host or list of host_names.

host The location of the host.

username The username used to login to the host, such as `root`.

password The password used to login to the host.

protocol Optionally set to alternate protocol if the host is not using the default protocol. Default protocol is `https`.

port Optionally set to alternate port if the host is not using the default port. Default port is `443`.

host_names List of ESXi host names. When the host, username, and password credentials are provided for a vCenter Server, the host_names argument is required to tell vCenter the hosts for which to get date/time information.

If host_names is not provided, the date/time information will be retrieved for the `host` location instead. This is useful for when service instance connection information is used for a single ESXi host.

CLI Example:

```
# Used for single ESXi host connection information
salt '*' vsphere.get_host_datetime my.esxi.host root bad-password

# Used for connecting to a vCenter Server
salt '*' vsphere.get_host_datetime my.vcenter.location root bad-password      ⊠
↪host_names='[esxi-1.host.com, esxi-2.host.com]'
```

`salt.modules.vsphere.`**`get_ntp_config`**(*host, username, password, protocol=None, port=None, host_names=None*)

Get the NTP configuration information for a given host or list of host_names.

host The location of the host.

username The username used to login to the host, such as `root`.

password The password used to login to the host.

protocol Optionally set to alternate protocol if the host is not using the default protocol. Default protocol is `https`.

port Optionally set to alternate port if the host is not using the default port. Default port is `443`.

host_names List of ESXi host names. When the host, username, and password credentials are provided for a vCenter Server, the host_names argument is required to tell vCenter the hosts for which to get ntp configuration information.

If host_names is not provided, the NTP configuration will be retrieved for the `host` location instead. This is useful for when service instance connection information is used for a single ESXi host.

CLI Example:

```
# Used for single ESXi host connection information
salt '*' vsphere.get_ntp_config my.esxi.host root bad-password

# Used for connecting to a vCenter Server
salt '*' vsphere.get_ntp_config my.vcenter.location root bad-password      ⊠
↪host_names='[esxi-1.host.com, esxi-2.host.com]'
```

`salt.modules.vsphere.`**`get_proxy_type`**()

Returns the proxy type retrieved either from the pillar of from the proxy minion's config. Returns `<undefined>` otherwise.

CLI Example:

```
salt '*' vsphere.get_proxy_type
```

`salt.modules.vsphere.`**`get_service_instance_via_proxy`**(*service_instance=None*)

Returns a service instance to the proxied endpoint (vCenter/ESXi host).

service_instance Service instance (vim.ServiceInstance) of the vCenter. Default is None.

Note: Should be used by state functions not invoked directly.

CLI Example: See note above

`salt.modules.vsphere.`**`get_service_policy`**(*host, username, password, service_name, protocol=None, port=None, host_names=None*)

Get the service name's policy for a given host or list of hosts.

host The location of the host.

username The username used to login to the host, such as `root`.

password The password used to login to the host.

service_name

The name of the service for which to retrieve the policy. Supported service names are:

- DCUI
- TSM
- SSH
- lbtd
- lsassd
- lwiod
- netlogond
- ntpd
- sfcbd-watchdog
- snmpd
- vprobed
- vpxa
- xorg

protocol Optionally set to alternate protocol if the host is not using the default protocol. Default protocol is `https`.

port Optionally set to alternate port if the host is not using the default port. Default port is `443`.

host_names List of ESXi host names. When the host, username, and password credentials are provided for a vCenter Server, the host_names argument is required to tell vCenter the hosts for which to get service policy information.

If host_names is not provided, the service policy information will be retrieved for the `host` location instead. This is useful for when service instance connection information is used for a single ESXi host.

CLI Example:

```
# Used for single ESXi host connection information
salt '*' vsphere.get_service_policy my.esxi.host root bad-password 'ssh'

# Used for connecting to a vCenter Server
salt '*' vsphere.get_service_policy my.vcenter.location root bad-password 'ntpd' ␣
↪       host_names='[esxi-1.host.com, esxi-2.host.com]'
```

`salt.modules.vsphere.`**`get_service_running`**(*host, username, password, service_name, protocol=None, port=None, host_names=None*)

Get the service name's running state for a given host or list of hosts.

host The location of the host.

username The username used to login to the host, such as `root`.

password The password used to login to the host.

service_name
The name of the service for which to retrieve the policy. Supported service names are:

- DCUI
- TSM
- SSH
- lbtd
- lsassd
- lwiod
- netlogond
- ntpd
- sfcbd-watchdog
- snmpd
- vprobed
- vpxa
- xorg

protocol Optionally set to alternate protocol if the host is not using the default protocol. Default protocol is `https`.

port Optionally set to alternate port if the host is not using the default port. Default port is `443`.

host_names List of ESXi host names. When the host, username, and password credentials are provided for a vCenter Server, the host_names argument is required to tell vCenter the hosts for which to get the service's running state.

If host_names is not provided, the service's running state will be retrieved for the `host` location instead. This is useful for when service instance connection information is used for a single ESXi host.

CLI Example:

```
# Used for single ESXi host connection information
salt '*' vsphere.get_service_running my.esxi.host root bad-password 'ssh'

# Used for connecting to a vCenter Server
salt '*' vsphere.get_service_running my.vcenter.location root bad-password 'ntpd'
↪        host_names='[esxi-1.host.com, esxi-2.host.com]'
```

`salt.modules.vsphere.`**`get_ssh_key`**(*host, username, password, protocol=None, port=None, certificate_verify=False*)

Retrieve the authorized_keys entry for root. This function only works for ESXi, not vCenter.

Parameters

- **`host`** – The location of the ESXi Host
- **`username`** – Username to connect as
- **`password`** – Password for the ESXi web endpoint
- **`protocol`** – defaults to https, can be http if ssl is disabled on ESXi
- **`port`** – defaults to 443 for https
- **`certificate_verify`** – If true require that the SSL connection present a valid certificate

Returns True if upload is successful

CLI Example:

```
salt '*' vsphere.get_ssh_key my.esxi.host root bad-password certificate_
↪verify=True
```

`salt.modules.vsphere.`**`get_syslog_config`**(*host, username, password, protocol=None, port=None, esxi_hosts=None, credstore=None*)

Retrieve the syslog configuration.

host The location of the host.

username The username used to login to the host, such as `root`.

password The password used to login to the host.

protocol Optionally set to alternate protocol if the host is not using the default protocol. Default protocol is `https`.

port Optionally set to alternate port if the host is not using the default port. Default port is `443`.

esxi_hosts If `host` is a vCenter host, then use esxi_hosts to execute this function on a list of one or more ESXi machines.

credstore Optionally set to path to the credential store file.

Returns Dictionary with keys and values corresponding to the syslog configuration, per host.

CLI Example:

```
# Used for ESXi host connection information
salt '*' vsphere.get_syslog_config my.esxi.host root bad-password

# Used for connecting to a vCenter Server
salt '*' vsphere.get_syslog_config my.vcenter.location root bad-password
↪   esxi_hosts='[esxi-1.host.com, esxi-2.host.com]'
```

`salt.modules.vsphere.`**`get_vm`**(*name*, *datacenter=None*, *vm_properties=None*, *traversal_spec=None*, *parent_ref=None*, *service_instance=None*)

Returns vm object properties.

name Name of the virtual machine.

datacenter Datacenter name

vm_properties List of vm properties.

traversal_spec Traversal Spec object(s) for searching.

parent_ref Container Reference object for searching under a given object.

service_instance Service instance (vim.ServiceInstance) of the vCenter. Default is None.

`salt.modules.vsphere.`**`get_vm_config`**(*name*, *datacenter=None*, *objects=True*, *service_instance=None*)

Queries and converts the virtual machine properties to the available format from the schema. If the objects attribute is True the config objects will have extra properties, like 'object' which will include the vim.vm.device.VirtualDevice, this is necessary for deletion and update actions.

name Name of the virtual machine

datacenter Datacenter's name where the virtual machine is available

objects Indicates whether to return the vmware object properties (eg. object, key) or just the properties which can be set

service_instance vCenter service instance for connection and configuration

`salt.modules.vsphere.`**`get_vm_config_file`**(*name*, *datacenter*, *placement*, *datastore*, *service_instance=None*)

Queries the virtual machine config file and returns vim.host.DatastoreBrowser.SearchResults object on success None on failure

name Name of the virtual machine

datacenter Datacenter name

datastore Datastore where the virtual machine files are stored

service_instance Service instance (vim.ServiceInstance) of the vCenter. Default is None.

`salt.modules.vsphere.`**`get_vmotion_enabled`**(*host*, *username*, *password*, *protocol=None*, *port=None*, *host_names=None*)

Get the VMotion enabled status for a given host or a list of host_names. Returns `True` if VMotion is enabled, `False` if it is not enabled.

host The location of the host.

username The username used to login to the host, such as `root`.

password The password used to login to the host.

protocol Optionally set to alternate protocol if the host is not using the default protocol. Default protocol is `https`.

port Optionally set to alternate port if the host is not using the default port. Default port is `443`.

host_names List of ESXi host names. When the host, username, and password credentials are provided for a vCenter Server, the host_names argument is required to tell vCenter which hosts to check if VMotion is enabled.

If host_names is not provided, the VMotion status will be retrieved for the `host` location instead. This is useful for when service instance connection information is used for a single ESXi host.

CLI Example:

```
# Used for single ESXi host connection information
salt '*' vsphere.get_vmotion_enabled my.esxi.host root bad-password

# Used for connecting to a vCenter Server
salt '*' vsphere.get_vmotion_enabled my.vcenter.location root bad-password
↪ host_names='[esxi-1.host.com, esxi-2.host.com]'
```

`salt.modules.vsphere.`**`get_vsan_eligible_disks`**(*host*, *username*, *password*, *protocol=None*, *port=None*, *host_names=None*)

Returns a list of VSAN-eligible disks for a given host or list of host_names.

host The location of the host.

username The username used to login to the host, such as `root`.

password The password used to login to the host.

protocol Optionally set to alternate protocol if the host is not using the default protocol. Default protocol is `https`.

port Optionally set to alternate port if the host is not using the default port. Default port is `443`.

host_names List of ESXi host names. When the host, username, and password credentials are provided for a vCenter Server, the host_names argument is required to tell vCenter which hosts to check if any VSAN-eligible disks are available.

If host_names is not provided, the VSAN-eligible disks will be retrieved for the `host` location instead. This is useful for when service instance connection information is used for a single ESXi host.

CLI Example:

```
# Used for single ESXi host connection information
salt '*' vsphere.get_vsan_eligible_disks my.esxi.host root bad-password

# Used for connecting to a vCenter Server
salt '*' vsphere.get_vsan_eligible_disks my.vcenter.location root bad-password
↪     host_names='[esxi-1.host.com, esxi-2.host.com]'
```

`salt.modules.vsphere.`**`get_vsan_enabled`**(*host*, *username*, *password*, *protocol=None*, *port=None*, *host_names=None*)

Get the VSAN enabled status for a given host or a list of host_names. Returns `True` if VSAN is enabled, `False` if it is not enabled, and `None` if a VSAN Host Config is unset, per host.

host The location of the host.

username The username used to login to the host, such as `root`.

password The password used to login to the host.

protocol Optionally set to alternate protocol if the host is not using the default protocol. Default protocol is `https`.

port Optionally set to alternate port if the host is not using the default port. Default port is `443`.

host_names List of ESXi host names. When the host, username, and password credentials are provided for a vCenter Server, the host_names argument is required to tell vCenter which hosts to check if VSAN enabled.

If host_names is not provided, the VSAN status will be retrieved for the `host` location instead. This is useful for when service instance connection information is used for a single ESXi host.

CLI Example:

```
# Used for single ESXi host connection information
salt '*' vsphere.get_vsan_enabled my.esxi.host root bad-password

# Used for connecting to a vCenter Server
salt '*' vsphere.get_vsan_enabled my.vcenter.location root bad-password
↪host_names='[esxi-1.host.com, esxi-2.host.com]'
```

`salt.modules.vsphere.`**`gets_service_instance_via_proxy`**(*fn*)

Decorator that connects to a target system (vCenter or ESXi host) using the proxy details and passes the connection (vim.ServiceInstance) to the decorated function.

Supported proxies: esxi, esxcluster, esxdatacenter.

Notes

1. The decorated function must have a `service_instance` parameter or a `**kwarg` type argument (name of argument is not important); 2. If the `service_instance` parameter is already defined, the value is passed through to the decorated function; 3. If the `service_instance` parameter in not defined, the connection is created using the proxy details and the service instance is returned.
CLI Example: None, this is a decorator

`salt.modules.vsphere.`**`list_assigned_licenses`**(*entity*, *entity_display_name*, *license_keys=None*, *service_instance=None*)

Lists the licenses assigned to an entity
entity Dictionary representation of an entity. See `_get_entity` docstrings for format.
entity_display_name Entity name used in logging
license_keys: List of license keys to be retrieved. Default is None.
service_instance Service instance (vim.ServiceInstance) of the vCenter/ESXi host. Default is None.

```
salt '*' vsphere.list_assigned_licenses
    entity={type:cluster,datacenter:dc,cluster:cl}
    entiy_display_name=cl
```

`salt.modules.vsphere.`**`list_attached_tags`**(*object_id*, *managed_obj='ClusterComputeResource'*, *server=None*, *username=None*, *password=None*, *service_instance=None*)

List existing tags a user has access to.

CLI Example:

```
salt vm_minion vsphere.list_attached_tags domain-c2283
```

Parameters
- **`object_id`** (*`str`*) – The identifier of the input object.
- **`managed_obj`** (*`str`*) – Classes that contain methods for creating and deleting resources typically contain a class attribute specifying the resource type for the resources being created and deleted.
- **`server`** (*`basestring`*) – Target DNS or IP of vCenter center.
- **`username`** (*`basestring`*) – Username associated with the vCenter center.
- **`password`** (*`basestring`*) – Password associated with the vCenter center.

Returns The list of all tag identifiers that correspond to the tags attached to the given object.
Return type list of tags
Raise Unauthorized if you do not have the privilege to read the object.
Raise Unauthenticated if the user can not be authenticated.

`salt.modules.vsphere.`**`list_capability_definitions`**(*service_instance=None*)

Returns a list of the metadata of all capabilities in the vCenter.

service_instance Service instance (vim.ServiceInstance) of the vCenter. Default is None.

```
salt '*' vsphere.list_capabilities
```

`salt.modules.vsphere.`**`list_cluster`**(*datacenter=None, cluster=None, service_instance=None*)

Returns a dict representation of an ESX cluster.

datacenter Name of datacenter containing the cluster. Ignored if already contained by proxy details. Default value is None.

cluster Name of cluster. Ignored if already contained by proxy details. Default value is None.

service_instance Service instance (vim.ServiceInstance) of the vCenter. Default is None.

```
# vcenter proxy
salt '*' vsphere.list_cluster datacenter=dc1 cluster=cl1

# esxdatacenter proxy
salt '*' vsphere.list_cluster cluster=cl1

# esxcluster proxy
salt '*' vsphere.list_cluster
```

`salt.modules.vsphere.`**`list_clusters`**(*host, username, password, protocol=None, port=None*)

Returns a list of clusters for the the specified host.

host The location of the host.

username The username used to login to the host, such as `root`.

password The password used to login to the host.

protocol Optionally set to alternate protocol if the host is not using the default protocol. Default protocol is `https`.

port Optionally set to alternate port if the host is not using the default port. Default port is `443`.

CLI Example:

```
salt '*' vsphere.list_clusters 1.2.3.4 root bad-password
```

`salt.modules.vsphere.`**`list_datacenters`**(*host, username, password, protocol=None, port=None*)

Returns a list of datacenters for the the specified host.

host The location of the host.

username The username used to login to the host, such as `root`.

password The password used to login to the host.

protocol Optionally set to alternate protocol if the host is not using the default protocol. Default protocol is `https`.

port Optionally set to alternate port if the host is not using the default port. Default port is `443`.

CLI Example:

```
salt '*' vsphere.list_datacenters 1.2.3.4 root bad-password
```

`salt.modules.vsphere.`**`list_datacenters_via_proxy`**(*datacenter_names=None, service_instance=None*)

Returns a list of dict representations of VMware datacenters. Connection is done via the proxy details.

Supported proxies: esxdatacenter

datacenter_names List of datacenter names. Default is None.

service_instance Service instance (vim.ServiceInstance) of the vCenter. Default is None.

```
salt '*' vsphere.list_datacenters_via_proxy
```

(continues on next page)

(continued from previous page)

```
salt '*' vsphere.list_datacenters_via_proxy dc1

salt '*' vsphere.list_datacenters_via_proxy dc1,dc2

salt '*' vsphere.list_datacenters_via_proxy datacenter_names=[dc1, dc2]
```

`salt.modules.vsphere.`**`list_datastore_clusters`**(*host*, *username*, *password*, *protocol=None*, *port=None*)

Returns a list of datastore clusters for the the specified host.

host The location of the host.

username The username used to login to the host, such as `root`.

password The password used to login to the host.

protocol Optionally set to alternate protocol if the host is not using the default protocol. Default protocol is `https`.

port Optionally set to alternate port if the host is not using the default port. Default port is `443`.

CLI Example:

```
salt '*' vsphere.list_datastore_clusters 1.2.3.4 root bad-password
```

`salt.modules.vsphere.`**`list_datastores`**(*host*, *username*, *password*, *protocol=None*, *port=None*)

Returns a list of datastores for the the specified host.

host The location of the host.

username The username used to login to the host, such as `root`.

password The password used to login to the host.

protocol Optionally set to alternate protocol if the host is not using the default protocol. Default protocol is `https`.

port Optionally set to alternate port if the host is not using the default port. Default port is `443`.

CLI Example:

```
salt '*' vsphere.list_datastores 1.2.3.4 root bad-password
```

`salt.modules.vsphere.`**`list_datastores_via_proxy`**(*datastore_names=None*, *backing_disk_ids=None*, *backing_disk_scsi_addresses=None*, *service_instance=None*)

Returns a list of dict representations of the datastores visible to the proxy object. The list of datastores can be filtered by datastore names, backing disk ids (canonical names) or backing disk scsi addresses.

Supported proxy types: esxi, esxcluster, esxdatacenter

datastore_names List of the names of datastores to filter on

backing_disk_ids List of canonical names of the backing disks of the datastores to filer. Default is None.

backing_disk_scsi_addresses List of scsi addresses of the backing disks of the datastores to filter. Default is None.

service_instance Service instance (vim.ServiceInstance) of the vCenter/ESXi host. Default is None.

```
salt '*' vsphere.list_datastores_via_proxy

salt '*' vsphere.list_datastores_via_proxy datastore_names=[ds1, ds2]
```

`salt.modules.vsphere.`**`list_default_storage_policy_of_datastore`**(*datastore*, *service_instance=None*)

Returns a list of datastores assign the the storage policies.

datastore Name of the datastore to assign. The datastore needs to be visible to the VMware entity the proxy points to.

service_instance Service instance (vim.ServiceInstance) of the vCenter. Default is None.

```
salt '*' vsphere.list_default_storage_policy_of_datastore datastore=ds1
```

`salt.modules.vsphere.`**`list_default_vsan_policy`**(*service_instance=None*)

Returns the default vsan storage policy.

service_instance Service instance (vim.ServiceInstance) of the vCenter. Default is None.

```
salt '*' vsphere.list_storage_policies

salt '*' vsphere.list_storage_policy policy_names=[policy_name]
```

`salt.modules.vsphere.`**`list_disk_partitions`**(*disk_id=None, scsi_address=None, service_instance=None*)

Lists the partitions on a disk. The disk can be specified either by the canonical name, or by the scsi_address.

disk_id Canonical name of the disk. Either `disk_id` or `scsi_address` needs to be specified (`disk_id` supersedes `scsi_address`.

scsi_address` Scsi address of the disk. `disk_id` or `scsi_address` needs to be specified (`disk_id` supersedes `scsi_address`.

service_instance Service instance (vim.ServiceInstance) of the vCenter/ESXi host. Default is None.

```
salt '*' vsphere.list_disk_partitions scsi_address='vmhaba0:C0:T0:L0'

salt '*' vsphere.list_disk_partitions disk_id='naa.000000000000001'
```

`salt.modules.vsphere.`**`list_diskgroups`**(*cache_disk_ids=None, service_instance=None*)

Returns a list of disk group dict representation on an ESXi host. The list of disk groups can be filtered by the cache disks canonical names. If no filtering is applied, all disk groups are returned.

cache_disk_ids: List of cache disk canonical names of the disk groups to be retrieved. Default is None.

use_proxy_details Specify whether to use the proxy minion's details instead of the arguments

service_instance Service instance (vim.ServiceInstance) of the vCenter/ESXi host. Default is None.

```
salt '*' vsphere.list_diskgroups

salt '*' vsphere.list_diskgroups cache_disk_ids='[naa.000000000000001]'
```

`salt.modules.vsphere.`**`list_disks`**(*disk_ids=None, scsi_addresses=None, service_instance=None*)

Returns a list of dict representations of the disks in an ESXi host. The list of disks can be filtered by disk canonical names or scsi addresses.

disk_ids: List of disk canonical names to be retrieved. Default is None.

scsi_addresses List of scsi addresses of disks to be retrieved. Default is None

service_instance Service instance (vim.ServiceInstance) of the vCenter/ESXi host. Default is None.

```
salt '*' vsphere.list_disks

salt '*' vsphere.list_disks disk_ids='[naa.00, naa.001]'

salt '*' vsphere.list_disks
    scsi_addresses='[vmhba0:C0:T0:L0, vmhba1:C0:T0:L0]'
```

`salt.modules.vsphere.`**`list_dvportgroups`**(*dvs=None, portgroup_names=None, service_instance=None*)

Returns a list of distributed virtual switch portgroups. The list can be filtered by the portgroup names or by the DVS.

dvs Name of the DVS containing the portgroups. Default value is None.

portgroup_names List of portgroup names to look for. If None, all portgroups are returned. Default value is None

service_instance Service instance (vim.ServiceInstance) of the vCenter. Default is None.

```
salt '*' vsphere.list_dvporgroups

salt '*' vsphere.list_dvportgroups dvs=dvs1

salt '*' vsphere.list_dvportgroups portgroup_names=[pg1]

salt '*' vsphere.list_dvportgroups dvs=dvs1 portgroup_names=[pg1]
```

`salt.modules.vsphere.`**`list_dvs`**(*host, username, password, protocol=None, port=None*)

Returns a list of distributed virtual switches for the the specified host.

host The location of the host.

username The username used to login to the host, such as `root`.

password The password used to login to the host.

protocol Optionally set to alternate protocol if the host is not using the default protocol. Default protocol is `https`.

port Optionally set to alternate port if the host is not using the default port. Default port is `443`.

CLI Example:

```
salt '*' vsphere.list_dvs 1.2.3.4 root bad-password
```

`salt.modules.vsphere.`**`list_dvss`**(*datacenter=None, dvs_names=None, service_instance=None*)

Returns a list of distributed virtual switches (DVSs). The list can be filtered by the datacenter or DVS names.

datacenter The datacenter to look for DVSs in. Default value is None.

dvs_names List of DVS names to look for. If None, all DVSs are returned. Default value is None.

```
salt '*' vsphere.list_dvss

salt '*' vsphere.list_dvss dvs_names=[dvs1,dvs2]
```

`salt.modules.vsphere.`**`list_folders`**(*host, username, password, protocol=None, port=None*)

Returns a list of folders for the the specified host.

host The location of the host.

username The username used to login to the host, such as `root`.

password The password used to login to the host.

protocol Optionally set to alternate protocol if the host is not using the default protocol. Default protocol is `https`.

port Optionally set to alternate port if the host is not using the default port. Default port is `443`.

CLI Example:

```
salt '*' vsphere.list_folders 1.2.3.4 root bad-password
```

`salt.modules.vsphere.`**`list_hosts`**(*host, username, password, protocol=None, port=None*)

Returns a list of hosts for the the specified VMware environment.

host The location of the host.

username The username used to login to the host, such as `root`.

password The password used to login to the host.

protocol Optionally set to alternate protocol if the host is not using the default protocol. Default protocol is `https`.

port Optionally set to alternate port if the host is not using the default port. Default port is `443`.

CLI Example:

```
salt '*' vsphere.list_hosts 1.2.3.4 root bad-password
```

`salt.modules.vsphere.`**`list_hosts_via_proxy`**(*hostnames=None, datacenter=None, cluster=None, service_instance=None*)

Returns a list of hosts for the the specified VMware environment. The list of hosts can be filtered by datacenter name and/or cluster name

hostnames Hostnames to filter on.

datacenter_name Name of datacenter. Only hosts in this datacenter will be retrieved. Default is None.

cluster_name Name of cluster. Only hosts in this cluster will be retrieved. If a datacenter is not specified the first cluster with this name will be considerred. Default is None.

service_instance Service instance (vim.ServiceInstance) of the vCenter/ESXi host. Default is None.

CLI Example:

```
salt '*' vsphere.list_hosts_via_proxy

salt '*' vsphere.list_hosts_via_proxy hostnames=[esxi1.example.com]

salt '*' vsphere.list_hosts_via_proxy datacenter=dc1 cluster=cluster1
```

`salt.modules.vsphere.`**`list_licenses`**(*service_instance=None*)

Lists all licenses on a vCenter.

service_instance Service instance (vim.ServiceInstance) of the vCenter/ESXi host. Default is None.

```
salt '*' vsphere.list_licenses
```

`salt.modules.vsphere.`**`list_networks`**(*host*, *username*, *password*, *protocol=None*, *port=None*)

Returns a list of networks for the the specified host.

host The location of the host.

username The username used to login to the host, such as `root`.

password The password used to login to the host.

protocol Optionally set to alternate protocol if the host is not using the default protocol. Default protocol is `https`.

port Optionally set to alternate port if the host is not using the default port. Default port is `443`.

CLI Example:

```
salt '*' vsphere.list_networks 1.2.3.4 root bad-password
```

`salt.modules.vsphere.`**`list_non_ssds`**(*host*, *username*, *password*, *protocol=None*, *port=None*, *host_names=None*)

Returns a list of Non-SSD disks for the given host or list of host_names.

Note: In the pyVmomi StorageSystem, ScsiDisks may, or may not have an `ssd` attribute. This attribute indicates if the ScsiDisk is SSD backed. As this option is optional, if a relevant disk in the StorageSystem does not have `ssd = true`, it will end up in the `non_ssds` list here.

---

host The location of the host.

username The username used to login to the host, such as `root`.

password The password used to login to the host.

protocol Optionally set to alternate protocol if the host is not using the default protocol. Default protocol is `https`.

port Optionally set to alternate port if the host is not using the default port. Default port is `443`.

host_names List of ESXi host names. When the host, username, and password credentials are provided for a vCenter Server, the host_names argument is required to tell vCenter the hosts for which to retrieve Non-SSD disks.

If host_names is not provided, Non-SSD disks will be retrieved for the `host` location instead. This is useful for when service instance connection information is used for a single ESXi host.

CLI Example:

```
# Used for single ESXi host connection information
salt '*' vsphere.list_non_ssds my.esxi.host root bad-password

# Used for connecting to a vCenter Server
salt '*' vsphere.list_non_ssds my.vcenter.location root bad-password         host_
↪names='[esxi-1.host.com, esxi-2.host.com]'
```

salt.modules.vsphere.**list_resourcepools**(*host*, *username*, *password*, *protocol=None*, *port=None*)

Returns a list of resource pools for the the specified host.

host The location of the host.

username The username used to login to the host, such as `root`.

password The password used to login to the host.

protocol Optionally set to alternate protocol if the host is not using the default protocol. Default protocol is `https`.

port Optionally set to alternate port if the host is not using the default port. Default port is `443`.

CLI Example:

```
salt '*' vsphere.list_resourcepools 1.2.3.4 root bad-password
```

salt.modules.vsphere.**list_ssds**(*host*, *username*, *password*, *protocol=None*, *port=None*, *host_names=None*)

Returns a list of SSDs for the given host or list of host_names.

host The location of the host.

username The username used to login to the host, such as `root`.

password The password used to login to the host.

protocol Optionally set to alternate protocol if the host is not using the default protocol. Default protocol is `https`.

port Optionally set to alternate port if the host is not using the default port. Default port is `443`.

host_names List of ESXi host names. When the host, username, and password credentials are provided for a vCenter Server, the host_names argument is required to tell vCenter the hosts for which to retrieve SSDs.

If host_names is not provided, SSDs will be retrieved for the `host` location instead. This is useful for when service instance connection information is used for a single ESXi host.

CLI Example:

```
# Used for single ESXi host connection information
salt '*' vsphere.list_ssds my.esxi.host root bad-password

# Used for connecting to a vCenter Server
salt '*' vsphere.list_ssds my.vcenter.location root bad-password         host_
↪names='[esxi-1.host.com, esxi-2.host.com]'
```

salt.modules.vsphere.**list_storage_policies**(*policy_names=None*, *service_instance=None*)

Returns a list of storage policies.

policy_names Names of policies to list. If None, all policies are listed. Default is None.

service_instance Service instance (vim.ServiceInstance) of the vCenter. Default is None.

```
salt '*' vsphere.list_storage_policies

salt '*' vsphere.list_storage_policy policy_names=[policy_name]
```

salt.modules.vsphere.**list_tag_categories**(*server=None*, *username=None*, *password=None*, *service_instance=None*)

List existing categories a user has access to.

CLI Example:

```
salt vm_minion vsphere.list_tag_categories
```

Parameters
- **server** (*basestring*) – Target DNS or IP of vCenter center.
- **username** (*basestring*) – Username associated with the vCenter center.
- **password** (*basestring*) – Password associated with the vCenter center.

Returns Value(s) of category_id.

Return type list of str

`salt.modules.vsphere.`**`list_tags`**(*server=None, username=None, password=None, service_instance=None*)

List existing tags a user has access to.

CLI Example:

```
salt vm_minion vsphere.list_tags
```

Parameters
- **server** (*basestring*) – Target DNS or IP of vCenter center.
- **username** (*basestring*) – Username associated with the vCenter center.
- **password** (*basestring*) – Password associated with the vCenter center.

Returns Value(s) of tag_id.

Return type list of str

`salt.modules.vsphere.`**`list_uplink_dvportgroup`**(*dvs, service_instance=None*)

Returns the uplink portgroup of a distributed virtual switch.

dvs Name of the DVS containing the portgroup.

service_instance Service instance (vim.ServiceInstance) of the vCenter. Default is None.

```
salt '*' vsphere.list_uplink_dvportgroup dvs=dvs_name
```

`salt.modules.vsphere.`**`list_vapps`**(*host, username, password, protocol=None, port=None*)

Returns a list of vApps for the the specified host.

host The location of the host.

username The username used to login to the host, such as `root`.

password The password used to login to the host.

protocol Optionally set to alternate protocol if the host is not using the default protocol. Default protocol is `https`.

port Optionally set to alternate port if the host is not using the default port. Default port is `443`.

CLI Example:

```
# List vapps from all minions
salt '*' vsphere.list_vapps 1.2.3.4 root bad-password
```

`salt.modules.vsphere.`**`list_vms`**(*host, username, password, protocol=None, port=None*)

Returns a list of VMs for the the specified host.

host The location of the host.

username The username used to login to the host, such as `root`.

password The password used to login to the host.

protocol Optionally set to alternate protocol if the host is not using the default protocol. Default protocol is `https`.

port Optionally set to alternate port if the host is not using the default port. Default port is `443`.

CLI Example:

```
salt '*' vsphere.list_vms 1.2.3.4 root bad-password
```

`salt.modules.vsphere.`**`power_off_vm`**(*name*, *datacenter=None*, *service_instance=None*)

Powers off a virtual machine specified by it's name.

name Name of the virtual machine

datacenter Datacenter of the virtual machine

service_instance Service instance (vim.ServiceInstance) of the vCenter. Default is None.

```
salt '*' vsphere.power_off_vm name=my_vm
```

`salt.modules.vsphere.`**`power_on_vm`**(*name*, *datacenter=None*, *service_instance=None*)

Powers on a virtual machine specified by it's name.

name Name of the virtual machine

datacenter Datacenter of the virtual machine

service_instance Service instance (vim.ServiceInstance) of the vCenter. Default is None.

```
salt '*' vsphere.power_on_vm name=my_vm
```

`salt.modules.vsphere.`**`register_vm`**(*name*, *datacenter*, *placement*, *vmx_path*, *service_instance=None*)

Registers a virtual machine to the inventory with the given vmx file. Returns comments and change list

name Name of the virtual machine

datacenter Datacenter of the virtual machine

placement Placement dictionary of the virtual machine, host or cluster

vmx_path: Full path to the vmx file, datastore name should be included

service_instance Service instance (vim.ServiceInstance) of the vCenter. Default is None.

`salt.modules.vsphere.`**`remove_capacity_from_diskgroup`**(*cache_disk_id*, *capacity_disk_ids*, *data_evacuation=True*, *safety_checks=True*, *service_instance=None*)

Remove capacity disks from the disk group with the specified cache disk.

cache_disk_id The canonical name of the cache disk.

capacity_disk_ids A list containing canonical names of the capacity disks to add.

data_evacuation Specifies whether to gracefully evacuate the data on the capacity disks before removing them from the disk group. Default value is True.

safety_checks Specify whether to perform safety check or to skip the checks and try performing the required task. Default value is True.

service_instance Service instance (vim.ServiceInstance) of the vCenter/ESXi host. Default is None.

```
salt '*' vsphere.remove_capacity_from_diskgroup
    cache_disk_id='naa.000000000000001'
    capacity_disk_ids='[naa.000000000000002, naa.000000000000003]'
```

`salt.modules.vsphere.`**`remove_datastore`**(*datastore*, *service_instance=None*)

Removes a datastore. If multiple datastores an error is raised.

datastore Datastore name

service_instance Service instance (vim.ServiceInstance) of the vCenter/ESXi host. Default is None.

```
salt '*' vsphere.remove_datastore ds_name
```

`salt.modules.vsphere.`**`remove_diskgroup`**(*cache_disk_id*, *data_accessibility=True*, *service_instance=None*)

Remove the diskgroup with the specified cache disk.

cache_disk_id The canonical name of the cache disk.

data_accessibility Specifies whether to ensure data accessibility. Default value is True.

service_instance Service instance (vim.ServiceInstance) of the vCenter/ESXi host. Default is None.

```
salt '*' vsphere.remove_diskgroup cache_disk_id='naa.000000000000001'
```

salt.modules.vsphere.**remove_dvportgroup**(*portgroup, dvs, service_instance=None*)

Removes a distributed virtual portgroup.

portgroup Name of the portgroup to be removed.

dvs Name of the DVS containing the portgroups.

service_instance Service instance (vim.ServiceInstance) of the vCenter. Default is None.

```
salt '*' vsphere.remove_dvportgroup portgroup=pg1 dvs=dvs1
```

salt.modules.vsphere.**rename_datastore**(*datastore_name, new_datastore_name, service_instance=None*)

Renames a datastore. The datastore needs to be visible to the proxy.

datastore_name Current datastore name.

new_datastore_name New datastore name.

service_instance Service instance (vim.ServiceInstance) of the vCenter/ESXi host. Default is None.

```
salt '*' vsphere.rename_datastore old_name new_name
```

salt.modules.vsphere.**reset_syslog_config**(*host, username, password, protocol=None, port=None, syslog_config=None, esxi_hosts=None, credstore=None*)

Reset the syslog service to its default settings.

Valid syslog_config values are `logdir`, `loghost`, `logdir-unique`, `default-rotate`, `default-size`, `default-timeout`, or `all` for all of these.

host The location of the host.

username The username used to login to the host, such as `root`.

password The password used to login to the host.

protocol Optionally set to alternate protocol if the host is not using the default protocol. Default protocol is `https`.

port Optionally set to alternate port if the host is not using the default port. Default port is `443`.

syslog_config List of parameters to reset, provided as a comma-delimited string, or 'all' to reset all syslog configuration parameters. Required.

esxi_hosts If `host` is a vCenter host, then use esxi_hosts to execute this function on a list of one or more ESXi machines.

credstore Optionally set to path to the credential store file.

Returns Dictionary with a top-level key of 'success' which indicates if all the parameters were reset, and individual keys for each parameter indicating which succeeded or failed, per host.

CLI Example:

`syslog_config` can be passed as a quoted, comma-separated string, e.g.

```
# Used for ESXi host connection information
salt '*' vsphere.reset_syslog_config my.esxi.host root bad-password ␣
↪syslog_config='logdir,loghost'

# Used for connecting to a vCenter Server
salt '*' vsphere.reset_syslog_config my.vcenter.location root bad-password ␣
↪     syslog_config='logdir,loghost' esxi_hosts='[esxi-1.host.com, esxi-2.host.
↪com]'
```

salt.modules.vsphere.**service_restart**(*host, username, password, service_name, protocol=None, port=None, host_names=None*)

Restart the named service for the given host or list of hosts.

host The location of the host.

username The username used to login to the host, such as `root`.

password The password used to login to the host.

service_name

The name of the service for which to set the policy. Supported service names are:

- DCUI
- TSM
- SSH
- lbtd
- lsassd
- lwiod
- netlogond
- ntpd
- sfcbd-watchdog
- snmpd
- vprobed
- vpxa
- xorg

protocol Optionally set to alternate protocol if the host is not using the default protocol. Default protocol is `https`.

port Optionally set to alternate port if the host is not using the default port. Default port is `443`.

host_names List of ESXi host names. When the host, username, and password credentials are provided for a vCenter Server, the host_names argument is required to tell vCenter the hosts for which to restart the service.

If host_names is not provided, the service will be restarted for the `host` location instead. This is useful for when service instance connection information is used for a single ESXi host.

CLI Example:

```
# Used for single ESXi host connection information
salt '*' vsphere.service_restart my.esxi.host root bad-password 'ntpd'

# Used for connecting to a vCenter Server
salt '*' vsphere.service_restart my.vcenter.location root bad-password 'ntpd'
↪    host_names='[esxi-1.host.com, esxi-2.host.com]'
```

`salt.modules.vsphere.`**`service_start`**(*host, username, password, service_name, protocol=None, port=None, host_names=None*)

Start the named service for the given host or list of hosts.

host The location of the host.

username The username used to login to the host, such as `root`.

password The password used to login to the host.

service_name

The name of the service for which to set the policy. Supported service names are:

- DCUI
- TSM
- SSH
- lbtd
- lsassd
- lwiod
- netlogond
- ntpd
- sfcbd-watchdog
- snmpd

  - vprobed
  - vpxa
  - xorg

protocol Optionally set to alternate protocol if the host is not using the default protocol. Default protocol is `https`.

port Optionally set to alternate port if the host is not using the default port. Default port is `443`.

host_names List of ESXi host names. When the host, username, and password credentials are provided for a vCenter Server, the host_names argument is required to tell vCenter the hosts for which to start the service.

If host_names is not provided, the service will be started for the `host` location instead. This is useful for when service instance connection information is used for a single ESXi host.

CLI Example:

```
# Used for single ESXi host connection information
salt '*' vsphere.service_start my.esxi.host root bad-password 'ntpd'

# Used for connecting to a vCenter Server
salt '*' vsphere.service_start my.vcenter.location root bad-password 'ntpd'   ⏎
↪  host_names='[esxi-1.host.com, esxi-2.host.com]'
```

`salt.modules.vsphere.`**`service_stop`**(*host*, *username*, *password*, *service_name*, *protocol=None*, *port=None*, *host_names=None*)

Stop the named service for the given host or list of hosts.

host The location of the host.

username The username used to login to the host, such as `root`.

password The password used to login to the host.

service_name

The name of the service for which to set the policy. Supported service names are:

  - DCUI
  - TSM
  - SSH
  - lbtd
  - lsassd
  - lwiod
  - netlogond
  - ntpd
  - sfcbd-watchdog
  - snmpd
  - vprobed
  - vpxa
  - xorg

protocol Optionally set to alternate protocol if the host is not using the default protocol. Default protocol is `https`.

port Optionally set to alternate port if the host is not using the default port. Default port is `443`.

host_names List of ESXi host names. When the host, username, and password credentials are provided for a vCenter Server, the host_names argument is required to tell vCenter the hosts for which to stop the service.

If host_names is not provided, the service will be stopped for the `host` location instead. This is useful for when service instance connection information is used for a single ESXi host.

CLI Example:

```
# Used for single ESXi host connection information
salt '*' vsphere.service_stop my.esxi.host root bad-password 'ssh'
```

(continues on next page)

(continued from previous page)

```
# Used for connecting to a vCenter Server
salt '*' vsphere.service_stop my.vcenter.location root bad-password 'ssh'        ⊠
↪host_names='[esxi-1.host.com, esxi-2.host.com]'
```

`salt.modules.vsphere.`**`set_advanced_configs`**(*vm_name*, *datacenter*, *advanced_configs*, *service_instance=None*)

Appends extra config parameters to a virtual machine advanced config list

vm_name Virtual machine name

datacenter Datacenter name where the virtual machine is available

advanced_configs Dictionary with advanced parameter key value pairs

service_instance vCenter service instance for connection and configuration

`salt.modules.vsphere.`**`set_coredump_network_config`**(*host*, *username*, *password*, *dump_ip*, *protocol=None*, *port=None*, *host_vnic='vmk0'*, *dump_port=6500*, *esxi_hosts=None*, *credstore=None*)

Set the network parameters for a network coredump collection. Note that ESXi requires that the dumps first be enabled (see *coredump_network_enable*) before these parameters may be set.

host The location of the host.

username The username used to login to the host, such as `root`.

password The password used to login to the host.

dump_ip IP address of host that will accept the dump.

protocol Optionally set to alternate protocol if the host is not using the default protocol. Default protocol is `https`.

port Optionally set to alternate port if the host is not using the default port. Default port is `443`.

esxi_hosts If `host` is a vCenter host, then use esxi_hosts to execute this function on a list of one or more ESXi machines.

host_vnic Host VNic port through which to communicate. Defaults to `vmk0`.

dump_port TCP port to use for the dump, defaults to `6500`.

credstore Optionally set to path to the credential store file.

Returns A standard cmd.run_all dictionary with a *success* key added, per host. *success* will be True if the set succeeded, False otherwise.

CLI Example:

```
# Used for ESXi host connection information
salt '*' vsphere.set_coredump_network_config my.esxi.host root bad-password 'dump_
↪ip.host.com'

# Used for connecting to a vCenter Server
salt '*' vsphere.set_coredump_network_config my.vcenter.location root bad-
↪password 'dump_ip.host.com'           esxi_hosts='[esxi-1.host.com, esxi-2.
↪host.com]'
```

`salt.modules.vsphere.`**`set_ntp_config`**(*host*, *username*, *password*, *ntp_servers*, *protocol=None*, *port=None*, *host_names=None*)

Set NTP configuration for a given host of list of host_names.

host The location of the host.

username The username used to login to the host, such as `root`.

password The password used to login to the host.

ntp_servers A list of servers that should be added to and configured for the specified host's NTP configuration.

protocol Optionally set to alternate protocol if the host is not using the default protocol. Default protocol is `https`.

port Optionally set to alternate port if the host is not using the default port. Default port is `443`.

host_names List of ESXi host names. When the host, username, and password credentials are provided for a vCenter Server, the host_names argument is required to tell vCenter which hosts to configure ntp servers.

If host_names is not provided, the NTP servers will be configured for the `host` location instead. This is useful for when service instance connection information is used for a single ESXi host.

CLI Example:

```
# Used for single ESXi host connection information
salt '*' vsphere.ntp_configure my.esxi.host root bad-password '[192.174.1.100,
↪192.174.1.200]'

# Used for connecting to a vCenter Server
salt '*' vsphere.ntp_configure my.vcenter.location root bad-password '[192.174.1.
↪100, 192.174.1.200]'         host_names='[esxi-1.host.com, esxi-2.host.com]'
```

`salt.modules.vsphere.`**`set_service_policy`**(*host, username, password, service_name, service_policy, protocol=None, port=None, host_names=None*)

Set the service name's policy for a given host or list of hosts.

host The location of the host.

username The username used to login to the host, such as `root`.

password The password used to login to the host.

service_name

The name of the service for which to set the policy. Supported service names are:

- DCUI
- TSM
- SSH
- lbtd
- lsassd
- lwiod
- netlogond
- ntpd
- sfcbd-watchdog
- snmpd
- vprobed
- vpxa
- xorg

service_policy The policy to set for the service. For example, 'automatic'.

protocol Optionally set to alternate protocol if the host is not using the default protocol. Default protocol is `https`.

port Optionally set to alternate port if the host is not using the default port. Default port is `443`.

host_names List of ESXi host names. When the host, username, and password credentials are provided for a vCenter Server, the host_names argument is required to tell vCenter the hosts for which to set the service policy.

If host_names is not provided, the service policy information will be retrieved for the `host` location instead. This is useful for when service instance connection information is used for a single ESXi host.

CLI Example:

```
# Used for single ESXi host connection information
salt '*' vsphere.set_service_policy my.esxi.host root bad-password 'ntpd'
↪'automatic'

# Used for connecting to a vCenter Server
salt '*' vsphere.set_service_policy my.vcenter.location root bad-password 'ntpd'
↪'automatic'         host_names='[esxi-1.host.com, esxi-2.host.com]'
```

(continues on next page)

(continued from previous page)

`salt.modules.vsphere.`**`set_syslog_config`**(*host*, *username*, *password*, *syslog_config*, *config_value*, *protocol=None*, *port=None*, *firewall=True*, *reset_service=True*, *esxi_hosts=None*, *credstore=None*)

Set the specified syslog configuration parameter. By default, this function will reset the syslog service after the configuration is set.

host ESXi or vCenter host to connect to.

username User to connect as, usually root.

password Password to connect with.

syslog_config Name of parameter to set (corresponds to the command line switch for esxcli without the double dashes (–))

Valid syslog_config values are `logdir`, `loghost`, `default-rotate`, ``default-size`, `default-timeout`, and `logdir-unique`.

config_value Value for the above parameter. For `loghost`, URLs or IP addresses to use for logging. Multiple log servers can be specified by listing them, comma-separated, but without spaces before or after commas.

(reference: https://blogs.vmware.com/vsphere/2012/04/configuring-multiple-syslog-servers-for-esxi-5.html)

protocol Optionally set to alternate protocol if the host is not using the default protocol. Default protocol is `https`.

port Optionally set to alternate port if the host is not using the default port. Default port is `443`.

firewall Enable the firewall rule set for syslog. Defaults to `True`.

reset_service After a successful parameter set, reset the service. Defaults to `True`.

esxi_hosts If `host` is a vCenter host, then use esxi_hosts to execute this function on a list of one or more ESXi machines.

credstore Optionally set to path to the credential store file.

Returns Dictionary with a top-level key of 'success' which indicates if all the parameters were reset, and individual keys for each parameter indicating which succeeded or failed, per host.

CLI Example:

```
# Used for ESXi host connection information
salt '*' vsphere.set_syslog_config my.esxi.host root bad-password
↪loghost ssl://localhost:5432,tcp://10.1.0.1:1514

# Used for connecting to a vCenter Server
salt '*' vsphere.set_syslog_config my.vcenter.location root bad-password
↪   loghost ssl://localhost:5432,tcp://10.1.0.1:1514          esxi_hosts=
↪'[esxi-1.host.com, esxi-2.host.com]'
```

`salt.modules.vsphere.`**`supports_proxies`**(**proxy_types*)

Decorator to specify which proxy types are supported by a function

proxy_types: Arbitrary list of strings with the supported types of proxies

`salt.modules.vsphere.`**`syslog_service_reload`**(*host*, *username*, *password*, *protocol=None*, *port=None*, *esxi_hosts=None*, *credstore=None*)

Reload the syslog service so it will pick up any changes.

host The location of the host.

username The username used to login to the host, such as `root`.

password The password used to login to the host.

protocol Optionally set to alternate protocol if the host is not using the default protocol. Default protocol is `https`.

port Optionally set to alternate port if the host is not using the default port. Default port is `443`.

esxi_hosts If `host` is a vCenter host, then use esxi_hosts to execute this function on a list of one or more ESXi machines.

credstore Optionally set to path to the credential store file.

> **Returns** A standard cmd.run_all dictionary. This dictionary will at least have a *retcode* key. If *retcode* is 0 the command was successful.

CLI Example:

```
# Used for ESXi host connection information
salt '*' vsphere.syslog_service_reload my.esxi.host root bad-password

# Used for connecting to a vCenter Server
salt '*' vsphere.syslog_service_reload my.vcenter.location root bad-password     ⤶
↪       esxi_hosts='[esxi-1.host.com, esxi-2.host.com]'
```

`salt.modules.vsphere.`**`system_info`**(*host*, *username*, *password*, *protocol=None*, *port=None*)

Return system information about a VMware environment.

host The location of the host.

username The username used to login to the host, such as `root`.

password The password used to login to the host.

protocol Optionally set to alternate protocol if the host is not using the default protocol. Default protocol is `https`.

port Optionally set to alternate port if the host is not using the default port. Default port is `443`.

CLI Example:

```
salt '*' vsphere.system_info 1.2.3.4 root bad-password
```

`salt.modules.vsphere.`**`test_vcenter_connection`**(*service_instance=None*)

Checks if a connection is to a vCenter

CLI Example:

```
salt '*' vsphere.test_vcenter_connection
```

`salt.modules.vsphere.`**`unregister_vm`**(*name*, *datacenter*, *placement=None*, *power_off=False*, *service_instance=None*)

Unregisters a virtual machine defined by name and placement

name Name of the virtual machine

datacenter Datacenter of the virtual machine

placement Placement information of the virtual machine

service_instance vCenter service instance for connection and configuration

```
salt '*' vsphere.unregister_vm name=my_vm datacenter=my_datacenter
```

`salt.modules.vsphere.`**`update_cluster`**(*cluster_dict*, *datacenter=None*, *cluster=None*, *service_instance=None*)

Updates a cluster.

config_dict Dictionary with the config values of the new cluster.

datacenter Name of datacenter containing the cluster. Ignored if already contained by proxy details. Default value is None.

cluster Name of cluster. Ignored if already contained by proxy details. Default value is None.

service_instance Service instance (vim.ServiceInstance) of the vCenter. Default is None.

```
# esxdatacenter proxy
salt '*' vsphere.update_cluster cluster_dict=$cluster_dict cluster=cl1
```

(continues on next page)

(continued from previous page)

```
# esxcluster proxy
salt '*' vsphere.update_cluster cluster_dict=$cluster_dict
```

`salt.modules.vsphere.`**`update_dvportgroup`**(*portgroup_dict, portgroup, dvs, service_instance=True*)

Updates a distributed virtual portgroup.

portgroup_dict Dictionary with the values the portgroup should be update with (example in salt.states.dvs).

portgroup Name of the portgroup to be updated.

dvs Name of the DVS containing the portgroups.

service_instance Service instance (vim.ServiceInstance) of the vCenter. Default is None.

```
salt '*' vsphere.update_dvportgroup portgroup_dict=<dict>
    portgroup=pg1

salt '*' vsphere.update_dvportgroup portgroup_dict=<dict>
    portgroup=pg1 dvs=dvs1
```

`salt.modules.vsphere.`**`update_dvs`**(*dvs_dict, dvs, service_instance=None*)

Updates a distributed virtual switch (DVS).

Note: Updating the product info, capability, uplinks of a DVS is not supported so the corresponding entries in `dvs_dict` will be ignored.

dvs_dict Dictionary with the values the DVS should be update with (example in salt.states.dvs)

dvs Name of the DVS to be updated.

service_instance Service instance (vim.ServiceInstance) of the vCenter. Default is None.

```
salt '*' vsphere.update_dvs dvs_dict=$dvs_dict dvs=dvs1
```

`salt.modules.vsphere.`**`update_host_datetime`**(*host, username, password, protocol=None, port=None, host_names=None*)

Update the date/time on the given host or list of host_names. This function should be used with caution since network delays and execution delays can result in time skews.

host The location of the host.

username The username used to login to the host, such as `root`.

password The password used to login to the host.

protocol Optionally set to alternate protocol if the host is not using the default protocol. Default protocol is `https`.

port Optionally set to alternate port if the host is not using the default port. Default port is `443`.

host_names List of ESXi host names. When the host, username, and password credentials are provided for a vCenter Server, the host_names argument is required to tell vCenter which hosts should update their date/time.

If host_names is not provided, the date/time will be updated for the `host` location instead. This is useful for when service instance connection information is used for a single ESXi host.

CLI Example:

```
# Used for single ESXi host connection information
salt '*' vsphere.update_date_time my.esxi.host root bad-password

# Used for connecting to a vCenter Server
salt '*' vsphere.update_date_time my.vcenter.location root bad-password
↪host_names='[esxi-1.host.com, esxi-2.host.com]'
```

`salt.modules.vsphere.`**`update_host_password`**(*host, username, password, new_password, protocol=None, port=None*)

Update the password for a given host.

Note: Currently only works with connections to ESXi hosts. Does not work with vCenter servers.

---

host The location of the ESXi host.
username The username used to login to the ESXi host, such as `root`.
password The password used to login to the ESXi host.
new_password The new password that will be updated for the provided username on the ESXi host.
protocol Optionally set to alternate protocol if the host is not using the default protocol. Default protocol is `https`.
port Optionally set to alternate port if the host is not using the default port. Default port is `443`.

CLI Example:

```
salt '*' vsphere.update_host_password my.esxi.host root original-bad-password new-
↪bad-password
```

`salt.modules.vsphere.`**`update_storage_policy`**(*policy*, *policy_dict*, *service_instance=None*)

Updates a storage policy.

Supported capability types: scalar, set, range.
policy Name of the policy to update.
policy_dict Dictionary containing the changes to apply to the policy. (example in salt.states.pbm)
service_instance Service instance (vim.ServiceInstance) of the vCenter. Default is None.

```
salt '*' vsphere.update_storage_policy policy='policy name'
    policy_dict="$policy_dict"
```

`salt.modules.vsphere.`**`update_vm`**(*vm_name*, *cpu=None*, *memory=None*, *image=None*, *version=None*, *interfaces=None*, *disks=None*, *scsi_devices=None*, *serial_ports=None*, *datacenter=None*, *datastore=None*, *cd_dvd_drives=None*, *sata_controllers=None*, *advanced_configs=None*, *service_instance=None*)

Updates the configuration of the virtual machine if the config differs
vm_name Virtual Machine name to be updated
cpu CPU configuration options
memory Memory configuration options
version Virtual machine container hardware version
image Virtual machine guest OS version identifier VirtualMachineGuestOsIdentifier
interfaces Network interfaces configuration options
disks Disks configuration options
scsi_devices SCSI devices configuration options
serial_ports Serial ports configuration options
datacenter Datacenter where the virtual machine is available
datastore Datastore where the virtual machine config files are available
cd_dvd_drives CD/DVD drives configuration options
advanced_config Advanced config parameters to be set for the virtual machine
service_instance vCenter service instance for connection and configuration

`salt.modules.vsphere.`**`upload_ssh_key`**(*host*, *username*, *password*, *ssh_key=None*, *ssh_key_file=None*, *protocol=None*, *port=None*, *certificate_verify=False*)

Upload an ssh key for root to an ESXi host via http PUT. This function only works for ESXi, not vCenter. Only one ssh key can be uploaded for root. Uploading a second key will replace any existing key.

Parameters

- **`host`** – The location of the ESXi Host
- **`username`** – Username to connect as

- **`password`** – Password for the ESXi web endpoint
- **`ssh_key`** – Public SSH key, will be added to authorized_keys on ESXi
- **`ssh_key_file`** – File containing the SSH key. Use 'ssh_key' or ssh_key_file, but not both.
- **`protocol`** – defaults to https, can be http if ssl is disabled on ESXi
- **`port`** – defaults to 443 for https
- **`certificate_verify`** – If true require that the SSL connection present a valid certificate

Returns Dictionary with a 'status' key, True if upload is successful. If upload is unsuccessful, 'status' key will be False and an 'Error' key will have an informative message.

CLI Example:

```
salt '*' vsphere.upload_ssh_key my.esxi.host root bad-password ssh_key_file='/etc/
↪salt/my_keys/my_key.pub'
```

`salt.modules.vsphere.`**`vmotion_disable`**(*host, username, password, protocol=None, port=None, host_names=None*)

Disable vMotion for a given host or list of host_names.

host The location of the host.

username The username used to login to the host, such as `root`.

password The password used to login to the host.

protocol Optionally set to alternate protocol if the host is not using the default protocol. Default protocol is `https`.

port Optionally set to alternate port if the host is not using the default port. Default port is `443`.

host_names List of ESXi host names. When the host, username, and password credentials are provided for a vCenter Server, the host_names argument is required to tell vCenter which hosts should disable VMotion.

If host_names is not provided, VMotion will be disabled for the `host` location instead. This is useful for when service instance connection information is used for a single ESXi host.

CLI Example:

```
# Used for single ESXi host connection information
salt '*' vsphere.vmotion_disable my.esxi.host root bad-password

# Used for connecting to a vCenter Server
salt '*' vsphere.vmotion_disable my.vcenter.location root bad-password
↪host_names='[esxi-1.host.com, esxi-2.host.com]'
```

`salt.modules.vsphere.`**`vmotion_enable`**(*host, username, password, protocol=None, port=None, host_names=None, device='vmk0'*)

Enable vMotion for a given host or list of host_names.

host The location of the host.

username The username used to login to the host, such as `root`.

password The password used to login to the host.

protocol Optionally set to alternate protocol if the host is not using the default protocol. Default protocol is `https`.

port Optionally set to alternate port if the host is not using the default port. Default port is `443`.

host_names List of ESXi host names. When the host, username, and password credentials are provided for a vCenter Server, the host_names argument is required to tell vCenter which hosts should enable VMotion.

If host_names is not provided, VMotion will be enabled for the `host` location instead. This is useful for when service instance connection information is used for a single ESXi host.

device The device that uniquely identifies the VirtualNic that will be used for VMotion for each host. Defaults to `vmk0`.

CLI Example:

```
# Used for single ESXi host connection information
salt '*' vsphere.vmotion_enable my.esxi.host root bad-password

# Used for connecting to a vCenter Server
salt '*' vsphere.vmotion_enable my.vcenter.location root bad-password
↪host_names='[esxi-1.host.com, esxi-2.host.com]'
```

salt.modules.vsphere.**vsan_add_disks**(*host*, *username*, *password*, *protocol=None*, *port=None*, *host_names=None*)

Add any VSAN-eligible disks to the VSAN System for the given host or list of host_names.

host The location of the host.

username The username used to login to the host, such as `root`.

password The password used to login to the host.

protocol Optionally set to alternate protocol if the host is not using the default protocol. Default protocol is `https`.

port Optionally set to alternate port if the host is not using the default port. Default port is `443`.

host_names List of ESXi host names. When the host, username, and password credentials are provided for a vCenter Server, the host_names argument is required to tell vCenter which hosts need to add any VSAN-eligible disks to the host's VSAN system.

If host_names is not provided, VSAN-eligible disks will be added to the hosts's VSAN system for the `host` location instead. This is useful for when service instance connection information is used for a single ESXi host.

CLI Example:

```
# Used for single ESXi host connection information
salt '*' vsphere.vsan_add_disks my.esxi.host root bad-password

# Used for connecting to a vCenter Server
salt '*' vsphere.vsan_add_disks my.vcenter.location root bad-password
↪host_names='[esxi-1.host.com, esxi-2.host.com]'
```

salt.modules.vsphere.**vsan_disable**(*host*, *username*, *password*, *protocol=None*, *port=None*, *host_names=None*)

Disable VSAN for a given host or list of host_names.

host The location of the host.

username The username used to login to the host, such as `root`.

password The password used to login to the host.

protocol Optionally set to alternate protocol if the host is not using the default protocol. Default protocol is `https`.

port Optionally set to alternate port if the host is not using the default port. Default port is `443`.

host_names List of ESXi host names. When the host, username, and password credentials are provided for a vCenter Server, the host_names argument is required to tell vCenter which hosts should disable VSAN.

If host_names is not provided, VSAN will be disabled for the `host` location instead. This is useful for when service instance connection information is used for a single ESXi host.

CLI Example:

```
# Used for single ESXi host connection information
salt '*' vsphere.vsan_disable my.esxi.host root bad-password

# Used for connecting to a vCenter Server
salt '*' vsphere.vsan_disable my.vcenter.location root bad-password        host_
↪names='[esxi-1.host.com, esxi-2.host.com]'
```

salt.modules.vsphere.**vsan_enable**(*host*, *username*, *password*, *protocol=None*, *port=None*, *host_names=None*)

Enable VSAN for a given host or list of host_names.

host The location of the host.

username The username used to login to the host, such as `root`.

password The password used to login to the host.

protocol Optionally set to alternate protocol if the host is not using the default protocol. Default protocol is `https`.

port Optionally set to alternate port if the host is not using the default port. Default port is `443`.

host_names List of ESXi host names. When the host, username, and password credentials are provided for a vCenter Server, the host_names argument is required to tell vCenter which hosts should enable VSAN.

If host_names is not provided, VSAN will be enabled for the `host` location instead. This is useful for when service instance connection information is used for a single ESXi host.

CLI Example:

```
# Used for single ESXi host connection information
salt '*' vsphere.vsan_enable my.esxi.host root bad-password

# Used for connecting to a vCenter Server
salt '*' vsphere.vsan_enable my.vcenter.location root bad-password        host_
↪names='[esxi-1.host.com, esxi-2.host.com]'
```

21.9.453 salt.modules.webutil

Support for htpasswd command. Requires the apache2-utils package for Debian-based distros.

New in version 2014.1.0.

The functions here will load inside the webutil module. This allows other functions that don't use htpasswd to use the webutil module name.

`salt.modules.webutil.`**`useradd`**(*pwfile, user, password, opts=", runas=None*)

Add a user to htpasswd file using the htpasswd command. If the htpasswd file does not exist, it will be created.

pwfile Path to htpasswd file

user User name

password User password

opts Valid options that can be passed are:

- *n* Don't update file; display results on stdout.
- *m* Force MD5 encryption of the password (default).
- *d* Force CRYPT encryption of the password.
- *p* Do not encrypt the password (plaintext).
- *s* Force SHA encryption of the password.

runas The system user to run htpasswd command with

CLI Examples:

```
salt '*' webutil.useradd /etc/httpd/htpasswd larry badpassword
salt '*' webutil.useradd /etc/httpd/htpasswd larry badpass opts=ns
```

`salt.modules.webutil.`**`userdel`**(*pwfile, user, runas=None, all_results=False*)

Delete a user from the specified htpasswd file.

pwfile Path to htpasswd file

user User name

runas The system user to run htpasswd command with

all_results Return stdout, stderr, and retcode, not just stdout

CLI Examples:

```
salt '*' webutil.userdel /etc/httpd/htpasswd larry
```

`salt.modules.webutil.`**`verify`**`(`*pwfile, user, password, opts='', runas=None*`)`

Return True if the htpasswd file exists, the user has an entry, and their password matches.

pwfile Fully qualified path to htpasswd file

user User name

password User password

opts Valid options that can be passed are:

- *m* Force MD5 encryption of the password (default).
- *d* Force CRYPT encryption of the password.
- *p* Do not encrypt the password (plaintext).
- *s* Force SHA encryption of the password.

runas The system user to run htpasswd command with

CLI Examples:

```
salt '*' webutil.verify /etc/httpd/htpasswd larry maybepassword
salt '*' webutil.verify /etc/httpd/htpasswd larry maybepassword opts=ns
```

21.9.454 salt.modules.win_autoruns

Module for listing programs that automatically run on startup (very alpha...not tested on anything but my Win 7x64)

`salt.modules.win_autoruns.`**`list_`**`()`

Get a list of automatically running programs

CLI Example:

```
salt '*' autoruns.list
```

21.9.455 salt.modules.win_certutil module

This module allows you to install certificates into the windows certificate manager.

```
salt '*' certutil.add_store salt://cert.cer "TrustedPublisher"
```

`salt.modules.win_certutil.`**`add_store`**`(`*source, store, saltenv='base'*`)`

Add the given cert into the given Certificate Store

source The source certificate file this can be in the form salt://path/to/file

store The certificate store to add the certificate to

saltenv The salt environment to use this is ignored if the path is local

CLI Example:

```
salt '*' certutil.add_store salt://cert.cer TrustedPublisher
```

`salt.modules.win_certutil.`**`del_store`**`(`*source, store, saltenv='base'*`)`

Delete the given cert into the given Certificate Store

source The source certificate file this can be in the form salt://path/to/file

store The certificate store to delete the certificate from

saltenv The salt environment to use this is ignored if the path is local

CLI Example:

```
salt '*' certutil.del_store salt://cert.cer TrustedPublisher
```

`salt.modules.win_certutil.`**`get_cert_serial`**(*cert_file*)

Get the serial number of a certificate file

cert_file The certificate file to find the serial for

CLI Example:

```
salt '*' certutil.get_cert_serial <certificate name>
```

`salt.modules.win_certutil.`**`get_stored_cert_serials`**(*store*)

Get all of the certificate serials in the specified store

store The store to get all the certificate serials from

CLI Example:

```
salt '*' certutil.get_stored_cert_serials <store>
```

21.9.456 salt.modules.win_dacl

Manage DACLs on Windows

depends

- winreg Python module

`salt.modules.win_dacl.`**`add_ace`**(*path*, *objectType*, *user*, *permission*, *acetype*, *propagation*)

add an ace to an object

path: path to the object (i.e. c:\temp\file, HKEY_LOCAL_MACHINE\SOFTWARE\KEY, etc) user: user to add permission: permissions for the user acetype: either allow/deny for each user/permission (ALLOW, DENY) propagation: how the ACE applies to children for Registry Keys and Directories(KEY, KEY&SUBKEYS, SUBKEYS)

CLI Example:

```
allow domain\fakeuser full control on HKLM\\SOFTWARE\\somekey, propagate to this
↪key and subkeys
    salt 'myminion' win_dacl.add_ace 'HKEY_LOCAL_MACHINE\\SOFTWARE\\somekey'
↪'Registry' 'domain\fakeuser' 'FULLCONTROL' 'ALLOW' 'KEY&SUBKEYS'
```

`salt.modules.win_dacl.`**`check_ace`**(*path*, *objectType*, *user*, *permission=None*, *acetype=None*, *propagation=None*, *exactPermissionMatch=False*)

Checks a path to verify the ACE (access control entry) specified exists

Parameters

- **`path`** – path to the file/reg key
- **`objectType`** – The type of object (FILE, DIRECTORY, REGISTRY)
- **`user`** – user that the ACL is for
- **`permission`** – permission to test for (READ, FULLCONTROL, etc)
- **`acetype`** – the type of ACE (ALLOW or DENY)
- **`propagation`** – the propagation type of the ACE (FILES, FOLDERS, KEY, KEY&SUBKEYS, SUBKEYS, etc)
- **`exactPermissionMatch`** – the ACL must match exactly, IE if READ is specified, the user must have READ exactly and not FULLCONTROL (which also has the READ permission obviously)

Returns (dict): 'Exists' true if the ACE exists, false if it does not

CLI Example:

```
salt 'minion-id' win_dacl.check_ace c:  emp directory <username> fullcontrol
```

`salt.modules.win_dacl.`**`check_inheritance`**(*path*, *objectType*, *user=None*)

Check a specified path to verify if inheritance is enabled

Parameters

- **path** – path of the registry key or file system object to check
- **objectType** – The type of object (FILE, DIRECTORY, REGISTRY)
- **user** – if provided, will consider only the ACEs for that user

Returns (bool): 'Inheritance' of True/False

CLI Example:

```
salt 'minion-id' win_dacl.check_inheritance c:  emp directory <username>
```

class `salt.modules.win_dacl.`**`daclConstants`**

DACL constants used throughout the module

`getAceTypeBit`(*t*)

returns the acetype bit of a text value

`getAceTypeText`(*t*)

returns the textual representation of a acetype bit

`getObjectTypeBit`(*t*)

returns the bit value of the string object type

`getPermissionBit`(*t*, *m*)

returns a permission bit of the string permission value for the specified object type

`getPermissionText`(*t*, *m*)

returns the permission textual representation of a specified permission bit/object type

`getPropagationBit`(*t*, *p*)

returns the propagation bit of a text value

`getPropagationText`(*t*, *p*)

returns the textual representation of a propagation bit

`getSecurityHkey`(*s*)

returns the necessary string value for an HKEY for the win32security module

`processPath`(*path*, *objectType*)

processes a path/object type combo and returns: registry types with the correct HKEY text representation files/directories with environment variables expanded

`salt.modules.win_dacl.`**`disable_inheritance`**(*path*, *objectType*, *copy=True*)

Disable inheritance on an object

Parameters

- **path** – The path to the object
- **objectType** – The type of object (FILE, DIRECTORY, REGISTRY)
- **copy** – True will copy the Inherited ACEs to the DACL before disabling inheritance

Returns (dict): A dictionary containing the results

CLI Example:

```
salt 'minion-id' win_dacl.disable_inheritance c:        emp directory
```

`salt.modules.win_dacl.`**`enable_inheritance`**(*path*, *objectType*, *clear=False*)

enable/disable inheritance on an object

Parameters

- **path** – The path to the object
- **objectType** – The type of object (FILE, DIRECTORY, REGISTRY)
- **clear** – True will remove non-Inherited ACEs from the ACL

Returns (dict): A dictionary containing the results

CLI Example:

```
salt 'minion-id' win_dacl.enable_inheritance c: emp directory
```

`salt.modules.win_dacl.`**`get`**(*path*, *objectType*, *user=None*)

Get the ACL of an object. Will filter by user if one is provided.

Parameters

- **path** – The path to the object
- **objectType** – The type of object (FILE, DIRECTORY, REGISTRY)
- **user** – A user name to filter by

Returns (dict): A dictionary containing the ACL

CLI Example:

```
salt 'minion-id' win_dacl.get c:        emp directory
```

`salt.modules.win_dacl.`**`rm_ace`**(*path*, *objectType*, *user*, *permission=None*, *acetype=None*, *propagation=None*)

remove an ace to an object

path: path to the object (i.e. c:\temp\file, HKEY_LOCAL_MACHINE\SOFTWARE\KEY, etc) user: user to remove permission: permissions for the user acetypes: either allow/deny for each user/permission (ALLOW, DENY) propagation: how the ACE applies to children for Registry Keys and Directories(KEY, KEY&SUBKEYS, SUBKEYS)

If any of the optional parameters are omitted (or set to None) they act as wildcards.

CLI Example:

```
remove allow domain\fakeuser full control on HKLM\\SOFTWARE\\somekey propagated
↪to this key and subkeys
    salt 'myminion' win_dacl.rm_ace 'Registry' 'HKEY_LOCAL_
↪MACHINE\\SOFTWARE\\somekey' 'domain\fakeuser' 'FULLCONTROL' 'ALLOW' 'KEY&SUBKEYS
↪'
```

21.9.457 salt.modules.win_disk

Module for gathering disk information on Windows

depends

- win32api Python module

`salt.modules.win_disk.`**`usage`**()

Return usage information for volumes mounted on this minion

CLI Example:

```
salt '*' disk.usage
```

21.9.458 salt.modules.win_dism module

Install features/packages for Windows using DISM, which is useful for minions not running server versions of Windows. Some functions are only available on Windows 10.

`salt.modules.win_dism.`**`add_capability`**(*capability, source=None, limit_access=False, image=None, restart=False*)

Install a capability

Parameters

- **`capability`** (`str`) – The capability to install
- **`source`** (`Optional[str]`) – The optional source of the capability. Default is set by group policy and can be Windows Update.
- **`limit_access`** (`Optional[bool]`) – Prevent DISM from contacting Windows Update for the source package
- **`image`** (`Optional[str]`) – The path to the root directory of an offline Windows image. If *None* is passed, the running operating system is targeted. Default is None.
- **`restart`** (`Optional[bool]`) – Reboot the machine if required by the install

Raises

- **`NotImplementedError`** – For all versions of Windows that are not Windows 10
- **`and later. Server editions of Windows use ServerManager instead.`** –

Returns A dictionary containing the results of the command

Return type dict

CLI Example:

```
salt '*' dism.add_capability Tools.Graphics.DirectX~~~~0.0.1.0
```

`salt.modules.win_dism.`**`add_feature`**(*feature, package=None, source=None, limit_access=False, enable_parent=False, image=None, restart=False*)

Install a feature using DISM

Parameters

- **`feature`** (`str`) – The feature to install
- **`package`** (`Optional[str]`) – The parent package for the feature. You do not have to specify the package if it is the Windows Foundation Package. Otherwise, use package to specify the parent package of the feature
- **`source`** (`Optional[str]`) – The optional source of the capability. Default is set by group policy and can be Windows Update
- **`limit_access`** (`Optional[bool]`) – Prevent DISM from contacting Windows Update for the source package
- **`enable_parent`** (`Optional[bool]`) – True will enable all parent features of the specified feature
- **`image`** (`Optional[str]`) – The path to the root directory of an offline Windows image. If *None* is passed, the running operating system is targeted. Default is None.
- **`restart`** (`Optional[bool]`) – Reboot the machine if required by the install

Returns A dictionary containing the results of the command

Return type dict

CLI Example:

```
salt '*' dism.add_feature NetFx3
```

`salt.modules.win_dism.`**`add_package`**(*package, ignore_check=False, prevent_pending=False, image=None, restart=False*)

Install a package using DISM

Parameters

- **`package`** (`str`) – The package to install. Can be a .cab file, a .msu file, or a folder

---

Note: An *.msu* package is supported only when the target image is offline, either mounted or applied.

- **ignore_check** (`Optional[bool]`) – Skip installation of the package if the applicability checks fail
- **prevent_pending** (`Optional[bool]`) – Skip the installation of the package if there are pending online actions
- **image** (`Optional[str]`) – The path to the root directory of an offline Windows image. If `None` is passed, the running operating system is targeted. Default is None.
- **restart** (`Optional[bool]`) – Reboot the machine if required by the install

Returns A dictionary containing the results of the command

Return type dict

CLI Example:

```
salt '*' dism.add_package C:\Packages\package.cab
```

`salt.modules.win_dism.`**`available_capabilities`**`(`*image=None*`)`

List the capabilities available on the system

Parameters **`image`** (`Optional[str]`) – The path to the root directory of an offline Windows image. If *None* is passed, the running operating system is targeted. Default is None.

Raises

- **`NotImplementedError`** – For all versions of Windows that are not Windows 10
- **`and later. Server editions of Windows use ServerManager instead.`** –

Returns A list of available capabilities

Return type list

CLI Example:

```
salt '*' dism.installed_capabilities
```

`salt.modules.win_dism.`**`available_features`**`(`*image=None*`)`

List the features available on the system

Parameters **`image`** (`Optional[str]`) – The path to the root directory of an offline Windows image. If *None* is passed, the running operating system is targeted. Default is None.

Returns A list of available features

Return type list

CLI Example:

```
salt '*' dism.available_features
```

`salt.modules.win_dism.`**`get_capabilities`**`(`*image=None*`)`

List all capabilities on the system

Parameters **`image`** (`Optional[str]`) – The path to the root directory of an offline Windows image. If *None* is passed, the running operating system is targeted. Default is None.

Raises

- **`NotImplementedError`** – For all versions of Windows that are not Windows 10
- **`and later. Server editions of Windows use ServerManager instead.`** –

Returns A list of capabilities

Return type list

CLI Example:

```
salt '*' dism.get_capabilities
```

`salt.modules.win_dism.`**`get_features`**(*package=None*, *image=None*)

List features on the system or in a package

Parameters

- **`package`** (`Optional[str]`) – The full path to the package. Can be either a .cab file or a folder. Should point to the original source of the package, not to where the file is installed. You cannot use this command to get package information for .msu files

  This can also be the name of a package as listed in `dism.installed_packages`
- **`image`** (`Optional[str]`) – The path to the root directory of an offline Windows image. If *None* is passed, the running operating system is targeted. Default is None.

Returns A list of features

Return type list

CLI Example:

```
# Return all features on the system
salt '*' dism.get_features

# Return all features in package.cab
salt '*' dism.get_features C:\packages\package.cab

# Return all features in the calc package
salt '*' dism.get_features Microsoft.Windows.Calc.Demo~6595b6144ccf1df~
↪x86~en~1.0.0.0
```

`salt.modules.win_dism.`**`installed_capabilities`**(*image=None*)

List the capabilities installed on the system

Parameters **`image`** (`Optional[str]`) – The path to the root directory of an offline Windows image. If *None* is passed, the running operating system is targeted. Default is None.

Raises

- **`NotImplementedError`** – For all versions of Windows that are not Windows 10
- **`and later. Server editions of Windows use ServerManager instead.`** –

Returns A list of installed capabilities

Return type list

CLI Example:

```
salt '*' dism.installed_capabilities
```

`salt.modules.win_dism.`**`installed_features`**(*image=None*)

List the features installed on the system

Parameters **`image`** (`Optional[str]`) – The path to the root directory of an offline Windows image. If *None* is passed, the running operating system is targeted. Default is None.

Returns A list of installed features

Return type list

CLI Example:

```
salt '*' dism.installed_features
```

`salt.modules.win_dism.`**`installed_packages`**(*image=None*)

List the packages installed on the system

Parameters **`image`** (`Optional[str]`) – The path to the root directory of an offline Windows image. If *None* is passed, the running operating system is targeted. Default is None.

Returns A list of installed packages

Return type list

CLI Example:

```
salt '*' dism.installed_packages
```

`salt.modules.win_dism.`**`package_info`**(*package*, *image=None*)

Display information about a package

Parameters

- **`package`** (*`str`*) – The full path to the package. Can be either a .cab file or a folder. Should point to the original source of the package, not to where the file is installed. You cannot use this command to get package information for .msu files
- **`image`** (*`Optional[str]`*) – The path to the root directory of an offline Windows image. If *None* is passed, the running operating system is targeted. Default is None.

Returns A dictionary containing the results of the command

Return type dict

CLI Example:

```
salt '*' dism. package_info C:\packages\package.cab
```

`salt.modules.win_dism.`**`remove_capability`**(*capability*, *image=None*, *restart=False*)

Uninstall a capability

Parameters

- **`capability`** (*`str`*) – The capability to be removed
- **`image`** (*`Optional[str]`*) – The path to the root directory of an offline Windows image. If *None* is passed, the running operating system is targeted. Default is None.
- **`restart`** (*`Optional[bool]`*) – Reboot the machine if required by the install

Raises

- **`NotImplementedError`** – For all versions of Windows that are not Windows 10
- **`and later. Server editions of Windows use ServerManager instead.`** –

Returns A dictionary containing the results of the command

Return type dict

CLI Example:

```
salt '*' dism.remove_capability Tools.Graphics.DirectX~~~~0.0.1.0
```

`salt.modules.win_dism.`**`remove_feature`**(*feature*, *remove_payload=False*, *image=None*, *restart=False*)

Disables the feature.

Parameters

- **`feature`** (*`str`*) – The feature to uninstall
- **`remove_payload`** (*`Optional[bool]`*) – Remove the feature's payload. Must supply source when enabling in the future.
- **`image`** (*`Optional[str]`*) – The path to the root directory of an offline Windows image. If *None* is passed, the running operating system is targeted. Default is None.
- **`restart`** (*`Optional[bool]`*) – Reboot the machine if required by the install

Returns A dictionary containing the results of the command

Return type dict

CLI Example:

```
salt '*' dism.remove_feature NetFx3
```

`salt.modules.win_dism.`**`remove_package`**(*package*, *image=None*, *restart=False*)

Uninstall a package

Parameters

- **`package`** (*`str`*) – The full path to the package. Can be either a .cab file or a folder. Should point to the original source of the package, not to where the file is installed. This can also be the name of a package as listed in `dism.installed_packages`

- **`image`** (`Optional[str]`) – The path to the root directory of an offline Windows image. If *None* is passed, the running operating system is targeted. Default is None.
- **`restart`** (`Optional[bool]`) – Reboot the machine if required by the install

Returns A dictionary containing the results of the command

Return type dict

CLI Example:

```
# Remove the Calc Package
salt '*' dism.remove_package Microsoft.Windows.Calc.Demo~6595b6144ccf1df~x86~en~1.
↪0.0.0

# Remove the package.cab (does not remove C:\packages\package.cab)
salt '*' dism.remove_package C:\packages\package.cab
```

21.9.459 salt.modules.win_dns_client

Module for configuring DNS Client on Windows systems

`salt.modules.win_dns_client.`**`add_dns`**(*ip*, *interface='Local Area Connection'*, *index=1*)

Add the DNS server to the network interface (index starts from 1)

Note: if the interface DNS is configured by DHCP, all the DNS servers will be removed from the interface and the requested DNS will be the only one

CLI Example:

```
salt '*' win_dns_client.add_dns <ip> <interface> <index>
```

`salt.modules.win_dns_client.`**`dns_dhcp`**(*interface='Local Area Connection'*)

Configure the interface to get its DNS servers from the DHCP server

CLI Example:

```
salt '*' win_dns_client.dns_dhcp <interface>
```

`salt.modules.win_dns_client.`**`get_dns_config`**(*interface='Local Area Connection'*)

Get the type of DNS configuration (dhcp / static).

Parameters

- **`interface`** (`str`) – The name of the network interface. This is the
- **`in the Network Connection Details for the device`** (`Description`) –

Returns `True` if DNS is configured, otherwise `False`

Return type bool

CLI Example:

```
salt '*' win_dns_client.get_dns_config 'Local Area Connection'
```

`salt.modules.win_dns_client.`**`get_dns_servers`**(*interface='Local Area Connection'*)

Return a list of the configured DNS servers of the specified interface

Parameters

- **`interface`** (`str`) – The name of the network interface. This is the name as
- **`appears in the Control Panel under Network Connections`** (`it`) –

Returns A list of dns servers

Return type list

CLI Example:

```
salt '*' win_dns_client.get_dns_servers 'Local Area Connection'
```

`salt.modules.win_dns_client.`**`rm_dns`**(*ip*, *interface='Local Area Connection'*)

Remove the DNS server from the network interface

CLI Example:

```
salt '*' win_dns_client.rm_dns <ip> <interface>
```

21.9.460 salt.modules.win_dsc

Module for working with Windows PowerShell DSC (Desired State Configuration)

This module is Alpha

This module applies DSC Configurations in the form of PowerShell scripts or MOF (Managed Object Format) schema files.

Use the `psget` module to manage PowerShell resources.

The idea is to leverage Salt to push DSC configuration scripts or MOF files to the Minion.

depends

- PowerShell 5.0

`salt.modules.win_dsc.`**`apply_config`**(*path*, *source=None*, *salt_env='base'*)

Run an compiled DSC configuration (a folder containing a .mof file). The folder can be cached from the salt master using the `source` option.

Parameters

- **path** (*str*) – Local path to the directory that contains the .mof configuration file to apply. Required.
- **source** (*str*) – Path to the directory that contains the .mof file on the `file_roots`. The source directory will be copied to the path directory and then executed. If the path and source directories differ, the source directory will be applied. If source is not passed, the config located at `path` will be applied. Optional.
- **salt_env** (*str*) – The salt environment to use when copying your source. Default is 'base'

Returns True if successful, otherwise False

Return type bool

CLI Example:

To apply a config that already exists on the the system

```
salt '*' dsc.apply_config C:\\DSC\\WebSiteConfiguration
```

To cache a configuration from the master and apply it:

```
salt '*' dsc.apply_config C:\\DSC\\WebSiteConfiguration salt://dsc/configs/
↪WebSiteConfiguration
```

`salt.modules.win_dsc.`**`compile_config`**(*path*, *source=None*, *config_name=None*, *config_data=None*, *config_data_source=None*, *script_parameters=None*, *salt_env='base'*)

Compile a config from a PowerShell script (`.ps1`)

Parameters

- **path** (*str*) – Path (local) to the script that will create the `.mof` configuration file. If no source is passed, the file must exist locally. Required.

- **`source`** (*`str`*) – Path to the script on `file_roots` to cache at the location specified by `path`. The source file will be cached locally and then executed. If source is not passed, the config script located at `path` will be compiled. Optional.
- **`config_name`** (*`str`*) – The name of the Configuration within the script to apply. If the script contains multiple configurations within the file a `config_name` must be specified. If the `config_name` is not specified, the name of the file will be used as the `config_name` to run. Optional.
- **`config_data`** (*`str`*) – Configuration data in the form of a hash table that will be passed to the `ConfigurationData` parameter when the `config_name` is compiled. This can be the path to a `.psd1` file containing the proper hash table or the PowerShell code to create the hash table.

  New in version 2017.7.0.
- **`config_data_source`** (*`str`*) – The path to the `.psd1` file on `file_roots` to cache at the location specified by `config_data`. If this is specified, `config_data` must be a local path instead of a hash table.

  New in version 2017.7.0.
- **`script_parameters`** (*`str`*) – Any additional parameters expected by the configuration script. These must be defined in the script itself.

  New in version 2017.7.0.
- **`salt_env`** (*`str`*) – The salt environment to use when copying the source. Default is 'base'

Returns A dictionary containing the results of the compilation

Return type dict

CLI Example:

To compile a config from a script that already exists on the system:

```
salt '*' dsc.compile_config C:\\DSC\\WebsiteConfig.ps1
```

To cache a config script to the system from the master and compile it:

```
salt '*' dsc.compile_config C:\\DSC\\WebsiteConfig.ps1 salt://dsc/configs/
↪WebsiteConfig.ps1
```

`salt.modules.win_dsc.`**`get_config`**`()`

Get the current DSC Configuration

Returns A dictionary representing the DSC Configuration on the machine

Return type dict

Raises ***`CommandExecutionError`*** – On failure

CLI Example:

```
salt '*' dsc.get_config
```

`salt.modules.win_dsc.`**`get_config_status`**`()`

Get the status of the current DSC Configuration

Returns

A dictionary representing the status of the current DSC Configuration on the machine

Return type dict

CLI Example:

```
salt '*' dsc.get_config_status
```

`salt.modules.win_dsc.`**`get_lcm_config`**`()`

Get the current Local Configuration Manager settings

Returns
A dictionary representing the Local Configuration Manager settings on the machine

Return type dict

CLI Example:

```
salt '*' dsc.get_lcm_config
```

`salt.modules.win_dsc.`**`remove_config`**`(`*reset=False*`)`

Remove the current DSC Configuration. Removes current, pending, and previous dsc configurations.

New in version 2017.7.5.

Parameters **`reset`** (*`bool`*) – Attempts to reset the DSC configuration by removing the following from `C:\Windows\System32\Configuration`:

- File: DSCStatusHistory.mof
- File: DSCEngineCache.mof
- Dir: ConfigurationStatus

Default is False

> **Warning:** `remove_config` may fail to reset the DSC environment if any of the files in the `ConfigurationStatus` directory. If you wait a few minutes and run again, it may complete successfully.

Returns True if successful

Return type bool

Raises ***`CommandExecutionError`*** – On failure

CLI Example:

```
salt '*' dsc.remove_config True
```

`salt.modules.win_dsc.`**`restore_config`**`()`

Reapplies the previous configuration.

New in version 2017.7.5.

> **Note:** The current configuration will be come the previous configuration. If run a second time back-to-back it is like toggling between two configs.

Returns True if successfully restored

Return type bool

Raises ***`CommandExecutionError`*** – On failure

CLI Example:

```
salt '*' dsc.restore_config
```

`salt.modules.win_dsc.`**`run_config`**`(`*path, source=None, config_name=None, config_data=None, config_data_source=None, script_parameters=None, salt_env='base'*`)`

Compile a DSC Configuration in the form of a PowerShell script (.ps1) and apply it. The PowerShell script can be cached from the master using the `source` option. If there is more than one config within the PowerShell script, the desired configuration can be applied by passing the name in the `config` option.

This command would be the equivalent of running `dsc.compile_config` followed by `dsc.apply_config`.

Parameters

- **`path`** (*str*) – The local path to the PowerShell script that contains the DSC Configuration. Required.
- **`source`** (*str*) – The path to the script on `file_roots` to cache at the location specified by `path`. The source file will be cached locally and then executed. If source is not passed, the config script located at `path` will be compiled. Optional.
- **`config_name`** (*str*) – The name of the Configuration within the script to apply. If the script contains multiple configurations within the file a `config_name` must be specified. If the `config_name` is not specified, the name of the file will be used as the `config_name` to run. Optional.
- **`config_data`** (*str*) – Configuration data in the form of a hash table that will be passed to the `ConfigurationData` parameter when the `config_name` is compiled. This can be the path to a `.psd1` file containing the proper hash table or the PowerShell code to create the hash table.

  New in version 2017.7.0.
- **`config_data_source`** (*str*) – The path to the `.psd1` file on `file_roots` to cache at the location specified by `config_data`. If this is specified, `config_data` must be a local path instead of a hash table.

  New in version 2017.7.0.
- **`script_parameters`** (*str*) – Any additional parameters expected by the configuration script. These must be defined in the script itself.

  New in version 2017.7.0.
- **`salt_env`** (*str*) – The salt environment to use when copying the source. Default is ’base’

Returns True if successfully compiled and applied, otherwise False

Return type bool

CLI Example:

To compile a config from a script that already exists on the system:

```
salt '*' dsc.run_config C:\\DSC\\WebsiteConfig.ps1
```

To cache a config script to the system from the master and compile it:

```
salt '*' dsc.run_config C:\\DSC\\WebsiteConfig.ps1 salt://dsc/configs/
↪WebsiteConfig.ps1
```

`salt.modules.win_dsc.`**`set_lcm_config`**(*config_mode=None, config_mode_freq=None, refresh_freq=None, reboot_if_needed=None, action_after_reboot=None, refresh_mode=None, certificate_id=None, configuration_id=None, allow_module_overwrite=None, debug_mode=False, status_retention_days=None*)

For detailed descriptions of the parameters see: https://msdn.microsoft.com/en-us/PowerShell/DSC/metaConfig

config_mode (str): How the LCM applies the configuration. Valid values are:

- ApplyOnly
- ApplyAndMonitor
- ApplyAndAutoCorrect

config_mode_freq (int): How often, in minutes, the current configuration is checked and applied. Ignored if config_mode is set to ApplyOnly. Default is 15.

refresh_mode (str): How the LCM gets configurations. Valid values are:

- Disabled
- Push
- Pull

refresh_freq (int): How often, in minutes, the LCM checks for updated configurations. (pull mode only) Default is 30.

reboot_if_needed (bool): Reboot the machine if needed after a configuration is applied. Default is False.

action_after_reboot (str): Action to take after reboot. Valid values are:

- ContinueConfiguration
- StopConfiguration

certificate_id (guid): A GUID that specifies a certificate used to access the configuration: (pull mode)

configuration_id (guid): A GUID that identifies the config file to get from a pull server. (pull mode)

allow_module_overwrite (bool): New configs are allowed to overwrite old ones on the target node.

debug_mode (str): Sets the debug level. Valid values are:

- None
- ForceModuleImport
- All

status_retention_days (int): Number of days to keep status of the current config.

Note: Either `config_mode_freq` or `refresh_freq` needs to be a multiple of the other. See documentation on MSDN for more details.

Returns True if successful, otherwise False

Return type bool

CLI Example:

```
salt '*' dsc.set_lcm_config ApplyOnly
```

`salt.modules.win_dsc.`**`test_config`**`()`

Tests the current applied DSC Configuration

Returns True if successfully applied, otherwise False

Return type bool

CLI Example:

```
salt '*' dsc.test_config
```

21.9.461 salt.modules.win_file

Manage information about files on the minion, set/read user, group data, modify the ACL of files/directories

depends

- win32api
- win32file
- win32con
- salt.utils.win_dacl

`exception` `salt.modules.win_file.`**`WindowsError`**

`salt.modules.win_file.`**`check_perms`**`(`*path*, *ret=None*, *owner=None*, *grant_perms=None*, *deny_perms=None*, *inheritance=True*, *reset=False*`)`

Check owner and permissions for the passed directory. This function checks the permissions and sets them, returning the changes made. Used by the file state to populate the return dict

Parameters

- **path** (*str*) – The full path to the directory.

- **`ret`** (*dict*) – A dictionary to append changes to and return. If not passed, will create a new dictionary to return.
- **`owner`** (*str*) – The owner to set for the directory.
- **`grant_perms`** (*dict*) – A dictionary containing the user/group and the basic permissions to check/grant, ie: `{'user': {'perms': 'basic_permission'}}`. Default is `None`.
- **`deny_perms`** (*dict*) – A dictionary containing the user/group and permissions to check/deny. Default is `None`.
- **`inheritance`** (*bool*) – `True will check if inheritance is enabled and enable it. ``False` will check if inheritance is disabled and disable it. Default is `True`.
- **`reset`** (*bool*) – `True` will show what permissions will be removed by resetting the DACL. `False` will do nothing. Default is `False`.

Returns A dictionary of changes that have been made

Return type dict

CLI Example:

```
# To see changes to ``C:\Temp`` if the 'Users' group is given 'read & execute'
↪permissions.
salt '*' file.check_perms C:\Temp\ {} Administrators "{'Users': {'perms': 'read_
↪execute'}}"

# Locally using salt call
salt-call file.check_perms C:\Temp\ {} Administrators "{'Users': {'perms': 'read_
↪execute', 'applies_to': 'this_folder_only'}}"

# Specify advanced attributes with a list
salt '*' file.check_perms C:\Temp\ {} Administrators "{'jsnuffy': {'perms': [
↪'read_attributes', 'read_ea'], 'applies_to': 'files_only'}}"
```

`salt.modules.win_file.`**`chgrp`**(*path, group*)

Change the group of a file

Under Windows, this will do nothing.

While a file in Windows does have a 'primary group', this rarely used attribute generally has no bearing on permissions unless intentionally configured and is only used to support Unix compatibility features (e.g. Services For Unix, NFS services).

Salt, therefore, remaps this function to do nothing while still being compatible with Unix behavior. When managing Windows systems, this function is superfluous and will generate an info level log entry if used directly.

If you do actually want to set the 'primary group' of a file, use `file .chpgrp`.

To set group permissions use `file.set_perms`

Parameters

- **`path`** (*str*) – The path to the file or directory
- **`group`** (*str*) – The group (unused)

Returns None

CLI Example:

```
salt '*' file.chpgrp c:\temp\test.txt administrators
```

`salt.modules.win_file.`**`chown`**(*path, user, group=None, pgroup=None, follow_symlinks=True*)

Chown a file, pass the file the desired user and group

Under Windows, the group parameter will be ignored.

This is because while files in Windows do have a 'primary group' property, this is rarely used. It generally has no bearing on permissions unless intentionally configured and is most commonly used to provide Unix compatibility (e.g. Services For Unix, NFS services).

If you do want to change the 'primary group' property and understand the implications, pass the Windows only parameter, pgroup, instead.

Parameters

- **path** (*str*) – The path to the file or directory
- **user** (*str*) – The name of the user to own the file
- **group** (*str*) – The group (not used)
- **pgroup** (*str*) – The primary group to assign
- **follow_symlinks** (*bool*) – If the object specified by `path` is a symlink, get attributes of the linked file instead of the symlink itself. Default is True

Returns True if successful, otherwise error

Return type bool

CLI Example:

```
salt '*' file.chown c:\temp\test.txt myusername
salt '*' file.chown c:\temp\test.txt myusername pgroup=Administrators
salt '*' file.chown c:\temp\test.txt myusername "pgroup='None'"
```

`salt.modules.win_file.`**`chpgrp`**(*path*, *group*)

Change the group of a file

Under Windows, this will set the rarely used primary group of a file. This generally has no bearing on permissions unless intentionally configured and is most commonly used to provide Unix compatibility (e.g. Services For Unix, NFS services).

Ensure you know what you are doing before using this function.

Parameters

- **path** (*str*) – The path to the file or directory
- **pgroup** (*str*) – The primary group to assign

Returns True if successful, otherwise error

Return type bool

CLI Example:

```
salt '*' file.chpgrp c:\temp\test.txt Administrators
salt '*' file.chpgrp c:\temp\test.txt "'None'"
```

`salt.modules.win_file.`**`get_attributes`**(*path*)

Return a dictionary object with the Windows file attributes for a file.

Parameters **path** (*str*) – The path to the file or directory

Returns A dictionary of file attributes

Return type dict

CLI Example:

```
salt '*' file.get_attributes c:\temp\a.txt
```

`salt.modules.win_file.`**`get_gid`**(*path*, *follow_symlinks=True*)

Return the id of the group that owns a given file

Under Windows, this will return the uid of the file.

While a file in Windows does have a 'primary group', this rarely used attribute generally has no bearing on permissions unless intentionally configured and is only used to support Unix compatibility features (e.g. Services For Unix, NFS services).

Salt, therefore, remaps this function to provide functionality that somewhat resembles Unix behavior for API compatibility reasons. When managing Windows systems, this function is superfluous and will generate an info level log entry if used directly.

If you do actually want to access the 'primary group' of a file, use *file.get_pgid*.

Parameters

- **path** (*str*) – The path to the file or directory
- **follow_symlinks** (*bool*) – If the object specified by `path` is a symlink, get attributes of the linked file instead of the symlink itself. Default is True

Returns The gid of the owner

Return type str

CLI Example:

```
salt '*' file.get_gid c:\temp\test.txt
```

`salt.modules.win_file.`**`get_group`**`(`*path, follow_symlinks=True*`)`

Return the group that owns a given file

Under Windows, this will return the user (owner) of the file.

While a file in Windows does have a 'primary group', this rarely used attribute generally has no bearing on permissions unless intentionally configured and is only used to support Unix compatibility features (e.g. Services For Unix, NFS services).

Salt, therefore, remaps this function to provide functionality that somewhat resembles Unix behavior for API compatibility reasons. When managing Windows systems, this function is superfluous and will generate an info level log entry if used directly.

If you do actually want to access the 'primary group' of a file, use *file.get_pgroup*.

Parameters

- **path** (*str*) – The path to the file or directory
- **follow_symlinks** (*bool*) – If the object specified by `path` is a symlink, get attributes of the linked file instead of the symlink itself. Default is True

Returns The name of the owner

Return type str

CLI Example:

```
salt '*' file.get_group c:\temp\test.txt
```

`salt.modules.win_file.`**`get_mode`**`(`*path*`)`

Return the mode of a file

Right now we're just returning None because Windows' doesn't have a mode like Linux

Parameters **path** (*str*) – The path to the file or directory

Returns None

CLI Example:

```
salt '*' file.get_mode /etc/passwd
```

`salt.modules.win_file.`**`get_pgid`**`(`*path, follow_symlinks=True*`)`

Return the id of the primary group that owns a given file (Windows only)

This function will return the rarely used primary group of a file. This generally has no bearing on permissions unless intentionally configured and is most commonly used to provide Unix compatibility (e.g. Services For Unix, NFS services).

Ensure you know what you are doing before using this function.

Parameters

- **path** (*str*) – The path to the file or directory

- **follow_symlinks** (*bool*) – If the object specified by `path` is a symlink, get attributes of the linked file instead of the symlink itself. Default is True

Returns The gid of the primary group

Return type str

CLI Example:

```
salt '*' file.get_pgid c:\temp\test.txt
```

`salt.modules.win_file.`**`get_pgroup`**(*path*, *follow_symlinks=True*)

Return the name of the primary group that owns a given file (Windows only)

This function will return the rarely used primary group of a file. This generally has no bearing on permissions unless intentionally configured and is most commonly used to provide Unix compatibility (e.g. Services For Unix, NFS services).

Ensure you know what you are doing before using this function.

The return value may be 'None', e.g. if the user is not on a domain. This is a valid group - do not confuse this with the Salt/Python value of None which means no value was returned. To be certain, use the *get_pgid* function which will return the SID, including for the system 'None' group.

Parameters

- **path** (*str*) – The path to the file or directory
- **follow_symlinks** (*bool*) – If the object specified by `path` is a symlink, get attributes of the linked file instead of the symlink itself. Default is True

Returns The name of the primary group

Return type str

CLI Example:

```
salt '*' file.get_pgroup c:\temp\test.txt
```

`salt.modules.win_file.`**`get_uid`**(*path*, *follow_symlinks=True*)

Return the id of the user that owns a given file

Symlinks are followed by default to mimic Unix behavior. Specify *follow_symlinks=False* to turn off this behavior.

Parameters

- **path** (*str*) – The path to the file or directory
- **follow_symlinks** (*bool*) – If the object specified by `path` is a symlink, get attributes of the linked file instead of the symlink itself. Default is True

Returns The uid of the owner

Return type str

CLI Example:

```
salt '*' file.get_uid c:\temp\test.txt
salt '*' file.get_uid c:\temp\test.txt follow_symlinks=False
```

`salt.modules.win_file.`**`get_user`**(*path*, *follow_symlinks=True*)

Return the user that owns a given file

Symlinks are followed by default to mimic Unix behavior. Specify *follow_symlinks=False* to turn off this behavior.

Parameters

- **path** (*str*) – The path to the file or directory
- **follow_symlinks** (*bool*) – If the object specified by `path` is a symlink, get attributes of the linked file instead of the symlink itself. Default is True

Returns The name of the owner

Return type str

CLI Example:

```
salt '*' file.get_user c:\temp\test.txt
salt '*' file.get_user c:\temp\test.txt follow_symlinks=False
```

`salt.modules.win_file.`**`gid_to_group`**(*gid*)

Convert the group id to the group name on this system

Under Windows, because groups are just another ACL entity, this function behaves the same as uid_to_user.

For maintaining Windows systems, this function is superfluous and only exists for API compatibility with Unix. Use the uid_to_user function instead; an info level log entry will be generated if this function is used directly.

Parameters **`gid`** (*`str`*) – The gid of the group

Returns The name of the group

Return type str

CLI Example:

```
salt '*' file.gid_to_group S-1-5-21-626487655-2533044672-482107328-1010
```

`salt.modules.win_file.`**`group_to_gid`**(*group*)

Convert the group to the gid on this system

Under Windows, because groups are just another ACL entity, this function behaves the same as user_to_uid, except if None is given, ” is returned.

For maintaining Windows systems, this function is superfluous and only exists for API compatibility with Unix. Use the user_to_uid function instead; an info level log entry will be generated if this function is used directly.

Parameters **`group`** (*`str`*) – The name of the group

Returns The gid of the group

Return type str

CLI Example:

```
salt '*' file.group_to_gid administrators
```

`salt.modules.win_file.`**`is_link`**(*path*)

Check if the path is a symlink

This is only supported on Windows Vista or later.

Inline with Unix behavior, this function will raise an error if the path is not a symlink, however, the error raised will be a SaltInvocationError, not an OSError.

Parameters **`path`** (*`str`*) – The path to a file or directory

Returns True if path is a symlink, otherwise False

Return type bool

CLI Example:

```
salt '*' file.is_link /path/to/link
```

`salt.modules.win_file.`**`lchown`**(*path, user, group=None, pgroup=None*)

Chown a file, pass the file the desired user and group without following any symlinks.

Under Windows, the group parameter will be ignored.

This is because while files in Windows do have a ’primary group’ property, this is rarely used. It generally has no bearing on permissions unless intentionally configured and is most commonly used to provide Unix compatibility (e.g. Services For Unix, NFS services).

If you do want to change the 'primary group' property and understand the implications, pass the Windows only parameter, pgroup, instead.

To set the primary group to 'None', it must be specified in quotes. Otherwise Salt will interpret it as the Python value of None and no primary group changes will occur. See the example below.

Parameters

- **path** (*str*) – The path to the file or directory
- **user** (*str*) – The name of the user to own the file
- **group** (*str*) – The group (not used)
- **pgroup** (*str*) – The primary group to assign

Returns True if successful, otherwise error

Return type bool

CLI Example:

```
salt '*' file.lchown c:\temp\test.txt myusername
salt '*' file.lchown c:\temp\test.txt myusername pgroup=Administrators
salt '*' file.lchown c:\temp\test.txt myusername "pgroup='None'"
```

`salt.modules.win_file.`**`makedirs_`**(*path, owner=None, grant_perms=None, deny_perms=None, inheritance=True, reset=False*)

Ensure that the parent directory containing this path is available.

Parameters

- **path** (*str*) – The full path to the directory.

Note: The path must end with a trailing slash otherwise the directory(s) will be created up to the parent directory. For example if path is `C:\temp\test`, then it would be treated as `C:\temp\` but if the path ends with a trailing slash like `C:\temp\test\`, then it would be treated as `C:\temp\test\`.

- **owner** (*str*) – The owner of the directory. If not passed, it will be the account that created the directory, likely SYSTEM.
- **grant_perms** (*dict*) – A dictionary containing the user/group and the basic permissions to grant, ie: `{'user': {'perms': 'basic_permission'}}`. You can also set the `applies_to` setting here. The default is `this_folder_subfolders_files`. Specify another `applies_to` setting like this:

```
{'user': {'perms': 'full_control', 'applies_to': 'this_folder'}
↪}
```

To set advanced permissions use a list for the `perms` parameter, ie:

```
{'user': {'perms': ['read_attributes', 'read_ea'], 'applies_to
↪': 'this_folder'}}
```

- **deny_perms** (*dict*) – A dictionary containing the user/group and permissions to deny along with the `applies_to` setting. Use the same format used for the `grant_perms` parameter. Remember, deny permissions supersede grant permissions.
- **inheritance** (*bool*) – If True the object will inherit permissions from the parent, if False, inheritance will be disabled. Inheritance setting will not apply to parent directories if they must be created.
- **reset** (*bool*) – If `True` the existing DACL will be cleared and replaced with the settings defined in this function. If `False`, new entries will be appended to the existing DACL. Default is `False`.

New in version 2018.3.0.

Returns True if successful

Return type bool

Raises ***CommandExecutionError*** – If unsuccessful

CLI Example:

```
# To grant the 'Users' group 'read & execute' permissions.
salt '*' file.makedirs C:\Temp\ Administrators "{'Users': {'perms': 'read_execute
↪'}}"

# Locally using salt call
salt-call file.makedirs C:\Temp\ Administrators "{'Users': {'perms': 'read_execute
↪', 'applies_to': 'this_folder_only'}}"

# Specify advanced attributes with a list
salt '*' file.makedirs C:\Temp\ Administrators "{'jsnuffy': {'perms': ['read_
↪attributes', 'read_ea'], 'applies_to': 'this_folder_only'}}"
```

`salt.modules.win_file.`**`makedirs_perms`**(*path, owner=None, grant_perms=None, deny_perms=None, inheritance=True, reset=True*)

Set owner and permissions for each directory created.

Parameters

- **`path`** (*`str`*) – The full path to the directory.
- **`owner`** (*`str`*) – The owner of the directory. If not passed, it will be the account that created the directory, likely SYSTEM.
- **`grant_perms`** (*`dict`*) – A dictionary containing the user/group and the basic permissions to grant, ie: `{'user': {'perms': 'basic_permission'}}`. You can also set the `applies_to` setting here. The default is `this_folder_subfolders_files`. Specify another `applies_to` setting like this:

  ```
  {'user': {'perms': 'full_control', 'applies_to': 'this_folder'}
  ↪}
  ```

  To set advanced permissions use a list for the `perms` parameter, ie:

  ```
  {'user': {'perms': ['read_attributes', 'read_ea'], 'applies_to
  ↪': 'this_folder'}}
  ```

- **`deny_perms`** (*`dict`*) – A dictionary containing the user/group and permissions to deny along with the `applies_to` setting. Use the same format used for the `grant_perms` parameter. Remember, deny permissions supersede grant permissions.
- **`inheritance`** (*`bool`*) – If `True` the object will inherit permissions from the parent, if `False`, inheritance will be disabled. Inheritance setting will not apply to parent directories if they must be created
- **`reset`** (*`bool`*) – If `True` the existing DACL will be cleared and replaced with the settings defined in this function. If `False`, new entries will be appended to the existing DACL. Default is `False`.

  New in version 2018.3.0.

Returns True if successful, otherwise raises an error

Return type bool

CLI Example:

```
# To grant the 'Users' group 'read & execute' permissions.
salt '*' file.makedirs_perms C:\Temp\ Administrators "{'Users': {'perms': 'read_
↪execute'}}"
```

(continues on next page)

(continued from previous page)

```
# Locally using salt call
salt-call file.makedirs_perms C:\Temp\ Administrators "{'Users': {'perms': 'read_
↪execute', 'applies_to': 'this_folder_only'}}"

# Specify advanced attributes with a list
salt '*' file.makedirs_perms C:\Temp\ Administrators "{'jsnuffy': {'perms': [
↪'read_attributes', 'read_ea'], 'applies_to': 'this_folder_files'}}"
```

`salt.modules.win_file.`**`mkdir`**(*path*, *owner=None*, *grant_perms=None*, *deny_perms=None*, *inheritance=True*, *reset=False*)

Ensure that the directory is available and permissions are set.

Parameters

- **`path`** (*`str`*) – The full path to the directory.
- **`owner`** (*`str`*) – The owner of the directory. If not passed, it will be the account that created the directory, likely SYSTEM
- **`grant_perms`** (*`dict`*) – A dictionary containing the user/group and the basic permissions to grant, ie: `{'user': {'perms': 'basic_permission'}}`. You can also set the `applies_to` setting here. The default is `this_folder_subfolders_files`. Specify another `applies_to` setting like this:

  ```
  {'user': {'perms': 'full_control', 'applies_to': 'this_folder'}
  ↪}
  ```

  To set advanced permissions use a list for the `perms` parameter, ie:

  ```
  {'user': {'perms': ['read_attributes', 'read_ea'], 'applies_to
  ↪': 'this_folder'}}
  ```

- **`deny_perms`** (*`dict`*) – A dictionary containing the user/group and permissions to deny along with the `applies_to` setting. Use the same format used for the `grant_perms` parameter. Remember, deny permissions supersede grant permissions.
- **`inheritance`** (*`bool`*) – If True the object will inherit permissions from the parent, if `False`, inheritance will be disabled. Inheritance setting will not apply to parent directories if they must be created.
- **`reset`** (*`bool`*) – If `True` the existing DACL will be cleared and replaced with the settings defined in this function. If `False`, new entries will be appended to the existing DACL. Default is `False`.

  New in version 2018.3.0.

Returns True if successful

Return type bool

Raises ***`CommandExecutionError`*** – If unsuccessful

CLI Example:

```
# To grant the 'Users' group 'read & execute' permissions.
salt '*' file.mkdir C:\Temp\ Administrators "{'Users': {'perms': 'read_execute'}}"

# Locally using salt call
salt-call file.mkdir C:\Temp\ Administrators "{'Users': {'perms': 'read_execute',
↪'applies_to': 'this_folder_only'}}"

# Specify advanced attributes with a list
```

(continues on next page)

(continued from previous page)

```
salt '*' file.mkdir C:\Temp\ Administrators "{'jsnuffy': {'perms': ['read_
↪attributes', 'read_ea'], 'applies_to': 'this_folder_only'}}"
```

`salt.modules.win_file.`**`readlink`**(*path*)

Return the path that a symlink points to

This is only supported on Windows Vista or later.

Inline with Unix behavior, this function will raise an error if the path is not a symlink, however, the error raised will be a SaltInvocationError, not an OSError.

Parameters **`path`** (*`str`*) – The path to the symlink

Returns The path that the symlink points to

Return type str

CLI Example:

```
salt '*' file.readlink /path/to/link
```

`salt.modules.win_file.`**`remove`**(*path*, *force=False*)

Remove the named file or directory

Parameters

- **`path`** (*`str`*) – The path to the file or directory to remove.
- **`force`** (*`bool`*) – Remove even if marked Read-Only. Default is False

Returns True if successful, False if unsuccessful

Return type bool

CLI Example:

```
salt '*' file.remove C:\Temp
```

`salt.modules.win_file.`**`set_attributes`**(*path*, *archive=None*, *hidden=None*, *normal=None*, *notIndexed=None*, *readonly=None*, *system=None*, *temporary=None*)

Set file attributes for a file. Note that the normal attribute means that all others are false. So setting it will clear all others.

Parameters

- **`path`** (*`str`*) – The path to the file or directory
- **`archive`** (*`bool`*) – Sets the archive attribute. Default is None
- **`hidden`** (*`bool`*) – Sets the hidden attribute. Default is None
- **`normal`** (*`bool`*) – Resets the file attributes. Cannot be used in conjunction with any other attribute. Default is None
- **`notIndexed`** (*`bool`*) – Sets the indexed attribute. Default is None
- **`readonly`** (*`bool`*) – Sets the readonly attribute. Default is None
- **`system`** (*`bool`*) – Sets the system attribute. Default is None
- **`temporary`** (*`bool`*) – Sets the temporary attribute. Default is None

Returns True if successful, otherwise False

Return type bool

CLI Example:

```
salt '*' file.set_attributes c:\temp\a.txt normal=True
salt '*' file.set_attributes c:\temp\a.txt readonly=True hidden=True
```

`salt.modules.win_file.`**`set_mode`**(*path*, *mode*)

Set the mode of a file

This just calls get_mode, which returns None because we don't use mode on Windows

Parameters

- **path** – The path to the file or directory
- **mode** – The mode (not used)

Returns None

CLI Example:

```
salt '*' file.set_mode /etc/passwd 0644
```

`salt.modules.win_file.`**`set_perms`**(*path*, *grant_perms=None*, *deny_perms=None*, *inheritance=True*, *reset=False*)

Set permissions for the given path

Parameters

- **path** (*str*) – The full path to the directory.
- **grant_perms** (*dict*) – A dictionary containing the user/group and the basic permissions to grant, ie: `{'user': {'perms': 'basic_permission'}}`. You can also set the `applies_to` setting here. The default for `applise_to` is `this_folder_subfolders_files`. Specify another `applies_to` setting like this:

  ```
  {'user': {'perms': 'full_control', 'applies_to': 'this_folder'}
  ↪}
  ```

  To set advanced permissions use a list for the `perms` parameter, ie:

  ```
  {'user': {'perms': ['read_attributes', 'read_ea'], 'applies_to
  ↪': 'this_folder'}}
  ```

  To see a list of available attributes and applies to settings see the documentation for salt.utils.win_dacl.

  A value of `None` will make no changes to the `grant` portion of the DACL. Default is `None`.
- **deny_perms** (*dict*) – A dictionary containing the user/group and permissions to deny along with the `applies_to` setting. Use the same format used for the `grant_perms` parameter. Remember, deny permissions supersede grant permissions.

  A value of `None` will make no changes to the `deny` portion of the DACL. Default is `None`.
- **inheritance** (*bool*) – If `True` the object will inherit permissions from the parent, if `False`, inheritance will be disabled. Inheritance setting will not apply to parent directories if they must be created. Default is `False`.
- **reset** (*bool*) – If `True` the existing DCL will be cleared and replaced with the settings defined in this function. If `False`, new entries will be appended to the existing DACL. Default is `False`.

  New in version 2018.3.0.

Returns True if successful

Return type bool

Raises ***CommandExecutionError*** – If unsuccessful

CLI Example:

```
# To grant the 'Users' group 'read & execute' permissions.
salt '*' file.set_perms C:\Temp\ "{'Users': {'perms': 'read_execute'}}"

# Locally using salt call
salt-call file.set_perms C:\Temp\ "{'Users': {'perms': 'read_execute', 'applies_to
↪': 'this_folder_only'}}"
```

(continues on next page)

(continued from previous page)

```
# Specify advanced attributes with a list
salt '*' file.set_perms C:\Temp\ "{'jsnuffy': {'perms': ['read_attributes', 'read_
↪ea'], 'applies_to': 'this_folder_only'}}"
```

`salt.modules.win_file.`**`stats`**(*path*, *hash_type='sha256'*, *follow_symlinks=True*)

Return a dict containing the stats about a given file

Under Windows, *gid* will equal *uid* and *group* will equal *user*.

While a file in Windows does have a 'primary group', this rarely used attribute generally has no bearing on permissions unless intentionally configured and is only used to support Unix compatibility features (e.g. Services For Unix, NFS services).

Salt, therefore, remaps these properties to keep some kind of compatibility with Unix behavior. If the 'primary group' is required, it can be accessed in the *pgroup* and *pgid* properties.

Parameters

- **path** (*str*) – The path to the file or directory
- **hash_type** (*str*) – The type of hash to return
- **follow_symlinks** (*bool*) – If the object specified by `path` is a symlink, get attributes of the linked file instead of the symlink itself. Default is True

Returns A dictionary of file/directory stats

Return type dict

CLI Example:

```
salt '*' file.stats /etc/passwd
```

`salt.modules.win_file.`**`symlink`**(*src*, *link*)

Create a symbolic link to a file

This is only supported with Windows Vista or later and must be executed by a user with the SeCreateSymbolicLink privilege.

The behavior of this function matches the Unix equivalent, with one exception - invalid symlinks cannot be created. The source path must exist. If it doesn't, an error will be raised.

Parameters

- **src** (*str*) – The path to a file or directory
- **link** (*str*) – The path to the link

Returns True if successful, otherwise False

Return type bool

CLI Example:

```
salt '*' file.symlink /path/to/file /path/to/link
```

`salt.modules.win_file.`**`uid_to_user`**(*uid*)

Convert a uid to a user name

Parameters **uid** (*str*) – The user id to lookup

Returns The name of the user

Return type str

CLI Example:

```
salt '*' file.uid_to_user S-1-5-21-626487655-2533044672-482107328-1010
```

`salt.modules.win_file.`**`user_to_uid`**(*user*)

Convert user name to a uid

Parameters **user** (*str*) – The user to lookup

Returns The user id of the user

Return type str

CLI Example:

```
salt '*' file.user_to_uid myusername
```

21.9.462 salt.modules.win_firewall

Module for configuring Windows Firewall using `netsh`

`salt.modules.win_firewall.`**`add_rule`**(*name*, *localport*, *protocol='tcp'*, *action='allow'*, *dir='in'*, *remoteip='any'*)

New in version 2015.5.0.

Add a new inbound or outbound rule to the firewall policy

Parameters

- **`name`** (*str*) – The name of the rule. Must be unique and cannot be "all". Required.
- **`localport`** (*int*) – The port the rule applies to. Must be a number between 0 and 65535. Can be a range. Can specify multiple ports separated by commas. Required.
- **`protocol`** (*Optional[str]*) – The protocol. Can be any of the following:
  - A number between 0 and 255
  - icmpv4
  - icmpv6
  - tcp
  - udp
  - any
- **`action`** (*Optional[str]*) – The action the rule performs. Can be any of the following:
  - allow
  - block
  - bypass
- **`dir`** (*Optional[str]*) – The direction. Can be `in` or `out`.
- **`remoteip`** (*Optional [str]*) – The remote IP. Can be any of the following:
  - any
  - localsubnet
  - dns
  - dhcp
  - wins
  - defaultgateway
  - Any valid IPv4 address (192.168.0.12)
  - Any valid IPv6 address (2002:9b3b:1a31:4:208:74ff:fe39:6c43)
  - Any valid subnet (192.168.1.0/24)
  - Any valid range of IP addresses (192.168.0.1-192.168.0.12)
  - A list of valid IP addresses

  Can be combinations of the above separated by commas.

Returns True if successful

Return type bool

Raises ***CommandExecutionError*** – If the command fails

CLI Example:

```
salt '*' firewall.add_rule 'test' '8080' 'tcp'
salt '*' firewall.add_rule 'test' '1' 'icmpv4'
salt '*' firewall.add_rule 'test_remote_ip' '8000' 'tcp' 'allow' 'in' '192.168.0.1
↪'
```

`salt.modules.win_firewall.`**`delete_rule`**(*name=None, localport=None, protocol=None, dir=None, remoteip=None*)

New in version 2015.8.0.

Delete an existing firewall rule identified by name and optionally by ports, protocols, direction, and remote IP.

Parameters

- **`name`** (*`str`*) – The name of the rule to delete. If the name `all` is used you must specify additional parameters.
- **`localport`** (*`Optional[str]`*) – The port of the rule. If protocol is not specified, protocol will be set to `tcp`
- **`protocol`** (*`Optional[str]`*) – The protocol of the rule. Default is `tcp` when `localport` is specified
- **`dir`** (*`Optional[str]`*) – The direction of the rule.
- **`remoteip`** (*`Optional[str]`*) – The remote IP of the rule.

Returns True if successful

Return type bool

Raises ***`CommandExecutionError`*** – If the command fails

CLI Example:

```
# Delete incoming tcp port 8080 in the rule named 'test'
salt '*' firewall.delete_rule 'test' '8080' 'tcp' 'in'

# Delete the incoming tcp port 8000 from 192.168.0.1 in the rule named
# 'test_remote_ip'
salt '*' firewall.delete_rule 'test_remote_ip' '8000' 'tcp' 'in' '192.168.0.1'

# Delete all rules for local port 80:
salt '*' firewall.delete_rule all 80 tcp

# Delete a rule called 'allow80':
salt '*' firewall.delete_rule allow80
```

`salt.modules.win_firewall.`**`disable`**(*profile='allprofiles'*)

Disable firewall profile

Parameters **`profile`** (*`Optional[str]`*) – The name of the profile to disable. Default is `allprofiles`. Valid options are:

- allprofiles
- domainprofile
- privateprofile
- publicprofile

Returns True if successful

Return type bool

Raises ***`CommandExecutionError`*** – If the command fails

CLI Example:

```
salt '*' firewall.disable
```

`salt.modules.win_firewall.`**`enable`**(*profile='allprofiles'*)

New in version 2015.5.0.

Enable firewall profile

Parameters **`profile`** (*`Optional[str]`*) – The name of the profile to enable. Default is `allprofiles`. Valid options are:

- allprofiles
- domainprofile
- privateprofile
- publicprofile

Returns True if successful

Return type bool

Raises `CommandExecutionError` – If the command fails

CLI Example:

```
salt '*' firewall.enable
```

`salt.modules.win_firewall.`**`get_all_profiles`**(*store='local'*)

Gets all properties for all profiles in the specified store

New in version 2018.3.4.

New in version 2019.2.0.

Parameters **`store`** (*`str`*) – The store to use. This is either the local firewall policy or the policy defined by local group policy. Valid options are:

- lgpo
- local

Default is `local`

Returns A dictionary containing the specified settings for each profile

Return type dict

CLI Example:

```
# Get all firewall settings for all profiles
salt * firewall.get_all_settings

# Get all firewall settings for all profiles as defined by local group
# policy

salt * firewall.get_all_settings lgpo
```

`salt.modules.win_firewall.`**`get_all_settings`**(*domain, store='local'*)

Gets all the properties for the specified profile in the specified store

New in version 2018.3.4.

New in version 2019.2.0.

Parameters

- **`profile`** (*`str`*) – The firewall profile to query. Valid options are:
  - domain
  - public
  - private
- **`store`** (*`str`*) – The store to use. This is either the local firewall policy or the policy defined by local group policy. Valid options are:
  - lgpo
  - local

  Default is `local`

Returns A dictionary containing the specified settings

Return type dict

CLI Example:

```
# Get all firewall settings for connections on the domain profile
salt * win_firewall.get_all_settings domain

# Get all firewall settings for connections on the domain profile as
# defined by local group policy
salt * win_firewall.get_all_settings domain lgpo
```

`salt.modules.win_firewall.`**`get_config`**`()`

Get the status of all the firewall profiles

Returns A dictionary of all profiles on the system

Return type dict

Raises ***`CommandExecutionError`*** – If the command fails

CLI Example:

```
salt '*' firewall.get_config
```

`salt.modules.win_firewall.`**`get_rule`**`(`*name='all'*`)`

New in version 2015.5.0.

Display all matching rules as specified by name

Parameters **`name`** (*`Optional[str]`*) – The full name of the rule. `all` will return all rules. Default is `all`

Returns A dictionary of all rules or rules that match the name exactly

Return type dict

Raises ***`CommandExecutionError`*** – If the command fails

CLI Example:

```
salt '*' firewall.get_rule 'MyAppPort'
```

`salt.modules.win_firewall.`**`get_settings`**`(`*profile, section, store='local'*`)`

Get the firewall property from the specified profile in the specified store as returned by `netsh advfirewall`.

New in version 2018.3.4.

New in version 2019.2.0.

Parameters

- **`profile`** (*`str`*) – The firewall profile to query. Valid options are:
  - domain
  - public
  - private
- **`section`** (*`str`*) – The property to query within the selected profile. Valid options are:
  - firewallpolicy : inbound/outbound behavior
  - logging : firewall logging settings
  - settings : firewall properties
  - state : firewalls state (on | off)
- **`store`** (*`str`*) – The store to use. This is either the local firewall policy or the policy defined by local group policy. Valid options are:
  - lgpo
  - local

  Default is `local`

Returns A dictionary containing the properties for the specified profile

Return type dict

Raises

- ***`CommandExecutionError`*** – If an error occurs
- **`ValueError`** – If the parameters are incorrect

CLI Example:

```
# Get the inbound/outbound firewall settings for connections on the
# local domain profile
salt * win_firewall.get_settings domain firewallpolicy

```

(continues on next page)

(continued from previous page)

```
# Get the inbound/outbound firewall settings for connections on the
# domain profile as defined by local group policy
salt * win_firewall.get_settings domain firewallpolicy lgpo
```

`salt.modules.win_firewall.`**`rule_exists`**`(`*name*`)`

New in version 2016.11.6.

Checks if a firewall rule exists in the firewall policy

Parameters **`name`** (`str`) – The name of the rule

Returns True if exists, otherwise False

Return type bool

CLI Example:

```
# Is there a rule named RemoteDesktop
salt '*' firewall.rule_exists RemoteDesktop
```

`salt.modules.win_firewall.`**`set_firewall_settings`**`(`*profile, inbound=None, outbound=None, store='local'*`)`

Set the firewall inbound/outbound settings for the specified profile and store

New in version 2018.3.4.

New in version 2019.2.0.

Parameters

- **`profile`** (`str`) – The firewall profile to query. Valid options are:
    - domain
    - public
    - private
- **`inbound`** (`str`) – The inbound setting. If `None` is passed, the setting will remain unchanged. Valid values are:
    - blockinbound
    - blockinboundalways
    - allowinbound
    - notconfigured

  Default is `None`
- **`outbound`** (`str`) – The outbound setting. If `None` is passed, the setting will remain unchanged. Valid values are:
    - allowoutbound
    - blockoutbound
    - notconfigured

  Default is `None`
- **`store`** (`str`) – The store to use. This is either the local firewall policy or the policy defined by local group policy. Valid options are:
    - lgpo
    - local

  Default is `local`

Returns `True` if successful

Return type bool

Raises

- ***`CommandExecutionError`*** – If an error occurs
- **`ValueError`** – If the parameters are incorrect

CLI Example:

```
# Set the inbound setting for the domain profile to block inbound
# connections
```

(continues on next page)

(continued from previous page)

```
salt * firewall.set_firewall_settings domain='domain' inbound='blockinbound'

# Set the outbound setting for the domain profile to allow outbound
# connections
salt * firewall.set_firewall_settings domain='domain' outbound='allowoutbound'

# Set inbound/outbound settings for the domain profile in the group
# policy to block inbound and allow outbound
salt * firewall.set_firewall_settings domain='domain' inbound='blockinbound'
↪outbound='allowoutbound' store='lgpo'
```

`salt.modules.win_firewall.`**`set_logging_settings`**(*profile, setting, value, store='local'*)

Configure logging settings for the Windows firewall.

New in version 2018.3.4.

New in version 2019.2.0.

Parameters

- **profile** (*str*) – The firewall profile to configure. Valid options are:
  - domain
  - public
  - private
- **setting** (*str*) – The logging setting to configure. Valid options are:
  - allowedconnections
  - droppedconnections
  - filename
  - maxfilesize
- **value** (*str*) – The value to apply to the setting. Valid values are dependent upon the setting being configured. Valid options are:

  allowedconnections:
  - enable
  - disable
  - notconfigured

  droppedconnections:
  - enable
  - disable
  - notconfigured

  filename:
  - Full path and name of the firewall log file
  - notconfigured

  maxfilesize:
  - 1 - 32767
  - notconfigured

  Note: `notconfigured` can only be used when using the lgpo store

- **store** (*str*) – The store to use. This is either the local firewall policy or the policy defined by local group policy. Valid options are:
  - lgpo
  - local

  Default is `local`

Returns `True` if successful

Return type bool

Raises

- *CommandExecutionError* – If an error occurs
- **ValueError** – If the parameters are incorrect

CLI Example:

```
# Log allowed connections and set that in local group policy
salt * firewall.set_logging_settings domain allowedconnections enable lgpo

# Don't log dropped connections
salt * firewall.set_logging_settings profile=private setting=droppedconnections␣
↪value=disable

# Set the location of the log file
salt * firewall.set_logging_settings domain filename C:\windows\logs\firewall.log

# You can also use environment variables
salt * firewall.set_logging_settings domain filename %systemroot
↪%\system32\LogFiles\Firewall\pfirewall.log

# Set the max file size of the log to 2048 Kb
salt * firewall.set_logging_settings domain maxfilesize 2048
```

`salt.modules.win_firewall.`**`set_settings`**(*profile*, *setting*, *value*, *store='local'*)

Configure firewall settings.

New in version 2018.3.4.

New in version 2019.2.0.

Parameters

- **profile** (*str*) – The firewall profile to configure. Valid options are:
  - domain
  - public
  - private
- **setting** (*str*) – The firewall setting to configure. Valid options are:
  - localfirewallrules
  - localconsecrules
  - inboundusernotification
  - remotemanagement
  - unicastresponsetomulticast
- **value** (*str*) – The value to apply to the setting. Valid options are
  - enable
  - disable
  - notconfigured

  Note: `notconfigured` can only be used when using the lgpo store

- **store** (*str*) – The store to use. This is either the local firewall policy or the policy defined by local group policy. Valid options are:
  - lgpo
  - local

  Default is `local`

Returns `True` if successful

Return type bool

Raises

- *CommandExecutionError* – If an error occurs
- **ValueError** – If the parameters are incorrect

CLI Example:

```
# Merge local rules with those distributed through group policy
salt * firewall.set_settings domain localfirewallrules enable

# Allow remote management of Windows Firewall
salt * firewall.set_settings domain remotemanagement enable
```

`salt.modules.win_firewall.`**`set_state`**(*profile*, *state*, *store='local'*)

Configure the firewall state.

New in version 2018.3.4.

New in version 2019.2.0.

Parameters

- **`profile`** (*`str`*) – The firewall profile to configure. Valid options are:
  - domain
  - public
  - private
- **`state`** (*`str`*) – The firewall state. Valid options are:
  - on
  - off
  - notconfigured

  Note: `notconfigured` can only be used when using the lgpo store
- **`store`** (*`str`*) – The store to use. This is either the local firewall policy or the policy defined by local group policy. Valid options are:
  - lgpo
  - local

  Default is `local`

Returns `True` if successful

Return type bool

Raises

- ***`CommandExecutionError`*** – If an error occurs
- **`ValueError`** – If the parameters are incorrect

CLI Example:

```
# Turn the firewall off when the domain profile is active
salt * firewall.set_state domain off

# Turn the firewall on when the public profile is active and set that in
# the local group policy
salt * firewall.set_state public on lgpo
```

21.9.463 salt.modules.win_groupadd

Manage groups on Windows

Important: If you feel that Salt should be using this module to manage groups on a minion, and it is using a different module (or gives an error similar to *'group.info' is not available*), see *here*.

`salt.modules.win_groupadd.`**`add`**(*name*, ***kwargs*)

Add the specified group

Parameters **`name`** (*`str`*) – The name of the group to add

Returns `True` if successful, otherwise `False`

Return type bool

CLI Example:

```
salt '*' group.add foo
```

`salt.modules.win_groupadd.`**`adduser`**(*name*, *username*, ***kwargs*)

Add a user to a group

Parameters

- **`name`** (*`str`*) – The name of the group to modify
- **`username`** (*`str`*) – The name of the user to add to the group

Returns `True` if successful, otherwise `False`

Return type bool

CLI Example:

```
salt '*' group.adduser foo username
```

`salt.modules.win_groupadd.`**`delete`**(*name*, ***kwargs*)

Remove the named group

Parameters **`name`** (*`str`*) – The name of the group to remove

Returns `True` if successful, otherwise `False`

Return type bool

CLI Example:

```
salt '*' group.delete foo
```

`salt.modules.win_groupadd.`**`deluser`**(*name*, *username*, ***kwargs*)

Remove a user from a group

Parameters

- **`name`** (*`str`*) – The name of the group to modify
- **`username`** (*`str`*) – The name of the user to remove from the group

Returns `True` if successful, otherwise `False`

Return type bool

CLI Example:

```
salt '*' group.deluser foo username
```

`salt.modules.win_groupadd.`**`getent`**(*refresh=False*)

Return info on all groups

Parameters **`refresh`** (*`bool`*) – Refresh the info for all groups in `__context__`. If False only the groups in `__context__` will be returned. If True the `__context__` will be refreshed with current data and returned. Default is False

Returns A list of groups and their information

CLI Example:

```
salt '*' group.getent
```

`salt.modules.win_groupadd.`**`info`**(*name*)

Return information about a group

Parameters **`name`** (*`str`*) – The name of the group for which to get information

Returns A dictionary of information about the group

Return type dict

CLI Example:

```
salt '*' group.info foo
```

`salt.modules.win_groupadd.`**`list_groups`**(*refresh=False*)

Return a list of groups

Parameters **`refresh`** (*`bool`*) – Refresh the info for all groups in `__context__`. If False only the groups in `__context__` will be returned. If True, the `__context__` will be refreshed with current data and returned. Default is False

Returns A list of groups on the machine

Return type list

CLI Example:

```
salt '*' group.list_groups
```

`salt.modules.win_groupadd.`**`members`**(*name*, *members_list*, ***kwargs*)

Ensure a group contains only the members in the list

Parameters

- **`name`** (*`str`*) – The name of the group to modify
- **`members_list`** (*`str`*) – A single user or a comma separated list of users. The group will contain only the users specified in this list.

Returns `True` if successful, otherwise `False`

Return type bool

CLI Example:

```
salt '*' group.members foo 'user1,user2,user3'
```

21.9.464 salt.modules.win_iis module

Microsoft IIS site management via WebAdministration powershell module

maintainer Shane Lee <slee@saltstack.com>, Robert Booth <rbooth@saltstack.com>

platform Windows

depends PowerShell

depends WebAdministration module (PowerShell) (IIS)

New in version 2016.3.0.

`salt.modules.win_iis.`**`create_app`**(*name*, *site*, *sourcepath*, *apppool=None*)

Create an IIS application.

Note: This function only validates against the application name, and will return True even if the application already exists with a different configuration. It will not modify the configuration of an existing application.

Parameters

- **`name`** (*`str`*) – The IIS application.
- **`site`** (*`str`*) – The IIS site name.
- **`sourcepath`** (*`str`*) – The physical path.
- **`apppool`** (*`str`*) – The name of the IIS application pool.

Returns True if successful, otherwise False

Return type bool

CLI Example:

```
salt '*' win_iis.create_app name='app0' site='site0' sourcepath='C:\site0'␣
↪apppool='site0'
```

`salt.modules.win_iis.`**`create_apppool`**(*name*)

Create an IIS application pool.

Note: This function only validates against the application pool name, and will return True even if the application pool already exists with a different configuration. It will not modify the configuration of an existing application pool.

Parameters **`name`** (*`str`*) – The name of the IIS application pool.
Returns True if successful, otherwise False
Return type bool

CLI Example:

```
salt '*' win_iis.create_apppool name='MyTestPool'
```

`salt.modules.win_iis.`**`create_backup`**(*name*)

Backup an IIS Configuration on the System.

New in version 2017.7.0.

Note: Backups are stored in the `$env:Windir\System32\inetsrv\backup` folder.

Parameters **`name`** (*`str`*) – The name to give the backup
Returns True if successful, otherwise False
Return type bool

CLI Example:

```
salt '*' win_iis.create_backup good_config_20170209
```

`salt.modules.win_iis.`**`create_binding`**(*site, hostheader=”, ipaddress=’*’, port=80, protocol=’http’, sslflags=None*)

Create an IIS Web Binding.

Note: This function only validates against the binding ipaddress:port:hostheader combination, and will return True even if the binding already exists with a different configuration. It will not modify the configuration of an existing binding.

Parameters
- **`site`** (*`str`*) – The IIS site name.
- **`hostheader`** (*`str`*) – The host header of the binding. Usually a hostname.
- **`ipaddress`** (*`str`*) – The IP address of the binding.
- **`port`** (*`int`*) – The TCP port of the binding.
- **`protocol`** (*`str`*) – The application protocol of the binding.
- **`sslflags`** (*`str`*) – The flags representing certificate type and storage of the binding.

Returns True if successful, otherwise False
Return type bool

CLI Example:

```
salt '*' win_iis.create_binding site='site0' hostheader='example.com' ipaddress='*
↪' port='80'
```

`salt.modules.win_iis.`**`create_cert_binding`**(*name*, *site*, *hostheader=''*, *ipaddress='*'*, *port=443*, *sslflags=0*)

Assign a certificate to an IIS Web Binding.

New in version 2016.11.0.

Note: The web binding that the certificate is being assigned to must already exist.

Parameters
- **name** (*str*) – The thumbprint of the certificate.
- **site** (*str*) – The IIS site name.
- **hostheader** (*str*) – The host header of the binding.
- **ipaddress** (*str*) – The IP address of the binding.
- **port** (*int*) – The TCP port of the binding.
- **sslflags** (*int*) – Flags representing certificate type and certificate storage of the binding.

Returns True if successful, otherwise False

Return type bool

CLI Example:

```
salt '*' win_iis.create_cert_binding name='AAA000' site='site0' hostheader=
↪'example.com' ipaddress='*' port='443'
```

`salt.modules.win_iis.`**`create_site`**(*name*, *sourcepath*, *apppool=''*, *hostheader=''*, *ipaddress='*'*, *port=80*, *protocol='http'*)

Create a basic website in IIS.

Note: This function only validates against the site name, and will return True even if the site already exists with a different configuration. It will not modify the configuration of an existing site.

Parameters
- **name** (*str*) – The IIS site name.
- **sourcepath** (*str*) – The physical path of the IIS site.
- **apppool** (*str*) – The name of the IIS application pool.
- **hostheader** (*str*) – The host header of the binding. Usually the hostname or website name, ie: www.contoso.com
- **ipaddress** (*str*) – The IP address of the binding.
- **port** (*int*) – The TCP port of the binding.
- **protocol** (*str*) – The application protocol of the binding. (http, https, etc.)

Returns True if successful, otherwise False.

Return type bool

Note: If an application pool is specified, and that application pool does not already exist, it will be created.

CLI Example:

```
salt '*' win_iis.create_site name='My Test Site' sourcepath='c:\stage' apppool=
↪'TestPool'
```

`salt.modules.win_iis.`**`create_vdir`**(*name*, *site*, *sourcepath*, *app='/'*)

Create an IIS virtual directory.

Note: This function only validates against the virtual directory name, and will return True even if the virtual directory already exists with a different configuration. It will not modify the configuration of an existing virtual directory.

Parameters
- **`name`** (*`str`*) – The virtual directory name.
- **`site`** (*`str`*) – The IIS site name.
- **`sourcepath`** (*`str`*) – The physical path.
- **`app`** (*`str`*) – The IIS application.

Returns True if successful, otherwise False

Return type bool

CLI Example:

```
salt '*' win_iis.create_vdir name='vd0' site='site0' sourcepath=
↪'C:\inetpub\vdirs\vd0'
```

`salt.modules.win_iis.`**`get_container_setting`**(*name*, *container*, *settings*)

Get the value of the setting for the IIS container.

New in version 2016.11.0.

Parameters
- **`name`** (*`str`*) – The name of the IIS container.
- **`container`** (*`str`*) – The type of IIS container. The container types are: AppPools, Sites, SslBindings
- **`settings`** (*`dict`*) – A dictionary of the setting names and their values.

Returns A dictionary of the provided settings and their values.

Return type dict

CLI Example:

```
salt '*' win_iis.get_container_setting name='MyTestPool' container='AppPools'
    settings="['processModel.identityType']"
```

`salt.modules.win_iis.`**`get_webapp_settings`**(*name*, *site*, *settings*)

New in version 2017.7.0.

Get the value of the setting for the IIS web application.

Note: Params are case sensitive

Parameters
- **`name`** (*`str`*) – The name of the IIS web application.
- **`site`** (*`str`*) – The site name contains the web application. Example: Default Web Site
- **`settings`** (*`str`*) – A dictionary of the setting names and their values. Available settings: physicalPath, applicationPool, userName, password

Returns A dictionary of the provided settings and their values.

Return type dict

CLI Example:

```
salt '*' win_iis.get_webapp_settings name='app0' site='Default Web Site'
    settings="['physicalPath','applicationPool']"
```

`salt.modules.win_iis.`**`get_webconfiguration_settings`**`(`*name*, *settings*`)`

Get the webconfiguration settings for the IIS PSPath.

Parameters

- **name** (*str*) – The PSPath of the IIS webconfiguration settings.
- **settings** (*list*) – A list of dictionaries containing setting name and filter.

Returns A list of dictionaries containing setting name, filter and value.

Return type dict

CLI Example:

```
salt '*' win_iis.get_webconfiguration_settings name='IIS:\' settings="[{'name':
↪'enabled', 'filter': 'system.webServer/security/authentication/
↪anonymousAuthentication'}]"
```

`salt.modules.win_iis.`**`list_apppools`**`()`

List all configured IIS application pools.

Returns A dictionary of IIS application pools and their details.

Return type dict

CLI Example:

```
salt '*' win_iis.list_apppools
```

`salt.modules.win_iis.`**`list_apps`**`(`*site*`)`

Get all configured IIS applications for the specified site.

Parameters **site** (*str*) – The IIS site name.

Returns: A dictionary of the application names and properties.

CLI Example:

```
salt '*' win_iis.list_apps site
```

`salt.modules.win_iis.`**`list_backups`**`()`

List the IIS Configuration Backups on the System.

New in version 2017.7.0.

Note: Backups are made when a configuration is edited. Manual backups are stored in the `$env:Windir\System32\inetsrv\backup` folder.

Returns A dictionary of IIS Configurations backed up on the system.

Return type dict

CLI Example:

```
salt '*' win_iis.list_backups
```

`salt.modules.win_iis.`**`list_bindings`**`(`*site*`)`

Get all configured IIS bindings for the specified site.

Parameters **site** (*str*) – The name if the IIS Site

Returns A dictionary of the binding names and properties.

Return type dict

CLI Example:

```
salt '*' win_iis.list_bindings site
```

`salt.modules.win_iis.`**`list_cert_bindings`**`(`*site*`)`

List certificate bindings for an IIS site.

New in version 2016.11.0.

Parameters **site** (*str*) – The IIS site name.

Returns A dictionary of the binding names and properties.

Return type dict

CLI Example:

```
salt '*' win_iis.list_bindings site
```

`salt.modules.win_iis.`**`list_sites`**`()`

List all the currently deployed websites.

Returns A dictionary of the IIS sites and their properties.

Return type dict

CLI Example:

```
salt '*' win_iis.list_sites
```

`salt.modules.win_iis.`**`list_vdirs`**`(`*site, app='/'*`)`

Get all configured IIS virtual directories for the specified site, or for the combination of site and application.

Parameters

- **site** (*str*) – The IIS site name.
- **app** (*str*) – The IIS application.

Returns A dictionary of the virtual directory names and properties.

Return type dict

CLI Example:

```
salt '*' win_iis.list_vdirs site
```

`salt.modules.win_iis.`**`list_worker_processes`**`(`*apppool*`)`

Returns a list of worker processes that correspond to the passed application pool.

New in version 2017.7.0.

Parameters **apppool** (*str*) – The application pool to query

Returns A dictionary of worker processes with their process IDs

Return type dict

CLI Example:

```
salt '*' win_iis.list_worker_processes 'My App Pool'
```

`salt.modules.win_iis.`**`modify_binding`**`(`*site, binding, hostheader=None, ipaddress=None, port=None, sslflags=None*`)`

Modify an IIS Web Binding. Use `site` and `binding` to target the binding.

New in version 2017.7.0.

Parameters

- **site** (*str*) – The IIS site name.
- **binding** (*str*) – The binding to edit. This is a combination of the IP address, port, and hostheader. It is in the following format: ipaddress:port:hostheader. For example, `*:80:` or `*:80:salt.com`
- **hostheader** (*str*) – The host header of the binding. Usually the hostname.
- **ipaddress** (*str*) – The IP address of the binding.
- **port** (*int*) – The TCP port of the binding.

- **sslflags** (*str*) – The flags representing certificate type and storage of the binding.

Returns True if successful, otherwise False

Return type bool

CLI Example:

The following will seat the host header of binding `*:80:` for `site0` to `example.com`

```
salt '*' win_iis.modify_binding site='site0' binding='*:80:' hostheader='example.
↪com'
```

salt.modules.win_iis.**modify_site**(*name*, *sourcepath=None*, *apppool=None*)

Modify a basic website in IIS.

New in version 2017.7.0.

Parameters

- **name** (*str*) – The IIS site name.
- **sourcepath** (*str*) – The physical path of the IIS site.
- **apppool** (*str*) – The name of the IIS application pool.

Returns True if successful, otherwise False.

Return type bool

Note: If an application pool is specified, and that application pool does not already exist, it will be created.

CLI Example:

```
salt '*' win_iis.modify_site name='My Test Site' sourcepath='c:\new_path' apppool=
↪'NewTestPool'
```

salt.modules.win_iis.**remove_app**(*name*, *site*)

Remove an IIS application.

Parameters

- **name** (*str*) – The application name.
- **site** (*str*) – The IIS site name.

Returns True if successful, otherwise False

Return type bool

CLI Example:

```
salt '*' win_iis.remove_app name='app0' site='site0'
```

salt.modules.win_iis.**remove_apppool**(*name*)

Remove an IIS application pool.

Parameters **name** (*str*) – The name of the IIS application pool.

Returns True if successful, otherwise False

Return type bool

CLI Example:

```
salt '*' win_iis.remove_apppool name='MyTestPool'
```

salt.modules.win_iis.**remove_backup**(*name*)

Remove an IIS Configuration backup from the System.

New in version 2017.7.0.

Parameters **name** (*str*) – The name of the backup to remove

Returns True if successful, otherwise False

Return type bool

CLI Example:

```
salt '*' win_iis.remove_backup backup_20170209
```

`salt.modules.win_iis.`**`remove_binding`**(*site*, *hostheader=''*, *ipaddress='*'*, *port=80*)

Remove an IIS binding.

Parameters

- **site** (*str*) – The IIS site name.
- **hostheader** (*str*) – The host header of the binding.
- **ipaddress** (*str*) – The IP address of the binding.
- **port** (*int*) – The TCP port of the binding.

Returns True if successful, otherwise False

Return type bool

CLI Example:

```
salt '*' win_iis.remove_binding site='site0' hostheader='example.com' ipaddress='*
↪' port='80'
```

`salt.modules.win_iis.`**`remove_cert_binding`**(*name*, *site*, *hostheader=''*, *ipaddress='*'*, *port=443*)

Remove a certificate from an IIS Web Binding.

New in version 2016.11.0.

Note: This function only removes the certificate from the web binding. It does not remove the web binding itself.

Parameters

- **name** (*str*) – The thumbprint of the certificate.
- **site** (*str*) – The IIS site name.
- **hostheader** (*str*) – The host header of the binding.
- **ipaddress** (*str*) – The IP address of the binding.
- **port** (*int*) – The TCP port of the binding.

Returns True if successful, otherwise False

Return type bool

CLI Example:

```
salt '*' win_iis.remove_cert_binding name='AAA000' site='site0' hostheader=
↪'example.com' ipaddress='*' port='443'
```

`salt.modules.win_iis.`**`remove_site`**(*name*)

Delete a website from IIS.

Parameters **name** (*str*) – The IIS site name.

Returns True if successful, otherwise False

Return type bool

Note: This will not remove the application pool used by the site.

CLI Example:

```
salt '*' win_iis.remove_site name='My Test Site'
```

`salt.modules.win_iis.`**`remove_vdir`**(*name*, *site*, *app='/'*)

Remove an IIS virtual directory.

Parameters

- **name** (*str*) – The virtual directory name.

- **site** (*str*) – The IIS site name.
- **app** (*str*) – The IIS application.

Returns True if successful, otherwise False

Return type bool

CLI Example:

```
salt '*' win_iis.remove_vdir name='vdir0' site='site0'
```

`salt.modules.win_iis.`**`restart_apppool`**(*name*)

Restart an IIS application pool.

New in version 2016.11.0.

Parameters **name** (*str*) – The name of the IIS application pool.

Returns True if successful, otherwise False

Return type bool

CLI Example:

```
salt '*' win_iis.restart_apppool name='MyTestPool'
```

`salt.modules.win_iis.`**`restart_site`**(*name*)

Restart a Web Site in IIS.

New in version 2017.7.0.

Parameters **name** (*str*) – The name of the website to restart.

Returns True if successful, otherwise False

Return type bool

CLI Example:

```
salt '*' win_iis.restart_site name='My Test Site'
```

`salt.modules.win_iis.`**`set_container_setting`**(*name, container, settings*)

Set the value of the setting for an IIS container.

New in version 2016.11.0.

Parameters

- **name** (*str*) – The name of the IIS container.
- **container** (*str*) – The type of IIS container. The container types are: AppPools, Sites, SslBindings
- **settings** (*dict*) – A dictionary of the setting names and their values.

Returns True if successful, otherwise False

Return type bool

CLI Example:

```
salt '*' win_iis.set_container_setting name='MyTestPool' container='AppPools'
    settings="{'managedPipeLineMode': 'Integrated'}"
```

`salt.modules.win_iis.`**`set_webapp_settings`**(*name, site, settings*)

New in version 2017.7.0.

Configure an IIS application.

Note: This function only configures an existing app. Params are case sensitive.

Parameters

- **name** (*str*) – The IIS application.
- **site** (*str*) – The IIS site name.

- **settings** (*str*) – A dictionary of the setting names and their values. - physicalPath: The physical path of the webapp. - applicationPool: The application pool for the webapp. - userName: "connectAs" user - password: "connectAs" password for user

Returns A boolean representing whether all changes succeeded.

Return type bool

CLI Example:

```
salt '*' win_iis.set_webapp_settings name='app0' site='site0' settings="{
↪'physicalPath': 'C:\site0', 'apppool': 'site0'}"
```

`salt.modules.win_iis.`**`set_webconfiguration_settings`**(*name*, *settings*)

Set the value of the setting for an IIS container.

Parameters

- **name** (*str*) – The PSPath of the IIS webconfiguration settings.
- **settings** (*list*) – A list of dictionaries containing setting name, filter and value.

Returns True if successful, otherwise False

Return type bool

CLI Example:

```
salt '*' win_iis.set_webconfiguration_settings name='IIS:\' settings="[{'name':
↪'enabled', 'filter': 'system.webServer/security/authentication/
↪anonymousAuthentication', 'value': False}]"
```

`salt.modules.win_iis.`**`start_apppool`**(*name*)

Start an IIS application pool.

New in version 2017.7.0.

Parameters **name** (*str*) – The name of the App Pool to start.

Returns True if successful, otherwise False

Return type bool

CLI Example:

```
salt '*' win_iis.start_apppool name='MyTestPool'
```

`salt.modules.win_iis.`**`start_site`**(*name*)

Start a Web Site in IIS.

New in version 2017.7.0.

Parameters **name** (*str*) – The name of the website to start.

Returns True if successful, otherwise False

Return type bool

CLI Example:

```
salt '*' win_iis.start_site name='My Test Site'
```

`salt.modules.win_iis.`**`stop_apppool`**(*name*)

Stop an IIS application pool.

New in version 2017.7.0.

Parameters **name** (*str*) – The name of the App Pool to stop.

Returns True if successful, otherwise False

Return type bool

CLI Example:

```
salt '*' win_iis.stop_apppool name='MyTestPool'
```

`salt.modules.win_iis.`**`stop_site`**`(`*name*`)`

Stop a Web Site in IIS.

New in version 2017.7.0.

Parameters **`name`** (*`str`*) – The name of the website to stop.

Returns True if successful, otherwise False

Return type bool

CLI Example:

```
salt '*' win_iis.stop_site name='My Test Site'
```

21.9.465 salt.modules.win_ip

The networking module for Windows based systems

`salt.modules.win_ip.`**`disable`**`(`*iface*`)`

Disable an interface

CLI Example:

```
salt -G 'os_family:Windows' ip.disable 'Local Area Connection #2'
```

`salt.modules.win_ip.`**`enable`**`(`*iface*`)`

Enable an interface

CLI Example:

```
salt -G 'os_family:Windows' ip.enable 'Local Area Connection #2'
```

`salt.modules.win_ip.`**`get_all_interfaces`**`()`

Return configs for all interfaces

CLI Example:

```
salt -G 'os_family:Windows' ip.get_all_interfaces
```

`salt.modules.win_ip.`**`get_default_gateway`**`()`

Set DNS source to DHCP on Windows

CLI Example:

```
salt -G 'os_family:Windows' ip.get_default_gateway
```

`salt.modules.win_ip.`**`get_interface`**`(`*iface*`)`

Return the configuration of a network interface

CLI Example:

```
salt -G 'os_family:Windows' ip.get_interface 'Local Area Connection'
```

`salt.modules.win_ip.`**`get_subnet_length`**`(`*mask*`)`

Convenience function to convert the netmask to the CIDR subnet length

CLI Example:

```
salt -G 'os_family:Windows' ip.get_subnet_length 255.255.255.0
```

`salt.modules.win_ip.`**`is_disabled`**`(`*iface*`)`

Returns `True` if interface is disabled, otherwise `False`

CLI Example:

```
salt -G 'os_family:Windows' ip.is_disabled 'Local Area Connection #2'
```

`salt.modules.win_ip.`**`is_enabled`**`(`*iface*`)`

Returns `True` if interface is enabled, otherwise `False`

CLI Example:

```
salt -G 'os_family:Windows' ip.is_enabled 'Local Area Connection #2'
```

`salt.modules.win_ip.`**`raw_interface_configs`**`()`

Return raw configs for all interfaces

CLI Example:

```
salt -G 'os_family:Windows' ip.raw_interface_configs
```

`salt.modules.win_ip.`**`set_dhcp_all`**`(`*iface*`)`

Set both IP Address and DNS to DHCP

CLI Example:

```
salt -G 'os_family:Windows' ip.set_dhcp_all 'Local Area Connection'
```

`salt.modules.win_ip.`**`set_dhcp_dns`**`(`*iface*`)`

Set DNS source to DHCP on Windows

CLI Example:

```
salt -G 'os_family:Windows' ip.set_dhcp_dns 'Local Area Connection'
```

`salt.modules.win_ip.`**`set_dhcp_ip`**`(`*iface*`)`

Set Windows NIC to get IP from DHCP

CLI Example:

```
salt -G 'os_family:Windows' ip.set_dhcp_ip 'Local Area Connection'
```

`salt.modules.win_ip.`**`set_static_dns`**`(`*iface, *addrs*`)`

Set static DNS configuration on a Windows NIC

Parameters

- **`iface`** (*`str`*) – The name of the interface to set
- **`addrs`** – One or more DNS servers to be added. To clear the list of DNS servers pass an empty list (`[]`). If undefined or `None` no changes will be made.

Returns A dictionary containing the new DNS settings

Return type dict

CLI Example:

```
salt -G 'os_family:Windows' ip.set_static_dns 'Local Area Connection' '192.168.1.1
↪'
salt -G 'os_family:Windows' ip.set_static_dns 'Local Area Connection' '192.168.1.
↪252' '192.168.1.253'
```

`salt.modules.win_ip.`**`set_static_ip`**(*iface*, *addr*, *gateway=None*, *append=False*)

Set static IP configuration on a Windows NIC

iface The name of the interface to manage

addr IP address with subnet length (ex. `10.1.2.3/24`). The *`ip.get_subnet_length`* function can be used to calculate the subnet length from a netmask.

gateway [None] If specified, the default gateway will be set to this value.

append [False] If `True`, this IP address will be added to the interface. Default is `False`, which overrides any existing configuration for the interface and sets `addr` as the only address on the interface.

CLI Example:

```
salt -G 'os_family:Windows' ip.set_static_ip 'Local Area Connection' 10.1.2.3/24␣
↪gateway=10.1.2.1
salt -G 'os_family:Windows' ip.set_static_ip 'Local Area Connection' 10.1.2.4/24␣
↪append=True
```

21.9.466 salt.modules.win_lgpo module

Manage Local Policy on Windows

This module allows configuring local group policy (i.e. `gpedit.msc`) on a Windows server.

New in version 2016.11.0.

Administrative Templates

Administrative template policies are dynamically read from ADMX/ADML files on the server.

Windows Settings

Policies contained in the "Windows Settings" section of the `gpedit.msc` GUI are statically defined in this module. Each policy is configured for the section (Machine/User) in the module's _policy_info class. The `_policy_info` class contains a "policies" dict on how the module will configure the policy, where the policy resides in the GUI (for display purposes), data validation data, data transformation data, etc.

Current known limitations

- At this time, start/shutdown scripts policies are displayed, but are not configurable.
- Not all "Security Settings" policies exist in the _policy_info class

  depends

  - pywin32 Python module
  - lxml
  - uuid
  - struct
  - salt.utils.win_reg

`salt.modules.win_lgpo.`**`get`**(*policy_class=None*, *return_full_policy_names=True*, *hierarchical_return=False*, *adml_language='en-US'*, *return_not_configured=False*)

Get a policy value

Parameters

- **policy_class** (*str*) – Some policies are both user and computer, by default all policies will be pulled, but this can be used to retrieve only a specific policy class User/USER/user = retrieve user policies Machine/MACHINE/machine/Computer/COMPUTER/computer = retrieve machine/computer policies
- **return_full_policy_names** (*bool*) – True/False to return the policy name as it is seen in the `gpedit.msc` GUI or to only return the policy key/id.
- **hierarchical_return** (*bool*) – True/False to return the policy data in the hierarchy as seen in the `gpedit.msc` GUI. The default of False will return data split only into User/Computer configuration sections
- **adml_language** (*str*) – The ADML language to use for processing display/descriptive names and enumeration values of ADMX template data, defaults to en-US
- **return_not_configured** (*bool*) – Include Administrative Template policies that are 'Not Configured' in the return data

Returns A dictionary containing the policy values for the specified class

Return type dict

CLI Example:

```
salt '*' lgpo.get machine return_full_policy_names=True
```

`salt.modules.win_lgpo.`**`get_policy`**(*policy_name*, *policy_class*, *adml_language='en-US'*, *return_value_only=True*, *return_full_policy_names=True*, *hierarchical_return=False*)

Get the current settings for a single policy on the machine

Parameters

- **policy_name** (*str*) – The name of the policy to retrieve. Can be the any of the names or alieses returned by `lgpo.get_policy_info`
- **policy_class** (*str*) – The policy class. Must be one of `machine` or `user`
- **adml_language** (*str*) – The language code for the adml file to use for localization. The default is `en-US`
- **return_value_only** (*bool*) – `True` will return only the value for the policy, without the name of the policy. `return_full_policy_names` and `hierarchical_return` will be ignored. Default is `True`
- **return_full_policy_names** (*bool*) – Returns the full policy name regardless of what was passed in `policy_name`

  Note: This setting applies to sub-elements of the policy if they exist. The value passed in `policy_name` will always be used as the policy name when this setting is `False`

- **hierarchical_return** (*bool*) – Returns a hierarchical view of the policy showing its parents

Returns A dictionary containing the policy settings

Return type dict

CLI Example:

```
# Using the policy id
salt * lgpo.get_policy LockoutDuration machine
salt * lgpo.get_policy AutoUpdateCfg machine

# Using the full name
salt * lgpo.get_policy "Account lockout duration" machine
salt * lgpo.get_policy "Configure Automatic Updates" machine
```

(continues on next page)

(continued from previous page)

```
# Using full path and name
salt * lgpo.get_policy "Windows Components\Windows Update\Configure Automatic
↪Updates" machine
```

salt.modules.win_lgpo.**get_policy_info**(*policy_name, policy_class, adml_language='en-US'*)

Returns information about a specified policy

Parameters

- **policy_name** (*str*) – The name of the policy to lookup
- **policy_class** (*str*) – The class of policy, i.e. machine, user, both
- **adml_language** (*str*) – The ADML language to use for Administrative Template data lookup

Returns Information about the specified policy

Return type dict

CLI Example:

```
salt '*' lgpo.get_policy_info 'Maximum password age' machine
```

You can use `lgpo.get_policy_info` to get all the possible names that could be used in a state file or from the command line (along with elements that need to be set/etc). The key is to match the text you see in the `gpedit.msc` gui exactly, including quotes around words or phrases. The "full path" style is really only needed when there are multiple policies that use the same base name. For example, `Access data sources across domains` exists in ~10 different paths. If you put that through `get_policy_info` you'll get back a message that it is used for multiple policies and you need to be more specific.

CLI Example:

```
salt-call --local lgpo.get_policy_info ShellRemoveOrderPrints_2 machine

local:
    ----------
    message:
    policy_aliases:
        - Turn off the "Order Prints" picture task
        - ShellRemoveOrderPrints_2
        - System\Internet Communication Management\Internet Communication
↪settings\Turn off the "Order Prints" picture task
    policy_class:
        machine
    policy_elements:
    policy_found:
        True
    policy_name:
        ShellRemoveOrderPrints_2
    rights_assignment:
        False
```

Escaping can get tricky in cmd/Powershell. The following is an example of escaping in Powershell using backquotes:

```
PS>salt-call --local lgpo.get_policy_info "Turn off the `\`"Order Prints`\`"
↪picture task" machine

local:
    ----------
    message:
```

(continues on next page)

(continued from previous page)

```
    policy_aliases:
        - Turn off the "Order Prints" picture task
        - ShellRemoveOrderPrints_2
        - System\Internet Communication Management\Internet Communication
↪settings\Turn off the "Order Prints" picture task
    policy_class:
        machine
    policy_elements:
    policy_found:
        True
    policy_name:
        Turn off the "Order Prints" picture task
    rights_assignment:
        False
```

This function can then be used to get the options available for specifying Group Policy Objects to be used in state files. Based on the above any of these *should* be usable:

```
internet_communications_settings:
  lgpo.set:
    - computer_policy:
        Turn off the "Order Prints" picture task: Enabled
```

```
internet_communications_settings:
  lgpo.set:
    - computer_policy:
        ShellRemoveOrderPrints_2: Enabled
```

When using the full path, it might be a good idea to use single quotes around the path:

```
internet_communications_settings:
  lgpo.set:
    - computer_policy:
        'System\Internet Communication Management\Internet Communication
↪settings\Turn off the "Order Prints" picture task': 'Enabled'
```

If you struggle to find the policy from `get_policy_info` using the name as you see in `gpedit.msc`, the names such as "ShellRemoveOrderPrints_2" come from the `.admx` files. If you know nothing about `.admx/.adml` relationships (ADML holds what you see in the GUI, ADMX holds the more technical details), then this may be a little bit too much info, but here is an example with the above policy using Powershell:

```
PS>Get-ChildItem -Path C:\Windows\PolicyDefinitions -Recurse -Filter *.adml |
↪Select-String "Order Prints"

C:\windows\PolicyDefinitions\en-US\ICM.adml:152:      <string id=
↪"ShellRemoveOrderPrints">Turn off the "Order Prints" picture task</string>
C:\windows\PolicyDefinitions\en-US\ICM.adml:153:      <string id=
↪"ShellRemoveOrderPrints_Help">This policy setting specifies whether the "Order
↪Prints Online" task is available from Picture Tasks in Windows folders.
C:\windows\PolicyDefinitions\en-US\ICM.adml:155:The Order Prints Online Wizard is
↪used to download a list of providers and allow users to order prints online.
C:\windows\PolicyDefinitions\en-US\ICM.adml:157:If you enable this policy setting,
↪ the task "Order Prints Online" is removed from Picture Tasks in File Explorer
↪folders.
```

From this grep, we can see id "ShellRemoveOrderPrints" is the ID of the string used to describe this policy, then we search for it in the ADMX:

```
PS>Get-ChildItem -Path C:\Windows\PolicyDefinitions -Recurse -Filter *.admx |␣
↪Select-String "ShellRemoveOrderPrints"

C:\windows\PolicyDefinitions\ICM.admx:661:    <policy name=
↪"ShellRemoveOrderPrints_1" class="User" displayName="$(string.
↪ShellRemoveOrderPrints)" explainText="$(string.ShellRemoveOrderPrints_Help)"␣
↪key="Software\Microsoft\Windows\CurrentVersion\Policies\Explorer" valueName=
↪"NoOnlinePrintsWizard">
C:\windows\PolicyDefinitions\ICM.admx:671:    <policy name=
↪"ShellRemoveOrderPrints_2" class="Machine" displayName="$(string.
↪ShellRemoveOrderPrints)" explainText="$(string.ShellRemoveOrderPrints_Help)"␣
↪key="Software\Microsoft\Windows\CurrentVersion\Policies\Explorer" valueName=
↪"NoOnlinePrintsWizard">
```

Now we have two to pick from. And if you notice the `class="Machine"` and `class="User"` (which details if it is a computer policy or user policy respectively) the `ShellRemoveOrderPrints_2` is the "short name" we could use to pass through `get_policy_info` to see what the module itself is expecting.

`salt.modules.win_lgpo.`**`set_`**(*computer_policy=None*, *user_policy=None*, *cumulative_rights_assignments=True*, *adml_language='en-US'*)

Set a local server policy.

Parameters

- **`computer_policy`** (*dict*) – A dictionary of "policyname: value" pairs of computer policies to set. 'value' should be how it is displayed in the gpedit GUI, i.e. if a setting can be 'Enabled'/'Disabled', then that should be passed

  Administrative Template data may require dicts within dicts, to specify each element of the Administrative Template policy. Administrative Templates policies are always cumulative.

  Policy names can be specified in a number of ways based on the type of policy:

  Windows Settings Policies:

  These policies can be specified using the GUI display name or the key name from the _policy_info class in this module. The GUI display name is also contained in the _policy_info class in this module.

  Administrative Template Policies:

  These can be specified using the policy name as displayed in the GUI (case sensitive). Some policies have the same name, but a different location (for example, "Access data sources across domains"). These can be differentiated by the "path" in the GUI (for example, "Windows ComponentsInternet ExplorerInternet Control PanelSecurity PageInternet ZoneAccess data sources across domains").

  Additionally, policies can be specified using the "name" and "id" attributes from the ADMX files.

  For Administrative Templates that have policy elements, each element can be specified using the text string as seen in the GUI or using the ID attribute from the ADMX file. Due to the way some of the GUI text is laid out, some policy element names could include descriptive text that appears lbefore the policy element in the GUI.

  Use the get_policy_info function for the policy name to view

the element ID/names that the module will accept.
- **user_policy** (*dict*) – The same setup as the computer_policy, except with data to configure the local user policy.
- **cumulative_rights_assignments** (*bool*) – Determine how user rights assignment policies are configured.

  If True, user right assignment specifications are simply added to the existing policy

  If False, only the users specified will get the right (any existing will have the right revoked)
- **adml_language** (*str*) – The language files to use for looking up Administrative Template policy data (i.e. how the policy is displayed in the GUI). Defaults to 'en-US' (U.S. English).

Returns True is successful, otherwise False

Return type bool

CLI Example:

```
salt '*' lgpo.set computer_policy="{'LockoutDuration': 2, 'RestrictAnonymous':
↪'Enabled', 'AuditProcessTracking': 'Succes, Failure'}"
```

salt.modules.win_lgpo.**set_computer_policy**(*name, setting, cumulative_rights_assignments=True, adml_language='en-US'*)

Set a single computer policy

Parameters
- **name** (*str*) – The name of the policy to configure
- **setting** (*str*) – The setting to configure the named policy with
- **cumulative_rights_assignments** (*bool*) – Determine how user rights assignment policies are configured. If True, user right assignment specifications are simply added to the existing policy. If False, only the users specified will get the right (any existing will have the right revoked)
- **adml_language** (*str*) – The language files to use for looking up Administrative Template policy data (i.e. how the policy is displayed in the GUI). Defaults to 'en-US' (U.S. English).

Returns True if successful, otherwise False

Return type bool

CLI Example:

```
salt '*' lgpo.set_computer_policy LockoutDuration 1440
```

salt.modules.win_lgpo.**set_user_policy**(*name, setting, adml_language='en-US'*)

Set a single user policy

Parameters
- **name** (*str*) – The name of the policy to configure
- **setting** (*str*) – The setting to configure the named policy with
- **adml_language** (*str*) – The language files to use for looking up Administrative Template policy data (i.e. how the policy is displayed in the GUI). Defaults to 'en-US' (U.S. English).

Returns True if successful, Otherwise False

Return type bool

CLI Example:

```
salt '*' lgpo.set_user_policy "Control Panel\Display\Disable the Display Control
↪Panel" Enabled
```

21.9.467 salt.modules.win_license module

This module allows you to manage windows licensing via slmgr.vbs

```
salt '*' license.install XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
```

`salt.modules.win_license.`**`activate`**`()`

Attempt to activate the current machine via Windows Activation

CLI Example:

```
salt '*' license.activate
```

`salt.modules.win_license.`**`info`**`()`

Return information about the license, if the license is not correctly activated this will return None.

CLI Example:

```
salt '*' license.info
```

`salt.modules.win_license.`**`install`**`(`*product_key*`)`

Install the given product key

CLI Example:

```
salt '*' license.install XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
```

`salt.modules.win_license.`**`installed`**`(`*product_key*`)`

Check to see if the product key is already installed.
Note: This is not 100% accurate as we can only see the last 5 digits of the license.
CLI Example:

```
salt '*' license.installed XXXXX-XXXXX-XXXXX-XXXXX-XXXXX
```

`salt.modules.win_license.`**`licensed`**`()`

Return true if the current machine is licensed correctly

CLI Example:

```
salt '*' license.licensed
```

`salt.modules.win_license.`**`uninstall`**`()`

Uninstall the current product key

CLI Example:

```
salt '*' license.uninstall
```

21.9.468 salt.modules.win_network

Module for gathering and managing network information

`salt.modules.win_network.`**`connect`**`(`*host*, *port=None*, ***kwargs*`)`

Test connectivity to a host using a particular port from the minion.

New in version 2016.3.0.

CLI Example:

```
salt '*' network.connect archlinux.org 80

salt '*' network.connect archlinux.org 80 timeout=3

salt '*' network.connect archlinux.org 80 timeout=3 family=ipv4

salt '*' network.connect google-public-dns-a.google.com port=53 proto=udp␣
↪timeout=3
```

`salt.modules.win_network.`**`dig`**`(`*host*`)`

Performs a DNS lookup with dig

Note: dig must be installed on the Windows minion

CLI Example:

```
salt '*' network.dig archlinux.org
```

`salt.modules.win_network.`**`get_route`**`(`*ip*`)`

Return routing information for given destination ip

New in version 2016.11.5.

CLI Example:

```
salt '*' network.get_route 10.10.10.10
```

`salt.modules.win_network.`**`hw_addr`**`(`*iface*`)`

Return the hardware address (a.k.a. MAC address) for a given interface

CLI Example:

```
salt '*' network.hw_addr 'Wireless Connection #1'
```

`salt.modules.win_network.`**`hwaddr`**`(`*iface*`)`

This function is an alias of `hw_addr`.

> Return the hardware address (a.k.a. MAC address) for a given interface
>
> CLI Example:
>
> ```
> salt '*' network.hw_addr 'Wireless Connection #1'
> ```

`salt.modules.win_network.`**`in_subnet`**`(`*cidr*`)`

Returns True if host is within specified subnet, otherwise False

CLI Example:

```
salt '*' network.in_subnet 10.0.0.0/16
```

`salt.modules.win_network.`**`interfaces`**`()`

Return a dictionary of information about all the interfaces on the minion

CLI Example:

```
salt '*' network.interfaces
```

`salt.modules.win_network.`**`interfaces_names`**`()`

Return a list of all the interfaces names

CLI Example:

```
salt '*' network.interfaces_names
```

`salt.modules.win_network.`**`ip_addrs`**(*interface=None*, *include_loopback=False*, *cidr=None*, *type=None*)

Returns a list of IPv4 addresses assigned to the host.

interface Only IP addresses from that interface will be returned.

include_loopback [False] Include loopback 127.0.0.1 IPv4 address.

cidr

Describes subnet using CIDR notation and only IPv4 addresses that belong to this subnet will be returned.

Changed in version 2019.2.0.

type If option set to 'public' then only public addresses will be returned. Ditto for 'private'.

Changed in version 2019.2.0.

CLI Example:

```
salt '*' network.ip_addrs
salt '*' network.ip_addrs cidr=10.0.0.0/8
salt '*' network.ip_addrs cidr=192.168.0.0/16 type=private
```

`salt.modules.win_network.`**`ip_addrs6`**(*interface=None*, *include_loopback=False*, *cidr=None*)

Returns a list of IPv6 addresses assigned to the host.

interface Only IP addresses from that interface will be returned.

include_loopback [False] Include loopback ::1 IPv6 address.

cidr Describes subnet using CIDR notation and only IPv6 addresses that belong to this subnet will be returned.

Changed in version 2019.2.0.

CLI Example:

```
salt '*' network.ip_addrs6
salt '*' network.ip_addrs6 cidr=2000::/3
```

`salt.modules.win_network.`**`ipaddrs`**(*interface=None*, *include_loopback=False*, *cidr=None*, *type=None*)

This function is an alias of `ip_addrs`.

Returns a list of IPv4 addresses assigned to the host.

interface Only IP addresses from that interface will be returned.

include_loopback [False] Include loopback 127.0.0.1 IPv4 address.

cidr

Describes subnet using CIDR notation and only IPv4 addresses that belong to this subnet will be returned.

Changed in version 2019.2.0.

type If option set to 'public' then only public addresses will be returned. Ditto for 'private'.

Changed in version 2019.2.0.

CLI Example:

```
salt '*' network.ip_addrs
salt '*' network.ip_addrs cidr=10.0.0.0/8
salt '*' network.ip_addrs cidr=192.168.0.0/16 type=private
```

`salt.modules.win_network.`**`ipaddrs6`**(*interface=None*, *include_loopback=False*, *cidr=None*)

This function is an alias of `ip_addrs6`.

Returns a list of IPv6 addresses assigned to the host.

interface Only IP addresses from that interface will be returned.

include_loopback [False] Include loopback ::1 IPv6 address.

cidr Describes subnet using CIDR notation and only IPv6 addresses that belong to this subnet will be returned.

Changed in version 2019.2.0.

CLI Example:

```
salt '*' network.ip_addrs6
salt '*' network.ip_addrs6 cidr=2000::/3
```

`salt.modules.win_network.`**`is_private`**`(`*ip_addr*`)`

Check if the given IP address is a private address

New in version 2019.2.0.

CLI Example:

```
salt '*' network.is_private 10.0.0.3
```

`salt.modules.win_network.`**`netstat`**`()`

Return information on open ports and states

CLI Example:

```
salt '*' network.netstat
```

`salt.modules.win_network.`**`nslookup`**`(`*host*`)`

Query DNS for information about a domain or ip address

CLI Example:

```
salt '*' network.nslookup archlinux.org
```

`salt.modules.win_network.`**`ping`**`(`*host, timeout=False, return_boolean=False*`)`

Performs a ping to a host

CLI Example:

```
salt '*' network.ping archlinux.org
```

New in version 2016.11.0.

Return a True or False instead of ping output.

```
salt '*' network.ping archlinux.org return_boolean=True
```

Set the time to wait for a response in seconds.

```
salt '*' network.ping archlinux.org timeout=3
```

`salt.modules.win_network.`**`subnets`**`()`

Returns a list of subnets to which the host belongs

CLI Example:

```
salt '*' network.subnets
```

`salt.modules.win_network.`**`traceroute`**`(`*host*`)`

Performs a traceroute to a 3rd party host

CLI Example:

```
salt '*' network.traceroute archlinux.org
```

21.9.469 salt.modules.win_ntp

Management of NTP servers on Windows

New in version 2014.1.0.

`salt.modules.win_ntp.`**`get_servers`**`()`

Get list of configured NTP servers

CLI Example:

```
salt '*' ntp.get_servers
```

`salt.modules.win_ntp.`**`set_servers`**`(`**servers*`)`

Set Windows to use a list of NTP servers

CLI Example:

```
salt '*' ntp.set_servers 'pool.ntp.org' 'us.pool.ntp.org'
```

21.9.470 salt.modules.win_path

Manage the Windows System PATH

Note that not all Windows applications will rehash the PATH environment variable, Only the ones that listen to the WM_SETTINGCHANGE message http://support.microsoft.com/kb/104011

`salt.modules.win_path.`**`add`**`(`*path, index=None, **kwargs*`)`

Add the directory to the SYSTEM path in the index location. Returns `True` if successful, otherwise `False`.

path Directory to add to path

index Optionally specify an index at which to insert the directory

rehash [True] If the registry was updated, and this value is set to `True`, sends a WM_SETTINGCHANGE broadcast to refresh the environment variables. Set this to `False` to skip this broadcast.

CLI Examples:

```
# Will add to the beginning of the path
salt '*' win_path.add 'c:\python27' 0

# Will add to the end of the path
salt '*' win_path.add 'c:\python27' index='-1'
```

`salt.modules.win_path.`**`exists`**`(`*path*`)`

Check if the directory is configured in the SYSTEM path Case-insensitive and ignores trailing backslash

Returns boolean True if path exists, False if not

CLI Example:

```
salt '*' win_path.exists 'c:\python27'
salt '*' win_path.exists 'c:\python27\'
salt '*' win_path.exists 'C:\pyThon27'
```

`salt.modules.win_path.`**`get_path`**`()`

Returns a list of items in the SYSTEM path

CLI Example:

```
salt '*' win_path.get_path
```

`salt.modules.win_path.`**`rehash`**`()`

Send a WM_SETTINGCHANGE Broadcast to Windows to refresh the Environment variables for new processes.

Note: This will only affect new processes that aren't launched by services. To apply changes to the path to services, the host must be restarted. The `salt-minion`, if running as a service, will not see changes to the environment until the system is restarted. See MSDN Documentation

---

CLI Example:

```
salt '*' win_path.rehash
```

`salt.modules.win_path.`**`remove`**`(`*path*, ***kwargs*`)`

Remove the directory from the SYSTEM path

Returns boolean True if successful, False if unsuccessful

rehash [True] If the registry was updated, and this value is set to `True`, sends a WM_SETTINGCHANGE broadcast to refresh the environment variables. Set this to `False` to skip this broadcast.

CLI Example:

```
# Will remove C:\Python27 from the path
salt '*' win_path.remove 'c:\\python27'
```

21.9.471 salt.modules.win_pkg

A module to manage software on Windows

Important: If you feel that Salt should be using this module to manage packages on a minion, and it is using a different module (or gives an error similar to *'pkg.install' is not available*), see *here*.

The following functions require the existence of a *windows repository* metadata DB, typically created by running *`pkg.refresh_db`*:

- *`pkg.get_repo_data`*
- *`pkg.install`*
- *`pkg.latest_version`*
- *`pkg.list_available`*
- *`pkg.list_pkgs`*
- *`pkg.list_upgrades`*
- *`pkg.remove`*

If a metadata DB does not already exist and one of these functions is run, then one will be created from the repo SLS files that are present.

As the creation of this metadata can take some time, the *`winrepo_cache_expire_min`* minion config option can be used to suppress refreshes when the metadata is less than a given number of seconds old.

Note: Version numbers can be `version number string`, `latest` and `Not Found`, where `Not Found` means this module was not able to determine the version of the software installed, it can also be used as the version number in sls definitions file in these cases. Versions numbers are sorted in order of 0, `Not Found`, `order version numbers`, ..., `latest`.

`salt.modules.win_pkg.`**`compare_versions`**(*ver1=", oper='==', ver2="*)

Compare software package versions. Made public for use with Jinja

Parameters
- **`ver1`** (*`str`*) – A software version to compare
- **`oper`** (*`str`*) – The operand to use to compare
- **`ver2`** (*`str`*) – A software version to compare

Returns True if the comparison is valid, otherwise False

Return type bool

CLI Example:

```
salt '*' pkg.compare_versions 1.2 >= 1.3
```

`salt.modules.win_pkg.`**`genrepo`**(***kwargs*)

Generate package metadata db based on files within the winrepo_source_dir

Kwargs:

saltenv (str): Salt environment. Default: `base`

verbose (bool): Return verbose data structure which includes 'success_list', a list of all sls files and the package names contained within. Default `False`.

failhard (bool): If `True`, an error will be raised if any repo SLS files failed to process. If `False`, no error will be raised, and a dictionary containing the full results will be returned.

Note:
- Hidden directories (directories beginning with '.', such as '*.git*') will be ignored.

Returns A dictionary of the results of the command

Return type dict

CLI Example:

```
salt-run pkg.genrepo
salt -G 'os:windows' pkg.genrepo verbose=true failhard=false
salt -G 'os:windows' pkg.genrepo saltenv=base
```

`salt.modules.win_pkg.`**`get_package_info`**(*name, saltenv='base'*)

Return package info. Returns empty map if package not available.

`salt.modules.win_pkg.`**`get_repo_data`**(*saltenv='base'*)

Returns the existing package metadata db. Will create it, if it does not exist, however will not refresh it.

Parameters **`saltenv`** (*`str`*) – Salt environment. Default `base`

Returns A dict containing contents of metadata db.

Return type dict

CLI Example:

```
salt '*' pkg.get_repo_data
```

`salt.modules.win_pkg.`**`install`**(*name=None, refresh=False, pkgs=None, **kwargs*)

Install the passed package(s) on the system using winrepo

Parameters

- **name** (*str*) – The name of a single package, or a comma-separated list of packages to install. (no spaces after the commas)
- **refresh** (*bool*) – Boolean value representing whether or not to refresh the winrepo db. Default `False`.
- **pkgs** (*list*) – A list of packages to install from a software repository. All packages listed under `pkgs` will be installed via a single command.

  You can specify a version by passing the item as a dict:

  CLI Example:

  ```
  # will install the latest version of foo and bar
  salt '*' pkg.install pkgs='["foo", "bar"]'

  # will install the latest version of foo and version 1.2.3 of
  ↪bar
  salt '*' pkg.install pkgs='["foo", {"bar": "1.2.3"}]'
  ```

Kwargs:

version (str): The specific version to install. If omitted, the latest version will be installed. Recommend for use when installing a single package.

If passed with a list of packages in the `pkgs` parameter, the version will be ignored.

CLI Example:

```
# Version is ignored
salt '*' pkg.install pkgs="['foo', 'bar']" version=1.2.3
```

If passed with a comma separated list in the `name` parameter, the version will apply to all packages in the list.

CLI Example:

```
# Version 1.2.3 will apply to packages foo and bar
salt '*' pkg.install foo,bar version=1.2.3
```

extra_install_flags (str): Additional install flags that will be appended to the `install_flags` defined in the software definition file. Only applies when single package is passed.

saltenv (str): Salt environment. Default 'base'

report_reboot_exit_codes (bool): If the installer exits with a recognized exit code indicating that a reboot is required, the module function

win_system.set_reboot_required_witnessed

will be called, preserving the knowledge of this event for the remainder of the current boot session. For the time being, 3010 is the only recognized exit code. The value of this param defaults to True.

New in version 2016.11.0.

Returns

Return a dict containing the new package names and versions. If the package is already installed, an empty dict is returned.

If the package is installed by `pkg.install`:

```
{'<package>': {'old': '<old-version>',
               'new': '<new-version>'}}
```

Return type dict

The following example will refresh the winrepo and install a single package, 7zip.

CLI Example:

```
salt '*' pkg.install 7zip refresh=True
```

CLI Example:

```
salt '*' pkg.install 7zip
salt '*' pkg.install 7zip,filezilla
salt '*' pkg.install pkgs='["7zip","filezilla"]'
```

WinRepo Definition File Examples:

The following example demonstrates the use of `cache_file`. This would be used if you have multiple installers in the same directory that use the same `install.ini` file and you don't want to download the additional installers.

```
ntp:
  4.2.8:
    installer: 'salt://win/repo/ntp/ntp-4.2.8-win32-setup.exe'
    full_name: Meinberg NTP Windows Client
    locale: en_US
    reboot: False
    cache_file: 'salt://win/repo/ntp/install.ini'
    install_flags: '/
↪USEFILE=C:\salt\var\cache\salt\minion\files\base\win\repo\ntp\install.ini'
    uninstaller: 'NTP/uninst.exe'
```

The following example demonstrates the use of `cache_dir`. It assumes a file named `install.ini` resides in the same directory as the installer.

```
ntp:
  4.2.8:
    installer: 'salt://win/repo/ntp/ntp-4.2.8-win32-setup.exe'
    full_name: Meinberg NTP Windows Client
    locale: en_US
    reboot: False
    cache_dir: True
    install_flags: '/
↪USEFILE=C:\salt\var\cache\salt\minion\files\base\win\repo\ntp\install.ini'
    uninstaller: 'NTP/uninst.exe'
```

`salt.modules.win_pkg.`**`latest_version`**(**names*, ***kwargs*)

Return the latest version of the named package available for upgrade or installation. If more than one package name is specified, a dict of name/version pairs is returned.

If the latest version of a given package is already installed, an empty string will be returned for that package.

Note: Since this is looking for the latest version available, a refresh_db will be triggered by default. This can take some time. To avoid this set `refresh` to `False`.

Parameters **`names`** (*`str`*) – A single or multiple names to lookup

Kwargs: saltenv (str): Salt environment. Default `base` refresh (bool): Refresh package metadata. Default `True`

Returns A dictionary of packages with the latest version available

Return type dict

CLI Example:

```
salt '*' pkg.latest_version <package name>
salt '*' pkg.latest_version <package1> <package2> <package3> ...
```

`salt.modules.win_pkg.`**`list_available`**(**names*, ***kwargs*)

Return a list of available versions of the specified package.

Parameters **`names`** (*`str`*) – One or more package names

Kwargs:

saltenv (str): The salt environment to use. Default `base`.

refresh (bool): Refresh package metadata. Default `False`.

return_dict_always (bool): Default `False` dict when a single package name is queried.

Returns The package name with its available versions

Return type dict

```
{'<package name>': ['<version>', '<version>', ]}
```

CLI Example:

```
salt '*' pkg.list_available <package name> return_dict_always=True
salt '*' pkg.list_available <package name01> <package name02>
```

`salt.modules.win_pkg.`**`list_pkgs`**(*versions_as_list=False*, *include_components=True*, *include_updates=True*, ***kwargs*)

List the packages currently installed.

Note: To view installed software as displayed in the Add/Remove Programs, set `include_components` and `include_updates` to False.

Parameters

- **`versions_as_list`** (*`bool`*) – Returns the versions as a list
- **`include_components`** (*`bool`*) – Include sub components of installed software. Default is `True`
- **`include_updates`** (*`bool`*) – Include software updates and Windows updates. Default is `True`

Kwargs:

saltenv (str): The salt environment to use. Default `base`

refresh (bool): Refresh package metadata. Default `False`

Returns A dictionary of installed software with versions installed

Return type dict

```
{'<package_name>': '<version>'}
```

CLI Example:

```
salt '*' pkg.list_pkgs
salt '*' pkg.list_pkgs versions_as_list=True
```

`salt.modules.win_pkg.`**`list_upgrades`**(*refresh=True*, ***kwargs*)

List all available package upgrades on this system

Parameters **`refresh`** (*`bool`*) – Refresh package metadata. Default `True`

Kwargs: saltenv (str): Salt environment. Default `base`

Returns A dictionary of packages with available upgrades

Return type dict

CLI Example:

```
salt '*' pkg.list_upgrades
```

`salt.modules.win_pkg.`**`purge`**`(`*name=None*, *pkgs=None*, ***kwargs*`)`

Package purges are not supported, this function is identical to `remove()`.

New in version 0.16.0.

Parameters

- **`name`** (*`str`*) – The name of the package to be deleted.
- **`version`** (*`str`*) – The version of the package to be deleted. If this option is used in combination with the `pkgs` option below, then this version will be applied to all targeted packages.
- **`pkgs`** (*`list`*) – A list of packages to delete. Must be passed as a python list. The `name` parameter will be ignored if this option is passed.

Kwargs: saltenv (str): Salt environment. Default `base` refresh (bool): Refresh package metadata. Default `False`

Returns A dict containing the changes.

Return type dict

CLI Example:

```
salt '*' pkg.purge <package name>
salt '*' pkg.purge <package1>,<package2>,<package3>
salt '*' pkg.purge pkgs='["foo", "bar"]'
```

`salt.modules.win_pkg.`**`refresh_db`**`(`***kwargs*`)`

Generates the local software metadata database (*winrepo.p*) on the minion. The database is stored in a serialized format located by default at the following location:

`C:\salt\var\cache\salt\minion\files\base\win\repo-ng\winrepo.p`

This module performs the following steps to generate the software metadata database:

- Fetch the package definition files (.sls) from *winrepo_source_dir* (default *salt://win/repo-ng*) and cache them in *<cachedir>files<saltenv><winrepo_source_dir>* (default: `C:\salt\var\cache\salt\minion\files\base\win\repo-ng`)
- Call *`pkg.genrepo`* to parse the package definition files and generate the repository metadata database file (*winrepo.p*)
- Return the report received from *`pkg.genrepo`*

The default winrepo directory on the master is */srv/salt/win/repo-ng*. All files that end with *.sls* in this and all subdirectories will be used to generate the repository metadata database (*winrepo.p*).

Note:

- Hidden directories (directories beginning with '.', such as '*.git*') will be ignored.

Note: There is no need to call *pkg.refresh_db* every time you work with the pkg module. Automatic refresh will occur based on the following minion configuration settings:

- *winrepo_cache_expire_min*
- *winrepo_cache_expire_max*

However, if the package definition files have changed, as would be the case if you are developing a new package definition, this function should be called to ensure the minion has the latest information about packages available to it.

Warning: Directories and files fetched from <winrepo_source_dir> (*/srv/salt/win/repo-ng*) will be processed in alphabetical order. If two or more software definition files contain the same name, the last one processed replaces all data from the files processed before it.

For more information see *Windows Software Repository*

Arguments:

saltenv (str): Salt environment. Default: `base`

verbose (bool): Return a verbose data structure which includes 'success_list', a list of all sls files and the package names contained within. Default is 'False'

failhard (bool): If `True`, an error will be raised if any repo SLS files fails to process. If `False`, no error will be raised, and a dictionary containing the full results will be returned.

Returns A dictionary containing the results of the database refresh.

Return type dict

Note: A result with a *total: 0* generally means that the files are in the wrong location on the master. Try running the following command on the minion: *salt-call -l debug pkg.refresh saltenv=base*

Warning: When calling this command from a state using *module.run* be sure to pass *failhard: False*. Otherwise the state will report failure if it encounters a bad software definition file.

CLI Example:

```
salt '*' pkg.refresh_db
salt '*' pkg.refresh_db saltenv=base
```

`salt.modules.win_pkg.`**`remove`**(*name=None*, *pkgs=None*, ***kwargs*)

Remove the passed package(s) from the system using winrepo

New in version 0.16.0.

Parameters

- **name** (*str*) – The name(s) of the package(s) to be uninstalled. Can be a single package or a comma delimited list of packages, no spaces.
- **pkgs** (*list*) – A list of packages to delete. Must be passed as a python list. The `name` parameter will be ignored if this option is passed.

Kwargs:

version (str): The version of the package to be uninstalled. If this option is used to to uninstall multiple packages, then this version will be applied to all targeted packages. Recommended using only when uninstalling a single package. If this parameter is omitted, the latest version will be uninstalled.

saltenv (str): Salt environment. Default `base` refresh (bool): Refresh package metadata. Default `False`

Returns

Returns a dict containing the changes.

If the package is removed by `pkg.remove`:

{'<package>': {'old': '<old-version>', 'new': '<new-version>'}}

If the package is already uninstalled:

{'<package>': {'current': 'not installed'}}

Return type dict

CLI Example:

```
salt '*' pkg.remove <package name>
salt '*' pkg.remove <package1>,<package2>,<package3>
salt '*' pkg.remove pkgs='["foo", "bar"]'
```

`salt.modules.win_pkg.`**`upgrade`**`(`***kwargs*`)`

Upgrade all software. Currently not implemented

Kwargs: saltenv (str): The salt environment to use. Default `base`. refresh (bool): Refresh package metadata. Default `True`.

Note: This feature is not yet implemented for Windows.

Returns Empty dict, until implemented

Return type dict

CLI Example:

```
salt '*' pkg.upgrade
```

`salt.modules.win_pkg.`**`upgrade_available`**`(`*name, **kwargs*`)`

Check whether or not an upgrade is available for a given package

Parameters **name** (*str*) – The name of a single package

Kwargs: refresh (bool): Refresh package metadata. Default `True` saltenv (str): The salt environment. Default `base`

Returns True if new version available, otherwise False

Return type bool

CLI Example:

```
salt '*' pkg.upgrade_available <package name>
```

`salt.modules.win_pkg.`**`version`**`(`**names, **kwargs*`)`

Returns a string representing the package version or an empty string if not installed. If more than one package name is specified, a dict of name/version pairs is returned.

Parameters **name** (*str*) – One or more package names

Kwargs: saltenv (str): The salt environment to use. Default `base`. refresh (bool): Refresh package metadata. Default `False`.

Returns version string when a single package is specified. dict: The package name(s) with the installed versions.

Return type str

```
{['<version>', '<version>', ]} OR
{'<package name>': ['<version>', '<version>', ]}
```

CLI Example:

```
salt '*' pkg.version <package name>
salt '*' pkg.version <package name01> <package name02>
```

21.9.472 salt.modules.win_pki module

Microsoft certificate management via the PKI Client PowerShell module. https://technet.microsoft.com/en-us/itpro/powershell/windows/pkiclient/pkiclient

The PKI Client PowerShell module is only available on Windows 8+ and Windows Server 2012+. https://technet.microsoft.com/en-us/library/hh848636(v=wps.620).aspx

platform Windows

depends

- PowerShell 4
- PKI Client Module (Windows 8+ / Windows Server 2012+)

New in version 2016.11.0.

`salt.modules.win_pki.`**`export_cert`**(*name, thumbprint, cert_format='cer', context='LocalMachine', store='My', password=''*)

Export the certificate to a file from the given certificate store.

Parameters

- **name** (*str*) – The destination path for the exported certificate file.
- **thumbprint** (*str*) – The thumbprint value of the target certificate.
- **cert_format** (*str*) – The certificate format. Specify 'cer' for X.509, or 'pfx' for PKCS #12.
- **context** (*str*) – The name of the certificate store location context.
- **store** (*str*) – The name of the certificate store.
- **password** (*str*) – The password of the certificate. Only applicable to pfx format. Note that if used interactively, the password will be seen by all minions. To protect the password, use a state and get the password from pillar.

Returns A boolean representing whether all changes succeeded.

Return type bool

CLI Example:

```
salt '*' win_pki.export_cert name='C:\certs\example.cer' thumbprint='AAA000'
```

`salt.modules.win_pki.`**`get_cert_file`**(*name, cert_format='cer', password=''*)

Get the details of the certificate file.

Parameters

- **name** (*str*) – The filesystem path of the certificate file.
- **cert_format** (*str*) – The certificate format. Specify 'cer' for X.509, or 'pfx' for PKCS #12.
- **password** (*str*) – The password of the certificate. Only applicable to pfx format. Note that if used interactively, the password will be seen by all minions. To protect the password, use a state and get the password from pillar.

Returns A dictionary of the certificate thumbprints and properties.

Return type dict

CLI Example:

```
salt '*' win_pki.get_cert_file name='C:\certs\example.cer'
```

`salt.modules.win_pki.`**`get_certs`**(*context='LocalMachine', store='My'*)

Get the available certificates in the given store.

Parameters

- **context** (*str*) – The name of the certificate store location context.
- **store** (*str*) – The name of the certificate store.

Returns A dictionary of the certificate thumbprints and properties.

Return type dict

CLI Example:

```
salt '*' win_pki.get_certs
```

`salt.modules.win_pki.`**`get_stores`**`()`

Get the certificate location contexts and their corresponding stores.

Returns A dictionary of the certificate location contexts and stores.

Return type dict

CLI Example:

```
salt '*' win_pki.get_stores
```

`salt.modules.win_pki.`**`import_cert`**`(`*name, cert_format='cer', context='LocalMachine', store='My', exportable=True, password='', saltenv='base'*`)`

Import the certificate file into the given certificate store.

Parameters

- **name** (*str*) – The path of the certificate file to import.
- **cert_format** (*str*) – The certificate format. Specify 'cer' for X.509, or 'pfx' for PKCS #12.
- **context** (*str*) – The name of the certificate store location context.
- **store** (*str*) – The name of the certificate store.
- **exportable** (*bool*) – Mark the certificate as exportable. Only applicable to pfx format.
- **password** (*str*) – The password of the certificate. Only applicable to pfx format. Note that if used interactively, the password will be seen by all minions. To protect the password, use a state and get the password from pillar.
- **saltenv** (*str*) – The environment the file resides in.

Returns A boolean representing whether all changes succeeded.

Return type bool

CLI Example:

```
salt '*' win_pki.import_cert name='salt://cert.cer'
```

`salt.modules.win_pki.`**`remove_cert`**`(`*thumbprint, context='LocalMachine', store='My'*`)`

Remove the certificate from the given certificate store.

Parameters

- **thumbprint** (*str*) – The thumbprint value of the target certificate.
- **context** (*str*) – The name of the certificate store location context.
- **store** (*str*) – The name of the certificate store.

Returns A boolean representing whether all changes succeeded.

Return type bool

CLI Example:

```
salt '*' win_pki.remove_cert thumbprint='AAA000'
```

`salt.modules.win_pki.`**`test_cert`**`(`*thumbprint, context='LocalMachine', store='My', untrusted_root=False, dns_name='', eku=''*`)`

Check the certificate for validity.

Parameters

- **thumbprint** (*str*) – The thumbprint value of the target certificate.
- **context** (*str*) – The name of the certificate store location context.
- **store** (*str*) – The name of the certificate store.
- **untrusted_root** (*bool*) – Whether the root certificate is required to be trusted in chain building.
- **dns_name** (*str*) – The DNS name to verify as valid for the certificate.

- **eku** (*str*) – The enhanced key usage object identifiers to verify for the certificate chain.

Returns A boolean representing whether the certificate was considered valid.

Return type bool

CLI Example:

```
salt '*' win_pki.test_cert thumbprint='AAA000' dns_name='example.test'
```

21.9.473 salt.modules.win_powercfg

This module allows you to control the power settings of a windows minion via powercfg.

New in version 2015.8.0.

```
# Set monitor to never turn off on Battery power
salt '*' powercfg.set_monitor_timeout 0 power=dc
# Set disk timeout to 120 minutes on AC power
salt '*' powercfg.set_disk_timeout 120 power=ac
```

`salt.modules.win_powercfg.`**`get_disk_timeout`**(*scheme=None*)

Get the current disk timeout of the given scheme

Parameters **scheme** (*str*) – The scheme to use, leave as `None` to use the current. Default is `None`. This can be the GUID or the Alias for the Scheme. Known Aliases are:

- `SCHEME_BALANCED` - Balanced
- `SCHEME_MAX` - Power saver
- `SCHEME_MIN` - High performance

Returns A dictionary of both the AC and DC settings

Return type dict

CLI Example:

```
salt '*' powercfg.get_disk_timeout
```

`salt.modules.win_powercfg.`**`get_hibernate_timeout`**(*scheme=None*)

Get the current hibernate timeout of the given scheme

scheme (str): The scheme to use, leave as `None` to use the current. Default is `None`. This can be the GUID or the Alias for the Scheme. Known Aliases are:

- `SCHEME_BALANCED` - Balanced
- `SCHEME_MAX` - Power saver
- `SCHEME_MIN` - High performance

Returns A dictionary of both the AC and DC settings

Return type dict

CLI Example:

```
salt '*' powercfg.get_hibernate_timeout
```

`salt.modules.win_powercfg.`**`get_monitor_timeout`**(*scheme=None*)

Get the current monitor timeout of the given scheme

Parameters **scheme** (*str*) – The scheme to use, leave as `None` to use the current. Default is `None`. This can be the GUID or the Alias for the Scheme. Known Aliases are:

- `SCHEME_BALANCED` - Balanced
- `SCHEME_MAX` - Power saver
- `SCHEME_MIN` - High performance

Returns A dictionary of both the AC and DC settings

Return type dict

CLI Example:

```
salt '*' powercfg.get_monitor_timeout
```

`salt.modules.win_powercfg.`**`get_standby_timeout`**`(`*scheme=None*`)`

Get the current standby timeout of the given scheme

scheme (str): The scheme to use, leave as `None` to use the current. Default is `None`. This can be the GUID or the Alias for the Scheme. Known Aliases are:

- `SCHEME_BALANCED` - Balanced
- `SCHEME_MAX` - Power saver
- `SCHEME_MIN` - High performance

Returns A dictionary of both the AC and DC settings

Return type dict

CLI Example:

```
salt '*' powercfg.get_standby_timeout
```

`salt.modules.win_powercfg.`**`set_disk_timeout`**`(`*timeout, power='ac', scheme=None*`)`

Set the disk timeout in minutes for the given power scheme

Parameters

- **`timeout`** (*int*) – The amount of time in minutes before the disk will timeout
- **`power`** (*str*) – Set the value for AC or DC power. Default is `ac`. Valid options are:
  - `ac` (AC Power)
  - `dc` (Battery)
- **`scheme`** (*str*) – The scheme to use, leave as `None` to use the current. Default is `None`. This can be the GUID or the Alias for the Scheme. Known Aliases are:
  - `SCHEME_BALANCED` - Balanced
  - `SCHEME_MAX` - Power saver
  - `SCHEME_MIN` - High performance

Returns `True` if successful, otherwise `False`

Return type bool

CLI Example:

```
# Sets the disk timeout to 30 minutes on battery
salt '*' powercfg.set_disk_timeout 30 power=dc
```

`salt.modules.win_powercfg.`**`set_hibernate_timeout`**`(`*timeout, power='ac', scheme=None*`)`

Set the hibernate timeout in minutes for the given power scheme

Parameters

- **`timeout`** (*int*) – The amount of time in minutes before the computer hibernates
- **`power`** (*str*) – Set the value for AC or DC power. Default is `ac`. Valid options are:
  - `ac` (AC Power)
  - `dc` (Battery)
- **`scheme`** (*str*) – The scheme to use, leave as `None` to use the current. Default is `None`. This can be the GUID or the Alias for the Scheme. Known Aliases are:
  - `SCHEME_BALANCED` - Balanced
  - `SCHEME_MAX` - Power saver
  - `SCHEME_MIN` - High performance

Returns `True` if successful, otherwise `False`

Return type bool

CLI Example:

```
# Sets the hibernate timeout to 30 minutes on Battery
salt '*' powercfg.set_hibernate_timeout 30 power=dc
```

`salt.modules.win_powercfg.`**`set_monitor_timeout`**(*timeout*, *power='ac'*, *scheme=None*)

Set the monitor timeout in minutes for the given power scheme

Parameters

- **`timeout`** (*`int`*) – The amount of time in minutes before the monitor will timeout
- **`power`** (*`str`*) – Set the value for AC or DC power. Default is `ac`. Valid options are:
  - `ac` (AC Power)
  - `dc` (Battery)
- **`scheme`** (*`str`*) – The scheme to use, leave as `None` to use the current. Default is `None`. This can be the GUID or the Alias for the Scheme. Known Aliases are:
  - `SCHEME_BALANCED` - Balanced
  - `SCHEME_MAX` - Power saver
  - `SCHEME_MIN` - High performance

Returns `True` if successful, otherwise `False`

Return type bool

CLI Example:

```
# Sets the monitor timeout to 30 minutes
salt '*' powercfg.set_monitor_timeout 30
```

`salt.modules.win_powercfg.`**`set_standby_timeout`**(*timeout*, *power='ac'*, *scheme=None*)

Set the standby timeout in minutes for the given power scheme

Parameters

- **`timeout`** (*`int`*) – The amount of time in minutes before the computer sleeps
- **`power`** (*`str`*) – Set the value for AC or DC power. Default is `ac`. Valid options are:
  - `ac` (AC Power)
  - `dc` (Battery)
- **`scheme`** (*`str`*) – The scheme to use, leave as `None` to use the current. Default is `None`. This can be the GUID or the Alias for the Scheme. Known Aliases are:
  - `SCHEME_BALANCED` - Balanced
  - `SCHEME_MAX` - Power saver
  - `SCHEME_MIN` - High performance

Returns `True` if successful, otherwise `False`

Return type bool

CLI Example:

```
# Sets the system standby timeout to 30 minutes on Battery
salt '*' powercfg.set_standby_timeout 30 power=dc
```

21.9.474 salt.modules.win_psget module

Module for managing PowerShell through PowerShellGet (PSGet)

depends

- PowerShell 5.0
- PSGet

Support for PowerShell

`salt.modules.win_psget.`**`avail_modules`**(*desc=False*)

List available modules in registered Powershell module repositories.

Parameters **desc** (`bool`) – If `True`, the verbose description will be returned.

CLI Example:

```
salt 'win01' psget.avail_modules
salt 'win01' psget.avail_modules desc=True
```

`salt.modules.win_psget.`**`bootstrap`**`()`

Make sure that nuget-anycpu.exe is installed. This will download the official nuget-anycpu.exe from the internet.

CLI Example:

```
salt 'win01' psget.bootstrap
```

`salt.modules.win_psget.`**`get_repository`**`(`*name*`)`

Get the details of a local PSGet repository

Parameters **name** (`str`) – Name of the repository

CLI Example:

```
salt 'win01' psget.get_repository MyRepo
```

`salt.modules.win_psget.`**`install`**`(`*name, minimum_version=None, required_version=None, scope=None, repository=None*`)`

Install a Powershell module from powershell gallery on the system.

Parameters

- **name** (`str`) – Name of a Powershell module
- **minimum_version** (`str`) – The maximum version to install, e.g. 1.23.2
- **required_version** (`str`) – Install a specific version
- **scope** (`str`) – The scope to install the module to, e.g. CurrentUser, Computer
- **repository** (`str`) – The friendly name of a private repository, e.g. MyREpo

CLI Example:

```
salt 'win01' psget.install PowerPlan
```

`salt.modules.win_psget.`**`list_modules`**`(`*desc=False*`)`

List currently installed PSGet Modules on the system.

Parameters **desc** (`bool`) – If `True`, the verbose description will be returned.

CLI Example:

```
salt 'win01' psget.list_modules
salt 'win01' psget.list_modules desc=True
```

`salt.modules.win_psget.`**`register_repository`**`(`*name, location, installation_policy=None*`)`

Register a PSGet repository on the local machine

Parameters

- **name** (`str`) – The name for the repository
- **location** (`str`) – The URI for the repository
- **installation_policy** (`str`) – The installation policy for packages, e.g. Trusted, Untrusted

CLI Example:

```
salt 'win01' psget.register_repository MyRepo https://myrepo.mycompany.com/
↪packages
```

`salt.modules.win_psget.`**`remove`**`(`*name*`)`

Remove a Powershell DSC module from the system.

Parameters **name** (`str`) – Name of a Powershell DSC module

CLI Example:

```
salt 'win01' psget.remove PowerPlan
```

`salt.modules.win_psget.`**`update`**`(`*name*, *maximum_version=None*, *required_version=None*`)`

Update a PowerShell module to a specific version, or the newest

Parameters

- **`name`** (`str`) – Name of a Powershell module
- **`maximum_version`** (`str`) – The maximum version to install, e.g. 1.23.2
- **`required_version`** (`str`) – Install a specific version

CLI Example:

```
salt 'win01' psget.update PowerPlan
```

21.9.475 salt.modules.winrepo

Module to manage Windows software repo on a Standalone Minion

`file_client: local` must be set in the minion config file.

For documentation on Salt's Windows Repo feature, see *here*.

`salt.modules.winrepo.`**`genrepo`**`()`

Generate winrepo_cachefile based on sls files in the winrepo_dir

CLI Example:

```
salt-call winrepo.genrepo
```

`salt.modules.winrepo.`**`show_sls`**`(`*name*, *saltenv='base'*`)`

New in version 2015.8.0.

Display the rendered software definition from a specific sls file in the local winrepo cache. This will parse all Jinja. Run pkg.refresh_db to pull the latest software definitions from the master.

Note: This function does not ask a master for an sls file to render. Instead it directly processes the file specified in *name*

Parameters

- **`str`** (*`saltenv`*) – The name/path of the package you want to view. This can be the
- **`path to a file on the minion file system or a file on the local`** (*`full`*) –
- **`cache.`** (*`minion`*) –
- **`str`** – The default environment is `base`

Returns Returns a dictionary containing the rendered data structure

Return type dict

Note: To use a file from the minion cache start from the local winrepo root (`C:\salt\var\cache\salt\minion\files\base\win\repo-ng`). If you have `.sls` files organized in subdirectories you'll have to denote them with `.`. For example, if you have a `test` directory in the winrepo root with a `gvim.sls` file inside, would target that file like so: `test.gvim`. Directories can be targeted as well as long as they contain an `init.sls` inside. For example, if you have a `node` directory with an `init.sls` inside, target that like so: `node`.

CLI Example:

```
salt '*' winrepo.show_sls gvim
salt '*' winrepo.show_sls test.npp
salt '*' winrepo.show_sls C:\test\gvim.sls
```

`salt.modules.winrepo.`**`update_git_repos`**(*clean=False*)

Checkout git repos containing *Windows Software Package Definitions*.

Important: This function requires Git for Windows to be installed in order to work. When installing, make sure to select an installation option which permits the git executable to be run from the Command Prompt.

---

clean [False] Clean repo cachedirs which are not configured under *`winrepo_remotes`*.

> **Note**: This option only applies if either pygit2 or GitPython is installed into Salt's bundled Python.

> **Warning**: This argument should not be set to `True` if a mix of git and non-git repo definitions are being used, as it will result in the non-git repo definitions being removed.

New in version 2015.8.0.

CLI Example:

```
salt-call winrepo.update_git_repos
```

21.9.476 salt.modules.win_servermanager

Manage Windows features via the ServerManager powershell module. Can list available and installed roles/features. Can install and remove roles/features.

maintainer Shane Lee <slee@saltstack.com>

platform Windows Server 2008R2 or greater

depends PowerShell module `ServerManager`

`salt.modules.win_servermanager.`**`install`**(*feature, recurse=False, restart=False, source=None, exclude=None*)

Install a feature

Note: Some features require reboot after un/installation, if so until the server is restarted other features can not be installed!

Note: Some features take a long time to complete un/installation, set -t with a long timeout

---

Parameters

- **`feature`** (*`str, list`*) – The name of the feature(s) to install. This can be a single feature, a string of features in a comma delimited list (no spaces), or a list of features.

  New in version 2018.3.0: Added the ability to pass a list of features to be installed.

- **recurse** (*Options[bool]*) – Install all sub-features. Default is False
- **restart** (*Optional[bool]*) – Restarts the computer when installation is complete, if required by the role/feature installed. Will also trigger a reboot if an item in `exclude` requires a reboot to be properly removed. Default is False
- **source** (*Optional[str]*) – Path to the source files if missing from the target system. None means that the system will use windows update services to find the required files. Default is None
- **exclude** (*Optional[str]*) – The name of the feature to exclude when installing the named feature. This can be a single feature, a string of features in a comma-delimited list (no spaces), or a list of features.

> **Warning:** As there is no exclude option for the `Add-WindowsFeature` or `Install-WindowsFeature` PowerShell commands the features named in `exclude` will be installed with other sub-features and will then be removed. **If the feature named in ``exclude`` is not a sub-feature of one of the installed items it will still be removed.**

Returns A dictionary containing the results of the install

Return type dict

CLI Example:

```
# Install the Telnet Client passing a single string
salt '*' win_servermanager.install Telnet-Client

# Install the TFTP Client and the SNMP Service passing a comma-delimited
# string. Install all sub-features
salt '*' win_servermanager.install TFTP-Client,SNMP-Service recurse=True

# Install the TFTP Client from d:\side-by-side
salt '*' win_servermanager.install TFTP-Client source=d:\\side-by-side

# Install the XPS Viewer, SNMP Service, and Remote Access passing a
# list. Install all sub-features, but exclude the Web Server
salt '*' win_servermanager.install "['XPS-Viewer', 'SNMP-Service', 'RemoteAccess']
↪" True recurse=True exclude="Web-Server"
```

`salt.modules.win_servermanager.`**`list_available`**`()`

List available features to install

Returns A list of available features as returned by the `Get-WindowsFeature` PowerShell command

Return type str

CLI Example:

```
salt '*' win_servermanager.list_available
```

`salt.modules.win_servermanager.`**`list_installed`**`()`

List installed features. Supported on Windows Server 2008 and Windows 8 and newer.

Returns A dictionary of installed features

Return type dict

CLI Example:

```
salt '*' win_servermanager.list_installed
```

`salt.modules.win_servermanager.`**`remove`**`(`*feature, remove_payload=False, restart=False*`)`

Remove an installed feature

Note: Some features require a reboot after installation/uninstallation. If one of these features are modified, then other features cannot be installed until the server is restarted. Additionally, some features take a while to complete installation/uninstallation, so it is a good idea to use the `-t` option to set a longer timeout.

Parameters

- **`feature`** (*`str, list`*) – The name of the feature(s) to remove. This can be a single feature, a string of features in a comma delimited list (no spaces), or a list of features.

  New in version 2018.3.0: Added the ability to pass a list of features to be removed.
- **`remove_payload`** (*`Optional[bool]`*) – True will cause the feature to be removed from the side-by-side store (`%SystemDrive%:\Windows\WinSxS`). Default is False
- **`restart`** (*`Optional[bool]`*) – Restarts the computer when uninstall is complete, if required by the role/feature removed. Default is False

Returns A dictionary containing the results of the uninstall

Return type dict

CLI Example:

```
salt -t 600 '*' win_servermanager.remove Telnet-Client
```

21.9.477 salt.modules.win_service

Windows Service module.

Changed in version 2016.11.0: - Rewritten to use PyWin32

`salt.modules.win_service.`**`available`**(*name*)

Check if a service is available on the system.

Parameters **`name`** (*`str`*) – The name of the service to check

Returns `True` if the service is available, `False` otherwise

Return type bool

CLI Example:

```
salt '*' service.available <service name>
```

`salt.modules.win_service.`**`create`**(*name, bin_path, exe_args=None, display_name=None, description=None, service_type='own', start_type='manual', start_delayed=False, error_control='normal', load_order_group=None, dependencies=None, account_name='.\\LocalSystem', account_password=None, run_interactive=False, **kwargs*)

Create the named service.

New in version 2015.8.0.

Parameters

- **`name`** (*`str`*) – Specifies the service name. This is not the display_name
- **`bin_path`** (*`str`*) – Specifies the path to the service binary file. Backslashes must be escaped, eg: `C:\path\to\binary.exe`
- **`exe_args`** (*`str`*) – Any additional arguments required by the service binary.
- **`display_name`** (*`str`*) – The name to be displayed in the service manager. If not passed, the `name` will be used
- **`description`** (*`str`*) – A description of the service

- **service_type** (*str*) – Specifies the service type. Default is `own`. Valid options are as follows:
  - kernel: Driver service
  - filesystem: File system driver service
  - adapter: Adapter driver service (reserved)
  - recognizer: Recognizer driver service (reserved)
  - own (default): Service runs in its own process
  - share: Service shares a process with one or more other services
- **start_type** (*str*) – Specifies the service start type. Valid options are as follows:
  - boot: Device driver that is loaded by the boot loader
  - system: Device driver that is started during kernel initialization
  - auto: Service that automatically starts
  - manual (default): Service must be started manually
  - disabled: Service cannot be started
- **start_delayed** (*bool*) – Set the service to Auto(Delayed Start). Only valid if the start_type is set to `Auto`. If service_type is not passed, but the service is already set to `Auto`, then the flag will be set. Default is `False`
- **error_control** (*str*) – The severity of the error, and action taken, if this service fails to start. Valid options are as follows:
  - normal (normal): Error is logged and a message box is displayed
  - severe: Error is logged and computer attempts a restart with the last known good configuration
  - critical: Error is logged, computer attempts to restart with the last known good configuration, system halts on failure
  - ignore: Error is logged and startup continues, no notification is given to the user
- **load_order_group** (*str*) – The name of the load order group to which this service belongs
- **dependencies** (*list*) – A list of services or load ordering groups that must start before this service
- **account_name** (*str*) – The name of the account under which the service should run. For `own` type services this should be in the `domain\username` format. The following are examples of valid built-in service accounts:
  - NT AuthorityLocalService
  - NT AuthorityNetworkService
  - NT AuthorityLocalSystem
  - .LocalSystem
- **account_password** (*str*) – The password for the account name specified in `account_name`. For the above built-in accounts, this can be None. Otherwise a password must be specified.
- **run_interactive** (*bool*) – If this setting is True, the service will be allowed to interact with the user. Not recommended for services that run with elevated privileges.

Returns A dictionary containing information about the new service

Return type dict

CLI Example:

```
salt '*' service.create <service name> <path to exe> display_name='<display name>'
```

`salt.modules.win_service.`**`create_win_salt_restart_task`**`()`

Create a task in Windows task scheduler to enable restarting the salt-minion

Returns `True` if successful, otherwise `False`

Return type bool

CLI Example:

```
salt '*' service.create_win_salt_restart_task()
```

`salt.modules.win_service.`**`delete`**`(`*name*, *timeout=90*`)`

Delete the named service

Parameters

- **`name`** (*`str`*) – The name of the service to delete
- **`timeout`** (*`int`*) – The time in seconds to wait for the service to be deleted before returning. This is necessary because a service must be stopped before it can be deleted. Default is 90 seconds

  New in version 2017.7.9,2018.3.4.

Returns

`True` if successful, otherwise `False`. Also returns `True` if the service is not present

Return type bool

CLI Example:

```
salt '*' service.delete <service name>
```

`salt.modules.win_service.`**`disable`**`(`*name*, ***kwargs*`)`

Disable the named service to start at boot

Parameters **`name`** (*`str`*) – The name of the service to disable

Returns `True` if disabled, `False` otherwise

Return type bool

CLI Example:

```
salt '*' service.disable <service name>
```

`salt.modules.win_service.`**`disabled`**`(`*name*`)`

Check to see if the named service is disabled to start on boot

Parameters **`name`** (*`str`*) – The name of the service to check

Returns True if the service is disabled

Return type bool

CLI Example:

```
salt '*' service.disabled <service name>
```

`salt.modules.win_service.`**`enable`**`(`*name*, *start_type='auto'*, *start_delayed=False*, ***kwargs*`)`

Enable the named service to start at boot

Parameters

- **`name`** (*`str`*) – The name of the service to enable.
- **`start_type`** (*`str`*) – Specifies the service start type. Valid options are as follows:
  - boot: Device driver that is loaded by the boot loader
  - system: Device driver that is started during kernel initialization
  - auto: Service that automatically starts
  - manual: Service must be started manually
  - disabled: Service cannot be started
- **`start_delayed`** (*`bool`*) – Set the service to Auto(Delayed Start). Only valid if the start_type is set to `Auto`. If service_type is not passed, but the service is already set to `Auto`, then the flag will be set.

Returns `True` if successful, `False` otherwise

Return type bool

CLI Example:

```
salt '*' service.enable <service name>
```

`salt.modules.win_service.`**`enabled`**`(`*name*, ***kwargs*`)`

Check to see if the named service is enabled to start on boot

Parameters **name** (*str*) – The name of the service to check

Returns True if the service is set to start

Return type bool

CLI Example:

```
salt '*' service.enabled <service name>
```

`salt.modules.win_service.`**`execute_salt_restart_task`**`()`

Run the Windows Salt restart task

Returns `True` if successful, otherwise `False`

Return type bool

CLI Example:

```
salt '*' service.execute_salt_restart_task()
```

`salt.modules.win_service.`**`get_all`**`()`

Return all installed services

Returns Returns a list of all services on the system.

Return type list

CLI Example:

```
salt '*' service.get_all
```

`salt.modules.win_service.`**`get_disabled`**`()`

Return a list of disabled services. Disabled is defined as a service that is marked 'Disabled' or 'Manual'.

Returns A list of disabled services.

Return type list

CLI Example:

```
salt '*' service.get_disabled
```

`salt.modules.win_service.`**`get_enabled`**`()`

Return a list of enabled services. Enabled is defined as a service that is marked to Auto Start.

Returns A list of enabled services

Return type list

CLI Example:

```
salt '*' service.get_enabled
```

`salt.modules.win_service.`**`get_service_name`**`(`**args*`)`

The Display Name is what is displayed in Windows when services.msc is executed. Each Display Name has an associated Service Name which is the actual name of the service. This function allows you to discover the Service Name by returning a dictionary of Display Names and Service Names, or filter by adding arguments of Display Names.

If no args are passed, return a dict of all services where the keys are the service Display Names and the values are the Service Names.

If arguments are passed, create a dict of Display Names and Service Names

Returns A dictionary of display names and service names

Return type dict

CLI Examples:

```
salt '*' service.get_service_name
salt '*' service.get_service_name 'Google Update Service (gupdate)' 'DHCP Client'
```

`salt.modules.win_service.`**`getsid`**(*name*)

Return the SID for this windows service

Parameters **`name`** (*`str`*) – The name of the service for which to return the SID

Returns A string representing the SID for the service

Return type str

CLI Example:

```
salt '*' service.getsid <service name>
```

`salt.modules.win_service.`**`info`**(*name*)

Get information about a service on the system

Parameters **`name`** (*`str`*) – The name of the service. This is not the display name. Use `get_service_name` to find the service name.

Returns A dictionary containing information about the service.

Return type dict

CLI Example:

```
salt '*' service.info spooler
```

`salt.modules.win_service.`**`missing`**(*name*)

The inverse of service.available.

Parameters **`name`** (*`str`*) – The name of the service to check

Returns `True` if the service is missing, `False` otherwise

Return type bool

CLI Example:

```
salt '*' service.missing <service name>
```

`salt.modules.win_service.`**`modify`**(*name, bin_path=None, exe_args=None, display_name=None, description=None, service_type=None, start_type=None, start_delayed=None, error_control=None, load_order_group=None, dependencies=None, account_name=None, account_password=None, run_interactive=None*)

Modify a service's parameters. Changes will not be made for parameters that are not passed.

New in version 2016.11.0.

Parameters

- **`name`** (*`str`*) – The name of the service. Can be found using the `service.get_service_name` function
- **`bin_path`** (*`str`*) – The path to the service executable. Backslashes must be escaped, eg: `C:\path\to\binary.exe`
- **`exe_args`** (*`str`*) – Any arguments required by the service executable
- **`display_name`** (*`str`*) – The name to display in the service manager
- **`description`** (*`str`*) – The description to display for the service
- **`service_type`** (*`str`*) – Specifies the service type. Default is `own`. Valid options are as follows:
  - kernel: Driver service
  - filesystem: File system driver service
  - adapter: Adapter driver service (reserved)
  - recognizer: Recognizer driver service (reserved)
  - own (default): Service runs in its own process

  - share: Service shares a process with one or more other services
- **start_type** (*str*) – Specifies the service start type. Valid options are as follows:
  - boot: Device driver that is loaded by the boot loader
  - system: Device driver that is started during kernel initialization
  - auto: Service that automatically starts
  - manual: Service must be started manually
  - disabled: Service cannot be started
- **start_delayed** (*bool*) – Set the service to Auto(Delayed Start). Only valid if the start_type is set to `Auto`. If service_type is not passed, but the service is already set to `Auto`, then the flag will be set.
- **error_control** (*str*) – The severity of the error, and action taken, if this service fails to start. Valid options are as follows:
  - normal: Error is logged and a message box is displayed
  - severe: Error is logged and computer attempts a restart with the last known good configuration
  - critical: Error is logged, computer attempts to restart with the last known good configuration, system halts on failure
  - ignore: Error is logged and startup continues, no notification is given to the user
- **load_order_group** (*str*) – The name of the load order group to which this service belongs
- **dependencies** (*list*) – A list of services or load ordering groups that must start before this service
- **account_name** (*str*) – The name of the account under which the service should run. For `own` type services this should be in the `domain\username` format. The following are examples of valid built-in service accounts:
  - NT AuthorityLocalService
  - NT AuthorityNetworkService
  - NT AuthorityLocalSystem
  - .LocalSystem
- **account_password** (*str*) – The password for the account name specified in `account_name`. For the above built-in accounts, this can be None. Otherwise a password must be specified.
- **run_interactive** (*bool*) – If this setting is True, the service will be allowed to interact with the user. Not recommended for services that run with elevated privileges.

Returns a dictionary of changes made

Return type dict

CLI Example:

```
salt '*' service.modify spooler start_type=disabled
```

`salt.modules.win_service.`**`restart`**(*name, timeout=90*)

Restart the named service. This issues a stop command followed by a start.

Parameters

- **name** – The name of the service to restart.

  Note: If the name passed is `salt-minion` a scheduled task is created and executed to restart the salt-minion service.

- **timeout** (*int*) – The time in seconds to wait for the service to stop and start before returning. Default is 90 seconds

> **Note:** The timeout is cumulative meaning it is applied to the stop and then to the start command. A timeout of 90 could take up to 180 seconds if the service is long in stopping and starting

New in version 2017.7.9,2018.3.4.

Returns `True` if successful, otherwise `False`

Return type bool

CLI Example:

```
salt '*' service.restart <service name>
```

`salt.modules.win_service.`**`start`**`(`*name*, *timeout=90*`)`

Start the specified service.

> **Warning:** You cannot start a disabled service in Windows. If the service is disabled, it will be changed to `Manual` start.

Parameters

- **`name`** (`str`) – The name of the service to start
- **`timeout`** (`int`) – The time in seconds to wait for the service to start before returning. Default is 90 seconds

  New in version 2017.7.9,2018.3.4.

Returns

`True` if successful, otherwise `False`. Also returns `True` if the service is already started

Return type bool

CLI Example:

```
salt '*' service.start <service name>
```

`salt.modules.win_service.`**`status`**`(`*name*, **args*, ***kwargs*`)`

Return the status for a service. If the name contains globbing, a dict mapping service name to True/False values is returned.

Changed in version 2018.3.0: The service name can now be a glob (e.g. `salt*`)

Parameters **`name`** (`str`) – The name of the service to check

Returns True if running, False otherwise dict: Maps service name to True if running, False otherwise

Return type bool

CLI Example:

```
salt '*' service.status <service name>
```

`salt.modules.win_service.`**`stop`**`(`*name*, *timeout=90*`)`

Stop the specified service

Parameters

- **`name`** (`str`) – The name of the service to stop
- **`timeout`** (`int`) – The time in seconds to wait for the service to stop before returning. Default is 90 seconds

  New in version 2017.7.9,2018.3.4.

Returns

`True` if successful, otherwise `False`. Also returns `True` if the service is already stopped

Return type bool

CLI Example:

```
salt '*' service.stop <service name>
```

21.9.478 salt.modules.win_shadow

Manage the shadow file

Important: If you feel that Salt should be using this module to manage passwords on a minion, and it is using a different module (or gives an error similar to *'shadow.info' is not available*), see *here*.

---

salt.modules.win_shadow.**info**(*name*)

Return information for the specified user This is just returns dummy data so that salt states can work.

Parameters **name** (*str*) – The name of the user account to show.

CLI Example:

```
salt '*' shadow.info root
```

salt.modules.win_shadow.**require_password_change**(*name*)

Require the user to change their password the next time they log in.

Parameters **name** – The name of the user account to require a password change.

Returns True if successful. False if unsuccessful.

Return type bool

CLI Example:

```
salt '*' shadow.require_password_change <username>
```

salt.modules.win_shadow.**set_expire**(*name*, *expire*)

Set the expiration date for a user account.

Parameters

- **name** – The name of the user account to edit.
- **expire** – The date the account will expire.

Returns True if successful. False if unsuccessful.

Return type bool

CLI Example:

```
salt '*' shadow.set_expire <username> 2016/7/1
```

salt.modules.win_shadow.**set_password**(*name*, *password*)

Set the password for a named user.

Parameters

- **name** (*str*) – The name of the user account
- **password** (*str*) – The new password

Returns True if successful. False if unsuccessful.

Return type bool

CLI Example:

```
salt '*' shadow.set_password root mysecretpassword
```

salt.modules.win_shadow.**unlock_account**(*name*)

Unlocks a user account.

Parameters **name** – The name of the user account to unlock.

Returns True if successful. False if unsuccessful.

Return type bool

CLI Example:

```
salt '*' shadow.unlock_account <username>
```

21.9.479 salt.modules.win_smtp_server module

Module for managing IIS SMTP server configuration on Windows servers. The Windows features 'SMTP-Server' and 'Web-WMI' must be installed.

depends wmi

salt.modules.win_smtp_server.**get_connection_ip_list**(*as_wmi_format=False, server='SmtpSvc/1'*)

Get the IPGrant list for the SMTP virtual server.

Parameters

- **as_wmi_format** (*bool*) – Returns the connection IPs as a list in the format WMI expects.
- **server** (*str*) – The SMTP server name.

Returns A dictionary of the IP and subnet pairs.

Return type dict

CLI Example:

```
salt '*' win_smtp_server.get_connection_ip_list
```

salt.modules.win_smtp_server.**get_log_format**(*server='SmtpSvc/1'*)

Get the active log format for the SMTP virtual server.

Parameters **server** (*str*) – The SMTP server name.

Returns A string of the log format name.

Return type str

CLI Example:

```
salt '*' win_smtp_server.get_log_format
```

salt.modules.win_smtp_server.**get_log_format_types**()

Get all available log format names and ids.

Returns A dictionary of the log format names and ids.

Return type dict

CLI Example:

```
salt '*' win_smtp_server.get_log_format_types
```

salt.modules.win_smtp_server.**get_relay_ip_list**(*server='SmtpSvc/1'*)

Get the RelayIpList list for the SMTP virtual server.

Parameters **server** (*str*) – The SMTP server name.

Returns A list of the relay IPs.

Return type list

Note: A return value of None corresponds to the restrictive 'Only the list below' GUI parameter with an empty access list, and setting an empty list/tuple corresponds to the more permissive 'All except the list below' GUI parameter.

CLI Example:

```
salt '*' win_smtp_server.get_relay_ip_list
```

`salt.modules.win_smtp_server.`**`get_server_setting`**(*settings*, *server='SmtpSvc/1'*)

Get the value of the setting for the SMTP virtual server.

Parameters

- **settings** (*str*) – A list of the setting names.
- **server** (*str*) – The SMTP server name.

Returns A dictionary of the provided settings and their values.

Return type dict

CLI Example:

```
salt '*' win_smtp_server.get_server_setting settings="['MaxRecipients']"
```

`salt.modules.win_smtp_server.`**`get_servers`**()

Get the SMTP virtual server names.

Returns A list of the SMTP virtual servers.

Return type list

CLI Example:

```
salt '*' win_smtp_server.get_servers
```

`salt.modules.win_smtp_server.`**`set_connection_ip_list`**(*addresses=None*, *grant_by_default=False*, *server='SmtpSvc/1'*)

Set the IPGrant list for the SMTP virtual server.

Parameters

- **addresses** (*str*) – A dictionary of IP + subnet pairs.
- **grant_by_default** (*bool*) – Whether the addresses should be a blacklist or whitelist.
- **server** (*str*) – The SMTP server name.

Returns A boolean representing whether the change succeeded.

Return type bool

CLI Example:

```
salt '*' win_smtp_server.set_connection_ip_list addresses="{'127.0.0.1': '255.255.
↪255.255'}"
```

`salt.modules.win_smtp_server.`**`set_log_format`**(*log_format*, *server='SmtpSvc/1'*)

Set the active log format for the SMTP virtual server.

Parameters

- **log_format** (*str*) – The log format name.
- **server** (*str*) – The SMTP server name.

Returns A boolean representing whether the change succeeded.

Return type bool

CLI Example:

```
salt '*' win_smtp_server.set_log_format 'Microsoft IIS Log File Format'
```

`salt.modules.win_smtp_server.`**`set_relay_ip_list`**(*addresses=None*, *server='SmtpSvc/1'*)

Set the RelayIpList list for the SMTP virtual server.

Due to the unusual way that Windows stores the relay IPs, it is advisable to retrieve the existing list you wish to set from a pre-configured server.

For example, setting '127.0.0.1' as an allowed relay IP through the GUI would generate an actual relay IP list similar to the following:

```
['24.0.0.128', '32.0.0.128', '60.0.0.128', '68.0.0.128', '1.0.0.0', '76.0.0.0',
 '0.0.0.0', '0.0.0.0', '1.0.0.0', '1.0.0.0', '2.0.0.0', '2.0.0.0', '4.0.0.0',
 '0.0.0.0', '76.0.0.128', '0.0.0.0', '0.0.0.0', '0.0.0.0', '0.0.0.0',
 '255.255.255.255', '127.0.0.1']
```

Note: Setting the list to None corresponds to the restrictive 'Only the list below' GUI parameter with an empty access list configured, and setting an empty list/tuple corresponds to the more permissive 'All except the list below' GUI parameter.

Parameters
- **addresses** (*str*) – A list of the relay IPs. The order of the list is important.
- **server** (*str*) – The SMTP server name.

Returns A boolean representing whether the change succeeded.

Return type bool

CLI Example:

```
salt '*' win_smtp_server.set_relay_ip_list addresses="['192.168.1.1', '172.16.1.1
↪']"
```

salt.modules.win_smtp_server.**set_server_setting**(*settings, server='SmtpSvc/1'*)

Set the value of the setting for the SMTP virtual server.

Note: The setting names are case-sensitive.

Parameters
- **settings** (*str*) – A dictionary of the setting names and their values.
- **server** (*str*) – The SMTP server name.

Returns A boolean representing whether all changes succeeded.

Return type bool

CLI Example:

```
salt '*' win_smtp_server.set_server_setting settings="{'MaxRecipients': '500'}"
```

21.9.480 salt.modules.win_snmp module

Module for managing SNMP service settings on Windows servers. The Windows feature 'SNMP-Service' must be installed.

salt.modules.win_snmp.**get_agent_service_types**()

Get the sysServices types that can be configured.

Returns A list of service types.

Return type list

CLI Example:

```
salt '*' win_snmp.get_agent_service_types
```

salt.modules.win_snmp.**get_agent_settings**()

Determine the value of the SNMP sysContact, sysLocation, and sysServices settings.

Returns A dictionary of the agent settings.

Return type dict

CLI Example:

```
salt '*' win_snmp.get_agent_settings
```

salt.modules.win_snmp.**get_auth_traps_enabled**()

Determine whether the host is configured to send authentication traps.

Returns True if traps are enabled, otherwise False

Return type bool

CLI Example:

```
salt '*' win_snmp.get_auth_traps_enabled
```

salt.modules.win_snmp.**get_community_names**()

Get the current accepted SNMP community names and their permissions.

If community names are being managed by Group Policy, those values will be returned instead like this:

```
TestCommunity:
    Managed by GPO
```

Community names managed normally will denote the permission instead:

```
TestCommunity:
    Read Only
```

Returns A dictionary of community names and permissions.

Return type dict

CLI Example:

```
salt '*' win_snmp.get_community_names
```

salt.modules.win_snmp.**get_permission_types**()

Get the permission types that can be configured for communities.

Returns A list of permission types.

Return type list

CLI Example:

```
salt '*' win_snmp.get_permission_types
```

salt.modules.win_snmp.**set_agent_settings**(*contact=None*, *location=None*, *services=None*)

Manage the SNMP sysContact, sysLocation, and sysServices settings.

Parameters

- **contact** (*str*, *optional*) – The SNMP contact.
- **location** (*str*, *optional*) – The SNMP location.
- **services** (*list*, *optional*) – A list of selected services. The possible service names can be found via `win_snmp.get_agent_service_types`. To disable all services pass a list of None, ie: ['None']

Returns True if successful, otherwise False

Return type bool

CLI Example:

```
salt '*' win_snmp.set_agent_settings contact='Contact Name' location='Place'
↪services="['Physical']"
```

salt.modules.win_snmp.**set_auth_traps_enabled**(*status=True*)

Manage the sending of authentication traps.

Parameters **`status`** (*bool*) – True to enable traps. False to disable.
Returns True if successful, otherwise False
Return type bool

CLI Example:

```
salt '*' win_snmp.set_auth_traps_enabled status='True'
```

`salt.modules.win_snmp.`**`set_community_names`**(*communities*)

Manage the SNMP accepted community names and their permissions.

Note: Settings managed by Group Policy will always take precedence over those set using the SNMP interface. Therefore if this function finds Group Policy settings it will raise a CommandExecutionError

Parameters **`communities`** (*dict*) – A dictionary of SNMP community names and permissions. The possible permissions can be found via `win_snmp.get_permission_types`.
Returns True if successful, otherwise False
Return type bool
Raises ***CommandExecutionError*** – If SNMP settings are being managed by Group Policy

CLI Example:

```
salt '*' win_snmp.set_community_names communities="{'TestCommunity': 'Read Only'}'
```

21.9.481 salt.modules.win_status

Module for returning various status data about a minion. These data can be useful for compiling into stats later, or for problem solving if your minion is having problems.

New in version 0.12.0.

depends

- wmi

class `salt.modules.win_status.`**`SYSTEM_PERFORMANCE_INFORMATION`**

`AvailablePagedPoolPages`
Structure/Union member

`AvailablePages`
Structure/Union member

`CacheIoCount`
Structure/Union member

`CacheReadCount`
Structure/Union member

`CacheTransitionCount`
Structure/Union member

`CcCopyReadNoWait`
Structure/Union member

`CcCopyReadNoWaitMiss`
Structure/Union member

CcCopyReadWait
Structure/Union member

CcCopyReadWaitMiss
Structure/Union member

CcDataFlushes
Structure/Union member

CcDataPages
Structure/Union member

CcDirtyPagesThreshold
Structure/Union member

CcFastMdlReadNoWait
Structure/Union member

CcFastMdlReadNotPossible
Structure/Union member

CcFastMdlReadResourceMiss
Structure/Union member

CcFastMdlReadWait
Structure/Union member

CcFastReadNoWait
Structure/Union member

CcFastReadNotPossible
Structure/Union member

CcFastReadResourceMiss
Structure/Union member

CcFastReadWait
Structure/Union member

CcLazyWriteIos
Structure/Union member

CcLazyWritePages
Structure/Union member

CcMapDataNoWait
Structure/Union member

CcMapDataNoWaitMiss
Structure/Union member

CcMapDataWait
Structure/Union member

CcMapDataWaitMiss
Structure/Union member

CcMdlReadNoWait
Structure/Union member

CcMdlReadNoWaitMiss
Structure/Union member

CcMdlReadWait
Structure/Union member

CcMdlReadWaitMiss
Structure/Union member

CcPinMappedDataCount
Structure/Union member

CcPinReadNoWait
Structure/Union member

CcPinReadNoWaitMiss
Structure/Union member

CcPinReadWait
Structure/Union member

CcPinReadWaitMiss
Structure/Union member

CcReadAheadIos
Structure/Union member

CcTotalDirtyPages
Structure/Union member

CommitLimit
Structure/Union member

CommittedPages
Structure/Union member

ContextSwitches
Structure/Union member

CopyOnWriteCount
Structure/Union member

DemandZeroCount
Structure/Union member

DirtyPagesWriteCount
Structure/Union member

DirtyWriteIoCount
Structure/Union member

FirstLevelTbFills
Structure/Union member

FreeSystemPtes
Structure/Union member

IdleProcessTime
Structure/Union member

IoOtherOperationCount
Structure/Union member

IoOtherTransferCount
Structure/Union member

IoReadOperationCount
Structure/Union member

IoReadTransferCount
Structure/Union member

IoWriteOperationCount
Structure/Union member

IoWriteTransferCount
Structure/Union member

MappedPagesWriteCount
Structure/Union member

MappedWriteIoCount
Structure/Union member

NonPagedPoolAllocs
Structure/Union member

NonPagedPoolFrees
Structure/Union member

NonPagedPoolLookasideHits
Structure/Union member

NonPagedPoolPages
Structure/Union member

PageFaultCount
Structure/Union member

PageReadCount
Structure/Union member

PageReadIoCount
Structure/Union member

PagedPoolAllocs
Structure/Union member

PagedPoolFrees
Structure/Union member

PagedPoolLookasideHits
Structure/Union member

PagedPoolPages
Structure/Union member

PeakCommitment
Structure/Union member

ResidentAvailablePages
Structure/Union member

ResidentPagedPoolPage
Structure/Union member

ResidentSystemCachePage
Structure/Union member

`ResidentSystemCodePage`
Structure/Union member

`ResidentSystemDriverPage`
Structure/Union member

`SecondLevelTbFills`
Structure/Union member

`SharedCommittedPages`
Structure/Union member

`SystemCalls`
Structure/Union member

`TotalSystemCodePages`
Structure/Union member

`TotalSystemDriverPages`
Structure/Union member

`TransitionCount`
Structure/Union member

`salt.modules.win_status.`**`cpuload`**`()`

New in version 2015.8.0.

Return the processor load as a percentage

CLI Example:

```
salt '*' status.cpuload
```

`salt.modules.win_status.`**`cpustats`**`()`

Return information about the CPU.
Returns dict: A dictionary containing information about the CPU stats
CLI Example:

```
salt * status.cpustats
```

`salt.modules.win_status.`**`diskusage`**`(`*human_readable=False, path=None*`)`

New in version 2015.8.0.

Return the disk usage for this minion
human_readable [False] If `True`, usage will be in KB/MB/GB etc.
CLI Example:

```
salt '*' status.diskusage path=c:/salt
```

`salt.modules.win_status.`**`loadavg`**`()`

Returns counter information related to the load of the machine
Returns A dictionary of counters
Return type dict
CLI Example:

```
salt * status.loadavg
```

`salt.modules.win_status.`**`master`**`(`*master=None, connected=True*`)`

New in version 2015.5.0.

Fire an event if the minion gets disconnected from its master. This function is meant to be run via a scheduled job from the minion. If master_ip is an FQDN/Hostname, is must be resolvable to a valid IPv4 address.

CLI Example:

```
salt '*' status.master
```

`salt.modules.win_status.`**`meminfo`**`()`

Return information about physical and virtual memory on the system

Returns A dictionary of information about memory on the system

Return type dict

CLI Example:

```
salt * status.meminfo
```

`salt.modules.win_status.`**`procs`**`(`*count=False*`)`

Return the process data

count [False] If `True`, this function will simply return the number of processes.

New in version 2015.8.0.

CLI Example:

```
salt '*' status.procs
salt '*' status.procs count
```

`salt.modules.win_status.`**`saltmem`**`(`*human_readable=False*`)`

New in version 2015.8.0.

Returns the amount of memory that salt is using

human_readable [False] return the value in a nicely formatted number

CLI Example:

```
salt '*' status.saltmem
salt '*' status.saltmem human_readable=True
```

`salt.modules.win_status.`**`uptime`**`(`*human_readable=False*`)`

New in version 2015.8.0.

Return the system uptime for the machine

Parameters **`human_readable`** (*`bool`*) – Return uptime in human readable format if `True`, otherwise return seconds. Default is `False`

Note: Human readable format is `days, hours:min:sec`. Days will only be displayed if more than 0

Returns The uptime in seconds or human readable format depending on the value of `human_readable`

Return type str

CLI Example:

```
salt '*' status.uptime
salt '*' status.uptime human_readable=True
```

`salt.modules.win_status.`**`vmstats`**`()`

Return information about the virtual memory on the machine

Returns A dictionary of virtual memory stats

Return type dict

CLI Example:

```
salt * status.vmstats
```

21.9.482 salt.modules.win_system

Module for managing windows systems.

depends

- pywintypes
- win32api
- win32con
- win32net
- wmi

Support for reboot, shutdown, etc

`salt.modules.win_system.`**`get_computer_desc`**`()`

Get the Windows computer description

Returns Returns the computer description if found. Otherwise returns `False`.

Return type str

CLI Example:

```
salt 'minion-id' system.get_computer_desc
```

`salt.modules.win_system.`**`get_computer_name`**`()`

Get the Windows computer name

Returns Returns the computer name if found. Otherwise returns `False`.

Return type str

CLI Example:

```
salt 'minion-id' system.get_computer_name
```

`salt.modules.win_system.`**`get_domain_workgroup`**`()`

Get the domain or workgroup the computer belongs to.

New in version 2015.5.7.

New in version 2015.8.2.

Returns The name of the domain or workgroup

Return type str

CLI Example:

```
salt 'minion-id' system.get_domain_workgroup
```

`salt.modules.win_system.`**`get_hostname`**`()`

Get the hostname of the windows minion

New in version 2016.3.0.

Returns Returns the hostname of the windows minion

Return type str

CLI Example:

```
salt 'minion-id' system.get_hostname
```

`salt.modules.win_system.`**`get_pending_component_servicing`**`()`

Determine whether there are pending Component Based Servicing tasks that require a reboot.

New in version 2016.11.0.

Returns `True` if there are pending Component Based Servicing tasks, otherwise `False`

Return type bool

CLI Example:

```
salt '*' system.get_pending_component_servicing
```

`salt.modules.win_system.`**`get_pending_computer_name`**`()`

Get a pending computer name. If the computer name has been changed, and the change is pending a system reboot, this function will return the pending computer name. Otherwise, `None` will be returned. If there was an error retrieving the pending computer name, `False` will be returned, and an error message will be logged to the minion log.

Returns Returns the pending name if pending restart. Returns `None` if not pending restart.

Return type str

CLI Example:

```
salt 'minion-id' system.get_pending_computer_name
```

`salt.modules.win_system.`**`get_pending_domain_join`**`()`

Determine whether there is a pending domain join action that requires a reboot.

New in version 2016.11.0.

Returns `True` if there is a pending domain join action, otherwise `False`

Return type bool

CLI Example:

```
salt '*' system.get_pending_domain_join
```

`salt.modules.win_system.`**`get_pending_file_rename`**`()`

Determine whether there are pending file rename operations that require a reboot.

New in version 2016.11.0.

Returns `True` if there are pending file rename operations, otherwise `False`

Return type bool

CLI Example:

```
salt '*' system.get_pending_file_rename
```

`salt.modules.win_system.`**`get_pending_reboot`**`()`

Determine whether there is a reboot pending.

New in version 2016.11.0.

Returns `True` if the system is pending reboot, otherwise `False`

Return type bool

CLI Example:

```
salt '*' system.get_pending_reboot
```

`salt.modules.win_system.`**`get_pending_servermanager`**`()`

Determine whether there are pending Server Manager tasks that require a reboot.

New in version 2016.11.0.

Returns `True` if there are pending Server Manager tasks, otherwise `False`

Return type bool

CLI Example:

```
salt '*' system.get_pending_servermanager
```

`salt.modules.win_system.`**`get_pending_update`**`()`

Determine whether there are pending updates that require a reboot.

New in version 2016.11.0.

Returns `True` if there are pending updates, otherwise `False`

Return type bool

CLI Example:

```
salt '*' system.get_pending_update
```

`salt.modules.win_system.`**`get_reboot_required_witnessed`**`()`

Determine if at any time during the current boot session the salt minion witnessed an event indicating that a reboot is required.

This function will return `True` if an install completed with exit code 3010 during the current boot session and can be extended where appropriate in the future.

New in version 2016.11.0.

Returns `True` if the `Requires reboot` registry flag is set to `1`, otherwise `False`

Return type bool

CLI Example:

```
salt '*' system.get_reboot_required_witnessed
```

`salt.modules.win_system.`**`get_system_date`**`()`

Get the Windows system date

Returns Returns the system date

Return type str

CLI Example:

```
salt '*' system.get_system_date
```

`salt.modules.win_system.`**`get_system_info`**`()`

Get system information.

Note: Not all system info is available across all versions of Windows. If it is not available on an older version, it will be skipped

---

Returns Dictionary containing information about the system to include name, description, version, etc…

Return type dict

CLI Example:

```
salt 'minion-id' system.get_system_info
```

`salt.modules.win_system.`**`get_system_time`**`()`

Get the system time.

Returns Returns the system time in HH:MM:SS AM/PM format.

Return type str

CLI Example:

```
salt 'minion-id' system.get_system_time
```

`salt.modules.win_system.`**`halt`**(*timeout=5*, *in_seconds=False*)

Halt a running system.

Parameters

- **`timeout`** (*`int`*) – Number of seconds before halting the system. Default is 5 seconds.
- **`in_seconds`** (*`bool`*) – Whether to treat timeout as seconds or minutes.

  New in version 2015.8.0.

Returns `True` if successful, otherwise `False`

Return type bool

CLI Example:

```
salt '*' system.halt 5 True
```

`salt.modules.win_system.`**`init`**(*runlevel*)

Change the system runlevel on sysV compatible systems. Not applicable to Windows

CLI Example:

```
salt '*' system.init 3
```

`salt.modules.win_system.`**`join_domain`**(*domain*, *username=None*, *password=None*, *account_ou=None*, *account_exists=False*, *restart=False*)

Join a computer to an Active Directory domain. Requires a reboot.

Parameters

- **`domain`** (*`str`*) – The domain to which the computer should be joined, e.g. `example.com`
- **`username`** (*`str`*) – Username of an account which is authorized to join computers to the specified domain. Needs to be either fully qualified like `user@domain.tld` or simply `user`
- **`password`** (*`str`*) – Password of the specified user
- **`account_ou`** (*`str`*) – The DN of the OU below which the account for this computer should be created when joining the domain, e.g. `ou=computers,ou=departm_432,dc=my-company,dc=com`
- **`account_exists`** (*`bool`*) – If set to `True` the computer will only join the domain if the account already exists. If set to `False` the computer account will be created if it does not exist, otherwise it will use the existing account. Default is `False`
- **`restart`** (*`bool`*) – `True` will restart the computer after a successful join. Default is `False`

  New in version 2015.8.2/2015.5.7.

Returns Returns a dictionary if successful, otherwise `False`

Return type dict

CLI Example:

```
salt 'minion-id' system.join_domain domain='domain.tld' \
                 username='joinuser' password='joinpassword' \
                 account_ou='ou=clients,ou=org,dc=domain,dc=tld' \
                 account_exists=False, restart=True
```

`salt.modules.win_system.`**`lock`**()

Lock the workstation.

Returns `True` if successful, otherwise `False`

Return type bool

CLI Example:

```
salt 'minion-id' system.lock
```

`salt.modules.win_system.`**`poweroff`**(*timeout=5*, *in_seconds=False*)

Power off a running system.

Parameters

- **`timeout`** (*`int`*) – Number of seconds before powering off the system. Default is 5 seconds.
- **`in_seconds`** (*`bool`*) – Whether to treat timeout as seconds or minutes.

  New in version 2015.8.0.

Returns `True` if successful, otherwise `False`

Return type bool

CLI Example:

```
salt '*' system.poweroff 5
```

`salt.modules.win_system.`**`reboot`**(*timeout=5*, *in_seconds=False*, *wait_for_reboot=False*, *only_on_pending_reboot=False*)

Reboot a running system.

Parameters

- **`timeout`** (*`int`*) – The number of minutes/seconds before rebooting the system. Use of minutes or seconds depends on the value of `in_seconds`. Default is 5 minutes.
- **`in_seconds`** (*`bool`*) –

  `True` will cause the `timeout` parameter to be in seconds. `False` will be in minutes. Default is `False`.

  New in version 2015.8.0.
- **`wait_for_reboot`** (*`bool`*) – `True` will sleep for timeout + 30 seconds after reboot has been initiated. This is useful for use in a highstate. For example, you may have states that you want to apply only after the reboot. Default is `False`.

  New in version 2015.8.0.
- **`only_on_pending_reboot`** (*`bool`*) – If this is set to `True`, then the reboot will only proceed if the system reports a pending reboot. Setting this parameter to `True` could be useful when calling this function from a final housekeeping state intended to be executed at the end of a state run (using *order: last*). Default is `False`.

Returns `True` if successful (a reboot will occur), otherwise `False`

Return type bool

CLI Example:

```
salt '*' system.reboot 5
salt '*' system.reboot 5 True
```

Invoking this function from a final housekeeping state:

```
final_housekeeping:
   module.run:
      - name: system.reboot
      - only_on_pending_reboot: True
      - order: last
```

`salt.modules.win_system.`**`set_computer_desc`**(*desc=None*)

Set the Windows computer description

Parameters **`desc`** (*`str`*) – The computer description

Returns Description if successful, otherwise `False`

Return type str

CLI Example:

```
salt 'minion-id' system.set_computer_desc 'This computer belongs to Dave!'
```

`salt.modules.win_system.`**`set_computer_name`**`(`*name*`)`

Set the Windows computer name

Parameters **`name`** (*`str`*) – The new name to give the computer. Requires a reboot to take effect.

Returns Returns a dictionary containing the old and new names if successful. `False` if not.

Return type dict

CLI Example:

```
salt 'minion-id' system.set_computer_name 'DavesComputer'
```

`salt.modules.win_system.`**`set_hostname`**`(`*hostname*`)`

Set the hostname of the windows minion, requires a restart before this will be updated.

New in version 2016.3.0.

Parameters **`hostname`** (*`str`*) – The hostname to set

Returns `True` if successful, otherwise `False`

Return type bool

CLI Example:

```
salt 'minion-id' system.set_hostname newhostname
```

`salt.modules.win_system.`**`set_reboot_required_witnessed`**`()`

This function is used to remember that an event indicating that a reboot is required was witnessed. This function relies on the salt-minion's ability to create the following volatile registry key in the *HKLM* hive:

SYSTEM\CurrentControlSet\Services\salt-minion\Volatile-Data

Because this registry key is volatile, it will not persist beyond the current boot session. Also, in the scope of this key, the name *'Reboot required'* will be assigned the value of *1*.

For the time being, this function is being used whenever an install completes with exit code 3010 and can be extended where appropriate in the future.

New in version 2016.11.0.

Returns `True` if successful, otherwise `False`

Return type bool

CLI Example:

```
salt '*' system.set_reboot_required_witnessed
```

`salt.modules.win_system.`**`set_system_date`**`(`*newdate*`)`

Set the Windows system date. Use <mm-dd-yy> format for the date.

Parameters **`newdate`** (*`str`*) – The date to set. Can be any of the following formats

- YYYY-MM-DD
- MM-DD-YYYY
- MM-DD-YY
- MM/DD/YYYY
- MM/DD/YY
- YYYY/MM/DD

Returns `True` if successful, otherwise `False`

Return type bool

CLI Example:

```
salt '*' system.set_system_date '03-28-13'
```

`salt.modules.win_system.`**`set_system_date_time`**`(`*years=None, months=None, days=None, hours=None, minutes=None, seconds=None*`)`

Set the system date and time. Each argument is an element of the date, but not required. If an element is not passed, the current system value for that element will be used. For example, if you don't pass the year, the current system year will be used. (Used by set_system_date and set_system_time)

Parameters

- **years** (*int*) – Years digit, ie: 2015
- **months** (*int*) – Months digit: 1 - 12
- **days** (*int*) – Days digit: 1 - 31
- **hours** (*int*) – Hours digit: 0 - 23
- **minutes** (*int*) – Minutes digit: 0 - 59
- **seconds** (*int*) – Seconds digit: 0 - 59

Returns `True` if successful, otherwise `False`

Return type bool

CLI Example:

```
salt '*' system.set_system_date_ time 2015 5 12 11 37 53
```

`salt.modules.win_system.`**`set_system_time`**`(`*newtime*`)`

Set the system time.

Parameters **newtime** (*str*) – The time to set. Can be any of the following formats:

- HH:MM:SS AM/PM
- HH:MM AM/PM
- HH:MM:SS (24 hour)
- HH:MM (24 hour)

Returns `True` if successful, otherwise `False`

Return type bool

CLI Example:

```
salt 'minion-id' system.set_system_time 12:01
```

`salt.modules.win_system.`**`shutdown`**`(`*message=None, timeout=5, force_close=True, reboot=False, in_seconds=False, only_on_pending_reboot=False*`)`

Shutdown a running system.

Parameters

- **message** (*str*) – The message to display to the user before shutting down.
- **timeout** (*int*) – The length of time (in seconds) that the shutdown dialog box should be displayed. While this dialog box is displayed, the shutdown can be aborted using the `system.shutdown_abort` function.

  If timeout is not zero, InitiateSystemShutdown displays a dialog box on the specified computer. The dialog box displays the name of the user who called the function, the message specified by the lpMessage parameter, and prompts the user to log off. The dialog box beeps when it is created and remains on top of other windows (system modal). The dialog box can be moved but not closed. A timer counts down the remaining time before the shutdown occurs.

  If timeout is zero, the computer shuts down immediately without displaying the dialog box and cannot be stopped by `system.shutdown_abort`.

  Default is 5 minutes
- **in_seconds** (*bool*) –

  `True` will cause the `timeout` parameter to be in seconds. `False` will be in minutes. Default is `False`.

  New in version 2015.8.0.
- **force_close** (*bool*) – `True` will force close all open applications. `False` will

display a dialog box instructing the user to close open applications. Default is `True`.

- **`reboot`** (*`bool`*) – `True` restarts the computer immediately after shutdown. `False` powers down the system. Default is `False`.
- **`only_on_pending_reboot`** (*`bool`*) – If this is set to True, then the shutdown will only proceed if the system reports a pending reboot. To optionally shutdown in a highstate, consider using the shutdown state instead of this module.
- **`only_on_pending_reboot`** – If `True` the shutdown will only proceed if there is a reboot pending. `False` will shutdown the system. Default is `False`.

Returns `True` if successful (a shutdown or reboot will occur), otherwise `False`

Return type bool

CLI Example:

```
salt '*' system.shutdown "System will shutdown in 5 minutes"
```

`salt.modules.win_system.`**`shutdown_abort`**`()`

Abort a shutdown. Only available while the dialog box is being displayed to the user. Once the shutdown has initiated, it cannot be aborted.

Returns `True` if successful, otherwise `False`

Return type bool

CLI Example:

```
salt 'minion-id' system.shutdown_abort
```

`salt.modules.win_system.`**`shutdown_hard`**`()`

Shutdown a running system with no timeout or warning.

Returns `True` if successful, otherwise `False`

Return type bool

CLI Example:

```
salt '*' system.shutdown_hard
```

`salt.modules.win_system.`**`start_time_service`**`()`

Start the Windows time service

Returns `True` if successful, otherwise `False`

Return type bool

CLI Example:

```
salt '*' system.start_time_service
```

`salt.modules.win_system.`**`stop_time_service`**`()`

Stop the Windows time service

Returns `True` if successful, otherwise `False`

Return type bool

CLI Example:

```
salt '*' system.stop_time_service
```

`salt.modules.win_system.`**`unjoin_domain`**`(`*username=None, password=None, domain=None, workgroup='WORKGROUP', disable=False, restart=False*`)`

Unjoin a computer from an Active Directory Domain. Requires a restart.

Parameters

- **`username`** (*`str`*) – Username of an account which is authorized to manage computer accounts on the domain. Needs to be a fully qualified name like `user@domain.tld` or `domain.tld\user`. If the domain is not specified, the passed domain will be used. If the computer account doesn't need to be disabled after the computer is unjoined, this can be `None`.

- **password** (*str*) – The password of the specified user
- **domain** (*str*) – The domain from which to unjoin the computer. Can be `None`
- **workgroup** (*str*) – The workgroup to join the computer to. Default is `WORKGROUP`

  New in version 2015.8.2/2015.5.7.
- **disable** (*bool*) – `True` to disable the computer account in Active Directory. Default is `False`
- **restart** (*bool*) – `True` will restart the computer after successful unjoin. Default is `False`

  New in version 2015.8.2/2015.5.7.

Returns Returns a dictionary if successful, otherwise `False`

Return type dict

CLI Example:

```
salt 'minion-id' system.unjoin_domain restart=True

salt 'minion-id' system.unjoin_domain username='unjoinuser' \
                 password='unjoinpassword' disable=True \
                 restart=True
```

21.9.483 salt.modules.win_task module

Windows Task Scheduler Module .. versionadded:: 2016.3.0

A module for working with the Windows Task Scheduler. You can add and edit existing tasks. You can add and clear triggers and actions. You can list all tasks, folders, triggers, and actions.

`salt.modules.win_task.`**`add_action`**(*name=None*, *location='\\'*, *action_type='Execute'*, ***kwargs*)

Add an action to a task.

Parameters

- **name** (*str*) – The name of the task to which to add the action.
- **location** (*str*) – A string value representing the location of the task. Default is `\` which is the root for the task scheduler (`C:\Windows\System32\tasks`).
- **action_type** (*str*) – The type of action to add. There are three action types. Each one requires its own set of Keyword Arguments (kwargs). Valid values are:
  - Execute
  - Email
  - Message

Required arguments for each action_type:

Execute

Execute a command or an executable

cmd (str): (required) The command or executable to run.

arguments (str): (optional) Arguments to be passed to the command or executable. To launch a script the first command will need to be the interpreter for the script. For example, to run a vbscript you would pass `cscript.exe` in the `cmd` parameter and pass the script in the `arguments` parameter as follows:

- `cmd='cscript.exe' arguments='c:\scripts\myscript.vbs'`

Batch files do not need an interpreter and may be passed to the cmd parameter directly.

start_in (str): (optional) The current working directory for the command.

Email

Send and email. Requires `server`, `from`, and `to` or `cc`.

from (str): The sender

reply_to (str): Who to reply to

to (str): The recipient

cc (str): The CC recipient

bcc (str): The BCC recipient

subject (str): The subject of the email

body (str): The Message Body of the email

server (str): The server used to send the email

attachments (list): A list of attachments. These will be the paths to the files to attach. ie: `attachments="['C:\attachment1.txt', 'C:\attachment2.txt']"`

Message

Display a dialog box. The task must be set to "Run only when user is logged on" in order for the dialog box to display. Both parameters are required.

title (str): The dialog box title.

message (str): The dialog box message body

Returns A dictionary containing the task configuration

Return type dict

CLI Example:

```
salt 'minion-id' task.add_action <task_name> cmd='del /Q /S C:\\Temp'
```

`salt.modules.win_task.`**`add_trigger`**(*name=None, location='\\', trigger_type=None, trigger_enabled=True, start_date=None, start_time=None, end_date=None, end_time=None, random_delay=None, repeat_interval=None, repeat_duration=None, repeat_stop_at_duration_end=False, execution_time_limit=None, delay=None, **kwargs*)

Add a trigger to a Windows Scheduled task

Note: Arguments are parsed by the YAML loader and are subject to yaml's idiosyncrasies. Therefore, time values in some formats (`%H:%M:%S` and `%H:%M`) should to be quoted. See YAML IDIOSYNCRASIES for more details.

Parameters

- **name** (*str*) – The name of the task to which to add the trigger.
- **location** (*str*) – A string value representing the location of the task. Default is \ which is the root for the task scheduler (`C:\Windows\System32\tasks`).
- **trigger_type** (*str*) – The type of trigger to create. This is defined when the trigger is created and cannot be changed later. Options are as follows:
  - Event
  - Once
  - Daily
  - Weekly
  - Monthly
  - MonthlyDay
  - OnIdle
  - OnTaskCreation
  - OnBoot

    - OnLogon
    - OnSessionChange
- **trigger_enabled** (*bool*) – Boolean value that indicates whether the trigger is enabled.
- **start_date** (*str*) – The date when the trigger is activated. If no value is passed, the current date will be used. Can be one of the following formats:
    - %Y-%m-%d
    - %m-%d-%y
    - %m-%d-%Y
    - %m/%d/%y
    - %m/%d/%Y
    - %Y/%m/%d
- **start_time** (*str*) – The time when the trigger is activated. If no value is passed, midnight will be used. Can be one of the following formats:
    - %I:%M:%S %p
    - %I:%M %p
    - %H:%M:%S
    - %H:%M
- **end_date** (*str*) – The date when the trigger is deactivated. The trigger cannot start the task after it is deactivated. Can be one of the following formats:
    - %Y-%m-%d
    - %m-%d-%y
    - %m-%d-%Y
    - %m/%d/%y
    - %m/%d/%Y
    - %Y/%m/%d
- **end_time** (*str*) – The time when the trigger is deactivated. If this is not passed with `end_date` it will be set to midnight. Can be one of the following formats:
    - %I:%M:%S %p
    - %I:%M %p
    - %H:%M:%S
    - %H:%M
- **random_delay** (*str*) – The delay time that is randomly added to the start time of the trigger. Valid values are:
    - 30 seconds
    - 1 minute
    - 30 minutes
    - 1 hour
    - 8 hours
    - 1 day

  Note: This parameter applies to the following trigger types
    - Once
    - Daily
    - Weekly
    - Monthly
    - MonthlyDay

- **repeat_interval** (*str*) – The amount of time between each restart of the task. Valid values are:
    - 5 minutes
    - 10 minutes
    - 15 minutes
    - 30 minutes

  - 1 hour
- **repeat_duration** (*str*) – How long the pattern is repeated. Valid values are:
  - Indefinitely
  - 15 minutes
  - 30 minutes
  - 1 hour
  - 12 hours
  - 1 day
- **repeat_stop_at_duration_end** (*bool*) – Boolean value that indicates if a running instance of the task is stopped at the end of the repetition pattern duration.
- **execution_time_limit** (*str*) – The maximum amount of time that the task launched by the trigger is allowed to run. Valid values are:
  - 30 minutes
  - 1 hour
  - 2 hours
  - 4 hours
  - 8 hours
  - 12 hours
  - 1 day
  - 3 days (default)
- **delay** (*str*) – The time the trigger waits after its activation to start the task. Valid values are:
  - 15 seconds
  - 30 seconds
  - 1 minute
  - 30 minutes
  - 1 hour
  - 8 hours
  - 1 day

Note: This parameter applies to the following trigger types:
  - OnLogon
  - OnBoot
  - Event
  - OnTaskCreation
  - OnSessionChange

kwargs

There are optional keyword arguments determined by the type of trigger being defined. They are as follows:

Event

The trigger will be fired by an event.

subscription (str): An event definition in xml format that fires the trigger. The easiest way to get this would is to create an event in Windows Task Scheduler and then copy the xml text.

Once

No special parameters required.

Daily

The task will run daily.

days_interval (int): The interval between days in the schedule. An interval of 1 produces a daily schedule. An interval of 2 produces an every-other day schedule. If no interval is specified, 1 is used. Valid entries are 1 - 999.

Weekly

The task will run weekly.

weeks_interval (int): The interval between weeks in the schedule. An interval of 1 produces a weekly schedule. An interval of 2 produces an every-other week schedule. If no interval is specified, 1 is used. Valid entries are 1 - 52.

days_of_week (list): Sets the days of the week on which the task runs. Should be a list. ie: `['Monday','Wednesday','Friday']`. Valid entries are the names of the days of the week.

Monthly

The task will run monthly.

months_of_year (list): Sets the months of the year during which the task runs. Should be a list. ie: `['January','July']`. Valid entries are the full names of all the months.

days_of_month (list): Sets the days of the month during which the task runs. Should be a list. ie: `[1, 15, 'Last']`. Options are all days of the month 1 - 31 and the word 'Last' to indicate the last day of the month.

last_day_of_month (bool): Boolean value that indicates that the task runs on the last day of the month regardless of the actual date of that day.

> **Note**: You can set the task to run on the last day of the month by either including the word 'Last' in the list of days, or setting the parameter 'last_day_of_month' equal to `True`.

MonthlyDay

The task will run monthly on the specified day.

months_of_year (list): Sets the months of the year during which the task runs. Should be a list. ie: `['January','July']`. Valid entries are the full names of all the months.

weeks_of_month (list): Sets the weeks of the month during which the task runs. Should be a list. ie: `['First','Third']`. Valid options are:

- First
- Second
- Third
- Fourth

last_week_of_month (bool): Boolean value that indicates that the task runs on the last week of the month.

days_of_week (list): Sets the days of the week during which the task runs. Should be a list. ie: `['Monday','Wednesday','Friday']`. Valid entries are the names of the days of the week.

OnIdle

No special parameters required.

OnTaskCreation

No special parameters required.

OnBoot

No special parameters required.

OnLogon

No special parameters required.

OnSessionChange

The task will be triggered by a session change.

session_user_name (str): Sets the user for the Terminal Server session. When a session state change is detected for this user, a task is started. To detect session status change for any user, do not pass this parameter.

state_change (str): Sets the kind of Terminal Server session change that would trigger a task launch. Valid options are:

- ConsoleConnect: When you connect to a user session (switch users)
- ConsoleDisconnect: When you disconnect a user session (switch users)
- RemoteConnect: When a user connects via Remote Desktop
- RemoteDisconnect: When a user disconnects via Remote Desktop
- SessionLock: When the workstation is locked
- SessionUnlock: When the workstation is unlocked

Returns `True` if successful, otherwise `False`

Return type bool

CLI Example:

```
salt 'minion-id' task.add_trigger <task_name> trigger_type=Once trigger_
↪enabled=True start_date=2016/12/1 start_time='"12:01"'
```

`salt.modules.win_task.`**`clear_triggers`**(*name*, *location='\\'*)

Remove all triggers from the task.

Parameters

- **`name`** (*str*) – The name of the task from which to clear all triggers.
- **`location`** (*str*) – A string value representing the location of the task. Default is \ which is the root for the task scheduler (`C:\Windows\System32\tasks`).

Returns `True` if successful, otherwise `False`

Return type bool

CLI Example:

```
salt 'minion-id' task.clear_trigger <task_name>
```

`salt.modules.win_task.`**`create_folder`**(*name*, *location='\\'*)

Create a folder in which to create tasks.

Parameters

- **`name`** (*str*) – The name of the folder. This will be displayed in the task scheduler.
- **`location`** (*str*) – A string value representing the location in which to create the folder. Default is \ which is the root for the task scheduler (`C:\Windows\System32\tasks`).

Returns `True` if successful, otherwise `False`

Return type bool

CLI Example:

```
salt 'minion-id' task.create_folder <folder_name>
```

`salt.modules.win_task.`**`create_task`**(*name*, *location='\\'*, *user_name='System'*, *password=None*, *force=False*, ***kwargs*)

Create a new task in the designated location. This function has many keyword arguments that are not listed here. For additional arguments see:

- *`edit_task()`*
- *`add_action()`*
- *`add_trigger()`*

Parameters

- **`name`** (*str*) – The name of the task. This will be displayed in the task scheduler.
- **`location`** (*str*) – A string value representing the location in which to create the task. Default is \ which is the root for the task scheduler (`C:\Windows\System32\tasks`).
- **`user_name`** (*str*) – The user account under which to run the task. To specify the 'System' account, use 'System'. The password will be ignored.
- **`password`** (*str*) – The password to use for authentication. This should set the task to run whether the user is logged in or not, but is currently not working.

- **force** (*bool*) – If the task exists, overwrite the existing task.

Returns `True` if successful, otherwise `False`

Return type bool

CLI Example:

```
salt 'minion-id' task.create_task <task_name> user_name=System force=True action_
↪type=Execute cmd='del /Q /S C:\\Temp' trigger_type=Once start_date=2016-12-1⊠
↪start_time=01:00
```

salt.modules.win_task.**create_task_from_xml**(*name, location='\\', xml_text=None, xml_path=None, user_name='System', password=None*)

Create a task based on XML. Source can be a file or a string of XML.

Parameters

- **name** (*str*) – The name of the task. This will be displayed in the task scheduler.
- **location** (*str*) – A string value representing the location in which to create the task. Default is \ which is the root for the task scheduler (`C:\Windows\System32\tasks`).
- **xml_text** (*str*) – A string of xml representing the task to be created. This will be overridden by `xml_path` if passed.
- **xml_path** (*str*) – The path to an XML file on the local system containing the xml that defines the task. This will override `xml_text`
- **user_name** (*str*) – The user account under which to run the task. To specify the 'System' account, use 'System'. The password will be ignored.
- **password** (*str*) – The password to use for authentication. This should set the task to run whether the user is logged in or not, but is currently not working.

Returns `True` if successful, otherwise `False`

Return type bool

CLI Example:

```
salt '*' task.create_task_from_xml <task_name> xml_path=C:\task.xml
```

salt.modules.win_task.**delete_folder**(*name, location='\\'*)

Delete a folder from the task scheduler.

Parameters

- **name** (*str*) – The name of the folder to delete.
- **location** (*str*) – A string value representing the location of the folder. Default is \ which is the root for the task scheduler (`C:\Windows\System32\tasks`).

Returns `True` if successful, otherwise `False`

Return type bool

CLI Example:

```
salt 'minion-id' task.delete_folder <folder_name>
```

salt.modules.win_task.**delete_task**(*name, location='\\'*)

Delete a task from the task scheduler.

Parameters

- **name** (*str*) – The name of the task to delete.
- **location** (*str*) – A string value representing the location of the task. Default is \ which is the root for the task scheduler (`C:\Windows\System32\tasks`).

Returns `True` if successful, otherwise `False`

Return type bool

CLI Example:

```
salt 'minion-id' task.delete_task <task_name>
```

`salt.modules.win_task.`**`edit_task`**(*name=None, location='\\', user_name=None, password=None, description=None, enabled=None, hidden=None, run_if_idle=None, idle_duration=None, idle_wait_timeout=None, idle_stop_on_end=None, idle_restart=None, ac_only=None, stop_if_on_batteries=None, wake_to_run=None, run_if_network=None, network_id=None, network_name=None, allow_demand_start=None, start_when_available=None, restart_every=None, restart_count=3, execution_time_limit=None, force_stop=None, delete_after=None, multiple_instances=None, **kwargs*)

Edit the parameters of a task. Triggers and Actions cannot be edited yet.

Parameters

- **`name`** (*`str`*) – The name of the task. This will be displayed in the task scheduler.
- **`location`** (*`str`*) – A string value representing the location in which to create the task. Default is \ which is the root for the task scheduler (`C:\Windows\System32\tasks`).
- **`user_name`** (*`str`*) – The user account under which to run the task. To specify the 'System' account, use 'System'. The password will be ignored.
- **`password`** (*`str`*) – The password to use for authentication. This should set the task to run whether the user is logged in or not, but is currently not working.

Note: The combination of user_name and password determine how the task runs. For example, if a username is passed without at password the task will only run when the user is logged in. If a password is passed as well the task will run whether the user is logged on or not. If you pass 'System' as the username the task will run as the system account (the password parameter is ignored).

- **`description`** (*`str`*) – A string representing the text that will be displayed in the description field in the task scheduler.
- **`enabled`** (*`bool`*) – A boolean value representing whether or not the task is enabled.
- **`hidden`** (*`bool`*) – A boolean value representing whether or not the task is hidden.
- **`run_if_idle`** (*`bool`*) – Boolean value that indicates that the Task Scheduler will run the task only if the computer is in an idle state.
- **`idle_duration`** (*`str`*) – A value that indicates the amount of time that the computer must be in an idle state before the task is run. Valid values are:
  - 1 minute
  - 5 minutes
  - 10 minutes
  - 15 minutes
  - 30 minutes
  - 1 hour
- **`idle_wait_timeout`** (*`str`*) – A value that indicates the amount of time that the Task Scheduler will wait for an idle condition to occur. Valid values are:
  - Do not wait
  - 1 minute
  - 5 minutes
  - 10 minutes
  - 15 minutes
  - 30 minutes
  - 1 hour
  - 2 hours

- **idle_stop_on_end** (*bool*) – Boolean value that indicates that the Task Scheduler will terminate the task if the idle condition ends before the task is completed.
- **idle_restart** (*bool*) – Boolean value that indicates whether the task is restarted when the computer cycles into an idle condition more than once.
- **ac_only** (*bool*) – Boolean value that indicates that the Task Scheduler will launch the task only while on AC power.
- **stop_if_on_batteries** (*bool*) – Boolean value that indicates that the task will be stopped if the computer begins to run on battery power.
- **wake_to_run** (*bool*) – Boolean value that indicates that the Task Scheduler will wake the computer when it is time to run the task.
- **run_if_network** (*bool*) – Boolean value that indicates that the Task Scheduler will run the task only when a network is available.
- **network_id** (*guid*) – GUID value that identifies a network profile.
- **network_name** (*str*) – Sets the name of a network profile. The name is used for display purposes.
- **allow_demand_start** (*bool*) – Boolean value that indicates that the task can be started by using either the Run command or the Context menu.
- **start_when_available** (*bool*) – Boolean value that indicates that the Task Scheduler can start the task at any time after its scheduled time has passed.
- **restart_every** (*str*) – A value that specifies the interval between task restart attempts. Valid values are:
  - False (to disable)
  - 1 minute
  - 5 minutes
  - 10 minutes
  - 15 minutes
  - 30 minutes
  - 1 hour
  - 2 hours
- **restart_count** (*int*) – The number of times the Task Scheduler will attempt to restart the task. Valid values are integers 1 - 999.
- **execution_time_limit** (*bool, str*) – The amount of time allowed to complete the task. Valid values are:
  - False (to disable)
  - 1 hour
  - 2 hours
  - 4 hours
  - 8 hours
  - 12 hours
  - 1 day
  - 3 days
- **force_stop** (*bool*) – Boolean value that indicates that the task may be terminated by using TerminateProcess.
- **delete_after** (*bool, str*) – The amount of time that the Task Scheduler will wait before deleting the task after it expires. Requires a trigger with an expiration date. Valid values are:
  - False (to disable)
  - Immediately
  - 30 days
  - 90 days
  - 180 days
  - 365 days
- **multiple_instances** (*str*) – Sets the policy that defines how the Task Scheduler deals with multiple instances of the task. Valid values are:

  - Parallel
  - Queue
  - No New Instance
  - Stop Existing

Returns `True` if successful, otherwise `False`

Return type bool

CLI Example:

```
salt '*' task.edit_task <task_name> description='This task is awesome'
```

`salt.modules.win_task.`**`info`**`(`*name, location='\\'*`)`

Get the details about a task in the task scheduler.

Parameters

- **`name`** (*str*) – The name of the task for which to return the status
- **`location`** (*str*) – A string value representing the location of the task. Default is \ which is the root for the task scheduler (`C:\Windows\System32\tasks`).

Returns A dictionary containing the task configuration

Return type dict

CLI Example:

```
salt 'minion-id' task.info <task_name>
```

`salt.modules.win_task.`**`list_actions`**`(`*name, location='\\'*`)`

List all actions that pertain to a task in the specified location.

Parameters

- **`name`** (*str*) – The name of the task for which list actions.
- **`location`** (*str*) – A string value representing the location of the task from which to list actions. Default is \ which is the root for the task scheduler (`C:\Windows\System32\tasks`).

Returns Returns a list of actions.

Return type list

CLI Example:

```
# List all actions for a task in the default location
salt 'minion-id' task.list_actions <task_name>

# List all actions for the XblGameSaveTask in the Microsoft\XblGameSave
# location
salt 'minion-id' task.list_actions XblGameSaveTask Microsoft\XblGameSave
```

`salt.modules.win_task.`**`list_folders`**`(`*location='\\'*`)`

List all folders located in a specific location in the task scheduler.

Parameters **`location`** (*str*) – A string value representing the folder from which you want to list tasks. Default is \ which is the root for the task scheduler (`C:\Windows\System32\tasks`).

Returns Returns a list of folders.

Return type list

CLI Example:

```
# List all folders in the default location
salt 'minion-id' task.list_folders

# List all folders in the Microsoft directory
salt 'minion-id' task.list_folders Microsoft
```

`salt.modules.win_task.`**`list_tasks`**`(`*location='\\'*`)`

List all tasks located in a specific location in the task scheduler.

Parameters **location** (*str*) – A string value representing the folder from which you want to list tasks. Default is \ which is the root for the task scheduler (`C:\Windows\System32\tasks`).

Returns Returns a list of tasks

Return type list

CLI Example:

```
# List all tasks in the default location
salt 'minion-id' task.list_tasks

# List all tasks in the Microsoft\XblGameSave Directory
salt 'minion-id' task.list_tasks Microsoft\XblGameSave
```

`salt.modules.win_task.`**`list_triggers`**(*name*, *location='\\'*)

List all triggers that pertain to a task in the specified location.

Parameters

- **name** (*str*) – The name of the task for which list triggers.
- **location** (*str*) – A string value representing the location of the task from which to list triggers. Default is \ which is the root for the task scheduler (`C:\Windows\System32\tasks`).

Returns Returns a list of triggers.

Return type list

CLI Example:

```
# List all triggers for a task in the default location
salt 'minion-id' task.list_triggers <task_name>

# List all triggers for the XblGameSaveTask in the Microsoft\XblGameSave
# location
salt '*' task.list_triggers XblGameSaveTask Microsoft\XblGameSave
```

`salt.modules.win_task.`**`run`**(*name*, *location='\\'*)

Run a scheduled task manually.

Parameters

- **name** (*str*) – The name of the task to run.
- **location** (*str*) – A string value representing the location of the task. Default is \ which is the root for the task scheduler (`C:\Windows\System32\tasks`).

Returns `True` if successful, otherwise `False`

Return type bool

CLI Example:

```
salt 'minion-id' task.list_run <task_name>
```

`salt.modules.win_task.`**`run_wait`**(*name*, *location='\\'*)

Run a scheduled task and return when the task finishes

Parameters

- **name** (*str*) – The name of the task to run.
- **location** (*str*) – A string value representing the location of the task. Default is \ which is the root for the task scheduler (`C:\Windows\System32\tasks`).

Returns `True` if successful, otherwise `False`

Return type bool

CLI Example:

```
salt 'minion-id' task.list_run_wait <task_name>
```

`salt.modules.win_task.`**`status`**(*name*, *location='\\'*)

Determine the status of a task. Is it Running, Queued, Ready, etc.

Parameters

- **name** (*str*) – The name of the task for which to return the status
- **location** (*str*) – A string value representing the location of the task. Default is \ which is the root for the task scheduler (`C:\Windows\System32\tasks`).

Returns

The current status of the task. Will be one of the following:

- Unknown
- Disabled
- Queued
- Ready
- Running

Return type str

CLI Example:

```
salt 'minion-id' task.list_status <task_name>
```

`salt.modules.win_task.`**`stop`**(*name*, *location='\\'*)

Stop a scheduled task.

Parameters

- **name** (*str*) – The name of the task to stop.
- **location** (*str*) – A string value representing the location of the task. Default is \ which is the root for the task scheduler (`C:\Windows\System32\tasks`).

Returns `True` if successful, otherwise `False`

Return type bool

CLI Example:

```
salt 'minion-id' task.list_stop <task_name>
```

21.9.484 salt.modules.win_timezone

Module for managing timezone on Windows systems.

`class` `salt.modules.win_timezone.`**`TzMapper`**(*unix_to_win*)

`add`(*k*, *v*)

`get_unix`(*key*, *default=None*)

`get_win`(*key*, *default=None*)

`list_unix`()

`list_win`()

`remove`(*k*)

`salt.modules.win_timezone.`**`get_hwclock`**()

Get current hardware clock setting (UTC or localtime)

Note: The hardware clock is always local time on Windows so this will always return "localtime"

CLI Example:

```
salt '*' timezone.get_hwclock
```

`salt.modules.win_timezone.`**`get_offset`**`()`

Get current numeric timezone offset from UTC (i.e. -0700)

Returns Offset from UTC

Return type str

CLI Example:

```
salt '*' timezone.get_offset
```

`salt.modules.win_timezone.`**`get_zone`**`()`

Get current timezone (i.e. America/Denver)

Returns Timezone in unix format

Return type str

CLI Example:

```
salt '*' timezone.get_zone
```

`salt.modules.win_timezone.`**`get_zonecode`**`()`

Get current timezone (i.e. PST, MDT, etc)

Returns An abbreviated timezone code

Return type str

CLI Example:

```
salt '*' timezone.get_zonecode
```

`salt.modules.win_timezone.`**`list`**`(`*unix_style=True*`)`

Return a list of Timezones that this module supports. These can be in either Unix or Windows format.

New in version 2018.3.3.

Parameters **`unix_style`** (*bool*) – `True` returns Unix-style timezones. `False` returns Windows-style timezones. Default is `True`

Returns A list of supported timezones

Return type list

CLI Example:

```
# Unix-style timezones
salt '*' timezone.list

# Windows-style timezones
salt '*' timezone.list unix_style=False
```

`salt.modules.win_timezone.`**`set_hwclock`**`(`*clock*`)`

Sets the hardware clock to be either UTC or localtime

Note: The hardware clock is always local time on Windows so this will always return `False`

CLI Example:

```
salt '*' timezone.set_hwclock UTC
```

`salt.modules.win_timezone.`**`set_zone`**`(`*timezone*`)`

Sets the timezone using the tzutil.

Parameters **`timezone`** (*str*) – A valid timezone

Returns `True` if successful, otherwise `False`

Return type bool

Raises ***CommandExecutionError*** – If invalid timezone is passed

CLI Example:

```
salt '*' timezone.set_zone 'America/Denver'
```

`salt.modules.win_timezone.`**`zone_compare`**(*timezone*)

Compares the given timezone with the machine timezone. Mostly useful for running state checks.

Parameters **`timezone`** (*`str`*) – The timezone to compare. This can be in Windows or Unix format. Can be any of the values returned by the `timezone.list` function

Returns `True` if they match, otherwise `False`

Return type bool

Example:

```
salt '*' timezone.zone_compare 'America/Denver'
```

21.9.485 salt.modules.win_useradd

Module for managing Windows Users

Important: If you feel that Salt should be using this module to manage users on a minion, and it is using a different module (or gives an error similar to *'user.info' is not available*), see *here*.

---

depends

- pywintypes
- win32api
- win32con
- win32net
- win32netcon
- win32profile
- win32security
- win32ts
- wmi

Note: This currently only works with local user accounts, not domain accounts

---

`salt.modules.win_useradd.`**`add`**(*name, password=None, fullname=None, description=None, groups=None, home=None, homedrive=None, profile=None, logonscript=None*)

Add a user to the minion.

Parameters

- **`name`** (*`str`*) – User name
- **`password`** (*`str, optional`*) – User's password in plain text.
- **`fullname`** (*`str, optional`*) – The user's full name.
- **`description`** (*`str, optional`*) – A brief description of the user account.
- **`groups`** (*`str, optional`*) – A list of groups to add the user to. (see chgroups)
- **`home`** (*`str, optional`*) – The path to the user's home directory.

- **homedrive** (*str, optional*) – The drive letter to assign to the home directory. Must be the Drive Letter followed by a colon. ie: U:
- **profile** (*str, optional*) – An explicit path to a profile. Can be a UNC or a folder on the system. If left blank, windows uses it's default profile directory.
- **logonscript** (*str, optional*) – Path to a login script to run when the user logs on.

Returns True if successful. False is unsuccessful.

Return type bool

CLI Example:

```
salt '*' user.add name password
```

salt.modules.win_useradd.**addgroup**(*name, group*)

Add user to a group

Parameters

- **name** (*str*) – The user name to add to the group
- **group** (*str*) – The name of the group to which to add the user

Returns True if successful, otherwise False

Return type bool

CLI Example:

```
salt '*' user.addgroup jsnuffy 'Power Users'
```

salt.modules.win_useradd.**chfullname**(*name, fullname*)

Change the full name of the user

Parameters

- **name** (*str*) – The user name for which to change the full name
- **fullname** (*str*) – The new value for the full name

Returns True if successful, otherwise False

Return type bool

CLI Example:

```
salt '*' user.chfullname user 'First Last'
```

salt.modules.win_useradd.**chgroups**(*name, groups, append=True*)

Change the groups this user belongs to, add append=False to make the user a member of only the specified groups

Parameters

- **name** (*str*) – The user name for which to change groups
- **groups** (*str, list*) – A single group or a list of groups to assign to the user. For multiple groups this can be a comma delimited string or a list.
- **append** (*bool, optional*) – True adds the passed groups to the user's current groups. False sets the user's groups to the passed groups only. Default is True.

Returns True if successful, otherwise False

Return type bool

CLI Example:

```
salt '*' user.chgroups jsnuffy Administrators,Users True
```

salt.modules.win_useradd.**chhome**(*name, home, **kwargs*)

Change the home directory of the user, pass True for persist to move files to the new home directory if the old home directory exist.

Parameters

- **name** (*str*) – The name of the user whose home directory you wish to change
- **home** (*str*) – The new location of the home directory

Returns True if successful, otherwise False

Return type bool

CLI Example:

```
salt '*' user.chhome foo \\fileserver\home\foo True
```

`salt.modules.win_useradd.`**`chprofile`**(*name*, *profile*)

Change the profile directory of the user

Parameters

- **name** (*str*) – The name of the user whose profile you wish to change
- **profile** (*str*) – The new location of the profile

Returns True if successful, otherwise False

Return type bool

CLI Example:

```
salt '*' user.chprofile foo \\fileserver\profiles\foo
```

`salt.modules.win_useradd.`**`current`**(*sam=False*)

Get the username that salt-minion is running under. If salt-minion is running as a service it should return the Local System account. If salt is running from a command prompt it should return the username that started the command prompt.

New in version 2015.5.6.

Parameters **sam** (*bool, optional*) – False returns just the username without any domain notation. True returns the domain with the username in the SAM format. Ie: `domain\username`

Returns Returns username

Return type str

CLI Example:

```
salt '*' user.current
```

`salt.modules.win_useradd.`**`delete`**(*name*, *purge=False*, *force=False*)

Remove a user from the minion

Parameters

- **name** (*str*) – The name of the user to delete
- **purge** (*bool, optional*) – Boolean value indicating that the user profile should also be removed when the user account is deleted. If set to True the profile will be removed. Default is False.
- **force** (*bool, optional*) – Boolean value indicating that the user account should be deleted even if the user is logged in. True will log the user out and delete user.

Returns True if successful, otherwise False

Return type bool

CLI Example:

```
salt '*' user.delete name
```

`salt.modules.win_useradd.`**`getUserSid`**(*username*)

Get the Security ID for the user

Parameters **username** (*str*) – The user name for which to look up the SID

Returns The user SID

Return type str

CLI Example:

```
salt '*' user.getUserSid jsnuffy
```

`salt.modules.win_useradd.`**`getent`**`(`*refresh=False*`)`

Return the list of all info for all users

Parameters **`refresh`** (*`bool, optional`*) – Refresh the cached user information. Useful when used from within a state function. Default is False.

Returns A dictionary containing information about all users on the system

Return type dict

CLI Example:

```
salt '*' user.getent
```

`salt.modules.win_useradd.`**`info`**`(`*name*`)`

Return user information

Parameters **`name`** (*`str`*) – Username for which to display information

Returns

A dictionary containing user information

- fullname
- username
- SID
- passwd (will always return None)
- comment (same as description, left here for backwards compatibility)
- description
- active
- logonscript
- profile
- home
- homedrive
- groups
- password_changed
- successful_logon_attempts
- failed_logon_attempts
- last_logon
- account_disabled
- account_locked
- password_never_expires
- disallow_change_password
- gid

Return type dict

CLI Example:

```
salt '*' user.info jsnuffy
```

`salt.modules.win_useradd.`**`list_groups`**`(`*name*`)`

Return a list of groups the named user belongs to

Parameters **`name`** (*`str`*) – The user name for which to list groups

Returns A list of groups to which the user belongs

Return type list

CLI Example:

```
salt '*' user.list_groups foo
```

`salt.modules.win_useradd.`**`list_users`**`()`

Return a list of all users on Windows

Returns A list of all users on the system

Return type list

CLI Example:

```
salt '*' user.list_users
```

`salt.modules.win_useradd.`**`removegroup`**(*name*, *group*)

Remove user from a group

Parameters

- **`name`** (*`str`*) – The user name to remove from the group
- **`group`** (*`str`*) – The name of the group from which to remove the user

Returns True if successful, otherwise False

Return type bool

CLI Example:

```
salt '*' user.removegroup jsnuffy 'Power Users'
```

`salt.modules.win_useradd.`**`rename`**(*name*, *new_name*)

Change the username for a named user

Parameters

- **`name`** (*`str`*) – The user name to change
- **`new_name`** (*`str`*) – The new name for the current user

Returns True if successful, otherwise False

Return type bool

CLI Example:

```
salt '*' user.rename jsnuffy jshmoe
```

`salt.modules.win_useradd.`**`setpassword`**(*name*, *password*)

Set the user's password

Parameters

- **`name`** (*`str`*) – The user name for which to set the password
- **`password`** (*`str`*) – The new password

Returns True if successful, otherwise False

Return type bool

CLI Example:

```
salt '*' user.setpassword jsnuffy sup3rs3cr3t
```

`salt.modules.win_useradd.`**`update`**(*name*, *password=None*, *fullname=None*, *description=None*, *home=None*, *homedrive=None*, *logonscript=None*, *profile=None*, *expiration_date=None*, *expired=None*, *account_disabled=None*, *unlock_account=None*, *password_never_expires=None*, *disallow_change_password=None*)

Updates settings for the windows user. Name is the only required parameter. Settings will only be changed if the parameter is passed a value.

New in version 2015.8.0.

Parameters

- **`name`** (*`str`*) – The user name to update.
- **`password`** (*`str, optional`*) – New user password in plain text.
- **`fullname`** (*`str, optional`*) – The user's full name.
- **`description`** (*`str, optional`*) – A brief description of the user account.
- **`home`** (*`str, optional`*) – The path to the user's home directory.
- **`homedrive`** (*`str, optional`*) – The drive letter to assign to the home directory. Must be the Drive Letter followed by a colon. ie: U:
- **`logonscript`** (*`str, optional`*) – The path to the logon script.
- **`profile`** (*`str, optional`*) – The path to the user's profile directory.

- **expiration_date** (*date, optional*) – The date and time when the account expires. Can be a valid date/time string. To set to never expire pass the string 'Never'.
- **expired** (*bool, optional*) – Pass *True* to expire the account. The user will be prompted to change their password at the next logon. Pass *False* to mark the account as 'not expired'. You can't use this to negate the expiration if the expiration was caused by the account expiring. You'll have to change the *expiration_date* as well.
- **account_disabled** (*bool, optional*) – True disables the account. False enables the account.
- **unlock_account** (*bool, optional*) – True unlocks a locked user account. False is ignored.
- **password_never_expires** (*bool, optional*) – True sets the password to never expire. False allows the password to expire.
- **disallow_change_password** (*bool, optional*) – True blocks the user from changing the password. False allows the user to change the password.

Returns True if successful. False is unsuccessful.

Return type bool

CLI Example:

```
salt '*' user.update bob password=secret profile=C:\Users\Bob
         home=\server\homeshareob homedrive=U:
```

21.9.486 salt.modules.win_wua

Module for managing Windows Updates using the Windows Update Agent.

List updates on the system using the following functions:

- *win_wua.available*
- `win_wua.list`

This is an easy way to find additional information about updates available to to the system, such as the GUID, KB number, or description.

Once you have the GUID or a KB number for the update you can get information about the update, download, install, or uninstall it using these functions:

- *win_wua.get*
- *win_wua.download*
- *win_wua.install*
- *win_wua.uninstall*

The get function expects a name in the form of a GUID, KB, or Title and should return information about a single update. The other functions accept either a single item or a list of items for downloading/installing/uninstalling a specific list of items.

The `win_wua.list` and *win_wua.get* functions are utility functions. In addition to returning information about updates they can also download and install updates by setting `download=True` or `install=True`. So, with py:func:*win_wua.list <salt.modules.win_wua.list_>* for example, you could run the function with the filters you want to see what is available. Then just add `install=True` to install everything on that list.

If you want to download, install, or uninstall specific updates, use *win_wua.download*, *win_wua.install*, or *win_wua.uninstall*. To update your system with the latest updates use `win_wua.list` and set `install=True`

You can also adjust the Windows Update settings using the *win_wua.set_wu_settings* function. This function is only supported on the following operating systems:

- Windows Vista / Server 2008
- Windows 7 / Server 2008R2
- Windows 8 / Server 2012
- Windows 8.1 / Server 2012R2

As of Windows 10 and Windows Server 2016, the ability to modify the Windows Update settings has been restricted. The settings can be modified in the Local Group Policy using the `lgpo` module.

New in version 2015.8.0.

depends salt.utils.win_update

`salt.modules.win_wua.`**`available`**(*software=True*, *drivers=True*, *summary=False*, *skip_installed=True*, *skip_hidden=True*, *skip_mandatory=False*, *skip_reboot=False*, *categories=None*, *severities=None*)

New in version 2017.7.0.

List updates that match the passed criteria. This allows for more filter options than `list()`. Good for finding a specific GUID or KB.

Parameters

- **`software`** (*`bool`*) – Include software updates in the results (default is True)
- **`drivers`** (*`bool`*) – Include driver updates in the results (default is True)
- **`summary`** (*`bool`*) –
  - True: Return a summary of updates available for each category.
  - False (default): Return a detailed list of available updates.
- **`skip_installed`** (*`bool`*) – Skip updates that are already installed. Default is False.
- **`skip_hidden`** (*`bool`*) – Skip updates that have been hidden. Default is True.
- **`skip_mandatory`** (*`bool`*) – Skip mandatory updates. Default is False.
- **`skip_reboot`** (*`bool`*) – Skip updates that require a reboot. Default is False.
- **`categories`** (*`list`*) – Specify the categories to list. Must be passed as a list. All categories returned by default.

  Categories include the following:
  - Critical Updates
  - Definition Updates
  - Drivers (make sure you set drivers=True)
  - Feature Packs
  - Security Updates
  - Update Rollups
  - Updates
  - Update Rollups
  - Windows 7
  - Windows 8.1
  - Windows 8.1 drivers
  - Windows 8.1 and later drivers
  - Windows Defender
- **`severities`** (*`list`*) – Specify the severities to include. Must be passed as a list. All severities returned by default.

  Severities include the following:
  - Critical
  - Important

Returns

Returns a dict containing either a summary or a list of updates:

```
List of Updates:
{'<GUID>': {'Title': <title>,
            'KB': <KB>,
            'GUID': <the globally unique identifier for the update>
            'Description': <description>,
            'Downloaded': <has the update been downloaded>,
            'Installed': <has the update been installed>,
            'Mandatory': <is the update mandatory>,
            'UserInput': <is user input required>,
            'EULAAccepted': <has the EULA been accepted>,
            'Severity': <update severity>,
            'NeedsReboot': <is the update installed and awaiting
↪reboot>,
            'RebootBehavior': <will the update require a reboot>,
            'Categories': [ '<category 1>',
                            '<category 2>',
                            ...]
            }
}

Summary of Updates:
{'Total': <total number of updates returned>,
 'Available': <updates that are not downloaded or installed>,
 'Downloaded': <updates that are downloaded but not installed>,
 'Installed': <updates installed (usually 0 unless installed=True)>,
 'Categories': { <category 1>: <total for that category>,
                 <category 2>: <total for category 2>,
                 ... }
}
```

Return type dict

CLI Examples:

```
# Normal Usage (list all software updates)
salt '*' win_wua.available

# List all updates with categories of Critical Updates and Drivers
salt '*' win_wua.available categories=["Critical Updates","Drivers"]

# List all Critical Security Updates
salt '*' win_wua.available categories=["Security Updates"] severities=["Critical"]

# List all updates with a severity of Critical
salt '*' win_wua.available severities=["Critical"]

# A summary of all available updates
salt '*' win_wua.available summary=True

# A summary of all Feature Packs and Windows 8.1 Updates
salt '*' win_wua.available categories=["Feature Packs","Windows 8.1"] summary=True
```

`salt.modules.win_wua.`**`download`**(*names*)

New in version 2017.7.0.

Downloads updates that match the list of passed identifiers. It's easier to use this function by using list_updates and setting install=True.

Parameters **names** (*str, list*) – A single update or a list of updates to download. This can be any combination of GUIDs, KB numbers, or names. GUIDs or KBs are preferred.

Note: An error will be raised if there are more results than there are items in the names parameter

Returns A dictionary containing the details about the downloaded updates

Return type dict

CLI Examples:

```
# Normal Usage
salt '*' win_wua.download names=['12345678-abcd-1234-abcd-1234567890ab',
↪'KB2131233']
```

`salt.modules.win_wua.get`(*name*, *download=False*, *install=False*)

New in version 2017.7.0.

Returns details for the named update

Parameters

- **name** (*str*) – The name of the update you're searching for. This can be the GUID, a KB number, or any part of the name of the update. GUIDs and KBs are preferred. Run `list` to get the GUID for the update you're looking for.
- **download** (*bool*) – Download the update returned by this function. Run this function first to see if the update exists, then set `download=True` to download the update.
- **install** (*bool*) – Install the update returned by this function. Run this function first to see if the update exists, then set `install=True` to install the update.

Returns

Returns a dict containing a list of updates that match the name if download and install are both set to False. Should usually be a single update, but can return multiple if a partial name is given.

If download or install is set to true it will return the results of the operation.

```
List of Updates:
{'<GUID>': {'Title': <title>,
            'KB': <KB>,
            'GUID': <the globally unique identifier for the update>
            'Description': <description>,
            'Downloaded': <has the update been downloaded>,
            'Installed': <has the update been installed>,
            'Mandatory': <is the update mandatory>,
            'UserInput': <is user input required>,
            'EULAAccepted': <has the EULA been accepted>,
            'Severity': <update severity>,
            'NeedsReboot': <is the update installed and awaiting
↪reboot>,
            'RebootBehavior': <will the update require a reboot>,
            'Categories': [ '<category 1>',
                            '<category 2>',
                            ...]
            }
}
```

Return type dict

CLI Examples:

```
# Recommended Usage using GUID without braces
# Use this to find the status of a specific update
```

(continues on next page)

(continued from previous page)

```
salt '*' win_wua.get 12345678-abcd-1234-abcd-1234567890ab

# Use the following if you don't know the GUID:

# Using a KB number
# Not all updates have an associated KB
salt '*' win_wua.get KB3030298

# Using part or all of the name of the update
# Could possibly return multiple results
# Not all updates have an associated KB
salt '*' win_wua.get 'Microsoft Camera Codec Pack'
```

`salt.modules.win_wua.`**`get_needs_reboot`**`()`

Determines if the system needs to be rebooted.

Returns True if the system requires a reboot, otherwise False

Return type bool

CLI Examples:

```
salt '*' win_wua.get_needs_reboot
```

`salt.modules.win_wua.`**`get_wu_settings`**`()`

Get current Windows Update settings.

Returns

A dictionary of Windows Update settings:

Featured Updates: Boolean value that indicates whether to display notifications for featured updates.

Group Policy Required (Read-only): Boolean value that indicates whether Group Policy requires the Automatic Updates service.

Microsoft Update: Boolean value that indicates whether to turn on Microsoft Update for other Microsoft Products

Needs Reboot: Boolean value that indicates whether the machine is in a reboot pending state.

Non Admins Elevated: Boolean value that indicates whether non-administrators can perform some update-related actions without administrator approval.

Notification Level:

Number 1 to 4 indicating the update level:

1. Never check for updates
2. Check for updates but let me choose whether to download and install them
3. Download updates but let me choose whether to install them
4. Install updates automatically

Read Only (Read-only): Boolean value that indicates whether the Automatic Update settings are read-only.

Recommended Updates: Boolean value that indicates whether to include optional or recommended updates when a search for updates and installation of updates is performed.

Scheduled Day: Days of the week on which Automatic Updates installs or uninstalls updates.

Scheduled Time: Time at which Automatic Updates installs or uninstalls updates.

Return type dict

CLI Examples:

```
salt '*' win_wua.get_wu_settings
```

`salt.modules.win_wua.`**`install`**(*names*)

New in version 2017.7.0.

Installs updates that match the list of identifiers. It may be easier to use the list_updates function and set install=True.

Parameters **`names`** (*`str, list`*) – A single update or a list of updates to install. This can be any combination of GUIDs, KB numbers, or names. GUIDs or KBs are preferred.

Note: An error will be raised if there are more results than there are items in the names parameter

---

Returns A dictionary containing the details about the installed updates

Return type dict

CLI Examples:

```
# Normal Usage
salt '*' win_wua.install KB12323211
```

`salt.modules.win_wua.`**`list`**(*software=True, drivers=False, summary=False, skip_installed=True, categories=None, severities=None, download=False, install=False*)

New in version 2017.7.0.

Returns a detailed list of available updates or a summary. If download or install is True the same list will be downloaded and/or installed.

Parameters

- **`software`** (*`bool`*) – Include software updates in the results (default is True)
- **`drivers`** (*`bool`*) – Include driver updates in the results (default is False)
- **`summary`** (*`bool`*) –
  - True: Return a summary of updates available for each category.
  - False (default): Return a detailed list of available updates.
- **`skip_installed`** (*`bool`*) – Skip installed updates in the results (default is False)
- **`download`** (*`bool`*) – (Overrides reporting functionality) Download the list of updates returned by this function. Run this function first with `download=False` to see what will be downloaded, then set `download=True` to download the updates.
- **`install`** (*`bool`*) – (Overrides reporting functionality) Install the list of updates returned by this function. Run this function first with `install=False` to see what will be installed, then set `install=True` to install the updates.
- **`categories`** (*`list`*) – Specify the categories to list. Must be passed as a list. All categories returned by default.

  Categories include the following:
  - Critical Updates
  - Definition Updates
  - Drivers (make sure you set drivers=True)
  - Feature Packs
  - Security Updates
  - Update Rollups
  - Updates
  - Update Rollups
  - Windows 7
  - Windows 8.1
  - Windows 8.1 drivers
  - Windows 8.1 and later drivers

    - Windows Defender
- **severities** (*list*) – Specify the severities to include. Must be passed as a list. All severities returned by default.

  Severities include the following:
    - Critical
    - Important

Returns

Returns a dict containing either a summary or a list of updates:

```
List of Updates:
{'<GUID>': {'Title': <title>,
            'KB': <KB>,
            'GUID': <the globally unique identifier for the update>
            'Description': <description>,
            'Downloaded': <has the update been downloaded>,
            'Installed': <has the update been installed>,
            'Mandatory': <is the update mandatory>,
            'UserInput': <is user input required>,
            'EULAAccepted': <has the EULA been accepted>,
            'Severity': <update severity>,
            'NeedsReboot': <is the update installed and awaiting
↪reboot>,
            'RebootBehavior': <will the update require a reboot>,
            'Categories': [ '<category 1>',
                            '<category 2>',
                            ...]
            }
}

Summary of Updates:
{'Total': <total number of updates returned>,
 'Available': <updates that are not downloaded or installed>,
 'Downloaded': <updates that are downloaded but not installed>,
 'Installed': <updates installed (usually 0 unless installed=True)>,
 'Categories': { <category 1>: <total for that category>,
                 <category 2>: <total for category 2>,
                 ... }
}
```

Return type dict

CLI Examples:

```
# Normal Usage (list all software updates)
salt '*' win_wua.list

# List all updates with categories of Critical Updates and Drivers
salt '*' win_wua.list categories=['Critical Updates','Drivers']

# List all Critical Security Updates
salt '*' win_wua.list categories=['Security Updates'] severities=['Critical']

# List all updates with a severity of Critical
salt '*' win_wua.list severities=['Critical']

# A summary of all available updates
salt '*' win_wua.list summary=True
```

(continues on next page)

(continued from previous page)

```
# A summary of all Feature Packs and Windows 8.1 Updates
salt '*' win_wua.list categories=['Feature Packs','Windows 8.1'] summary=True
```

salt.modules.win_wua.**set_wu_settings**(*level=None, recommended=None, featured=None, elevated=None, msupdate=None, day=None, time=None*)

Change Windows Update settings. If no parameters are passed, the current value will be returned.

Supported:

- Windows Vista / Server 2008
- Windows 7 / Server 2008R2
- Windows 8 / Server 2012
- Windows 8.1 / Server 2012R2

Parameters

- **level** (*int*) – Number from 1 to 4 indicating the update level:
    1. Never check for updates
    2. Check for updates but let me choose whether to download and install them
    3. Download updates but let me choose whether to install them
    4. Install updates automatically
- **recommended** (*bool*) – Boolean value that indicates whether to include optional or recommended updates when a search for updates and installation of updates is performed.
- **featured** (*bool*) – Boolean value that indicates whether to display notifications for featured updates.
- **elevated** (*bool*) – Boolean value that indicates whether non-administrators can perform some update-related actions without administrator approval.
- **msupdate** (*bool*) – Boolean value that indicates whether to turn on Microsoft Update for other Microsoft products
- **day** (*str*) – Days of the week on which Automatic Updates installs or uninstalls updates. Accepted values:
    - Everyday
    - Monday
    - Tuesday
    - Wednesday
    - Thursday
    - Friday
    - Saturday
- **time** (*str*) – Time at which Automatic Updates installs or uninstalls updates. Must be in the ##:## 24hr format, eg. 3:00 PM would be 15:00. Must be in 1 hour increments.

Returns Returns a dictionary containing the results.

Return type dict

CLI Examples:

```
salt '*' win_wua.set_wu_settings level=4 recommended=True featured=False
```

salt.modules.win_wua.**uninstall**(*names*)

New in version 2017.7.0.

Uninstall updates.

Parameters **names** (*str, list*) – A single update or a list of updates to uninstall. This can be any combination of GUIDs, KB numbers, or names. GUIDs or KBs are preferred.

Returns A dictionary containing the details about the uninstalled updates

Return type dict

CLI Examples:

```
# Normal Usage
salt '*' win_wua.uninstall KB3121212

# As a list
salt '*' win_wua.uninstall guid=['12345678-abcd-1234-abcd-1234567890ab',
↪'KB1231231']
```

21.9.487 salt.modules.win_wusa

Microsoft Update files management via wusa.exe

maintainer Thomas Lemarchand

platform Windows

depends PowerShell

New in version 2018.3.4.

`salt.modules.win_wusa.`**`install`**(*path*, *restart=False*)

Install a KB from a .msu file.

Parameters

- **`path`** (*`str`*) – The full path to the msu file to install
- **`restart`** (*`bool`*) – `True` to force a restart if required by the installation. Adds the `/forcerestart` switch to the `wusa.exe` command. `False` will add the `/norestart` switch instead. Default is `False`

Returns `True` if successful, otherwise `False`

Return type bool

Raise:

CommandExecutionError: If the package is already installed or an error is encountered

CLI Example:

```
salt '*' wusa.install C:/temp/KB123456.msu
```

`salt.modules.win_wusa.`**`is_installed`**(*name*)

Check if a specific KB is installed.

Parameters **`name`** (*`str`*) – The name of the KB to check

Returns `True` if installed, otherwise `False`

Return type bool

CLI Example:

```
salt '*' wusa.is_installed KB123456
```

`salt.modules.win_wusa.`**`list`**()

Get a list of updates installed on the machine

Returns A list of installed updates

Return type list

CLI Example:

```
salt '*' wusa.list
```

`salt.modules.win_wusa.`**`uninstall`**(*path*, *restart=False*)

Uninstall a specific KB.

Parameters

- **`path`** (*`str`*) – The full path to the msu file to uninstall. This can also be just the name of the KB to uninstall
- **`restart`** (*`bool`*) – `True` to force a restart if required by the installation. Adds the `/forcerestart` switch to the `wusa.exe` command. `False` will add the `/norestart` switch instead. Default is `False`

Returns `True` if successful, otherwise `False`

Return type bool

Raises ***`CommandExecutionError`*** – If an error is encountered

CLI Example:

```
salt '*' wusa.uninstall KB123456

# or

salt '*' wusa.uninstall C:/temp/KB123456.msu
```

21.9.488 salt.modules.x509

Manage X509 certificates

New in version 2015.8.0.

depends M2Crypto

`salt.modules.x509.`**`create_certificate`**(*path=None*, *text=False*, *overwrite=True*, *ca_server=None*, ***kwargs*)

Create an X509 certificate.

path: Path to write the certificate to.

text: If `True`, return the PEM text without writing to a file. Default `False`.

overwrite: If `True` (default), create_certificate will overwrite the entire PEM file. Set False to preserve existing private keys and dh params that may exist in the PEM file.

kwargs: Any of the properties below can be included as additional keyword arguments.

ca_server: Request a remotely signed certificate from ca_server. For this to work, a `signing_policy` must be specified, and that same policy must be configured on the ca_server. See `signing_policy` for details. Also, the salt master must permit peers to call the `sign_remote_certificate` function.

Example:

/etc/salt/master.d/peer.conf

```
peer:
  .*:
    - x509.sign_remote_certificate
```

subject properties: Any of the values below can be included to set subject properties Any other subject properties supported by OpenSSL should also work.

C: 2 letter Country code

CN: Certificate common name, typically the FQDN.

Email: Email address

GN: Given Name

L: Locality

O: Organization

OU: Organization Unit

SN: SurName

ST: State or Province

signing_private_key: A path or string of the private key in PEM format that will be used to sign this certificate. If neither `signing_cert`, `public_key`, or `csr` are included, it will be assumed that this is a self-signed certificate, and the public key matching `signing_private_key` will be used to create the certificate.

signing_private_key_passphrase: Passphrase used to decrypt the signing_private_key.

signing_cert: A certificate matching the private key that will be used to sign this certificate. This is used to populate the issuer values in the resulting certificate. Do not include this value for self-signed certificates.

public_key: The public key to be included in this certificate. This can be sourced from a public key, certificate, CSR or private key. If a private key is used, the matching public key from the private key will be generated before any processing is done. This means you can request a certificate from a remote CA using a private key file as your public_key and only the public key will be sent across the network to the CA. If neither `public_key` or `csr` are specified, it will be assumed that this is a self-signed certificate, and the public key derived from `signing_private_key` will be used. Specify either `public_key` or `csr`, not both. Because you can input a CSR as a public key or as a CSR, it is important to understand the difference. If you import a CSR as a public key, only the public key will be added to the certificate, subject or extension information in the CSR will be lost.

public_key_passphrase: If the public key is supplied as a private key, this is the passphrase used to decrypt it.

csr: A file or PEM string containing a certificate signing request. This will be used to supply the subject, extensions and public key of a certificate. Any subject or extensions specified explicitly will overwrite any in the CSR.

basicConstraints: X509v3 Basic Constraints extension.

extensions: The following arguments set X509v3 Extension values. If the value starts with `critical`, the extension will be marked as critical.

Some special extensions are `subjectKeyIdentifier` and `authorityKeyIdentifier`.

`subjectKeyIdentifier` can be an explicit value or it can be the special string `hash`. `hash` will set the subjectKeyIdentifier equal to the SHA1 hash of the modulus of the public key in this certificate. Note that this is not the exact same hashing method used by OpenSSL when using the hash value.

`authorityKeyIdentifier` Use values acceptable to the openssl CLI tools. This will automatically populate `authorityKeyIdentifier` with the `subjectKeyIdentifier` of `signing_cert`. If this is a self-signed cert these values will be the same.

basicConstraints: X509v3 Basic Constraints
keyUsage: X509v3 Key Usage
extendedKeyUsage: X509v3 Extended Key Usage
subjectKeyIdentifier: X509v3 Subject Key Identifier
issuerAltName: X509v3 Issuer Alternative Name
subjectAltName: X509v3 Subject Alternative Name
crlDistributionPoints: X509v3 CRL Distribution Points
issuingDistributionPoint: X509v3 Issuing Distribution Point
certificatePolicies: X509v3 Certificate Policies
policyConstraints: X509v3 Policy Constraints
inhibitAnyPolicy: X509v3 Inhibit Any Policy
nameConstraints: X509v3 Name Constraints
noCheck: X509v3 OCSP No Check
nsComment: Netscape Comment
nsCertType: Netscape Certificate Type

days_valid: The number of days this certificate should be valid. This sets the `notAfter` property of the certificate. Defaults to 365.

version: The version of the X509 certificate. Defaults to 3. This is automatically converted to the version value, so `version=3` sets the certificate version field to 0x2.

serial_number: The serial number to assign to this certificate. If omitted a random serial number of size `serial_bits` is generated.

serial_bits: The number of bits to use when randomly generating a serial number. Defaults to 64.

algorithm: The hashing algorithm to be used for signing this certificate. Defaults to sha256.

copypath: An additional path to copy the resulting certificate to. Can be used to maintain a copy of all certificates issued for revocation purposes.

prepend_cn: If set to True, the CN and a dash will be prepended to the copypath's filename.

Example: /etc/pki/issued_certs/www.example.com-DE:CA:FB:AD:00:00:00:00.crt

signing_policy: A signing policy that should be used to create this certificate. Signing policies should be defined in the minion configuration, or in a minion pillar. It should be a YAML formatted list of arguments which will override any arguments passed to this function. If the `minions` key is included in the signing policy, only minions matching that pattern (see match.glob and match.compound) will be permitted to remotely request certificates from that policy.

Example:

```
x509_signing_policies:
  www:
    - minions: 'www*'
    - signing_private_key: /etc/pki/ca.key
    - signing_cert: /etc/pki/ca.crt
    - C: US
    - ST: Utah
    - L: Salt Lake City
    - basicConstraints: "critical CA:false"
    - keyUsage: "critical cRLSign, keyCertSign"
    - subjectKeyIdentifier: hash
    - authorityKeyIdentifier: keyid,issuer:always
    - days_valid: 90
    - copypath: /etc/pki/issued_certs/
```

The above signing policy can be invoked with `signing_policy=www`

CLI Example:

```
salt '*' x509.create_certificate path=/etc/pki/myca.crt signing_private_key='/etc/
↪pki/myca.key' csr='/etc/pki/myca.csr'}
```

`salt.modules.x509.`**`create_crl`**(*path=None, text=False, signing_private_key=None, signing_private_key_passphrase=None, signing_cert=None, revoked=None, include_expired=False, days_valid=100, digest=''*)

Create a CRL

Depends

- PyOpenSSL Python module

path: Path to write the CRL to.

text: If `True`, return the PEM text without writing to a file. Default `False`.

signing_private_key: A path or string of the private key in PEM format that will be used to sign the CRL. This is required.

signing_private_key_passphrase: Passphrase to decrypt the private key.

signing_cert: A certificate matching the private key that will be used to sign the CRL. This is required.

revoked: A list of dicts containing all the certificates to revoke. Each dict represents one certificate. A dict must contain either the key `serial_number` with the value of the serial number to revoke, or `certificate` with either the PEM encoded text of the certificate, or a path to the certificate to revoke.

The dict can optionally contain the `revocation_date` key. If this key is omitted the revocation date will be set to now. If should be a string in the format "%Y-%m-%d %H:%M:%S".

The dict can also optionally contain the `not_after` key. This is redundant if the `certificate` key is included. If the `Certificate` key is not included, this can be used for the logic behind the `include_expired` parameter. If should be a string in the format "%Y-%m-%d %H:%M:%S".

The dict can also optionally contain the `reason` key. This is the reason code for the revocation. Available choices are `unspecified`, `keyCompromise`, `CACompromise`, `affiliationChanged`, `superseded`, `cessationOfOperation` and `certificateHold`.

include_expired: Include expired certificates in the CRL. Default is `False`.
days_valid: The number of days that the CRL should be valid. This sets the Next Update field in the CRL.
digest: The digest to use for signing the CRL. This has no effect on versions of pyOpenSSL less than 0.14

CLI Example:

```
salt '*' x509.create_crl path=/etc/pki/mykey.key signing_private_key=/etc/pki/ca.
↪key signing_cert=/etc/pki/ca.crt revoked="{'compromized-web-key': {'certificate
↪': '/etc/pki/certs/www1.crt', 'revocation_date': '2015-03-01 00:00:00'}}"
```

`salt.modules.x509.`**`create_csr`**(*path=None*, *text=False*, ***kwargs*)
Create a certificate signing request.
path: Path to write the certificate to.
text: If `True`, return the PEM text without writing to a file. Default `False`.
algorithm: The hashing algorithm to be used for signing this request. Defaults to sha256.
kwargs: The subject, extension and version arguments from `x509.create_certificate` can be used.
CLI Example:

```
salt '*' x509.create_csr path=/etc/pki/myca.csr public_key='/etc/pki/myca.key' CN=
↪'My Cert'
```

`salt.modules.x509.`**`create_private_key`**(*path=None*, *text=False*, *bits=2048*, *passphrase=None*, *cipher='aes_128_cbc'*, *verbose=True*)
Creates a private key in PEM format.
path: The path to write the file to, either `path` or `text` are required.
text: If `True`, return the PEM text without writing to a file. Default `False`.
bits: Length of the private key in bits. Default 2048
passphrase: Passphrase for encrypting the private key
cipher: Cipher for encrypting the private key. Has no effect if passphrase is None.
verbose: Provide visual feedback on stdout. Default True

New in version 2016.11.0.

CLI Example:

```
salt '*' x509.create_private_key path=/etc/pki/mykey.key
```

`salt.modules.x509.`**`expired`**(*certificate*)
Returns a dict containing limited details of a certificate and whether the certificate has expired.

New in version 2016.11.0.

certificate: The certificate to be read. Can be a path to a certificate file, or a string containing the PEM formatted text of the certificate.
CLI Example:

```
salt '*' x509.expired "/etc/pki/mycert.crt"
```

`salt.modules.x509.`**`get_pem_entries`**(*glob_path*)
Returns a dict containing PEM entries in files matching a glob
glob_path: A path to certificates to be read and returned.
CLI Example:

```
salt '*' x509.get_pem_entries "/etc/pki/*.crt"
```

`salt.modules.x509.`**`get_pem_entry`**(*text*, *pem_type=None*)
Returns a properly formatted PEM string from the input text fixing any whitespace or line-break issues

text: Text containing the X509 PEM entry to be returned or path to a file containing the text.

pem_type: If specified, this function will only return a pem of a certain type, for example 'CERTIFICATE' or 'CERTIFICATE REQUEST'.

CLI Example:

```
salt '*' x509.get_pem_entry "-----BEGIN CERTIFICATE REQUEST-----MIICyzCC Ar8CAQI..
↪.-----END CERTIFICATE REQUEST"
```

`salt.modules.x509.`**`get_private_key_size`**(*private_key*, *passphrase=None*)

Returns the bit length of a private key in PEM format.

private_key: A path or PEM encoded string containing a private key.

CLI Example:

```
salt '*' x509.get_private_key_size /etc/pki/mycert.key
```

`salt.modules.x509.`**`get_public_key`**(*key*, *passphrase=None*, *asObj=False*)

Returns a string containing the public key in PEM format.

key: A path or PEM encoded string containing a CSR, Certificate or Private Key from which a public key can be retrieved.

CLI Example:

```
salt '*' x509.get_public_key /etc/pki/mycert.cer
```

`salt.modules.x509.`**`get_signing_policy`**(*signing_policy_name*)

Returns the details of a names signing policy, including the text of the public key that will be used to sign it. Does not return the private key.

CLI Example:

```
salt '*' x509.get_signing_policy www
```

`salt.modules.x509.`**`read_certificate`**(*certificate*)

Returns a dict containing details of a certificate. Input can be a PEM string or file path.

certificate: The certificate to be read. Can be a path to a certificate file, or a string containing the PEM formatted text of the certificate.

CLI Example:

```
salt '*' x509.read_certificate /etc/pki/mycert.crt
```

`salt.modules.x509.`**`read_certificates`**(*glob_path*)

Returns a dict containing details of all certificates matching a glob

glob_path: A path to certificates to be read and returned.

CLI Example:

```
salt '*' x509.read_certificates "/etc/pki/*.crt"
```

`salt.modules.x509.`**`read_crl`**(*crl*)

Returns a dict containing details of a certificate revocation list. Input can be a PEM string or file path.

Depends

- OpenSSL command line tool

crl: A path or PEM encoded string containing the CRL to read.

CLI Example:

```
salt '*' x509.read_crl /etc/pki/mycrl.crl
```

`salt.modules.x509.`**`read_csr`**`(`*csr*`)`

Returns a dict containing details of a certificate request.

Depends

- OpenSSL command line tool

csr: A path or PEM encoded string containing the CSR to read.

CLI Example:

```
salt '*' x509.read_csr /etc/pki/mycert.csr
```

`salt.modules.x509.`**`sign_remote_certificate`**`(`*argdic*, ***kwargs*`)`

Request a certificate to be remotely signed according to a signing policy.

argdic: A dict containing all the arguments to be passed into the create_certificate function. This will become kwargs when passed to create_certificate.

kwargs: kwargs delivered from publish.publish

CLI Example:

```
salt '*' x509.sign_remote_certificate argdic="{'public_key': '/etc/pki/www.key',
↪'signing_policy': 'www'}" __pub_id='www1'
```

`salt.modules.x509.`**`verify_crl`**`(`*crl*, *cert*`)`

Validate a CRL against a certificate. Parses openssl command line output, this is a workaround for M2Crypto's inability to get them from CSR objects.

crl: The CRL to verify

cert: The certificate to verify the CRL against

CLI Example:

```
salt '*' x509.verify_crl crl=/etc/pki/myca.crl cert=/etc/pki/myca.crt
```

`salt.modules.x509.`**`verify_private_key`**`(`*private_key*, *public_key*, *passphrase=None*`)`

Verify that 'private_key' matches 'public_key'

private_key: The private key to verify, can be a string or path to a private key in PEM format.

public_key: The public key to verify, can be a string or path to a PEM formatted certificate, CSR, or another private key.

passphrase: Passphrase to decrypt the private key.

CLI Example:

```
salt '*' x509.verify_private_key private_key=/etc/pki/myca.key \
        public_key=/etc/pki/myca.crt
```

`salt.modules.x509.`**`verify_signature`**`(`*certificate*, *signing_pub_key=None*, *signing_pub_key_passphrase=None*`)`

Verify that `certificate` has been signed by `signing_pub_key`

certificate: The certificate to verify. Can be a path or string containing a PEM formatted certificate.

signing_pub_key: The public key to verify, can be a string or path to a PEM formatted certificate, CSR, or private key.

signing_pub_key_passphrase: Passphrase to the signing_pub_key if it is an encrypted private key.

CLI Example:

```
salt '*' x509.verify_signature /etc/pki/mycert.pem \
        signing_pub_key=/etc/pki/myca.crt
```

`salt.modules.x509.`**`will_expire`**`(`*certificate*, *days*`)`

Returns a dict containing details of a certificate and whether the certificate will expire in the specified number of days. Input can be a PEM string or file path.

New in version 2016.11.0.

certificate: The certificate to be read. Can be a path to a certificate file, or a string containing the PEM formatted text of the certificate.

CLI Example:

```
salt '*' x509.will_expire "/etc/pki/mycert.crt" days=30
```

`salt.modules.x509.`**`write_pem`**`(`*text, path, overwrite=True, pem_type=None*`)`

Writes out a PEM string fixing any formatting or whitespace issues before writing.

text: PEM string input to be written out.

path: Path of the file to write the PEM out to.

overwrite: If `True` (default), write_pem will overwrite the entire PEM file. Set False to preserve existing private keys and dh params that may exist in the PEM file.

pem_type: The PEM type to be saved, for example `CERTIFICATE` or `PUBLIC KEY`. Adding this will allow the function to take input that may contain multiple PEM types.

CLI Example:

```
salt '*' x509.write_pem "-----BEGIN CERTIFICATE-----MIIGMzCCBBugA..." path=/etc/
↪pki/mycert.crt
```

21.9.489 salt.modules.xapi_virt

This module (mostly) uses the XenAPI to manage Xen virtual machines.

Big fat warning: the XenAPI used in this file is the one bundled with Xen Source, NOT XenServer nor Xen Cloud Platform. As a matter of fact it *will* fail under those platforms. From what I've read, little work is needed to adapt this code to XS/XCP, mostly playing with XenAPI version, but as XCP is not taking precedence on Xen Source on many platforms, please keep compatibility in mind.

Useful documentation:

. http://downloads.xen.org/Wiki/XenAPI/xenapi-1.0.6.pdf . http://docs.vmd.citrix.com/XenServer/6.0.0/1.0/en_gb/api/ . https://github.com/xapi-project/xen-api/tree/master/scripts/examples/python . http://xenbits.xen.org/gitweb/?p=xen.git;a=tree;f=tools/python/xen/xm;hb=HEAD

`salt.modules.xapi_virt.`**`freecpu`**`()`

Return an int representing the number of unallocated cpus on this hypervisor

CLI Example:

```
salt '*' virt.freecpu
```

`salt.modules.xapi_virt.`**`freemem`**`()`

Return an int representing the amount of memory that has not been given to virtual machines on this node

CLI Example:

```
salt '*' virt.freemem
```

`salt.modules.xapi_virt.`**`full_info`**`()`

Return the node_info, vm_info and freemem

CLI Example:

```
salt '*' virt.full_info
```

`salt.modules.xapi_virt.`**`get_disks`**`(`*vm_*`)`

Return the disks of a named vm

CLI Example:

```
salt '*' virt.get_disks <vm name>
```

`salt.modules.xapi_virt.`**`get_macs`**`(`*vm_*`)`

Return a list off MAC addresses from the named vm

CLI Example:

```
salt '*' virt.get_macs <vm name>
```

`salt.modules.xapi_virt.`**`get_nics`**`(`*vm_*`)`

Return info about the network interfaces of a named vm

CLI Example:

```
salt '*' virt.get_nics <vm name>
```

`salt.modules.xapi_virt.`**`is_hyper`**`()`

Returns a bool whether or not this node is a hypervisor of any kind

CLI Example:

```
salt '*' virt.is_hyper
```

`salt.modules.xapi_virt.`**`list_domains`**`()`

Return a list of virtual machine names on the minion

CLI Example:

```
salt '*' virt.list_domains
```

`salt.modules.xapi_virt.`**`migrate`**`(`*vm_, target, live=1, port=0, node=-1, ssl=None, change_home_server=0*`)`

Migrates the virtual machine to another hypervisor

CLI Example:

```
salt '*' virt.migrate <vm name> <target hypervisor> [live] [port] [node] [ssl]
↪[change_home_server]
```

Optional values:
live Use live migration
port Use a specified port
node Use specified NUMA node on target
ssl use ssl connection for migration
change_home_server change home server for managed domains

`salt.modules.xapi_virt.`**`node_info`**`()`

Return a dict with information about this node

CLI Example:

```
salt '*' virt.node_info
```

`salt.modules.xapi_virt.`**`pause`**`(`*vm_*`)`

Pause the named vm

CLI Example:

```
salt '*' virt.pause <vm name>
```

`salt.modules.xapi_virt.`**`reboot`**`(`*vm_*`)`

Reboot a domain via ACPI request

CLI Example:

```
salt '*' virt.reboot <vm name>
```

`salt.modules.xapi_virt.`**`reset`**`(`*vm_*`)`

Reset a VM by emulating the reset button on a physical machine

CLI Example:

```
salt '*' virt.reset <vm name>
```

`salt.modules.xapi_virt.`**`resume`**`(`*vm_*`)`

Resume the named vm

CLI Example:

```
salt '*' virt.resume <vm name>
```

`salt.modules.xapi_virt.`**`setmem`**`(`*vm_*, *memory*`)`

Changes the amount of memory allocated to VM.

Memory is to be specified in MB

CLI Example:

```
salt '*' virt.setmem myvm 768
```

`salt.modules.xapi_virt.`**`setvcpus`**`(`*vm_*, *vcpus*`)`

Changes the amount of vcpus allocated to VM.

vcpus is an int representing the number to be assigned

CLI Example:

```
salt '*' virt.setvcpus myvm 2
```

`salt.modules.xapi_virt.`**`shutdown`**`(`*vm_*`)`

Send a soft shutdown signal to the named vm

CLI Example:

```
salt '*' virt.shutdown <vm name>
```

`salt.modules.xapi_virt.`**`start`**`(`*config_*`)`

Start a defined domain

CLI Example:

```
salt '*' virt.start <path to Xen cfg file>
```

`salt.modules.xapi_virt.`**`stop`**`(`*vm_*`)`

Hard power down the virtual machine, this is equivalent to pulling the power

CLI Example:

```
salt '*' virt.stop <vm name>
```

`salt.modules.xapi_virt.`**`vcpu_pin`**`(`*vm_, vcpu, cpus*`)`

Set which CPUs a VCPU can use.

CLI Example:

```
salt 'foo' virt.vcpu_pin domU-id 2 1
salt 'foo' virt.vcpu_pin domU-id 2 2-6
```

`salt.modules.xapi_virt.`**`vm_cputime`**`(`*vm_=None*`)`

Return cputime used by the vms on this hyper in a list of dicts:

```
[
    'your-vm': {
        'cputime' <int>
        'cputime_percent' <int>
        },
    ...
    ]
```

If you pass a VM name in as an argument then it will return info for just the named VM, otherwise it will return all VMs.

CLI Example:

```
salt '*' virt.vm_cputime
```

`salt.modules.xapi_virt.`**`vm_diskstats`**`(`*vm_=None*`)`

Return disk usage counters used by the vms on this hyper in a list of dicts:

```
[
    'your-vm': {
        'io_read_kbs'   : 0,
        'io_write_kbs'  : 0
        },
    ...
    ]
```

If you pass a VM name in as an argument then it will return info for just the named VM, otherwise it will return all VMs.

CLI Example:

```
salt '*' virt.vm_diskstats
```

`salt.modules.xapi_virt.`**`vm_info`**`(`*vm_=None*`)`

Return detailed information about the vms.

If you pass a VM name in as an argument then it will return info for just the named VM, otherwise it will return all VMs.

CLI Example:

```
salt '*' virt.vm_info
```

`salt.modules.xapi_virt.`**`vm_netstats`**`(`*vm_=None*`)`

Return combined network counters used by the vms on this hyper in a list of dicts:

```
[
    'your-vm': {
        'io_read_kbs'           : 0,
        'io_total_read_kbs'     : 0,
        'io_total_write_kbs'    : 0,
        'io_write_kbs'          : 0
        },
    ...
    ]
```

If you pass a VM name in as an argument then it will return info for just the named VM, otherwise it will return all VMs.

CLI Example:

```
salt '*' virt.vm_netstats
```

`salt.modules.xapi_virt.`**`vm_state`**`(`*vm_=None*`)`
Return list of all the vms and their state.

If you pass a VM name in as an argument then it will return info for just the named VM, otherwise it will return all VMs.

CLI Example:

```
salt '*' virt.vm_state <vm name>
```

21.9.490 salt.modules.xbpspkg module

Package support for XBPS package manager (used by VoidLinux)

New in version 2016.11.0.

`salt.modules.xbpspkg.`**`add_repo`**`(`*repo, conffile='/usr/share/xbps.d/15-saltstack.conf'*`)`
Add an XBPS repository to the system.
repo url of repo to add (persistent).
conffile path to xbps conf file to add this repo default: /usr/share/xbps.d/15-saltstack.conf
CLI Examples:

```
salt '*' pkg.add_repo <repo url> [conffile=/path/to/xbps/repo.conf]
```

`salt.modules.xbpspkg.`**`available_version`**`(`**names, **kwargs*`)`
Return the latest version of the named package available for upgrade or installation. If more than one package name is specified, a dict of name/version pairs is returned.

If the latest version of a given package is already installed, an empty string will be returned for that package.

CLI Example:

```
salt '*' pkg.latest_version <package name>
salt '*' pkg.latest_version <package1> <package2> <package3> ...
```

`salt.modules.xbpspkg.`**`del_repo`**`(`*repo*`)`
Remove an XBPS repository from the system.
repo url of repo to remove (persistent).
CLI Examples:

```
salt '*' pkg.del_repo <repo url>
```

`salt.modules.xbpspkg.`**`get_repo`**`(`*repo*, ***kwargs*`)`

Display information about the repo.

CLI Examples:

```
salt '*' pkg.get_repo 'repo-url'
```

`salt.modules.xbpspkg.`**`install`**`(`*name=None*, *refresh=False*, *fromrepo=None*, *pkgs=None*, *sources=None*, ***kwargs*`)`

Install the passed package

name The name of the package to be installed.

refresh Whether or not to refresh the package database before installing.

fromrepo Specify a package repository (url) to install from.

Multiple Package Installation Options:

pkgs A list of packages to install from a software repository. Must be passed as a python list.

CLI Example:

```
salt '*' pkg.install pkgs='["foo","bar"]'
```

sources A list of packages to install. Must be passed as a list of dicts, with the keys being package names, and the values being the source URI or local path to the package.

CLI Example:

```
salt '*' pkg.install sources='[{"foo": "salt://foo.deb"},{"bar": "salt://bar.
↪deb"}]'
```

Return a dict containing the new package names and versions:

```
{'<package>': {'old': '<old-version>',
               'new': '<new-version>'}}
```

CLI Example:

```
salt '*' pkg.install <package name>
```

`salt.modules.xbpspkg.`**`latest_version`**`(`**names*, ***kwargs*`)`

Return the latest version of the named package available for upgrade or installation. If more than one package name is specified, a dict of name/version pairs is returned.

If the latest version of a given package is already installed, an empty string will be returned for that package.

CLI Example:

```
salt '*' pkg.latest_version <package name>
salt '*' pkg.latest_version <package1> <package2> <package3> ...
```

`salt.modules.xbpspkg.`**`list_pkgs`**`(`*versions_as_list=False*, ***kwargs*`)`

List the packages currently installed as a dict:

```
{'<package_name>': '<version>'}
```

CLI Example:

```
salt '*' pkg.list_pkgs
```

`salt.modules.xbpspkg.`**`list_repos`**`()`

List all repos known by XBPS

CLI Example:

```
salt '*' pkg.list_repos
```

`salt.modules.xbpspkg.`**`list_upgrades`**`(`*refresh=True*`)`

Check whether or not an upgrade is available for all packages

CLI Example:

```
salt '*' pkg.list_upgrades
```

`salt.modules.xbpspkg.`**`refresh_db`**`()`

Update list of available packages from installed repos

CLI Example:

```
salt '*' pkg.refresh_db
```

`salt.modules.xbpspkg.`**`remove`**`(`*name=None, pkgs=None, recursive=True, **kwargs*`)`

name The name of the package to be deleted.

recursive Also remove dependent packages (not required elsewhere). Default mode: enabled.

Multiple Package Options:

pkgs A list of packages to delete. Must be passed as a python list. The `name` parameter will be ignored if this option is passed.

Returns a list containing the removed packages.

CLI Example:

```
salt '*' pkg.remove <package name> [recursive=False]
salt '*' pkg.remove <package1>,<package2>,<package3> [recursive=False]
salt '*' pkg.remove pkgs='["foo", "bar"]' [recursive=False]
```

`salt.modules.xbpspkg.`**`upgrade`**`(`*refresh=True*`)`

Run a full system upgrade

refresh Whether or not to refresh the package database before installing. Default is *True*.

Returns a dictionary containing the changes:

```
{'<package>':  {'old': '<old-version>',
                'new': '<new-version>'}}
```

CLI Example:

```
salt '*' pkg.upgrade
```

`salt.modules.xbpspkg.`**`upgrade_available`**`(`*name*`)`

Check whether or not an upgrade is available for a given package

CLI Example:

```
salt '*' pkg.upgrade_available <package name>
```

`salt.modules.xbpspkg.`**`version`**`(`**names, **kwargs*`)`

Returns a string representing the package version or an empty string if not installed. If more than one package name is specified, a dict of name/version pairs is returned.

CLI Example:

```
salt '*' pkg.version <package name>
salt '*' pkg.version <package1> <package2> <package3> ...
```

21.9.491 salt.modules.xfs

Module for managing XFS file systems.

`salt.modules.xfs.`**`defragment`**`(`*device*`)`

Defragment mounted XFS filesystem. In order to mount a filesystem, device should be properly mounted and writable.

CLI Example:

```
salt '*' xfs.defragment /dev/sda1
```

`salt.modules.xfs.`**`devices`**`()`

Get known XFS formatted devices on the system.

CLI Example:

```
salt '*' xfs.devices
```

`salt.modules.xfs.`**`dump`**`(`*device, destination, level=0, label=None, noerase=None*`)`

Dump filesystem device to the media (file, tape etc).

Required parameters:

- **device**: XFS device, content of which to be dumped.
- **destination**: Specifies a dump destination.

Valid options are:

- **label**: Label of the dump. Otherwise automatically generated label is used.
- **level**: Specifies a dump level of 0 to 9.
- **noerase**: Pre-erase media.

Other options are not used in order to let `xfsdump` use its default values, as they are most optimal. See the `xfsdump(8)` manpage for a more complete description of these options.

CLI Example:

```
salt '*' xfs.dump /dev/sda1 /detination/on/the/client
salt '*' xfs.dump /dev/sda1 /detination/on/the/client label='Company accountancy'
salt '*' xfs.dump /dev/sda1 /detination/on/the/client noerase=True
```

`salt.modules.xfs.`**`estimate`**`(`*path*`)`

Estimate the space that an XFS filesystem will take. For each directory estimate the space that directory would take if it were copied to an XFS filesystem. Estimation does not cross mount points.

CLI Example:

```
salt '*' xfs.estimate /path/to/file
salt '*' xfs.estimate /path/to/dir/*
```

`salt.modules.xfs.`**`info`**`(`*device*`)`

Get filesystem geometry information.

CLI Example:

```
salt '*' xfs.info /dev/sda1
```

`salt.modules.xfs.`**`inventory`**`()`

Display XFS dump inventory without restoration.

CLI Example:

```
salt '*' xfs.inventory
```

`salt.modules.xfs.`**`mkfs`**`(`*device, label=None, ssize=None, noforce=None, bso=None, gmo=None, ino=None, lso=None, rso=None, nmo=None, dso=None*`)`

Create a file system on the specified device. By default wipes out with force.

General options:

- **label**: Specify volume label.
- **ssize**: Specify the fundamental sector size of the filesystem.
- **noforce**: Do not force create filesystem, if disk is already formatted.

Filesystem geometry options:

- **bso**: Block size options.
- **gmo**: Global metadata options.
- **dso**: **Data section options. These options specify the location, size,** and other parameters of the data section of the filesystem.
- **ino**: Inode options to specify the inode size of the filesystem, and other inode allocation parameters.
- **lso**: Log section options.
- **nmo**: Naming options.
- **rso**: Realtime section options.

See the `mkfs.xfs(8)` manpage for a more complete description of corresponding options description.

CLI Example:

```
salt '*' xfs.mkfs /dev/sda1
salt '*' xfs.mkfs /dev/sda1 dso='su=32k,sw=6' noforce=True
salt '*' xfs.mkfs /dev/sda1 dso='su=32k,sw=6' lso='logdev=/dev/sda2,size=10000b'
```

`salt.modules.xfs.`**`modify`**`(`*device, label=None, lazy_counting=None, uuid=None*`)`

Modify parameters of an XFS filesystem.

CLI Example:

```
salt '*' xfs.modify /dev/sda1 label='My backup' lazy_counting=False
salt '*' xfs.modify /dev/sda1 uuid=False
salt '*' xfs.modify /dev/sda1 uuid=True
```

`salt.modules.xfs.`**`prune_dump`**`(`*sessionid*`)`

Prunes the dump session identified by the given session id.

CLI Example:

```
salt '*' xfs.prune_dump b74a3586-e52e-4a4a-8775-c3334fa8ea2c
```

21.9.492 salt.modules.xml

XML file manager

New in version 3000.

`salt.modules.xml.`**`get_attribute`**`(`*file, element*`)`

Return the attributes of the matched xpath element.

CLI Example:

```
salt '*' xml.get_attribute /tmp/test.xml ".//element[@id='3']"
```

`salt.modules.xml.`**`get_value`**(*file*, *element*)
Returns the value of the matched xpath element

CLI Example:

```
salt '*' xml.get_value /tmp/test.xml ".//element"
```

`salt.modules.xml.`**`set_attribute`**(*file*, *element*, *key*, *value*)
Set the requested attribute key and value for matched xpath element.

CLI Example:

```
salt '*' xml.set_attribute /tmp/test.xml ".//element[@id='3']" editedby "gal"
```

`salt.modules.xml.`**`set_value`**(*file*, *element*, *value*)
Sets the value of the matched xpath element

CLI Example:

```
salt '*' xml.set_value /tmp/test.xml ".//element" "new value"
```

21.9.493 salt.modules.xmpp

Module for Sending Messages via XMPP (a.k.a. Jabber)

New in version 2014.1.0.

depends

- sleekxmpp>=1.3.1
- pyasn1
- pyasn1-modules
- dnspython

configuration This module can be used by either passing a jid and password directly to send_message, or by specifying the name of a configuration profile in the minion config, minion pillar, or master config.

For example:

```
my-xmpp-login:
    xmpp.jid: myuser@jabber.example.org/resourcename
    xmpp.password: verybadpass
```

The resourcename refers to the resource that is using this account. It is user-definable, and optional. The following configurations are both valid:

```
my-xmpp-login:
    xmpp.jid: myuser@jabber.example.org/salt
    xmpp.password: verybadpass

my-xmpp-login:
    xmpp.jid: myuser@jabber.example.org
    xmpp.password: verybadpass
```

class `salt.modules.xmpp.`**`SendMsgBot`**(*jid, password, recipient, msg*)

classmethod create_multi(*jid, password, msg, recipients=None, rooms=None, nick='SaltStack Bot'*)

Alternate constructor that accept multiple recipients and rooms

`start`(*event*)

class `salt.modules.xmpp.`**`SleekXMPPMUC`**(*name=''*)

`filter`(*record*)

Determine if the specified record is to be logged.

Is the specified record to be logged? Returns 0 for no, nonzero for yes. If deemed appropriate, the record may be modified in-place.

`salt.modules.xmpp.`**`send_msg`**(*recipient, message, jid=None, password=None, profile=None*)

Send a message to an XMPP recipient. Designed for use in states.

CLI Examples:

```
xmpp.send_msg 'admins@xmpp.example.com' 'This is a salt module test'            ⊠
↪profile='my-xmpp-account'
xmpp.send_msg 'admins@xmpp.example.com' 'This is a salt module test'            ⊠
↪jid='myuser@xmpp.example.com/salt' password='verybadpass'
```

`salt.modules.xmpp.`**`send_msg_multi`**(*message, recipients=None, rooms=None, jid=None, password=None, nick='SaltStack Bot', profile=None*)

Send a message to an XMPP recipient, support send message to multiple recipients or chat room.

CLI Examples:

```
xmpp.send_msg recipients=['admins@xmpp.example.com']             rooms=[
↪'secret@conference.xmpp.example.com']             'This is a salt module test' ⊠
↪           profile='my-xmpp-account'
xmpp.send_msg recipients=['admins@xmpp.example.com']             rooms=[
↪'secret@conference.xmpp.example.com']            'This is a salt module test'  ⊠
↪          jid='myuser@xmpp.example.com/salt' password='verybadpass'
```

21.9.494 salt.modules.yumpkg

Support for YUM/DNF

Important: If you feel that Salt should be using this module to manage packages on a minion, and it is using a different module (or gives an error similar to *'pkg.install' is not available*), see *here*.

Note: DNF is fully supported as of version 2015.5.10 and 2015.8.4 (partial support for DNF was initially added in 2015.8.0), and DNF is used automatically in place of YUM in Fedora 22 and newer.

---

class `salt.modules.yumpkg.`**`AvailablePackages`**(**args, **kwargs*)

`salt.modules.yumpkg.`**`clean_metadata`**(***kwargs*)

New in version 2014.1.0.

Cleans local yum metadata. Functionally identical to *`refresh_db()`*.

CLI Example:

```
salt '*' pkg.clean_metadata
```

`salt.modules.yumpkg.`**`del_repo`**`(`*repo, basedir=None, **kwargs*`)`

Delete a repo from <basedir> (default basedir: all dirs in *reposdir* yum option).

If the .repo file in which the repo exists does not contain any other repo configuration, the file itself will be deleted.

CLI Examples:

```
salt '*' pkg.del_repo myrepo
salt '*' pkg.del_repo myrepo basedir=/path/to/dir
salt '*' pkg.del_repo myrepo basedir=/path/to/dir,/path/to/another/dir
```

`salt.modules.yumpkg.`**`diff`**`(`**paths*`)`

Return a formatted diff between current files and original in a package. NOTE: this function includes all files (configuration and not), but does not work on binary content.

Parameters **`path`** – Full path to the installed file

Returns Difference string or raises and exception if examined file is binary.

CLI example:

```
salt '*' pkg.diff /etc/apache2/httpd.conf /etc/sudoers
```

`salt.modules.yumpkg.`**`download`**`(`**packages*`)`

New in version 2015.5.0.

Download packages to the local disk. Requires `yumdownloader` from `yum-utils` package.

Note: `yum-utils` will already be installed on the minion if the package was installed from the Fedora / EPEL repositories.

CLI example:

```
salt '*' pkg.download httpd
salt '*' pkg.download httpd postfix
```

`salt.modules.yumpkg.`**`file_dict`**`(`**packages*`)`

New in version 2014.1.0.

List the files that belong to a package, grouped by package. Not specifying any packages will return a list of *every* file on the system's rpm database (not generally recommended).

CLI Examples:

```
salt '*' pkg.file_list httpd
salt '*' pkg.file_list httpd postfix
salt '*' pkg.file_list
```

`salt.modules.yumpkg.`**`file_list`**`(`**packages*`)`

New in version 2014.1.0.

List the files that belong to a package. Not specifying any packages will return a list of *every* file on the system's rpm database (not generally recommended).

CLI Examples:

```
salt '*' pkg.file_list httpd
salt '*' pkg.file_list httpd postfix
salt '*' pkg.file_list
```

`salt.modules.yumpkg.`**`get_repo`**`(`*name*, *basedir=None*, ***kwargs*`)`

Display a repo from <basedir> (default basedir: all dirs in `reposdir` yum option).

CLI Examples:

```
salt '*' pkg.get_repo myrepo
salt '*' pkg.get_repo myrepo basedir=/path/to/dir
salt '*' pkg.get_repo myrepo basedir=/path/to/dir,/path/to/another/dir
```

`salt.modules.yumpkg.`**`group_diff`**`(`*name*`)`

New in version 2014.1.0.

Changed in version 2016.3.0,2015.8.4,2015.5.10: Environment groups are now supported. The key names have been renamed, similar to the changes made in *`pkg.group_info`*.

Lists which of a group's packages are installed and which are not installed

CLI Example:

```
salt '*' pkg.group_diff 'Perl Support'
```

`salt.modules.yumpkg.`**`group_info`**`(`*name*, *expand=False*`)`

New in version 2014.1.0.

Changed in version 2016.3.0,2015.8.4,2015.5.10: The return data has changed. A new key `type` has been added to distinguish environment groups from package groups. Also, keys for the group name and group ID have been added. The `mandatory packages`, `optional packages`, and `default packages` keys have been renamed to `mandatory`, `optional`, and `default` for accuracy, as environment groups include other groups, and not packages. Finally, this function now properly identifies conditional packages.

Lists packages belonging to a certain group

name Name of the group to query

expand [False] If the specified group is an environment group, then the group will be expanded and the return data will include package names instead of group names.

New in version 2016.3.0.

CLI Example:

```
salt '*' pkg.group_info 'Perl Support'
```

`salt.modules.yumpkg.`**`group_install`**`(`*name*, *skip=()*, *include=()*, ***kwargs*`)`

New in version 2014.1.0.

Install the passed package group(s). This is basically a wrapper around *`pkg.install`*, which performs package group resolution for the user. This function is currently considered experimental, and should be expected to undergo changes.

name Package group to install. To install more than one group, either use a comma-separated list or pass the value as a python list.

CLI Examples:

```
salt '*' pkg.group_install 'Group 1'
salt '*' pkg.group_install 'Group 1,Group 2'
salt '*' pkg.group_install '["Group 1", "Group 2"]'
```

skip Packages that would normally be installed by the package group ("default" packages), which should not be installed. Can be passed either as a comma-separated list or a python list.

CLI Examples:

```
salt '*' pkg.group_install 'My Group' skip='foo,bar'
salt '*' pkg.group_install 'My Group' skip='["foo", "bar"]'
```

include Packages which are included in a group, which would not normally be installed by a `yum groupinstall` ("optional" packages). Note that this will not enforce group membership; if you include packages which are not members of the specified groups, they will still be installed. Can be passed either as a comma-separated list or a python list.

CLI Examples:

```
salt '*' pkg.group_install 'My Group' include='foo,bar'
salt '*' pkg.group_install 'My Group' include='["foo", "bar"]'
```

Note: Because this is essentially a wrapper around pkg.install, any argument which can be passed to pkg.install may also be included here, and it will be passed along wholesale.

`salt.modules.yumpkg.`**`group_list`**`()`

New in version 2014.1.0.

Lists all groups known by yum on this system

CLI Example:

```
salt '*' pkg.group_list
```

`salt.modules.yumpkg.`**`hold`**`(`*name=None, pkgs=None, sources=None, normalize=True, **kwargs*`)`

New in version 2014.7.0.

Version-lock packages

Note: Requires the appropriate `versionlock` plugin package to be installed:
- On RHEL 5: `yum-versionlock`
- On RHEL 6 & 7: `yum-plugin-versionlock`
- On Fedora: `python-dnf-plugins-extras-versionlock`

name The name of the package to be held.

Multiple Package Options:

pkgs A list of packages to hold. Must be passed as a python list. The `name` parameter will be ignored if this option is passed.

Returns a dict containing the changes.

CLI Example:

```
salt '*' pkg.hold <package name>
salt '*' pkg.hold pkgs='["foo", "bar"]'
```

`salt.modules.yumpkg.`**`info_installed`**`(`**names, **kwargs*`)`

New in version 2015.8.1.

Return the information of the named package(s), installed on the system.

Parameters **`all_versions`** – Include information for all versions of the packages installed on the minion.

CLI example:

```
salt '*' pkg.info_installed <package1>
salt '*' pkg.info_installed <package1> <package2> <package3> ...
salt '*' pkg.info_installed <package1> <package2> <package3> all_versions=True
```

`salt.modules.yumpkg.`**`install`**(*name=None, refresh=False, skip_verify=False, pkgs=None, sources=None, downloadonly=False, reinstall=False, normalize=True, update_holds=False, saltenv='base', ignore_epoch=False, **kwargs*)

Changed in version 2015.8.12,2016.3.3,2016.11.0: On minions running systemd>=205, systemd-run(1) is now used to isolate commands which modify installed packages from the `salt-minion` daemon's control group. This is done to keep systemd from killing any yum/dnf commands spawned by Salt when the `salt-minion` service is restarted. (see `KillMode` in the systemd.kill(5) manpage for more information). If desired, usage of systemd-run(1) can be suppressed by setting a *`config option`* called `systemd.scope`, with a value of `False` (no quotes).

Install the passed package(s), add refresh=True to clean the yum database before package is installed.

name The name of the package to be installed. Note that this parameter is ignored if either "pkgs" or "sources" is passed. Additionally, please note that this option can only be used to install packages from a software repository. To install a package file manually, use the "sources" option.

32-bit packages can be installed on 64-bit systems by appending the architecture designation (`.i686`, `.i586`, etc.) to the end of the package name.

CLI Example:

```
salt '*' pkg.install <package name>
```

refresh Whether or not to update the yum database before executing.

reinstall Specifying reinstall=True will use `yum reinstall` rather than `yum install` for requested packages that are already installed.

If a version is specified with the requested package, then `yum reinstall` will only be used if the installed version matches the requested version.

Works with `sources` when the package header of the source can be matched to the name and version of an installed package.

New in version 2014.7.0.

skip_verify Skip the GPG verification check (e.g., `--nogpgcheck`)

downloadonly Only download the packages, do not install.

version Install a specific version of the package, e.g. 1.2.3-4.el5. Ignored if "pkgs" or "sources" is passed.

Changed in version 2018.3.0: version can now contain comparison operators (e.g. `>1.2.3`, `<=2.0`, etc.)

update_holds [False] If `True`, and this function would update the package version, any packages held using the yum/dnf "versionlock" plugin will be unheld so that they can be updated. Otherwise, if this function attempts to update a held package, the held package(s) will be skipped and an error will be raised.

New in version 2016.11.0.

setopt A comma-separated or Python list of key=value options. This list will be expanded and `--setopt` prepended to each in the yum/dnf command that is run.

CLI Example:

```
salt '*' pkg.install foo setopt='obsoletes=0,plugins=0'
```

New in version 2019.2.0.

Repository Options:

fromrepo Specify a package repository (or repositories) from which to install. (e.g., `yum --disablerepo='*' --enablerepo='somerepo'`)

enablerepo (ignored if `fromrepo` is specified) Specify a disabled package repository (or repositories) to enable. (e.g., `yum --enablerepo='somerepo'`)

disablerepo (ignored if `fromrepo` is specified) Specify an enabled package repository (or repositories) to disable. (e.g., `yum --disablerepo='somerepo'`)

disableexcludes Disable exclude from main, for a repo or for everything. (e.g., `yum --disableexcludes='main'`)

New in version 2014.7.0.

ignore_epoch [False] Only used when the version of a package is specified using a comparison operator (e.g. `>4.1`). If set to `True`, then the epoch will be ignored when comparing the currently-installed version to the desired version.

New in version 2018.3.0.

Multiple Package Installation Options:

pkgs A list of packages to install from a software repository. Must be passed as a python list. A specific version number can be specified by using a single-element dict representing the package and its version.

CLI Examples:

```
salt '*' pkg.install pkgs='["foo", "bar"]'
salt '*' pkg.install pkgs='["foo", {"bar": "1.2.3-4.el5"}]'
```

sources A list of RPM packages to install. Must be passed as a list of dicts, with the keys being package names, and the values being the source URI or local path to the package.

CLI Example:

```
salt '*' pkg.install sources='[{"foo": "salt://foo.rpm"}, {"bar": "salt://
↪bar.rpm"}]'
```

normalize [True] Normalize the package name by removing the architecture. This is useful for poorly created packages which might include the architecture as an actual part of the name such as kernel modules which match a specific kernel version.

```
salt -G role:nsd pkg.install gpfs.gplbin-2.6.32-279.31.1.el6.x86_64␣
↪normalize=False
```

New in version 2014.7.0.

diff_attr: If a list of package attributes is specified, returned value will contain them, eg.:

```
{'<package>': {
    'old': {
        'version': '<old-version>',
        'arch': '<old-arch>'},

    'new': {
        'version': '<new-version>',
        'arch': '<new-arch>'}}}
```

Valid attributes are: `epoch`, `version`, `release`, `arch`, `install_date`, `install_date_time_t`.

If `all` is specified, all valid attributes will be returned.

New in version 2018.3.0.

Returns a dict containing the new package names and versions:

```
{'<package>': {'old': '<old-version>',
               'new': '<new-version>'}}
```

If an attribute list in diff_attr is specified, the dict will also contain any specified attribute, eg.:

```
{'<package>': {
    'old': {
        'version': '<old-version>',
        'arch': '<old-arch>'},

    'new': {
        'version': '<new-version>',
        'arch': '<new-arch>'}}}
```

`salt.modules.yumpkg.`**`latest_version`**`(`**names, **kwargs*`)`

Return the latest version of the named package available for upgrade or installation. If more than one package name is specified, a dict of name/version pairs is returned.

If the latest version of a given package is already installed, an empty string will be returned for that package.

A specific repo can be requested using the `fromrepo` keyword argument, and the `disableexcludes` option is also supported.

New in version 2014.7.0: Support for the `disableexcludes` option

CLI Example:

```
salt '*' pkg.latest_version <package name>
salt '*' pkg.latest_version <package name> fromrepo=epel-testing
salt '*' pkg.latest_version <package name> disableexcludes=main
salt '*' pkg.latest_version <package1> <package2> <package3> ...
```

`salt.modules.yumpkg.`**`list_downloaded`**`()`

New in version 2017.7.0.

List prefetched packages downloaded by Yum in the local disk.

CLI example:

```
salt '*' pkg.list_downloaded
```

`salt.modules.yumpkg.`**`list_holds`**`(`*pattern='[\\w+]+(?:[.-][^-]+)*', full=True*`)`

Changed in version 2016.3.0,2015.8.4,2015.5.10: Function renamed from `pkg.get_locked_pkgs` to `pkg.list_holds`.

List information on locked packages

Note: Requires the appropriate `versionlock` plugin package to be installed:

- On RHEL 5: `yum-versionlock`
- On RHEL 6 & 7: `yum-plugin-versionlock`
- On Fedora: `python-dnf-plugins-extras-versionlock`

---

pattern [\w+(?:[.-][^-]+)*] Regular expression used to match the package name

full [True] Show the full hold definition including version and epoch. Set to `False` to return just the name of the package(s) being held.

CLI Example:

```
salt '*' pkg.list_holds
salt '*' pkg.list_holds full=False
```

`salt.modules.yumpkg.`**`list_installed_patches`**`()`

New in version 2017.7.0.

List installed advisory patches on the system.

CLI Examples:

```
salt '*' pkg.list_installed_patches
```

`salt.modules.yumpkg.`**`list_patches`**`(`*refresh=False*`)`

New in version 2017.7.0.

List all known advisory patches from available repos.

refresh force a refresh if set to True. If set to False (default) it depends on yum if a refresh is executed.

CLI Examples:

```
salt '*' pkg.list_patches
```

`salt.modules.yumpkg.`**`list_pkgs`**`(`*versions_as_list=False*, ***kwargs*`)`

List the packages currently installed as a dict. By default, the dict contains versions as a comma separated string:

```
{'<package_name>': '<version>[,<version>...]'}
```

versions_as_list: If set to true, the versions are provided as a list

{'<package_name>': ['<version>', '<version>']}

attr: If a list of package attributes is specified, returned value will contain them in addition to version, eg.:

```
{'<package_name>': [{'version' : 'version', 'arch' : 'arch'}]}
```

Valid attributes are: `epoch`, `version`, `release`, `arch`, `install_date`, `install_date_time_t`.

If `all` is specified, all valid attributes will be returned.

New in version 2018.3.0.

CLI Example:

```
salt '*' pkg.list_pkgs
salt '*' pkg.list_pkgs attr=version,arch
salt '*' pkg.list_pkgs attr='["version", "arch"]'
```

`salt.modules.yumpkg.`**`list_repo_pkgs`**`(`**args*, ***kwargs*`)`

New in version 2014.1.0.

Changed in version 2014.7.0: All available versions of each package are now returned. This required a slight modification to the structure of the return dict. The return data shown below reflects the updated return dict structure. Note that packages which are version-locked using *`pkg.hold`* will only show the currently-installed version, as locking a package will make other versions appear unavailable to yum/dnf.

Changed in version 2017.7.0: By default, the versions for each package are no longer organized by repository. To get results organized by repository, use `byrepo=True`.

Returns all available packages. Optionally, package names (and name globs) can be passed and the results will be filtered to packages matching those names. This is recommended as it speeds up the function considerably.

> **Warning:** Running this function on RHEL/CentOS 6 and earlier will be more resource-intensive, as the version of yum that ships with older RHEL/CentOS has no yum subcommand for listing packages from a repository. Thus, a `yum list installed` and `yum list available` are run, which generates a lot of output, which must then be analyzed to determine which package information to include in the return data.

This function can be helpful in discovering the version or repo to specify in a *`pkg.installed`* state.

The return data will be a dictionary mapping package names to a list of version numbers, ordered from newest to oldest. If `byrepo` is set to `True`, then the return dictionary will contain repository names at the top level, and each repository will map packages to lists of version numbers. For example:

```
# With byrepo=False (default)
{
    'bash': ['4.1.2-15.el6_5.2',
             '4.1.2-15.el6_5.1',
             '4.1.2-15.el6_4'],
    'kernel': ['2.6.32-431.29.2.el6',
               '2.6.32-431.23.3.el6',
               '2.6.32-431.20.5.el6',
               '2.6.32-431.20.3.el6',
               '2.6.32-431.17.1.el6',
               '2.6.32-431.11.2.el6',
               '2.6.32-431.5.1.el6',
               '2.6.32-431.3.1.el6',
               '2.6.32-431.1.2.0.1.el6',
               '2.6.32-431.el6']
}
# With byrepo=True
{
    'base': {
        'bash': ['4.1.2-15.el6_4'],
        'kernel': ['2.6.32-431.el6']
    },
    'updates': {
        'bash': ['4.1.2-15.el6_5.2', '4.1.2-15.el6_5.1'],
        'kernel': ['2.6.32-431.29.2.el6',
                   '2.6.32-431.23.3.el6',
                   '2.6.32-431.20.5.el6',
                   '2.6.32-431.20.3.el6',
                   '2.6.32-431.17.1.el6',
                   '2.6.32-431.11.2.el6',
                   '2.6.32-431.5.1.el6',
                   '2.6.32-431.3.1.el6',
                   '2.6.32-431.1.2.0.1.el6']
    }
}
```

fromrepo [None] Only include results from the specified repo(s). Multiple repos can be specified, comma-separated.

enablerepo (ignored if `fromrepo` is specified) Specify a disabled package repository (or repositories) to enable. (e.g., `yum --enablerepo='somerepo'`)

New in version 2017.7.0.

disablerepo (ignored if `fromrepo` is specified) Specify an enabled package repository (or repositories) to disable. (e.g., `yum --disablerepo='somerepo'`)

New in version 2017.7.0.

byrepo [False] When `True`, the return data for each package will be organized by repository.

New in version 2017.7.0.

cacheonly [False] When `True`, the repo information will be retrieved from the cached repo metadata. This is equivalent to passing the `-C` option to yum/dnf.

New in version 2017.7.0.

setopt A comma-separated or Python list of key=value options. This list will be expanded and `--setopt` prepended to each in the yum/dnf command that is run.

New in version 2019.2.0.

CLI Examples:

```
salt '*' pkg.list_repo_pkgs
salt '*' pkg.list_repo_pkgs foo bar baz
salt '*' pkg.list_repo_pkgs 'samba4*' fromrepo=base,updates
salt '*' pkg.list_repo_pkgs 'python2-*' byrepo=True
```

`salt.modules.yumpkg.`**`list_repos`**`(`*basedir=None*`)`

Lists all repos in <basedir> (default: all dirs in *reposdir* yum option).

CLI Example:

```
salt '*' pkg.list_repos
salt '*' pkg.list_repos basedir=/path/to/dir
salt '*' pkg.list_repos basedir=/path/to/dir,/path/to/another/dir
```

`salt.modules.yumpkg.`**`list_upgrades`**`(`*refresh=True, **kwargs*`)`

Check whether or not an upgrade is available for all packages

The `fromrepo`, `enablerepo`, and `disablerepo` arguments are supported, as used in pkg states, and the `disableexcludes` option is also supported.

New in version 2014.7.0: Support for the `disableexcludes` option

CLI Example:

```
salt '*' pkg.list_upgrades
```

`salt.modules.yumpkg.`**`mod_repo`**`(`*repo, basedir=None, **kwargs*`)`

Modify one or more values for a repo. If the repo does not exist, it will be created, so long as the following values are specified:

repo name by which the yum refers to the repo

name a human-readable name for the repo

baseurl the URL for yum to reference

mirrorlist the URL for yum to reference

Key/Value pairs may also be removed from a repo's configuration by setting a key to a blank value. Bear in mind that a name cannot be deleted, and a baseurl can only be deleted if a mirrorlist is specified (or vice versa).

CLI Examples:

```
salt '*' pkg.mod_repo reponame enabled=1 gpgcheck=1
salt '*' pkg.mod_repo reponame basedir=/path/to/dir enabled=1
salt '*' pkg.mod_repo reponame baseurl= mirrorlist=http://host.com/
```

`salt.modules.yumpkg.`**`modified`**`(`**packages, **flags*`)`

List the modified files that belong to a package. Not specifying any packages will return a list of _all_ modified files on the system's RPM database.

New in version 2015.5.0.

Filtering by flags (True or False):
size Include only files where size changed.
mode Include only files which file's mode has been changed.
checksum Include only files which MD5 checksum has been changed.
device Include only files which major and minor numbers has been changed.
symlink Include only files which are symbolic link contents.
owner Include only files where owner has been changed.
group Include only files where group has been changed.
time Include only files where modification time of the file has been changed.
capabilities Include only files where capabilities differ or not. Note: supported only on newer RPM versions.
CLI Examples:

```
salt '*' pkg.modified
salt '*' pkg.modified httpd
salt '*' pkg.modified httpd postfix
salt '*' pkg.modified httpd owner=True group=False
```

`salt.modules.yumpkg.`**`normalize_name`**(*name*)

Strips the architecture from the specified package name, if necessary. Circumstances where this would be done include:

- If the arch is 32 bit and the package name ends in a 32-bit arch.
- If the arch matches the OS arch, or is `noarch`.

CLI Example:

```
salt '*' pkg.normalize_name zsh.x86_64
```

`salt.modules.yumpkg.`**`owner`**(**paths*)

New in version 2014.7.0.

Return the name of the package that owns the file. Multiple file paths can be passed. Like *`pkg.version`*, if a single path is passed, a string will be returned, and if multiple paths are passed, a dictionary of file/package name pairs will be returned.

If the file is not owned by a package, or is not present on the minion, then an empty string will be returned for that path.

CLI Examples:

```
salt '*' pkg.owner /usr/bin/apachectl
salt '*' pkg.owner /usr/bin/apachectl /etc/httpd/conf/httpd.conf
```

`salt.modules.yumpkg.`**`parse_arch`**(*name*)

Parse name and architecture from the specified package name.

CLI Example:

```
salt '*' pkg.parse_arch zsh.x86_64
```

`salt.modules.yumpkg.`**`purge`**(*name=None*, *pkgs=None*, ***kwargs*)

Changed in version 2015.8.12,2016.3.3,2016.11.0: On minions running systemd>=205, systemd-run(1) is now used to isolate commands which modify installed packages from the `salt-minion` daemon's control group. This is done to keep systemd from killing any yum/dnf commands spawned by Salt when the `salt-minion` service is restarted. (see `KillMode` in the systemd.kill(5) manpage for more information). If desired, usage of systemd-run(1) can be suppressed by setting a *`config option`* called `systemd.scope`, with a value of `False` (no quotes).

Package purges are not supported by yum, this function is identical to *`pkg.remove`*.

name The name of the package to be purged
Multiple Package Options:
pkgs A list of packages to delete. Must be passed as a python list. The `name` parameter will be ignored if this option is passed.
New in version 0.16.0.

Returns a dict containing the changes.

CLI Example:

```
salt '*' pkg.purge <package name>
salt '*' pkg.purge <package1>,<package2>,<package3>
salt '*' pkg.purge pkgs='["foo", "bar"]'
```

`salt.modules.yumpkg.`**`refresh_db`**`(`*`**kwargs`*`)`

Check the yum repos for updated packages

Returns:

- `True`: Updates are available
- `False`: An error occurred
- `None`: No updates are available

repo Refresh just the specified repo
disablerepo Do not refresh the specified repo
enablerepo Refresh a disabled repo using this option
branch Add the specified branch when refreshing
disableexcludes Disable the excludes defined in your config files. Takes one of three options: - `all` - disable all excludes - `main` - disable excludes defined in [main] in yum.conf - `repoid` - disable excludes defined for that repo
setopt A comma-separated or Python list of key=value options. This list will be expanded and `--setopt` prepended to each in the yum/dnf command that is run.

New in version 2019.2.0.

CLI Example:

```
salt '*' pkg.refresh_db
```

`salt.modules.yumpkg.`**`remove`**`(`*`name=None, pkgs=None, **kwargs`*`)`

Changed in version 2015.8.12,2016.3.3,2016.11.0: On minions running systemd>=205, systemd-run(1) is now used to isolate commands which modify installed packages from the `salt-minion` daemon's control group. This is done to keep systemd from killing any yum/dnf commands spawned by Salt when the `salt-minion` service is restarted. (see `KillMode` in the systemd.kill(5) manpage for more information). If desired, usage of systemd-run(1) can be suppressed by setting a *`config option`* called `systemd.scope`, with a value of `False` (no quotes).

Remove packages
name The name of the package to be removed
Multiple Package Options:
pkgs A list of packages to delete. Must be passed as a python list. The `name` parameter will be ignored if this option is passed.
New in version 0.16.0.

Returns a dict containing the changes.

CLI Example:

```
salt '*' pkg.remove <package name>
salt '*' pkg.remove <package1>,<package2>,<package3>
salt '*' pkg.remove pkgs='["foo", "bar"]'
```

`salt.modules.yumpkg.`**`unhold`**(*name=None*, *pkgs=None*, *sources=None*, ***kwargs*)

New in version 2014.7.0.

Remove version locks

Note: Requires the appropriate `versionlock` plugin package to be installed:

- On RHEL 5: `yum-versionlock`
- On RHEL 6 & 7: `yum-plugin-versionlock`
- On Fedora: `python-dnf-plugins-extras-versionlock`

---

name The name of the package to be unheld

Multiple Package Options:

pkgs A list of packages to unhold. Must be passed as a python list. The `name` parameter will be ignored if this option is passed.

Returns a dict containing the changes.

CLI Example:

```
salt '*' pkg.unhold <package name>
salt '*' pkg.unhold pkgs='["foo", "bar"]'
```

`salt.modules.yumpkg.`**`update`**(*name=None*, *pkgs=None*, *refresh=True*, *skip_verify=False*, *normalize=True*, *minimal=False*, *obsoletes=False*, ***kwargs*)

New in version 2019.2.0.

Calls *`pkg.upgrade`* with `obsoletes=False`. Mirrors the CLI behavior of `yum update`. See *`pkg.upgrade`* for further documentation.

```
salt '*' pkg.update
```

`salt.modules.yumpkg.`**`upgrade`**(*name=None*, *pkgs=None*, *refresh=True*, *skip_verify=False*, *normalize=True*, *minimal=False*, *obsoletes=True*, ***kwargs*)

Run a full system upgrade (a `yum upgrade` or `dnf upgrade`), or upgrade specified packages. If the packages aren't installed, they will not be installed.

Changed in version 2014.7.0.

Changed in version 2015.8.12,2016.3.3,2016.11.0: On minions running systemd>=205, systemd-run(1) is now used to isolate commands which modify installed packages from the `salt-minion` daemon's control group. This is done to keep systemd from killing any yum/dnf commands spawned by Salt when the `salt-minion` service is restarted. (see `KillMode` in the systemd.kill(5) manpage for more information). If desired, usage of systemd-run(1) can be suppressed by setting a *`config option`* called `systemd.scope`, with a value of `False` (no quotes).

Changed in version 2019.2.0: Added `obsoletes` and `minimal` arguments

Returns a dictionary containing the changes:

```
{'<package>':  {'old': '<old-version>',
                'new': '<new-version>'}}
```

CLI Example:

```
salt '*' pkg.upgrade
salt '*' pkg.upgrade name=openssl
```

Repository Options:

fromrepo Specify a package repository (or repositories) from which to install. (e.g., `yum --disablerepo='*' --enablerepo='somerepo'`)

enablerepo (ignored if `fromrepo` is specified) Specify a disabled package repository (or repositories) to enable. (e.g., `yum --enablerepo='somerepo'`)

disablerepo (ignored if `fromrepo` is specified) Specify an enabled package repository (or repositories) to disable. (e.g., `yum --disablerepo='somerepo'`)

disableexcludes Disable exclude from main, for a repo or for everything. (e.g., `yum --disableexcludes='main'`)

New in version 2014.7.

name The name of the package to be upgraded. Note that this parameter is ignored if "pkgs" is passed.

32-bit packages can be upgraded on 64-bit systems by appending the architecture designation (`.i686`, `.i586`, etc.) to the end of the package name.

Warning: if you forget 'name=' and run pkg.upgrade openssl, ALL packages are upgraded. This will be addressed in next releases.

CLI Example:

```
salt '*' pkg.upgrade name=openssl
```

New in version 2016.3.0.

pkgs A list of packages to upgrade from a software repository. Must be passed as a python list. A specific version number can be specified by using a single-element dict representing the package and its version. If the package was not already installed on the system, it will not be installed.

CLI Examples:

```
salt '*' pkg.upgrade pkgs='["foo", "bar"]'
salt '*' pkg.upgrade pkgs='["foo", {"bar": "1.2.3-4.el5"}]'
```

New in version 2016.3.0.

normalize [True] Normalize the package name by removing the architecture. This is useful for poorly created packages which might include the architecture as an actual part of the name such as kernel modules which match a specific kernel version.

```
salt -G role:nsd pkg.upgrade gpfs.gplbin-2.6.32-279.31.1.el6.x86_64␣
↪normalize=False
```

New in version 2016.3.0.

minimal [False] Use upgrade-minimal instead of upgrade (e.g., `yum upgrade-minimal`) Goes to the 'newest' package match which fixes a problem that affects your system.

```
salt '*' pkg.upgrade minimal=True
```

New in version 2019.2.0.

obsoletes [True] Controls wether yum/dnf should take obsoletes into account and remove them. If set to `False` yum will use `update` instead of `upgrade` and dnf will be run with `--obsoletes=False`

```
salt '*' pkg.upgrade obsoletes=False
```

New in version 2019.2.0.

setopt A comma-separated or Python list of key=value options. This list will be expanded and `--setopt` prepended to each in the yum/dnf command that is run.

New in version 2019.2.0.

Note: To add extra arguments to the `yum upgrade` command, pass them as key word arguments. For arguments without assignments, pass `True`

```
salt '*' pkg.upgrade security=True exclude='kernel*'
```

`salt.modules.yumpkg.`**`upgrade_available`**(*name*, ***kwargs*)

Check whether or not an upgrade is available for a given package

CLI Example:

```
salt '*' pkg.upgrade_available <package name>
```

`salt.modules.yumpkg.`**`verify`**(**names*, ***kwargs*)

New in version 2014.1.0.

Runs an rpm -Va on a system, and returns the results in a dict

Pass options to modify rpm verify behavior using the `verify_options` keyword argument

Files with an attribute of config, doc, ghost, license or readme in the package header can be ignored using the `ignore_types` keyword argument

CLI Example:

```
salt '*' pkg.verify
salt '*' pkg.verify httpd
salt '*' pkg.verify 'httpd postfix'
salt '*' pkg.verify 'httpd postfix' ignore_types=['config','doc']
salt '*' pkg.verify 'httpd postfix' verify_options=['nodeps','nosize']
```

`salt.modules.yumpkg.`**`version`**(**names*, ***kwargs*)

Returns a string representing the package version or an empty string if not installed. If more than one package name is specified, a dict of name/version pairs is returned.

CLI Example:

```
salt '*' pkg.version <package name>
salt '*' pkg.version <package1> <package2> <package3> ...
```

`salt.modules.yumpkg.`**`version_cmp`**(*pkg1*, *pkg2*, *ignore_epoch=False*)

New in version 2015.5.4.

Do a cmp-style comparison on two packages. Return -1 if pkg1 < pkg2, 0 if pkg1 == pkg2, and 1 if pkg1 > pkg2. Return None if there was a problem making the comparison.

ignore_epoch [False] Set to `True` to ignore the epoch when comparing versions

New in version 2015.8.10,2016.3.2.

CLI Example:

```
salt '*' pkg.version_cmp '0.2-001' '0.2.0.1-002'
```

21.9.495 salt.modules.zabbix module

Support for Zabbix

optdepends

- zabbix server

configuration This module is not usable until the zabbix user and zabbix password are specified either in a pillar or in the minion's config file. Zabbix url should be also specified.

```
zabbix.user: Admin
zabbix.password: mypassword
zabbix.url: http://127.0.0.1/zabbix/api_jsonrpc.php
```

Connection arguments from the minion config file can be overridden on the CLI by using arguments with `_connection_` prefix.

```
zabbix.apiinfo_version _connection_user=Admin _connection_
↪password=zabbix _connection_url=http://host/zabbix/
```

codeauthor Jiri Kotlin <jiri.kotlin@ultimum.io>

`salt.modules.zabbix.`**`apiinfo_version`**`(`*`**connection_args`*`)`

Retrieve the version of the Zabbix API.

New in version 2016.3.0.

Parameters

- **`_connection_user`** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **`_connection_password`** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **`_connection_url`** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

Returns On success string with Zabbix API version, False on failure.

CLI Example: .. code-block:: bash

salt '*' zabbix.apiinfo_version

`salt.modules.zabbix.`**`compare_params`**`(`*defined*, *existing*, *return_old_value=False*`)`

New in version 2017.7.

Compares Zabbix object definition against existing Zabbix object.

Parameters

- **`defined`** – Zabbix object definition taken from sls file.
- **`existing`** – Existing Zabbix object taken from result of an API call.
- **`return_old_value`** – Default False. If True, returns dict("old"=old_val, "new"=new_val) for rollback purpose.

Returns Params that are different from existing object. Result extended by object ID can be passed directly to Zabbix API update method.

`salt.modules.zabbix.`**`configuration_import`**`(`*config_file*, *rules=None*, *file_format='xml'*, ***connection_args*`)`

New in version 2017.7.

Imports Zabbix configuration specified in file to Zabbix server.

Parameters

- **`config_file`** – File with Zabbix config (local or remote)
- **`rules`** – Optional - Rules that have to be different from default (defaults are the same as in Zabbix web UI.)

- **file_format** – Config file format (default: xml)
- **_connection_user** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **_connection_password** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **_connection_url** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

CLI Example:

```
salt '*' zabbix.configuration_import salt://zabbix/config/zabbix_templates.xml
↪     "{'screens': {'createMissing': True, 'updateExisting': True}}"
```

`salt.modules.zabbix.`**`get_object_id_by_params`**(*obj*, *params=None*, ***connection_args*)

New in version 2017.7.

Get ID of single Zabbix object specified by its name.

Parameters

- **obj** – Zabbix object type
- **params** – Parameters by which object is uniquely identified
- **_connection_user** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **_connection_password** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **_connection_url** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

Returns object ID

`salt.modules.zabbix.`**`get_zabbix_id_mapper`**()

New in version 2017.7.

Make ZABBIX_ID_MAPPER constant available to state modules.

Returns ZABBIX_ID_MAPPER

`salt.modules.zabbix.`**`host_create`**(*host*, *groups*, *interfaces*, ***connection_args*)

New in version 2016.3.0.

Create new host

Note: This function accepts all standard host properties: keyword argument names differ depending on your zabbix version, see here.

Parameters

- **host** – technical name of the host
- **groups** – groupids of host groups to add the host to
- **interfaces** – interfaces to be created for the host
- **_connection_user** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **_connection_password** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **_connection_url** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)
- **visible_name** – string with visible name of the host, use 'visible_name' instead of 'name' parameter to not mess with value supplied from Salt sls file.

return: ID of the created host.

CLI Example:

```
salt '*' zabbix.host_create technicalname 4
interfaces='{type: 1, main: 1, useip: 1, ip: "192.168.3.1", dns: "", port: 10050}'
visible_name='Host Visible Name' inventory_mode=0 inventory='{"alias": "something
↪"}'
```

`salt.modules.zabbix.`**`host_delete`**(*hostids*, ***connection_args*)

Delete hosts.

New in version 2016.3.0.

Parameters

- **`hostids`** – Hosts (hostids) to delete.
- **`_connection_user`** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **`_connection_password`** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **`_connection_url`** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

Returns IDs of the deleted hosts.

CLI Example: .. code-block:: bash

salt '*' zabbix.host_delete 10106

`salt.modules.zabbix.`**`host_exists`**(*host=None*, *hostid=None*, *name=None*, *node=None*, *nodeids=None*, ***connection_args*)

Checks if at least one host that matches the given filter criteria exists.

New in version 2016.3.0.

Parameters

- **`host`** – technical name of the host
- **`hostids`** – Hosts (hostids) to delete.
- **`name`** – visible name of the host
- **`node`** – name of the node the hosts must belong to (zabbix API < 2.4)
- **`nodeids`** – IDs of the node the hosts must belong to (zabbix API < 2.4)
- **`_connection_user`** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **`_connection_password`** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **`_connection_url`** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

Returns IDs of the deleted hosts, False on failure.

CLI Example: .. code-block:: bash

salt '*' zabbix.host_exists 'Zabbix server'

`salt.modules.zabbix.`**`host_get`**(*host=None*, *name=None*, *hostids=None*, ***connection_args*)

New in version 2016.3.0.

Retrieve hosts according to the given parameters

Note: This function accepts all optional host.get parameters: keyword argument names differ depending on your zabbix version, see here.

Parameters

- **`host`** – technical name of the host
- **`name`** – visible name of the host
- **`hostids`** – ids of the hosts

- **_connection_user** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **_connection_password** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **_connection_url** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

Returns Array with convenient hosts details, False if no host found or on failure.

CLI Example: .. code-block:: bash

salt '*' zabbix.host_get 'Zabbix server'

`salt.modules.zabbix.`**`host_inventory_get`**`(`*hostids*, ***connection_args*`)`

Retrieve host inventory according to the given parameters. See: https://www.zabbix.com/documentation/2.4/manual/api/reference/host/object#host_inventory

New in version 2019.2.0.

Parameters

- **hostids** – Return only host interfaces used by the given hosts.
- **_connection_user** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **_connection_password** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **_connection_url** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

Returns Array with host interfaces details, False if no convenient host interfaces found or on failure.

CLI Example: .. code-block:: bash

salt '*' zabbix.host_inventory_get 101054

`salt.modules.zabbix.`**`host_inventory_set`**`(`*hostid*, ***connection_args*`)`

Update host inventory items NOTE: This function accepts all standard host: keyword argument names for inventory see: https://www.zabbix.com/documentation/2.4/manual/api/reference/host/object#host_inventory

New in version 2019.2.0.

Parameters

- **hostid** – ID of the host to update
- **clear_old** – Set to True in order to remove all existing inventory items before setting the specified items
- **_connection_user** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **_connection_password** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **_connection_url** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

Returns ID of the updated host, False on failure.

CLI Example: .. code-block:: bash

salt '*' zabbix.host_inventory_set 101054 asset_tag=jml3322 type=vm clear_old=True

`salt.modules.zabbix.`**`host_list`**`(`***connection_args*`)`

Retrieve all hosts.

New in version 2016.3.0.

Parameters

- **_connection_user** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **_connection_password** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)

- **`_connection_url`** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

Returns Array with details about hosts, False on failure.

CLI Example: .. code-block:: bash

salt '*' zabbix.host_list

`salt.modules.zabbix.`**`host_update`**`(`*hostid*, ***connection_args*`)`

New in version 2016.3.0.

Update existing hosts

Note: This function accepts all standard host and host.update properties: keyword argument names differ depending on your zabbix version, see the documentation for host objects and the documentation for updating hosts.

Parameters

- **`hostid`** – ID of the host to update
- **`_connection_user`** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **`_connection_password`** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **`_connection_url`** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)
- **`visible_name`** – string with visible name of the host, use 'visible_name' instead of 'name' parameter to not mess with value supplied from Salt sls file.

Returns ID of the updated host.

CLI Example: .. code-block:: bash

salt '*' zabbix.host_update 10084 name='Zabbix server2'

`salt.modules.zabbix.`**`hostgroup_create`**`(`*name*, ***connection_args*`)`

New in version 2016.3.0.

Create a host group

Note: This function accepts all standard host group properties: keyword argument names differ depending on your zabbix version, see here.

Parameters

- **`name`** – name of the host group
- **`_connection_user`** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **`_connection_password`** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **`_connection_url`** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

Returns ID of the created host group.

CLI Example: .. code-block:: bash

salt '*' zabbix.hostgroup_create MyNewGroup

`salt.modules.zabbix.`**`hostgroup_delete`**`(`*hostgroupids*, ***connection_args*`)`

Delete the host group.

New in version 2016.3.0.

Parameters

- **hostgroupids** – IDs of the host groups to delete
- **_connection_user** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **_connection_password** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **_connection_url** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

Returns ID of the deleted host groups, False on failure.

CLI Example: .. code-block:: bash

salt '*' zabbix.hostgroup_delete 23

`salt.modules.zabbix.`**`hostgroup_exists`**(*name=None*, *groupid=None*, *node=None*, *nodeids=None*, ***connection_args*)

Checks if at least one host group that matches the given filter criteria exists.

New in version 2016.3.0.

Parameters

- **name** – names of the host groups
- **groupid** – host group IDs
- **node** – name of the node the host groups must belong to (zabbix API < 2.4)
- **nodeids** – IDs of the nodes the host groups must belong to (zabbix API < 2.4)
- **_connection_user** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **_connection_password** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **_connection_url** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

Returns True if at least one host group exists, False if not or on failure.

CLI Example: .. code-block:: bash

salt '*' zabbix.hostgroup_exists MyNewGroup

`salt.modules.zabbix.`**`hostgroup_get`**(*name=None*, *groupids=None*, *hostids=None*, ***connection_args*)

New in version 2016.3.0.

Retrieve host groups according to the given parameters

Note: This function accepts all standard hostgroup.get properities: keyword argument names differ depending on your zabbix version, see here.

Parameters

- **name** – names of the host groups
- **groupid** – host group IDs
- **node** – name of the node the host groups must belong to
- **nodeids** – IDs of the nodes the host groups must belong to
- **hostids** – return only host groups that contain the given hosts
- **_connection_user** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **_connection_password** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **_connection_url** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

Returns Array with host groups details, False if no convenient host group found or on failure.

CLI Example: .. code-block:: bash

salt '*' zabbix.hostgroup_get MyNewGroup `salt.modules.zabbix.`**`hostgroup_list`**(***connection_args*)

Retrieve all host groups.

New in version 2016.3.0.

Parameters

- **`_connection_user`** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **`_connection_password`** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **`_connection_url`** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

Returns Array with details about host groups, False on failure.

CLI Example: .. code-block:: bash

salt '*' zabbix.hostgroup_list

`salt.modules.zabbix.`**`hostgroup_update`**(*groupid*, *name=None*, ***connection_args*)

New in version 2016.3.0.

Update existing hosts group

Note: This function accepts all standard host group properties: keyword argument names differ depending on your zabbix version, see here.

Parameters

- **`groupid`** – ID of the host group to update
- **`name`** – name of the host group
- **`_connection_user`** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **`_connection_password`** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **`_connection_url`** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

Returns IDs of updated host groups.

CLI Example: .. code-block:: bash

salt '*' zabbix.hostgroup_update 24 name='Renamed Name'

`salt.modules.zabbix.`**`hostinterface_create`**(*hostid*, *ip_*, *dns=''*, *main=1*, *if_type=1*, *useip=1*, *port=None*, ***connection_args*)

New in version 2016.3.0.

Create new host interface

Note: This function accepts all standard host group interface: keyword argument names differ depending on your zabbix version, see here.

Parameters

- **`hostid`** – ID of the host the interface belongs to
- **`ip`** – IP address used by the interface
- **`dns`** – DNS name used by the interface
- **`main`** – whether the interface is used as default on the host (0 - not default, 1 - default)
- **`port`** – port number used by the interface
- **`type`** – Interface type (1 - agent; 2 - SNMP; 3 - IPMI; 4 - JMX)
- **`useip`** – Whether the connection should be made via IP (0 - connect using host DNS name; 1 - connect using host IP address for this host interface)

- **`_connection_user`** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **`_connection_password`** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **`_connection_url`** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

Returns ID of the created host interface, False on failure.

CLI Example: .. code-block:: bash
salt '*' zabbix.hostinterface_create 10105 192.193.194.197

`salt.modules.zabbix.`**`hostinterface_delete`**`(`*interfaceids*, ***connection_args*`)`

Delete host interface

New in version 2016.3.0.

Parameters

- **`interfaceids`** – IDs of the host interfaces to delete
- **`_connection_user`** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **`_connection_password`** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **`_connection_url`** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

Returns ID of deleted host interfaces, False on failure.

CLI Example: .. code-block:: bash
salt '*' zabbix.hostinterface_delete 50

`salt.modules.zabbix.`**`hostinterface_get`**`(`*hostids*, ***connection_args*`)`

New in version 2016.3.0.

Retrieve host groups according to the given parameters

Note: This function accepts all standard hostinterface.get properities: keyword argument names differ depending on your zabbix version, see here.

Parameters

- **`hostids`** – Return only host interfaces used by the given hosts.
- **`_connection_user`** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **`_connection_password`** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **`_connection_url`** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

Returns Array with host interfaces details, False if no convenient host interfaces found or on failure.

CLI Example: .. code-block:: bash
salt '*' zabbix.hostinterface_get 101054

`salt.modules.zabbix.`**`hostinterface_update`**`(`*interfaceid*, ***connection_args*`)`

New in version 2016.3.0.

Update host interface

Note: This function accepts all standard hostinterface: keyword argument names differ depending on your zabbix version, see here.

Parameters

- **`interfaceid`** – ID of the hostinterface to update
- **`_connection_user`** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **`_connection_password`** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **`_connection_url`** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

Returns ID of the updated host interface, False on failure.

CLI Example: .. code-block:: bash

salt '*' zabbix.hostinterface_update 6 ip_=0.0.0.2

`salt.modules.zabbix.`**`mediatype_create`**`(`*name*, *mediatype*, ***connection_args*`)`

Create new mediatype

Note: This function accepts all standard mediatype properties: keyword argument names differ depending on your zabbix version, see here.

Parameters

- **`mediatype`** – media type - 0: email, 1: script, 2: sms, 3: Jabber, 100: Ez Texting
- **`exec_path`** – exec path - Required for script and Ez Texting types, see Zabbix API docs
- **`gsm_modem`** – exec path - Required for sms type, see Zabbix API docs
- **`smtp_email`** – email address from which notifications will be sent, required for email type
- **`smtp_helo`** – SMTP HELO, required for email type
- **`smtp_server`** – SMTP server, required for email type
- **`status`** – whether the media type is enabled - 0: enabled, 1: disabled
- **`username`** – authentication user, required for Jabber and Ez Texting types
- **`passwd`** – authentication password, required for Jabber and Ez Texting types
- **`_connection_user`** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **`_connection_password`** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **`_connection_url`** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

return: ID of the created mediatype.

CLI Example:

```
salt '*' zabbix.mediatype_create 'Email' 0 smtp_email='noreply@example.com'
smtp_server='mailserver.example.com' smtp_helo='zabbix.example.com'
```

`salt.modules.zabbix.`**`mediatype_delete`**`(`*mediatypeids*, ***connection_args*`)`

Delete mediatype

Parameters

- **`interfaceids`** – IDs of the mediatypes to delete
- **`_connection_user`** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **`_connection_password`** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)

- **`_connection_url`** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

Returns ID of deleted mediatype, False on failure.

CLI Example: .. code-block:: bash

salt '*' zabbix.mediatype_delete 3

`salt.modules.zabbix.`**`mediatype_get`**(*name=None*, *mediatypeids=None*, ***connection_args*)

Retrieve mediatypes according to the given parameters.

Parameters

- **`name`** – Name or description of the mediatype
- **`mediatypeids`** – ids of the mediatypes
- **`connection_args`** (*`optional`*) – _connection_user: zabbix user (can also be set in opts or pillar, see module's docstring) _connection_password: zabbix password (can also be set in opts or pillar, see module's docstring) _connection_url: url of zabbix frontend (can also be set in opts or pillar, see module's docstring)

  all optional mediatype.get parameters: keyword argument names depends on your zabbix version, see:

  https://www.zabbix.com/documentation/2.2/manual/api/reference/mediatype/get

Returns Array with mediatype details, False if no mediatype found or on failure.

CLI Example: .. code-block:: bash

salt '*' zabbix.mediatype_get name='Email' salt '*' zabbix.mediatype_get mediatypeids="['1', '2', '3']"

`salt.modules.zabbix.`**`mediatype_update`**(*mediatypeid*, *name=False*, *mediatype=False*, ***connection_args*)

Update existing mediatype

Note: This function accepts all standard mediatype properties: keyword argument names differ depending on your zabbix version, see here.

Parameters

- **`mediatypeid`** – ID of the mediatype to update
- **`_connection_user`** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **`_connection_password`** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **`_connection_url`** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

Returns IDs of the updated mediatypes, False on failure.

CLI Example: .. code-block:: bash

salt '*' zabbix.usergroup_update 8 name="Email update"

`salt.modules.zabbix.`**`run_query`**(*method*, *params*, ***connection_args*)

Send Zabbix API call

Parameters

- **`method`** – actual operation to perform via the API
- **`params`** – parameters required for specific method
- **`connection_args`** (*`optional`*) – _connection_user: zabbix user (can also be set in opts or pillar, see module's docstring) _connection_password: zabbix password (can also be set in opts or pillar, see module's docstring) _connection_url: url of zabbix frontend (can also be set in opts or pillar, see module's docstring)

  all optional template.get parameters: keyword argument names depends on your zabbix version, see:

https://www.zabbix.com/documentation/2.4/manual/api/reference/

Returns Response from Zabbix API

CLI Example: .. code-block:: bash

salt '*' zabbix.run_query proxy.create '{"host": "zabbixproxy.domain.com", "status": "5"}'

`salt.modules.zabbix.`**`substitute_params`**(*input_object*, *extend_params=None*, *filter_key='name'*, ***kwargs*)

New in version 2017.7.

Go through Zabbix object params specification and if needed get given object ID from Zabbix API and put it back as a value. Definition of the object is done via dict with keys "query_object" and "query_name".

Parameters

- **input_object** – Zabbix object type specified in state file
- **extend_params** – Specify query with params
- **filter_key** – Custom filtering key (default: name)
- **_connection_user** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **_connection_password** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **_connection_url** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

Returns Params structure with values converted to string for further comparison purposes

`salt.modules.zabbix.`**`template_get`**(*name=None*, *host=None*, *templateids=None*, ***connection_args*)

Retrieve templates according to the given parameters.

Parameters

- **host** – technical name of the template
- **name** – visible name of the template
- **hostids** – ids of the templates
- **connection_args** (*optional*) – _connection_user: zabbix user (can also be set in opts or pillar, see module's docstring) _connection_password: zabbix password (can also be set in opts or pillar, see module's docstring) _connection_url: url of zabbix frontend (can also be set in opts or pillar, see module's docstring)

  all optional template.get parameters: keyword argument names depends on your zabbix version, see:

  https://www.zabbix.com/documentation/2.4/manual/api/reference/template/get

Returns Array with convenient template details, False if no template found or on failure.

CLI Example: .. code-block:: bash

salt '*' zabbix.template_get name='Template OS Linux' salt '*' zabbix.template_get templateids="['10050', '10001']"

`salt.modules.zabbix.`**`user_addmedia`**(*userids*, *active*, *mediatypeid*, *period*, *sendto*, *severity*, ***connection_args*)

Add new media to multiple users.

New in version 2016.3.0.

Parameters

- **userids** – ID of the user that uses the media
- **active** – Whether the media is enabled (0 enabled, 1 disabled)
- **mediatypeid** – ID of the media type used by the media
- **period** – Time when the notifications can be sent as a time period
- **sendto** – Address, user name or other identifier of the recipient
- **severity** – Trigger severities to send notifications about
- **_connection_user** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)

- **`_connection_password`** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **`_connection_url`** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

Returns IDs of the created media.

CLI Example: .. code-block:: bash

salt '*' zabbix.user_addmedia 4 active=0 mediatypeid=1 period='1-7,00:00-24:00' sendto='support2@example.com' severity=63

`salt.modules.zabbix.`**`user_create`**(*alias*, *passwd*, *usrgrps*, ***connection_args*)

New in version 2016.3.0.

Create new zabbix user

Note: This function accepts all standard user properties: keyword argument names differ depending on your zabbix version, see here.

Parameters

- **`alias`** – user alias
- **`passwd`** – user's password
- **`usrgrps`** – user groups to add the user to
- **`_connection_user`** – zabbix user (can also be set in opts or pillar, see module's docstring)
- **`_connection_password`** – zabbix password (can also be set in opts or pillar, see module's docstring)
- **`_connection_url`** – url of zabbix frontend (can also be set in opts or pillar, see module's docstring)
- **`firstname`** – string with firstname of the user, use 'firstname' instead of 'name' parameter to not mess with value supplied from Salt sls file.

Returns On success string with id of the created user.

CLI Example: .. code-block:: bash

salt '*' zabbix.user_create james password007 '[7, 12]' firstname='James Bond'

`salt.modules.zabbix.`**`user_delete`**(*users*, ***connection_args*)

Delete zabbix users.

New in version 2016.3.0.

Parameters

- **`users`** – array of users (userids) to delete
- **`_connection_user`** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **`_connection_password`** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **`_connection_url`** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

Returns On success array with userids of deleted users.

CLI Example: .. code-block:: bash

salt '*' zabbix.user_delete 15

`salt.modules.zabbix.`**`user_deletemedia`**(*mediaids*, ***connection_args*)

Delete media by id.

New in version 2016.3.0.

Parameters

- **`mediaids`** – IDs of the media to delete

- **_connection_user** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **_connection_password** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **_connection_url** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

Returns IDs of the deleted media, False on failure.

CLI Example: .. code-block:: bash

salt '*' zabbix.user_deletemedia 27

`salt.modules.zabbix.`**`user_exists`**`(`*alias*, ***connection_args*`)`

Checks if user with given alias exists.

New in version 2016.3.0.

Parameters

- **alias** – user alias
- **_connection_user** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **_connection_password** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **_connection_url** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

Returns True if user exists, else False.

CLI Example: .. code-block:: bash

salt '*' zabbix.user_exists james

`salt.modules.zabbix.`**`user_get`**`(`*alias=None*, *userids=None*, ***connection_args*`)`

Retrieve users according to the given parameters.

New in version 2016.3.0.

Parameters

- **alias** – user alias
- **userids** – return only users with the given IDs
- **_connection_user** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **_connection_password** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **_connection_url** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

Returns Array with details of convenient users, False on failure of if no user found.

CLI Example: .. code-block:: bash

salt '*' zabbix.user_get james

`salt.modules.zabbix.`**`user_getmedia`**`(`*userids=None*, ***connection_args*`)`

New in version 2016.3.0.

Retrieve media according to the given parameters

Note: This function accepts all standard usermedia.get properties: keyword argument names differ depending on your zabbix version, see here.

Parameters

- **userids** – return only media that are used by the given users
- **_connection_user** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)

- **`_connection_password`** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **`_connection_url`** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

Returns List of retrieved media, False on failure.

CLI Example: .. code-block:: bash
salt '*' zabbix.user_getmedia

`salt.modules.zabbix.`**`user_list`**`(`*`**connection_args`*`)`

Retrieve all of the configured users.

New in version 2016.3.0.

Parameters

- **`_connection_user`** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **`_connection_password`** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **`_connection_url`** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

Returns Array with user details.

CLI Example: .. code-block:: bash
salt '*' zabbix.user_list

`salt.modules.zabbix.`**`user_update`**`(`*`userid`*`, `*`**connection_args`*`)`

New in version 2016.3.0.

Update existing users

Note: This function accepts all standard user properties: keyword argument names differ depending on your zabbix version, see here.

Parameters

- **`userid`** – id of the user to update
- **`_connection_user`** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **`_connection_password`** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **`_connection_url`** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

Returns Id of the updated user on success.

CLI Example: .. code-block:: bash
salt '*' zabbix.user_update 16 visible_name='James Brown'

`salt.modules.zabbix.`**`usergroup_create`**`(`*`name`*`, `*`**connection_args`*`)`

New in version 2016.3.0.

Create new user group

Note: This function accepts all standard user group properties: keyword argument names differ depending on your zabbix version, see here.

Parameters

- **`name`** – name of the user group

- **`_connection_user`** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **`_connection_password`** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **`_connection_url`** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

Returns IDs of the created user groups.

CLI Example: .. code-block:: bash

salt '*' zabbix.usergroup_create GroupName

`salt.modules.zabbix.`**`usergroup_delete`**`(`*usergroupids*, ***connection_args*`)`

New in version 2016.3.0.

Parameters

- **`usergroupids`** – IDs of the user groups to delete
- **`_connection_user`** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **`_connection_password`** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **`_connection_url`** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

Returns IDs of the deleted user groups.

CLI Example: .. code-block:: bash

salt '*' zabbix.usergroup_delete 28

`salt.modules.zabbix.`**`usergroup_exists`**`(`*name=None*, *node=None*, *nodeids=None*, ***connection_args*`)`

Checks if at least one user group that matches the given filter criteria exists

New in version 2016.3.0.

Parameters

- **`name`** – names of the user groups
- **`node`** – name of the node the user groups must belong to (This will override the nodeids parameter.)
- **`nodeids`** – IDs of the nodes the user groups must belong to
- **`_connection_user`** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **`_connection_password`** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **`_connection_url`** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

Returns True if at least one user group that matches the given filter criteria exists, else False.

CLI Example: .. code-block:: bash

salt '*' zabbix.usergroup_exists Guests

`salt.modules.zabbix.`**`usergroup_get`**`(`*name=None*, *usrgrpids=None*, *userids=None*, ***connection_args*`)`

New in version 2016.3.0.

Retrieve user groups according to the given parameters

Note: This function accepts all usergroup_get properties: keyword argument names differ depending on your zabbix version, see here.

Parameters

- **`name`** – names of the user groups

- **usrgrpids** – return only user groups with the given IDs
- **userids** – return only user groups that contain the given users
- **_connection_user** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **_connection_password** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **_connection_url** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

Returns Array with convenient user groups details, False if no user group found or on failure.

CLI Example: .. code-block:: bash
salt '*' zabbix.usergroup_get Guests

salt.modules.zabbix.**usergroup_list**(***connection_args*)

Retrieve all enabled user groups.

New in version 2016.3.0.

Parameters

- **_connection_user** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **_connection_password** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **_connection_url** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

Returns Array with enabled user groups details, False on failure.

CLI Example: .. code-block:: bash
salt '*' zabbix.usergroup_list

salt.modules.zabbix.**usergroup_update**(*usrgrpid*, ***connection_args*)

New in version 2016.3.0.

Update existing user group

Note: This function accepts all standard user group properties: keyword argument names differ depending on your zabbix version, see here.

Parameters

- **usrgrpid** – ID of the user group to update.
- **_connection_user** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **_connection_password** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **_connection_url** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

Returns IDs of the updated user group, False on failure.

CLI Example: .. code-block:: bash
salt '*' zabbix.usergroup_update 8 name=guestsRenamed

salt.modules.zabbix.**usermacro_create**(*macro*, *value*, *hostid*, ***connection_args*)

Create new host usermacro.

Parameters

- **macro** – name of the host usermacro
- **value** – value of the host usermacro
- **hostid** – hostid or templateid

- **_connection_user** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **_connection_password** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **_connection_url** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

return: ID of the created host usermacro.

CLI Example:

```
salt '*' zabbix.usermacro_create '{$SNMP_COMMUNITY}' 'public' 1
```

salt.modules.zabbix.**usermacro_createglobal**(*macro*, *value*, ***connection_args*)

Create new global usermacro.

Parameters

- **macro** – name of the global usermacro
- **value** – value of the global usermacro
- **_connection_user** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **_connection_password** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **_connection_url** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

return: ID of the created global usermacro.

CLI Example:

```
salt '*' zabbix.usermacro_createglobal '{$SNMP_COMMUNITY}' 'public'
```

salt.modules.zabbix.**usermacro_delete**(*macroids*, ***connection_args*)

Delete host usermacros.

Parameters

- **macroids** – macroids of the host usermacros
- **_connection_user** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **_connection_password** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **_connection_url** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

return: IDs of the deleted host usermacro.

CLI Example:

```
salt '*' zabbix.usermacro_delete 21
```

salt.modules.zabbix.**usermacro_deleteglobal**(*macroids*, ***connection_args*)

Delete global usermacros.

Parameters

- **macroids** – macroids of the global usermacros
- **_connection_user** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **_connection_password** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **_connection_url** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

return: IDs of the deleted global usermacro.

CLI Example:

```
salt '*' zabbix.usermacro_deleteglobal 21
```

`salt.modules.zabbix.`**`usermacro_get`**`(`*macro=None*, *hostids=None*, *templateids=None*, *hostmacroids=None*, *globalmacroids=None*, *globalmacro=False*, ***connection_args*`)`

Retrieve user macros according to the given parameters.

Parameters

- **`macro`** – name of the usermacro
- **`hostids`** – Return macros for the given hostids
- **`templateids`** – Return macros for the given templateids
- **`hostmacroids`** – Return macros with the given hostmacroids
- **`globalmacroids`** – Return macros with the given globalmacroids (implies globalmacro=True)
- **`globalmacro`** – if True, returns only global macros
- **`connection_args`** (*`optional`*) – _connection_user: zabbix user (can also be set in opts or pillar, see module's docstring) _connection_password: zabbix password (can also be set in opts or pillar, see module's docstring) _connection_url: url of zabbix frontend (can also be set in opts or pillar, see module's docstring)

Returns Array with usermacro details, False if no usermacro found or on failure.

CLI Example: .. code-block:: bash

salt '*' zabbix.usermacro_get macro='{$SNMP_COMMUNITY}'

`salt.modules.zabbix.`**`usermacro_update`**`(`*hostmacroid*, *value*, ***connection_args*`)`

Update existing host usermacro.

Parameters

- **`hostmacroid`** – id of the host usermacro
- **`value`** – new value of the host usermacro
- **`_connection_user`** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **`_connection_password`** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **`_connection_url`** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

return: ID of the update host usermacro.

CLI Example:

```
salt '*' zabbix.usermacro_update 1 'public'
```

`salt.modules.zabbix.`**`usermacro_updateglobal`**`(`*globalmacroid*, *value*, ***connection_args*`)`

Update existing global usermacro.

Parameters

- **`globalmacroid`** – id of the host usermacro
- **`value`** – new value of the host usermacro
- **`_connection_user`** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **`_connection_password`** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **`_connection_url`** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

return: ID of the update global usermacro.

CLI Example:

```
salt '*' zabbix.usermacro_updateglobal 1 'public'
```

21.9.496 salt.modules.zcbuildout

Management of zc.buildout

New in version 2014.1.0.

This module is inspired by minitage's buildout maker

Note: The zc.buildout integration is still in beta; the API is subject to change

---

General notes

You have those following methods:

- upgrade_bootstrap
- bootstrap
- run_buildout
- buildout

`salt.modules.zcbuildout.`**`bootstrap`**`(*a, **kw)`

Run the buildout bootstrap dance (python bootstrap.py).

directory directory to execute in
config alternative buildout configuration file to use
runas User used to run buildout as
env environment variables to set when running
buildout_ver force a specific buildout version (1 | 2)
test_release buildout accept test release
offline are we executing buildout in offline mode
distribute Forcing use of distribute
new_st Forcing use of setuptools >= 0.7
python path to a python executable to use in place of default (salt one)
onlyif Only execute cmd if statement on the host return 0
unless Do not execute cmd if statement on the host return 0
use_vt Use the new salt VT to stream output [experimental]
CLI Example:

```
salt '*' buildout.bootstrap /srv/mybuildout
```

`salt.modules.zcbuildout.`**`buildout`**`(*a, **kw)`

Run buildout in a directory.

directory directory to execute in
config buildout config to use
parts specific buildout parts to run
runas user used to run buildout as
env environment variables to set when running
buildout_ver force a specific buildout version (1 | 2)
test_release buildout accept test release
new_st Forcing use of setuptools >= 0.7
distribute use distribute over setuptools if possible

offline does buildout run offline
python python to use
debug run buildout with -D debug flag
onlyif Only execute cmd if statement on the host return 0
unless Do not execute cmd if statement on the host return 0
newest run buildout in newest mode
verbose run buildout in verbose mode (-vvvvv)
use_vt Use the new salt VT to stream output [experimental]
CLI Example:

```
salt '*' buildout.buildout /srv/mybuildout
```

salt.modules.zcbuildout.**run_buildout**(**a*, ***kw*)

Run a buildout in a directory.
directory directory to execute in
config alternative buildout configuration file to use
offline are we executing buildout in offline mode
runas user used to run buildout as
env environment variables to set when running
onlyif Only execute cmd if statement on the host return 0
unless Do not execute cmd if statement on the host return 0
newest run buildout in newest mode
force run buildout unconditionally
verbose run buildout in verbose mode (-vvvvv)
use_vt Use the new salt VT to stream output [experimental]
CLI Example:

```
salt '*' buildout.run_buildout /srv/mybuildout
```

salt.modules.zcbuildout.**upgrade_bootstrap**(**a*, ***kw*)

Upgrade current bootstrap.py with the last released one.

Indeed, when we first run a buildout, a common source of problem is to have a locally stale bootstrap, we just try to grab a new copy
directory directory to execute in
offline are we executing buildout in offline mode
buildout_ver forcing to use a specific buildout version (1 | 2)
onlyif Only execute cmd if statement on the host return 0
unless Do not execute cmd if statement on the host return 0
CLI Example:

```
salt '*' buildout.upgrade_bootstrap /srv/mybuildout
```

21.9.497 salt.modules.zenoss

Module for working with the Zenoss API

New in version 2016.3.0.

depends requests

configuration This module requires a 'zenoss' entry in the master/minion config.

For example:

```
zenoss:
  hostname: https://zenoss.example.com
  username: admin
  password: admin123
```

`salt.modules.zenoss.`**`add_device`**(*device=None*, *device_class=None*, *collector='localhost'*, *prod_state=1000*)

A function to connect to a zenoss server and add a new device entry.

Parameters

- **`device`** – (Optional) Will use the grain 'fqdn' by default.
- **`device_class`** – (Optional) The device class to use. If none, will determine based on kernel grain.
- **`collector`** – (Optional) The collector to use for this device. Defaults to 'localhost'.
- **`prod_state`** – (Optional) The prodState to set on the device. If none, defaults to 1000 (production)

CLI Example: salt '*' zenoss.add_device

`salt.modules.zenoss.`**`device_exists`**(*device=None*)

Check to see if a device already exists in Zenoss.

Parameters **`device`** – (Optional) Will use the grain 'fqdn' by default

CLI Example: salt '*' zenoss.device_exists

`salt.modules.zenoss.`**`find_device`**(*device=None*)

Find a device in Zenoss. If device not found, returns None.

Parameters **`device`** – (Optional) Will use the grain 'fqdn' by default

CLI Example: salt '*' zenoss.find_device

`salt.modules.zenoss.`**`set_prod_state`**(*prod_state*, *device=None*)

A function to set the prod_state in zenoss.

Parameters

- **`prod_state`** – (Required) Integer value of the state
- **`device`** – (Optional) Will use the grain 'fqdn' by default.

CLI Example: salt zenoss.set_prod_state 1000 hostname

21.9.498 salt.modules.zfs

Module for running ZFS command

codeauthor Nitin Madhok <nmadhok@clemson.edu>, Jorge Schrauwen <sjorge@blackdot.be>

maintainer Jorge Schrauwen <sjorge@blackdot.be>

maturity new

depends salt.utils.zfs

platform illumos,freebsd,linux

Changed in version 2018.3.1: Big refactor to remove duplicate code, better type converions and improved consistancy in output.

`salt.modules.zfs.`**`bookmark`**(*snapshot*, *bookmark*)

Creates a bookmark of the given snapshot

Note: Bookmarks mark the point in time when the snapshot was created, and can be used as the incremental source for a zfs send command.

This feature must be enabled to be used. See zpool-features(5) for details on ZFS feature flags and the bookmarks feature.

---

snapshot [string] name of snapshot to bookmark
bookmark [string] name of bookmark

New in version 2016.3.0.

CLI Example:

```
salt '*' zfs.bookmark myzpool/mydataset@yesterday myzpool/mydataset#complete
```

`salt.modules.zfs.`**`clone`**(*name_a*, *name_b*, ***kwargs*)

Creates a clone of the given snapshot.
name_a [string] name of snapshot
name_b [string] name of filesystem or volume
create_parent [boolean] creates all the non-existing parent datasets. any property specified on the command line using the -o option is ignored.
properties [dict] additional zfs properties (-o)

Note: ZFS properties can be specified at the time of creation of the filesystem by passing an additional argument called "properties" and specifying the properties with their respective values in the form of a python dictionary:

```
properties="{'property1': 'value1', 'property2': 'value2'}"
```

New in version 2016.3.0.

CLI Example:

```
salt '*' zfs.clone myzpool/mydataset@yesterday myzpool/mydataset_yesterday
```

`salt.modules.zfs.`**`create`**(*name*, ***kwargs*)

Create a ZFS File System.
name [string] name of dataset or volume
volume_size [string] if specified, a zvol will be created instead of a dataset
sparse [boolean] create sparse volume
create_parent [boolean] creates all the non-existing parent datasets. any property specified on the command line using the -o option is ignored.
properties [dict] additional zfs properties (-o)

Note: ZFS properties can be specified at the time of creation of the filesystem by passing an additional argument called "properties" and specifying the properties with their respective values in the form of a python dictionary:

```
properties="{'property1': 'value1', 'property2': 'value2'}"
```

New in version 2015.5.0.

CLI Example:

```
salt '*' zfs.create myzpool/mydataset [create_parent=True|False]
salt '*' zfs.create myzpool/mydataset properties="{'mountpoint': '/export/zfs',
↪'sharenfs': 'on'}"
salt '*' zfs.create myzpool/volume volume_size=1G [sparse=True|False]`
salt '*' zfs.create myzpool/volume volume_size=1G properties="{'volblocksize':
↪'512'}" [sparse=True|False]
```

`salt.modules.zfs.`**`destroy`**`(`*name*, ***kwargs*`)`

Destroy a ZFS File System.

name [string] name of dataset, volume, or snapshot

force [boolean] force an unmount of any file systems using the unmount -f command.

recursive [boolean] recursively destroy all children. (-r)

recursive_all [boolean] recursively destroy all dependents, including cloned file systems outside the target hierarchy. (-R)

> **Warning:** watch out when using recursive and recursive_all

New in version 2015.5.0.

CLI Example:

```
salt '*' zfs.destroy myzpool/mydataset [force=True|False]
```

`salt.modules.zfs.`**`diff`**`(`*name_a*, *name_b=None*, ***kwargs*`)`

Display the difference between a snapshot of a given filesystem and another snapshot of that filesystem from a later time or the current contents of the filesystem.

name_a [string] name of snapshot

name_b [string] (optional) name of snapshot or filesystem

show_changetime [boolean] display the path's inode change time as the first column of output. (default = True)

show_indication [boolean] display an indication of the type of file. (default = True)

parsable [boolean] if true we don't parse the timestamp to a more readable date (default = True)

New in version 2016.3.0.

CLI Example:

```
salt '*' zfs.diff myzpool/mydataset@yesterday myzpool/mydataset
```

`salt.modules.zfs.`**`exists`**`(`*name*, ***kwargs*`)`

Check if a ZFS filesystem or volume or snapshot exists.

name [string] name of dataset

type [string] also check if dataset is of a certain type, valid choices are: filesystem, snapshot, volume, bookmark, or all.

New in version 2015.5.0.

CLI Example:

```
salt '*' zfs.exists myzpool/mydataset
salt '*' zfs.exists myzpool/myvolume type=volume
```

`salt.modules.zfs.`**`get`**`(`**dataset*, ***kwargs*`)`

Displays properties for the given datasets.

dataset [string] name of snapshot(s), filesystem(s), or volume(s)

properties [string] comma-separated list of properties to list, defaults to all

recursive [boolean] recursively list children

depth [int] recursively list children to depth

fields [string] comma-separated list of fields to include, the name and property field will always be added

type [string] comma-separated list of types to display, where type is one of filesystem, snapshot, volume, bookmark, or all.

source [string] comma-separated list of sources to display. Must be one of the following: local, default, inherited, temporary, and none. The default value is all sources.

parsable [boolean] display numbers in parsable (exact) values (default = True) .. versionadded:: 2018.3.0

Note: If no datasets are specified, then the command displays properties for all datasets on the system.

New in version 2016.3.0.

CLI Example:

```
salt '*' zfs.get
salt '*' zfs.get myzpool/mydataset [recursive=True|False]
salt '*' zfs.get myzpool/mydataset properties="sharenfs,mountpoint"
↪[recursive=True|False]
salt '*' zfs.get myzpool/mydataset myzpool/myotherdataset properties=available
↪fields=value depth=1
```

`salt.modules.zfs.`**`hold`**(*tag*, **snapshot*, ***kwargs*)

Adds a single reference, named with the tag argument, to the specified snapshot or snapshots.

Note: Each snapshot has its own tag namespace, and tags must be unique within that space.

If a hold exists on a snapshot, attempts to destroy that snapshot by using the zfs destroy command return EBUSY.

---

tag [string] name of tag

snapshot [string] name of snapshot(s)

recursive [boolean] specifies that a hold with the given tag is applied recursively to the snapshots of all descendent file systems.

New in version 2016.3.0.

Changed in version 2018.3.1.

Warning: As of 2018.3.1 the tag parameter no longer accepts a comma-separated value. It's is now possible to create a tag that contains a comma, this was impossible before.

CLI Example:

```
salt '*' zfs.hold mytag myzpool/mydataset@mysnapshot [recursive=True]
salt '*' zfs.hold mytag myzpool/mydataset@mysnapshot myzpool/
↪mydataset@myothersnapshot
```

`salt.modules.zfs.`**`holds`**(*snapshot*, ***kwargs*)

Lists all existing user references for the given snapshot or snapshots.

snapshot [string] name of snapshot

recursive [boolean] lists the holds that are set on the named descendent snapshots also.

New in version 2016.3.0.

CLI Example:

```
salt '*' zfs.holds myzpool/mydataset@baseline
```

`salt.modules.zfs.`**`inherit`**`(`*prop*, *name*, ***kwargs*`)`

Clears the specified property

prop [string] name of property

name [string] name of the filesystem, volume, or snapshot

recursive [boolean] recursively inherit the given property for all children.

revert [boolean] revert the property to the received value if one exists; otherwise operate as if the -S option was not specified.

New in version 2016.3.0.

CLI Example:

```
salt '*' zfs.inherit canmount myzpool/mydataset [recursive=True|False]
```

`salt.modules.zfs.`**`list_`**`(`*name=None*, ***kwargs*`)`

Return a list of all datasets or a specified dataset on the system and the values of their used, available, referenced, and mountpoint properties.

name [string] name of dataset, volume, or snapshot

recursive [boolean] recursively list children

depth [int] limit recursion to depth

properties [string] comma-separated list of properties to list, the name property will always be added

type [string] comma-separated list of types to display, where type is one of filesystem, snapshot, volume, bookmark, or all.

sort [string] property to sort on (default = name)

order [string [ascending|descending]] sort order (default = ascending)

parsable [boolean] display numbers in parsable (exact) values .. versionadded:: 2018.3.0

New in version 2015.5.0.

CLI Example:

```
salt '*' zfs.list
salt '*' zfs.list myzpool/mydataset [recursive=True|False]
salt '*' zfs.list myzpool/mydataset properties="sharenfs,mountpoint"
```

`salt.modules.zfs.`**`list_mount`**`()`

List mounted zfs filesystems

New in version 2018.3.1.

CLI Example:

```
salt '*' zfs.list_mount
```

`salt.modules.zfs.`**`mount`**`(`*name=None*, ***kwargs*`)`

Mounts ZFS file systems

name [string] name of the filesystem, having this set to None will mount all filesystems. (this is the default)

overlay [boolean] perform an overlay mount.

options [string] optional comma-separated list of mount options to use temporarily for the duration of the mount.

New in version 2016.3.0.

Changed in version 2018.3.1.

Warning: Passing '-a' as name is deprecated and will be removed in Sodium.

CLI Example:

```
salt '*' zfs.mount
salt '*' zfs.mount myzpool/mydataset
salt '*' zfs.mount myzpool/mydataset options=ro
```

`salt.modules.zfs.`**`promote`**(*name*)

Promotes a clone file system to no longer be dependent on its "origin" snapshot.

Note: This makes it possible to destroy the file system that the clone was created from. The clone parent-child dependency relationship is reversed, so that the origin file system becomes a clone of the specified file system.

The snapshot that was cloned, and any snapshots previous to this snapshot, are now owned by the promoted clone. The space they use moves from the origin file system to the promoted clone, so enough space must be available to accommodate these snapshots. No new space is consumed by this operation, but the space accounting is adjusted. The promoted clone must not have any conflicting snapshot names of its own. The rename subcommand can be used to rename any conflicting snapshots.

---

name [string] name of clone-filesystem

New in version 2016.3.0.

CLI Example:

```
salt '*' zfs.promote myzpool/myclone
```

`salt.modules.zfs.`**`release`**(*tag*, **snapshot*, ***kwargs*)

Removes a single reference, named with the tag argument, from the specified snapshot or snapshots.

Note: The tag must already exist for each snapshot. If a hold exists on a snapshot, attempts to destroy that snapshot by using the zfs destroy command return EBUSY.

---

tag [string] name of tag
snapshot [string] name of snapshot(s)
recursive [boolean] recursively releases a hold with the given tag on the snapshots of all descendent file systems.

New in version 2016.3.0.

Changed in version 2018.3.1.

Warning: As of 2018.3.1 the tag parameter no longer accepts a comma-separated value. It's is now possible to create a tag that contains a comma, this was impossible before.

CLI Example:

```
salt '*' zfs.release mytag myzpool/mydataset@mysnapshot [recursive=True]
salt '*' zfs.release mytag myzpool/mydataset@mysnapshot myzpool/
↪mydataset@myothersnapshot
```

`salt.modules.zfs.`**`rename`**(*name*, *new_name*, ***kwargs*)

Rename or Relocate a ZFS File System.
name [string] name of dataset, volume, or snapshot
new_name [string] new name of dataset, volume, or snapshot

force [boolean] force unmount any filesystems that need to be unmounted in the process.

create_parent [boolean] creates all the nonexistent parent datasets. Datasets created in this manner are automatically mounted according to the mountpoint property inherited from their parent.

recursive [boolean] recursively rename the snapshots of all descendent datasets. snapshots are the only dataset that can be renamed recursively.

New in version 2015.5.0.

CLI Example:

```
salt '*' zfs.rename myzpool/mydataset myzpool/renameddataset
```

`salt.modules.zfs.`**`rollback`**`(`*name*, ***kwargs*`)`

Roll back the given dataset to a previous snapshot.

name [string] name of snapshot

recursive [boolean] destroy any snapshots and bookmarks more recent than the one specified.

recursive_all [boolean] destroy any more recent snapshots and bookmarks, as well as any clones of those snapshots.

force [boolean] used with the -R option to force an unmount of any clone file systems that are to be destroyed.

Warning: When a dataset is rolled back, all data that has changed since the snapshot is discarded, and the dataset reverts to the state at the time of the snapshot. By default, the command refuses to roll back to a snapshot other than the most recent one.

In order to do so, all intermediate snapshots and bookmarks must be destroyed by specifying the -r option.

New in version 2016.3.0.

CLI Example:

```
salt '*' zfs.rollback myzpool/mydataset@yesterday
```

`salt.modules.zfs.`**`set`**`(`**dataset*, ***kwargs*`)`

Sets the property or list of properties to the given value(s) for each dataset.

dataset [string] name of snapshot(s), filesystem(s), or volume(s)

properties [string] additional zfs properties pairs

Note: properties are passed as key-value pairs. e.g.

compression=off

Note: Only some properties can be edited.

See the Properties section for more information on what properties can be set and acceptable values.

Numeric values can be specified as exact values, or in a human-readable form with a suffix of B, K, M, G, T, P, E (for bytes, kilobytes, megabytes, gigabytes, terabytes, petabytes, or exabytes respectively).

New in version 2016.3.0.

CLI Example:

```
salt '*' zfs.set myzpool/mydataset compression=off
salt '*' zfs.set myzpool/mydataset myzpool/myotherdataset compression=off
salt '*' zfs.set myzpool/mydataset myzpool/myotherdataset compression=lz4␣
↪canmount=off
```

`salt.modules.zfs.`**`snapshot`**`(`**snapshot*, ***kwargs*`)`

Creates snapshots with the given names.

snapshot [string] name of snapshot(s)

recursive [boolean] recursively create snapshots of all descendent datasets.

properties [dict] additional zfs properties (-o)

Note: ZFS properties can be specified at the time of creation of the filesystem by passing an additional argument called "properties" and specifying the properties with their respective values in the form of a python dictionary:

```
properties="{'property1': 'value1', 'property2': 'value2'}"
```

New in version 2016.3.0.

CLI Example:

```
salt '*' zfs.snapshot myzpool/mydataset@yesterday [recursive=True]
salt '*' zfs.snapshot myzpool/mydataset@yesterday myzpool/
↪myotherdataset@yesterday [recursive=True]
```

`salt.modules.zfs.`**`unmount`**`(`*name*, ***kwargs*`)`

Unmounts ZFS file systems

name [string] name of the filesystem, you can use None to unmount all mounted filesystems.

force [boolean] forcefully unmount the file system, even if it is currently in use.

Warning: Using `-a` for the name parameter will probably break your system, unless your rootfs is not on zfs.

New in version 2016.3.0.

Changed in version 2018.3.1.

Warning: Passing '-a' as name is deprecated and will be removed in Sodium.

CLI Example:

```
salt '*' zfs.unmount myzpool/mydataset [force=True|False]
```

21.9.499 salt.modules.zk_concurrency

Concurrency controls in zookeeper

depends kazoo

configuration See *`salt.modules.zookeeper`* for setup instructions.

This module allows you to acquire and release a slot. This is primarily useful for ensureing that no more than N hosts take a specific action at once. This can also be used to coordinate between masters.

`salt.modules.zk_concurrency.`**`lock`**`(`*path*, *zk_hosts=None*, *identifier=None*, *max_concurrency=1*, *timeout=None*, *ephemeral_lease=False*, *force=False*, *profile=None*, *scheme=None*, *username=None*, *password=None*, *default_acl=None*`)`

Get lock (with optional timeout)

path The path in zookeeper where the lock is

zk_hosts zookeeper connect string

identifier Name to identify this minion, if unspecified defaults to the hostname

max_concurrency Maximum number of lock holders

timeout timeout to wait for the lock. A None timeout will block forever

ephemeral_lease Whether the locks in zookeper should be ephemeral

force Forcibly acquire the lock regardless of available slots

Example:

```
salt minion zk_concurrency.lock /lock/path host1:1234,host2:1234
```

`salt.modules.zk_concurrency.`**`lock_holders`**(*path, zk_hosts=None, identifier=None, max_concurrency=1, timeout=None, ephemeral_lease=False, profile=None, scheme=None, username=None, password=None, default_acl=None*)

Return an un-ordered list of lock holders

path The path in zookeeper where the lock is

zk_hosts zookeeper connect string

identifier Name to identify this minion, if unspecified defaults to hostname

max_concurrency Maximum number of lock holders

timeout timeout to wait for the lock. A None timeout will block forever

ephemeral_lease Whether the locks in zookeper should be ephemeral

Example:

```
salt minion zk_concurrency.lock_holders /lock/path host1:1234,host2:1234
```

`salt.modules.zk_concurrency.`**`party_members`**(*path, zk_hosts=None, min_nodes=1, blocking=False, profile=None, scheme=None, username=None, password=None, default_acl=None*)

Get the List of identifiers in a particular party, optionally waiting for the specified minimum number of nodes (min_nodes) to appear

path The path in zookeeper where the lock is

zk_hosts zookeeper connect string

min_nodes The minimum number of nodes expected to be present in the party

blocking The boolean indicating if we need to block until min_nodes are available

Example:

```
salt minion zk_concurrency.party_members /lock/path host1:1234,host2:1234
salt minion zk_concurrency.party_members /lock/path host1:1234,host2:1234 min_
↪nodes=3 blocking=True
```

`salt.modules.zk_concurrency.`**`unlock`**(*path, zk_hosts=None, identifier=None, max_concurrency=1, ephemeral_lease=False, scheme=None, profile=None, username=None, password=None, default_acl=None*)

Remove lease from semaphore

path The path in zookeeper where the lock is

zk_hosts zookeeper connect string

identifier Name to identify this minion, if unspecified defaults to hostname

max_concurrency Maximum number of lock holders

timeout timeout to wait for the lock. A None timeout will block forever

ephemeral_lease Whether the locks in zookeper should be ephemeral

Example:

```
salt minion zk_concurrency.unlock /lock/path host1:1234,host2:1234
```

21.9.500 salt.modules.znc

znc - An advanced IRC bouncer

New in version 2014.7.0.

Provides an interface to basic ZNC functionality

`salt.modules.znc.`**`buildmod`**`(`**modules*`)`

Build module using znc-buildmod

CLI Example:

```
salt '*' znc.buildmod module.cpp [...]
```

`salt.modules.znc.`**`dumpconf`**`()`

Write the active configuration state to config file

CLI Example:

```
salt '*' znc.dumpconf
```

`salt.modules.znc.`**`rehashconf`**`()`

Rehash the active configuration state from config file

CLI Example:

```
salt '*' znc.rehashconf
```

`salt.modules.znc.`**`version`**`()`

Return server version from znc –version

CLI Example:

```
salt '*' znc.version
```

21.9.501 salt.modules.zoneadm

Module for Solaris 10's zoneadm

maintainer Jorge Schrauwen <sjorge@blackdot.be>

maturity new

platform OmniOS,OpenIndiana,SmartOS,OpenSolaris,Solaris 10

New in version 2017.7.0.

Warning: Oracle Solaris 11's zoneadm is not supported by this module!

`salt.modules.zoneadm.`**`attach`**`(`*zone*, *force=False*, *brand_opts=None*`)`

Attach the specified zone.
zone [string] name of the zone
force [boolean] force the zone into the "installed" state with no validation

brand_opts [string] brand specific options to pass

CLI Example:

```
salt '*' zoneadm.attach lawrence
salt '*' zoneadm.attach lawrence True
```

`salt.modules.zoneadm.`**`boot`**`(`*zone*, *single=False*, *altinit=None*, *smf_options=None*`)`

Boot (or activate) the specified zone.

zone [string] name or uuid of the zone

single [boolean] boots only to milestone svc:/milestone/single-user:default.

altinit [string] valid path to an alternative executable to be the primordial process.

smf_options [string] include two categories of options to control booting behavior of the service management facility: recovery options and messages options.

CLI Example:

```
salt '*' zoneadm.boot clementine
salt '*' zoneadm.boot maeve single=True
salt '*' zoneadm.boot teddy single=True smf_options=verbose
```

`salt.modules.zoneadm.`**`clone`**`(`*zone*, *source*, *snapshot=None*`)`

Install a zone by copying an existing installed zone.

zone [string] name of the zone

source [string] zone to clone from

snapshot [string] optional name of snapshot to use as source

CLI Example:

```
salt '*' zoneadm.clone clementine dolores
```

`salt.modules.zoneadm.`**`detach`**`(`*zone*`)`

Detach the specified zone.

zone [string] name or uuid of the zone

CLI Example:

```
salt '*' zoneadm.detach kissy
```

`salt.modules.zoneadm.`**`halt`**`(`*zone*`)`

Halt the specified zone.

zone [string] name or uuid of the zone

Note: To cleanly shutdown the zone use the shutdown function.

CLI Example:

```
salt '*' zoneadm.halt hector
```

`salt.modules.zoneadm.`**`install`**`(`*zone*, *nodataset=False*, *brand_opts=None*`)`

Install the specified zone from the system.

zone [string] name of the zone

nodataset [boolean] do not create a ZFS file system

brand_opts [string] brand specific options to pass

CLI Example:

```
salt '*' zoneadm.install dolores
salt '*' zoneadm.install teddy True
```

`salt.modules.zoneadm.`**`list_zones`**`(`*verbose=True, installed=False, configured=False, hide_global=True*`)`

List all zones

verbose [boolean] display additional zone information

installed [boolean] include installed zones in output

configured [boolean] include configured zones in output

hide_global [boolean] do not include global zone

CLI Example:

```
salt '*' zoneadm.list
```

`salt.modules.zoneadm.`**`move`**`(`*zone, zonepath*`)`

Move zone to new zonepath.

zone [string] name or uuid of the zone

zonepath [string] new zonepath

CLI Example:

```
salt '*' zoneadm.move meave /sweetwater/meave
```

`salt.modules.zoneadm.`**`ready`**`(`*zone*`)`

Prepares a zone for running applications.

zone [string] name or uuid of the zone

CLI Example:

```
salt '*' zoneadm.ready clementine
```

`salt.modules.zoneadm.`**`reboot`**`(`*zone, single=False, altinit=None, smf_options=None*`)`

Restart the zone. This is equivalent to a halt boot sequence.

zone [string] name or uuid of the zone

single [boolean] boots only to milestone svc:/milestone/single-user:default.

altinit [string] valid path to an alternative executable to be the primordial process.

smf_options [string] include two categories of options to control booting behavior of the service management facility: recovery options and messages options.

CLI Example:

```
salt '*' zoneadm.reboot dolores
salt '*' zoneadm.reboot teddy single=True
```

`salt.modules.zoneadm.`**`shutdown`**`(`*zone, reboot=False, single=False, altinit=None, smf_options=None*`)`

Gracefully shutdown the specified zone.

zone [string] name or uuid of the zone

reboot [boolean] reboot zone after shutdown (equivalent of shutdown -i6 -g0 -y)

single [boolean] boots only to milestone svc:/milestone/single-user:default.

altinit [string] valid path to an alternative executable to be the primordial process.

smf_options [string] include two categories of options to control booting behavior of the service management facility: recovery options and messages options.

CLI Example:

```
salt '*' zoneadm.shutdown peter
salt '*' zoneadm.shutdown armistice reboot=True
```

`salt.modules.zoneadm.`**`uninstall`**`(`*zone*`)`

Uninstall the specified zone from the system.

zone [string] name or uuid of the zone

> **Warning:** The -F flag is always used to avoid the prompts when uninstalling.

CLI Example:

```
salt '*' zoneadm.uninstall teddy
```

`salt.modules.zoneadm.`**`verify`**`(`*zone*`)`

Check to make sure the configuration of the specified zone can safely be installed on the machine.

zone [string] name of the zone

CLI Example:

```
salt '*' zoneadm.verify dolores
```

21.9.502 salt.modules.zonecfg

Module for Solaris 10's zonecfg

maintainer Jorge Schrauwen <sjorge@blackdot.be>

maturity new

platform OmniOS,OpenIndiana,SmartOS,OpenSolaris,Solaris 10

depend salt.modules.file

New in version 2017.7.0.

> **Warning:** Oracle Solaris 11's zonecfg is not supported by this module!

`salt.modules.zonecfg.`**`add_resource`**`(`*zone, resource_type, **kwargs*`)`

Add a resource

zone [string] name of zone

resource_type [string] type of resource

kwargs [string|int|...] resource properties

CLI Example:

```
salt '*' zonecfg.add_resource tallgeese rctl name=zone.max-locked-memory value=
↪'(priv=privileged,limit=33554432,action=deny)'
```

`salt.modules.zonecfg.`**`clear_property`**`(`*zone, key*`)`

Clear a property

zone [string] name of zone

key [string] name of property

CLI Example:

```
salt '*' zonecfg.clear_property deathscythe cpu-shares
```

`salt.modules.zonecfg.`**`create`**`(`*zone, brand, zonepath, force=False*`)`

Create an in-memory configuration for the specified zone.

zone [string] name of zone

brand [string] brand name

zonepath [string] path of zone

force [boolean] overwrite configuration

CLI Example:

```
salt '*' zonecfg.create deathscythe ipkg /zones/deathscythe
```

`salt.modules.zonecfg.`**`create_from_template`**`(`*zone*, *template*`)`

Create an in-memory configuration from a template for the specified zone.

zone [string] name of zone

template [string] name of template

Warning: existing config will be overwritten!

CLI Example:

```
salt '*' zonecfg.create_from_template leo tallgeese
```

`salt.modules.zonecfg.`**`delete`**`(`*zone*`)`

Delete the specified configuration from memory and stable storage.

zone [string] name of zone

CLI Example:

```
salt '*' zonecfg.delete epyon
```

`salt.modules.zonecfg.`**`export`**`(`*zone*, *path=None*`)`

Export the configuration from memory to stable storage.

zone [string] name of zone

path [string] path of file to export to

CLI Example:

```
salt '*' zonecfg.export epyon
salt '*' zonecfg.export epyon /zones/epyon.cfg
```

`salt.modules.zonecfg.`**`import_`**`(`*zone*, *path*`)`

Import the configuration to memory from stable storage.

zone [string] name of zone

path [string] path of file to export to

CLI Example:

```
salt '*' zonecfg.import epyon /zones/epyon.cfg
```

`salt.modules.zonecfg.`**`info`**`(`*zone*, *show_all=False*`)`

Display the configuration from memory

zone [string] name of zone

show_all [boolean] also include calculated values like capped-cpu, cpu-shares, …

CLI Example:

```
salt '*' zonecfg.info tallgeese
```

`salt.modules.zonecfg.`**`remove_resource`**`(`*zone*, *resource_type*, *resource_key*, *resource_value*`)`

Remove a resource

zone [string] name of zone

resource_type [string] type of resource

resource_key [string] key for resource selection

resource_value [string] value for resource selection

Note: Set resource_selector to None for resource that do not require one.

CLI Example:

```
salt '*' zonecfg.remove_resource tallgeese rctl name zone.max-locked-memory
```

`salt.modules.zonecfg.`**`set_property`**(*zone*, *key*, *value*)

Set a property

zone [string] name of zone

key [string] name of property

value [string] value of property

CLI Example:

```
salt '*' zonecfg.set_property deathscythe cpu-shares 100
```

`salt.modules.zonecfg.`**`update_resource`**(*zone*, *resource_type*, *resource_selector*, ***kwargs*)

Add a resource

zone [string] name of zone

resource_type [string] type of resource

resource_selector [string] unique resource identifier

kwargs [string|int|...] resource properties

Note: Set resource_selector to None for resource that do not require one.

CLI Example:

```
salt '*' zonecfg.update_resource tallgeese rctl name name=zone.max-locked-memory␣
↪value='(priv=privileged,limit=33554432,action=deny)'
```

21.9.503 salt.modules.zookeeper

Zookeeper Module

maintainer SaltStack

maturity new

platform all

depends kazoo

New in version 2018.3.0.

Configuration

configuration This module is not usable until the following are specified either in a pillar or in the minion's config file:

```
zookeeper:
  hosts: zoo1,zoo2,zoo3
  default_acl:
    - username: daniel
      password: test
      read: true
      write: true
      create: true
```

(continues on next page)

(continued from previous page)

```
      delete: true
      admin: true
  username: daniel
  password: test
```

If configuration for multiple zookeeper environments is required, they can be set up as different configuration profiles. For example:

```
zookeeper:
  prod:
    hosts: zoo1,zoo2,zoo3
    default_acl:
      - username: daniel
        password: test
        read: true
        write: true
        create: true
        delete: true
        admin: true
    username: daniel
    password: test
  dev:
    hosts:
      - dev1
      - dev2
      - dev3
    default_acl:
      - username: daniel
        password: test
        read: true
        write: true
        create: true
        delete: true
        admin: true
    username: daniel
    password: test
```

`salt.modules.zookeeper.`**`create`**(*path, value='', acls=None, ephemeral=False, sequence=False, makepath=False, profile=None, hosts=None, scheme=None, username=None, password=None, default_acl=None*)

Create Znode

path path of znode to create

value value to assign to znode (Default: '')

acls list of acl dictionaries to be assigned (Default: None)

ephemeral indicate node is ephemeral (Default: False)

sequence indicate node is suffixed with a unique index (Default: False)

makepath Create parent paths if they do not exist (Default: False)

profile Configured Zookeeper profile to authenticate with (Default: None)

hosts Lists of Zookeeper Hosts (Default: '127.0.0.1:2181)

scheme Scheme to authenticate with (Default: 'digest')

username Username to authenticate (Default: None)

password Password to authenticate (Default: None)

default_acl Default acls to assign if a node is created in this connection (Default: None)

CLI Example:

```
salt minion1 zookeeper.create /test/name daniel profile=prod
```

`salt.modules.zookeeper.`**`delete`**(*path, version=-1, recursive=False, profile=None, hosts=None, scheme=None, username=None, password=None, default_acl=None*)

Delete znode

path path to znode

version only delete if version matches (Default: -1 (always matches))

profile Configured Zookeeper profile to authenticate with (Default: None)

hosts Lists of Zookeeper Hosts (Default: '127.0.0.1:2181)

scheme Scheme to authenticate with (Default: 'digest')

username Username to authenticate (Default: None)

password Password to authenticate (Default: None)

default_acl Default acls to assign if a node is created in this connection (Default: None)

CLI Example:

```
salt minion1 zookeeper.delete /test/name profile=prod
```

`salt.modules.zookeeper.`**`ensure_path`**(*path, acls=None, profile=None, hosts=None, scheme=None, username=None, password=None, default_acl=None*)

Ensure Znode path exists

path Parent path to create

acls list of acls dictionaries to be assigned (Default: None)

profile Configured Zookeeper profile to authenticate with (Default: None)

hosts Lists of Zookeeper Hosts (Default: '127.0.0.1:2181)

scheme Scheme to authenticate with (Default: 'digest')

username Username to authenticate (Default: None)

password Password to authenticate (Default: None)

default_acl Default acls to assign if a node is created in this connection (Default: None)

CLI Example:

```
salt minion1 zookeeper.ensure_path /test/name profile=prod
```

`salt.modules.zookeeper.`**`exists`**(*path, profile=None, hosts=None, scheme=None, username=None, password=None, default_acl=None*)

Check if path exists

path path to check

profile Configured Zookeeper profile to authenticate with (Default: None)

hosts Lists of Zookeeper Hosts (Default: '127.0.0.1:2181)

scheme Scheme to authenticate with (Default: 'digest')

username Username to authenticate (Default: None)

password Password to authenticate (Default: None)

default_acl Default acls to assign if a node is created in this connection (Default: None)

CLI Example:

```
salt minion1 zookeeper.exists /test/name profile=prod
```

`salt.modules.zookeeper.`**`get`**(*path, profile=None, hosts=None, scheme=None, username=None, password=None, default_acl=None*)

Get value saved in znode

path path to check

profile Configured Zookeeper profile to authenticate with (Default: None)

hosts Lists of Zookeeper Hosts (Default: '127.0.0.1:2181)

scheme Scheme to authenticate with (Default: 'digest')

username Username to authenticate (Default: None)

password Password to authenticate (Default: None)
default_acl Default acls to assign if a node is created in this connection (Default: None)
CLI Example:

```
salt minion1 zookeeper.get /test/name profile=prod
```

`salt.modules.zookeeper.`**`get_acls`**(*path*, *profile=None*, *hosts=None*, *scheme=None*, *username=None*, *password=None*, *default_acl=None*)

Get acls on a znode
path path to znode
profile Configured Zookeeper profile to authenticate with (Default: None)
hosts Lists of Zookeeper Hosts (Default: '127.0.0.1:2181)
scheme Scheme to authenticate with (Default: 'digest')
username Username to authenticate (Default: None)
password Password to authenticate (Default: None)
default_acl Default acls to assign if a node is created in this connection (Default: None)
CLI Example:

```
salt minion1 zookeeper.get_acls /test/name profile=prod
```

`salt.modules.zookeeper.`**`get_children`**(*path*, *profile=None*, *hosts=None*, *scheme=None*, *username=None*, *password=None*, *default_acl=None*)

Get children in znode path
path path to check
profile Configured Zookeeper profile to authenticate with (Default: None)
hosts Lists of Zookeeper Hosts (Default: '127.0.0.1:2181)
scheme Scheme to authenticate with (Default: 'digest')
username Username to authenticate (Default: None)
password Password to authenticate (Default: None)
default_acl Default acls to assign if a node is created in this connection (Default: None)
CLI Example:

```
salt minion1 zookeeper.get_children /test profile=prod
```

`salt.modules.zookeeper.`**`make_digest_acl`**(*username*, *password*, *read=False*, *write=False*, *create=False*, *delete=False*, *admin=False*, *allperms=False*)

Generate acl object

Note: This is heavily used in the zookeeper state and probably is not useful as a cli module

username username of acl
password plain text password of acl
read read acl
write write acl
create create acl
delete delete acl
admin admin acl
allperms set all other acls to True

CLI Example:

```
salt minion1 zookeeper.make_digest_acl username=daniel password=mypass
↪allperms=True
```

`salt.modules.zookeeper.`**`set`**(*path, value, version=-1, profile=None, hosts=None, scheme=None, username=None, password=None, default_acl=None*)

Update znode with new value

path znode to update

value value to set in znode

version only update znode if version matches (Default: -1 (always matches))

profile Configured Zookeeper profile to authenticate with (Default: None)

hosts Lists of Zookeeper Hosts (Default: '127.0.0.1:2181)

scheme Scheme to authenticate with (Default: 'digest')

username Username to authenticate (Default: None)

password Password to authenticate (Default: None)

default_acl Default acls to assign if a node is created in this connection (Default: None)

CLI Example:

```
salt minion1 zookeeper.set /test/name gtmanfred profile=prod
```

`salt.modules.zookeeper.`**`set_acls`**(*path, acls, version=-1, profile=None, hosts=None, scheme=None, username=None, password=None, default_acl=None*)

Set acls on a znode

path path to znode

acls list of acl dictionaries to set on the znode

version only set acls if version matches (Default: -1 (always matches))

profile Configured Zookeeper profile to authenticate with (Default: None)

hosts Lists of Zookeeper Hosts (Default: '127.0.0.1:2181)

scheme Scheme to authenticate with (Default: 'digest')

username Username to authenticate (Default: None)

password Password to authenticate (Default: None)

default_acl Default acls to assign if a node is created in this connection (Default: None)

CLI Example:

```
salt minion1 zookeeper.set_acls /test/name acls='[{"username": "gtmanfred",
↪"password": "test", "all": True}]' profile=prod
```

21.9.504 salt.modules.zpool

Module for running ZFS zpool command

codeauthor Nitin Madhok <nmadhok@clemson.edu>, Jorge Schrauwen <sjorge@blackdot.be>

maintainer Jorge Schrauwen <sjorge@blackdot.be>

maturity new

depends salt.utils.zfs

platform illumos,freebsd,linux

Changed in version 2018.3.1: Big refactor to remove duplicate code, better type conversions and improved consistency in output.

`salt.modules.zpool.`**`add`**(*zpool, *vdevs, **kwargs*)

Add the specified vdev's to the given storage pool

zpool [string] Name of storage pool

vdevs [string] One or more devices

force [boolean] Forces use of device

CLI Example:

```
salt '*' zpool.add myzpool /path/to/vdev1 /path/to/vdev2 [...]
```

`salt.modules.zpool.`**`attach`**(*zpool*, *device*, *new_device*, *force=False*)

Attach specified device to zpool

zpool [string] Name of storage pool

device [string] Existing device name too

new_device [string] New device name (to be attached to `device`)

force [boolean] Forces use of device

CLI Example:

```
salt '*' zpool.attach myzpool /path/to/vdev1 /path/to/vdev2 [...]
```

`salt.modules.zpool.`**`clear`**(*zpool*, *device=None*)

Clears device errors in a pool.

> **Warning:** The device must not be part of an active pool configuration.

zpool [string] name of storage pool

device [string] (optional) specific device to clear

New in version 2018.3.1.

CLI Example:

```
salt '*' zpool.clear mypool
salt '*' zpool.clear mypool /path/to/dev
```

`salt.modules.zpool.`**`create`**(*zpool*, **vdevs*, ***kwargs*)

New in version 2015.5.0.

Create a simple zpool, a mirrored zpool, a zpool having nested VDEVs, a hybrid zpool with cache, spare and log drives or a zpool with RAIDZ-1, RAIDZ-2 or RAIDZ-3

zpool [string] Name of storage pool

vdevs [string] One or move devices

force [boolean] Forces use of vdevs, even if they appear in use or specify a conflicting replication level.

mountpoint [string] Sets the mount point for the root dataset

altroot [string] Equivalent to "-o cachefile=none,altroot=root"

properties [dict] Additional pool properties

filesystem_properties [dict] Additional filesystem properties

createboot [boolean] create a boot partition

New in version 2018.3.0.

CLI Examples:

```
salt '*' zpool.create myzpool /path/to/vdev1 [...] [force=True|False]
salt '*' zpool.create myzpool mirror /path/to/vdev1 /path/to/vdev2 [...]
↪[force=True|False]
salt '*' zpool.create myzpool raidz1 /path/to/vdev1 /path/to/vdev2 raidz2 /path/
↪to/vdev3 /path/to/vdev4 /path/to/vdev5 [...] [force=True|False]
salt '*' zpool.create myzpool mirror /path/to/vdev1 [...] mirror /path/to/vdev2 /
↪path/to/vdev3 [...] [force=True|False]
salt '*' zpool.create myhybridzpool mirror /tmp/file1 [...] log mirror /path/to/
↪vdev1 [...] cache /path/to/vdev2 [...] spare /path/to/vdev3 [...]
↪[force=True|False]
```

Note: Zpool properties can be specified at the time of creation of the pool by passing an additional argument called "properties" and specifying the properties with their respective values in the form of a python dictionary:

```
properties="{'property1': 'value1', 'property2': 'value2'}"
```

Filesystem properties can be specified at the time of creation of the pool by passing an additional argument called "filesystem_properties" and specifying the properties with their respective values in the form of a python dictionary:

```
filesystem_properties="{'property1': 'value1', 'property2': 'value2'}"
```

Example:

```
salt '*' zpool.create myzpool /path/to/vdev1 [...] properties="{'property1':
↪'value1', 'property2': 'value2'}"
```

CLI Example:

```
salt '*' zpool.create myzpool /path/to/vdev1 [...] [force=True|False]
salt '*' zpool.create myzpool mirror /path/to/vdev1 /path/to/vdev2 [...]
↪[force=True|False]
salt '*' zpool.create myzpool raidz1 /path/to/vdev1 /path/to/vdev2 raidz2 /path/
↪to/vdev3 /path/to/vdev4 /path/to/vdev5 [...] [force=True|False]
salt '*' zpool.create myzpool mirror /path/to/vdev1 [...] mirror /path/to/vdev2 /
↪path/to/vdev3 [...] [force=True|False]
salt '*' zpool.create myhybridzpool mirror /tmp/file1 [...] log mirror /path/to/
↪vdev1 [...] cache /path/to/vdev2 [...] spare /path/to/vdev3 [...]
↪[force=True|False]
```

`salt.modules.zpool.`**`create_file_vdev`**`(`*size, *vdevs*`)`

Creates file based virtual devices for a zpool

CLI Example:

```
salt '*' zpool.create_file_vdev 7G /path/to/vdev1 [/path/to/vdev2] [...]
```

Note: Depending on file size, the above command may take a while to return.

`salt.modules.zpool.`**`destroy`**`(`*zpool, force=False*`)`

Destroys a storage pool

zpool [string] Name of storage pool

force [boolean] Force destroy of pool

CLI Example:

```
salt '*' zpool.destroy myzpool
```

`salt.modules.zpool.`**`detach`**`(`*zpool, device*`)`

Detach specified device to zpool

zpool [string] Name of storage pool

device [string] Device to detach

CLI Example:

```
salt '*' zpool.detach myzpool /path/to/vdev1
```

`salt.modules.zpool.`**`exists`**`(`*zpool*`)`

Check if a ZFS storage pool is active

zpool [string] Name of storage pool

CLI Example:

```
salt '*' zpool.exists myzpool
```

`salt.modules.zpool.`**`export`**`(`**pools, **kwargs*`)`

New in version 2015.5.0.

Export storage pools

pools [string] One or more storage pools to export

force [boolean] Force export of storage pools

CLI Example:

```
salt '*' zpool.export myzpool ... [force=True|False]
salt '*' zpool.export myzpool2 myzpool2 ... [force=True|False]
```

`salt.modules.zpool.`**`get`**`(`*zpool, prop=None, show_source=False, parsable=True*`)`

New in version 2016.3.0.

Retrieves the given list of properties

zpool [string] Name of storage pool

prop [string] Optional name of property to retrieve

show_source [boolean] Show source of property

parsable [boolean] Display numbers in parsable (exact) values

New in version 2018.3.0.

CLI Example:

```
salt '*' zpool.get myzpool
```

`salt.modules.zpool.`**`healthy`**`()`

Check if all zpools are healthy

New in version 2016.3.0.

CLI Example:

```
salt '*' zpool.healthy
```

`salt.modules.zpool.`**`history`**`(`*zpool=None, internal=False, verbose=False*`)`

New in version 2016.3.0.

Displays the command history of the specified pools, or all pools if no pool is specified

zpool [string] Optional storage pool

internal [boolean] Toggle display of internally logged ZFS events

verbose [boolean] Toggle display of the user name, the hostname, and the zone in which the operation was performed

CLI Example:

```
salt '*' zpool.upgrade myzpool
```

`salt.modules.zpool.`**`import_`**`(`*zpool=None, new_name=None, **kwargs*`)`

New in version 2015.5.0.

Import storage pools or list pools available for import

zpool [string] Optional name of storage pool

new_name [string] Optional new name for the storage pool

mntopts [string] Comma-separated list of mount options to use when mounting datasets within the pool.
force [boolean] Forces import, even if the pool appears to be potentially active.
altroot [string] Equivalent to "-o cachefile=none,altroot=root"
dir [string] Searches for devices or files in dir, multiple dirs can be specified as follows: `dir="dir1,dir2"`
no_mount [boolean] Import the pool without mounting any file systems.
only_destroyed [boolean] Imports destroyed pools only. This also sets `force=True`.
recovery [bool|str] false: do not try to recovery broken pools true: try to recovery the pool by rolling back the latest transactions test: check if a pool can be recovered, but don't import it nolog: allow import without log device, recent transactions might be lost

Note: If feature flags are not support this forced to the default of 'false'

Warning: When recovery is set to 'test' the result will be have imported set to True if the pool can be imported. The pool might also be imported if the pool was not broken to begin with.

properties [dict] Additional pool properties

Note: Zpool properties can be specified at the time of creation of the pool by passing an additional argument called "properties" and specifying the properties with their respective values in the form of a python dictionary:

```
properties="{'property1': 'value1', 'property2': 'value2'}"
```

CLI Example:

```
salt '*' zpool.import [force=True|False]
salt '*' zpool.import myzpool [mynewzpool] [force=True|False]
salt '*' zpool.import myzpool dir='/tmp'
```

`salt.modules.zpool.`**`iostat`**(*zpool=None*, *sample_time=5*, *parsable=True*)

Display I/O statistics for the given pools
zpool [string] optional name of storage pool
sample_time [int] seconds to capture data before output default a sample of 5 seconds is used
parsable [boolean] display data in pythonc values (True, False, Bytes,...)
New in version 2016.3.0.

Changed in version 2018.3.1: Added `` `parsable` `` parameter that defaults to True

CLI Example:

```
salt '*' zpool.iostat myzpool
```

`salt.modules.zpool.`**`labelclear`**(*device*, *force=False*)

New in version 2018.3.0.

Removes ZFS label information from the specified device
device [string] Device name; must not be part of an active pool configuration.
force [boolean] Treat exported or foreign devices as inactive
CLI Example:

```
salt '*' zpool.labelclear /path/to/dev
```

`salt.modules.zpool.`**`list_`**(*properties='size, alloc, free, cap, frag, health'*, *zpool=None*, *parsable=True*)

New in version 2015.5.0.

Return information about (all) storage pools
zpool [string] optional name of storage pool
properties [string] comma-separated list of properties to list
parsable [boolean] display numbers in parsable (exact) values

New in version 2018.3.0.

Note: The `name` property will always be included, while the `frag` property will get removed if not available

zpool [string] optional zpool

Note: Multiple storage pool can be provided as a space separated list

CLI Example:

```
salt '*' zpool.list
salt '*' zpool.list zpool=tank
salt '*' zpool.list 'size,free'
salt '*' zpool.list 'size,free' tank
```

`salt.modules.zpool.`**`offline`**(*zpool*, **vdevs*, ***kwargs*)
New in version 2015.5.0.

Ensure that the specified devices are offline

Warning: By default, the `OFFLINE` state is persistent. The device remains offline when the system is rebooted. To temporarily take a device offline, use `temporary=True`.

zpool [string] name of storage pool
vdevs [string] One or more devices
temporary [boolean] Enable temporarily offline

CLI Example:

```
salt '*' zpool.offline myzpool /path/to/vdev1 [...] [temporary=True|False]
```

`salt.modules.zpool.`**`online`**(*zpool*, **vdevs*, ***kwargs*)
New in version 2015.5.0.

Ensure that the specified devices are online
zpool [string] name of storage pool
vdevs [string] one or more devices
expand [boolean] Expand the device to use all available space.

Note: If the device is part of a mirror or raidz then all devices must be expanded before the new space will become available to the pool.

CLI Example:

```
salt '*' zpool.online myzpool /path/to/vdev1 [...]
```

`salt.modules.zpool.`**`reguid`**(*zpool*)
Generates a new unique identifier for the pool

Warning: You must ensure that all devices in this pool are online and healthy before performing this action.

zpool [string] name of storage pool

New in version 2016.3.0.

CLI Example:

```
salt '*' zpool.reguid myzpool
```

`salt.modules.zpool.`**`reopen`**(*zpool*)

Reopen all the vdevs associated with the pool
zpool [string] name of storage pool
New in version 2016.3.0.

CLI Example:

```
salt '*' zpool.reopen myzpool
```

`salt.modules.zpool.`**`replace`**(*zpool, old_device, new_device=None, force=False*)

Replaces `old_device` with `new_device`

Note: This is equivalent to attaching `new_device`, waiting for it to resilver, and then detaching `old_device`.

The size of `new_device` must be greater than or equal to the minimum size of all the devices in a mirror or raidz configuration.

---

zpool [string] Name of storage pool
old_device [string] Old device to replace
new_device [string] Optional new device
force [boolean] Forces use of new_device, even if its appears to be in use.

CLI Example:

```
salt '*' zpool.replace myzpool /path/to/vdev1 /path/to/vdev2
```

`salt.modules.zpool.`**`scrub`**(*zpool, stop=False, pause=False*)

Scrub a storage pool
zpool [string] Name of storage pool
stop [boolean] If `True`, cancel ongoing scrub
pause [boolean] If `True`, pause ongoing scrub

New in version 2018.3.0.

Note: Pause is only available on recent versions of ZFS.

If both `pause` and `stop` are `True`, then `stop` will win.

CLI Example:

```
salt '*' zpool.scrub myzpool
```

`salt.modules.zpool.`**`set`**(*zpool, prop, value*)

Sets the given property on the specified pool

zpool [string] Name of storage pool
prop [string] Name of property to set
value [string] Value to set for the specified property
New in version 2016.3.0.

CLI Example:

```
salt '*' zpool.set myzpool readonly yes
```

`salt.modules.zpool.`**`split`**`(`*zpool, newzpool, **kwargs*`)`
New in version 2018.3.0.

Splits devices off pool creating newpool.

Note: All vdevs in pool must be mirrors. At the time of the split, `newzpool` will be a replica of `zpool`.

After splitting, do not forget to import the new pool!

---

zpool [string] Name of storage pool
newzpool [string] Name of new storage pool
mountpoint [string] Sets the mount point for the root dataset
altroot [string] Sets altroot for newzpool
properties [dict] Additional pool properties for newzpool

CLI Examples:

```
salt '*' zpool.split datamirror databackup
salt '*' zpool.split datamirror databackup altroot=/backup
```

Note: Zpool properties can be specified at the time of creation of the pool by passing an additional argument called ”properties” and specifying the properties with their respective values in the form of a python dictionary:

```
properties="{'property1': 'value1', 'property2': 'value2'}"
```

Example:

```
salt '*' zpool.split datamirror databackup properties="{'readonly': 'on'}"
```

CLI Example:

```
salt '*' zpool.split datamirror databackup
salt '*' zpool.split datamirror databackup altroot=/backup
```

`salt.modules.zpool.`**`status`**`(`*zpool=None*`)`
Return the status of the named zpool
zpool [string] optional name of storage pool
New in version 2016.3.0.

CLI Example:

```
salt '*' zpool.status myzpool
```

`salt.modules.zpool.`**`upgrade`**`(`*zpool=None, version=None*`)`
New in version 2016.3.0.

Enables all supported features on the given pool

zpool [string] Optional storage pool, applies to all otherwize
version [int] Version to upgrade to, if unspecified upgrade to the highest possible

> **Warning:** Once this is done, the pool will no longer be accessible on systems that do not support feature flags. See zpool-features(5) for details on compatibility with systems that support feature flags, but do not support all features enabled on the pool.

CLI Example:

```
salt '*' zpool.upgrade myzpool
```

21.9.505 salt.modules.zypperpkg

Package support for openSUSE via the zypper package manager

depends

- `rpm` Python module. Install with `zypper install rpm-python`

Important: If you feel that Salt should be using this module to manage packages on a minion, and it is using a different module (or gives an error similar to *'pkg.install' is not available*), see *here*.

---

class `salt.modules.zypperpkg.`**`Wildcard`**(*zypper*)

New in version 2017.7.0.

Converts string wildcard to a zypper query. .. rubric:: Example

'1.2.3.4*' is '1.2.3.4.whatever.is.here' and is equal to: '1.2.3.4 >= and < 1.2.3.5'

Parameters **ptn** – Pattern

Returns Query range

`Z_OP = ['<', '<=', '=', '>=', '>']`

`salt.modules.zypperpkg.`**`add_lock`**(*packages*, ***kwargs*)

Add a package lock. Specify packages to lock by exact name.

CLI Example:

```
salt '*' pkg.add_lock <package name>
salt '*' pkg.add_lock <package1>,<package2>,<package3>
salt '*' pkg.add_lock pkgs='["foo", "bar"]'
```

`salt.modules.zypperpkg.`**`clean_locks`**()

Remove unused locks that do not currently (with regard to repositories used) lock any package.

CLI Example:

```
salt '*' pkg.clean_locks
```

`salt.modules.zypperpkg.`**`del_repo`**(*repo*)

Delete a repo.

CLI Examples:

```
salt '*' pkg.del_repo alias
```

`salt.modules.zypperpkg.`**`diff`**`(`**paths*`)`

Return a formatted diff between current files and original in a package. NOTE: this function includes all files (configuration and not), but does not work on binary content.

Parameters **`path`** – Full path to the installed file

Returns Difference string or raises and exception if examined file is binary.

CLI example:

```
salt '*' pkg.diff /etc/apache2/httpd.conf /etc/sudoers
```

`salt.modules.zypperpkg.`**`download`**`(`**packages, **kwargs*`)`

Download packages to the local disk.

refresh force a refresh if set to True. If set to False (default) it depends on zypper if a refresh is executed.

CLI example:

```
salt '*' pkg.download httpd
salt '*' pkg.download httpd postfix
```

`salt.modules.zypperpkg.`**`file_dict`**`(`**packages*`)`

List the files that belong to a package, grouped by package. Not specifying any packages will return a list of *every* file on the system's rpm database (not generally recommended).

CLI Examples:

```
salt '*' pkg.file_list httpd
salt '*' pkg.file_list httpd postfix
salt '*' pkg.file_list
```

`salt.modules.zypperpkg.`**`file_list`**`(`**packages*`)`

List the files that belong to a package. Not specifying any packages will return a list of *every* file on the system's rpm database (not generally recommended).

CLI Examples:

```
salt '*' pkg.file_list httpd
salt '*' pkg.file_list httpd postfix
salt '*' pkg.file_list
```

`salt.modules.zypperpkg.`**`get_repo`**`(`*repo, **kwargs*`)`

Display a repo.

CLI Example:

```
salt '*' pkg.get_repo alias
```

`salt.modules.zypperpkg.`**`info_available`**`(`**names, **kwargs*`)`

Return the information of the named package available for the system.

refresh force a refresh if set to True (default). If set to False it depends on zypper if a refresh is executed or not.

CLI example:

```
salt '*' pkg.info_available <package1>
salt '*' pkg.info_available <package1> <package2> <package3> ...
```

`salt.modules.zypperpkg.`**`info_installed`**`(`**names, **kwargs*`)`

Return the information of the named package(s), installed on the system.

Parameters

- **`names`** – Names of the packages to get information about.

- **`attr`** – Comma-separated package attributes. If no 'attr' is specified, all available attributes returned.

  Valid attributes are: version, vendor, release, build_date, build_date_time_t, install_date, install_date_time_t, build_host, group, source_rpm, arch, epoch, size, license, signature, packager, url, summary, description.
- **`errors`** – Handle RPM field errors. If 'ignore' is chosen, then various mistakes are simply ignored and omitted from the texts or strings. If 'report' is chonen, then a field with a mistake is not returned, instead a 'N/A (broken)' (not available, broken) text is placed.

  Valid attributes are: ignore, report
- **`all_versions`** – Include information for all versions of the packages installed on the minion.

CLI example:

```
salt '*' pkg.info_installed <package1>
salt '*' pkg.info_installed <package1> <package2> <package3> ...
salt '*' pkg.info_installed <package1> <package2> <package3> all_versions=True
salt '*' pkg.info_installed <package1> attr=version,vendor all_versions=True
salt '*' pkg.info_installed <package1> <package2> <package3> ... attr=version,
↪vendor
salt '*' pkg.info_installed <package1> <package2> <package3> ... attr=version,
↪vendor errors=ignore
salt '*' pkg.info_installed <package1> <package2> <package3> ... attr=version,
↪vendor errors=report
```

`salt.modules.zypperpkg.`**`install`**(*name=None, refresh=False, fromrepo=None, pkgs=None, sources=None, downloadonly=None, skip_verify=False, version=None, ignore_repo_failure=False, **kwargs*)

Changed in version 2015.8.12,2016.3.3,2016.11.0: On minions running systemd>=205, systemd-run(1) is now used to isolate commands which modify installed packages from the `salt-minion` daemon's control group. This is done to keep systemd from killing any zypper commands spawned by Salt when the `salt-minion` service is restarted. (see `KillMode` in the systemd.kill(5) manpage for more information). If desired, usage of systemd-run(1) can be suppressed by setting a *`config option`* called `systemd.scope`, with a value of `False` (no quotes).

Install the passed package(s), add refresh=True to force a 'zypper refresh' before package is installed.

name The name of the package to be installed. Note that this parameter is ignored if either `pkgs` or `sources` is passed. Additionally, please note that this option can only be used to install packages from a software repository. To install a package file manually, use the `sources` option.

CLI Example:

```
salt '*' pkg.install <package name>
```

refresh force a refresh if set to True. If set to False (default) it depends on zypper if a refresh is executed.

fromrepo Specify a package repository to install from.

downloadonly Only download the packages, do not install.

skip_verify Skip the GPG verification check (e.g., `--no-gpg-checks`)

version Can be either a version number, or the combination of a comparison operator (<, >, <=, >=, =) and a version number (ex. '>1.2.3-4'). This parameter is ignored if `pkgs` or `sources` is passed.

resolve_capabilities If this option is set to True zypper will take capabilities into account. In this case names which are just provided by a package will get installed. Default is False.

Multiple Package Installation Options:

pkgs A list of packages to install from a software repository. Must be passed as a python list. A specific version number can be specified by using a single-element dict representing the package and its version. As with the `version` parameter above, comparison operators can be used to target a specific version of a package.

CLI Examples:

```
salt '*' pkg.install pkgs='["foo", "bar"]'
salt '*' pkg.install pkgs='["foo", {"bar": "1.2.3-4"}]'
salt '*' pkg.install pkgs='["foo", {"bar": "<1.2.3-4"}]'
```

sources A list of RPM packages to install. Must be passed as a list of dicts, with the keys being package names, and the values being the source URI or local path to the package.

CLI Example:

```
salt '*' pkg.install sources='[{"foo": "salt://foo.rpm"},{"bar": "salt://bar.
↪rpm"}]'
```

ignore_repo_failure Zypper returns error code 106 if one of the repositories are not available for various reasons. In case to set strict check, this parameter needs to be set to True. Default: False.

diff_attr: If a list of package attributes is specified, returned value will contain them, eg.:

```
{'<package>': {
    'old': {
        'version': '<old-version>',
        'arch': '<old-arch>'},

    'new': {
        'version': '<new-version>',
        'arch': '<new-arch>'}}}
```

Valid attributes are: `epoch`, `version`, `release`, `arch`, `install_date`, `install_date_time_t`.

If `all` is specified, all valid attributes will be returned.

New in version 2018.3.0.

Returns a dict containing the new package names and versions:

```
{'<package>': {'old': '<old-version>',
               'new': '<new-version>'}}
```

If an attribute list is specified in `diff_attr`, the dict will also contain any specified attribute, eg.:

```
{'<package>': {
    'old': {
        'version': '<old-version>',
        'arch': '<old-arch>'},

    'new': {
        'version': '<new-version>',
        'arch': '<new-arch>'}}}
```

`salt.modules.zypperpkg.`**`latest_version`**(**names*, ***kwargs*)

Return the latest version of the named package available for upgrade or installation. If more than one package name is specified, a dict of name/version pairs is returned.

If the latest version of a given package is already installed, an empty dict will be returned for that package.

refresh force a refresh if set to True (default). If set to False it depends on zypper if a refresh is executed or not.

CLI example:

```
salt '*' pkg.latest_version <package name>
salt '*' pkg.latest_version <package1> <package2> <package3> ...
```

`salt.modules.zypperpkg.`**`list_downloaded`**`()`

New in version 2017.7.0.

List prefetched packages downloaded by Zypper in the local disk.

CLI example:

```
salt '*' pkg.list_downloaded
```

`salt.modules.zypperpkg.`**`list_installed_patches`**`()`

New in version 2017.7.0.

List installed advisory patches on the system.

CLI Examples:

```
salt '*' pkg.list_installed_patches
```

`salt.modules.zypperpkg.`**`list_installed_patterns`**`()`

List installed patterns on the system.

CLI Examples:

```
salt '*' pkg.list_installed_patterns
```

`salt.modules.zypperpkg.`**`list_locks`**`()`

List current package locks.

Return a dict containing the locked package with attributes:

```
{'<package>': {'case_sensitive': '<case_sensitive>',
               'match_type': '<match_type>'
               'type': '<type>'}}
```

CLI Example:

```
salt '*' pkg.list_locks
```

`salt.modules.zypperpkg.`**`list_patches`**`(`*refresh=False*`)`

New in version 2017.7.0.

List all known advisory patches from available repos.

refresh force a refresh if set to True. If set to False (default) it depends on zypper if a refresh is executed.

CLI Examples:

```
salt '*' pkg.list_patches
```

`salt.modules.zypperpkg.`**`list_patterns`**`(`*refresh=False*`)`

List all known patterns from available repos.

refresh force a refresh if set to True. If set to False (default) it depends on zypper if a refresh is executed.

CLI Examples:

```
salt '*' pkg.list_patterns
```

`salt.modules.zypperpkg.`**`list_pkgs`**`(`*versions_as_list=False, **kwargs*`)`

List the packages currently installed as a dict. By default, the dict contains versions as a comma separated string:

```
{'<package_name>': '<version>[,<version>...]'}
```

versions_as_list: If set to true, the versions are provided as a list

{'<package_name>': ['<version>', '<version>']}

attr: If a list of package attributes is specified, returned value will contain them in addition to version, eg.:

```
{'<package_name>': [{'version' : 'version', 'arch' : 'arch'}]}
```

Valid attributes are: `epoch`, `version`, `release`, `arch`, `install_date`, `install_date_time_t`.

If `all` is specified, all valid attributes will be returned.

New in version 2018.3.0.

removed: not supported

purge_desired: not supported

CLI Example:

```
salt '*' pkg.list_pkgs
salt '*' pkg.list_pkgs attr=version,arch
salt '*' pkg.list_pkgs attr='["version", "arch"]'
```

`salt.modules.zypperpkg.`**`list_products`**`(`*all=False*, *refresh=False*`)`

List all available or installed SUSE products.

all List all products available or only installed. Default is False.

refresh force a refresh if set to True. If set to False (default) it depends on zypper if a refresh is executed.

Includes handling for OEM products, which read the OEM productline file and overwrite the release value.

CLI Examples:

```
salt '*' pkg.list_products
salt '*' pkg.list_products all=True
```

`salt.modules.zypperpkg.`**`list_provides`**`(`***kwargs*`)`

New in version 2018.3.0.

List package provides of installed packages as a dict. {'<provided_name>': ['<package_name>', '<package_name>', ...]}

CLI Examples:

```
salt '*' pkg.list_provides
```

`salt.modules.zypperpkg.`**`list_repo_pkgs`**`(`**args*, ***kwargs*`)`

New in version 2017.7.5,2018.3.1.

Returns all available packages. Optionally, package names (and name globs) can be passed and the results will be filtered to packages matching those names. This is recommended as it speeds up the function considerably.

This function can be helpful in discovering the version or repo to specify in a *`pkg.installed`* state.

The return data will be a dictionary mapping package names to a list of version numbers, ordered from newest to oldest. If `byrepo` is set to `True`, then the return dictionary will contain repository names at the top level, and each repository will map packages to lists of version numbers. For example:

```
# With byrepo=False (default)
{
    'bash': ['4.3-83.3.1',
             '4.3-82.6'],
    'vim': ['7.4.326-12.1']
}
```

(continues on next page)

(continued from previous page)

```
{
    'OSS': {
        'bash': ['4.3-82.6'],
        'vim': ['7.4.326-12.1']
    },
    'OSS Update': {
        'bash': ['4.3-83.3.1']
    }
}
```

fromrepo [None] Only include results from the specified repo(s). Multiple repos can be specified, comma-separated.

byrepo [False] When `True`, the return data for each package will be organized by repository.

CLI Examples:

```
salt '*' pkg.list_repo_pkgs
salt '*' pkg.list_repo_pkgs foo bar baz
salt '*' pkg.list_repo_pkgs 'python2-*' byrepo=True
salt '*' pkg.list_repo_pkgs 'python2-*' fromrepo='OSS Updates'
```

`salt.modules.zypperpkg.`**`list_repos`**`()`

Lists all repos.

CLI Example:

```
salt '*' pkg.list_repos
```

`salt.modules.zypperpkg.`**`list_upgrades`**`(`*refresh=True*, ***kwargs*`)`

List all available package upgrades on this system

refresh force a refresh if set to True (default). If set to False it depends on zypper if a refresh is executed.

CLI Example:

```
salt '*' pkg.list_upgrades
```

`salt.modules.zypperpkg.`**`mod_repo`**`(`*repo*, ***kwargs*`)`

Modify one or more values for a repo. If the repo does not exist, it will be created, so long as the following values are specified:

repo or alias alias by which Zypper refers to the repo

url, mirrorlist or baseurl the URL for Zypper to reference

enabled Enable or disable (True or False) repository, but do not remove if disabled.

refresh Enable or disable (True or False) auto-refresh of the repository.

cache Enable or disable (True or False) RPM files caching.

gpgcheck Enable or disable (True or False) GPG check for this repository.

gpgautoimport [False] If set to True, automatically trust and import public GPG key for the repository.

Key/Value pairs may also be removed from a repo's configuration by setting a key to a blank value. Bear in mind that a name cannot be deleted, and a URL can only be deleted if a `mirrorlist` is specified (or vice versa).

CLI Examples:

```
salt '*' pkg.mod_repo alias alias=new_alias
salt '*' pkg.mod_repo alias url= mirrorlist=http://host.com/
```

`salt.modules.zypperpkg.`**`modified`**`(`**packages*, ***flags*`)`

List the modified files that belong to a package. Not specifying any packages will return a list of _all_ modified files on the system's RPM database.

New in version 2015.5.0.

Filtering by flags (True or False):
size Include only files where size changed.
mode Include only files which file's mode has been changed.
checksum Include only files which MD5 checksum has been changed.
device Include only files which major and minor numbers has been changed.
symlink Include only files which are symbolic link contents.
owner Include only files where owner has been changed.
group Include only files where group has been changed.
time Include only files where modification time of the file has been changed.
capabilities Include only files where capabilities differ or not. Note: supported only on newer RPM versions.
CLI Examples:

```
salt '*' pkg.modified
salt '*' pkg.modified httpd
salt '*' pkg.modified httpd postfix
salt '*' pkg.modified httpd owner=True group=False
```

`salt.modules.zypperpkg.`**`normalize_name`**`(`*name*`)`

Strips the architecture from the specified package name, if necessary. Circumstances where this would be done include:

- If the arch is 32 bit and the package name ends in a 32-bit arch.
- If the arch matches the OS arch, or is `noarch`.

CLI Example:

```
salt '*' pkg.normalize_name zsh.x86_64
```

`salt.modules.zypperpkg.`**`owner`**`(`**paths*`)`

Return the name of the package that owns the file. Multiple file paths can be passed. If a single path is passed, a string will be returned, and if multiple paths are passed, a dictionary of file/package name pairs will be returned.

If the file is not owned by a package, or is not present on the minion, then an empty string will be returned for that path.

CLI Examples:

```
salt '*' pkg.owner /usr/bin/apachectl
salt '*' pkg.owner /usr/bin/apachectl /etc/httpd/conf/httpd.conf
```

`salt.modules.zypperpkg.`**`parse_arch`**`(`*name*`)`

Parse name and architecture from the specified package name.

CLI Example:

```
salt '*' pkg.parse_arch zsh.x86_64
```

`salt.modules.zypperpkg.`**`purge`**`(`*name=None*, *pkgs=None*, ***kwargs*`)`

Changed in version 2015.8.12,2016.3.3,2016.11.0: On minions running systemd>=205, systemd-run(1) is now used to isolate commands which modify installed packages from the `salt-minion` daemon's control group. This is done to keep systemd from killing any zypper commands spawned by Salt when the `salt-minion` service is restarted. (see `KillMode` in the systemd.kill(5) manpage for more information). If desired, usage of systemd-run(1) can be suppressed by setting a *`config option`* called `systemd.scope`, with a value of `False` (no quotes).

Recursively remove a package and all dependencies which were installed with it, this will call a `zypper -n remove -u`

name The name of the package to be deleted.
Multiple Package Options:
pkgs A list of packages to delete. Must be passed as a python list. The `name` parameter will be ignored if this option is passed.
New in version 0.16.0.

Returns a dict containing the changes.

CLI Example:

```
salt '*' pkg.purge <package name>
salt '*' pkg.purge <package1>,<package2>,<package3>
salt '*' pkg.purge pkgs='["foo", "bar"]'
```

`salt.modules.zypperpkg.`**`refresh_db`**`()`

Force a repository refresh by calling `zypper refresh --force`, return a dict:

```
{'<database name>': Bool}
```

CLI Example:

```
salt '*' pkg.refresh_db
```

`salt.modules.zypperpkg.`**`remove`**`(`*name=None, pkgs=None, **kwargs*`)`

Changed in version 2015.8.12,2016.3.3,2016.11.0: On minions running systemd>=205, systemd-run(1) is now used to isolate commands which modify installed packages from the `salt-minion` daemon's control group. This is done to keep systemd from killing any zypper commands spawned by Salt when the `salt-minion` service is restarted. (see `KillMode` in the systemd.kill(5) manpage for more information). If desired, usage of systemd-run(1) can be suppressed by setting a *`config option`* called `systemd.scope`, with a value of `False` (no quotes).

Remove packages with `zypper -n remove`
name The name of the package to be deleted.
Multiple Package Options:
pkgs A list of packages to delete. Must be passed as a python list. The `name` parameter will be ignored if this option is passed.
New in version 0.16.0.

Returns a dict containing the changes.

CLI Example:

```
salt '*' pkg.remove <package name>
salt '*' pkg.remove <package1>,<package2>,<package3>
salt '*' pkg.remove pkgs='["foo", "bar"]'
```

`salt.modules.zypperpkg.`**`remove_lock`**`(`*packages, **kwargs*`)`

Remove specified package lock.

CLI Example:

```
salt '*' pkg.remove_lock <package name>
salt '*' pkg.remove_lock <package1>,<package2>,<package3>
salt '*' pkg.remove_lock pkgs='["foo", "bar"]'
```

`salt.modules.zypperpkg.`**`resolve_capabilities`**`(`*pkgs, refresh, **kwargs*`)`

New in version 2018.3.0.

Convert name provides in `pkgs` into real package names if `resolve_capabilities` parameter is set to True. In case of `resolve_capabilities` is set to False the package list is returned unchanged.

refresh force a refresh if set to True. If set to False (default) it depends on zypper if a refresh is executed.

resolve_capabilities If this option is set to True the input will be checked if a package with this name exists. If not, this function will search for a package which provides this name. If one is found the output is exchanged with the real package name. In case this option is set to False (Default) the input will be returned unchanged.

CLI Examples:

```
salt '*' pkg.resolve_capabilities resolve_capabilities=True w3m_ssl
```

`salt.modules.zypperpkg.`**`search`**(*criteria*, *refresh=False*, ***kwargs*)

List known packags, available to the system.

refresh force a refresh if set to True. If set to False (default) it depends on zypper if a refresh is executed.

match (str) One of *exact*, *words*, *substrings*. Search for an *exact* match or for the whole *words* only. Default to *substrings* to patch partial words.

provides (bool) Search for packages which provide the search strings.

recommends (bool) Search for packages which recommend the search strings.

requires (bool) Search for packages which require the search strings.

suggests (bool) Search for packages which suggest the search strings.

conflicts (bool) Search packages conflicting with search strings.

obsoletes (bool) Search for packages which obsolete the search strings.

file_list (bool) Search for a match in the file list of packages.

search_descriptions (bool) Search also in package summaries and descriptions.

case_sensitive (bool) Perform case-sensitive search.

installed_only (bool) Show only installed packages.

not_installed_only (bool) Show only packages which are not installed.

details (bool) Show version and repository

CLI Examples:

```
salt '*' pkg.search <criteria>
```

`salt.modules.zypperpkg.`**`upgrade`**(*refresh=True*, *dryrun=False*, *dist_upgrade=False*, *fromrepo=None*, *novendorchange=False*, *skip_verify=False*, ***kwargs*)

Changed in version 2015.8.12,2016.3.3,2016.11.0: On minions running systemd>=205, systemd-run(1) is now used to isolate commands which modify installed packages from the `salt-minion` daemon's control group. This is done to keep systemd from killing any zypper commands spawned by Salt when the `salt-minion` service is restarted. (see `KillMode` in the systemd.kill(5) manpage for more information). If desired, usage of systemd-run(1) can be suppressed by setting a *`config option`* called `systemd.scope`, with a value of `False` (no quotes).

Run a full system upgrade, a zypper upgrade

refresh force a refresh if set to True (default). If set to False it depends on zypper if a refresh is executed.

dryrun If set to True, it creates a debug solver log file and then perform a dry-run upgrade (no changes are made). Default: False

dist_upgrade Perform a system dist-upgrade. Default: False

fromrepo Specify a list of package repositories to upgrade from. Default: None

novendorchange If set to True, no allow vendor changes. Default: False

skip_verify Skip the GPG verification check (e.g., `--no-gpg-checks`)

Returns a dictionary containing the changes:

```
{'<package>':  {'old': '<old-version>',
                'new': '<new-version>'}}
```

CLI Example:

```
salt '*' pkg.upgrade
salt '*' pkg.upgrade dist_upgrade=True fromrepo='["MyRepoName"]'␣
↪novendorchange=True
salt '*' pkg.upgrade dist_upgrade=True dryrun=True
```

`salt.modules.zypperpkg.`**`upgrade_available`**(*name*, ***kwargs*)

Check whether or not an upgrade is available for a given package

refresh force a refresh if set to True (default). If set to False it depends on zypper if a refresh is executed or not.

CLI Example:

```
salt '*' pkg.upgrade_available <package name>
```

`salt.modules.zypperpkg.`**`verify`**(**names*, ***kwargs*)

Runs an rpm -Va on a system, and returns the results in a dict

Files with an attribute of config, doc, ghost, license or readme in the package header can be ignored using the `ignore_types` keyword argument

CLI Example:

```
salt '*' pkg.verify
salt '*' pkg.verify httpd
salt '*' pkg.verify 'httpd postfix'
salt '*' pkg.verify 'httpd postfix' ignore_types=['config','doc']
```

`salt.modules.zypperpkg.`**`version`**(**names*, ***kwargs*)

Returns a string representing the package version or an empty dict if not installed. If more than one package name is specified, a dict of name/version pairs is returned.

CLI Example:

```
salt '*' pkg.version <package name>
salt '*' pkg.version <package1> <package2> <package3> ...
```

`salt.modules.zypperpkg.`**`version_cmp`**(*ver1*, *ver2*, *ignore_epoch=False*)

New in version 2015.5.4.

Do a cmp-style comparison on two packages. Return -1 if ver1 < ver2, 0 if ver1 == ver2, and 1 if ver1 > ver2. Return None if there was a problem making the comparison.

ignore_epoch [False] Set to `True` to ignore the epoch when comparing versions

New in version 2015.8.10,2016.3.2.

CLI Example:

```
salt '*' pkg.version_cmp '0.2-001' '0.2.0.1-002'
```

21.10 netapi modules

21.10.1 rest_cherrypy

- *A REST API for Salt*

  - *Authentication*
  - *Usage*
  - *Content negotiation*
- *Performance Expectations and Recommended Usage*
  - *Long-Running HTTP Connections*
  - *Timeouts*
  - *Best Practices*
  - *Performance Tuning*
  - *Future Plans*
- *Deployment*
  - ***`salt-api`*** *using the CherryPy server*
  - *Using a WSGI-compliant web server*
- *REST URI Reference*
  - `/`
  - `/login`
  - `/logout`
  - `/minions`
  - `/jobs`
  - `/run`
  - `/events`
  - `/hook`
  - `/keys`
  - `/ws`
  - `/stats`

A REST API for Salt

depends

- CherryPy Python module.

  Note: there is a known SSL traceback for CherryPy versions 3.2.5 through 3.7.x. Please use version 3.2.3 or the latest 10.x version instead.

optdepends

- ws4py Python module for websockets support.

client_libraries

- Java: https://github.com/SUSE/salt-netapi-client
- Python: https://github.com/saltstack/pepper

setup All steps below are performed on the machine running the Salt Master daemon. Configuration goes into the Master configuration file.

1. Install `salt-api`. (This step varies between OS and Linux distros. Some package systems have a split package, others include salt-api in the main Salt package. Ensure the `salt-api --version` output matches the `salt --version` output.)
2. Install CherryPy. (Read the version caveat in the section above.)
3. Optional: generate self-signed SSL certificates.

   Using a secure HTTPS connection is strongly recommended since Salt eauth authentication credentials will be sent over the wire.

   1. Install the PyOpenSSL package.
   2. Generate a self-signed certificate using the *`create_self_signed_cert()`* execution function.

      ```
      salt-call --local tls.create_self_signed_cert
      ```

4. Edit the master config to create at least one external auth user or group following the *full external auth instructions*.
5. Edit the master config with the following production-ready example to enable the `rest_cherrypy` module. (Adjust cert paths as needed, or disable SSL (not recommended!).)

   ```
   rest_cherrypy:
     port: 8000
     ssl_crt: /etc/pki/tls/certs/localhost.crt
     ssl_key: /etc/pki/tls/certs/localhost.key
   ```

6. Restart the `salt-master` daemon.
7. Start the `salt-api` daemon.

configuration All available configuration options are detailed below. These settings configure the CherryPy HTTP server and do not apply when using an external server such as Apache or Nginx.

port **Required**

The port for the webserver to listen on.

host [`0.0.0.0`] The socket interface for the HTTP server to listen on.

debug [`False`] Starts the web server in development mode. It will reload itself when the underlying code is changed and will output more debugging info.

log_access_file Path to a file to write HTTP access logs.

New in version 2016.11.0.

log_error_file Path to a file to write HTTP error logs.

New in version 2016.11.0.

ssl_crt The path to a SSL certificate. (See below)

ssl_key The path to the private key for your SSL certificate. (See below)

ssl_chain (Optional when using PyOpenSSL) the certificate chain to pass to `Context.load_verify_locations`.

disable_ssl A flag to disable SSL. Warning: your Salt authentication credentials will be sent in the clear!

webhook_disable_auth [False] The *Webhook* URL requires authentication by default but external services cannot always be configured to send authentication. See the Webhook documentation for suggestions on securing this interface.

webhook_url [/hook] Configure the URL endpoint for the *Webhook* entry point.

thread_pool [`100`] The number of worker threads to start up in the pool.

socket_queue_size [`30`] Specify the maximum number of HTTP connections to queue.

expire_responses [True] Whether to check for and kill HTTP responses that have exceeded the default timeout.

Deprecated since version 2016.11.9,2017.7.3,2018.3.0: The "expire_responses" configuration setting, which corresponds to the `timeout_monitor` setting in CherryPy, is no longer supported in CherryPy versions >= 12.0.0.

max_request_body_size [`1048576`] Maximum size for the HTTP request body.

collect_stats [False] Collect and report statistics about the CherryPy server

Reports are available via the *Stats* URL.

stats_disable_auth [False] Do not require authentication to access the `/stats` endpoint.

New in version 2018.3.0.

static A filesystem path to static HTML/JavaScript/CSS/image assets.

static_path [`/static`] The URL prefix to use when serving static assets out of the directory specified in the `static` setting.

enable_sessions [`True`] Enable or disable all endpoints that rely on session cookies. This can be useful to enforce only header-based authentication.

New in version 2017.7.0.

app [`index.html`] A filesystem path to an HTML file that will be served as a static file. This is useful for bootstrapping a single-page JavaScript app.

Warning! If you set this option to a custom web application, anything that uses cookie-based authentication is vulnerable to XSRF attacks. Send the custom `X-Auth-Token` header instead and consider disabling the `enable_sessions` setting.

Changed in version 2017.7.0: Add a proof-of-concept JavaScript single-page app.

app_path [`/app`] The URL prefix to use for serving the HTML file specified in the `app` setting. This should be a simple name containing no slashes.

Any path information after the specified path is ignored; this is useful for apps that utilize the HTML5 history API.

root_prefix [`/`] A URL path to the main entry point for the application. This is useful for serving multiple applications from the same URL.

Authentication

Authentication is performed by passing a session token with each request. Tokens are generated via the *Login* URL.

The token may be sent in one of two ways: as a custom header or as a session cookie. The latter is far more convenient for clients that support cookies.

- Include a custom header named *X-Auth-Token*.

  For example, using curl:

  ```
  curl -sSk https://localhost:8000/login \
      -H 'Accept: application/x-yaml' \
      -d username=saltdev \
      -d password=saltdev \
      -d eauth=pam
  ```

  Copy the `token` value from the output and include it in subsequent requests:

  ```
  curl -sSk https://localhost:8000 \
      -H 'Accept: application/x-yaml' \
      -H 'X-Auth-Token: 697adbdc8fe971d09ae4c2a3add7248859c87079'\
      -d client=local \
      -d tgt='*' \
      -d fun=test.ping
  ```

- Sent via a cookie. This option is a convenience for HTTP clients that automatically handle cookie support (such as browsers).

  For example, using curl:

  ```
  # Write the cookie file:
  curl -sSk https://localhost:8000/login \
        -c ~/cookies.txt \
        -H 'Accept: application/x-yaml' \
        -d username=saltdev \
        -d password=saltdev \
        -d eauth=auto

  # Read the cookie file:
  curl -sSk https://localhost:8000 \
        -b ~/cookies.txt \
        -H 'Accept: application/x-yaml' \
        -d client=local \
        -d tgt='*' \
        -d fun=test.ping
  ```

  Another example using the **`requests`** library in Python:

  ```
  >>> import requests
  >>> session = requests.Session()
  >>> session.post('http://localhost:8000/login', json={
      'username': 'saltdev',
      'password': 'saltdev',
      'eauth': 'auto',
  })
  <Response [200]>
  >>> resp = session.post('http://localhost:8000', json=[{
      'client': 'local',
      'tgt': '*',
      'fun': 'test.arg',
      'arg': ['foo', 'bar'],
      'kwarg': {'baz': 'Baz!'},
  ```

(continues on next page)

(continued from previous page)

```
}])
>>> resp.json()
{u'return': [{
    ...snip...
}]}
```

See also:

You can bypass the session handling via the *Run* URL.

Usage

This interface directly exposes Salt's *Python API*. Everything possible at the CLI is possible through the Python API. Commands are executed on the Salt Master.

The root URL (/) is RPC-like in that it accepts instructions in the request body for what Salt functions to execute, and the response contains the result of those function calls.

For example:

```
% curl -sSi https://localhost:8000         -H 'Content-type: application/json'      ␣
↪ -d '[{
        "client": "local",
        "tgt": "*",
        "fun": "test.ping"
    }]'
HTTP/1.1 200 OK
Content-Type: application/json
[...snip...]

{"return": [{"jerry": true}]}
```

The request body must be an array of commands. Use this workflow to build a command:

1. Choose a client interface.
2. Choose a function.
3. Fill out the remaining parameters needed for the chosen client.

The `client` field is a reference to the main Python classes used in Salt's Python API. Read the full *Client APIs* documentation, but in short:

- "local" uses `LocalClient` which sends commands to Minions. Equivalent to the `salt` CLI command.
- "runner" uses `RunnerClient` which invokes runner modules on the Master. Equivalent to the `salt-run` CLI command.
- "wheel" uses `WheelClient` which invokes wheel modules on the Master. Wheel modules do not have a direct CLI equivalent but they typically manage Master-side resources such as state files, pillar files, the Salt config files, and the `key wheel module` exposes similar functionality as the `salt-key` CLI command.

Most clients have variants like synchronous or asynchronous execution as well as others like batch execution. See the *full list of client interfaces*.

Each client requires different arguments and sometimes has different syntax. For example, `LocalClient` requires the `tgt` argument because it forwards the command to Minions and the other client interfaces do not.

`LocalClient` also takes `arg` (array) and `kwarg` (dictionary) arguments because these values are sent to the Minions and used to execute the requested function there. `RunnerClient` and `WheelClient` are executed directly on the Master and thus do not need or accept those arguments.

Read the method signatures in the client documentation linked above, but hopefully an example will help illustrate the concept. This example causes Salt to execute two functions – the *`test.arg execution function`* using `LocalClient` and the *`test.arg runner function`* using `RunnerClient`; note the different structure for each command. The results for both are combined and returned as one response.

```
% curl -b ~/cookies.txt -sSi localhost:8000         -H 'Content-type: application/json
↪'         -d '
[
    {
        "client": "local",
        "tgt": "*",
        "fun": "test.arg",
        "arg": ["positional arg one", "positional arg two"],
        "kwarg": {
            "keyword arg one": "Hello from a minion",
            "keyword arg two": "Hello again from a minion"
        }
    },
    {
        "client": "runner",
        "fun": "test.arg",
        "keyword arg one": "Hello from a master",
        "keyword arg two": "Runners do not support positional args"
    }
]
'
HTTP/1.1 200 OK
[...snip...]
{
  "return": [
    {
      "jerry": {
        "args": [
          "positional arg one",
          "positional arg two"
        ],
        "kwargs": {
          "keyword arg one": "Hello from a minion",
          "keyword arg two": "Hello again from a minion",
          [...snip...]
        }
      },
      [...snip; other minion returns here...]
    },
    {
      "args": [],
      "kwargs": {
        "keyword arg two": "Runners do not support positional args",
        "keyword arg one": "Hello from a master"
      }
    }
  ]
}
```

One more example, this time with more commonly used functions:

```
curl -b /tmp/cookies.txt -sSi localhost:8000         -H 'Content-type: application/
↪json'         -d '
[
    {
        "client": "local",
        "tgt": "*",
        "fun": "state.sls",
        "kwarg": {
            "mods": "apache",
            "pillar": {
                "lookup": {
                    "wwwdir": "/srv/httpd/htdocs"
                }
            }
        }
    },
    {
        "client": "runner",
        "fun": "cloud.create",
        "provider": "my-ec2-provider",
        "instances": "my-centos-6",
        "image": "ami-1624987f",
        "delvol_on_destroy", true
    }
]
'
HTTP/1.1 200 OK
[...snip...]
{
  "return": [
    {
      "jerry": {
        "pkg_|-install_apache_|-httpd_|-installed": {
            [...snip full state return here...]
        }
      }
      [...snip other minion returns here...]
    },
    {
        [...snip full salt-cloud output here...]
    }
  ]
}
```

Content negotiation

This REST interface is flexible in what data formats it will accept as well as what formats it will return (e.g., JSON, YAML, urlencoded).

- Specify the format of data in the request body by including the *`Content-Type`* header.
- Specify the desired data format for the response body with the *`Accept`* header.

We recommend the JSON format for most HTTP requests. urlencoded data is simple and cannot express complex data structures – and that is often required for some Salt commands, such as starting a state run that uses Pillar data. Salt's CLI tool can reformat strings passed in at the CLI into complex data structures, and that behavior also works via salt-api, but that can be brittle and since salt-api can accept JSON it is best just to send JSON.

Here is an example of sending urlencoded data:

```
curl -sSik https://localhost:8000 \
    -b ~/cookies.txt \
    -d client=runner \
    -d fun='jobs.lookup_jid' \
    -d jid='20150129182456704682'
```

---

urlencoded data caveats

- Only a single command may be sent per HTTP request.
- Repeating the `arg` parameter multiple times will cause those parameters to be combined into a single list.

  Note, some popular frameworks and languages (notably jQuery, PHP, and Ruby on Rails) will automatically append empty brackets onto repeated query string parameters. E.g., `?foo[]=fooone&foo[]=footwo`. This is **not** supported; send `?foo=fooone&foo=footwo` instead, or send JSON or YAML.

A note about `curl`

The `-d` flag to curl does *not* automatically urlencode data which can affect passwords and other data that contains characters that must be encoded. Use the `--data-urlencode` flag instead. E.g.:

```
curl -ksi http://localhost:8000/login \
-H "Accept: application/json" \
-d username='myapiuser' \
--data-urlencode password='1234+' \
-d eauth='pam'
```

Performance Expectations and Recommended Usage

This module provides a thin wrapper around *Salt's Python API*. Executing a Salt command via rest_cherrypy is directly analogous to executing a Salt command via Salt's CLI (which also uses the Python API) – they share the same semantics, performance characteristics, and 98% of the same code. As a rule-of-thumb: if you wouldn't do it at the CLI don't do it via this API.

Long-Running HTTP Connections

The CherryPy server is a production-ready, threading HTTP server written in Python. Because it makes use of a thread pool to process HTTP requests it is not ideally suited to maintaining large numbers of concurrent, synchronous connections. On moderate hardware with default settings it should top-out at around 30 to 50 concurrent connections.

That number of long-running, synchronous Salt processes is also not ideal. Like at the CLI, each Salt command run will start a process that instantiates its own `LocalClient`, which instantiates its own listener to the Salt event bus, and sends out its own periodic `saltutil.find_job` queries to determine if a Minion is still running the command. Not exactly a lightweight operation.

Timeouts

In addition to the above resource overhead for long-running connections, there are the usual HTTP timeout semantics for the CherryPy server, any HTTP client being used, as well as any hardware in between such as proxies, gateways, or load balancers. rest_cherrypy can be configured not to time-out long responses via the `expire_responses` setting, and both *`LocalClient`* and *`RunnerClient`* have their own timeout parameters that may be passed as top-level keywords:

```
curl -b /tmp/cookies.txt -sSi localhost:8000         -H 'Content-type: application/json'         -d '
[
    {
        "client": "local",
        "tgt": "*",
        "fun": "test.sleep",
        "kwarg": {"length": 30},
        "timeout": 60
    },
    {
        "client": "runner",
        "fun": "test.sleep",
        "kwarg": {"s_time": 30},
        "timeout": 60
    }
]
'
```

Best Practices

Given the performance overhead and HTTP timeouts for long-running operations described above, the most effective and most scalable way to use both Salt and salt-api is to run commands asynchronously using the `local_async`, `runner_async`, and `wheel_async` clients.

Running asynchronous jobs results in being able to process 3x more commands per second for `LocalClient` and 17x more commands per second for `RunnerClient`, in addition to much less network traffic and memory requirements. Job returns can be fetched from Salt's job cache via the `/jobs/<jid>` endpoint, or they can be collected into a data store using Salt's *Returner system*.

The `/events` endpoint is specifically designed to handle long-running HTTP connections and it exposes Salt's event bus which includes job returns. Watching this endpoint first, then executing asynchronous Salt commands second, is the most lightweight and scalable way to use `rest_cherrypy` while still receiving job returns in real-time. But this requires clients that can properly handle the inherent asynchronicity of that workflow.

Performance Tuning

The `thread_pool` and `socket_queue_size` settings can be used to increase the capacity of rest_cherrypy to handle incoming requests. Keep an eye on RAM usage as well as available file handles while testing changes to these settings. As salt-api is a thin wrapper around Salt's Python API, also keep an eye on the performance of Salt when testing.

Future Plans

Now that Salt uses the Tornado concurrency library internally, we plan to improve performance in the API by taking advantage of existing processes and event listeners and to use lightweight coroutines to facilitate more simultaneous HTTP connections and better support for synchronous operations. That effort can be tracked in issue 26505, but until that issue is closed rest_cherrypy will remain the officially recommended REST API.

Deployment

The `rest_cherrypy` netapi module is a standard Python WSGI app. It can be deployed one of two ways.

`salt-api` using the CherryPy server

The default configuration is to run this module using **`salt-api`** to start the Python-based CherryPy server. This server is lightweight, multi-threaded, encrypted with SSL, and should be considered production-ready. See the section above for performance expectations.

Using a WSGI-compliant web server

This module may be deployed on any WSGI-compliant server such as Apache with mod_wsgi or Nginx with FastCGI, to name just two (there are many).

Note, external WSGI servers handle URLs, paths, and SSL certs directly. The `rest_cherrypy` configuration options are ignored and the `salt-api` daemon does not need to be running at all. Remember Salt authentication credentials are sent in the clear unless SSL is being enforced!

An example Apache virtual host configuration:

```
<VirtualHost *:80>
    ServerName example.com
    ServerAlias *.example.com

    ServerAdmin webmaster@example.com

    LogLevel warn
    ErrorLog /var/www/example.com/logs/error.log
    CustomLog /var/www/example.com/logs/access.log combined

    DocumentRoot /var/www/example.com/htdocs

    WSGIScriptAlias / /path/to/salt/netapi/rest_cherrypy/wsgi.py
</VirtualHost>
```

REST URI Reference

/

class `salt.netapi.rest_cherrypy.app.`**`LowDataAdapter`**

The primary entry point to Salt's REST API

`GET``()`

An explanation of the API with links of where to go next

`GET /`

Request Headers
- **`Accept`** – the desired response format.

Status Codes
- **`200`** – success
- **`401`** – authentication required
- **`406`** – requested Content-Type not available

Example request:

```
curl -i localhost:8000
```

```
GET / HTTP/1.1
Host: localhost:8000
Accept: application/json
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
```

POST

Mock out specified imports.

This allows autodoc to do its thing without having oodles of req'd installed libs. This doesn't work with `import *` imports.

This Mock class can be configured to return a specific values at specific names, if required.

http://read-the-docs.readthedocs.org/en/latest/faq.html#i-get-import-errors-on-libraries-that-depend-on-c-modules

/login

class `salt.netapi.rest_cherrypy.app.`**`Login`**(**args*, ***kwargs*)

Log in to receive a session token

Authentication information.

GET()

Present the login interface

GET /login

An explanation of how to log in.

Status Codes

- **200** – success
- **401** – authentication required
- **406** – requested Content-Type not available

Example request:

```
curl -i localhost:8000/login
```

```
GET /login HTTP/1.1
Host: localhost:8000
Accept: text/html
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: text/html
```

POST(***kwargs*)

Authenticate against Salt's eauth system

POST /login

Request Headers

- **`X-Auth-Token`** – a session token from *`Login`*.
- **`Accept`** – the desired response format.
- **`Content-Type`** – the format of the request body.

Form Parameters

- **`eauth`** – the eauth backend configured for the user
- **`username`** – username
- **`password`** – password

Status Codes

- **`200`** – success
- **`401`** – authentication required
- **`406`** – requested Content-Type not available

Example request:

```
curl -si localhost:8000/login \
    -c ~/cookies.txt \
    -H "Accept: application/json" \
    -H "Content-type: application/json" \
    -d '{
        "username": "saltuser",
        "password": "saltuser",
        "eauth": "auto"
    }'
```

```
POST / HTTP/1.1
Host: localhost:8000
Content-Length: 42
Content-Type: application/json
Accept: application/json

{"username": "saltuser", "password": "saltuser", "eauth": "auto"}
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 206
X-Auth-Token: 6d1b722e
Set-Cookie: session_id=6d1b722e; expires=Sat, 17 Nov 2012 03:23:52 GMT;
↪Path=/

{"return": {
    "token": "6d1b722e",
    "start": 1363805943.776223,
    "expire": 1363849143.776224,
    "user": "saltuser",
    "eauth": "pam",
    "perms": [
        "grains.*",
        "status.*",
        "sys.*",
        "test.*"
    ]
}}
```

`/logout`

class `salt.netapi.rest_cherrypy.app.`**Logout**

Class to remove or invalidate sessions

POST()

Destroy the currently active session and expire the session cookie

`/minions`

class `salt.netapi.rest_cherrypy.app.`**Minions**

Convenience URLs for working with minions

GET(*mid=None*)

A convenience URL for getting lists of minions or getting minion details

GET /minions/(*mid*)

Request Headers

- **X-Auth-Token** – a session token from `Login`.
- **Accept** – the desired response format.

Status Codes

- **200** – success
- **401** – authentication required
- **406** – requested Content-Type not available

Example request:

```
curl -i localhost:8000/minions/ms-3
```

```
GET /minions/ms-3 HTTP/1.1
Host: localhost:8000
Accept: application/x-yaml
```

Example response:

```
HTTP/1.1 200 OK
Content-Length: 129005
Content-Type: application/x-yaml

return:
- ms-3:
    grains.items:
        ...
```

POST(***kwargs*)

Start an execution command and immediately return the job id

POST /minions

Request Headers

- **X-Auth-Token** – a session token from `Login`.
- **Accept** – the desired response format.
- **Content-Type** – the format of the request body.

Response Headers

- **Content-Type** – the format of the response body; depends on the *`Accept`* request header.

Status Codes

- **`200`** – success
- **`400`** – bad or malformed request
- **`401`** – authentication required
- **`406`** – requested Content-Type not available

Lowstate data describing Salt commands must be sent in the request body. The `client` option will be set to `local_async()`.

Example request:

```
curl -sSi localhost:8000/minions \
    -b ~/cookies.txt \
    -H "Accept: application/x-yaml" \
    -d '[{"tgt": "*", "fun": "status.diskusage"}]'
```

```
POST /minions HTTP/1.1
Host: localhost:8000
Accept: application/x-yaml
Content-Type: application/json

tgt=*&fun=status.diskusage
```

Example response:

```
HTTP/1.1 202 Accepted
Content-Length: 86
Content-Type: application/x-yaml

return:
- jid: '20130603122505459265'
  minions: [ms-4, ms-3, ms-2, ms-1, ms-0]
_links:
  jobs:
    - href: /jobs/20130603122505459265
```

`/jobs`

class `salt.netapi.rest_cherrypy.app.`**`Jobs`**

`GET`(*jid=None, timeout=”*)

A convenience URL for getting lists of previously run jobs or getting the return from a single job

`GET /jobs/`(*jid*)

List jobs or show a single job from the job cache.

Request Headers

- **`X-Auth-Token`** – a session token from *`Login`*.
- **`Accept`** – the desired response format.

Status Codes

- **`200`** – success
- **`401`** – authentication required
- **`406`** – requested Content-Type not available

Example request:

```
curl -i localhost:8000/jobs
```

```
GET /jobs HTTP/1.1
Host: localhost:8000
Accept: application/x-yaml
```

Example response:

```
HTTP/1.1 200 OK
Content-Length: 165
Content-Type: application/x-yaml

return:
- '20121130104633606931':
    Arguments:
    - '3'
    Function: test.fib
    Start Time: 2012, Nov 30 10:46:33.606931
    Target: jerry
    Target-type: glob
```

Example request:

```
curl -i localhost:8000/jobs/20121130104633606931
```

```
GET /jobs/20121130104633606931 HTTP/1.1
Host: localhost:8000
Accept: application/x-yaml
```

Example response:

```
HTTP/1.1 200 OK
Content-Length: 73
Content-Type: application/x-yaml

info:
- Arguments:
    - '3'
    Function: test.fib
    Minions:
    - jerry
    Start Time: 2012, Nov 30 10:46:33.606931
    Target: '*'
    Target-type: glob
    User: saltdev
    jid: '20121130104633606931'
return:
- jerry:
    - - 0
    - 1
    - 1
    - 2
    - 6.9141387939453125e-06
```

/run

class `salt.netapi.rest_cherrypy.app.`**`Run`**

Run commands bypassing the *normal session handling* salt-api does not enforce authorization, Salt's eauth system does that. Local/Runner/WheelClient all accept `username`/`password`/`eauth` **or** `token` kwargs that are then checked by the eauth system. The session mechanism in `rest_cherrypy` simply pairs a session with a Salt eauth token and then passes the `token` kwarg in automatically.

If you already have a Salt eauth token, perhaps generated by the *`mk_token`* function in the Auth Runner module, then there is no reason to use sessions.

This endpoint accepts either a `username`, `password`, `eauth` trio, **or** a `token` kwarg and does not make use of sessions at all.

`POST`(***kwargs*)

Run commands bypassing the *normal session handling* Other than that this URL is identical to the *`root URL (/)`*.

`POST /run`

An array of lowstate data describing Salt commands must be sent in the request body.

Status Codes

- **`200`** – success
- **`400`** – bad or malformed request
- **`401`** – authentication required
- **`406`** – requested Content-Type not available

Example request:

```
curl -sS localhost:8000/run \
    -H 'Accept: application/x-yaml' \
    -H 'Content-type: application/json' \
    -d '[{
        "client": "local",
        "tgt": "*",
        "fun": "test.ping",
        "username": "saltdev",
        "password": "saltdev",
        "eauth": "auto"
    }]'
```

Or using a Salt Eauth token:

```
curl -sS localhost:8000/run \
    -H 'Accept: application/x-yaml' \
    -H 'Content-type: application/json' \
    -d '[{
        "client": "local",
        "tgt": "*",
        "fun": "test.ping",
        "token": "<salt eauth token here>"
    }]'
```

```
POST /run HTTP/1.1
Host: localhost:8000
Accept: application/x-yaml
Content-Length: 75
Content-Type: application/json

[{"client": "local", "tgt": "*", "fun": "test.ping", "username": "saltdev",
↪"password": "saltdev", "eauth": "auto"}]
```

Example response:

```
HTTP/1.1 200 OK
Content-Length: 73
Content-Type: application/x-yaml

return:
- ms-0: true
  ms-1: true
  ms-2: true
  ms-3: true
  ms-4: true
```

The /run enpoint can also be used to issue commands using the salt-ssh subsystem.

When using salt-ssh, eauth credentials should not be supplied. Instead, authentication should be handled by the SSH layer itself. The use of the salt-ssh client does not require a salt master to be running. Instead, only a roster file must be present in the salt configuration directory.

All SSH client requests are synchronous.

Example SSH client request:

```
curl -sS localhost:8000/run \
    -H 'Accept: application/x-yaml' \
    -d client='ssh' \
    -d tgt='*' \
    -d fun='test.ping'
```

```
POST /run HTTP/1.1
Host: localhost:8000
Accept: application/x-yaml
Content-Length: 75
Content-Type: application/x-www-form-urlencoded

client=ssh&tgt=*&fun=test.ping
```

Example SSH response:

```
return:
- silver:
  fun: test.ping
  fun_args: []
  id: silver
  jid: '20141203103525666185'
  retcode: 0
  return: true
  success: true
```

/events

class salt.netapi.rest_cherrypy.app.**Events**

Expose the Salt event bus

The event bus on the Salt master exposes a large variety of things, notably when executions are started on the master and also when minions ultimately return their results. This URL provides a real-time window into a running Salt infrastructure.

See also:

Events & Reactor

GET(*token=None*, *salt_token=None*)

An HTTP stream of the Salt master event bus

This stream is formatted per the Server Sent Events (SSE) spec. Each event is formatted as JSON.

GET /events

Status Codes

- **200** – success
- **401** – authentication required
- **406** – requested Content-Type not available

Query Parameters

- **token** – **optional** parameter containing the token ordinarily supplied via the X-Auth-Token header in order to allow cross-domain requests in browsers that do not include CORS support in the EventSource API. E.g., `curl -NsS localhost:8000/events?token=308650d`
- **salt_token** – **optional** parameter containing a raw Salt *eauth token* (not to be confused with the token returned from the /login URL). E.g., `curl -NsS localhost:8000/events?salt_token=30742765`

Example request:

```
curl -NsS localhost:8000/events
```

```
GET /events HTTP/1.1
Host: localhost:8000
```

Example response:

Note, the `tag` field is not part of the spec. SSE compliant clients should ignore unknown fields. This addition allows non-compliant clients to only watch for certain tags without having to deserialze the JSON object each time.

```
HTTP/1.1 200 OK
Connection: keep-alive
Cache-Control: no-cache
Content-Type: text/event-stream;charset=utf-8

retry: 400

tag: salt/job/20130802115730568475/new
data: {'tag': 'salt/job/20130802115730568475/new', 'data': {'minions': ['ms-4
↪', 'ms-3', 'ms-2', 'ms-1', 'ms-0']}}

tag: salt/job/20130802115730568475/ret/jerry
data: {'tag': 'salt/job/20130802115730568475/ret/jerry', 'data': {'jid':
↪'20130802115730568475', 'return': True, 'retcode': 0, 'success': True, 'cmd
↪': '_return', 'fun': 'test.ping', 'id': 'ms-1'}}
```

The event stream can be easily consumed via JavaScript:

```
var source = new EventSource('/events');
source.onopen = function() { console.info('Listening ...') };
source.onerror = function(err) { console.error(err) };
source.onmessage = function(message) {
    var saltEvent = JSON.parse(message.data);
```

(continues on next page)

(continued from previous page)

```
    console.log(saltEvent.tag, saltEvent.data);
};
```

Note, the SSE stream is fast and completely asynchronous and Salt is very fast. If a job is created using a regular POST request, it is possible that the job return will be available on the SSE stream before the response for the POST request arrives. It is important to take that asynchronicity into account when designing an application. Below are some general guidelines.

- Subscribe to the SSE stream _before_ creating any events.
- Process SSE events directly as they arrive and don't wait for any other process to "complete" first (like an ajax request).
- Keep a buffer of events if the event stream must be used for synchronous lookups.
- Be cautious in writing Salt's event stream directly to the DOM. It is very busy and can quickly overwhelm the memory allocated to a browser tab.

A full, working proof-of-concept JavaScript application is available adjacent to this file. It can be viewed by pointing a browser at the `/app` endpoint in a running `rest_cherrypy` instance.

Or using CORS:

```
var source = new EventSource('/events?
↪token=ecd589e4e01912cf3c4035afad73426dbb8dba75', {withCredentials: true});
```

It is also possible to consume the stream via the shell.

Records are separated by blank lines; the `data:` and `tag:` prefixes will need to be removed manually before attempting to unserialize the JSON.

curl's `-N` flag turns off input buffering which is required to process the stream incrementally.

Here is a basic example of printing each event as it comes in:

```
curl -NsS localhost:8000/events |\
        while IFS= read -r line ; do
            echo $line
        done
```

Here is an example of using awk to filter events based on tag:

```
curl -NsS localhost:8000/events |\
        awk '
            BEGIN { RS=""; FS="\\n" }
            $1 ~ /^tag: salt\/job\/[0-9]+\/new$/ { print $0 }
        '
tag: salt/job/20140112010149808995/new
data: {"tag": "salt/job/20140112010149808995/new", "data": {"tgt_type": "glob
↪", "jid": "20140112010149808995", "tgt": "jerry", "_stamp": "2014-01-12_
↪01:01:49.809617", "user": "shouse", "arg": [], "fun": "test.ping", "minions
↪": ["jerry"]}}
tag: 20140112010149808995
data: {"tag": "20140112010149808995", "data": {"fun_args": [], "jid":
↪"20140112010149808995", "return": true, "retcode": 0, "success": true, "cmd
↪": "_return", "_stamp": "2014-01-12_01:01:49.819316", "fun": "test.ping",
↪"id": "jerry"}}
```

/hook

class `salt.netapi.rest_cherrypy.app.`**`Webhook`**

A generic web hook entry point that fires an event on Salt's event bus

External services can POST data to this URL to trigger an event in Salt. For example, Amazon SNS, Jenkins-CI or Travis-CI, or GitHub web hooks.

Note: Be mindful of security

Salt's Reactor can run any code. A Reactor SLS that responds to a hook event is responsible for validating that the event came from a trusted source and contains valid data.

This is a generic interface and securing it is up to you!

This URL requires authentication however not all external services can be configured to authenticate. For this reason authentication can be selectively disabled for this URL. Follow best practices – always use SSL, pass a secret key, configure the firewall to only allow traffic from a known source, etc.

The event data is taken from the request body. The *`Content-Type`* header is respected for the payload.

The event tag is prefixed with `salt/netapi/hook` and the URL path is appended to the end. For example, a `POST` request sent to `/hook/mycompany/myapp/mydata` will produce a Salt event with the tag `salt/netapi/hook/mycompany/myapp/mydata`.

The following is an example `.travis.yml` file to send notifications to Salt of successful test runs:

```
language: python
script: python -m unittest tests
after_success:
    - |
        curl -sSk https://saltapi-url.example.com:8000/hook/travis/build/success
↪                          -d branch="${TRAVIS_BRANCH}"                          -d
↪commit="${TRAVIS_COMMIT}"
```

See also:

Events & Reactor, *reactor*

`POST`(**args*, ***kwargs*)

Fire an event in Salt with a custom event tag and data

`POST /hook`

Status Codes

- **`200`** – success
- **`401`** – authentication required
- **`406`** – requested Content-Type not available
- **`413`** – request body is too large

Example request:

```
curl -sS localhost:8000/hook \
    -H 'Content-type: application/json' \
    -d '{"foo": "Foo!", "bar": "Bar!"}'
```

```
POST /hook HTTP/1.1
Host: localhost:8000
Content-Length: 16
```

(continues on next page)

(continued from previous page)

```
Content-Type: application/json

{"foo": "Foo!", "bar": "Bar!"}
```

Example response:

```
HTTP/1.1 200 OK
Content-Length: 14
Content-Type: application/json

{"success": true}
```

As a practical example, an internal continuous-integration build server could send an HTTP POST request to the URL `https://localhost:8000/hook/mycompany/build/success` which contains the result of a build and the SHA of the version that was built as JSON. That would then produce the following event in Salt that could be used to kick off a deployment via Salt's Reactor:

```
Event fired at Fri Feb 14 17:40:11 2014
*************************
Tag: salt/netapi/hook/mycompany/build/success
Data:
{'_stamp': '2014-02-14_17:40:11.440996',
    'headers': {
        'X-My-Secret-Key': 'F0fAgoQjIT@W',
        'Content-Length': '37',
        'Content-Type': 'application/json',
        'Host': 'localhost:8000',
        'Remote-Addr': '127.0.0.1'},
    'post': {'revision': 'aa22a3c4b2e7', 'result': True}}
```

Salt's Reactor could listen for the event:

```
reactor:
  - 'salt/netapi/hook/mycompany/build/*':
    - /srv/reactor/react_ci_builds.sls
```

And finally deploy the new build:

```
{% set secret_key = data.get('headers', {}).get('X-My-Secret-Key') %}
{% set build = data.get('post', {}) %}

{% if secret_key == 'F0fAgoQjIT@W' and build.result == True %}
deploy_my_app:
  cmd.state.sls:
    - tgt: 'application*'
    - arg:
      - myapp.deploy
    - kwarg:
        pillar:
          revision: {{ revision }}
{% endif %}
```

/keys

class `salt.netapi.rest_cherrypy.app.`**Keys**

Convenience URLs for working with minion keys

New in version 2014.7.0.

These URLs wrap the functionality provided by the *`key wheel module`* functions.

GET(*mid=None*)

Show the list of minion keys or detail on a specific key

New in version 2014.7.0.

GET /keys/(*mid*)

List all keys or show a specific key

Request Headers

- **X-Auth-Token** – a session token from *`Login`*.
- **Accept** – the desired response format.

Status Codes

- **200** – success
- **401** – authentication required
- **406** – requested Content-Type not available

Example request:

```
curl -i localhost:8000/keys
```

```
GET /keys HTTP/1.1
Host: localhost:8000
Accept: application/x-yaml
```

Example response:

```
HTTP/1.1 200 OK
Content-Length: 165
Content-Type: application/x-yaml

return:
  local:
  - master.pem
  - master.pub
  minions:
  - jerry
  minions_pre: []
  minions_rejected: []
```

Example request:

```
curl -i localhost:8000/keys/jerry
```

```
GET /keys/jerry HTTP/1.1
Host: localhost:8000
Accept: application/x-yaml
```

Example response:

```
HTTP/1.1 200 OK
Content-Length: 73
Content-Type: application/x-yaml

return:
  minions:
    jerry: 51:93:b3:d0:9f:3a:6d:e5:28:67:c2:4b:27:d6:cd:2b
```

POST (***kwargs*)

Easily generate keys for a minion and auto-accept the new key

Accepts all the same parameters as the `key.gen_accept`.

Note: A note about `curl` Avoid using the `-i` flag or HTTP headers will be written and produce an invalid tar file.

Example partial kickstart script to bootstrap a new minion:

```
%post
mkdir -p /etc/salt/pki/minion
curl -sSk https://localhost:8000/keys \
        -d mid=jerry \
        -d username=kickstart \
        -d password=kickstart \
        -d eauth=pam \
    | tar -C /etc/salt/pki/minion -xf -

mkdir -p /etc/salt/minion.d
printf 'master: 10.0.0.5\nid: jerry' > /etc/salt/minion.d/id.conf
%end
```

POST /keys

Generate a public and private key and return both as a tarball

Authentication credentials must be passed in the request.

Status Codes

- **200** – success
- **401** – authentication required
- **406** – requested Content-Type not available

Example request:

```
curl -sSk https://localhost:8000/keys \
        -d mid=jerry \
        -d username=kickstart \
        -d password=kickstart \
        -d eauth=pam \
        -o jerry-salt-keys.tar
```

```
POST /keys HTTP/1.1
Host: localhost:8000
```

Example response:

```
HTTP/1.1 200 OK
Content-Length: 10240
```

(continues on next page)

(continued from previous page)

```
Content-Disposition: attachment; filename="saltkeys-jerry.tar"
Content-Type: application/x-tar

jerry.pub0000644000000000000000000000070300000000000010730 0ustar ␣
↪00000000000000
```

/ws

class salt.netapi.rest_cherrypy.app.**WebsocketEndpoint**

Open a WebSocket connection to Salt's event bus

The event bus on the Salt master exposes a large variety of things, notably when executions are started on the master and also when minions ultimately return their results. This URL provides a real-time window into a running Salt infrastructure. Uses websocket as the transport mechanism.

See also:

Events & Reactor

GET(*token=None*, ***kwargs*)

Return a websocket connection of Salt's event stream

GET /ws/(*token*)

Query format_events The event stream will undergo server-side formatting if the `format_events` URL parameter is included in the request. This can be useful to avoid formatting on the client-side:

```
curl -NsS <...snip...> localhost:8000/ws?format_events
```

Reqheader X-Auth-Token an authentication token from `Login`.

Status 101 switching to the websockets protocol

Status 401 authentication required

Status 406 requested Content-Type not available

Example request:

```
curl -NsSk \
    -H 'X-Auth-Token: ffedf49d' \
    -H 'Host: localhost:8000' \
    -H 'Connection: Upgrade' \
    -H 'Upgrade: websocket' \
    -H 'Origin: https://localhost:8000' \
    -H 'Sec-WebSocket-Version: 13' \
    -H 'Sec-WebSocket-Key: '"$(echo -n $RANDOM | base64)" \
    localhost:8000/ws
```

```
GET /ws HTTP/1.1
Connection: Upgrade
Upgrade: websocket
Host: localhost:8000
Origin: https://localhost:8000
Sec-WebSocket-Version: 13
Sec-WebSocket-Key: s65VsgHigh7v/Jcf4nXHnA==
X-Auth-Token: ffedf49d
```

Example response:

```
HTTP/1.1 101 Switching Protocols
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Accept: mWZjBV9FCglzn1rIKJAxrTFlnJE=
Sec-WebSocket-Version: 13
```

An authentication token **may optionally** be passed as part of the URL for browsers that cannot be configured to send the authentication header or cookie:

```
curl -NsS <...snip...> localhost:8000/ws/ffedf49d
```

The event stream can be easily consumed via JavaScript:

```
// Note, you must be authenticated!
var source = new Websocket('ws://localhost:8000/ws/d0ce6c1a');
source.onerror = function(e) { console.debug('error!', e); };
source.onmessage = function(e) { console.debug(e.data); };

source.send('websocket client ready')

source.close();
```

Or via Python, using the Python module websocket-client for example.

```
# Note, you must be authenticated!

from websocket import create_connection

ws = create_connection('ws://localhost:8000/ws/d0ce6c1a')
ws.send('websocket client ready')

# Look at https://pypi.python.org/pypi/websocket-client/ for more
# examples.
while listening_to_events:
    print ws.recv()

ws.close()
```

Above examples show how to establish a websocket connection to Salt and activating real time updates from Salt's event stream by signaling `websocket client ready`.

/stats

class `salt.netapi.rest_cherrypy.app.`**Stats**

Expose statistics on the running CherryPy server

GET()

Return a dump of statistics collected from the CherryPy server

GET /stats

Request Headers

- **X-Auth-Token** – a session token from *Login*.
- **Accept** – the desired response format.

Response Headers

- **Content-Type** – the format of the response body; depends on the *Accept* request header.

Status Codes

- **200** – success
- **401** – authentication required
- **406** – requested Content-Type not available

21.10.2 rest_tornado

A non-blocking REST API for Salt

depends

- tornado Python module

configuration All authentication is done through Salt's *external auth* system which requires additional configuration not described here.

In order to run rest_tornado with the salt-master add the following to the Salt master config file.

```
rest_tornado:
    # can be any port
    port: 8000
    # address to bind to (defaults to 0.0.0.0)
    address: 0.0.0.0
    # socket backlog
    backlog: 128
    ssl_crt: /etc/pki/api/certs/server.crt
    # no need to specify ssl_key if cert and key
    # are in one single file
    ssl_key: /etc/pki/api/certs/server.key
    debug: False
    disable_ssl: False
    webhook_disable_auth: False
    cors_origin: null
```

Authentication

Authentication is performed by passing a session token with each request. Tokens are generated via the `SaltAuthHandler` URL.

The token may be sent in one of two ways:

- Include a custom header named `X-Auth-Token`.
- Sent via a cookie. This option is a convenience for HTTP clients that automatically handle cookie support (such as browsers).

See also:

You can bypass the session handling via the `RunSaltAPIHandler` URL.

CORS

rest_tornado supports Cross-site HTTP requests out of the box. It is by default deactivated and controlled by the *cors_origin* config key.

You can allow all origins by settings *cors_origin* to *.

You can allow only one origin with this configuration:

```
rest_tornado:
    cors_origin: http://salt.yourcompany.com
```

You can also be more specific and select only a few allowed origins by using a list. For example:

```
rest_tornado:
    cors_origin:
        - http://salt.yourcompany.com
        - http://salt-preprod.yourcampany.com
```

The format for origin are full URL, with both scheme and port if not standard.

In this case, rest_tornado will check if the Origin header is in the allowed list if it's the case allow the origin. Else it will returns nothing, effectively preventing the origin to make request.

For reference, CORS is a mechanism used by browser to allow (or disallow) requests made from browser from a different origin than salt-api. It's complementary to Authentication and mandatory only if you plan to use a salt client developed as a Javascript browser application.

Usage

Commands are sent to a running Salt master via this module by sending HTTP requests to the URLs detailed below.

Content negotiation

This REST interface is flexible in what data formats it will accept as well as what formats it will return (e.g., JSON, YAML, x-www-form-urlencoded).

- Specify the format of data in the request body by including the `Content-Type` header.
- Specify the desired data format for the response body with the `Accept` header.

Data sent in POST and PUT requests must be in the format of a list of lowstate dictionaries. This allows multiple commands to be executed in a single HTTP request.

lowstate A dictionary containing various keys that instruct Salt which command to run, where that command lives, any parameters for that command, any authentication credentials, what returner to use, etc.

Salt uses the lowstate data format internally in many places to pass command data between functions. Salt also uses lowstate for the *LocalClient()* Python API interface.

The following example (in JSON format) causes Salt to execute two commands:

```
[{
    "client": "local",
    "tgt": "*",
    "fun": "test.fib",
    "arg": ["10"]
},
{
    "client": "runner",
    "fun": "jobs.lookup_jid",
    "jid": "20130603122505459265"
}]
```

Multiple commands in a Salt API request will be executed in serial and makes no gaurantees that all commands will run. Meaning that if test.fib (from the example above) had an exception, the API would still execute "jobs.lookup_jid".

Responses to these lowstates are an in-order list of dicts containing the return data, a yaml response could look like:

```
- ms-1: true
  ms-2: true
- ms-1: foo
  ms-2: bar
```

In the event of an exception while executing a command the return for that lowstate will be a string, for example if no minions matched the first lowstate we would get a return like:

```
- No minions matched the target. No command was sent, no jid was assigned.
- ms-1: true
  ms-2: true
```

---

x-www-form-urlencoded

Sending JSON or YAML in the request body is simple and most flexible, however sending data in urlencoded format is also supported with the caveats below. It is the default format for HTML forms, many JavaScript libraries, and the **`curl`** command.

For example, the equivalent to running `salt '*' test.ping` is sending `fun=test.ping&arg&client=local&tgt=*` in the HTTP request body.

Caveats:

- Only a single command may be sent per HTTP request.
- Repeating the `arg` parameter multiple times will cause those parameters to be combined into a single list.

  Note, some popular frameworks and languages (notably jQuery, PHP, and Ruby on Rails) will automatically append empty brackets onto repeated parameters. E.g., `arg=one`, `arg=two` will be sent as `arg[]=one`, `arg[]=two`. This is not supported; send JSON or YAML instead.

A Websockets add-on to saltnado

depends

- tornado Python module

In order to enable saltnado_websockets you must add websockets: True to your saltnado config block.

```
rest_tornado:
    # can be any port
    port: 8000
    ssl_crt: /etc/pki/api/certs/server.crt
    # no need to specify ssl_key if cert and key
    # are in one single file
    ssl_key: /etc/pki/api/certs/server.key
    debug: False
    disable_ssl: False
    websockets: True
```

All Events

Exposes `all` "real-time" events from Salt's event bus on a websocket connection. It should be noted that "Real-time" here means these events are made available to the server as soon as any salt related action (changes to minions, new jobs etc) happens. Clients are however assumed to be able to tolerate any network transport related latencies. Functionality provided by this endpoint is similar to the `/events` end point.

The event bus on the Salt master exposes a large variety of things, notably when executions are started on the master and also when minions ultimately return their results. This URL provides a real-time window into a running Salt infrastructure. Uses websocket as the transport mechanism.

Exposes GET method to return websocket connections. All requests should include an auth token. A way to obtain obtain authentication tokens is shown below.

```
% curl -si localhost:8000/login \
    -H "Accept: application/json" \
    -d username='salt' \
    -d password='salt' \
    -d eauth='pam'
```

Which results in the response

```
{
    "return": [{
        "perms": [".*", "@runner", "@wheel"],
        "start": 1400556492.277421,
        "token": "d0ce6c1a37e99dcc0374392f272fe19c0090cca7",
        "expire": 1400599692.277422,
        "user": "salt",
        "eauth": "pam"
    }]
}
```

In this example the `token` returned is `d0ce6c1a37e99dcc0374392f272fe19c0090cca7` and can be included in subsequent websocket requests (as part of the URL).

The event stream can be easily consumed via JavaScript:

```
// Note, you must be authenticated!

// Get the Websocket connection to Salt
var source = new Websocket('wss://localhost:8000/all_events/
↪d0ce6c1a37e99dcc0374392f272fe19c0090cca7');

// Get Salt's "real time" event stream.
source.onopen = function() { source.send('websocket client ready'); };

// Other handlers
source.onerror = function(e) { console.debug('error!', e); };

// e.data represents Salt's "real time" event data as serialized JSON.
source.onmessage = function(e) { console.debug(e.data); };

// Terminates websocket connection and Salt's "real time" event stream on the server.
source.close();
```

Or via Python, using the Python module websocket-client for example. Or the tornado client.

```
# Note, you must be authenticated!

from websocket import create_connection

# Get the Websocket connection to Salt
ws = create_connection('wss://localhost:8000/all_events/
↪d0ce6c1a37e99dcc0374392f272fe19c0090cca7')

# Get Salt's "real time" event stream.
ws.send('websocket client ready')


# Simple listener to print results of Salt's "real time" event stream.
# Look at https://pypi.python.org/pypi/websocket-client/ for more examples.
while listening_to_events:
    print ws.recv()       #  Salt's "real time" event data as serialized JSON.

# Terminates websocket connection and Salt's "real time" event stream on the server.
ws.close()

# Please refer to https://github.com/liris/websocket-client/issues/81 when using a
↪self signed cert
```

Above examples show how to establish a websocket connection to Salt and activating real time updates from Salt's event stream by signaling `websocket client ready`.

Formatted Events

Exposes `formatted` "real-time" events from Salt's event bus on a websocket connection. It should be noted that "Real-time" here means these events are made available to the server as soon as any salt related action (changes to minions, new jobs etc) happens. Clients are however assumed to be able to tolerate any network transport related latencies. Functionality provided by this endpoint is similar to the `/events` end point.

The event bus on the Salt master exposes a large variety of things, notably when executions are started on the master and also when minions ultimately return their results. This URL provides a real-time window into a running Salt infrastructure. Uses websocket as the transport mechanism.

Formatted events parses the raw "real time" event stream and maintains a current view of the following:

- minions
- jobs

A change to the minions (such as addition, removal of keys or connection drops) or jobs is processed and clients are updated. Since we use salt's presence events to track minions, please enable `presence_events` and set a small value for the `loop_interval` in the salt master config file.

Exposes GET method to return websocket connections. All requests should include an auth token. A way to obtain obtain authentication tokens is shown below.

```
% curl -si localhost:8000/login \
    -H "Accept: application/json" \
    -d username='salt' \
    -d password='salt' \
    -d eauth='pam'
```

Which results in the response

```
{
    "return": [{
        "perms": [".*", "@runner", "@wheel"],
        "start": 1400556492.277421,
        "token": "d0ce6c1a37e99dcc0374392f272fe19c0090cca7",
        "expire": 1400599692.277422,
        "user": "salt",
        "eauth": "pam"
    }]
}
```

In this example the `token` returned is `d0ce6c1a37e99dcc0374392f272fe19c0090cca7` and can be included in subsequent websocket requests (as part of the URL).

The event stream can be easily consumed via JavaScript:

```
// Note, you must be authenticated!

// Get the Websocket connection to Salt
var source = new Websocket('wss://localhost:8000/formatted_events/
↪d0ce6c1a37e99dcc0374392f272fe19c0090cca7');

// Get Salt's "real time" event stream.
source.onopen = function() { source.send('websocket client ready'); };

// Other handlers
source.onerror = function(e) { console.debug('error!', e); };

// e.data represents Salt's "real time" event data as serialized JSON.
source.onmessage = function(e) { console.debug(e.data); };

// Terminates websocket connection and Salt's "real time" event stream on the server.
source.close();
```

Or via Python, using the Python module websocket-client for example. Or the tornado client.

```
# Note, you must be authenticated!

from websocket import create_connection

# Get the Websocket connection to Salt
ws = create_connection('wss://localhost:8000/formatted_events/
↪d0ce6c1a37e99dcc0374392f272fe19c0090cca7')

# Get Salt's "real time" event stream.
ws.send('websocket client ready')


# Simple listener to print results of Salt's "real time" event stream.
# Look at https://pypi.python.org/pypi/websocket-client/ for more examples.
while listening_to_events:
    print ws.recv()       #  Salt's "real time" event data as serialized JSON.

# Terminates websocket connection and Salt's "real time" event stream on the server.
ws.close()

# Please refer to https://github.com/liris/websocket-client/issues/81 when using a
↪self signed cert
```

Above examples show how to establish a websocket connection to Salt and activating real time updates from Salt's event stream by signaling `websocket client ready`.

Example responses

`Minion information` is a dictionary keyed by each connected minion's `id` (`mid`), grains information for each minion is also included.

Minion information is sent in response to the following minion events:

- **connection drops**
    - requires running `manage.present` periodically every `loop_interval` seconds
- minion addition
- minon removal

```
# Not all grains are shown
data: {
    "minions": {
        "minion1": {
            "id": "minion1",
            "grains": {
                "kernel": "Darwin",
                "domain": "local",
                "zmqversion": "4.0.3",
                "kernelrelease": "13.2.0"
            }
        }
    }
}
```

`Job information` is also tracked and delivered.

Job information is also a dictionary in which each job's information is keyed by salt's `jid`.

```
data: {
    "jobs": {
        "20140609153646699137": {
            "tgt_type": "glob",
            "jid": "20140609153646699137",
            "tgt": "*",
            "start_time": "2014-06-09T15:36:46.700315",
            "state": "complete",
            "fun": "test.ping",
            "minions": {
                "minion1": {
                    "return": true,
                    "retcode": 0,
                    "success": true
                }
            }
        }
    }
}
```

Setup

REST URI Reference

- */*
- */login*
- */minions*
- */jobs*
- */run*
- */events*
- */hook*

/

class `salt.netapi.rest_tornado.saltnado.`**`SaltAPIHandler`**(*application*, *request*, ***kwargs*)

Main API handler for base "/"

`disbatch`()

Disbatch all lowstates to the appropriate clients

`get`()

An endpoint to determine salt-api capabilities

`GET /`

Request Headers

- **`Accept`** – the desired response format.

Status Codes

- **`200`** – success
- **`401`** – authentication required
- **`406`** – requested Content-Type not available

Example request:

```
curl -i localhost:8000
```

```
GET / HTTP/1.1
Host: localhost:8000
Accept: application/json
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Legnth: 83

{"clients": ["local", "local_async", "runner", "runner_async"], "return":
↪"Welcome"}
```

`post`()

Send one or more Salt commands (lowstates) in the request body

POST /

Request Headers

- **X-Auth-Token** – a session token from *SaltAuthHandler*.
- **Accept** – the desired response format.
- **Content-Type** – the format of the request body.

Response Headers

- **Content-Type** – the format of the response body; depends on the *Accept* request header.

Status Codes

- **200** – success
- **401** – authentication required
- **406** – requested Content-Type not available

lowstate data describing Salt commands must be sent in the request body.

Example request:

```
curl -si https://localhost:8000 \
        -H "Accept: application/x-yaml" \
        -H "X-Auth-Token: d40d1e1e" \
        -d client=local \
        -d tgt='*' \
        -d fun='test.ping' \
        -d arg
```

```
POST / HTTP/1.1
Host: localhost:8000
Accept: application/x-yaml
X-Auth-Token: d40d1e1e
Content-Length: 36
Content-Type: application/x-www-form-urlencoded

fun=test.ping&arg&client=local&tgt=*
```

Example response:

Responses are an in-order list of the lowstate's return data. In the event of an exception running a command the return will be a string instead of a mapping.

```
HTTP/1.1 200 OK
Content-Length: 200
Allow: GET, HEAD, POST
Content-Type: application/x-yaml

return:
- ms-0: true
    ms-1: true
    ms-2: true
    ms-3: true
    ms-4: true
```

---

multiple commands

Note that if multiple *lowstate* structures are sent, the Salt API will execute them in serial, and will not stop execution upon failure of a previous job. If you need to have commands executed in order and stop on failure please use compound-command-execution.

`/login`

class `salt.netapi.rest_tornado.saltnado.`**`SaltAuthHandler`**(*application*, *request*, ***kwargs*)

Handler for login requests

`get`()

All logins are done over post, this is a parked endpoint

GET /login

Status Codes

- **401** – authentication required
- **406** – requested Content-Type not available

Example request:

```
curl -i localhost:8000/login
```

```
GET /login HTTP/1.1
Host: localhost:8000
Accept: application/json
```

Example response:

```
HTTP/1.1 401 Unauthorized
Content-Type: application/json
Content-Length: 58

{"status": "401 Unauthorized", "return": "Please log in"}
```

`post`()

Authenticate against Salt's eauth system

POST /login

Request Headers

- **X-Auth-Token** – a session token from `SaltAuthHandler`.
- **Accept** – the desired response format.
- **Content-Type** – the format of the request body.

Form Parameters

- **eauth** – the eauth backend configured for the user
- **username** – username
- **password** – password

Status Codes

- **200** – success
- **400** – bad request
- **401** – authentication required
- **406** – requested Content-Type not available
- **500** – internal server error

Example request:

```
curl -si localhost:8000/login \
        -H "Accept: application/json" \
        -d username='saltuser' \
        -d password='saltpass' \
        -d eauth='pam'
```

```
POST / HTTP/1.1
Host: localhost:8000
Content-Length: 42
Content-Type: application/x-www-form-urlencoded
Accept: application/json

username=saltuser&password=saltpass&eauth=pam
```

Example response:

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 206
X-Auth-Token: 6d1b722e
Set-Cookie: session_id=6d1b722e; expires=Sat, 17 Nov 2012 03:23:52 GMT;
↪Path=/

{"return": {
    "token": "6d1b722e",
    "start": 1363805943.776223,
    "expire": 1363849143.776224,
    "user": "saltuser",
    "eauth": "pam",
    "perms": [
        "grains.*",
        "status.*",
        "sys.*",
        "test.*"
    ]
}}
```

`/minions`

class `salt.netapi.rest_tornado.saltnado.`**`MinionSaltAPIHandler`**(*application*, *request*, ***kwargs*)

A convenience endpoint for minion related functions

`get`(*mid=None*)

A convenience URL for getting lists of minions or getting minion details

`GET /minions/`(*mid*)

Request Headers

- **`X-Auth-Token`** – a session token from `SaltAuthHandler`.
- **`Accept`** – the desired response format.

Status Codes

- **`200`** – success
- **`401`** – authentication required
- **`406`** – requested Content-Type not available

Example request:

```
curl -i localhost:8000/minions/ms-3
```

```
GET /minions/ms-3 HTTP/1.1
Host: localhost:8000
Accept: application/x-yaml
```

Example response:

```
HTTP/1.1 200 OK
Content-Length: 129005
Content-Type: application/x-yaml

return:
- ms-3:
    grains.items:
        ...
```

post()

Start an execution command and immediately return the job id

POST /minions

Request Headers

- **X-Auth-Token** – a session token from *SaltAuthHandler*.
- **Accept** – the desired response format.
- **Content-Type** – the format of the request body.

Response Headers

- **Content-Type** – the format of the response body; depends on the *Accept* request header.

Status Codes

- **200** – success
- **401** – authentication required
- **406** – requested Content-Type not available

lowstate data describing Salt commands must be sent in the request body. The `client` option will be set to `local_async()`.

Example request:

```
curl -sSi localhost:8000/minions \
    -H "Accept: application/x-yaml" \
    -d tgt='*' \
    -d fun='status.diskusage'
```

```
POST /minions HTTP/1.1
Host: localhost:8000
Accept: application/x-yaml
Content-Length: 26
Content-Type: application/x-www-form-urlencoded

tgt=*&fun=status.diskusage
```

Example response:

```
HTTP/1.1 202 Accepted
Content-Length: 86
Content-Type: application/x-yaml

return:
- jid: '20130603122505459265'
  minions: [ms-4, ms-3, ms-2, ms-1, ms-0]
```

/jobs

class salt.netapi.rest_tornado.saltnado.**JobsSaltAPIHandler**(*application*, *request*, ***kwargs*)

A convenience endpoint for job cache data

get(*jid=None*)

A convenience URL for getting lists of previously run jobs or getting the return from a single job

GET /jobs/(*jid*)

List jobs or show a single job from the job cache.

Status Codes

- **200** – success
- **401** – authentication required
- **406** – requested Content-Type not available

Example request:

```
curl -i localhost:8000/jobs
```

```
GET /jobs HTTP/1.1
Host: localhost:8000
Accept: application/x-yaml
```

Example response:

```
HTTP/1.1 200 OK
Content-Length: 165
Content-Type: application/x-yaml

return:
- '20121130104633606931':
    Arguments:
    - '3'
    Function: test.fib
    Start Time: 2012, Nov 30 10:46:33.606931
    Target: jerry
    Target-type: glob
```

Example request:

```
curl -i localhost:8000/jobs/20121130104633606931
```

```
GET /jobs/20121130104633606931 HTTP/1.1
Host: localhost:8000
Accept: application/x-yaml
```

Example response:

```
HTTP/1.1 200 OK
Content-Length: 73
Content-Type: application/x-yaml

info:
- Arguments:
    - '3'
    Function: test.fib
```

(continues on next page)

(continued from previous page)

```
    Minions:
    - jerry
    Start Time: 2012, Nov 30 10:46:33.606931
    Target: '*'
    Target-type: glob
    User: saltdev
    jid: '20121130104633606931'
return:
- jerry:
    - - 0
    - 1
    - 1
    - 2
    - 6.9141387939453125e-06
```

/run

class `salt.netapi.rest_tornado.saltnado.`**`RunSaltAPIHandler`**(*application*, *request*, ***kwargs*)

Endpoint to run commands without normal session handling

`post()`

Run commands bypassing the *normal session handling*

POST /run

This entry point is primarily for ”one-off” commands. Each request must pass full Salt authentication credentials. Otherwise this URL is identical to the `root URL (/)`.

lowstate data describing Salt commands must be sent in the request body.

Status Codes

- **200** – success
- **401** – authentication required
- **406** – requested Content-Type not available

Example request:

```
curl -sS localhost:8000/run \
    -H 'Accept: application/x-yaml' \
    -d client='local' \
    -d tgt='*' \
    -d fun='test.ping' \
    -d username='saltdev' \
    -d password='saltdev' \
    -d eauth='pam'
```

```
POST /run HTTP/1.1
Host: localhost:8000
Accept: application/x-yaml
Content-Length: 75
Content-Type: application/x-www-form-urlencoded

client=local&tgt=*&fun=test.ping&username=saltdev&password=saltdev&eauth=pam
```

Example response:

```
HTTP/1.1 200 OK
Content-Length: 73
Content-Type: application/x-yaml

return:
- ms-0: true
    ms-1: true
    ms-2: true
    ms-3: true
    ms-4: true
```

`/events`

class `salt.netapi.rest_tornado.saltnado.`**`EventsSaltAPIHandler`**(*application*, *request*, ***kwargs*)

Expose the Salt event bus

The event bus on the Salt master exposes a large variety of things, notably when executions are started on the master and also when minions ultimately return their results. This URL provides a real-time window into a running Salt infrastructure.

See also:

Events & Reactor

get()

An HTTP stream of the Salt master event bus

This stream is formatted per the Server Sent Events (SSE) spec. Each event is formatted as JSON.

GET /events

Status Codes

- **200** – success
- **401** – authentication required
- **406** – requested Content-Type not available

Example request:

```
curl -NsS localhost:8000/events
```

```
GET /events HTTP/1.1
Host: localhost:8000
```

Example response:

```
HTTP/1.1 200 OK
Connection: keep-alive
Cache-Control: no-cache
Content-Type: text/event-stream;charset=utf-8

retry: 400
data: {'tag': '', 'data': {'minions': ['ms-4', 'ms-3', 'ms-2', 'ms-1', 'ms-0
↪']}}

data: {'tag': '20130802115730568475', 'data': {'jid': '20130802115730568475',
↪ 'return': True, 'retcode': 0, 'success': True, 'cmd': '_return', 'fun':
↪'test.ping', 'id': 'ms-1'}}
```

The event stream can be easily consumed via JavaScript:

```
<!-- Note, you must be authenticated! -->
var source = new EventSource('/events');
source.onopen = function() { console.debug('opening') };
source.onerror = function(e) { console.debug('error!', e) };
source.onmessage = function(e) { console.debug(e.data) };
```

Or using CORS:

```
var source = new EventSource('/events', {withCredentials: true});
```

Some browser clients lack CORS support for the `EventSource()` API. Such clients may instead pass the *`X-Auth-Token`* value as an URL parameter:

```
curl -NsS localhost:8000/events/6d1b722e
```

It is also possible to consume the stream via the shell.

Records are separated by blank lines; the `data:` and `tag:` prefixes will need to be removed manually before attempting to unserialize the JSON.

curl's `-N` flag turns off input buffering which is required to process the stream incrementally.

Here is a basic example of printing each event as it comes in:

```
curl -NsS localhost:8000/events |\
        while IFS= read -r line ; do
            echo $line
        done
```

Here is an example of using awk to filter events based on tag:

```
curl -NsS localhost:8000/events |\
        awk '
            BEGIN { RS=""; FS="\\n" }
            $1 ~ /^tag: salt\/job\/[0-9]+\/new$/ { print $0 }
        '
tag: salt/job/20140112010149808995/new
data: {"tag": "salt/job/20140112010149808995/new", "data": {"tgt_type": "glob
↪", "jid": "20140112010149808995", "tgt": "jerry", "_stamp": "2014-01-12_
↪01:01:49.809617", "user": "shouse", "arg": [], "fun": "test.ping", "minions
↪": ["jerry"]}}
tag: 20140112010149808995
data: {"tag": "20140112010149808995", "data": {"fun_args": [], "jid":
↪"20140112010149808995", "return": true, "retcode": 0, "success": true, "cmd
↪": "_return", "_stamp": "2014-01-12_01:01:49.819316", "fun": "test.ping",
↪"id": "jerry"}}
```

/hook

class `salt.netapi.rest_tornado.saltnado.`**`WebhookSaltAPIHandler`**(*application*, *request*, ***kwargs*)

A generic web hook entry point that fires an event on Salt's event bus

External services can POST data to this URL to trigger an event in Salt. For example, Amazon SNS, Jenkins-CI or Travis-CI, or GitHub web hooks.

Note: Be mindful of security

Salt's Reactor can run any code. A Reactor SLS that responds to a hook event is responsible for validating that the event came from a trusted source and contains valid data.

This is a generic interface and securing it is up to you!

This URL requires authentication however not all external services can be configured to authenticate. For this reason authentication can be selectively disabled for this URL. Follow best practices – always use SSL, pass a secret key, configure the firewall to only allow traffic from a known source, etc.

The event data is taken from the request body. The *`Content-Type`* header is respected for the payload.

The event tag is prefixed with `salt/netapi/hook` and the URL path is appended to the end. For example, a POST request sent to `/hook/mycompany/myapp/mydata` will produce a Salt event with the tag `salt/netapi/hook/mycompany/myapp/mydata`.

The following is an example `.travis.yml` file to send notifications to Salt of successful test runs:

```
language: python
script: python -m unittest tests
after_success:
    - 'curl -sS http://saltapi-url.example.com:8000/hook/travis/build/success -d␣
↪branch="${TRAVIS_BRANCH}" -d commit="${TRAVIS_COMMIT}"'
```

See also:

Events, *Reactor*

`post`(*tag_suffix=None*)

Fire an event in Salt with a custom event tag and data

`POST /hook`

Status Codes

- **`200`** – success
- **`401`** – authentication required
- **`406`** – requested Content-Type not available
- **`413`** – request body is too large

Example request:

```
curl -sS localhost:8000/hook -d foo='Foo!' -d bar='Bar!'
```

```
POST /hook HTTP/1.1
Host: localhost:8000
Content-Length: 16
Content-Type: application/x-www-form-urlencoded

foo=Foo&bar=Bar!
```

Example response:

```
HTTP/1.1 200 OK
Content-Length: 14
Content-Type: application/json

{"success": true}
```

As a practical example, an internal continuous-integration build server could send an HTTP POST request to the URL `http://localhost:8000/hook/mycompany/build/success` which contains the result of a build and the SHA of the version that was built as JSON. That would then produce the following event in Salt that could be used to kick off a deployment via Salt's Reactor:

```
Event fired at Fri Feb 14 17:40:11 2014
*************************
Tag: salt/netapi/hook/mycompany/build/success
Data:
{'_stamp': '2014-02-14_17:40:11.440996',
    'headers': {
        'X-My-Secret-Key': 'F0fAgoQjIT@W',
        'Content-Length': '37',
        'Content-Type': 'application/json',
        'Host': 'localhost:8000',
        'Remote-Addr': '127.0.0.1'},
    'post': {'revision': 'aa22a3c4b2e7', 'result': True}}
```

Salt's Reactor could listen for the event:

```
reactor:
  - 'salt/netapi/hook/mycompany/build/*':
    - /srv/reactor/react_ci_builds.sls
```

And finally deploy the new build:

```
{% set secret_key = data.get('headers', {}).get('X-My-Secret-Key') %}
{% set build = data.get('post', {}) %}

{% if secret_key == 'F0fAgoQjIT@W' and build.result == True %}
deploy_my_app:
  cmd.state.sls:
    - tgt: 'application*'
    - arg:
      - myapp.deploy
    - kwarg:
        pillar:
          revision: {{ revision }}
{% endif %}
```

21.10.3 rest_wsgi

A minimalist REST API for Salt

This `rest_wsgi` module provides a no-frills REST interface for sending commands to the Salt master. There are no dependencies.

Extra care must be taken when deploying this module into production. Please read this documentation in entirety.

All authentication is done through Salt's *external auth* system.

Usage

- All requests must be sent to the root URL (/).
- All requests must be sent as a POST request with JSON content in the request body.

- All responses are in JSON.

See also:

rest_cherrypy

The *rest_cherrypy* module is more full-featured, production-ready, and has builtin security features.

Deployment

The `rest_wsgi` netapi module is a standard Python WSGI app. It can be deployed one of two ways.

Using a WSGI-compliant web server

This module may be run via any WSGI-compliant production server such as Apache with mod_wsgi or Nginx with FastCGI.

It is strongly recommended that this app be used with a server that supports HTTPS encryption since raw Salt authentication credentials must be sent with every request. Any apps that access Salt through this interface will need to manually manage authentication credentials (either username and password or a Salt token). Tread carefully.

`salt-api` using a development-only server

If run directly via the salt-api daemon it uses the wsgiref.simple_server() that ships in the Python standard library. This is a single-threaded server that is intended for testing and development. **This server does not use encryption**; please note that raw Salt authentication credentials must be sent with every HTTP request.

Running this module via salt-api is not recommended!

In order to start this module via the `salt-api` daemon the following must be put into the Salt master config:

```
rest_wsgi:
    port: 8001
```

Usage examples

POST /

Example request for a basic `test.ping`:

```
% curl -sS -i \
        -H 'Content-Type: application/json' \
        -d '[{"eauth":"pam","username":"saltdev","password":"saltdev","client":
↪"local","tgt":"*","fun":"test.ping"}]' localhost:8001
```

Example response:

```
HTTP/1.0 200 OK
Content-Length: 89
Content-Type: application/json

{"return": [{"ms--4": true, "ms--3": true, "ms--2": true, "ms--1": true, "ms--0":
↪true}]}
```

Example request for an asynchronous `test.ping`:

```
% curl -sS -i \
        -H 'Content-Type: application/json' \
        -d '[{"eauth":"pam","username":"saltdev","password":"saltdev","client":
↪"local_async","tgt":"*","fun":"test.ping"}]' localhost:8001
```

Example response:

```
HTTP/1.0 200 OK
Content-Length: 103
Content-Type: application/json

{"return": [{"jid": "20130412192112593739", "minions": ["ms--4", "ms--3", "ms--2",
↪ "ms--1", "ms--0"]}]}
```

Example request for looking up a job ID:

```
% curl -sS -i \
        -H 'Content-Type: application/json' \
        -d '[{"eauth":"pam","username":"saltdev","password":"saltdev","client":
↪"runner","fun":"jobs.lookup_jid","jid":"20130412192112593739"}]' localhost:8001
```

Example response:

```
HTTP/1.0 200 OK
Content-Length: 89
Content-Type: application/json

{"return": [{"ms--4": true, "ms--3": true, "ms--2": true, "ms--1": true, "ms--0":
↪true}]}
```

form lowstate A list of lowstate data appropriate for the *client* interface you are calling.

status 200 success

status 401 authentication required

21.11 output modules

Follow one of the below links for further information and examples

highstate	Outputter for displaying results of state runs
json_out	Display return data in JSON format
key	Display salt-key output
nested	Recursively display nested data
newline_values_only	Display values only, separated by newlines
no_out_quiet	Display no output
no_return	Display output for minions that did not return
overstatestage	Display clean output of an overstate stage
pony	Display Pony output data structure
pprint_out	Python pretty-print (pprint)
profile	Display profiling data in a table format
progress	Display return data as a progress bar
raw	Display raw output data structure

Continued on next page

Table 11 – continued from previous page

table_out	Display output in a table format
txt	Simple text outputter
virt_query	virt.query outputter
yaml_out	Display return data in YAML format

21.11.1 salt.output.highstate

Outputter for displaying results of state runs

The return data from the Highstate command is a standard data structure which is parsed by the highstate outputter to deliver a clean and readable set of information about the HighState run on minions.

Two configurations can be set to modify the highstate outputter. These values can be set in the master config to change the output of the `salt` command or set in the minion config to change the output of the `salt-call` command.

state_verbose By default *state_verbose* is set to *True*, setting this to *False* will instruct the highstate outputter to omit displaying anything in green, this means that nothing with a result of True and no changes will not be printed

state_output: The highstate outputter has six output modes, `full`, `terse`, `mixed`, `changes` and `filter`

- The default is set to `full`, which will display many lines of detailed information for each executed chunk.
- If `terse` is used, then the output is greatly simplified and shown in only one line.
- If `mixed` is used, then terse output will be used unless a state failed, in which case full output will be used.
- If `changes` is used, then terse output will be used if there was no error and no changes, otherwise full output will be used.
- If `filter` is used, then either or both of two different filters can be used: `exclude` or `terse`.
    - for `exclude`, state.highstate expects a list of states to be excluded (or `None`) followed by `True` for terse output or `False` for regular output. Because of parsing nuances, if only one of these is used, it must still contain a comma. For instance: *exclude=True,*.
    - for `terse`, state.highstate expects simply `True` or `False`.

  These can be set as such from the command line, or in the Salt config as *state_output_exclude* or *state_output_terse*, respectively.

The output modes have one modifier:

`full_id`, `terse_id`, `mixed_id`, `changes_id` and `filter_id` If `_id` is used, then the corresponding form will be used, but the value for `name` will be drawn from the state ID. This is useful for cases where the name value might be very long and hard to read.

state_tabular: If *state_output* uses the terse output, set this to *True* for an aligned output format. If you wish to use a custom format, this can be set to a string.

Example usage:

If `state_output: filter` is set in the configuration file:

```
salt '*' state.highstate exclude=None,True
```

means to exclude no states from the highstate and turn on terse output.

```
salt twd state.highstate exclude=problemstate1,problemstate2,False
```

means to exclude states `problemstate1` and `problemstate2` from the highstate, and use regular output.

Example output for the above highstate call when `top.sls` defines only one other state to apply to minion `twd`:

```
twd:

Summary for twd
------------
Succeeded: 1 (changed=1)
Failed:    0
------------
Total states run:     1
```

Example output with no special settings in configuration files:

```
myminion:
----------
          ID: test.ping
    Function: module.run
      Result: True
     Comment: Module function test.ping executed
     Changes:
              ----------
              ret:
                  True

Summary for myminion
------------
Succeeded: 1
Failed:    0
------------
Total:     0
```

`salt.output.highstate.`**`output`**(*data*, ***kwargs*)

The HighState Outputter is only meant to be used with the state.highstate function, or a function that returns highstate return data.

21.11.2 salt.output.json_out

Display return data in JSON format

configuration The output format can be configured in two ways: Using the `--out-indent` CLI flag and specifying a positive integer or a negative integer to group JSON from each minion to a single line.

Or setting the `output_indent` setting in the Master or Minion configuration file with one of the following values:

- `Null`: put each minion return on a single line.
- `pretty`: use four-space indents and sort the keys.
- An integer: specify the indentation level.

Salt's outputters operate on a per-minion basis. Each minion return will be output as a single JSON object once it comes in to the master.

Some JSON parsers can guess when an object ends and a new one begins but many can not. A good way to differentiate between each minion return is to use the single-line output format and to parse each line individually. Example output (truncated):

```
{"dave": {"en0": {"hwaddr": "02:b0:26:32:4c:69", ...}}}
{"jerry": {"en0": {"hwaddr": "02:26:ab:0d:b9:0d", ...}}}
{"kevin": {"en0": {"hwaddr": "02:6d:7f:ce:9f:ee", ...}}}
{"mike": {"en0": {"hwaddr": "02:48:a2:4b:70:a0", ...}}}
{"phill": {"en0": {"hwaddr": "02:1d:cc:a2:33:55", ...}}}
{"stuart": {"en0": {"hwaddr": "02:9a:e0:ea:9e:3c", ...}}}
```

CLI Example:

```
salt '*' foo.bar --out=json
```

`salt.output.json_out.`**`output`**(*data*, ***kwargs*)
Print the output data in JSON

21.11.3 salt.output.key

Display salt-key output

The `salt-key` command makes use of this outputter to format its output.

`salt.output.key.`**`output`**(*data*, ***kwargs*)
Read in the dict structure generated by the salt key API methods and print the structure.

21.11.4 salt.output.nested

Recursively display nested data

This is the default outputter for most execution functions.

Example output:

```
myminion:
    ----------
    foo:
        ----------
        bar:
            baz
        dictionary:
            ----------
            abc:
                123
            def:
                456
        list:
            - Hello
            - World
```

class `salt.output.nested.`**`NestDisplay`**(*retcode=0*)
Manage the nested display contents

`display`(*ret*, *indent*, *prefix*, *out*)
Recursively iterate down through data structures to determine output

ustring(*indent, color, msg, prefix='', suffix='', endc=None*)

`salt.output.nested.`**output**(*ret, **kwargs*)
Display ret data

21.11.5 salt.output.newline_values_only

Display values only, separated by newlines

New in version 2015.5.0.

This outputter is designed for Salt CLI return data. It will do the following to the return dict:

1. Get just the values (ignoring the minion IDs).
2. Each value, if it is iterable, is split a separate line.
3. Each minion's values are separated by newlines.

This results in a single string of return data containing all the values from the various minions.

Warning: As noted above, this outputter will discard the minion ID. If the minion ID is important, then an outputter that returns the full return dictionary in a parsable format (such as `json`, `pprint,`, or `yaml`) may be more suitable.

Example 1

```
salt '*' foo.bar --out=newline_values_only
```

Input

```
{
    'myminion': ['127.0.0.1', '10.0.0.1'],
    'second-minion': ['127.0.0.1', '10.0.0.2']
}
```

Output

```
127.0.0.1
10.0.0.1
127.0.0.1
10.0.0.2
```

Example 2

```
salt '*' foo.bar --out=newline_values_only
```

Input

```
{
    'myminion': 8,
    'second-minion': 10
}
```

Output

```
8
10
```

`salt.output.newline_values_only.`**`output`**(*data*, ***kwargs*)
Display modified ret data

21.11.6 salt.output.no_out_quiet

Display no output

No output is produced when this outputter is selected

CLI Example:

```
salt '*' foo.bar --out=quiet
```

`salt.output.no_out_quiet.`**`output`**(*ret*, ***kwargs*)
Don't display data. Used when you only are interested in the return.

21.11.7 salt.output.no_return

Display output for minions that did not return

This outputter is used to display notices about which minions failed to return when a salt function is run with `-v` or `--verbose`. It should not be called directly from the CLI.

Example output:

```
virtucentos:
    Minion did not return
```

`class` `salt.output.no_return.`**`NestDisplay`**
Create generator for nested output

`display`(*ret*, *indent*, *prefix*, *out*)
Recursively iterate down through data structures to determine output

`salt.output.no_return.`**`output`**(*ret*, ***kwargs*)
Display ret data

21.11.8 salt.output.overstatestage

Display clean output of an overstate stage

This outputter is used to display *Orchestrate Runner* stages, and should not be called directly.

`salt.output.overstatestage.`**`output`**`(`*data*, ***kwargs*`)`
Format the data for printing stage information from the overstate system

21.11.9 salt.output.pony module

Display Pony output data structure

depends

- ponysay CLI program

Display output from a pony. Ponies are better than cows because everybody wants a pony.

Example output:

```
< {'local': True} >
 -----------------
 \
  \
   \
```

CLI Example:

```
salt '*' foo.bar --out=pony
```

`salt.output.pony.`**`output`**`(`*data*, ***kwargs*`)`
Mane function

21.11.10 salt.output.pprint_out

Python pretty-print (pprint)

The python pretty-print system was once the default outputter. It simply passes the return data through to `pprint.pformat` and prints the results.

CLI Example:

```
salt '*' foo.bar --out=pprint
```

Example output:

```
{'saltmine': {'foo': {'bar': 'baz',
                      'dictionary': {'abc': 123, 'def': 456},
                      'list': ['Hello', 'World']}}}
```

`salt.output.pprint_out.`**`output`**`(`*data*, ***kwargs*`)`
Print out via pretty print

21.11.11 salt.output.profile

Display profiling data in a table format

Show profile data for returners that would normally show a highstate output.

CLI Example:

```
salt '*' state.apply something --out=profile
```

Attempt to output the returns of state.sls and state.highstate as a table of names, modules and durations that looks somewhat like the following:

```
name                 mod.fun                duration (ms)
--------------------------------------------------------
I-fail-unless-stmt   other.function               -1.0000
old-minion-config    grains.list_present           1.1200
salt-data            group.present                48.3800
/etc/salt/minion     file.managed                 63.1450
```

To get the above appearance, use settings something like these:

```
out.table.separate_rows: False
out.table.justify: left
out.table.delim: '  '
out.table.prefix: ''
out.table.suffix: ''
```

`salt.output.profile.`**`output`**`(`*data*, ***kwargs*`)`
Display the profiling data in a table format.

21.11.12 salt.output.progress

Display return data as a progress bar

`salt.output.progress.`**`output`**(*ret*, *bar*, ***kwargs*)
Update the progress bar

`salt.output.progress.`**`progress_iter`**(*progress*)
Initialize and return a progress bar iter

21.11.13 salt.output.raw

Display raw output data structure

This outputter simply displays the output as a python data structure, by printing a string representation of it. It is similar to the `pprint` outputter, only the data is not nicely formatted/indented.

This was the original outputter used by Salt before the outputter system was developed.

CLI Example:

```
salt '*' foo.bar --out=raw
```

Example output:

```
salt '*' foo.bar --out=table
{'myminion': {'foo': {'list': ['Hello', 'World'], 'bar': 'baz', 'dictionary': {'abc':
↪123, 'def': 456}}}}
```

`salt.output.raw.`**`output`**(*data*, ***kwargs*)
Rather basic....

21.11.14 salt.output.table_out

Display output in a table format

New in version 2017.7.0.

The `table` outputter displays a sequence of rows as table.

Example output:

```
edge01.bjm01:
----------
    comment:
    ----------
    out:
    ----------
        ______________________________________________________________________________
        | Active | Interface | Last Move |        Mac        | Moves | Static | Vlan |
        ______________________________________________________________________________
        |  True  |  ae1.900  |    0.0    | 40:A6:77:5A:50:01 |   0   | False  | 111  |
        ______________________________________________________________________________
        |  True  |  ae1.111  |    0.0    | 64:16:8D:32:26:58 |   0   | False  | 111  |
        ______________________________________________________________________________
        |  True  |  ae1.111  |    0.0    | 8C:60:4F:73:2D:57 |   0   | False  | 111  |
        ______________________________________________________________________________
        |  True  |  ae1.111  |    0.0    | 8C:60:4F:73:2D:7C |   0   | False  | 111  |
        ______________________________________________________________________________
        |  True  |  ae1.222  |    0.0    | 8C:60:4F:73:2D:57 |   0   | False  | 222  |
```

(continues on next page)

(continued from previous page)

```
        ------------------------------------------------------------------------------
        |  True  |  ae1.222  |    0.0    | F4:0F:1B:76:9D:97 |   0   | False  | 222  |
        ------------------------------------------------------------------------------
    result:
    ----------
```

CLI Example:

```
salt '*' foo.bar --out=table
```

class salt.output.table_out.**TableDisplay**(*has_header=True*, *row_delimiter='-'*, *delim=' | '*, *justify='center'*, *separate_rows=True*, *prefix='| '*, *suffix=' |'*, *width=50*, *wrapfunc=None*)

Manage the table display content.

display(*ret*, *indent*, *out*, *rows_key=None*, *labels_key=None*)

Display table(s).

display_rows(*rows*, *labels*, *indent*)

Prepares row content and displays.

prepare_rows(*rows*, *indent*, *has_header*)

Prepare rows content to be displayed.

ustring(*indent*, *color*, *msg*, *prefix=''*, *suffix=''*, *endc=None*)

Build the unicode string to be displayed.

wrap_onspace(*text*)

When the text inside the column is longer then the width, will split by space and continue on the next line.

salt.output.table_out.**output**(*ret*, ***kwargs*)

Display the output as table.

Parameters

- **nested_indent** (***) – integer, specify the left alignment.
- **has_header** (***) – boolean specifying if header should be displayed. Default: True.
- **row_delimiter** (***) – character to separate rows. Default: `_`.
- **delim** (***) – character to separate columns. Default: `" | "`.
- **justify** (***) – text alignment. Default: `center`.
- **separate_rows** (***) – boolean specifying if row separator will be displayed between consecutive rows. Default: True.
- **prefix** (***) – character at the beginning of the row. Default: `"| "`.
- **suffix** (***) – character at the end of the row. Default: `" |"`.
- **width** (***) – column max width. Default: `50`.
- **rows_key** (***) – display the rows under a specific key.
- **labels_key** (***) – use the labels under a certain key. Otherwise will try to use the dictionary keys (if any).
- **title** (***) – display title when only one table is selected (using the `rows_key` argument).

21.11.15 salt.output.txt

Simple text outputter

The `txt` outputter has been developed to make the output from shell commands on minions appear as they do when the command is executed on the minion.

CLI Example:

```
salt '*' foo.bar --out=txt
```

`salt.output.txt.`**`output`**(*data*, ***kwargs*)
Output the data in lines, very nice for running commands

21.11.16 salt.output.virt_query

virt.query outputter

Used to display the output from the *`virt.query`* runner.

`salt.output.virt_query.`**`output`**(*data*, ***kwargs*)
Display output for the salt-run virt.query function

21.11.17 salt.output.yaml_out

Display return data in YAML format

This outputter defaults to printing in YAML block mode for better readability.

CLI Example:

```
salt '*' foo.bar --out=yaml
```

Example output:

CLI Example:

```
saltmine:
  foo:
    bar: baz
    dictionary:
      abc: 123
      def: 456
    list:
      - Hello
      - World
```

`salt.output.yaml_out.`**`output`**(*data*, ***kwargs*)
Print out YAML using the block mode

21.12 pillar modules

`cmd_json`	Execute a command and read the output as JSON.
`cmd_yaml`	Execute a command and read the output as YAML.
`cmd_yamlex`	Execute a command and read the output as YAMLEX.
`cobbler`	A module to pull data from Cobbler via its API into the Pillar dictionary

Continued on next page

Table 12 – continued from previous page

`confidant`	An external pillar module for getting credentials from confidant.
`consul_pillar`	Use Consul K/V as a Pillar source with values parsed as YAML
`csvpillar`	Store key/value pairs in a CSV file
`digicert`	Digicert Pillar Certificates
`django_orm`	Generate Pillar data from Django models through the Django ORM
`ec2_pillar`	Retrieve EC2 instance data for minions for ec2_tags and ec2_tags_list
`etcd_pillar`	Use etcd data as a Pillar source
`file_tree`	The `file_tree` external pillar allows values from all files in a directory tree to be imported as Pillar data.
`foreman`	A module to pull data from Foreman via its API into the Pillar dictionary
`git_pillar`	Use a git repository as a Pillar source
`gpg`	Decrypt pillar data through the builtin GPG renderer
`hg_pillar`	Use remote Mercurial repository as a Pillar source.
`hiera`	Use hiera data as a Pillar source
`http_json`	A module that adds data to the Pillar structure retrieved by an http request
`http_yaml`	A module that adds data to the Pillar structure retrieved by an http request
`libvirt`	Load up the libvirt keys into Pillar for a given minion if said keys have been generated using the libvirt key runner
`makostack`	Simple and flexible YAML ext_pillar which can read pillar from within pillar.
`mongo`	Read Pillar data from a mongodb collection
`mysql`	Retrieve Pillar data by doing a MySQL query
`netbox`	A module that adds data to the Pillar structure from a NetBox API.
`neutron`	Use Openstack Neutron data as a Pillar source.
`nodegroups`	Nodegroups Pillar
`pepa`	Pepa
`pillar_ldap`	Use LDAP data as a Pillar source
`postgres`	Retrieve Pillar data by doing a postgres query
`puppet`	Execute an unmodified puppet_node_classifier and read the output as YAML.
`reclass_adapter`	Use the "reclass" database as a Pillar source
`redismod`	Read pillar data from a Redis backend
`s3`	Copy pillar data from a bucket in Amazon S3
`saltclass`	SaltClass Pillar Module
`sql_base`	Retrieve Pillar data by doing a SQL query
`sqlcipher`	Retrieve Pillar data by running a SQLCipher query
`sqlite3`	Retrieve Pillar data by doing a SQLite3 query
`stack`	Simple and flexible YAML ext_pillar which can read pillar from within pillar.
`svn_pillar`	Clone a remote SVN repository and use the filesystem as a Pillar source
`varstack_pillar`	Use Varstack data as a Pillar source

Continued on next page

Table 12 – continued from previous page

vault	Vault Pillar Module
venafi	Venafi Pillar Certificates
virtkey	Accept a key from a hypervisor if the virt runner has already submitted an authorization request
vmware_pillar	Pillar data from vCenter or an ESXi host

21.12.1 salt.pillar.cmd_json

Execute a command and read the output as JSON. The JSON data is then directly overlaid onto the minion's Pillar data.

`salt.pillar.cmd_json.`**`ext_pillar`**(*minion_id*, *pillar*, *command*)
Execute a command and read the output as JSON

21.12.2 salt.pillar.cmd_yaml

Execute a command and read the output as YAML. The YAML data is then directly overlaid onto the minion's Pillar data

`salt.pillar.cmd_yaml.`**`ext_pillar`**(*minion_id*, *pillar*, *command*)
Execute a command and read the output as YAML

21.12.3 salt.pillar.cmd_yamlex

Execute a command and read the output as YAMLEX.

The YAMLEX data is then directly overlaid onto the minion's Pillar data

`salt.pillar.cmd_yamlex.`**`ext_pillar`**(*minion_id*, *pillar*, *command*)
Execute a command and read the output as YAMLEX

21.12.4 salt.pillar.cobbler

A module to pull data from Cobbler via its API into the Pillar dictionary

Configuring the Cobbler ext_pillar

The same cobbler.* parameters are used for both the Cobbler tops and Cobbler pillar modules.

```
ext_pillar:
- cobbler:
    key: cobbler # Nest results within this key. By default, values are not nested.
    only: [parameters] # Add only these keys to pillar.

cobbler.url: https://example.com/cobbler_api #default is http://localhost/cobbler_api
cobbler.user: username # default is no username
cobbler.password: password # default is no password
```

Module Documentation

`salt.pillar.cobbler.`**`ext_pillar`**`(`*minion_id, pillar, key=None, only=()*`)`

Read pillar data from Cobbler via its API.

21.12.5 salt.pillar.confidant

An external pillar module for getting credentials from confidant.

Configuring the Confidant module

The module can be configured via ext_pillar in the minion config:

```

```

ext_pillar:

- **confidant**:

  profile: # The URL of the confidant web service url: 'https://confidant-production.example.com'
  # The context to use for KMS authentication auth_context: from: example-production-iad to: confidant-production-iad user_type: service # The KMS master key to use for authentication auth_key: "alias/authnz" # Cache file for KMS auth token token_cache_file: /run/confidant/confidant_token # The duration of the validity of a token, in minutes token_duration: 60 # key, keyid and region can be defined in the profile, but it's # generally best to use IAM roles or environment variables for AWS # auth. keyid: 98nh9h9h908h09kjjk key: jhf908gyeghehe0he0g8h9u0j0n0n09hj09h0 region: us-east-1

depends confidant-common, confidant-client

Module Documentation

`salt.pillar.confidant.`**`ext_pillar`**`(`*minion_id, pillar, profile=None*`)`

Read pillar data from Confidant via its API.

21.12.6 salt.pillar.consul_pillar module

Use Consul K/V as a Pillar source with values parsed as YAML

depends

- python-consul

In order to use an consul server, a profile must be created in the master configuration file:

```
my_consul_config:
  consul.host: 127.0.0.1
  consul.port: 8500
  consul.token: b6376760-a8bb-edd5-fcda-33bc13bfc556
  consul.scheme: http
  consul.consistency: default
  consul.dc: dev
  consul.verify: True
```

All parameters are optional.

The `consul.token` requires python-consul >= 0.4.7.

If you have a multi-datacenter Consul cluster you can map your `pillarenv``s to your data centers by providing a dictionary of mappings in ``consul.dc` field:

```
my_consul_config:
  consul.dc:
    dev: us-east-1
    prod: us-west-1
```

In the example above we specifying static mapping between Pillar environments and data centers: the data for `dev` and `prod` Pillar environments will be fetched from `us-east-1` and `us-west-1` datacenter respectively.

In fact when `consul.dc` is set to dictionary keys are processed as regular expressions (that can capture named parameters) and values are processed as string templates as per PEP 3101.

```
my_consul_config:
  consul.dc:
    ^dev-.*$: dev-datacenter
    ^(?P<region>.*)-prod$: prod-datacenter-{region}
```

This example maps all Pillar environments starting with `dev-` to `dev-datacenter` whereas Pillar environment like `eu-prod` will be mapped to `prod-datacenter-eu`.

Before evaluation patterns are sorted by length in descending order.

If Pillar environment names correspond to data center names a single pattern can be used:

```
my_consul_config:
  consul.dc:
    ^(?P<env>.*)$: '{env}'
```

After the profile is created, configure the external pillar system to use it. Optionally, a root may be specified.

```
ext_pillar:
  - consul: my_consul_config

ext_pillar:
  - consul: my_consul_config root=salt
```

Using these configuration profiles, multiple consul sources may also be used:

```
ext_pillar:
  - consul: my_consul_config
  - consul: my_other_consul_config
```

Either the `minion_id`, or the `role`, or the `environment` grain may be used in the `root` path to expose minion-specific information stored in consul.

```
ext_pillar:
  - consul: my_consul_config root=salt/%(minion_id)s
  - consul: my_consul_config root=salt/%(role)s
  - consul: my_consul_config root=salt/%(environment)s
```

Minion-specific values may override shared values when the minion-specific root appears after the shared root:

```
ext_pillar:
  - consul: my_consul_config root=salt-shared
  - consul: my_other_consul_config root=salt-private/%(minion_id)s
```

If using the `role` or `environment` grain in the consul key path, be sure to define it using */etc/salt/grains*, or similar:

```
role: my-minion-role
environment: dev
```

It's possible to lock down where the pillar values are shared through minion targeting. Note that double quotes `"` are required around the target value and cannot be used inside the matching statement. See the section on Compound Matchers for more examples.

```
ext_pillar:
  - consul: my_consul_config root=salt target="L@salt.example.com and G@osarch:x86_64"
```

The data from Consul can be merged into a nested key in Pillar.

```
ext_pillar:
  - consul: my_consul_config pillar_root=consul_data
```

By default, keys containing YAML data will be deserialized before being merged into Pillar. This behavior can be disabled by setting `expand_keys` to `false`.

```
ext_pillar:
  - consul: my_consul_config expand_keys=false
```

`salt.pillar.consul_pillar.`**`consul_fetch`**(*client*, *path*)
Query consul for all keys/values within base path

`salt.pillar.consul_pillar.`**`ext_pillar`**(*minion_id*, *pillar*, *conf*)
Check consul for all data

`salt.pillar.consul_pillar.`**`fetch_tree`**(*client*, *path*, *expand_keys*)
Grab data from consul, trim base path and remove any keys which are folders. Take the remaining data and send it to be formatted in such a way as to be used as pillar data.

`salt.pillar.consul_pillar.`**`get_conn`**(*opts*, *profile*)
Return a client object for accessing consul

`salt.pillar.consul_pillar.`**`pillar_format`**(*ret*, *keys*, *value*, *expand_keys*)
Perform data formatting to be used as pillar data and merge it with the current pillar data

21.12.7 salt.pillar.csvpillar module

Store key/value pairs in a CSV file

New in version 2016.11.0.

Example configuration:

```
ext_pillar:
  - csv: /path/to/file.csv

# or

ext_pillar:
```

(continues on next page)

(continued from previous page)

```
  - csv:
      path: /path/to/file.csv
      namespace: 'subkey'
      fieldnames:
      - col1
      - col2
      - col2
```

The first column must be minion IDs and the first row must be dictionary keys. E.g.:

id	role	env
jerry	web	prod
stuart	web	stage
dave	web	qa
phil	db	prod
kevin	db	stage
mike	db	qa

Will produce the following Pillar values for a minion named "jerry":

```
{
    'role': 'web',
    'env': 'prod',
}
```

`salt.pillar.csvpillar.`**`ext_pillar`**(*mid*, *pillar*, *path*, *idkey='id'*, *namespace=None*, *fieldnames=None*, *restkey=None*, *restval=None*, *dialect='excel'*)

Read a CSV into Pillar

Parameters

- **path** (*str*) – Absolute path to a CSV file.
- **idkey** (*str*) – (Optional) The column name of minion IDs.
- **namespace** (*str*) – (Optional) A pillar key to namespace the values under.
- **fieldnames** (*list*) – (Optional) if the first row of the CSV is not column names they may be specified here instead.

21.12.8 salt.pillar.digicert module

Digicert Pillar Certificates

This module will only return pillar data if the `digicert` runner module has already been used to create certificates.

To configure this module, set `digicert` to `True` in the `ext_pillar` section of your `master` configuration file:

```
ext_pillar:
  - digicert: True
```

`salt.pillar.digicert.`**`ext_pillar`**(*minion_id*, *pillar*, *conf*)

Return an existing set of certificates

21.12.9 salt.pillar.django_orm

Generate Pillar data from Django models through the Django ORM

maintainer Micah Hausler <micah.hausler@gmail.com>

maturity new

Configuring the django_orm ext_pillar

To use this module, your Django project must be on the salt master server with database access. This assumes you are using virtualenv with all the project's requirements installed.

```
ext_pillar:
  - django_orm:
      pillar_name: my_application
      project_path: /path/to/project/
      settings_module: my_application.settings
      env_file: /path/to/env/file.sh
      # Optional: If your project is not using the system python,
      # add your virtualenv path below.
      env: /path/to/virtualenv/

      django_app:

        # Required: the app that is included in INSTALLED_APPS
        my_application.clients:

          # Required: the model name
          Client:

            # Required: model field to use as the key in the rendered
            # Pillar. Must be unique; must also be included in the
            # ``fields`` list below.
            name: shortname

            # Optional:
            # See Django's QuerySet documentation for how to use .filter()
            filter:  {'kw': 'args'}

            # Required: a list of field names
            # List items will be used as arguments to the .values() method.
            # See Django's QuerySet documentation for how to use .values()
            fields:
              - field_1
              - field_2
```

This would return pillar data that would look like

```
my_application:
  my_application.clients:
    Client:
      client_1:
        field_1: data_from_field_1
        field_2: data_from_field_2
      client_2:
        field_1: data_from_field_1
        field_2: data_from_field_2
```

As another example, data from multiple database tables can be fetched using Django's regular lookup syntax. Note, using ManyToManyFields will not currently work since the return from values() changes if a ManyToMany is present.

```
ext_pillar:
  - django_orm:
      pillar_name: djangotutorial
      project_path: /path/to/mysite
      settings_module: mysite.settings

      django_app:
        mysite.polls:
          Choices:
            name: poll__question
            fields:
              - poll__question
              - poll__id
              - choice_text
              - votes
```

Module Documentation

`salt.pillar.django_orm.`**`ext_pillar`**(*minion_id, pillar, pillar_name, project_path, settings_module, django_app, env=None, env_file=None, *args, **kwargs*)

Connect to a Django database through the ORM and retrieve model fields

Parameters

- **`pillar_name`** (*`str`*) – The name of the pillar to be returned
- **`project_path`** (*`str`*) – The full path to your Django project (the directory manage.py is in)
- **`settings_module`** (*`str`*) – The settings module for your project. This can be found in your manage.py file
- **`django_app`** (*`str`*) – A dictionary containing your apps, models, and fields
- **`env`** (*`str`*) – The full path to the virtualenv for your Django project
- **`env_file`** (*`str`*) – An optional bash file that sets up your environment. The file is run in a subprocess and the changed variables are then added

21.12.10 salt.pillar.ec2_pillar

Retrieve EC2 instance data for minions for ec2_tags and ec2_tags_list

The minion id must be the AWS instance-id or value in `tag_match_key`. For example set `tag_match_key` to `Name` to have the minion-id matched against the tag 'Name'. The tag contents must be unique. The value of `tag_match_value` can be 'uqdn' or 'asis'. if 'uqdn', then the domain will be stripped before comparison.

Additionally, the `use_grain` option can be set to `True`. This allows the use of an instance-id grain instead of the minion-id. Since this is a potential security risk, the configuration can be further expanded to include a list of minions that are trusted to only allow the alternate id of the instances to specific hosts. There is no glob matching at this time.

Note: If you are using `use_grain: True` in the configuration for this external pillar module, the minion must have *`metadata_server_grains`* enabled in the minion config file (see also *`here`*).

It is important to also note that enabling the `use_grain` option allows the minion to manipulate the pillar data returned, as described above.

The optional `tag_list_key` indicates which keys should be added to `ec2_tags_list` and be split by `tag_list_sep` (by default `;`). If a tag key is included in `tag_list_key` it is removed from ec2_tags. If a tag does not exist it is still included as an empty list.

..**note**:: As with any master configuration change, restart the salt-master daemon for changes to take effect.

```
ext_pillar:
  - ec2_pillar:
      tag_match_key: 'Name'
      tag_match_value: 'asis'
      tag_list_key:
        - Role
      tag_list_sep: ';'
      use_grain: True
      minion_ids:
        - trusted-minion-1
        - trusted-minion-2
        - trusted-minion-3
```

This is a very simple pillar configuration that simply retrieves the instance data from AWS. Currently the only portion implemented are EC2 tags, which returns a list of key/value pairs for all of the EC2 tags assigned to the instance.

`salt.pillar.ec2_pillar.`**`ext_pillar`**`(`*minion_id, pillar, use_grain=False, minion_ids=None, tag_match_key=None, tag_match_value='asis', tag_list_key=None, tag_list_sep=';'*`)`

Execute a command and read the output as YAML

21.12.11 salt.pillar.etcd_pillar

Use etcd data as a Pillar source

New in version 2014.7.0.

depends

- python-etcd

In order to use an etcd server, a profile must be created in the master configuration file:

```
my_etcd_config:
  etcd.host: 127.0.0.1
  etcd.port: 4001
```

After the profile is created, configure the external pillar system to use it. Optionally, a root may be specified.

```
ext_pillar:
  - etcd: my_etcd_config

ext_pillar:
  - etcd: my_etcd_config root=/salt
```

Using these configuration profiles, multiple etcd sources may also be used:

```
ext_pillar:
  - etcd: my_etcd_config
  - etcd: my_other_etcd_config
```

The `minion_id` may be used in the `root` path to expose minion-specific information stored in etcd.

```
ext_pillar:
  - etcd: my_etcd_config root=/salt/%(minion_id)s
```

Minion-specific values may override shared values when the minion-specific root appears after the shared root:

```
ext_pillar:
  - etcd: my_etcd_config root=/salt-shared
  - etcd: my_other_etcd_config root=/salt-private/%(minion_id)s
```

Using the configuration above, the following commands could be used to share a key with all minions but override its value for a specific minion:

```
etcdctl set /salt-shared/mykey my_value
etcdctl set /salt-private/special_minion_id/mykey my_other_value
```

`salt.pillar.etcd_pillar.`**`ext_pillar`**(*minion_id*, *pillar*, *conf*)
Check etcd for all data

21.12.12 salt.pillar.file_tree

The `file_tree` external pillar allows values from all files in a directory tree to be imported as Pillar data.

Note: This is an external pillar and is subject to the *rules and constraints* governing external pillars.

New in version 2015.5.0.

In this pillar, data is organized by either Minion ID or Nodegroup name. To setup pillar data for a specific Minion, place it in `<root_dir>/hosts/<minion_id>`. To setup pillar data for an entire Nodegroup, place it in `<root_dir>/nodegroups/<node_group>` where `<node_group>` is the Nodegroup's name.

Example `file_tree` Pillar

Master Configuration

```
ext_pillar:
  - file_tree:
      root_dir: /srv/ext_pillar
      follow_dir_links: False
      keep_newline: True
```

The `root_dir` parameter is required and points to the directory where files for each host are stored. The `follow_dir_links` parameter is optional and defaults to False. If `follow_dir_links` is set to True, this external pillar will follow symbolic links to other directories.

Warning: Be careful when using `follow_dir_links`, as a recursive symlink chain will result in unexpected results.

Changed in version 2018.3.0: If `root_dir` is a relative path, it will be treated as relative to the *`pillar_roots`* of the environment specified by *`pillarenv`*. If an environment specifies multiple roots, this module will search for files relative to all of them, in order, merging the results.

If `keep_newline` is set to `True`, then the pillar values for files ending in newlines will keep that newline. The default behavior is to remove the end-of-file newline. `keep_newline` should be turned on if the pillar data is intended to be used to deploy a file using `contents_pillar` with a *`file.managed`* state.

Changed in version 2015.8.4: The `raw_data` parameter has been renamed to `keep_newline`. In earlier releases, `raw_data` must be used. Also, this parameter can now be a list of globs, allowing for more granular control over which pillar values keep their end-of-file newline. The globs match paths relative to the directories named for minion IDs and nodegroups underneath the `root_dir` (see the layout examples in the below sections).

```
ext_pillar:
  - file_tree:
      root_dir: /path/to/root/directory
      keep_newline:
        - files/testdir/*
```

Note: In earlier releases, this documentation incorrectly stated that binary files would not affected by the `keep_newline` configuration. However, this module does not actually distinguish between binary and text files.

Changed in version 2017.7.0: Templating/rendering has been added. You can now specify a default render pipeline and a black- and whitelist of (dis)allowed renderers.

`template` must be set to `True` for templating to happen.

```
ext_pillar:
  - file_tree:
    root_dir: /path/to/root/directory
    render_default: jinja|yaml
    renderer_blacklist:
      - gpg
    renderer_whitelist:
      - jinja
      - yaml
    template: True
```

Assigning Pillar Data to Individual Hosts

To configure pillar data for each host, this external pillar will recursively iterate over `root_dir`/hosts/`id` (where `id` is a minion ID), and compile pillar data with each subdirectory as a dictionary key and each file as a value.

For example, the following `root_dir` tree:

```
./hosts/
./hosts/test-host/
./hosts/test-host/files/
./hosts/test-host/files/testdir/
./hosts/test-host/files/testdir/file1.txt
./hosts/test-host/files/testdir/file2.txt
./hosts/test-host/files/another-testdir/
./hosts/test-host/files/another-testdir/symlink-to-file1.txt
```

will result in the following pillar tree for minion with ID `test-host`:

```
test-host:
    ----------
    apache:
```

(continues on next page)

(continued from previous page)

```
        ----------
        config.d:
            ----------
            00_important.conf:
                <important_config important_setting="yes" />
            20_bob_extra.conf:
                <bob_specific_cfg has_freeze_ray="yes" />
    corporate_app:
        ----------
        settings:
            ----------
            common_settings:
                // This is the main settings file for the corporate
                // internal web app
                main_setting: probably
            bob_settings:
                role: bob
```

Note: The leaf data in the example shown is the contents of the pillar files.

`salt.pillar.file_tree.`**`ext_pillar`**(*minion_id, pillar, root_dir=None, follow_dir_links=False, debug=False, keep_newline=False, render_default=None, renderer_blacklist=None, renderer_whitelist=None, template=False*)

Compile pillar data from the given `root_dir` specific to Nodegroup names and Minion IDs.

If a Minion's ID is not found at `<root_dir>/host/<minion_id>` or if it is not included in any Nodegroups named at `<root_dir>/nodegroups/<node_group>`, no pillar data provided by this pillar module will be available for that Minion.

Changed in version 2017.7.0: Templating/rendering has been added. You can now specify a default render pipeline and a black- and whitelist of (dis)allowed renderers.

`template` must be set to `True` for templating to happen.

```
ext_pillar:
  - file_tree:
    root_dir: /path/to/root/directory
    render_default: jinja|yaml
    renderer_blacklist:
      - gpg
    renderer_whitelist:
      - jinja
      - yaml
    template: True
```

Parameters

- **`minion_id`** – The ID of the Minion whose pillar data is to be collected
- **`pillar`** – Unused by the `file_tree` pillar module
- **`root_dir`** – Filesystem directory used as the root for pillar data (e.g. `/srv/ext_pillar`)

  Changed in version 2018.3.0: If `root_dir` is a relative path, it will be treated as relative to the *`pillar_roots`* of the environment specified by *`pillarenv`*. If an environment specifies multiple roots, this module will search for files relative to all of them, in order, merging the results.

- **follow_dir_links** – Follow symbolic links to directories while collecting pillar files. Defaults to `False`.

  > **Warning:** Care should be exercised when enabling this option as it will follow links that point outside of `root_dir`.

  > **Warning:** Symbolic links that lead to infinite recursion are not filtered.

- **debug** – Enable debug information at log level `debug`. Defaults to `False`. This option may be useful to help debug errors when setting up the `file_tree` pillar module.
- **keep_newline** – Preserve the end-of-file newline in files. Defaults to `False`. This option may either be a boolean or a list of file globs (as defined by the Python fnmatch package) for which end-of-file newlines are to be kept.

  `keep_newline` should be turned on if the pillar data is intended to be used to deploy a file using `contents_pillar` with a *`file.managed`* state.

  Changed in version 2015.8.4: The `raw_data` parameter has been renamed to `keep_newline`. In earlier releases, `raw_data` must be used. Also, this parameter can now be a list of globs, allowing for more granular control over which pillar values keep their end-of-file newline. The globs match paths relative to the directories named for Minion IDs and Nodegroup namess underneath the `root_dir`.

  ```
  ext_pillar:
    - file_tree:
        root_dir: /srv/ext_pillar
        keep_newline:
          - apache/config.d/*
          - corporate_app/settings/*
  ```

  > **Note:** In earlier releases, this documentation incorrectly stated that binary files would not affected by the `keep_newline`. However, this module does not actually distinguish between binary and text files.

- **render_default** – Override Salt's *`default global renderer`* for the `file_tree` pillar.

  ```
  render_default: jinja
  ```

- **renderer_blacklist** – Disallow renderers for pillar files.

  ```
  renderer_blacklist:
    - json
  ```

- **renderer_whitelist** – Allow renderers for pillar files.

  ```
  renderer_whitelist:
    - yaml
    - jinja
  ```

- **template** – Enable templating of pillar files. Defaults to `False`.

21.12.13 salt.pillar.foreman

A module to pull data from Foreman via its API into the Pillar dictionary

Configuring the Foreman ext_pillar

Set the following Salt config to setup Foreman as external pillar source:

```
ext_pillar:
  - foreman:
      key: foreman # Nest results within this key
      only: ['hostgroup_name', 'parameters'] # Add only these keys to pillar

foreman.url: https://example.com/foreman_api
foreman.user: username # default is admin
foreman.password: password # default is changeme
```

The following options are optional:

```
foreman.api: apiversion # default is 2 (1 is not supported yet)
foreman.verifyssl: False # default is True
foreman.certfile: /etc/ssl/certs/mycert.pem # default is None
foreman.keyfile: /etc/ssl/private/mykey.pem # default is None
foreman.cafile: /etc/ssl/certs/mycert.ca.pem # default is None
foreman.lookup_parameters: True # default is True
```

An alternative would be to use the Foreman modules integrating Salt features in the Smart Proxy and the webinterface.

Further information can be found on GitHub.

Module Documentation

`salt.pillar.foreman.`**`ext_pillar`**`(`*minion_id, pillar, key=None, only=()*`)`

Read pillar data from Foreman via its API.

21.12.14 salt.pillar.git_pillar

Use a git repository as a Pillar source

This external pillar allows for a Pillar top file and Pillar SLS files to be sourced from a git repository.

However, since git_pillar does not have an equivalent to the *`pillar_roots`* parameter, configuration is slightly different. A Pillar top file is required to be in the git repository and must still contain the relevant environment, like so:

```
base:
  '*':
    - foo
```

The branch/tag which maps to that environment must then be specified along with the repo's URL. Configuration details can be found below.

Important: Each branch/tag used for git_pillar must have its own top file. This is different from how the top file works when configuring *States*. The reason for this is that each git_pillar branch/tag is processed separately from the rest. Therefore, if the `qa` branch is to be used for git_pillar, it would need to have its own top file, with the `qa` environment defined within it, like this:

```
qa:
  'dev-*':
    - bar
```

Additionally, while git_pillar allows for the branch/tag to be overridden (see *here*), keep in mind that the top file must reference the actual environment name. It is common practice to make the environment in a git_pillar top file match the branch/tag name, but when remapping, the environment of course no longer matches the branch/tag, and the top file needs to be adjusted accordingly. When expected Pillar values configured in git_pillar are missing, this is a common misconfiguration that may be to blame, and is a good first step in troubleshooting.

Configuring git_pillar for Salt

Beginning with Salt version 2015.8.0, pygit2 is now supported in addition to GitPython. The requirements for GitPython and pygit2 are the same as for GitFS, as described *here*.

Important: git_pillar has its own set of global configuration parameters. While it may seem intuitive to use the global gitfs configuration parameters (*`gitfs_base`*, etc.) to manage git_pillar, this will not work. The main difference for this is the fact that the different components which use Salt's git backend code do not all function identically. For instance, in git_pillar it is necessary to specify which branch/tag to be used for git_pillar remotes. This is the reverse behavior from gitfs, where branches/tags make up your environments.

See *here* for documentation on the git_pillar configuration options and their usage.

Here is an example git_pillar configuration:

```
ext_pillar:
  - git:
    # Use 'prod' instead of the branch name 'production' as the environment
    - production https://gitserver/git-pillar.git:
      - env: prod
    # Use 'dev' instead of the branch name 'develop' as the environment
    - develop https://gitserver/git-pillar.git:
      - env: dev
    # No per-remote config parameters (and no trailing colon), 'qa' will
    # be used as the environment
    - qa https://gitserver/git-pillar.git
    # SSH key authentication
    - master git@other-git-server:pillardata-ssh.git:
      # Pillar SLS files will be read from the 'pillar' subdirectory in
      # this repository
      - root: pillar
      - privkey: /path/to/key
      - pubkey: /path/to/key.pub
      - passphrase: CorrectHorseBatteryStaple
    # HTTPS authentication
    - master https://other-git-server/pillardata-https.git:
      - user: git
      - password: CorrectHorseBatteryStaple
```

The main difference between this and the old way of configuring git_pillar is that multiple remotes can be configured under one `git` section under *`ext_pillar`*. More than one `git` section can be used, but it is not necessary. Remotes will be evaluated sequentially.

Per-remote configuration parameters are supported (similar to *gitfs*), and global versions of the git_pillar configuration parameters can also be set.

To remap a specific branch to a specific Pillar environment, use the `env` per-remote parameter:

```
ext_pillar:
  - git:
    - production https://gitserver/git-pillar.git:
      - env: prod
```

If `__env__` is specified as the branch name, then git_pillar will decide which branch to use based on the following criteria:

- If the minion has a *`pillarenv`* configured, it will use that pillar environment. (2016.11.2 and later)
- Otherwise, if the minion has an `environment` configured, it will use that environment.
- Otherwise, the master's *`git_pillar_base`* will be used.

Note: The use of *`environment`* to choose the pillar environment dates from a time before the *`pillarenv`* parameter was added. In a future release, it will be ignored and either the minion's *`pillarenv`* or the master's *`git_pillar_base`* will be used.

Here's an example of using `__env__` as the git_pillar environment:

```
ext_pillar:
  - git:
    - __env__ https://gitserver/git-pillar.git:
      - root: pillar
```

The corresponding Pillar top file would look like this:

```
"{{saltenv}}":
  '*':
    - bar
```

With the addition of pygit2 support, git_pillar can now interact with authenticated remotes. Authentication works just like in gitfs (as outlined in the *Git Fileserver Backend Walkthrough*), only with the global authenication parameter names prefixed with `git_pillar` instead of `gitfs` (e.g. *`git_pillar_pubkey`*, *`git_pillar_privkey`*, *`git_pillar_passphrase`*, etc.).

Note: The `name` parameter can be used to further differentiate between two remotes with the same URL and branch. When using two remotes with the same URL, the `name` option is required.

How Multiple Remotes Are Handled

As noted above, multiple remotes can be included in the same `git` ext_pillar configuration. Consider the following:

```
my_etcd_config:
  etcd.host: 127.0.0.1
  etcd.port: 4001

ext_pillar:
  - etcd: my_etcd_config
  - git:
    - master https://mydomain.tld/foo.git:
      - root: pillar
    - master https://mydomain.tld/bar.git
    - master https://mydomain.tld/baz.git
    - dev https://mydomain.tld/qux.git
  - git:
    - master https://mydomain.tld/abc.git
    - dev https://mydomain.tld/123.git
```

To understand how pillar data from these repos will be compiled, it's important to know how Salt will process them. The following points should be kept in mind:

1. Each ext_pillar is called separately from the others. So, in the above example, the `etcd` ext_pillar will be evaluated first, with the first group of git_pillar remotes evaluated next (and merged into the etcd pillar data). Lastly, the second group of git_pillar remotes will be evaluated, and then merged into the ext_pillar data evaluated before it.
2. Within a single group of git_pillar remotes, each remote will be evaluated in order, with results merged together as each remote is evaluated.

   Note: Prior to the 2017.7.0 release, remotes would be evaluated in a non-deterministic order.

3. By default, when a repo is evaluated, other remotes' which share its pillar environment will have their files made available to the remote being processed.

The first point should be straightforward enough, but the second and third could use some additional clarification.

First, point #2. In the first group of git_pillar remotes, the top file and pillar SLS files in the `foo` remote will be evaluated first. The `bar` remote will be evaluated next, and its results will be merged into the pillar data compiled when the `foo` remote was evaluated. As the subsequent remotes are evaluated, their data will be merged in the same fashion.

But wait, don't these repositories belong to more than one pillar environments? Well, yes. The default method of generating pillar data compiles pillar data from all environments. This behavior can be overridden using a `pillarenv`. Setting a *`pillarenv`* in the minion config file will make that minion tell the master to ignore any pillar data from environments which don't match that pillarenv. A pillarenv can also be specified for a given minion or set of minions when *`running states`*, by using the `pillarenv` argument. The CLI pillarenv will override one set in the minion config file. So, assuming that a pillarenv of `base` was set for a minion, it would not get any of the pillar variables configured in the `qux` remote, since that remote is assigned to the `dev` environment. The only way to get its pillar data would be to specify a pillarenv of `dev`, which would mean that it would then ignore any items from the `base` pillarenv. A more detailed explanation of pillar environments can be found *here*.

Moving on to point #3, and looking at the example ext_pillar configuration, as the `foo` remote is evaluated, it will also have access to the files from the `bar` and `baz` remotes, since all three are assigned to the `base` pillar environment. So, if an SLS file referenced by the `foo` remotes's top file does not exist in the `foo` remote, it will be searched for in the `bar` remote, followed by the `baz` remote. When it comes time to evaluate the `bar` remote, SLS files referenced by the `bar` remote's top file will first be looked for in the `bar` remote, followed by `foo`, and `baz`, and when the `baz` remote is processed, SLS files will be looked for in `baz`, followed by `foo` and `bar`. This "failover" logic is called a *directory overlay*, and it is also used by *`file_roots`* and :conf_minion'pillar_roots'. The ordering of which remote is checked for SLS files is determined by the order they are listed. First the remote being processed is checked, then the others that share the same environment are checked. However, before the 2017.7.0 release, since evaluation was unordered, the remote being processed would be checked, followed in no specific order by the other repos which share the same environment.

Beginning with the 2017.7.0 release, this behavior of git_pillar remotes having access to files in other repos which share the same environment can be disabled by setting *git_pillar_includes* to `False`. If this is done, then all git_pillar remotes will only have access to their own SLS files. Another way of ensuring that a git_pillar remote will not have access to SLS files from other git_pillar remotes which share the same pillar environment is to put them in a separate `git` section under `ext_pillar`. Look again at the example configuration above. In the second group of git_pillar remotes, the `abc` remote would not have access to the SLS files from the `foo`, `bar`, and `baz` remotes, and vice-versa.

Mountpoints

New in version 2017.7.0.

Assume the following pillar top file:

```
base:
  'web*':
    - common
    - web.server.nginx
    - web.server.appdata
```

Now, assume that you would like to configure the `web.server.nginx` and `web.server.appdata` SLS files in separate repos. This could be done using the following ext_pillar configuration (assuming that *git_pillar_includes* has not been set to `False`):

```
ext_pillar:
  - git:
    - master https://mydomain.tld/pillar-common.git
    - master https://mydomain.tld/pillar-nginx.git
    - master https://mydomain.tld/pillar-appdata.git
```

However, in order to get the files in the second and third git_pillar remotes to work, you would need to first create the directory structure underneath it (i.e. place them underneath `web/server/` in the repository). This also makes it tedious to reorganize the configuration, as changing `web.server.nginx` to `web.nginx` in the top file would require you to also move the SLS files in the `pillar-nginx` up a directory level.

For these reasons, much like gitfs, git_pillar now supports a "mountpoint" feature. Using the following ext_pillar configuration, the SLS files in the second and third git_pillar remotes can be placed in the root of the git repository:

```
ext_pillar:
  - git:
    - master https://mydomain.tld/pillar-common.git
    - master https://mydomain.tld/pillar-nginx.git:
      - mountpoint: web/server/
    - master https://mydomain.tld/pillar-appdata.git:
      - mountpoint: web/server/
```

Now, if the top file changed the SLS target from `web.server.nginx`, instead of reorganizing the git repository, you would just need to adjust the mountpoint to `web/` (and restart the `salt-master` daemon).

Note:

- Leading and trailing slashes on the mountpoints are optional.

- Use of the `mountpoint` feature requires that `git_pillar_includes` is not disabled.
- Content from mounted git_pillar repos can only be referenced by a top file in the same pillar environment.
- Salt versions prior to 2018.3.4 ignore the `root` parameter when `mountpoint` is set.

all_saltenvs

New in version 2018.3.4.

When `__env__` is specified as the branch name, `all_saltenvs` per-remote configuration parameter overrides the logic Salt uses to map branches/tags to pillar environments (i.e. pillarenvs). This allows a single branch/tag to appear in all saltenvs. Example:

```
ext_pillar:
  - git:
    - __env__ https://mydomain.tld/top.git
      - all_saltenvs: master
    - __env__ https://mydomain.tld/pillar-nginx.git:
      - mountpoint: web/server/
    - __env__ https://mydomain.tld/pillar-appdata.git:
      - mountpoint: web/server/
```

git_pillar_update_interval

New in version 3000.

This option defines the default update interval (in seconds) for git_pillar remotes. The update is handled within the global loop, hence `git_pillar_update_interval` should be a multiple of `loop_interval`.

```
git_pillar_update_interval: 120
```

`salt.pillar.git_pillar.`**`ext_pillar`**(*minion_id, pillar, *repos*)
Checkout the ext_pillar sources and compile the resulting pillar SLS

21.12.15 salt.pillar.gpg module

Decrypt pillar data through the builtin GPG renderer

In most cases, you'll want to make this the last external pillar used. For example, to pair with the builtin stack pillar you could do something like this:

```
ext_pillar:
  - stack: /path/to/stack.cfg
  - gpg: {}
```

Set `gpg_keydir` in your config to adjust the homedir the renderer uses.

`salt.pillar.gpg.`**`ext_pillar`**(*minion_id, pillar, *args, **kwargs*)

21.12.16 salt.pillar.hg_pillar

Use remote Mercurial repository as a Pillar source.

New in version 2015.8.0.

The module depends on the `hglib` python module being available. This is the same requirement as for hgfs_ so should not pose any extra hurdles.

This external Pillar source can be configured in the master config file as such:

```
ext_pillar:
  - hg: ssh://hg@example.co/user/repo
```

class `salt.pillar.hg_pillar.`**`Repo`**(*repo_uri*)
Deal with remote hg (mercurial) repository for Pillar

`close`()
Cleanup mercurial command server

`pull`()

`update`(*branch='default'*)
Ensure we are using the latest revision in the hg repository

`salt.pillar.hg_pillar.`**`ext_pillar`**(*minion_id, pillar, repo, branch='default', root=None*)
Extract pillar from an hg repository

`salt.pillar.hg_pillar.`**`update`**(*repo_uri*)
Execute an hg pull on all the repos

21.12.17 salt.pillar.hiera

Use hiera data as a Pillar source

`salt.pillar.hiera.`**`ext_pillar`**(*minion_id, pillar, conf*)
Execute hiera and return the data

21.12.18 salt.pillar.http_json module

A module that adds data to the Pillar structure retrieved by an http request

Configuring the HTTP_JSON ext_pillar

Set the following Salt config to setup http json result as external pillar source:

```
ext_pillar:
  - http_json:
      url: http://example.com/api/minion_id
      ::TODO::
      username: username
      password: password
```

If the with_grains parameter is set, grain keys wrapped in can be provided (wrapped in <> brackets) in the url in order to populate pillar data based on the grain value.

```
ext_pillar:
  - http_json:
      url: http://example.com/api/<nodename>
      with_grains: True
```

Changed in version 2018.3.0: If %s is present in the url, it will be automatically replaced by the minion_id:

```
ext_pillar:
  - http_json:
      url: http://example.com/api/%s
```

Module Documentation

salt.pillar.http_json.**ext_pillar**(*minion_id*, *pillar*, *url*, *with_grains=False*)

Read pillar data from HTTP response.

Parameters

- **url** (*str*) – Url to request.
- **with_grains** (*bool*) – Whether to substitute strings in the url with their grain values.

Returns A dictionary of the pillar data to add.

Return type dict

21.12.19 salt.pillar.http_yaml module

A module that adds data to the Pillar structure retrieved by an http request

Configuring the HTTP_YAML ext_pillar

Set the following Salt config to setup an http endpoint as the external pillar source:

```
ext_pillar:
  - http_yaml:
      url: http://example.com/api/minion_id
      ::TODO::
      username: username
      password: password
```

If the with_grains parameter is set, grain keys wrapped in can be provided (wrapped in <> brackets) in the url in order to populate pillar data based on the grain value.

```
ext_pillar:
  - http_yaml:
      url: http://example.com/api/<nodename>
      with_grains: True
```

Changed in version 2018.3.0: If %s is present in the url, it will be automatically replaced by the minion_id:

```
ext_pillar:
  - http_json:
      url: http://example.com/api/%s
```

Module Documentation

`salt.pillar.http_yaml.`**`ext_pillar`**`(`*minion_id, pillar, url, with_grains=False*`)`

Read pillar data from HTTP response.

Parameters

- **`url`** (*`str`*) – Url to request.
- **`with_grains`** (*`bool`*) – Whether to substitute strings in the url with their grain values.

Returns A dictionary of the pillar data to add.

Return type dict

21.12.20 salt.pillar.libvirt

Load up the libvirt keys into Pillar for a given minion if said keys have been generated using the libvirt key runner

depends certtool

`salt.pillar.libvirt.`**`ext_pillar`**`(`*minion_id, pillar, command*`)`

Read in the generated libvirt keys

`salt.pillar.libvirt.`**`gen_hyper_keys`**`(`*minion_id, country='US', state='Utah', locality='Salt Lake City', organization='Salted', expiration_days='365'*`)`

Generate the keys to be used by libvirt hypervisors, this routine gens the keys and applies them to the pillar for the hypervisor minions

21.12.21 salt.pillar.makostack module

Simple and flexible YAML ext_pillar which can read pillar from within pillar.

New in version 2016.3.0.

This custom saltstack `ext_pillar` is a direct ripoff of the 'stack' ext_pillar, simply ported to use mako instead of jinja2 for templating.

It supports the following features:

- multiple config files that are mako templates with support for `pillar`, `__grains__`, `__salt__`, `__opts__` objects.
- a config file renders as an ordered list of files. Unless absolute, the paths of these files are relative to the current config file - if absolute, they will be treated literally.
- this list of files are read in order as mako templates with support for `stack`, `pillar`, `__grains__`, `__salt__`, `__opts__` objects.
- all these rendered files are then parsed as `yaml`.
- then all yaml dicts are merged in order, with support for the following. merging strategies: `merge-first`, `merge-last`, `remove`, and `overwrite`.
- stack config files can be matched based on `pillar`, `grains`, or `opts` values, which make it possible to support kind of self-contained environments.

Configuration in Salt

Like any other external pillar, its configuration takes place through the `ext_pillar` key in the master config file.

However, you can configure MakoStack in 3 different ways:

Single config file

This is the simplest option, you just need to set the path to your single MakoStack config file like below:

```
ext_pillar:
  - makostack: /path/to/stack.cfg
```

List of config files

You can also provide a list of config files:

```
ext_pillar:
  - makostack:
      - /path/to/stack1.cfg
      - /path/to/stack2.cfg
```

Select config files through grains|pillar|opts matching

You can also opt for a much more flexible configuration: MakoStack allows one to select the config files for the current minion based on matching values from either grains, or pillar, or opts objects.

Here is an example of such a configuration, which should speak by itself:

```
ext_pillar:
  - makostack:
      pillar:environment:
        dev: /path/to/dev/stack.cfg
        prod: /path/to/prod/stack.cfg
      grains:custom:grain:
        value:
          - /path/to/stack1.cfg
          - /path/to/stack2.cfg
      opts:custom:opt:
        value: /path/to/stack0.cfg
```

Grafting data from files to arbitrary namespaces

An extended syntax for config files permits defining ”graft points” on a per-config-file basis. As an example, if the file foo.cfg would produce the following:

```
foo:
  - bar
  - baz
```

and you specified the cfg file as /path/to/foo.cfg:yummy:fur, the following would actually end up in pillar after all merging was complete:

```
yummy:
  fur:
    foo:
      - bar
      - baz
```

MakoStack configuration files

The config files that are referenced in the above `ext_pillar` configuration are mako templates which must render as a simple ordered list of `yaml` files that will then be merged to build pillar data.

Unless an absolute path name is specified, the path of these `yaml` files is assumed to be relative to the directory containing the MakoStack config file. If a path begins with '/', however, it will be treated literally and can be anywhere on the filesystem.

The following variables are available in mako templating of makostack configuration files:

- `pillar`: the pillar data (as passed by Salt to our `ext_pillar` function)
- `minion_id`: the minion id ;-)
- `__opts__`: a dictionary of mostly Salt configuration options
- `__grains__`: a dictionary of the grains of the minion making this pillar call
- `__salt__`: a dictionary of Salt module functions, useful so you don't have to duplicate functions that already exist (note: runs on the master)

So you can use all the power of mako to build your list of `yaml` files that will be merged in pillar data.

For example, you could have a MakoStack config file which looks like:

```
$ cat /path/to/stack/config.cfg
core.yml
osarchs/%{ __grains__['osarch'] }}.yml
oscodenames/%{ __grains__['oscodename'] }.yml
% for role in pillar.get('roles', []):
roles/%{ role }.yml
% endfor
minions/%{ minion_id }.yml
```

And the whole directory structure could look like:

```
$ tree /path/to/stack/
/path/to/stack/
├── config.cfg
├── core.yml
├── osarchs/
│   ├── amd64.yml
│   └── armhf.yml
├── oscodenames/
│   ├── wheezy.yml
│   └── jessie.yml
├── roles/
│   ├── web.yml
│   └── db.yml
└── minions/
    ├── test-1-dev.yml
    └── test-2-dev.yml
```

Overall process

In the above MakoStack configuration, given that test-1-dev minion is an amd64 platform running Debian Jessie, and which pillar `roles` is `["db"]`, the following `yaml` files would be merged in order:

- `core.yml`
- `osarchs/amd64.yml`
- `oscodenames/jessie.yml`
- `roles/db.yml`
- `minions/test-1-dev.yml`

Before merging, every files above will be preprocessed as mako templates. The following variables are available in mako templating of `yaml` files:

- `stack`: the MakoStack pillar data object that has currently been merged (data from previous `yaml` files in MakoStack configuration)
- `pillar`: the pillar data (as passed by Salt to our `ext_pillar` function)
- `minion_id`: the minion id ;-)
- `__opts__`: a dictionary of mostly Salt configuration options
- `__grains__`: a dictionary of the grains of the minion making this pillar call
- `__salt__`: a dictionary of Salt module functions, useful so you don't have to duplicate functions that already exist (note: runs on the master)

So you can use all the power of mako to build your pillar data, and even use other pillar values that has already been merged by MakoStack (from previous `yaml` files in MakoStack configuration) through the `stack` variable.

Once a `yaml` file has been preprocessed by mako, we obtain a Python dict - let's call it `yml_data` - then, MakoStack will merge this `yml_data` dict in the main `stack` dict (which contains already merged MakoStack pillar data). By default, MakoStack will deeply merge `yml_data` in `stack` (similarly to the `recurse` salt `pillar_source_merging_strategy`), but 3 merging strategies are currently available for you to choose (see next section).

Once every `yaml` files have been processed, the `stack` dict will contain your whole own pillar data, merged in order by MakoStack. So MakoStack `ext_pillar` returns the `stack` dict, the contents of which Salt takes care to merge in with all of the other pillars and finally return the whole pillar to the minion.

Merging strategies

The way the data from a new `yaml_data` dict is merged with the existing `stack` data can be controlled by specifying a merging strategy. Right now this strategy can either be `merge-last` (the default), `merge-first`, `remove`, or `overwrite`.

Note that scalar values like strings, integers, booleans, etc. are always evaluated using the `overwrite` strategy (other strategies don't make sense in that case).

The merging strategy can be set by including a dict in the form of:

```
__: <merging strategy>
```

as the first item of the dict or list. This allows fine grained control over the merging process.

`merge-last` (default) strategy

If the `merge-last` strategy is selected (the default), then content of dict or list variables is merged recursively with previous definitions of this variable (similarly to the `recurse` salt `pillar_source_merging_strategy`). This allows for extending previously defined data.

`merge-first` strategy

If the `merge-first` strategy is selected, then the content of dict or list variables are swapped between the `yaml_data` and `stack` objects before being merged recursively with the `merge-last` previous strategy.

`remove` strategy

If the `remove` strategy is selected, then content of dict or list variables in `stack` are removed only if the corresponding item is present in the `yaml_data` dict. This allows for removing items from previously defined data.

`overwrite` strategy

If the `overwrite` strategy is selected, then the content of dict or list variables in `stack` is overwritten by the content of `yaml_data` dict. So this allows one to overwrite variables from previous definitions.

Merging examples

Let's go through small examples that should clarify what's going on when a `yaml_data` dict is merged in the `stack` dict.

When you don't specify any strategy, the default `merge-last` strategy is selected:

`stack`

```yaml
users:
  tom:
    uid: 500
    roles:
      - sysadmin
  root:
    uid: 0
```

`yaml_data`

```yaml
users:
  tom:
    uid: 1000
    roles:
      - developer
  mat:
    uid: 1001
```

`stack` (after merge)

```yaml
users:
  tom:
    uid: 1000
    roles:
      - sysadmin
      - developer
  mat:
    uid: 1001
  root:
    uid: 0
```

Then you can select a custom merging strategy using the `__` key in a dict:

<table>
<tr><th><code>stack</code></th><th><code>yaml_data</code></th><th><code>stack</code> (after merge)</th></tr>
<tr><td><pre>
users:
  tom:
    uid: 500
    roles:
      - sysadmin
  root:
    uid: 0
</pre></td><td><pre>
users:
  __: merge-last
  tom:
    uid: 1000
    roles:
      - developer
  mat:
    uid: 1001
</pre></td><td><pre>
users:
  tom:
    uid: 1000
    roles:
      - sysadmin
      - developer
  mat:
    uid: 1001
  root:
    uid: 0
</pre></td></tr>
<tr><td><pre>
users:
  tom:
    uid: 500
    roles:
      - sysadmin
  root:
    uid: 0
</pre></td><td><pre>
users:
  __: merge-first
  tom:
    uid: 1000
    roles:
      - developer
  mat:
    uid: 1001
</pre></td><td><pre>
users:
  tom:
    uid: 500
    roles:
      - developer
      - sysadmin
  mat:
    uid: 1001
  root:
    uid: 0
</pre></td></tr>
<tr><td><pre>
users:
  tom:
    uid: 500
    roles:
      - sysadmin
  root:
    uid: 0
</pre></td><td><pre>
users:
  __: remove
  tom:
  mat:
</pre></td><td><pre>
users:
  root:
    uid: 0
</pre></td></tr>
<tr><td><pre>
users:
  tom:
    uid: 500
    roles:
      - sysadmin
  root:
    uid: 0
</pre></td><td><pre>
users:
  __: overwrite
  tom:
    uid: 1000
    roles:
      - developer
  mat:
    uid: 1001
</pre></td><td><pre>
users:
  tom:
    uid: 1000
    roles:
      - developer
  mat:
    uid: 1001
</pre></td></tr>
</table>

You can also select a custom merging strategy using a `__` object in a list:

`stack`	`yaml_data`	`stack` (after merge)

`users:`
`  - tom`
`  - root`
	`users:`
	`  - __: merge-last`
	`  - mat`
		`users:`
		`  - tom`
		`  - root`
		`  - mat`

`users:`
`  - tom`
`  - root`
	`users:`
	`  - __: merge-first`
	`  - mat`
		`users:`
		`  - mat`
		`  - tom`
		`  - root`

`users:`
`  - tom`
`  - root`
	`users:`
	`  - __: remove`
	`  - mat`
	`  - tom`
		`users:`
		`  - root`

`users:`
`  - tom`
`  - root`
	`users:`
	`  - __: overwrite`
	`  - mat`
		`users:`
		`  - mat`

`salt.pillar.makostack.`**`ext_pillar`**`(`*minion_id*, *pillar*, **args*, ***kwargs*`)`

21.12.22 salt.pillar.mongo

Read Pillar data from a mongodb collection

depends pymongo (for salt-master)

This module will load a node-specific pillar dictionary from a mongo collection. It uses the node's id for lookups and can load either the whole document, or just a specific field from that document as the pillar dictionary.

Salt Master Mongo Configuration

The module shares the same base mongo connection variables as *`salt.returners.mongo_return`*. These variables go in your master config file.

- `mongo.db` - The mongo database to connect to. Defaults to `'salt'`.
- `mongo.host` - The mongo host to connect to. Supports replica sets by specifying all hosts in the set, comma-delimited. Defaults to `'salt'`.
- `mongo.port` - The port that the mongo database is running on. Defaults to `27017`.
- `mongo.user` - The username for connecting to mongo. Only required if you are using mongo authentication. Defaults to `''`.
- `mongo.password` - The password for connecting to mongo. Only required if you are using mongo authentication. Defaults to `''`.

Configuring the Mongo ext_pillar

The Mongo ext_pillar takes advantage of the fact that the Salt Master configuration file is yaml. It uses a sub-dictionary of values to adjust specific features of the pillar. This is the explicit single-line dictionary notation for yaml. One may be able to get the easier-to-read multi-line dict to work correctly with some experimentation.

```
ext_pillar:
  - mongo: {collection: vm, id_field: name, re_pattern: \.example\.com, fields:
↪[customer_id, software, apache_vhosts]}
```

In the example above, we've decided to use the `vm` collection in the database to store the data. Minion ids are stored in the `name` field on documents in that collection. And, since minion ids are FQDNs in most cases, we'll need to trim the domain name in order to find the minion by hostname in the collection. When we find a minion, return only the `customer_id`, `software`, and `apache_vhosts` fields, as that will contain the data we want for a given node. They will be available directly inside the `pillar` dict in your SLS templates.

Module Documentation

`salt.pillar.mongo.`**`ext_pillar`**(*minion_id, pillar, collection='pillar', id_field='_id', re_pattern=None, re_replace='', fields=None*)

Connect to a mongo database and read per-node pillar information.

Parameters

- **`collection`** (*) – The mongodb collection to read data from. Defaults to `'pillar'`.
- **`id_field`** (*) – The field in the collection that represents an individual minion id. Defaults to `'_id'`.
- **`re_pattern`** (*) – If your naming convention in the collection is shorter than the minion id, you can use this to trim the name. *re_pattern* will be used to match the name, and *re_replace* will be used to replace it. Backrefs are supported as they are in the Python standard library. If `None`, no mangling of the name will be performed - the collection will be searched with the entire minion id. Defaults to `None`.
- **`re_replace`** (*) – Use as the replacement value in node ids matched with *re_pattern*. Defaults to ''. Feel free to use backreferences here.
- **`fields`** (*) – The specific fields in the document to use for the pillar data. If `None`, will use the entire document. If using the entire document, the `_id` field will be converted to string. Be careful with other fields in the document as they must be string serializable. Defaults to `None`.

21.12.23 salt.pillar.mysql

Retrieve Pillar data by doing a MySQL query

MariaDB provides Python support through the MySQL Python package. Therefore, you may use this module with both MySQL or MariaDB.

This module is a concrete implementation of the sql_base ext_pillar for MySQL.

maturity new

depends python-mysqldb

platform all

Configuring the mysql ext_pillar

Use the 'mysql' key under ext_pillar for configuration of queries.

MySQL configuration of the MySQL returner is being used (mysql.db, mysql.user, mysql.pass, mysql.port, mysql.host) for database connection info.

Required python modules: MySQLdb

Complete example

```
mysql:
  user: 'salt'
  pass: 'super_secret_password'
  db: 'salt_db'
  port: 3306
  ssl:
    cert: /etc/mysql/client-cert.pem
    key: /etc/mysql/client-key.pem

ext_pillar:
  - mysql:
      fromdb:
        query: 'SELECT col1,col2,col3,col4,col5,col6,col7
                  FROM some_random_table
                 WHERE minion_pattern LIKE %s'
        depth: 5
        as_list: True
        with_lists: [1,3]
```

class salt.pillar.mysql.**MySQLExtPillar**

This class receives and processes the database rows from MySQL.

extract_queries(*args, kwargs*)

This function normalizes the config block into a set of queries we can use. The return is a list of consistently laid out dicts.

salt.pillar.mysql.**ext_pillar**(*minion_id, pillar, *args, **kwargs*)

Execute queries against MySQL, merge and return as a dict

21.12.24 salt.pillar.netbox

A module that adds data to the Pillar structure from a NetBox API.

New in version 2019.2.0.

Configuring the NetBox ext_pillar

```
ext_pillar:
  - netbox:
      api_url: http://netbox_url.com/api/
      api_token: 123abc
```

Create a token in your NetBox instance at http://netbox_url.com/user/api-tokens/

The following options are optional, and determine whether or not the module will attempt to configure the `proxy` pillar data for use with the napalm proxy-minion:

```
proxy_return: True
proxy_username: admin
```

By default, this module will query the NetBox API for the platform associated with the device, and use the 'NAPALM driver' field to set the napalm proxy-minion driver. (Currently only 'napalm' is supported for drivertype.)

This module currently only supports the napalm proxy minion and assumes you will use SSH keys to authenticate to the network device. If password authentication is desired, it is recommended to create another `proxy` key in pillar_roots (or git_pillar) with just the `passwd` key and use `salt.renderers.gpg` to encrypt the value. If any additional options for the proxy setup are needed they should also be configured in pillar_roots.

Other available options:

site_details: **`True`** Whether should retrieve details of the site the device belongs to.

site_prefixes: **`True`** Whether should retrieve the prefixes of the site the device belongs to.

`salt.pillar.netbox.`**`ext_pillar`**`(`*minion_id*, *pillar*, **args*, ***kwargs*`)`
Query NetBox API for minion data

21.12.25 salt.pillar.neutron module

Use Openstack Neutron data as a Pillar source. Will list all networks listed inside of Neutron, to all minions.

New in version 2015.5.1.

depends

- python-neutronclient

A keystone profile must be used for the pillar to work (no generic keystone configuration here). For example:

```
my openstack_config:
  keystone.user: 'admin'
  keystone.password: 'password'
  keystone.tenant: 'admin'
  keystone.auth_url: 'http://127.0.0.1:5000/v2.0/'
  keystone.region_name: 'RegionOne'
  keystone.service_type: 'network'
```

After the profile is created, configure the external pillar system to use it.

```
ext_pillar:
  - neutron: my_openstack_config
```

Using these configuration profiles, multiple neutron sources may also be used:

```
ext_pillar:
  - neutron: my_openstack_config
  - neutron: my_other_openstack_config
```

By default, these networks will be returned as a pillar item called `networks`. In order to have them returned under a different name, add the name after the Keystone profile name:

ext_pillar:

- neutron: my_openstack_config neutron_networks

`salt.pillar.neutron.`**`ext_pillar`**(*minion_id*, *pillar*, *conf*)
Check neutron for all data

21.12.26 salt.pillar.nodegroups

Nodegroups Pillar

Introspection: to which nodegroups does my minion belong? Provides a pillar with the default name of *nodegroups* which contains a list of nodegroups which match for a given minion.

New in version 2016.11.0.

Command Line

```
salt-call pillar.get nodegroups
local:
    - class_infra
    - colo_sj
    - state_active
    - country_US
    - type_saltmaster
```

Configuring Nodegroups Pillar

```
extension_modules: /srv/salt/ext
ext_pillar:
  - nodegroups:
      pillar_name: 'nodegroups'
```

`salt.pillar.nodegroups.`**`ext_pillar`**(*minion_id*, *pillar*, *pillar_name=None*)
A salt external pillar which provides the list of nodegroups of which the minion is a member.

Parameters

- **`minion_id`** – used for compound matching nodegroups
- **`pillar`** – provided by salt, but not used by nodegroups ext_pillar
- **`pillar_name`** – optional name to use for the pillar, defaults to 'nodegroups'

Returns a dictionary which is included by the salt master in the pillars returned to the minion

21.12.27 salt.pillar.pepa

Pepa

Configuration templating for SaltStack using Hierarchical substitution and Jinja.

Configuring Pepa

```
extension_modules: /srv/salt/ext

ext_pillar:
  - pepa:
```

(continues on next page)

(continued from previous page)

```
      resource: host                # Name of resource directory and sub-key in
↪pillars
      sequence:                     # Sequence used for hierarchical substitution
        - hostname:                 # Name of key
            name: input             # Alias used for template directory
            base_only: True         # Only use templates from Base environment, i.e.
↪no staging
        - default:
        - environment:
        - location..region:
            name: region
        - location..country:
            name: country
        - location..datacenter:
            name: datacenter
        - roles:
        - osfinger:
            name: os
        - hostname:
            name: override
            base_only: True
      subkey: True                  # Create a sub-key in pillars, named after the
↪resource in this case [host]
      subkey_only: True             # Only create a sub-key, and leave the top level
↪untouched

pepa_roots:                         # Base directory for each environment
  base: /srv/pepa/base              # Path for base environment
  dev: /srv/pepa/base               # Associate dev with base
  qa: /srv/pepa/qa
  prod: /srv/pepa/prod

# Use a different delimiter for nested dictionaries, defaults to '..' since some keys
↪may use '.' in the name
#pepa_delimiter: ..

# Supply Grains for Pepa, this should **ONLY** be used for testing or validation
#pepa_grains:
#  environment: dev

# Supply Pillar for Pepa, this should **ONLY** be used for testing or validation
#pepa_pillars:
#  saltversion: 0.17.4

# Enable debug for Pepa, and keep Salt on warning
#log_level: debug

#log_granular_levels:
#  salt: warning
#  salt.loaded.ext.pillar.pepa: debug
```

Pepa can also be used in Master-less SaltStack setup.

Command line

```
usage: pepa.py [-h] [-c CONFIG] [-d] [-g GRAINS] [-p PILLAR] [-n] [-v]
               hostname

positional arguments:
  hostname              Hostname

optional arguments:
  -h, --help            show this help message and exit
  -c CONFIG, --config CONFIG
                        Configuration file
  -d, --debug           Print debug info
  -g GRAINS, --grains GRAINS
                        Input Grains as YAML
  -p PILLAR, --pillar PILLAR
                        Input Pillar as YAML
  -n, --no-color        No color output
  -v, --validate        Validate output
```

Templates

Templates is configuration for a host or software, that can use information from Grains or Pillars. These can then be used for hierarchically substitution.

Example File: host/input/test_example_com.yaml

```
location..region: emea
location..country: nl
location..datacenter: foobar
environment: dev
roles:
  - salt.master
network..gateway: 10.0.0.254
network..interfaces..eth0..hwaddr: 00:20:26:a1:12:12
network..interfaces..eth0..dhcp: False
network..interfaces..eth0..ipv4: 10.0.0.3
network..interfaces..eth0..netmask: 255.255.255.0
network..interfaces..eth0..fqdn: {{ hostname }}
cobbler..profile: fedora-19-x86_64
```

As you see in this example you can use Jinja directly inside the template.

Example File: host/region/amer.yaml

```
network..dns..servers:
  - 10.0.0.1
  - 10.0.0.2
time..ntp..servers:
  - ntp1.amer.example.com
  - ntp2.amer.example.com
  - ntp3.amer.example.com
time..timezone: America/Chihuahua
yum..mirror: yum.amer.example.com
```

Each template is named after the value of the key using lowercase and all extended characters are replaced with underscore.

Example:

osfinger: Fedora-19

Would become:

fedora_19.yaml

Nested dictionaries

In order to create nested dictionaries as output you can use double dot ".." as a delimiter. You can change this using "pepa_delimiter" we choose double dot since single dot is already used by key names in some modules, and using ":" requires quoting in the YAML.

Example:

```yaml
network..dns..servers:
  - 10.0.0.1
  - 10.0.0.2
network..dns..options:
  - timeout:2
  - attempts:1
  - ndots:1
network..dns..search:
  - example.com
```

Would become:

```yaml
network:
  dns:
    servers:
      - 10.0.0.1
      - 10.0.0.2
    options:
      - timeout:2
      - attempts:1
      - ndots:1
    search:
      - example.com
```

Operators

Operators can be used to merge/unset a list/hash or set the key as immutable, so it can't be changed.

Operator	Description
merge()	Merge list or hash
unset()	Unset key
immutable()	Set the key as immutable, so it can't be changed
imerge()	Set immutable and merge
iunset()	Set immutable and unset

Example:

```
network..dns..search..merge():
  - foobar.com
  - dummy.nl
owner..immutable(): Operations
host..printers..unset():
```

Validation

Since it's very hard to test Jinja as is, the best approach is to run all the permutations of input and validate the output, i.e. Unit Testing.

To facilitate this in Pepa we use YAML, Jinja and Cerberus <https://github.com/nicolaiarocci/cerberus>.

Schema

So this is a validation schema for network configuration, as you see it can be customized with Jinja just as Pepa templates.

This was designed to be run as a build job in Jenkins or similar tool. You can provide Grains/Pillar input using either the config file or command line arguments.

File Example: host/validation/network.yaml

```
network..dns..search:
  type: list
  allowed:
    - example.com

network..dns..options:
  type: list
  allowed: ['timeout:2', 'attempts:1', 'ndots:1']

network..dns..servers:
  type: list
  schema:
    regex: ^([0-9]{1,3}\.){3}[0-9]{1,3}$

network..gateway:
  type: string
  regex: ^([0-9]{1,3}\.){3}[0-9]{1,3}$

{% if network.interfaces is defined %}
{% for interface in network.interfaces %}

network..interfaces..{{ interface }}..dhcp:
  type: boolean

network..interfaces..{{ interface }}..fqdn:
  type: string
  regex: ^([a-z0-9]([a-z0-9-]{0,61}[a-z0-9])?\.)+[a-zA-Z]{2,6}$

network..interfaces..{{ interface }}..hwaddr:
  type: string
  regex: ^([0-9a-f]{1,2}\:){5}[0-9a-f]{1,2}$
```

(continues on next page)

(continued from previous page)

```
network..interfaces..{{ interface }}..ipv4:
  type: string
  regex: ^([0-9]{1,3}\.){3}[0-9]{1,3}$

network..interfaces..{{ interface }}..netmask:
  type: string
  regex: ^([0-9]{1,3}\.){3}[0-9]{1,3}$

{% endfor %}
{% endif %}
```

Links

For more examples and information see <https://github.com/mickep76/pepa>.

`salt.pillar.pepa.`**`ext_pillar`**(*minion_id, pillar, resource, sequence, subkey=False, subkey_only=False*)
Evaluate Pepa templates

`salt.pillar.pepa.`**`key_value_to_tree`**(*data*)
Convert key/value to tree

`salt.pillar.pepa.`**`validate`**(*output, resource*)
Validate Pepa templates

21.12.28 salt.pillar.pillar_ldap

Use LDAP data as a Pillar source

This pillar module executes a series of LDAP searches. Data returned by these searches are aggregated, whereby data returned by later searches override data by previous searches with the same key.

The final result is merged with existing pillar data.

The configuration of this external pillar module is done via an external file which provides the actual configuration for the LDAP searches.

Configuring the LDAP ext_pillar

The basic configuration is part of the master configuration.

```
ext_pillar:
  - pillar_ldap: /etc/salt/master.d/pillar_ldap.yaml
```

Note: When placing the file in the `master.d` directory, make sure its name doesn't end in `.conf`, otherwise the salt-master process will attempt to parse its content.

Warning: Make sure this file has very restrictive permissions, as it will contain possibly sensitive LDAP credentials!

The only required key in the master configuration is `pillar_ldap` pointing to a file containing the actual configuration.

Configuring the LDAP searches

The file is processed using *Salt's Renderers <renderers>* which makes it possible to reference grains within the configuration.

> **Warning:** When using Jinja in this file, make sure to do it in a way which prevents leaking sensitive information. A rogue minion could send arbitrary grains to trick the master into returning secret data. Use only the 'id' grain which is verified through the minion's key/cert.

Map Mode

The `it-admins` configuration below returns the Pillar `it-admins` by:

- filtering for: - members of the group `it-admins` - objects with `objectclass=user`
- returning the data of users, where each user is a dictionary containing the configured string or list attributes.

Configuration

```
salt-users:
  server:    ldap.company.tld
  port:      389
  tls:       true
  dn:        'dc=company,dc=tld'
  binddn:    'cn=salt-pillars,ou=users,dc=company,dc=tld'
  bindpw:    bi7ieBai5Ano
  referrals: false
  anonymous: false
  mode:      map
  dn:        'ou=users,dc=company,dc=tld'
  filter:    '(&(memberof=cn=it-admins,ou=groups,dc=company,dc=tld)(objectclass=user))
↪'
  attrs:
    - cn
    - displayName
    - givenName
    - sn
  lists:
    - memberOf

search_order:
  - salt-users
```

Result

```
{
    'salt-users': [
        {
            'cn': 'cn=johndoe,ou=users,dc=company,dc=tld',
            'displayName': 'John Doe'
            'givenName': 'John'
            'sn': 'Doe'
            'memberOf': [
              'cn=it-admins,ou=groups,dc=company,dc=tld',
              'cn=team01,ou=groups,dc=company'
            ]
        },
        {
            'cn': 'cn=janedoe,ou=users,dc=company,dc=tld',
            'displayName': 'Jane Doe',
            'givenName': 'Jane',
            'sn': 'Doe',
            'memberOf': [
              'cn=it-admins,ou=groups,dc=company,dc=tld',
              'cn=team02,ou=groups,dc=company'
            ]
        }
    ]
}
```

`salt.pillar.pillar_ldap.`**`ext_pillar`**(*minion_id*, *pillar*, *config_file*)

Execute LDAP searches and return the aggregated data

21.12.29 salt.pillar.postgres module

Retrieve Pillar data by doing a postgres query

New in version 2017.7.0.

maturity new

depends psycopg2

platform all

Complete Example

```
postgres:
  user: 'salt'
  pass: 'super_secret_password'
  db: 'salt_db'

ext_pillar:
  - postgres:
      fromdb:
        query: 'SELECT col1,col2,col3,col4,col5,col6,col7
                  FROM some_random_table
                 WHERE minion_pattern LIKE %s'
        depth: 5
        as_list: True
        with_lists: [1,3]
```

class `salt.pillar.postgres.`**`POSTGRESExtPillar`**

This class receives and processes the database rows from POSTGRES.

`extract_queries`(*args*, *kwargs*)

This function normalizes the config block into a set of queries we can use. The return is a list of consistently laid out dicts.

`salt.pillar.postgres.`**`ext_pillar`**(*minion_id*, *pillar*, **args*, ***kwargs*)

Execute queries against POSTGRES, merge and return as a dict

21.12.30 salt.pillar.puppet

Execute an unmodified puppet_node_classifier and read the output as YAML. The YAML data is then directly overlaid onto the minion's Pillar data.

`salt.pillar.puppet.`**`ext_pillar`**(*minion_id*, *pillar*, *command*)

Execute an unmodified puppet_node_classifier and read the output as YAML

21.12.31 salt.pillar.reclass_adapter

Use the "reclass" database as a Pillar source

This `ext_pillar` plugin provides access to the **reclass** database, such that Pillar data for a specific minion are fetched using **reclass**.

You can find more information about **reclass** at http://reclass.pantsfullofunix.net.

To use the plugin, add it to the `ext_pillar` list in the Salt master config and tell **reclass** by way of a few options how and where to find the inventory:

```
ext_pillar:
    - reclass:
        storage_type: yaml_fs
        inventory_base_uri: /srv/salt
```

This would cause **reclass** to read the inventory from YAML files in `/srv/salt/nodes` and `/srv/salt/classes`.

If you are also using **reclass** as `master_tops` plugin, and you want to avoid having to specify the same information for both, use YAML anchors (take note of the differing data types for `ext_pillar` and `master_tops`):

```
reclass: &reclass
    storage_type: yaml_fs
    inventory_base_uri: /srv/salt
    reclass_source_path: ~/code/reclass

ext_pillar:
    - reclass: *reclass

master_tops:
    reclass: *reclass
```

If you want to run reclass from source, rather than installing it, you can either let the master know via the `PYTHONPATH` environment variable, or by setting the configuration option, like in the example above.

`salt.pillar.reclass_adapter.`**`ext_pillar`**(*minion_id*, *pillar*, ***kwargs*)

Obtain the Pillar data from **reclass** for the given `minion_id`.

21.12.32 salt.pillar.redismod

Read pillar data from a Redis backend

New in version 2014.7.0.

depends

- redis Python module (on master)

Salt Master Redis Configuration

The module shares the same base Redis connection variables as *salt.returners.redis_return*. These variables go in your master config file.

- `redis.db` - The Redis database to use. Defaults to `0`.
- `redis.host` - The Redis host to connect to. Defaults to `'salt'`.
- `redis.port` - The port that the Redis database is listening on. Defaults to `6379`.
- `redis.password` - The password for authenticating with Redis. Only required if you are using master auth. Defaults to `None`.

Configuring the Redis ext_pillar

```
ext_pillar:
  - redis: {function: key_value}
```

`salt.pillar.redismod.`**`ext_pillar`**(*minion_id, pillar, function, **kwargs*)

Grabs external pillar data based on configured function

`salt.pillar.redismod.`**`key_json`**(*minion_id, pillar, pillar_key=None*)

Pulls a string from redis and deserializes it from json. Deserialized dictionary data loaded directly into top level if pillar_key is not set.

pillar_key Pillar key to return data into

`salt.pillar.redismod.`**`key_value`**(*minion_id, pillar, pillar_key='redis_pillar'*)

Looks for key in redis matching minion_id, returns a structure based on the data type of the redis key. String for string type, dict for hash type and lists for lists, sets and sorted sets.

pillar_key Pillar key to return data into

21.12.33 salt.pillar.s3

Copy pillar data from a bucket in Amazon S3

The S3 pillar can be configured in the master config file with the following options

```
ext_pillar:
  - s3:
      bucket: my.fancy.pillar.bucket
      keyid: KASKFJWAKJASJKDAJKSD
      key: ksladfDLKDALSFKSD93q032sdDasdfasdflsadkf
      multiple_env: False
      environment: base
      prefix: somewhere/overthere
```

(continues on next page)

(continued from previous page)

```
      verify_ssl: True
      service_url: s3.amazonaws.com
      kms_keyid: 01234567-89ab-cdef-0123-4567890abcde
      s3_cache_expire: 30
      s3_sync_on_update: True
      path_style: False
      https_enable: True
```

The `bucket` parameter specifies the target S3 bucket. It is required.

The `keyid` parameter specifies the key id to use when access the S3 bucket. If it is not provided, an attempt to fetch it from EC2 instance meta-data will be made.

The `key` parameter specifies the key to use when access the S3 bucket. If it is not provided, an attempt to fetch it from EC2 instance meta-data will be made.

The `multiple_env` defaults to False. It specifies whether the pillar should interpret top level folders as pillar environments (see mode section below).

The `environment` defaults to 'base'. It specifies which environment the bucket represents when in single environments mode (see mode section below). It is ignored if multiple_env is True.

The `prefix` defaults to ''. It specifies a key prefix to use when searching for data in the bucket for the pillar. It works when multiple_env is True or False. Essentially it tells ext_pillar to look for your pillar data in a 'subdirectory' of your S3 bucket

The `verify_ssl` parameter defaults to True. It specifies whether to check for valid S3 SSL certificates. *NOTE* If you use bucket names with periods, this must be set to False else an invalid certificate error will be thrown (issue #12200).

The `service_url` parameter defaults to 's3.amazonaws.com'. It specifies the base url to use for accessing S3.

The `kms_keyid` parameter is optional. It specifies the ID of the Key Management Service (KMS) master key that was used to encrypt the object.

The `s3_cache_expire` parameter defaults to 30s. It specifies expiration time of S3 metadata cache file.

The `s3_sync_on_update` parameter defaults to True. It specifies if cache is synced on update rather than jit.

The `path_style` parameter defaults to False. It specifies whether to use path style requests or dns style requests

The `https_enable` parameter defaults to True. It specifies whether to use https protocol or http protocol

This pillar can operate in two modes, single environment per bucket or multiple environments per bucket.

Single environment mode must have this bucket structure:

```
s3://<bucket name>/<prefix>/<files>
```

Multiple environment mode must have this bucket structure:

```
s3://<bucket name>/<prefix>/<environment>/<files>
```

If you wish to define your pillar data entirely within S3 it's recommended that you use the *prefix=* parameter and specify one entry in ext_pillar for each environment rather than specifying multiple_env. This is due to issue #22471 (https://github.com/saltstack/salt/issues/22471)

class `salt.pillar.s3.`**`S3Credentials`**(*key, keyid, bucket, service_url, verify_ssl=True, kms_keyid=None, location=None, path_style=False, https_enable=True*) `salt.pillar.s3.`**`ext_pillar`**(*minion_id, pillar, bucket, key=None, keyid=None, verify_ssl=True, location=None, multiple_env=False, environment='base', prefix='', service_url=None, kms_keyid=None, s3_cache_expire=30, s3_sync_on_update=True, path_style=False, https_enable=True*)

Execute a command and read the output as YAML

21.12.34 salt.pillar.saltclass

SaltClass Pillar Module

```
ext_pillar:
  - saltclass:
    - path: /srv/saltclass
```

For additional configuration instructions, see the *saltclass* module

`salt.pillar.saltclass.`**`ext_pillar`**(*minion_id, pillar, *args, **kwargs*)

Compile pillar data

21.12.35 salt.pillar.sql_base module

Retrieve Pillar data by doing a SQL query

This module is not meant to be used directly as an ext_pillar. It is a place to put code common to PEP 249 compliant SQL database adapters. It exposes a python ABC that can be subclassed for new database providers.

maturity new

platform all

Theory of sql_base ext_pillar

Ok, here's the theory for how this works...

- First, any non-keyword args are processed in order.
- Then, remaining keywords are processed.

We do this so that it's backward compatible with older configs. Keyword arguments are sorted before being appended, so that they're predictable, but they will always be applied last so overall it's moot.

For each of those items we process, it depends on the object type:

- Strings are executed as is and the pillar depth is determined by the number of fields returned.
- A list has the first entry used as the query, the second as the pillar depth.
- A mapping uses the keys "query" and "depth" as the tuple

You can retrieve as many fields as you like, how they get used depends on the exact settings.

Configuring a sql_base ext_pillar

The sql_base ext_pillar cannot be used directly, but shares query configuration with its implementations. These examples use a fake 'sql_base' adapter, which should be replaced with the name of the adapter you are using.

A list of queries can be passed in

```
ext_pillar:
  - sql_base:
      - "SELECT pillar,value FROM pillars WHERE minion_id = %s"
      - "SELECT pillar,value FROM more_pillars WHERE minion_id = %s"
```

Or you can pass in a mapping

```
ext_pillar:
  - sql_base:
      main: "SELECT pillar,value FROM pillars WHERE minion_id = %s"
      extras: "SELECT pillar,value FROM more_pillars WHERE minion_id = %s"
```

The query can be provided as a string as we have just shown, but they can be provided as lists

```
ext_pillar:
  - sql_base:
      - "SELECT pillar,value FROM pillars WHERE minion_id = %s"
        2
```

Or as a mapping

```
ext_pillar:
  - sql_base:
      - query: "SELECT pillar,value FROM pillars WHERE minion_id = %s"
        depth: 2
```

The depth defines how the dicts are constructed. Essentially if you query for fields a,b,c,d for each row you'll get:

- With depth 1: {a: {"b": b, "c": c, "d": d}}
- With depth 2: {a: {b: {"c": c, "d": d}}}
- With depth 3: {a: {b: {c: d}}}

Depth greater than 3 wouldn't be different from 3 itself. Depth of 0 translates to the largest depth needed, so 3 in this case. (max depth == key count - 1)

Then they are merged in a similar way to plain pillar data, in the order returned by the SQL database.

Thus subsequent results overwrite previous ones when they collide.

The ignore_null option can be used to change the overwrite behavior so that only non-NULL values in subsequent results will overwrite. This can be used to selectively overwrite default values.

```
ext_pillar:
  - sql_base:
      - query: "SELECT pillar,value FROM pillars WHERE minion_id = 'default' and
↪minion_id != %s"
        depth: 2
      - query: "SELECT pillar,value FROM pillars WHERE minion_id = %s"
        depth: 2
        ignore_null: True
```

If you specify *as_list: True* in the mapping expression it will convert collisions to lists.

If you specify *with_lists: '...'* in the mapping expression it will convert the specified depths to list. The string provided is a sequence numbers that are comma separated. The string '1,3' will result in:

```
a,b,c,d,e,1  # field 1 same, field 3 differs
a,b,c,f,g,2  # ^^^^
a,z,h,y,j,3  # field 1 same, field 3 same
a,z,h,y,k,4  # ^^^^
  ^   ^
```

These columns define list grouping

```
{a: [
      {c: [
          {e: 1},
          {g: 2}
          ]
      },
      {h: [
          {j: 3, k: 4 }
          ]
      }
]}
```

The range for with_lists is 1 to number_of_fields, inclusive. Numbers outside this range are ignored.

Finally, if you pass the queries in via a mapping, the key will be the first level name where as passing them in as a list will place them in the root. This isolates the query results into their own subtrees. This may be a help or hindrance to your aims and can be used as such.

You can basically use any SELECT query that gets you the information, you could even do joins or subqueries in case your minion_id is stored elsewhere. It is capable of handling single rows or multiple rows per minion.

Configuration of the connection depends on the adapter in use.

More complete example for MySQL (to also show configuration)

```
mysql:
  user: 'salt'
  pass: 'super_secret_password'
  db: 'salt_db'

ext_pillar:
  - mysql:
      fromdb:
        query: 'SELECT col1,col2,col3,col4,col5,col6,col7
                  FROM some_random_table
                 WHERE minion_pattern LIKE %s'
        depth: 5
        as_list: True
        with_lists: [1,3]
```

class salt.pillar.sql_base.**SqlBaseExtPillar**

This class receives and processes the database rows in a database agnostic way.

as_list = False

depth = 0

enter_root(*root*)

Set self.focus for kwarg queries

extract_queries(*args*, *kwargs*)
This function normalizes the config block into a set of queries we can use. The return is a list of consistently laid out dicts.

fetch(*minion_id*, *pillar*, **args*, ***kwargs*)
Execute queries, merge and return as a dict.

field_names = None

focus = None

ignore_null = False

num_fields = 0

process_fields(*field_names*, *depth*)
The primary purpose of this function is to store the sql field list and the depth to which we process.

process_results(*rows*)
This function takes a list of database results and iterates over, merging them into a dict form.

result = None

with_lists = None

21.12.36 salt.pillar.sqlcipher module

Retrieve Pillar data by running a SQLCipher query

New in version 2016.3.0.

Python SQLCipher support is provided by the pysqlcipher Python package. You need this module installed to query Pillar data from a SQLCipher database.

This module is a concrete implementation of the sql_base ext_pillar for SQLCipher.

maturity new

depends pysqlcipher (for py2) or pysqlcipher3 (for py3)

platform all

Configuring the sqlcipher ext_pillar

Use the 'sqlcipher' key under ext_pillar for configuration of queries.

SQLCipher database connection configuration requires the following values configured in the master config:

- `sqlcipher.database` - The SQLCipher database to connect to. Defaults to `'/var/lib/salt/pillar-sqlcipher.db'`.
- `sqlcipher.pass` - The SQLCipher database decryption password.
- `sqlcipher.timeout` - The connection timeout in seconds.

Example configuration

```
sqlcipher:
  database: /var/lib/salt/pillar-sqlcipher.db
  pass: strong_pass_phrase
  timeout: 5.0
```

Complete Example

```
sqlcipher:
  database: '/var/lib/salt/pillar-sqlcipher.db'
  pass: strong_pass_phrase
  timeout: 5.0

ext_pillar:
  - sqlcipher:
      fromdb:
        query: 'SELECT col1,col2,col3,col4,col5,col6,col7
                  FROM some_random_table
                 WHERE minion_pattern LIKE ?'
        depth: 5
        as_list: True
        with_lists: [1,3]
```

class `salt.pillar.sqlcipher.`**`SQLCipherExtPillar`**

This class receives and processes the database rows from SQLCipher.

`salt.pillar.sqlcipher.`**`ext_pillar`**(*minion_id*, *pillar*, **args*, ***kwargs*)

Execute queries against SQLCipher, merge and return as a dict

21.12.37 salt.pillar.sqlite3 module

Retrieve Pillar data by doing a SQLite3 query

New in version 2015.8.0.

`sqlite3` is included in the stdlib since Python 2.5.

This module is a concrete implementation of the sql_base ext_pillar for SQLite3.

platform all

Configuring the sqlite3 ext_pillar

Use the 'sqlite3' key under ext_pillar for configuration of queries.

SQLite3 database connection configuration requires the following values configured in the master config:

Note, timeout is in seconds.

```
sqlite3.database: /var/lib/salt/pillar.db
sqlite3.timeout: 5.0
```

Complete Example

```
sqlite3:
  database: '/var/lib/salt/pillar.db'
  timeout: 5.0

ext_pillar:
  - sqlite3:
      fromdb:
```

(continues on next page)

(continued from previous page)

```
        query: 'SELECT col1,col2,col3,col4,col5,col6,col7
                 FROM some_random_table
                WHERE minion_pattern LIKE ?'
        depth: 5
        as_list: True
        with_lists: [1,3]
```

class `salt.pillar.sqlite3.`**`SQLite3ExtPillar`**

This class receives and processes the database rows from SQLite3.

`salt.pillar.sqlite3.`**`ext_pillar`**(*minion_id*, *pillar*, **args*, ***kwargs*)

Execute queries against SQLite3, merge and return as a dict

21.12.38 salt.pillar.stack

Simple and flexible YAML ext_pillar which can read pillar from within pillar.

New in version 2016.3.0.

PillarStack is a custom saltstack `ext_pillar` which was inspired by varstack but is heavily based on Jinja2 for maximum flexibility.

Any issue should be reported to the upstream project at: https://github.com/bbinet/pillarstack/issues

It supports the following features:

- multiple config files that are jinja2 templates with support for `pillar`, `__grains__`, `__salt__`, `__opts__` objects
- a config file renders as an ordered list of files (paths of these files are relative to the current config file)
- this list of files are read in ordered as jinja2 templates with support for `stack`, `pillar`, `__grains__`, `__salt__`, `__opts__` objects
- all these rendered files are then parsed as `yaml`
- then all yaml dicts are merged in order with support for the following merging strategies: `merge-first`, `merge-last`, `remove`, and `overwrite`
- stack config files can be matched based on `pillar`, `grains`, or `opts` values, which make it possible to support kind of self-contained environments

Installation

PillarStack is already bundled with Salt since 2016.3.0 version so there is nothing to install from version 2016.3.0.

If you use an older Salt version or you want to override PillarStack with a more recent one, follow the installation procedure below.

Installing the PillarStack `ext_pillar` is as simple as dropping the `stack.py` file in the `<extension_modules>/pillar` directory (no external python module required), given that `extension_modules` is set in your salt-master configuration, see: http://docs.saltstack.com/en/latest/ref/configuration/master.html#extension-modules

Configuration in Salt

Like any other external pillar, its configuration takes place through the `ext_pillar` key in the master config file.

However, you can configure PillarStack in 3 different ways:

Single config file

This is the simplest option, you just need to set the path to your single PillarStack config file like below:

```
ext_pillar:
  - stack: /path/to/stack.cfg
```

List of config files

You can also provide a list of config files:

```
ext_pillar:
  - stack:
      - /path/to/stack1.cfg
      - /path/to/stack2.cfg
```

Select config files through grains|pillar|opts matching

You can also opt for a much more flexible configuration: PillarStack allows one to select the config files for the current minion based on matching values from either grains, or pillar, or opts objects.

Here is an example of such a configuration, which should speak by itself:

```
ext_pillar:
  - stack:
      pillar:environment:
        dev: /path/to/dev/stack.cfg
        prod: /path/to/prod/stack.cfg
      grains:custom:grain:
        value:
          - /path/to/stack1.cfg
          - /path/to/stack2.cfg
      opts:custom:opt:
        value: /path/to/stack0.cfg
```

PillarStack configuration files

The config files that are referenced in the above `ext_pillar` configuration are jinja2 templates which must render as a simple ordered list of `yaml` files that will then be merged to build pillar data.

The path of these `yaml` files must be relative to the directory of the PillarStack config file. These paths support unix style pathname pattern expansion through the *Python glob module <https://docs.python.org/2/library/glob.html>*.

The following variables are available in jinja2 templating of PillarStack configuration files:

- `pillar`: the pillar data (as passed by Salt to our `ext_pillar` function)
- `minion_id`: the minion id ;-)

- `__opts__`: a dictionary of mostly Salt configuration options
- `__grains__`: a dictionary of the grains of the minion making this pillar call
- `__salt__`: a dictionary of Salt module functions, useful so you don't have to duplicate functions that already exist (note: runs on the master)

So you can use all the power of jinja2 to build your list of `yaml` files that will be merged in pillar data.

For example, you could have a PillarStack config file which looks like:

```
$ cat /path/to/stack/config.cfg
core.yml
common/*.yml
osarchs/{{ __grains__['osarch'] }}.yml
oscodenames/{{ __grains__['oscodename'] }}.yml
{%- for role in pillar.get('roles', []) %}
roles/{{ role }}.yml
{%- endfor %}
minions/{{ minion_id }}.yml
```

And the whole directory structure could look like:

```
$ tree /path/to/stack/
/path/to/stack/
├── config.cfg
├── core.yml
├── common/
│   ├── xxx.yml
│   └── yyy.yml
├── osarchs/
│   ├── amd64.yml
│   └── armhf.yml
├── oscodenames/
│   ├── wheezy.yml
│   └── jessie.yml
├── roles/
│   ├── web.yml
│   └── db.yml
└── minions/
    ├── test-1-dev.yml
    └── test-2-dev.yml
```

Overall process

In the above PillarStack configuration, given that test-1-dev minion is an amd64 platform running Debian Jessie, and which pillar `roles` is `["db"]`, the following `yaml` files would be merged in order:

- `core.yml`
- `common/xxx.yml`
- `common/yyy.yml`
- `osarchs/amd64.yml`
- `oscodenames/jessie.yml`

- `roles/db.yml`
- `minions/test-1-dev.yml`

Before merging, every files above will be preprocessed as Jinja2 templates. The following variables are available in Jinja2 templating of `yaml` files:

- `stack`: the PillarStack pillar data object that has currently been merged (data from previous `yaml` files in PillarStack configuration)
- `pillar`: the pillar data (as passed by Salt to our `ext_pillar` function)
- `minion_id`: the minion id ;-)
- `__opts__`: a dictionary of mostly Salt configuration options
- `__grains__`: a dictionary of the grains of the minion making this pillar call
- `__salt__`: a dictionary of Salt module functions, useful so you don't have to duplicate functions that already exist (note: runs on the master)

So you can use all the power of jinja2 to build your pillar data, and even use other pillar values that has already been merged by PillarStack (from previous `yaml` files in PillarStack configuration) through the `stack` variable.

Once a `yaml` file has been preprocessed by Jinja2, we obtain a Python dict - let's call it `yml_data` - then, PillarStack will merge this `yml_data` dict in the main `stack` dict (which contains already merged PillarStack pillar data). By default, PillarStack will deeply merge `yml_data` in `stack` (similarly to the `recurse` salt `pillar_source_merging_strategy`), but 3 merging strategies are currently available for you to choose (see next section).

Once every `yaml` files have been processed, the `stack` dict will contain your whole own pillar data, merged in order by PillarStack. So PillarStack `ext_pillar` returns the `stack` dict, the contents of which Salt takes care to merge in with all of the other pillars and finally return the whole pillar to the minion.

Merging strategies

The way the data from a new `yaml_data` dict is merged with the existing `stack` data can be controlled by specifying a merging strategy. Right now this strategy can either be `merge-last` (the default), `merge-first`, `remove`, or `overwrite`.

Note that scalar values like strings, integers, booleans, etc. are always evaluated using the `overwrite` strategy (other strategies don't make sense in that case).

The merging strategy can be set by including a dict in the form of:

```
__: <merging strategy>
```

as the first item of the dict or list. This allows fine grained control over the merging process.

`merge-last` (default) strategy

If the `merge-last` strategy is selected (the default), then content of dict or list variables is merged recursively with previous definitions of this variable (similarly to the `recurse` salt `pillar_source_merging_strategy`). This allows for extending previously defined data.

`merge-first` strategy

If the `merge-first` strategy is selected, then the content of dict or list variables are swapped between the `yaml_data` and `stack` objects before being merged recursively with the `merge-last` previous strategy.

`remove` strategy

If the `remove` strategy is selected, then content of dict or list variables in `stack` are removed only if the corresponding item is present in the `yaml_data` dict. This allows for removing items from previously defined data.

`overwrite` strategy

If the `overwrite` strategy is selected, then the content of dict or list variables in `stack` is overwritten by the content of `yaml_data` dict. So this allows one to overwrite variables from previous definitions.

Merging examples

Let's go through small examples that should clarify what's going on when a `yaml_data` dict is merged in the `stack` dict.

When you don't specify any strategy, the default `merge-last` strategy is selected:

`stack`

```yaml
users:
  tom:
    uid: 500
    roles:
      - sysadmin
  root:
    uid: 0
```

`yaml_data`

```yaml
users:
  tom:
    uid: 1000
    roles:
      - developer
  mat:
    uid: 1001
```

`stack` (after merge)

```yaml
users:
  tom:
    uid: 1000
    roles:
      - sysadmin
      - developer
  mat:
    uid: 1001
  root:
    uid: 0
```

Then you can select a custom merging strategy using the `__` key in a dict:

`stack` | `yaml_data` | `stack` (after merge)

```yaml
users:
  tom:
    uid: 500
    roles:
      - sysadmin
  root:
    uid: 0
```

```yaml
users:
  __: merge-last
  tom:
    uid: 1000
    roles:
      - developer
  mat:
    uid: 1001
```

```yaml
users:
  tom:
    uid: 1000
    roles:
      - sysadmin
      - developer
  mat:
    uid: 1001
  root:
    uid: 0
```

```yaml
users:
  tom:
    uid: 500
    roles:
      - sysadmin
  root:
    uid: 0
```

```yaml
users:
  __: merge-first
  tom:
    uid: 1000
    roles:
      - developer
  mat:
    uid: 1001
```

```yaml
users:
  tom:
    uid: 500
    roles:
      - developer
      - sysadmin
  mat:
    uid: 1001
  root:
    uid: 0
```

```yaml
users:
  tom:
    uid: 500
    roles:
      - sysadmin
  root:
    uid: 0
```

```yaml
users:
  __: remove
  tom:
  mat:
```

```yaml
users:
  root:
    uid: 0
```

```yaml
users:
  tom:
    uid: 500
    roles:
      - sysadmin
  root:
    uid: 0
```

```yaml
users:
  __: overwrite
  tom:
    uid: 1000
    roles:
      - developer
  mat:
    uid: 1001
```

```yaml
users:
  tom:
    uid: 1000
    roles:
      - developer
  mat:
    uid: 1001
```

You can also select a custom merging strategy using a `__` object in a list:

`stack` | `yaml_data` | `stack` (after merge)

`stack`
```
users:
  - tom
  - root
```
`yaml_data`
```
users:
  - __: merge-last
  - mat
```
`stack` (after merge)
```
users:
  - tom
  - root
  - mat
```

`stack`
```
users:
  - tom
  - root
```
`yaml_data`
```
users:
  - __: merge-first
  - mat
```
`stack` (after merge)
```
users:
  - mat
  - tom
  - root
```

`stack`
```
users:
  - tom
  - root
```
`yaml_data`
```
users:
  - __: remove
  - mat
  - tom
```
`stack` (after merge)
```
users:
  - root
```

`stack`
```
users:
  - tom
  - root
```
`yaml_data`
```
users:
  - __: overwrite
  - mat
```
`stack` (after merge)
```
users:
  - mat
```

`salt.pillar.stack.`**`ext_pillar`**(*minion_id*, *pillar*, **args*, ***kwargs*)

21.12.39 salt.pillar.svn_pillar

Clone a remote SVN repository and use the filesystem as a Pillar source

This external Pillar source can be configured in the master config file like so:

```
ext_pillar:
  - svn: trunk svn://svnserver/repo root=subdirectory
```

The *root=* parameter is optional and used to set the subdirectory from where to look for Pillar files (such as `top.sls`).

Changed in version 2014.7.0: The optional `root` parameter will be added.

Note that this is not the same thing as configuring pillar data using the *`pillar_roots`* parameter. The branch referenced in the *`ext_pillar`* entry above (`master`), would evaluate to the `base` environment, so this branch needs to contain a `top.sls` with a `base` section in it, like this:

```
base:
  '*':
    - foo
```

To use other environments from the same SVN repo as svn_pillar sources, just add additional lines, like so:

```
ext_pillar:
  - svn: trunk svn://svnserver/repo
  - svn: dev svn://svnserver/repo
```

In this case, the `dev` branch would need its own `top.sls` with a `dev` section in it, like this:

```
dev:
  '*':
    - bar
```

class salt.pillar.svn_pillar.**SvnPillar**(*branch*, *repo_location*, *root*, *opts*)

Deal with the remote SVN repository for Pillar

pillar_dir()

Returns the directory of the pillars (repo cache + branch + root)

update()

salt.pillar.svn_pillar.**ext_pillar**(*minion_id*, *pillar*, *repo_string*)

Execute a command and read the output as YAML

21.12.40 salt.pillar.varstack_pillar

Use Varstack data as a Pillar source

Configuring Varstack

Using varstack in Salt is fairly simple. Just put the following into the config file of your master:

```
ext_pillar:
  - varstack: /etc/varstack.yaml
```

Varstack will then use /etc/varstack.yaml to determine which configuration data to return as pillar information. From there you can take a look at the README of varstack on how this file is evaluated.

salt.pillar.varstack_pillar.**ext_pillar**(*minion_id*, *pillar*, *conf*)

Parse varstack data and return the result

21.12.41 salt.pillar.vault module

Vault Pillar Module

maintainer SaltStack

maturity New

platform all

New in version 2016.11.0.

This module allows pillar data to be stored in Hashicorp Vault.

Base configuration instructions are documented in the *execution module docs*. Below are noted extra configuration required for the pillar module, but the base configuration must also be completed.

After the base Vault configuration is created, add the configuration below to the ext_pillar section in the Salt master configuration.

```
ext_pillar:
  - vault: path=secret/salt
```

Each key needs to have all the key-value pairs with the names you require. Avoid naming every key 'password' as you they will collide:

If you want to nest results under a nesting_key name use the following format:

ext_pillar:

- **vault**: conf: path=secret/salt nesting_key: vault_key_name

```
$ vault write secret/salt auth=my_password master=127.0.0.1
```

The above will result in two pillars being available, `auth` and `master`.

You can then use normal pillar requests to get each key pair directly from pillar root. Example:

```
$ salt-ssh '*' pillar.get auth
```

Multiple Vault sources may also be used:

```
ext_pillar:
  - vault: path=secret/salt
  - vault: path=secret/root
  - vault: path=secret/minions/{minion}/pass
```

You can also use nesting here as well. Identical nesting keys will get merged.

```
ext_pillar:
  - vault:
      conf: path=secret/salt
      nesting_key: keyname1
  - vault:
      conf: path=secret/root
      nesting_key: keyname1
  - vault:
      conf: path=secret/minions/{minion}/pass
      nesting_key: keyname2
```

The difference between the return with and without the nesting key is shown below. This example takes the key value pairs returned from vault as follows:

path=secret/salt

Key Value — —– salt-passwd badpasswd1

path=secret/root

Key Value — —– root-passwd rootbadpasswd1

path=secret/minions/{minion}/pass

Key Value — —– minion-passwd minionbadpasswd1

```
#Nesting Key not defined

local:
    ----------
    salt-passwd:
        badpasswd1
    root-passwd:
        rootbadpasswd1
    minion-passwd:
```

(continues on next page)

(continued from previous page)

```
        minionbadpasswd1

#Nesting Key defined

local:
    ----------
    keyname1:
        ----------
            salt-passwd:
                badpasswd1
            root-passwd:
                rootbadpasswd1
    keyname2:
        ----------
            minion-passwd:
                minionbadpasswd1
```

`salt.pillar.vault.`**`ext_pillar`**(*minion_id*, *pillar*, *conf*, *nesting_key=None*)
Get pillar data from Vault for the configuration `conf`.

21.12.42 salt.pillar.venafi module

Venafi Pillar Certificates

This module will only return pillar data if the `venafi` runner module has already been used to create certificates.

To configure this module, set `venafi` to `True` in the `ext_pillar` section of your `master` configuration file:

```
ext_pillar:
  - venafi: True
```

`salt.pillar.venafi.`**`ext_pillar`**(*minion_id*, *pillar*, *conf*)
Return an existing set of certificates

21.12.43 salt.pillar.virtkey

Accept a key from a hypervisor if the virt runner has already submitted an authorization request

`salt.pillar.virtkey.`**`ext_pillar`**(*hyper_id*, *pillar*, *name*, *key*)
Accept the key for the VM on the hyper, if authorized.

21.12.44 salt.pillar.vmware_pillar module

Pillar data from vCenter or an ESXi host

New in version 2017.7.0.

depends

- pyVmomi

This external pillar can pull attributes from objects in vCenter or an ESXi host and provide those attributes as pillar data to minions. This can allow for pillar based targeting of minions on ESXi host, Datastore, VM configuration, etc. This setup requires only the salt master have access to the vCenter server/ESXi hosts.

The pillar will return an empty dict if the 'os' or 'virtual' grain are not 'VMWare', 'ESXi', or 'VMWare ESXi'.

Defaults

- The external pillar will search for Virtual Machines with the VM name matching the minion id.
- Data will be returned into the 'vmware' pillar key.
- The external pillar has a default set of properties to return for both VirtualMachine and HostSystem types.

Configuring the VMWare pillar

The required minimal configuration in the salt master ext_pillar setup:

```
ext_pillar:
    - vmware:
        host: <vcenter/esx host>
        username: <user to connect with>
        password: <password>
```

Optionally, the following keyword arguments can be passed to the ext_pillar for customized configuration:

pillar_key Optionally set the pillar key to return the data into. Default is `vmware`.

protocol Optionally set to alternate protocol if the vCenter server or ESX/ESXi host is not using the default protocol. Default protocol is `https`.

port Optionally set to alternate port if the vCenter server or ESX/ESXi host is not using the default port. Default port is `443`.

property_name Property name to match the minion id against. Defaults to `name`.

property_types Optionally specify a list of pyVmomi vim types to search for the minion id in 'property_name'. Default is `['VirtualMachine']`.

For example, to search both vim.VirtualMachine and vim.HostSystem object types:

```
ext_pillar:
    - vmware:
        host: myesx
        username: root
        password: complex_password
        property_types:
          - VirtualMachine
          - HostSystem
```

Additionally, the list of property types can be dicts, the item of the dict being a list specifying the attribute to return for that vim object type.

The pillar will attempt to recurse the attribute and return all child attributes.

To explicitly specify deeper attributes without attempting to recurse an attribute, convert the list item to a dict with the item of the dict being the child attributes to return. Follow this pattern to return attributes as deep within the object as necessary.

Note: Be careful when specifying custom attributes! Many attributes have objects as attributes which have the parent object as an attribute and which will cause the pillar to fail due to the attempt to convert all sub-objects recursively (i.e. infinite attribute loops). Specifying only the sub-attributes you would like returned will keep the infinite recursion from occurring.

A maximum recursion exception will occur in this case and the pillar will not return as desired.

```
ext_pillar:
    - vmware:
        host: myvcenter
        username: my_user
        password: my_pass
        replace_default_attributes: True
        property_types:
          - VirtualMachine:
              - config:
                 - bootOptions:
                     - bootDelay
                     - bootRetryDelay
          - HostSystem:
              - datastore:
                 - name
```

The above ext_pillar example would return a pillar like the following for a VirtualMachine object that's name matched the minion id:

```
vmware:
  config:
    bootOptions:
      bootDelay: 1000
      bootRetryDelay: 1000
```

If you were to retrieve these virtual machine attributes via pyVmomi directly, this would be the same as

```
vmObject.config.bootOptions.bootDelay
vmObject.config.bootOptionis.bootRetryDelay
```

The above ext_pillar example would return a pillar like the following for a HostySystem object that's name matched the minion id:

```
vmware:
  datastore:
     - name: Datastore1
     - name: Datastore2
```

The 'datastore' property of a HostSystem object is a list of datastores, thus a list is returned.

replace_default_attributes If custom attributes are specified by the property_types parameter, replace_default_attributes determines if those will be added to default attributes (False) or replace the default attributes completely (True). The default setting is 'False'.

Note: vCenter "Custom Attributes" (i.e. Annotations) will always be returned if it exists on the object as part of the pillar regardless of this setting.

`salt.pillar.vmware_pillar.`**`ext_pillar`**`(`*minion_id*, *pillar*, ***kwargs*`)`
Check vmware/vcenter for all data

21.13 proxy modules

arista_pyeapi	Arista pyeapi
cimc	Proxy Minion interface module for managing Cisco Integrated Management Controller devices
chronos	Chronos
cimc	Proxy Minion interface module for managing Cisco Integrated Management Controller devices
cisconso	Proxy Minion interface module for managing (practically) any network device with Cisco Network Services Orchestrator (Cisco NSO).
dummy	This is the a dummy proxy-minion designed for testing the proxy minion subsystem.
esxi	Proxy Minion interface module for managing VMware ESXi hosts.
fx2	Dell FX2 chassis
junos	Interface with a Junos device via proxy-minion.
marathon	Marathon
napalm	NAPALM: Network Automation and Programmability Abstraction Layer with Multivendor support
netmiko_px	Netmiko
nxos	Proxy Minion for Cisco NX OS Switches
nxos_api	Proxy Minion to manage Cisco Nexus Switches (NX-OS) over the NX-API
panos	Proxy Minion interface module for managing Palo Alto firewall devices
philips_hue	Philips HUE lamps module for proxy.
rest_sample	This is a simple proxy-minion designed to connect to and communicate with the bottle-based web service contained in https://github.com/saltstack/salt-contrib/tree/master/proxyminion_rest_example
ssh_sample	This is a simple proxy-minion designed to connect to and communicate with a server that exposes functionality via SSH.

21.13.1 salt.proxy.arista_pyeapi

Arista pyeapi

New in version 2019.2.0.

Proxy module for managing Arista switches via the eAPI using the pyeapi library.

codeauthor Mircea Ulinic <ping@mirceaulinic.net>

maturity new

depends pyeapi

platform unix

Note: To understand how to correctly enable the eAPI on your switch, please check https://eos.arista.com/arista-eapi-101/.

Dependencies

The `pyeapi` Proxy module requires pyeapi to be installed: `pip install pyeapi`.

Pillar

The `pyeapi` proxy configuration requires the following parameters in order to connect to the network device:

transport: `https` Specifies the type of connection transport to use. Valid values for the connection are `socket`, `http_local`, `http`, and `https`.

host: `localhost` The IP address or DNS host name of the connection device.

username: `admin` The username to pass to the device to authenticate the eAPI connection.

password The password to pass to the device to authenticate the eAPI connection.

port The TCP port of the endpoint for the eAPI connection. If this keyword is not specified, the default value is automatically determined by the transport type (`80` for `http`, or `443` for `https`).

enablepwd The enable mode password if required by the destination node.

All the arguments may be optional, depending on your setup.

Proxy Pillar Example

```
proxy:
  proxytype: pyeapi
  host: router1.example.com
  username: example
  password: example
```

`salt.proxy.arista_pyeapi.`**`call`**`(`*method*, **args*, ***kwargs*`)`
Calls an arbitrary pyeapi method.

`salt.proxy.arista_pyeapi.`**`conn`**`()`
Return the connection object.

`salt.proxy.arista_pyeapi.`**`init`**`(`*opts*`)`
Open the connection to the Arista switch over the eAPI.

`salt.proxy.arista_pyeapi.`**`initialized`**`()`
Connection finished initializing?

`salt.proxy.arista_pyeapi.`**`ping`**`()`
Connection open successfully?

`salt.proxy.arista_pyeapi.`**`shutdown`**`(`*opts*`)`
Closes connection with the device.

21.13.2 salt.proxy.cimc module

Proxy Minion interface module for managing Cisco Integrated Management Controller devices

New in version 2018.3.0.

codeauthor `Spencer Ervin <spencer_ervin@hotmail.com>`

maturity new

depends none

platform unix

This proxy minion enables Cisco Integrated Management Controller devices (hereafter referred to as simply 'cimc' devices to be treated individually like a Salt Minion.

The cimc proxy leverages the XML API functionality on the Cisco Integrated Management Controller. The Salt proxy must have access to the cimc on HTTPS (tcp/443).

More in-depth conceptual reading on Proxy Minions can be found in the *Proxy Minion* section of Salt's documentation.

Configuration

To use this integration proxy module, please configure the following:

Pillar

Proxy minions get their configuration from Salt's Pillar. Every proxy must have a stanza in Pillar and a reference in the Pillar top-file that matches the ID.

```
proxy:
  proxytype: cimc
  host: <ip or dns name of cimc host>
  username: <cimc username>
  password: <cimc password>
```

proxytype

The `proxytype` key and value pair is critical, as it tells Salt which interface to load from the `proxy` directory in Salt's install hierarchy, or from `/srv/salt/_proxy` on the Salt Master (if you have created your own proxy module, for example). To use this cimc Proxy Module, set this to `cimc`.

host

The location, or ip/dns, of the cimc host. Required.

username

The username used to login to the cimc host. Required.

password

The password used to login to the cimc host. Required.

`salt.proxy.cimc.`**`get_config_resolver_class`**(*cid=None*, *hierarchical=False*)

The configResolveClass method returns requested managed object in a given class.

`salt.proxy.cimc.`**`grains`**`()`
Get the grains from the proxied device

`salt.proxy.cimc.`**`grains_refresh`**`()`
Refresh the grains from the proxied device

`salt.proxy.cimc.`**`init`**`(`*opts*`)`
This function gets called when the proxy starts up.

`salt.proxy.cimc.`**`initialized`**`()`
Since grains are loaded in many different places and some of those places occur before the proxy can be initialized, return whether our init() function has been called

`salt.proxy.cimc.`**`logon`**`()`
Logs into the cimc device and returns the session cookie.

`salt.proxy.cimc.`**`logout`**`(`*cookie=None*`)`
Closes the session with the device.

`salt.proxy.cimc.`**`ping`**`()`
Returns true if the device is reachable, else false.

`salt.proxy.cimc.`**`prepare_return`**`(`*x*`)`
Converts the etree to dict

`salt.proxy.cimc.`**`set_config_modify`**`(`*dn=None, inconfig=None, hierarchical=False*`)`
The configConfMo method configures the specified managed object in a single subtree (for example, DN).

`salt.proxy.cimc.`**`shutdown`**`()`
Shutdown the connection to the proxy device. For this proxy, shutdown is a no-op.

21.13.3 salt.proxy.chronos module

Chronos

Proxy minion for managing a Chronos cluster.

Dependencies

- *`chronos execution module (salt.modules.chronos)`*

Pillar

The chronos proxy configuration requires a 'base_url' property that points to the chronos endpoint:

```
proxy:
  proxytype: chronos
  base_url: http://my-chronos-master.mydomain.com:4400
```

New in version 2015.8.2.

`salt.proxy.chronos.`**`init`**`(`*opts*`)`
Perform any needed setup.

`salt.proxy.chronos.`**`ping`**`()`
Is the chronos api responding?

`salt.proxy.chronos.`**`shutdown`**`(`*opts*`)`

For this proxy shutdown is a no-op

21.13.4 salt.proxy.cisconso

Proxy Minion interface module for managing (practically) any network device with Cisco Network Services Orchestrator (Cisco NSO). Cisco NSO uses a series of remote polling agents, APIs and SSH commands to fetch network configuration and represent it in a data model. PyNSO, the Python module used by this proxy minion does the task of converting native Python dictionaries into NETCONF/YANG syntax that the REST API for Cisco NSO can then use to set the configuration of the target network device.

Supported devices:

- A10 AX Series
- Arista 7150 Series
- Ciena 3000, 5000, ESM
- H3c S5800 Series
- Overture 1400, 2200, 5000, 5100, 6000
- Accedian MetroNID
- Avaya ERS 4000, SR8000, VSP 9000
- **Cisco: APIC-DC, ASA, IOS, IOS XE, IOS XR, er, ME-4600, NX OS,** Prime Network Registrar, Quantum, StarOS, UCS ManagWSA
- Huawei: NE40E, quidway series, Enterprise Network Simulation Framework
- PaloAlto PA-2000, PA-3000, Virtualized Firewalls
- Adtran 900 Series
- Brocade ADX, MLX, Netiron, Vyatta
- Dell Force 10 Networking S-Series
- Infinera DTN-X Multi-Terabit Packet Optical Network Platform
- Pulsecom SuperG
- Adva 150CC Series
- CableLabs Converged Cable Access Platform
- Ericsson EFN324 Series, SE family
- Juniper: Contrail, EX, M, MX, QFX, SRX, Virtual SRX
- Quagga Routing Software
- Affirmed Networks
- Citrix Netscaler
- F5 BIG-IP
- NEC iPasolink
- Riverbed Steelhead Series
- Alcatel-Lucent 7XXX, SAM
- Clavister

- Fortinet
- Nominum DCS
- Sonus SBC 5000 Series
- Allied Telesys
- Open vSwitch

New in version 2016.11.0.

codeauthor *Anthony Shaw <anthony.shaw@dimensiondata.com>*

This proxy minion enables a consistent interface to fetch, control and maintain the configuration of network devices via a NETCONF-compliant control plane. Cisco Network Services Orchestrator.

More in-depth conceptual reading on Proxy Minions can be found in the *Proxy Minion* section of Salt's documentation.

Dependencies

- pynso Python module

PyNSO

PyNSO can be installed via pip:

```
pip install pynso
```

Configuration

To use this integration proxy module, please configure the following:

Pillar

Proxy minions get their configuration from Salt's Pillar. Every proxy must have a stanza in Pillar and a reference in the Pillar top-file that matches the ID. At a minimum for communication with the NSO host, the pillar should look like this:

```
proxy:
  proxytype: cisconso
  host: <ip or dns name of host>
  port: 8080
  use_ssl: false
  username: <username>
  password: password
```

proxytype

The `proxytype` key and value pair is critical, as it tells Salt which interface to load from the `proxy` directory in Salt's install hierarchy, or from `/srv/salt/_proxy` on the Salt Master (if you have created your own proxy module, for example). To use this Cisco NSO Proxy Module, set this to `cisconso`.

host

The location, or IP/dns, of the Cisco NSO API host. Required.

username

The username used to login to the Cisco NSO host, such as `admin`. Required.

passwords

The password for the given user. Required.

use_ssl

Whether to use HTTPS messaging to speak to the API.

port

The port that the Cisco NSO API is running on, 8080 by default

Salt Proxy

After your pillar is in place, you can test the proxy. The proxy can run on any machine that has network connectivity to your Salt Master and to the Cisco NSO host in question. SaltStack recommends that the machine running the salt-proxy process also run a regular minion, though it is not strictly necessary.

On the machine that will run the proxy, make sure there is an `/etc/salt/proxy` file with at least the following in it:

```
master: <ip or hostname of salt-master>
```

You can then start the salt-proxy process with:

```
salt-proxy --proxyid <id you want to give the host>
```

You may want to add `-l debug` to run the above in the foreground in debug mode just to make sure everything is OK.

Next, accept the key for the proxy on your salt-master, just like you would for a regular minion:

```
salt-key -a <id you gave the cisconso host>
```

You can confirm that the pillar data is in place for the proxy:

```
salt <id> pillar.items
```

And now you should be able to ping the Cisco NSO host to make sure it is responding:

```
salt <id> test.ping
```

`salt.proxy.cisconso.`**`apply_rollback`**`(`*datastore*, *name*`)`

Apply a system rollback

Parameters

- **`datastore`** (`DatastoreType` (`str` enum).) – The datastore, e.g. running, operational. One of the NETCONF store IETF types
- **`name`** (`str`) – an ID of the rollback to restore

`salt.proxy.cisconso.`**`get_data`**`(`*datastore*, *path*`)`

Get the configuration of the device tree at the given path

Parameters

- **`datastore`** (`DatastoreType` (`str` enum).) – The datastore, e.g. running, operational. One of the NETCONF store IETF types
- **`path`** (`list` of `str` OR `tuple`) – The device path, a list of element names in order, comma separated

Returns The network configuration at that tree

Return type `dict`

```
salt cisco-nso cisconso.get_data devices
```

`salt.proxy.cisconso.`**`get_rollback`**`(`*name*`)`

Get the backup of stored a configuration rollback

Parameters **`name`** (`str`) – Typically an ID of the backup

Return type `str`

Returns the contents of the rollback snapshot

`salt.proxy.cisconso.`**`get_rollbacks`**`()`

Get a list of stored configuration rollbacks

`salt.proxy.cisconso.`**`grains`**`()`

Get the grains from the proxy device.

`salt.proxy.cisconso.`**`init`**`(`*opts*`)`

`salt.proxy.cisconso.`**`ping`**`()`

Check to see if the host is responding. Returns False if the host didn't respond, True otherwise.

CLI Example:

```
salt cisco-nso test.ping
```

`salt.proxy.cisconso.`**`set_data_value`**`(`*datastore*, *path*, *data*`)`

Get a data entry in a datastore

Parameters

- **`datastore`** (`DatastoreType` (`str` enum).) – The datastore, e.g. running, operational. One of the NETCONF store IETF types
- **`path`** (`list` of `str` OR `tuple`) – The device path to set the value at, a list of element names in order, comma separated
- **`data`** (`dict`) – The new value at the given path

Return type `bool`

Returns `True` if successful, otherwise error.

`salt.proxy.cisconso.`**`shutdown`**`()`

Shutdown the connection to the proxy device. For this proxy, shutdown is a no-op.

21.13.5 salt.proxy.dummy module

This is the a dummy proxy-minion designed for testing the proxy minion subsystem.

`salt.proxy.dummy.`**`fns`**`()`

`salt.proxy.dummy.`**`grains`**`()`
Make up some grains

`salt.proxy.dummy.`**`grains_refresh`**`()`
Refresh the grains

`salt.proxy.dummy.`**`init`**`(`*opts*`)`

`salt.proxy.dummy.`**`initialized`**`()`
Since grains are loaded in many different places and some of those places occur before the proxy can be initialized, return whether our init() function has been called

`salt.proxy.dummy.`**`package_install`**`(`*name*, ***kwargs*`)`
Install a "package" on the REST server

`salt.proxy.dummy.`**`package_list`**`()`
List "packages" installed on the REST server

`salt.proxy.dummy.`**`package_remove`**`(`*name*`)`
Remove a "package" on the REST server

`salt.proxy.dummy.`**`package_status`**`(`*name*`)`
Check the installation status of a package on the REST server

`salt.proxy.dummy.`**`ping`**`()`
Degenerate ping

`salt.proxy.dummy.`**`service_list`**`()`
List "services" on the REST server

`salt.proxy.dummy.`**`service_restart`**`(`*name*`)`
Restart a "service" on the REST server

`salt.proxy.dummy.`**`service_start`**`(`*name*`)`
Start a "service" on the dummy server

`salt.proxy.dummy.`**`service_status`**`(`*name*`)`
Check if a service is running on the REST server

`salt.proxy.dummy.`**`service_stop`**`(`*name*`)`
Stop a "service" on the dummy server

`salt.proxy.dummy.`**`shutdown`**`(`*opts*`)`
For this proxy shutdown is a no-op

`salt.proxy.dummy.`**`test_from_state`**`()`
Test function so we have something to call from a state :return:

`salt.proxy.dummy.`**`upgrade`**`()`
"Upgrade" packages

`salt.proxy.dummy.`**`uptodate`**`()`
Call the REST endpoint to see if the packages on the "server" are up to date.

21.13.6 salt.proxy.esxi

Proxy Minion interface module for managing VMware ESXi hosts.

New in version 2015.8.4.

Special Note: SaltStack thanks Adobe Corporation **for their support in creating this Proxy Minion integration.**

This proxy minion enables VMware ESXi (hereafter referred to as simply 'ESXi') hosts to be treated individually like a Salt Minion.

Since the ESXi host may not necessarily run on an OS capable of hosting a Python stack, the ESXi host can't run a Salt Minion directly. Salt's "Proxy Minion" functionality enables you to designate another machine to host a minion process that "proxies" communication from the Salt Master. The master does not know nor care that the target is not a "real" Salt Minion.

More in-depth conceptual reading on Proxy Minions can be found in the *Proxy Minion* section of Salt's documentation.

Dependencies

- pyVmomi Python Module
- ESXCLI

pyVmomi

PyVmomi can be installed via pip:

```
pip install pyVmomi
```

Note: Version 6.0 of pyVmomi has some problems with SSL error handling on certain versions of Python. If using version 6.0 of pyVmomi, Python 2.6, Python 2.7.9, or newer must be present. This is due to an upstream dependency in pyVmomi 6.0 that is not supported in Python versions 2.7 to 2.7.8. If the version of Python is not in the supported range, you will need to install an earlier version of pyVmomi. See Issue #29537 for more information.

Based on the note above, to install an earlier version of pyVmomi than the version currently listed in PyPi, run the following:

```
pip install pyVmomi==5.5.0.2014.1.1
```

The 5.5.0.2014.1.1 is a known stable version that this original ESXi State Module was developed against.

ESXCLI

Currently, about a third of the functions used in the vSphere Execution Module require the ESXCLI package be installed on the machine running the Proxy Minion process.

The ESXCLI package is also referred to as the VMware vSphere CLI, or vCLI. VMware provides vCLI package installation instructions for vSphere 5.5 and vSphere 6.0.

Once all of the required dependencies are in place and the vCLI package is installed, you can check to see if you can connect to your ESXi host or vCenter server by running the following command:

```
esxcli -s <host-location> -u <username> -p <password> system syslog config get
```

If the connection was successful, ESXCLI was successfully installed on your system. You should see output related to the ESXi host's syslog configuration.

Configuration

To use this integration proxy module, please configure the following:

Pillar

Proxy minions get their configuration from Salt's Pillar. Every proxy must have a stanza in Pillar and a reference in the Pillar top-file that matches the ID. At a minimum for communication with the ESXi host, the pillar should look like this:

```
proxy:
  proxytype: esxi
  host: <ip or dns name of esxi host>
  username: <ESXi username>
  passwords:
    - first_password
    - second_password
    - third_password
  credstore: <path to credential store>
```

proxytype

The `proxytype` key and value pair is critical, as it tells Salt which interface to load from the `proxy` directory in Salt's install hierarchy, or from `/srv/salt/_proxy` on the Salt Master (if you have created your own proxy module, for example). To use this ESXi Proxy Module, set this to `esxi`.

host

The location, or ip/dns, of the ESXi host. Required.

username

The username used to login to the ESXi host, such as `root`. Required.

passwords

A list of passwords to be used to try and login to the ESXi host. At least one password in this list is required.

The proxy integration will try the passwords listed in order. It is configured this way so you can have a regular password and the password you may be updating for an ESXi host either via the *`vsphere.update_host_password`* execution module function or via the *`esxi.password_present`* state function. This way, after the password is changed, you should not need to restart the proxy minion–it should just pick up the the new password provided in the list. You can then change pillar at will to move that password to the front and retire the unused ones.

This also allows you to use any number of potential fallback passwords.

Note: When a password is changed on the host to one in the list of possible passwords, the further down on the list the password is, the longer individual commands will take to return. This is due to the nature of pyVmomi's login system. We have to wait for the first attempt to fail before trying the next password on the list.

This scenario is especially true, and even slower, when the proxy minion first starts. If the correct password is not the first password on the list, it may take up to a minute for `test.ping` to respond with a `True` result. Once the initial authorization is complete, the responses for commands will be a little faster.

To avoid these longer waiting periods, SaltStack recommends moving the correct password to the top of the list and restarting the proxy minion at your earliest convenience.

protocol

If the ESXi host is not using the default protocol, set this value to an alternate protocol. Default is `https`.

port

If the ESXi host is not using the default port, set this value to an alternate port. Default is `443`.

credstore

If the ESXi host is using an untrusted SSL certificate, set this value to the file path where the credential store is located. This file is passed to `esxcli`. Default is `<HOME>/.vmware/credstore/vicredentials.xml` on Linux and `<APPDATA>/VMware/credstore/vicredentials.xml` on Windows.

Note: `HOME` variable is sometimes not set for processes running as system services. If you want to rely on the default credential store location, make sure `HOME` is set for the proxy process.

Salt Proxy

After your pillar is in place, you can test the proxy. The proxy can run on any machine that has network connectivity to your Salt Master and to the ESXi host in question. SaltStack recommends that the machine running the salt-proxy process also run a regular minion, though it is not strictly necessary.

On the machine that will run the proxy, make sure there is an `/etc/salt/proxy` file with at least the following in it:

```
master: <ip or hostname of salt-master>
```

You can then start the salt-proxy process with:

```
salt-proxy --proxyid <id you want to give the host>
```

You may want to add `-l debug` to run the above in the foreground in debug mode just to make sure everything is OK.

Next, accept the key for the proxy on your salt-master, just like you would for a regular minion:

```
salt-key -a <id you gave the esxi host>
```

You can confirm that the pillar data is in place for the proxy:

```
salt <id> pillar.items
```

And now you should be able to ping the ESXi host to make sure it is responding:

```
salt <id> test.ping
```

At this point you can execute one-off commands against the host. For example, you can get the ESXi host's system information:

```
salt <id> esxi.cmd system_info
```

Note that you don't need to provide credentials or an ip/hostname. Salt knows to use the credentials you stored in Pillar.

It's important to understand how this particular proxy works. *`Salt.modules.vsphere`* is a standard Salt execution module. If you pull up the docs for it you'll see that almost every function in the module takes credentials and a target host. When credentials and a host aren't passed, Salt runs commands through `pyVmomi` against the local machine. If you wanted, you could run functions from this module on any host where an appropriate version of `pyVmomi` is installed, and that host would reach out over the network and communicate with the ESXi host.

`esxi.cmd` acts as a "shim" between the execution module and the proxy. Its first parameter is always the function from salt.modules.vsphere. If the function takes more positional or keyword arguments you can append them to the call. It's this shim that speaks to the ESXi host through the proxy, arranging for the credentials and hostname to be pulled from the Pillar section for this Proxy Minion.

Because of the presence of the shim, to lookup documentation for what functions you can use to interface with the ESXi host, you'll want to look in *`salt.modules.vsphere`* instead of *`salt.modules.esxi`*.

States

Associated states are thoroughly documented in *`salt.states.esxi`*. Look there to find an example structure for Pillar as well as an example `.sls` file for standing up an ESXi host from scratch.

`salt.proxy.esxi.`**`ch_config`**`(`*cmd*, **args*, ***kwargs*`)`
This function is called by the *`salt.modules.esxi.cmd`* shim. It then calls whatever is passed in `cmd` inside the *`salt.modules.vsphere`* module. Passes the return through from the vsphere module.
cmd The command to call inside salt.modules.vsphere
args Arguments that need to be passed to that command.
kwargs Keyword arguments that need to be passed to that command.

`salt.proxy.esxi.`**`find_credentials`**`(`*host*`)`
Cycle through all the possible credentials and return the first one that works.

`salt.proxy.esxi.`**`get_details`**`()`
Return the proxy details

`salt.proxy.esxi.`**`grains`**`()`
Get the grains from the proxy device.

`salt.proxy.esxi.`**`grains_refresh`**`()`
Refresh the grains from the proxy device.

`salt.proxy.esxi.`**`init`**`(`*opts*`)`
This function gets called when the proxy starts up. For ESXi devices, the host, login credentials, and, if configured, the protocol and port are cached.

`salt.proxy.esxi.`**`is_connected_via_vcenter`**`()`

`salt.proxy.esxi.`**`ping`**`()`
Returns True if connection is to be done via a vCenter (no connection is attempted). Check to see if the host is responding when connecting directly via an ESXi host.

CLI Example:

```
salt esxi-host test.ping
```

`salt.proxy.esxi.`**`shutdown`**`()`

Shutdown the connection to the proxy device. For this proxy, shutdown is a no-op.

21.13.7 salt.proxy.fx2

Dell FX2 chassis

New in version 2015.8.2.

Proxy minion interface module for managing Dell FX2 chassis (Dell Chassis Management Controller version 1.2 and above, iDRAC8 version 2.00 and above)

Dependencies

- *iDRAC Remote execution module (salt.modules.dracr)*
- *Chassis command shim (salt.modules.chassis)*
- *Dell Chassis States (salt.states.dellchassis)*
- Dell's `racadm` command line interface to CMC and iDRAC devices.

Special Note: SaltStack thanks Adobe Corporation **for their support in creating this proxy minion integration.**

This proxy minion enables Dell FX2 and FX2s (hereafter referred to as simply "chassis", "CMC", or "FX2") chassis to be treated individually like a salt-minion.

Since the CMC embedded in the chassis does not run an OS capable of hosting a Python stack, the chassis can't run a minion directly. Salt's "Proxy Minion" functionality enables you to designate another machine to host a minion process that "proxies" communication from the salt-master. The master does not know nor care that the target is not a real minion.

More in-depth conceptual reading on Proxy Minions can be found *in the Proxy Minion section* of Salt's documentation.

To configure this integration, follow these steps:

Pillar

Proxy minions get their configuration from Salt's Pillar. Every proxy must have a stanza in Pillar, and a reference in the Pillar topfile that matches the ID. At a minimum for communication with the chassis the pillar should look like this:

```
proxy:
  host: <ip or dns name of chassis controller>
  admin_username: <iDRAC username for the CMC, usually 'root'>
  fallback_admin_username: <username to try if the first fails>
  passwords:
    - first_password
    - second_password
    - third-password
  proxytype: fx2
```

The `proxytype` line above is critical, it tells Salt which interface to load from the `proxy` directory in Salt's install hierarchy, or from `/srv/salt/_proxy` on the salt-master (if you have created your own proxy module, for example).

The proxy integration will try the passwords listed in order. It is configured this way so you can have a regular password, a potential fallback password, and the third password can be the one you intend to change the chassis to use. This way, after it is changed, you should not need to restart the proxy minion–it should just pick up the third password in the list. You can then change pillar at will to move that password to the front and retire the unused ones.

Beware, many Dell CMC and iDRAC units are configured to lockout IP addresses or users after too many failed password attempts. This can generate user panic in the form of "I no longer know what the password is!!!". To mitigate panic try the web interface from a different IP, or setup a emergency administrator user in the CMC before doing a wholesale password rotation.

The automatic lockout can be disabled via Salt with the following:

```
salt <cmc> chassis.cmd set_general cfgRacTuning cfgRacTuneIpBlkEnable 0
```

and then verified with

```
salt <cmc> chassis.cmd get_general cfgRacTuning cfgRacTuneIpBlkEnable
```

salt-proxy

After your pillar is in place, you can test the proxy. The proxy can run on any machine that has network connectivity to your salt-master and to the chassis in question. SaltStack recommends that this machine also run a regular minion, though it is not strictly necessary.

On the machine that will run the proxy, make sure there is an `/etc/salt/proxy` file with at least the following in it:

```
master: <ip or hostname of salt-master>
```

You can start the proxy with

```
salt-proxy --proxyid <id you want to give the chassis>
```

You may want to add `-l debug` to run the above in the foreground in debug mode just to make sure everything is OK.

Next, accept the key for the proxy on your salt-master, just like you would for a regular minion:

```
salt-key -a <id you want to give the chassis>
```

You can confirm that the pillar data is in place for the proxy:

```
salt <id> pillar.items
```

And now you should be able to ping the chassis to make sure it is responding:

```
salt <id> test.ping
```

At this point you can execute one-off commands against the chassis. For example, you can get the chassis inventory:

```
salt <id> chassis.cmd inventory
```

Note that you don't need to provide credentials or an ip/hostname. Salt knows to use the credentials you stored in Pillar.

It's important to understand how this particular proxy works. *`Salt.modules.dracr`* is a standard Salt execution module. If you pull up the docs for it you'll see that almost every function in the module takes credentials and a target host. When credentials and a host aren't passed, Salt runs `racadm` against the local machine. If you wanted you could run functions from this module on any host where an appropriate version of `racadm` is installed, and that host would reach out over the network and communicate with the chassis.

`Chassis.cmd` acts as a "shim" between the execution module and the proxy. It's first parameter is always the function from salt.modules.dracr to execute. If the function takes more positional or keyword arguments you can append them to the call. It's this shim that speaks to the chassis through the proxy, arranging for the credentials and hostname to be pulled from the pillar section for this proxy minion.

Because of the presence of the shim, to lookup documentation for what functions you can use to interface with the chassis, you'll want to look in *`salt.modules.dracr`* instead of *`salt.modules.chassis`*.

States

Associated states are thoroughly documented in *`salt.states.dellchassis`*. Look there to find an example structure for pillar as well as an example `.sls` file for standing up a Dell Chassis from scratch.

`salt.proxy.fx2.`**`admin_password`**`()`

Return the admin_password in the DETAILS dictionary, or 'calvin' (the Dell default) if there is none present

`salt.proxy.fx2.`**`admin_username`**`()`

Return the admin_username in the DETAILS dictionary, or root if there is none present

`salt.proxy.fx2.`**`chconfig`**`(`*cmd*, **args*, ***kwargs*`)`

This function is called by the *`salt.modules.chassis.cmd`* shim. It then calls whatever is passed in `cmd` inside the *`salt.modules.dracr`* module.

Parameters
- **`cmd`** – The command to call inside salt.modules.dracr
- **`args`** – Arguments that need to be passed to that command
- **`kwargs`** – Keyword arguments that need to be passed to that command

Returns Passthrough the return from the dracr module.

`salt.proxy.fx2.`**`find_credentials`**`()`

Cycle through all the possible credentials and return the first one that works

`salt.proxy.fx2.`**`grains`**`()`

Get the grains from the proxied device

`salt.proxy.fx2.`**`grains_refresh`**`()`

Refresh the grains from the proxied device

`salt.proxy.fx2.`**`host`**`()`

`salt.proxy.fx2.`**`init`**`(`*opts*`)`

This function gets called when the proxy starts up. We check opts to see if a fallback user and password are supplied. If they are present, and the primary credentials don't work, then we try the backup before failing.

Whichever set of credentials works is placed in the persistent DETAILS dictionary and will be used for further communication with the chassis.

`salt.proxy.fx2.`**`ping`**`()`

Is the chassis responding?

Returns Returns False if the chassis didn't respond, True otherwise.

`salt.proxy.fx2.`**`shutdown`**`(`*opts*`)`
Shutdown the connection to the proxied device. For this proxy shutdown is a no-op.

21.13.8 salt.proxy.junos

Interface with a Junos device via proxy-minion. To connect to a junos device via junos proxy, specify the host information in the pillar in '/srv/pillar/details.sls'

```
proxy:
  proxytype: junos
  host: <ip or dns name of host>
  username: <username>
  port: 830
  password: <secret>
```

In '/srv/pillar/top.sls' map the device details with the proxy name.

```
base:
  'vmx':
    - details
```

After storing the device information in the pillar, configure the proxy in '/etc/salt/proxy'

```
master: <ip or hostname of salt-master>
```

Run the salt proxy via the following command:

```
salt-proxy --proxyid=vmx
```

`salt.proxy.junos.`**`alive`**`(`*opts*`)`
Validate and return the connection status with the remote device.

New in version 2018.3.0.

`salt.proxy.junos.`**`conn`**`()`

`salt.proxy.junos.`**`get_serialized_facts`**`()`

`salt.proxy.junos.`**`init`**`(`*opts*`)`
Open the connection to the Junos device, login, and bind to the Resource class

`salt.proxy.junos.`**`initialized`**`()`

`salt.proxy.junos.`**`ping`**`()`
Ping? Pong!

`salt.proxy.junos.`**`proxytype`**`()`
Returns the name of this proxy

`salt.proxy.junos.`**`shutdown`**`(`*opts*`)`
This is called when the proxy-minion is exiting to make sure the connection to the device is closed cleanly.

21.13.9 salt.proxy.marathon module

Marathon

Proxy minion for managing a Marathon cluster.

Dependencies

- *marathon execution module (salt.modules.marathon)*

Pillar

The marathon proxy configuration requires a 'base_url' property that points to the marathon endpoint:

```
proxy:
  proxytype: marathon
  base_url: http://my-marathon-master.mydomain.com:8080
```

New in version 2015.8.2.

`salt.proxy.marathon.`**`init`**`(`*opts*`)`
Perform any needed setup.

`salt.proxy.marathon.`**`ping`**`()`
Is the marathon api responding?

`salt.proxy.marathon.`**`shutdown`**`(`*opts*`)`
For this proxy shutdown is a no-op

21.13.10 salt.proxy.napalm

NAPALM: Network Automation and Programmability Abstraction Layer with Multivendor support

New in version 2016.11.0.

Proxy minion for managing network devices via NAPALM library.

codeauthor Mircea Ulinic <ping@mirceaulinic.net> & Jerome Fleury <jf@cloudflare.com>

maturity new

depends napalm

platform unix

Dependencies

The `napalm` proxy module requires NAPALM library to be installed: `pip install napalm` Please check Installation for complete details.

Note: Beginning with Salt release 2017.7.3, it is recommended to use `napalm >= 2.0.0`. The library has been unified into a monolithic package, as in opposite to separate packages per driver. For more details you can check this document. While it will still work with the old packages, bear in mind that the NAPALM core team will maintain only the main `napalm` package.

Moreover, for additional capabilities, the users can always define a library that extends NAPALM's base capabilities and configure the `provider` option (see below).

Pillar

The napalm proxy configuration requires the following parameters in order to connect to the network device:

driver Specifies the network device operating system. For a complete list of the supported operating systems please refer to the NAPALM Read the Docs page.

host The IP Address or FQDN to use when connecting to the device. Alternatively, the following field names can be used instead: `hostname`, `fqdn`, `ip`.

username The username to be used when connecting to the device.

passwd The password needed to establish the connection.

> **Note:** This field may not be mandatory when working with SSH-based drivers, and the username has a SSH key properly configured on the device targeted to be managed.

optional_args Dictionary with the optional arguments. Check the complete list of supported optional arguments.

always_alive: **`True`** In certain less dynamic environments, maintaining the remote connection permanently open with the network device is not always beneficial. In that case, the user can select to initialize the connection only when needed, by specifying this field to `false`. Default: `true` (maintains the connection with the remote network device).

New in version 2017.7.0.

provider: **`napalm_base`** The library that provides the `get_network_device` function. This option is useful when the user has more specific needs and requires to extend the NAPALM capabilities using a private library implementation. The only constraint is that the alternative library needs to have the `get_network_device` function available.

New in version 2017.7.1.

multiprocessing: **`False`** Overrides the *`multiprocessing`* option, per proxy minion. The `multiprocessing` option must be turned off for SSH-based proxies. However, some NAPALM drivers (e.g. Arista, NX-OS) are not SSH-based. As multiple proxy minions may share the same configuration file, this option permits the configuration of the `multiprocessing` option more specifically, for some proxy minions.

New in version 2017.7.2.

Proxy pillar file example:

```yaml
proxy:
  proxytype: napalm
  driver: junos
  host: core05.nrt02
  username: my_username
  passwd: my_password
  optional_args:
    port: 12201
```

Example using a user-specific library, extending NAPALM's capabilities, e.g. `custom_napalm_base`:

```yaml
proxy:
  proxytype: napalm
  driver: ios
  fqdn: cr1.th2.par.as1234.net
  username: salt
```

(continues on next page)

(continued from previous page)

```
  password: ''
  provider: custom_napalm_base
```

See also:

- *NAPALM grains: select network devices based on their characteristics*
- *NET module: network basic features*
- *Network config state: Manage the configuration using arbitrary templates*
- NAPALM YANG state: Manage the configuration according to the YANG models (OpenConfig/IETF)
- Network ACL module: Generate and load ACL (firewall) configuration
- *Network ACL state: Manage the firewall configuration*
- *NTP operational and configuration management module*
- *BGP operational and configuration management module*
- *Routes details*
- *SNMP configuration module*
- *Users configuration management*

Note: Beginning with release codename 2019.2.0, any NAPALM command executed when running under a NAPALM Proxy Minion supports the `force_reconnect` magic argument.

Proxy Minions generally establish a connection with the remote network device at the time of the Minion startup and that connection is going to be used forever.

If one would need execute a command on the device but connecting using different parameters (due to various causes, e.g., unable to authenticate the user specified in the Pillar as the authentication system - say TACACS+ is not available, or the DNS resolver is currently down and would like to temporarily use the IP address instead, etc.), it implies updating the Pillar data and restarting the Proxy Minion process restart. In particular cases like that, you can pass the `force_reconnect=True` keyword argument, together with the alternative connection details, to enforce the command to be executed over a separate connection.

For example, if the usual command is `salt '*' net.arp`, you can use the following to connect using a different username instead: `salt '*' net.arp username=my-alt-usr force_reconnect=True`.

`salt.proxy.napalm.`**`alive`**`(`*opts*`)`

Return the connection status with the remote device.

New in version 2017.7.0.

`salt.proxy.napalm.`**`call`**`(`*method, *args, **kwargs*`)`

Calls a specific method from the network driver instance. Please check the readthedocs page for the updated list of getters.

Parameters

- **`method`** – specifies the name of the method to be called
- **`params`** – contains the mapping between the name and the values of the parameters needed to call the method

Returns A dictionary with three keys:

- result (True/False): if the operation succeeded
- out (object): returns the object as-is from the call

- comment (string): provides more details in case the call failed
- traceback (string): complete traceback in case of exception. Please submit an issue including this traceback on the correct driver repo and make sure to read the FAQ

Example:

```
__proxy__['napalm.call']('cli'
                         **{
                            'commands': [
                                'show version',
                                'show chassis fan'
                            ]
                         })
```

`salt.proxy.napalm.`**`fns`**`()`
Method called by NAPALM grains module.

`salt.proxy.napalm.`**`get_device`**`()`
Returns the network device object.

`salt.proxy.napalm.`**`get_grains`**`()`
Retrieve facts from the network device.

`salt.proxy.napalm.`**`grains_refresh`**`()`
Refresh the grains.

`salt.proxy.napalm.`**`init`**`(`*opts*`)`
Opens the connection with the network device.

`salt.proxy.napalm.`**`initialized`**`()`
Connection finished initializing?

`salt.proxy.napalm.`**`ping`**`()`
Connection open successfully?

`salt.proxy.napalm.`**`shutdown`**`(`*opts*`)`
Closes connection with the device.

21.13.11 salt.proxy.netmiko_px

Netmiko

New in version 2019.2.0.

Proxy module for managing network devices via Netmiko.

codeauthor Mircea Ulinic <ping@mirceaulinic.net> & Kirk Byers <ktbyers@twb-tech.com>

maturity new

depends netmiko

platform unix

Dependencies

The `netmiko` proxy modules requires Netmiko to be installed: `pip install netmiko`.

Pillar

The `netmiko` proxy configuration requires the following parameters in order to connect to the network device:

- `device_type` - Class selection based on device type. Supported options:
    - `a10`: A10 Networks
    - `accedian`: Accedian Networks
    - `alcatel_aos`: Alcatel AOS
    - `alcatel_sros`: Alcatel SROS
    - `apresia_aeos`: Apresia AEOS
    - `arista_eos`: Arista EOS
    - `aruba_os`: Aruba
    - `avaya_ers`: Avaya ERS
    - `avaya_vsp`: Avaya VSP
    - `brocade_fastiron`: Brocade Fastiron
    - `brocade_netiron`: Brocade Netiron
    - `brocade_nos`: Brocade NOS
    - `brocade_vdx`: Brocade NOS
    - `brocade_vyos`: VyOS
    - `checkpoint_gaia`: Check Point GAiA
    - `calix_b6`: Calix B6
    - `ciena_saos`: Ciena SAOS
    - `cisco_asa`: Cisco SA
    - `cisco_ios`: Cisco IOS
    - `cisco_nxos`: Cisco NX-oS
    - `cisco_s300`: Cisco S300
    - `cisco_tp`: Cisco TpTcCe
    - `cisco_wlc`: Cisco WLC
    - `cisco_xe`: Cisco IOS
    - `cisco_xr`: Cisco XR
    - `coriant`: Coriant
    - `dell_force10`: Dell Force10
    - `dell_os10`: Dell OS10
    - `dell_powerconnect`: Dell PowerConnect
    - `eltex`: Eltex
    - `enterasys`: Enterasys
    - `extreme`: Extreme
    - `extreme_wing`: Extreme Wing

  - `f5_ltm`: F5 LTM
  - `fortinet`: Fortinet
  - `generic_termserver`: TerminalServer
  - `hp_comware`: HP Comware
  - `hp_procurve`: HP Procurve
  - `huawei`: Huawei
  - `huawei_vrpv8`: Huawei VRPV8
  - `juniper`: Juniper Junos
  - `juniper_junos`: Juniper Junos
  - `linux`: Linux
  - `mellanox`: Mellanox
  - `mrv_optiswitch`: MrvOptiswitch
  - `netapp_cdot`: NetAppcDot
  - `netscaler`: Netscaler
  - `ovs_linux`: OvsLinux
  - `paloalto_panos`: PaloAlto Panos
  - `pluribus`: Pluribus
  - `quanta_mesh`: Quanta Mesh
  - `ruckus_fastiron`: Ruckus Fastiron
  - `ubiquiti_edge`: Ubiquiti Edge
  - `ubiquiti_edgeswitch`: Ubiquiti Edge
  - `vyatta_vyos`: VyOS
  - `vyos`: VyOS
  - `brocade_fastiron_telnet`: Brocade Fastiron over Telnet
  - `brocade_netiron_telnet`: Brocade Netiron over Telnet
  - `cisco_ios_telnet`: Cisco IOS over Telnet
  - `apresia_aeos_telnet`: Apresia AEOS over Telnet
  - `arista_eos_telnet`: Arista EOS over Telnet
  - `hp_procurve_telnet`: HP Procurve over Telnet
  - `hp_comware_telnet`: HP Comware over Telnet
  - `juniper_junos_telnet`: Juniper Junos over Telnet
  - `calix_b6_telnet`: Calix B6 over Telnet
  - `dell_powerconnect_telnet`: Dell PowerConnect over Telnet
  - `generic_termserver_telnet`: TerminalServer over Telnet
  - `extreme_telnet`: Extreme Networks over Telnet
  - `ruckus_fastiron_telnet`: Ruckus Fastiron over Telnet

  - `cisco_ios_serial`: Cisco IOS over serial port
- `ip` - IP address of target device (not required if `host` is provided)
- `host` - Hostname of target device (not required if `ip` is provided)
- `username` - Username to authenticate against target device, if required
- `password` - Password to authenticate against target device, if required
- `secret` - The enable password if target device requires one
- `port` - The destination port used to connect to the target device
- `global_delay_factor` - Multiplication factor affecting Netmiko delays (default: `1`)
- `use_keys` - Connect to target device using SSH keys (default: `False`)
- `key_file` - Filename path of the SSH key file to use
- `allow_agent` - Enable use of SSH key-agent
- `ssh_strict` - Automatically reject unknown SSH host keys (default: `False`, which means unknown SSH host keys will be accepted)
- `system_host_keys` - Load host keys from the user's "known_hosts" file (default: `False`)
- `alt_host_keys` - If `True`, host keys will be loaded from the file specified in `alt_key_file` (default: `False`)
- `alt_key_file` - SSH host key file to use (if `alt_host_keys=True`)
- `ssh_config_file` - File name of OpenSSH configuration file
- `timeout` - Connection timeout, in seconds (default: `90`)
- `session_timeout` - Set a timeout for parallel requests, in seconds (default: `60`)
- `keepalive` - Send SSH keepalive packets at a specific interval, in seconds. Currently defaults to `0`, for backwards compatibility (it will not attempt to keep the connection alive using the KEEPALIVE packets).
- `default_enter` - Character(s) to send to correspond to enter key (default: `\n`)
- `response_return` - Character(s) to use in normalized return data to represent enter key (default: `\n`)
- `always_alive` - In certain less dynamic environments, maintaining the remote connection permanently open with the network device is not always beneficial. In that case, the user can select to initialize the connection only when needed, by setting this option to `False`. By default this option is set to `True` (maintains the connection with the remote network device)
- `multiprocessing` - Overrides the *`multiprocessing`* option, per proxy minion, as the Netmiko communication channel is mainly SSH (default: `False`)

Proxy Pillar Example

```
proxy:
  proxytype: netmiko
  device_type: juniper_junos
  host: router1.example.com
  username: example
  password: example
```

```
proxy:
  proxytype: netmiko
  device_type: cisco_ios
  ip: 1.2.3.4
  username: test
  use_keys: true
  secret: w3@k
```

`salt.proxy.netmiko_px.`**`alive`**`(`*opts*`)`
Return the connection status with the network device.

`salt.proxy.netmiko_px.`**`args`**`()`
Return the Netmiko device args.

`salt.proxy.netmiko_px.`**`call`**`(`*method*, **args*, ***kwargs*`)`
Calls an arbitrary netmiko method.

`salt.proxy.netmiko_px.`**`conn`**`()`
Return the connection object.

`salt.proxy.netmiko_px.`**`init`**`(`*opts*`)`
Open the connection to the network device managed through netmiko.

`salt.proxy.netmiko_px.`**`initialized`**`()`
Connection finished initializing?

`salt.proxy.netmiko_px.`**`ping`**`()`
Connection open successfully?

`salt.proxy.netmiko_px.`**`shutdown`**`(`*opts*`)`
Closes connection with the device.

21.13.12 salt.proxy.nxos module

Proxy Minion for Cisco NX OS Switches

The Cisco NX OS Proxy Minion uses the built in SSHConnection module in `salt.utils.vt_helper`

To configure the proxy minion:

```
proxy:
  proxytype: nxos
  host: 192.168.187.100
  username: admin
  password: admin
  prompt_name: switch
  ssh_args: '-o PubkeyAuthentication=no'
  key_accept: True
```

proxytype (REQUIRED) Use this proxy minion *nxos*

host (REQUIRED) ip address or hostname to connect to

username (REQUIRED) username to login with

password (REQUIRED) password to use to login with

prompt_name (REQUIRED, this or *prompt_regex* below, but not both) The name in the prompt on the switch. Recommended to use your device's hostname.

prompt_regex (REQUIRED, this or *prompt_name* above, but not both) A regular expression that matches the prompt on the switch and any other possible prompt at which you need the proxy minion to continue sending input. This feature was specifically developed for situations where the switch may ask for confirmation. *prompt_name* above would not match these, and so the session would timeout.

Example:

```
dc01-switch-01#.*|\(y\/n\)\?.*
```

This should match

```
dc01-switch-01#
```

or

```
Flash complete.  Reboot this switch (y/n)? [n]
```

If neither *prompt_name* nor *prompt_regex* is specified the prompt will be defaulted to

```
.+#$
```

which should match any number of characters followed by a # at the end of the line. This may be far too liberal for most installations.

ssh_args Any extra args to use to connect to the switch.

key_accept Whether or not to accept a the host key of the switch on initial login. Defaults to False.

The functions from the proxy minion can be run from the salt commandline using the `salt.modules.nxos` execution module.

`salt.proxy.nxos.`**`add_config`**(*lines*)

Add one or more config lines to the switch running config

```
salt '*' nxos.cmd add_config 'snmp-server community TESTSTRINGHERE group network-
↪operator'
```

Note: For more than one config added per command, lines should be a list.

`salt.proxy.nxos.`**`check_password`**(*username, password, encrypted=False*)

Check if passed password is the one assigned to user

```
salt '*' nxos.cmd check_password username=admin password=admin
salt '*' nxos.cmd check_password username=admin \
    password='$5$2fWwO2vK$s7.Hr3YltMNHuhywQQ3nfOd.gAPHgs3SOBYYdGT3E.A' \
    encrypted=True
```

`salt.proxy.nxos.`**`check_role`**(*username, role*)

Check if user is assigned a specific role on switch

```
salt '*' nxos.cmd check_role username=admin role=network-admin
```

`salt.proxy.nxos.`**`delete_config`**(*lines*)

Delete one or more config lines to the switch running config

```
salt '*' nxos.cmd delete_config 'snmp-server community TESTSTRINGHERE group
↪network-operator'
```

Note: For more than one config deleted per command, lines should be a list.

`salt.proxy.nxos.`**`find`**`(`*pattern*`)`

Find all instances where the pattern is in the running command

```
salt '*' nxos.cmd find '^snmp-server.*$'
```

Note: This uses the *re.MULTILINE* regex format for python, and runs the regex against the whole show_run output.

`salt.proxy.nxos.`**`get_roles`**`(`*username*`)`

Get roles that the username is assigned from switch

```
salt '*' nxos.cmd get_roles username=admin
```

`salt.proxy.nxos.`**`get_user`**`(`*username*`)`

Get username line from switch

```
salt '*' nxos.cmd get_user username=admin
```

`salt.proxy.nxos.`**`grains`**`()`

Get grains for proxy minion

```
salt '*' nxos.cmd grains
```

`salt.proxy.nxos.`**`grains_refresh`**`()`

Refresh the grains from the proxy device.

```
salt '*' nxos.cmd grains_refresh
```

`salt.proxy.nxos.`**`init`**`(`*opts=None*`)`

Required. Can be used to initialize the server connection.

`salt.proxy.nxos.`**`initialized`**`()`

`salt.proxy.nxos.`**`ping`**`()`

Ping the device on the other end of the connection

```
salt '*' nxos.cmd ping
```

`salt.proxy.nxos.`**`remove_user`**`(`*username*`)`

Remove user from switch

```
salt '*' nxos.cmd remove_user username=daniel
```

`salt.proxy.nxos.`**`replace`**`(`*old_value, new_value, full_match=False*`)`

Replace string or full line matches in switch's running config

If full_match is set to True, then the whole line will need to be matched as part of the old value.

```
salt '*' nxos.cmd replace 'TESTSTRINGHERE' 'NEWTESTSTRINGHERE'
```

`salt.proxy.nxos.`**`sendline`**`(`*command*`)`

Run command through switch's cli

```
salt '*' nxos.cmd sendline 'show run | include "^username admin password"'
```

`salt.proxy.nxos.`**`set_password`**(*username, password, encrypted=False, role=None, crypt_salt=None, algorithm='sha256'*)

Set users password on switch

```
salt '*' nxos.cmd set_password admin TestPass
salt '*' nxos.cmd set_password admin \
    password='$5$2fWwO2vK$s7.Hr3YltMNHuhywQQ3nfOd.gAPHgs3SOBYYdGT3E.A' \
    encrypted=True
```

`salt.proxy.nxos.`**`set_role`**(*username, role*)

Assign role to username

```
salt '*' nxos.cmd set_role username=daniel role=vdc-admin
```

`salt.proxy.nxos.`**`show_run`**()

Shortcut to run *show run* on switch

```
salt '*' nxos.cmd show_run
```

`salt.proxy.nxos.`**`show_ver`**()

Shortcut to run *show ver* on switch

```
salt '*' nxos.cmd show_ver
```

`salt.proxy.nxos.`**`shutdown`**(*opts*)

Disconnect

`salt.proxy.nxos.`**`system_info`**()

Return system information for grains of the NX OS proxy minion

```
salt '*' nxos.system_info
```

`salt.proxy.nxos.`**`unset_role`**(*username, role*)

Remove role from username

```
salt '*' nxos.cmd unset_role username=daniel role=vdc-admin
```

21.13.13 salt.proxy.nxos_api module

Proxy Minion to manage Cisco Nexus Switches (NX-OS) over the NX-API

New in version 2019.2.0.

Proxy module for managing Cisco Nexus switches via the NX-API.

codeauthor Mircea Ulinic <ping@mirceaulinic.net>

maturity new

platform any

Usage

Note: To be able to use this module you need to enable to NX-API on your switch, by executing `feature nxapi` in configuration mode.

Configuration example:

```
switch# conf t
switch(config)# feature nxapi
```

To check that NX-API is properly enabled, execute `show nxapi`.

Output example:

```
switch# show nxapi
nxapi enabled
HTTPS Listen on port 443
```

Note: NX-API requires modern NXOS distributions, typically at least 7.0 depending on the hardware. Due to reliability reasons it is recommended to run the most recent version.

Check https://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus7000/sw/programmability/guide/b_Cisco_Nexus_7000_Series_NX-OS_Programmability_Guide/b_Cisco_Nexus_7000_Series_NX-OS_Programmability_Guide_chapter_0101.html for more details.

Pillar

The `nxos_api` proxy configuration requires the following parameters in order to connect to the network switch:

transport: `https` Specifies the type of connection transport to use. Valid values for the connection are `http`, and `https`.

host: `localhost` The IP address or DNS host name of the connection device.

username: `admin` The username to pass to the device to authenticate the NX-API connection.

password The password to pass to the device to authenticate the NX-API connection.

port The TCP port of the endpoint for the NX-API connection. If this keyword is not specified, the default value is automatically determined by the transport type (`80` for `http`, or `443` for `https`).

timeout: `60` Time in seconds to wait for the device to respond. Default: 60 seconds.

verify: `True` Either a boolean, in which case it controls whether we verify the NX-API TLS certificate, or a string, in which case it must be a path to a CA bundle to use. Defaults to `True`.

When there is no certificate configuration on the device and this option is set as `True` (default), the commands will fail with the following error: `SSLError: [SSL: CERTIFICATE_VERIFY_FAILED] certificate verify failed (_ssl.c:581)`. In this case, you either need to configure a proper certificate on the device (*recommended*), or bypass the checks setting this argument as `False` with all the security risks considered.

Check https://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus3000/sw/programmability/6_x/b_Cisco_Nexus_3000_Series_NX-OS_Programmability_Guide/b_Cisco_Nexus_3000_Series_NX-OS_Programmability_Guide_chapter_01.html to see how to properly configure the certificate.

All the arguments may be optional, depending on your setup.

Proxy Pillar Example

```
proxy:
  proxytype: nxos_api
  host: switch1.example.com
  username: example
  password: example
```

`salt.proxy.nxos_api.`**`get_conn_args`**`()`

Returns the connection arguments of the Proxy Minion.

`salt.proxy.nxos_api.`**`init`**`(`*opts*`)`

Open the connection to the Nexsu switch over the NX-API.

As the communication is HTTP based, there is no connection to maintain, however, in order to test the connectivity and make sure we are able to bring up this Minion, we are executing a very simple command (`show clock`) which doesn't come with much overhead and it's sufficient to confirm we are indeed able to connect to the NX-API endpoint as configured.

`salt.proxy.nxos_api.`**`initialized`**`()`

Connection finished initializing?

`salt.proxy.nxos_api.`**`ping`**`()`

Connection open successfully?

`salt.proxy.nxos_api.`**`rpc`**`(`*commands*, *method='cli'*, ***kwargs*`)`

Executes an RPC request over the NX-API.

`salt.proxy.nxos_api.`**`shutdown`**`(`*opts*`)`

Closes connection with the device.

21.13.14 salt.proxy.panos module

Proxy Minion interface module for managing Palo Alto firewall devices

New in version 2018.3.0.

codeauthor `Spencer Ervin <spencer_ervin@hotmail.com>`

maturity new

depends none

platform unix

This proxy minion enables Palo Alto firewalls (hereafter referred to as simply 'panos') to be treated individually like a Salt Minion.

The panos proxy leverages the XML API functionality on the Palo Alto firewall. The Salt proxy must have access to the Palo Alto firewall on HTTPS (tcp/443).

More in-depth conceptual reading on Proxy Minions can be found in the *Proxy Minion* section of Salt's documentation.

Configuration

To use this integration proxy module, please configure the following:

Pillar

Proxy minions get their configuration from Salt's Pillar. Every proxy must have a stanza in Pillar and a reference in the Pillar top-file that matches the ID. There are four connection options available for the panos proxy module.

- Direct Device (Password)
- Direct Device (API Key)
- Panorama Pass-Through (Password)
- Panorama Pass-Through (API Key)

Direct Device (Password)

The direct device configuration configures the proxy to connect directly to the device with username and password.

```
proxy:
  proxytype: panos
  host: <ip or dns name of panos host>
  username: <panos username>
  password: <panos password>
```

proxytype

The `proxytype` key and value pair is critical, as it tells Salt which interface to load from the `proxy` directory in Salt's install hierarchy, or from `/srv/salt/_proxy` on the Salt Master (if you have created your own proxy module, for example). To use this panos Proxy Module, set this to `panos`.

host

The location, or ip/dns, of the panos host. Required.

username

The username used to login to the panos host. Required.

password

The password used to login to the panos host. Required.

Direct Device (API Key)

Palo Alto devices allow for access to the XML API with a generated 'API key'_ instead of username and password.

```
proxy:
  proxytype: panos
  host: <ip or dns name of panos host>
  apikey: <panos generated api key>
```

proxytype

The `proxytype` key and value pair is critical, as it tells Salt which interface to load from the `proxy` directory in Salt's install hierarchy, or from `/srv/salt/_proxy` on the Salt Master (if you have created your own proxy module, for example). To use this panos Proxy Module, set this to `panos`.

host

The location, or ip/dns, of the panos host. Required.

apikey

The generated XML API key for the panos host. Required.

Panorama Pass-Through (Password)

The Panorama pass-through method sends all connections through the Panorama management system. It passes the connections to the appropriate device using the serial number of the Palo Alto firewall.

This option will reduce the number of connections that must be present for the proxy server. It will only require a connection to the Panorama server.

The username and password will be for authentication to the Panorama server, not the panos device.

```
proxy:
  proxytype: panos
  serial: <serial number of panos host>
  host: <ip or dns name of the panorama server>
  username: <panorama server username>
  password: <panorama server password>
```

proxytype

The `proxytype` key and value pair is critical, as it tells Salt which interface to load from the `proxy` directory in Salt's install hierarchy, or from `/srv/salt/_proxy` on the Salt Master (if you have created your own proxy module, for example). To use this panos Proxy Module, set this to `panos`.

serial

The serial number of the panos host. Required.

host

The location, or ip/dns, of the Panorama server. Required.

username

The username used to login to the Panorama server. Required.

password

The password used to login to the Panorama server. Required.

Panorama Pass-Through (API Key)

The Panorama server can also utilize a generated 'API key'_ for authentication.

```
proxy:
  proxytype: panos
  serial: <serial number of panos host>
  host: <ip or dns name of the panorama server>
  apikey: <panos generated api key>
```

proxytype

The `proxytype` key and value pair is critical, as it tells Salt which interface to load from the `proxy` directory in Salt's install hierarchy, or from `/srv/salt/_proxy` on the Salt Master (if you have created your own proxy module, for example). To use this panos Proxy Module, set this to `panos`.

serial

The serial number of the panos host. Required.

host

The location, or ip/dns, of the Panorama server. Required.

apikey

The generated XML API key for the Panorama server. Required.

`salt.proxy.panos.`**`call`**`(`*payload=None*`)`
This function captures the query string and sends it to the Palo Alto device.

`salt.proxy.panos.`**`grains`**`()`
Get the grains from the proxied device

`salt.proxy.panos.`**`grains_refresh`**`()`
Refresh the grains from the proxied device

`salt.proxy.panos.`**`init`**`(`*opts*`)`
This function gets called when the proxy starts up. For panos devices, a determination is made on the connection type and the appropriate connection details that must be cached.

`salt.proxy.panos.`**`initialized`**`()`
Since grains are loaded in many different places and some of those places occur before the proxy can be initialized, return whether our init() function has been called

`salt.proxy.panos.`**`is_required_version`**`(`*required_version='0.0.0'*`)`
Because different versions of Palo Alto support different command sets, this function will return true if the current version of Palo Alto supports the required command.

`salt.proxy.panos.`**`ping`**`()`
Returns true if the device is reachable, else false.

`salt.proxy.panos.`**`shutdown`**`()`
Shutdown the connection to the proxy device. For this proxy, shutdown is a no-op.

21.13.15 salt.proxy.philips_hue module

Philips HUE lamps module for proxy.

New in version 2015.8.3.

First create a new user on the Hue bridge by following the Meet hue instructions.

To configure the proxy minion:

```
proxy:
  proxytype: philips_hue
  host: [hostname or ip]
  user: [username]
```

`class` `salt.proxy.philips_hue.`**`Const`**
Constants for the lamp operations.

`COLOR_BLUE = {'hue': 46920, 'sat': 254}`

`COLOR_DAYLIGHT = {'xy': [0.3806, 0.3576]}`

`COLOR_GREEN = {'hue': 25500, 'sat': 254}`

`COLOR_ORANGE = {'hue': 12000, 'sat': 254}`

`COLOR_PINK = {'xy': [0.3688, 0.2095]}`

`COLOR_PURPLE = {'xy': [0.3787, 0.1724]}`

`COLOR_RED = {'hue': 0, 'sat': 254}`

`COLOR_WHITE = {'xy': [0.3227, 0.329]}`

`COLOR_YELLOW = {'xy': [0.4432, 0.5154]}`

`LAMP_OFF = {'on': False, 'transitiontime': 0}`

`LAMP_ON = {'on': True, 'transitiontime': 0}`

`salt.proxy.philips_hue.`**`call_alert`**`(`**args, **kwargs*`)`
Lamp alert

Options:
- **id**: Specifies a device ID. Can be a comma-separated values. All, if omitted.
- **on**: Turns on or off an alert. Default is True.

CLI Example:

```
salt '*' hue.alert
salt '*' hue.alert id=1
salt '*' hue.alert id=1,2,3 on=false
```

`salt.proxy.philips_hue.`**`call_blink`**`(`**args, **kwargs*`)`
Blink a lamp. If lamp is ON, then blink ON-OFF-ON, otherwise OFF-ON-OFF.

Options:
- **id**: Specifies a device ID. Can be a comma-separated values. All, if omitted.

- **pause**: Time in seconds. Can be less than 1, i.e. 0.7, 0.5 sec.

CLI Example:

```
salt '*' hue.blink id=1
salt '*' hue.blink id=1,2,3
```

`salt.proxy.philips_hue.`**`call_brightness`**`(`**args, **kwargs*`)`

Set an effect to the lamp.

Arguments:

- **value**: 0~255 brightness of the lamp.

Options:

- **id**: Specifies a device ID. Can be a comma-separated values. All, if omitted.
- **transition**: Transition 0~200. Default 0.

CLI Example:

```
salt '*' hue.brightness value=100
salt '*' hue.brightness id=1 value=150
salt '*' hue.brightness id=1,2,3 value=255
```

`salt.proxy.philips_hue.`**`call_color`**`(`**args, **kwargs*`)`

Set a color to the lamp.

Options:

- **id**: Specifies a device ID. Can be a comma-separated values. All, if omitted.
- **color**: **Fixed color. Values are**: **red**, **green**, **blue**, **orange**, **pink**, **white**, yellow, daylight, purple. Default white.
- **transition**: Transition 0~200.

Advanced:

- **gamut**: **XY coordinates. Use gamut according to the Philips HUE devices documentation**. More: http://www.developers.meethue.com/documentation/hue-xy-values

CLI Example:

```
salt '*' hue.color
salt '*' hue.color id=1
salt '*' hue.color id=1,2,3 oolor=red transition=30
salt '*' hue.color id=1 gamut=0.3,0.5
```

`salt.proxy.philips_hue.`**`call_effect`**`(`**args, **kwargs*`)`

Set an effect to the lamp.

Options:

- **id**: Specifies a device ID. Can be a comma-separated values. All, if omitted.
- **type**: Type of the effect. Possible values are ”none” or ”colorloop”. Default ”none”.

CLI Example:

```
salt '*' hue.effect
salt '*' hue.effect id=1
salt '*' hue.effect id=1,2,3 type=colorloop
```

`salt.proxy.philips_hue.`**`call_lights`**`(`**args, **kwargs*`)`

Get info about all available lamps.

Options:

- **id**: Specifies a device ID. Can be a comma-separated values. All, if omitted.

CLI Example:

```
salt '*' hue.lights
salt '*' hue.lights id=1
salt '*' hue.lights id=1,2,3
```

salt.proxy.philips_hue.**call_ping**(**args*, ***kwargs*)

Ping the lamps by issuing a short inversion blink to all available devices.

CLI Example:

```
salt '*' hue.ping
```

salt.proxy.philips_hue.**call_rename**(**args*, ***kwargs*)

Rename a device.

Options:

- **id**: Specifies a device ID. Only one device at a time.
- **title**: Title of the device.

CLI Example:

```
salt '*' hue.rename id=1 title='WC for cats'
```

salt.proxy.philips_hue.**call_status**(**args*, ***kwargs*)

Return the status of the lamps.

Options:

- **id**: Specifies a device ID. Can be a comma-separated values. All, if omitted.

CLI Example:

```
salt '*' hue.status
salt '*' hue.status id=1
salt '*' hue.status id=1,2,3
```

salt.proxy.philips_hue.**call_switch**(**args*, ***kwargs*)

Switch lamp ON/OFF.

If no particular state is passed, then lamp will be switched to the opposite state.

Options:

- **id**: Specifies a device ID. Can be a comma-separated values. All, if omitted.
- **on**: True or False. Inverted current, if omitted

CLI Example:

```
salt '*' hue.switch
salt '*' hue.switch id=1
salt '*' hue.switch id=1,2,3 on=True
```

salt.proxy.philips_hue.**call_temperature**(**args*, ***kwargs*)

Set the mired color temperature. More: http://en.wikipedia.org/wiki/Mired

Arguments:

- **value**: 150~500.

Options:

- **id**: Specifies a device ID. Can be a comma-separated values. All, if omitted.

CLI Example:

```
salt '*' hue.temperature value=150
salt '*' hue.temperature value=150 id=1
salt '*' hue.temperature value=150 id=1,2,3
```

`salt.proxy.philips_hue.`**`init`**`(`*cnf*`)`
Initialize the module.

`salt.proxy.philips_hue.`**`ping`**`(`**args*, ***kw*`)`
Ping the lamps.

`salt.proxy.philips_hue.`**`shutdown`**`(`*opts*, **args*, ***kw*`)`
Shuts down the service.

21.13.16 salt.proxy.rest_sample

This is a simple proxy-minion designed to connect to and communicate with the bottle-based web service contained in https://github.com/saltstack/salt-contrib/tree/master/proxyminion_rest_example

`salt.proxy.rest_sample.`**`alive`**`(`*opts*`)`

`salt.proxy.rest_sample.`**`fix_outage`**`()`

`salt.proxy.rest_sample.`**`fns`**`()`

`salt.proxy.rest_sample.`**`grains`**`()`
Get the grains from the proxied device

`salt.proxy.rest_sample.`**`grains_refresh`**`()`
Refresh the grains from the proxied device

`salt.proxy.rest_sample.`**`id`**`(`*opts*`)`
Return a unique ID for this proxy minion. This ID MUST NOT CHANGE. If it changes while the proxy is running the salt-master will get really confused and may stop talking to this minion

`salt.proxy.rest_sample.`**`init`**`(`*opts*`)`

`salt.proxy.rest_sample.`**`initialized`**`()`
Since grains are loaded in many different places and some of those places occur before the proxy can be initialized, return whether our init() function has been called

`salt.proxy.rest_sample.`**`package_install`**`(`*name*, ***kwargs*`)`
Install a ”package” on the REST server

`salt.proxy.rest_sample.`**`package_list`**`()`
List ”packages” installed on the REST server

`salt.proxy.rest_sample.`**`package_remove`**`(`*name*`)`
Remove a ”package” on the REST server

`salt.proxy.rest_sample.`**`package_status`**`(`*name*`)`
Check the installation status of a package on the REST server

`salt.proxy.rest_sample.`**`ping`**`()`
Is the REST server up?

`salt.proxy.rest_sample.`**`service_list`**`()`
List ”services” on the REST server

`salt.proxy.rest_sample.`**`service_restart`**`(`*name*`)`
Restart a ”service” on the REST server

`salt.proxy.rest_sample.`**`service_start`**`(`*name*`)`
Start a ”service” on the REST server

`salt.proxy.rest_sample.`**`service_status`**`(`*name*`)`
Check if a service is running on the REST server

`salt.proxy.rest_sample.`**`service_stop`**`(`*name*`)`
Stop a "service" on the REST server

`salt.proxy.rest_sample.`**`shutdown`**`(`*opts*`)`
For this proxy shutdown is a no-op

`salt.proxy.rest_sample.`**`test_from_state`**`()`
Test function so we have something to call from a state :return:

`salt.proxy.rest_sample.`**`uptodate`**`(`*name*`)`
Call the REST endpoint to see if the packages on the "server" are up to date.

21.13.17 salt.proxy.ssh_sample

This is a simple proxy-minion designed to connect to and communicate with a server that exposes functionality via SSH. This can be used as an option when the device does not provide an api over HTTP and doesn't have the python stack to run a minion.

`salt.proxy.ssh_sample.`**`fns`**`()`

`salt.proxy.ssh_sample.`**`grains`**`()`
Get the grains from the proxied device

`salt.proxy.ssh_sample.`**`grains_refresh`**`()`
Refresh the grains from the proxied device

`salt.proxy.ssh_sample.`**`init`**`(`*opts*`)`
Required. Can be used to initialize the server connection.

`salt.proxy.ssh_sample.`**`initialized`**`()`
Since grains are loaded in many different places and some of those places occur before the proxy can be initialized, return whether our init() function has been called

`salt.proxy.ssh_sample.`**`package_install`**`(`*name*, ***kwargs*`)`
Install a "package" on the ssh server

`salt.proxy.ssh_sample.`**`package_list`**`()`
List "packages" by executing a command via ssh This function is called in response to the salt command
..**code-block::bash** salt target_minion pkg.list_pkgs

`salt.proxy.ssh_sample.`**`package_remove`**`(`*name*`)`
Remove a "package" on the ssh server

`salt.proxy.ssh_sample.`**`parse`**`(`*out*`)`
Extract json from out.
Parameter out: Type string. The data returned by the ssh command.

`salt.proxy.ssh_sample.`**`ping`**`()`
Required. Ping the device on the other end of the connection

`salt.proxy.ssh_sample.`**`service_list`**`()`
Start a "service" on the ssh server

New in version 2015.8.2.

`salt.proxy.ssh_sample.`**`service_restart`**`(`*name*`)`
Restart a "service" on the ssh server

New in version 2015.8.2.

`salt.proxy.ssh_sample.`**`service_start`**`(`*name*`)`
Start a "service" on the ssh server

New in version 2015.8.2.

`salt.proxy.ssh_sample.`**`service_stop`**(*name*)

Stop a "service" on the ssh server

New in version 2015.8.2.

`salt.proxy.ssh_sample.`**`shutdown`**(*opts*)

Disconnect

21.14 queue modules

pgjsonb_queue	New in version 2016.3.0.
sqlite_queue	New in version 2014.7.0.

21.14.1 salt.queues.pgjsonb_queue module

New in version 2016.3.0.

This is a queue with postgres as the backend. It uses the jsonb store to store information for queues.

depends python-psycopg2

To enable this queue, the following needs to be configured in your master config. These are the defaults:

```
queue.pgjsonb.host: 'salt'
queue.pgjsonb.user: 'salt'
queue.pgjsonb.password: 'salt'
queue.pgjsonb.dbname: 'salt'
queue.pgjsonb.port: 5432
```

Use the following Pg database schema:

```
CREATE DATABASE  salt WITH ENCODING 'utf-8';

--
-- Table structure for table `salt`
--
DROP TABLE IF EXISTS salt;
CREATE OR REPLACE TABLE salt(
   id SERIAL PRIMARY KEY,
   data jsonb NOT NULL
);
```

```
salt-run queue.insert test '{"name": "redis", "host": "172.16.0.8", "port": 6379}'↩
↪backend=pgjsonb
salt-run queue.process_queue test all backend=pgjsonb
```

`salt.queues.pgjsonb_queue.`**`delete`**(*queue, items*)

Delete an item or items from a queue

`salt.queues.pgjsonb_queue.`**`handle_queue_creation`**(*queue*)

`salt.queues.pgjsonb_queue.`**`insert`**(*queue, items*)

Add an item or items to a queue

`salt.queues.pgjsonb_queue.`**`list_items`**`(`*queue*`)`
List contents of a queue

`salt.queues.pgjsonb_queue.`**`list_length`**`(`*queue*`)`
Provide the number of items in a queue

`salt.queues.pgjsonb_queue.`**`list_queues`**`()`
Return a list of Salt Queues on the Salt Master

`salt.queues.pgjsonb_queue.`**`pop`**`(`*queue, quantity=1, is_runner=False*`)`
Pop one or more or all items from the queue return them.

21.14.2 salt.queues.sqlite_queue module

New in version 2014.7.0.

This is the default local master event queue built on sqlite. By default, an sqlite3 database file is created in the *sqlite_queue_dir* which is found at:

```
/var/cache/salt/master/queues
```

It's possible to store the sqlite3 database files by setting *sqlite_queue_dir* to another location:

```
sqlite_queue_dir: /home/myuser/salt/master/queues
```

`salt.queues.sqlite_queue.`**`delete`**`(`*queue, items*`)`
Delete an item or items from a queue

`salt.queues.sqlite_queue.`**`insert`**`(`*queue, items*`)`
Add an item or items to a queue

`salt.queues.sqlite_queue.`**`list_items`**`(`*queue*`)`
List contents of a queue

`salt.queues.sqlite_queue.`**`list_length`**`(`*queue*`)`
Provide the number of items in a queue

`salt.queues.sqlite_queue.`**`list_queues`**`()`
Return a list of Salt Queues on the Salt Master

`salt.queues.sqlite_queue.`**`pop`**`(`*queue, quantity=1, is_runner=False*`)`
Pop one or more or all items from the queue return them.

21.15 roster modules

`ansible`	Read in an Ansible inventory file or script
`cache`	The `cache` roster provides a flexible interface to the Salt Masters' minion cache to access regular minions over `salt-ssh`.
`cloud`	Use the cloud cache on the master to derive IPv4 addresses based on minion ID.
`clustershell`	This roster resolves hostname in a pdsh/clustershell style.
`flat`	Read in the roster from a flat file using the renderer system

Continued on next page

Table 15 – continued from previous page

range	This roster resolves targets from a range server.
scan	Scan a netmask or ipaddr for open ssh ports
sshconfig	Parses roster entries out of Host directives from SSH config
terraform	Dynamic roster from terraform current state

21.15.1 salt.roster.ansible

Read in an Ansible inventory file or script

Flat inventory files should be in the regular ansible inventory format.

```
[servers]
salt.gtmanfred.com ansible_ssh_user=gtmanfred ansible_ssh_host=127.0.0.1 ansible_ssh_
↪port=22 ansible_ssh_pass='password'

[desktop]
home ansible_ssh_user=gtmanfred ansible_ssh_host=12.34.56.78 ansible_ssh_port=23
↪ansible_ssh_pass='password'

[computers:children]
desktop
servers

[names:vars]
http_port=80
```

then salt-ssh can be used to hit any of them

```
[~]# salt-ssh -N all test.ping
salt.gtmanfred.com:
    True
home:
    True
[~]# salt-ssh -N desktop test.ping
home:
    True
[~]# salt-ssh -N computers test.ping
salt.gtmanfred.com:
    True
home:
    True
[~]# salt-ssh salt.gtmanfred.com test.ping
salt.gtmanfred.com:
    True
```

There is also the option of specifying a dynamic inventory, and generating it on the fly

```
#!/bin/bash
echo '{
  "servers": [
    "salt.gtmanfred.com"
  ],
  "desktop": [
    "home"
  ],
```

(continues on next page)

(continued from previous page)

```
  "computers": {
    "hosts": [],
    "children": [
      "desktop",
      "servers"
    ]
  },
  "_meta": {
    "hostvars": {
      "salt.gtmanfred.com": {
        "ansible_ssh_user": "gtmanfred",
        "ansible_ssh_host": "127.0.0.1",
        "ansible_sudo_pass": "password",
        "ansible_ssh_port": 22
      },
      "home": {
        "ansible_ssh_user": "gtmanfred",
        "ansible_ssh_host": "12.34.56.78",
        "ansible_sudo_pass": "password",
        "ansible_ssh_port": 23
      }
    }
  }
}'
```

This is the format that an inventory script needs to output to work with ansible, and thus here.

```
[~]# salt-ssh --roster-file /etc/salt/hosts salt.gtmanfred.com test.ping
salt.gtmanfred.com:
        True
```

Any of the [groups] or direct hostnames will return. The 'all' is special, and returns everything.

`salt.roster.ansible.`**`targets`**(*tgt*, *tgt_type='glob'*, ***kwargs*)

Return the targets from the ansible inventory_file Default: /etc/salt/roster

21.15.2 salt.roster.cache

The `cache` roster provides a flexible interface to the Salt Masters' minion cache to access regular minions over `salt-ssh`.

New in version 2017.7.0: grains, pillar, mine data matching SDB URLs IPv6 support roster_order per config key default order changed to industry-wide best practices CIDR range selection

Targeting

This roster supports all matching and targeting of the Salt Master. The matching will be done using only the Salt Master's cache.

The Roster Order

The roster's composition can be configured using `roster_order`. In the `roster_order` you can define *any* roster key and fill it with a parameter overriding the one in `roster_defaults`:

-
  ```
  roster_order:
      host: id          # use the minion id as hostname
  ```

You can define lists of parameters as well, the first result from the list will become the value.

Selecting a host

```
# default
roster_order:
    host:
      - ipv6-private  # IPv6 addresses in private ranges
      - ipv6-global   # IPv6 addresses in global ranges
      - ipv4-private  # IPv4 addresses in private ranges
      - ipv4-public   # IPv4 addresses in public ranges
      - ipv4-local    # loopback addresses
```

This is the default `roster_order`. It prefers IPv6 over IPv4 addresses and private addresses over public ones. The relevant data will be fetched from the cache in-order, and the first match will fill the `host` key.

Other address selection parameters are also possible:

```
roster_order:
  host:
    - global|public|private|local    # Both IPv6 and IPv4 addresses in that range
    - 2000::/3                       # CIDR networks, both IPv4 and IPv6 are supported
```

Using cached data

Several cached libraries can be selected using the `library: `` prefix, followed by the library key. This can be referenced using the same ``:` syntax as e.g. *`pillar.get`*. Lists of references are also supported during the lookup, as are Salt SDB URLs.

This should be especially useful for the other roster keys:

```
roster_order:
  host:
    - grain: fqdn_ip4                # Lookup this grain
    - mine: network.ip_addrs         # Mine data lookup works the same

  password: sdb://vault/ssh_pass     # Salt SDB URLs are also supported

  user:
    - pillar: ssh:auth:user          # Lookup this pillar key
    - sdb://osenv/USER               # Lookup this env var through sdb

  priv:
    - pillar:                        # Lists are also supported
        - salt:ssh:private_key
        - ssh:auth:private_key
```

`salt.roster.cache.`**`targets`**(*tgt*, *tgt_type='glob'*, ***kwargs*)

Return the targets from the Salt Masters' minion cache. All targets and matchers are supported.

The resulting roster can be configured using `roster_order` and `roster_default`.

21.15.3 salt.roster.cloud

Use the cloud cache on the master to derive IPv4 addresses based on minion ID.

This roster requires that the minion in question was created using at least the 2015.5.0 version of Salt Cloud. Starting with the 2015.5.0 release, Salt Cloud maintains an index of minions that it creates and deletes. This index tracks the provider and profile configuration used to provision the minion, including authentication information. So long as this configuration remains current, it can be used by Salt SSH to log into any minion in the index.

To connect as a user other than root, modify the cloud configuration file usually located at /etc/salt/cloud. For example, add the following:

```
ssh_username: my_user
sudo: True
```

`salt.roster.cloud.`**`extract_ipv4`**(*roster_order*, *ipv4*)
Extract the preferred IP address from the ipv4 grain

`salt.roster.cloud.`**`targets`**(*tgt*, *tgt_type='glob'*, ***kwargs*)
Return the targets from the flat yaml file, checks opts for location but defaults to /etc/salt/roster

21.15.4 salt.roster.clustershell

This roster resolves hostname in a pdsh/clustershell style.

depends clustershell, https://github.com/cea-hpc/clustershell

When you want to use host globs for target matching, use `--roster clustershell`. For example:

```
salt-ssh --roster clustershell 'server_[1-10,21-30],test_server[5,7,9]' test.ping
```

`salt.roster.clustershell.`**`targets`**(*tgt*, *tgt_type='glob'*, ***kwargs*)
Return the targets

21.15.5 salt.roster.flat

Read in the roster from a flat file using the renderer system

`salt.roster.flat.`**`targets`**(*tgt*, *tgt_type='glob'*, ***kwargs*)
Return the targets from the flat yaml file, checks opts for location but defaults to /etc/salt/roster

21.15.6 salt.roster.range module

This roster resolves targets from a range server.

depends seco.range, https://github.com/ytoolshed/range

When you want to use a range query for target matching, use `--roster range`. For example:

```
salt-ssh --roster range '%%%example.range.cluster' test.ping
```

`salt.roster.range.`**`target_glob`**(*tgt*, *hosts*)

`salt.roster.range.`**`target_range`**(*tgt*, *hosts*)

`salt.roster.range.`**`targets`**(*tgt*, *tgt_type='range'*, ***kwargs*)
Return the targets from a range query

21.15.7 salt.roster.scan

Scan a netmask or ipaddr for open ssh ports

class `salt.roster.scan.`**`RosterMatcher`**(*tgt*, *tgt_type*)
Matcher for the roster data structure

`targets`()
Return ip addrs based on netmask, sitting in the "glob" spot because it is the default

`salt.roster.scan.`**`targets`**(*tgt*, *tgt_type='glob'*, ***kwargs*)
Return the targets from the flat yaml file, checks opts for location but defaults to /etc/salt/roster

21.15.8 salt.roster.sshconfig

Parses roster entries out of Host directives from SSH config

```
salt-ssh --roster sshconfig '*' -r "echo hi"
```

class `salt.roster.sshconfig.`**`RosterMatcher`**(*raw*, *tgt*, *tgt_type*)
Matcher for the roster data structure

`get_data`(*minion*)
Return the configured ip

`ret_glob_minions`()
Return minions that match via glob

`targets`()
Execute the correct tgt_type routine and return

`salt.roster.sshconfig.`**`parse_ssh_config`**(*lines*)
Parses lines from the SSH config to create roster targets.

Parameters **`lines`** – Individual lines from the ssh config file
Returns Dictionary of targets in similar style to the flat roster

`salt.roster.sshconfig.`**`targets`**(*tgt*, *tgt_type='glob'*, ***kwargs*)
Return the targets from the flat yaml file, checks opts for location but defaults to /etc/salt/roster

21.15.9 salt.roster.terraform

Dynamic roster from terraform current state

This roster module allows you dynamically generate the roster from the terraform resources defined with the Terraform Salt provider.

It exposes all salt_host resources with the same attributes to the salt-ssh roster, making it completely independent of the type of terraform resource, and providing the integration using terraform constructs with interpolation.

Basic Example

Given a simple salt-ssh tree with a Saltfile:

```
salt-ssh:
  config_dir: etc/salt
  max_procs: 30
  wipe_ssh: True
```

and `etc/salt/master`:

```
root_dir: .
  file_roots:
    base:
      - srv/salt
  pillar_roots:
    base:
      - srv/pillar
  roster: terraform
```

In the same folder as your `Saltfile`, create terraform file with resources like cloud instances, virtual machines, etc. For every single one of those that you want to manage with Salt, create a `salt_host` resource:

```
resource "salt_host" "dbminion" {
  salt_id = "dbserver"
  host = "${libvirt_domain.vm-db.network_interface.0.addresses.0}"
  user = "root"
  passwd = "linux"
}
```

You can use the count attribute to create multiple roster entries with a single definition. Please refer to the Terraform Salt provider for more detailed examples.

`salt.roster.terraform.`**`targets`**(*tgt*, *tgt_type='glob'*, ***kwargs*)

Returns the roster from the terraform state file, checks opts for location, but defaults to terraform.tfstate

21.16 runner modules

asam	Novell ASAM Runner
auth	Authentication runner for creating, deleting, and managing eauth tokens.
bgp	BGP Finder
cache	Return cached data from minions
cloud	The Salt Cloud Runner
ddns	Dynamic DNS Runner
digicertapi	Support for Digicert.
doc	A runner module to collect and display the inline documentation from the various module types
drac	Manage Dell DRAC from the Master
error	Error generator to enable integration testing of salt runner error handling
event	Module for sending events using the runner system.
f5	Runner to provide F5 Load Balancer functionality
fileserver	Directly manage the Salt fileserver plugins
git_pillar	Runner module to directly manage the git external pillar

Continued on next page

Table 16 – continued from previous page

http	Module for making various web calls.
jobs	A convenience system to manage jobs, both active and already run
launchd	Manage launchd plist files
lxc	Control Linux Containers via Salt
manage	General management functions for salt, tools like seeing what hosts are up and what hosts are down
mattermost	Module for sending messages to Mattermost
mine	A runner to access data from the salt mine
nacl	This module helps include encrypted passwords in pillars, grains and salt state files.
net	NET Finder
network	Network tools to run from the Master
pagerduty	Runner Module for Firing Events via PagerDuty
pillar	Functions to interact with the pillar compiler on the master
pkg	Package helper functions using `salt.modules.pkg`
queue	General management and processing of queues.
reactor	A convenience system to manage reactors
salt	This runner makes Salt's execution modules available on the salt master.
saltutil	The Saltutil runner is used to sync custom types to the Master.
sdb	Runner for setting and querying data via the sdb API on the master
smartos_vmadm	Runner for SmartOS minions control vmadm
spacewalk	Spacewalk Runner
ssh	A Runner module interface on top of the salt-ssh Python API.
state	Execute orchestration functions
survey	A general map/reduce style salt runner for aggregating results returned by several different minions.
test	This runner is used only for test purposes and servers no production purpose
thin	The thin runner is used to manage the salt thin systems.
vault	**maintainer** SaltStack
venafiapi	Support for Venafi
virt	Control virtual machines via Salt
vistara	Vistara Runner
winrepo	Runner to manage Windows software repo

21.16.1 salt.runners.asam

Novell ASAM Runner

New in version Beryllium.

Runner to interact with Novell ASAM Fan-Out Driver

codeauthor Nitin Madhok <nmadhok@clemson.edu>

To use this runner, set up the Novell Fan-Out Driver URL, username and password in the master configuration at `/etc/salt/master` or `/etc/salt/master.d/asam.conf`:

```
asam:
  prov1.domain.com
    username: "testuser"
    password: "verybadpass"
  prov2.domain.com
    username: "testuser"
    password: "verybadpass"
```

Note: Optionally, `protocol` and `port` can be specified if the Fan-Out Driver server is not using the defaults. Default is `protocol: https` and `port: 3451`.

`salt.runners.asam.`**`add_platform`**(*name*, *platform_set*, *server_url*)

To add an ASAM platform using the specified ASAM platform set on the Novell Fan-Out Driver

CLI Example:

```
salt-run asam.add_platform my-test-vm test-platform-set prov1.domain.com
```

`salt.runners.asam.`**`list_platform_sets`**(*server_url*)

To list all ASAM platform sets present on the Novell Fan-Out Driver

CLI Example:

```
salt-run asam.list_platform_sets prov1.domain.com
```

`salt.runners.asam.`**`list_platforms`**(*server_url*)

To list all ASAM platforms present on the Novell Fan-Out Driver

CLI Example:

```
salt-run asam.list_platforms prov1.domain.com
```

`salt.runners.asam.`**`remove_platform`**(*name*, *server_url*)

To remove specified ASAM platform from the Novell Fan-Out Driver

CLI Example:

```
salt-run asam.remove_platform my-test-vm prov1.domain.com
```

21.16.2 salt.runners.auth

Authentication runner for creating, deleting, and managing eauth tokens.

New in version 2016.11.0.

`salt.runners.auth.`**`del_token`**(*token*)

Delete an eauth token by name

CLI Example:

```
salt-run auth.del_token 6556760736e4077daa601baec2b67c24
```

`salt.runners.auth.`**`mk_token`**`(`*`**load`*`)`

Create an eauth token using provided credentials

Non-root users may specify an expiration date – if allowed via the *`token_expire_user_override`* setting – by passing an additional `token_expire` param. This overrides the *`token_expire`* setting of the same name in the Master config and is how long a token should live in seconds.

CLI Example:

```
salt-run auth.mk_token username=saltdev password=saltdev eauth=auto

# Create a token valid for three years.
salt-run auth.mk_token username=saltdev password=saltdev eauth=auto \
    token_expire=94670856

# Calculate the number of seconds using expr.
salt-run auth.mk_token username=saltdev password=saltdev eauth=auto \
    token_expire=$(expr \( 365 \* 24 \* 60 \* 60 \) \* 3)
```

21.16.3 salt.runners.bgp

BGP Finder

New in version 2017.7.0.

Runner to search BGP neighbors details.

Configuration

- Minion (proxy) config

  The `bgp.neighbors` function must be appened in the list of `mine_functions`:

  ```
  mine_functions:
    bgp.neighbors: []
  ```

  Which instructs Salt to cache the data returned by the `neighbors` function from the *`NAPALM BGP module`*.

  How often the mines are refreshed, can be specified using:

  ```
  mine_interval: <X minutes>
  ```

- Master config

  By default the following options can be configured on the master. They are not mandatory, but available in case the user has different requirements.

  tgt: `*` From what minions will collect the mine data. Default: `*` (collect mine data from all minions)

  tgt_type: `glob` Minion matching expression form. Default: `glob`.

  return_fields What fields to return in the output. It can display all the fields from the `neighbors` function from the *`NAPALM BGP module`*.

  Some fields cannot be removed:

  - `as_number`: the AS number of the neighbor

- `device`: the minion ID
- `neighbor_address`: the neighbor remote IP address

By default, the following extra fields are returned (displayed):

- `connection_stats`: connection stats, as described below
- `import_policy`: the name of the import policy
- `export_policy`: the name of the export policy

Special fields:

- `vrf`: return the name of the VRF.
- `connection_stats`: returning an output of the form `<State> <Active>/<Received>/<Accepted>/<Damped>`, e.g. `Established 398/399/399/0` similar to the usual output from network devices.
- `interface_description`: matches the neighbor details with the corresponding interface and returns its description. This will reuse functionality from the *`net runner`*, so the user needs to enable the mines as specified in the documentation.
- `interface_name`: matches the neighbor details with the corresponding interface and returns the name. Similar to `interface_description`, this will reuse functionality from the *`net runner`*, so the user needs to enable the mines as specified in the documentation.

display: **`True`** Display on the screen or return structured object? Default: `True` (return on the CLI).

outputter: **`table`** Specify the outputter name when displaying on the CLI. Default: *`table`*.

Configuration example:

```
runners:
  bgp:
    tgt: 'edge*'
    tgt_type: 'glob'
    return_fields:
      - up
      - connection_state
      - previous_connection_state
      - suppress_4byte_as
      - holdtime
      - flap_count
    outputter: yaml
```

`salt.runners.bgp.`**`neighbors`**`(`**asns, **kwargs*`)`

Search for BGP neighbors details in the mines of the `bgp.neighbors` function.

Arguments:

asns A list of AS numbers to search for. The runner will return only the neighbors of these AS numbers.

device Filter by device name (minion ID).

ip Search BGP neighbor using the IP address. In multi-VRF environments, the same IP address could be used by more than one neighbors, in different routing tables.

network Search neighbors within a certain IP network.

title Custom title.

display: **`True`** Display on the screen or return structured object? Default: `True` (return on the CLI).

outputter: **`table`** Specify the outputter name when displaying on the CLI. Default: *`table`*.

In addition, any field from the output of the `neighbors` function from the *`NAPALM BGP module`* can be used as a filter.

CLI Example:

```
salt-run bgp.neighbors 13335 15169
salt-run bgp.neighbors 13335 ip=172.17.19.1
salt-run bgp.neighbors multipath=True
salt-run bgp.neighbors up=False export_policy=my-export-policy multihop=False
salt-run bgp.neighbors network=192.168.0.0/16
```

Output example:

```
BGP Neighbors for 13335, 15169
----------------------------------------------------------------------------------
↪------------------------------------------------------------------------------
|    Device     | AS Number |         Neighbor Address         | State|#Active/
↪Received/Accepted/Damped |         Policy IN           |         Policy OUT      ⊠
↪   |
----------------------------------------------------------------------------------
↪------------------------------------------------------------------------------
| edge01.bjm01 |   13335   |          172.17.109.11           |        Established⊠
↪0/398/398/0         |       import-policy       |        export-policy       |
----------------------------------------------------------------------------------
↪------------------------------------------------------------------------------
| edge01.bjm01 |   13335   |          172.17.109.12           |       Established⊠
↪397/398/398/0         |       import-policy       |        export-policy       |
----------------------------------------------------------------------------------
↪------------------------------------------------------------------------------
| edge01.flw01 |   13335   |          192.168.172.11          |        Established⊠
↪1/398/398/0         |       import-policy       |        export-policy       |
----------------------------------------------------------------------------------
↪------------------------------------------------------------------------------
| edge01.oua01 |   13335   |          172.17.109.17           |         ⊠
↪Established 0/0/0/0           |       import-policy       |        export-
↪policy       |
----------------------------------------------------------------------------------
↪------------------------------------------------------------------------------
| edge01.bjm01 |   15169   |             2001::1              |       Established⊠
↪102/102/102/0         |       import-policy       |        export-policy       |
----------------------------------------------------------------------------------
↪------------------------------------------------------------------------------
| edge01.bjm01 |   15169   |             2001::2              |       Established⊠
↪102/102/102/0         |       import-policy       |        export-policy       |
----------------------------------------------------------------------------------
↪------------------------------------------------------------------------------
| edge01.tbg01 |   13335   |          192.168.172.17          |         ⊠
↪Established 0/1/1/0           |       import-policy       |        export-
↪policy       |
----------------------------------------------------------------------------------
↪------------------------------------------------------------------------------
```

21.16.4 salt.runners.cache

Return cached data from minions

`salt.runners.cache.`**`clear_all`**(*tgt=None, tgt_type='glob'*)

Changed in version 2017.7.0: The `expr_form` argument has been renamed to `tgt_type`, earlier releases must use `expr_form`.

Clear the cached pillar, grains, and mine data of the targeted minions

CLI Example:

```
salt-run cache.clear_all
```

`salt.runners.cache.`**`clear_git_lock`**(*role, remote=None, **kwargs*)

New in version 2015.8.2.

Remove the update locks for Salt components (gitfs, git_pillar, winrepo) which use gitfs backend code from salt.utils.gitfs.

Note: Running *`cache.clear_all`* will not include this function as it does for pillar, grains, and mine.

Additionally, executing this function with a `role` of `gitfs` is equivalent to running `salt-run fileserver.clear_lock backend=git`.

---

role Which type of lock to remove (`gitfs`, `git_pillar`, or `winrepo`)

remote If specified, then any remotes which contain the passed string will have their lock cleared. For example, a `remote` value of **github** will remove the lock from all github.com remotes.

type [update,checkout,mountpoint] The types of lock to clear. Can be one or more of `update`, `checkout`, and `mountpoint`, and can be passed either as a comma-separated or Python list.

New in version 2015.8.8.

Changed in version 2018.3.0: `mountpoint` lock type added

CLI Examples:

```
salt-run cache.clear_git_lock gitfs
salt-run cache.clear_git_lock git_pillar
salt-run cache.clear_git_lock git_pillar type=update
salt-run cache.clear_git_lock git_pillar type=update,checkout
salt-run cache.clear_git_lock git_pillar type='["update", "mountpoint"]'
```

`salt.runners.cache.`**`clear_grains`**(*tgt=None, tgt_type='glob'*)

Changed in version 2017.7.0: The `expr_form` argument has been renamed to `tgt_type`, earlier releases must use `expr_form`.

Clear the cached grains data of the targeted minions

CLI Example:

```
salt-run cache.clear_grains
```

`salt.runners.cache.`**`clear_mine`**(*tgt=None, tgt_type='glob'*)

Changed in version 2017.7.0: The `expr_form` argument has been renamed to `tgt_type`, earlier releases must use `expr_form`.

Clear the cached mine data of the targeted minions

CLI Example:

```
salt-run cache.clear_mine
```

`salt.runners.cache.`**`clear_mine_func`**(*tgt=None, tgt_type='glob', clear_mine_func_flag=None*)

Changed in version 2017.7.0: The `expr_form` argument has been renamed to `tgt_type`, earlier releases must use `expr_form`.

Clear the cached mine function data of the targeted minions

CLI Example:

```
salt-run cache.clear_mine_func tgt='*' clear_mine_func_flag='network.interfaces'
```

`salt.runners.cache.`**`clear_pillar`**(*tgt=None*, *tgt_type='glob'*)

Changed in version 2017.7.0: The `expr_form` argument has been renamed to `tgt_type`, earlier releases must use `expr_form`.

Clear the cached pillar data of the targeted minions

CLI Example:

```
salt-run cache.clear_pillar
```

`salt.runners.cache.`**`cloud`**(*tgt*, *provider=None*)

Return cloud cache data for target.

Note: Only works with glob matching

tgt Glob Target to match minion ids

provider Cloud Provider

CLI Example:

```
salt-run cache.cloud 'salt*'
salt-run cache.cloud glance.example.org provider=openstack
```

`salt.runners.cache.`**`fetch`**(*bank*, *key*, *cachedir=None*)

Fetch data from a salt.cache bank.

CLI Example:

```
salt-run cache.fetch cloud/active/ec2/myec2 myminion cachedir=/var/cache/salt/
```

`salt.runners.cache.`**`flush`**(*bank*, *key=None*, *cachedir=None*)

Remove the key from the cache bank with all the key content. If no key is specified remove the entire bank with all keys and sub-banks inside.

CLI Examples:

```
salt-run cache.flush cloud/active/ec2/myec2 cachedir=/var/cache/salt/
salt-run cache.flush cloud/active/ec2/myec2 myminion cachedir=/var/cache/salt/
```

`salt.runners.cache.`**`grains`**(*tgt=None*, *tgt_type='glob'*, ***kwargs*)

Changed in version 2017.7.0: The `expr_form` argument has been renamed to `tgt_type`, earlier releases must use `expr_form`.

Return cached grains of the targeted minions.

tgt Target to match minion ids.

> Changed in version 2017.7.5,2018.3.0: The `tgt` argument is now required to display cached grains. If not used, the function will not return grains. This optional argument will become mandatory in the Salt `Sodium` release.

tgt_type The type of targeting to use for matching, such as `glob`, `list`, etc.

CLI Example:

```
salt-run cache.grains '*'
```

`salt.runners.cache.`**`list_`**(*bank*, *cachedir=None*)

Lists entries stored in the specified bank.

CLI Example:

```
salt-run cache.list cloud/active/ec2/myec2 cachedir=/var/cache/salt/
```

`salt.runners.cache.`**`mine`**(*tgt=None*, *tgt_type='glob'*, ***kwargs*)

Changed in version 2017.7.0: The `expr_form` argument has been renamed to `tgt_type`, earlier releases must use `expr_form`.

Return cached mine data of the targeted minions

CLI Example:

```
salt-run cache.mine
```

`salt.runners.cache.`**`pillar`**(*tgt=None*, *tgt_type='glob'*, ***kwargs*)

Changed in version 2017.7.0: The `expr_form` argument has been renamed to `tgt_type`, earlier releases must use `expr_form`.

Return cached pillars of the targeted minions if tgt is set. If tgt is not set will return cached pillars for all minions.

CLI Example:

```
salt-run cache.pillar
```

`salt.runners.cache.`**`store`**(*bank*, *key*, *data*, *cachedir=None*)

Lists entries stored in the specified bank.

CLI Example:

```
salt-run cache.store mycache mykey 'The time has come the walrus said'
```

21.16.5 salt.runners.cloud

The Salt Cloud Runner

This runner wraps the functionality of salt cloud making salt cloud routines available to all internal apis via the runner system

`salt.runners.cloud.`**`action`**(*func=None*, *cloudmap=None*, *instances=None*, *provider=None*, *instance=None*, *opts=None*, ***kwargs*)

Execute a single action on the given map/provider/instance

CLI Example:

```
salt-run cloud.action start my-salt-vm
```

`salt.runners.cloud.`**`create`**(*provider*, *instances*, *opts=None*, ***kwargs*)

Create an instance using Salt Cloud

CLI Example:

```
salt-run cloud.create my-ec2-config myinstance \
    image=ami-1624987f size='t1.micro' ssh_username=ec2-user \
    securitygroup=default delvol_on_destroy=True
```

`salt.runners.cloud.`**`destroy`**(*instances*, *opts=None*)
Destroy the named vm(s)

`salt.runners.cloud.`**`full_query`**(*query_type='list_nodes_full'*)
List all available cloud provider data

`salt.runners.cloud.`**`list_images`**(*provider='all'*)
List cloud provider images for the given providers

`salt.runners.cloud.`**`list_locations`**(*provider='all'*)
List cloud provider sizes for the given providers

`salt.runners.cloud.`**`list_sizes`**(*provider='all'*)
List cloud provider sizes for the given providers

`salt.runners.cloud.`**`map_run`**(*path=None*, *opts=None*, ***kwargs*)
Execute a salt cloud map file

`salt.runners.cloud.`**`profile`**(*prof=None*, *instances=None*, *opts=None*, ***kwargs*)
Create a cloud vm with the given profile and instances, instances can be a list or comma-delimited string

CLI Example:

```
salt-run cloud.profile prof=my-ec2 instances=node1,node2,node3
```

`salt.runners.cloud.`**`query`**(*query_type='list_nodes'*)
List cloud provider data for all providers

`salt.runners.cloud.`**`select_query`**(*query_type='list_nodes_select'*)
List selected nodes

21.16.6 salt.runners.ddns

Dynamic DNS Runner

New in version Beryllium.

Runner to interact with DNS server and create/delete/update DNS records

codeauthor Nitin Madhok <nmadhok@clemson.edu>

`salt.runners.ddns.`**`add_host`**(*zone*, *name*, *ttl*, *ip*, *keyname*, *keyfile*, *nameserver*, *timeout*, *port=53*, *keyalgorithm='hmac-md5'*)
Create both A and PTR (reverse) records for a host.

CLI Example:

```
salt-run ddns.add_host domain.com my-test-vm 3600 10.20.30.40 my-tsig-key /etc/
↪salt/tsig.keyring 10.0.0.1 5
```

`salt.runners.ddns.`**`create`**(*zone*, *name*, *ttl*, *rdtype*, *data*, *keyname*, *keyfile*, *nameserver*, *timeout*, *port=53*, *keyalgorithm='hmac-md5'*)
Create a DNS record. The nameserver must be an IP address and the master running this runner must have create privileges on that server.

CLI Example:

```
salt-run ddns.create domain.com my-test-vm 3600 A 10.20.30.40 my-tsig-key /etc/
↪salt/tsig.keyring 10.0.0.1 5
```

`salt.runners.ddns.`**`delete`**`(`*zone, name, keyname, keyfile, nameserver, timeout, rdtype=None, data=None, port=53, keyalgorithm='hmac-md5'*`)`

Delete a DNS record.

CLI Example:

```
salt-run ddns.delete domain.com my-test-vm my-tsig-key /etc/salt/tsig.keyring 10.
↪0.0.1 5 A
```

`salt.runners.ddns.`**`delete_host`**`(`*zone, name, keyname, keyfile, nameserver, timeout, port=53, keyalgorithm='hmac-md5'*`)`

Delete both forward (A) and reverse (PTR) records for a host only if the forward (A) record exists.

CLI Example:

```
salt-run ddns.delete_host domain.com my-test-vm my-tsig-key /etc/salt/tsig.
↪keyring 10.0.0.1 5
```

`salt.runners.ddns.`**`update`**`(`*zone, name, ttl, rdtype, data, keyname, keyfile, nameserver, timeout, replace=False, port=53, keyalgorithm='hmac-md5'*`)`

Replace, or update a DNS record. The nameserver must be an IP address and the master running this runner must have update privileges on that server.

Note: If `replace` is set to True, all records for this name and type will first be deleted and then recreated. Default is `replace=False`.

CLI Example:

```
salt-run ddns.update domain.com my-test-vm 3600 A 10.20.30.40 my-tsig-key /etc/
↪salt/tsig.keyring 10.0.0.1 5
```

21.16.7 salt.runners.digicertapi

Support for Digicert. Heavily based on the Venafi runner by Joseph Hall (jphall@saltstack.com).

Before using this module you need to register an account with Digicert's CertCentral.

Login to CertCentral, ensure you have a payment method configured and/or there are adequate funds attached to your account. Click the `Account` item in the left sidebar, and select `Account Access`. The right hand pane should show "Account Access" and a link to create an API key. Create a new API key and assign it to the user that should be attached to requests coming from Salt.

NOTE CertCentral will not show the API key again after revealing it the first time. Make sure you copy it right away or you will have to revoke it and generate a new one.

Now open `/etc/salt/master` and add the API key as shown below.

```
digicert:
  api_key:␣
↪ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABC
```

Restart your Salt Master.

You can also include default values of the following variables to help with creating CSRs:

```
digicert:
  api_key:
↪ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZABC
  shatype: sha256
```

This API currently only supports RSA key types. Support for other key types will be added if interest warrants.

`salt.runners.digicertapi.`**`del_cached_domain`**(*domains*)

Delete cached domains from the master

CLI Example:

```
salt-run digicert.del_cached_domain domain1.example.com,domain2.example.com
```

`salt.runners.digicertapi.`**`gen_csr`**(*minion_id, dns_name, organization_id, ou_name=None, key_len=2048, shatype='sha256', password=None*)

CLI Example:

```
salt-run digicert.gen_csr <minion_id> <dns_name>
```

`salt.runners.digicertapi.`**`gen_key`**(*minion_id, dns_name=None, password=None, key_len=2048*)

Generate and return a private_key. If a `dns_name` is passed in, the private_key will be cached under that name.

CLI Example:

```
salt-run digicert.gen_key <minion_id> [dns_name] [password]
```

`salt.runners.digicertapi.`**`get_certificate`**(*order_id=None, certificate_id=None, minion_id=None, cert_format='pem_all', filename=None*)

Retrieve a certificate by order_id or certificate_id and write it to stdout or a filename.

A list of permissible cert_formats is here: https://www.digicert.com/services/v2/documentation/appendix-certificate-formats

CLI Example:

```
salt-run digicert.get_certificate order_id=48929454 cert_format=apache
```

Including a 'filename' will write the certificate to the desired file. Note that some cert formats are zipped files, and some are binary.

If the certificate has not been issued, this function will return the order details inside of which will be a status (one of pending, rejected, processing, issued, revoked, canceled, needs_csr, and needs_approval)

If for some reason you want to pipe the output of this command to a file or other command you will want to leave off the `filename` argument and make sure to include `--no-color` so there will be no terminal ANSI escape sequences.

`salt.runners.digicertapi.`**`get_org_details`**(*organization_id*)

Return the details for an organization

CLI Example:

```
salt-run digicert.get_org_details 34
```

Returns a dictionary with the org details, or with 'error' and 'status' keys.

`salt.runners.digicertapi.`**`list_domain_cache`**()

List domains that have been cached

CLI Example:

```
salt-run digicert.list_domain_cache
```

`salt.runners.digicertapi.`**`list_domains`**(*container_id=None*)

List domains that CertCentral knows about. You can filter by container_id (also known as "Division") by passing a container_id.

CLI Example:

```
salt-run digicert.list_domains
```

`salt.runners.digicertapi.`**`list_orders`**(*status=None*)

List certificate orders made to CertCentral.

CLI Example:

```
salt-run digicert.list_orders
```

`salt.runners.digicertapi.`**`list_organizations`**(*container_id=None, include_validation=True*)

List organizations that CertCentral knows about. You can filter by container_id (also known as "Division") by passing a container_id. This function returns validation information by default; pass `include_validation=False` to turn it off.

CLI Example:

```
salt-run digicert.list_organizations
```

`salt.runners.digicertapi.`**`list_requests`**(*status=None*)

List certificate requests made to CertCentral. You can filter by status: `pending`, `approved`, `rejected`

CLI Example:

```
salt-run digicert.list_requests pending
```

`salt.runners.digicertapi.`**`order_certificate`**(*minion_id, common_name, organization_id, validity_years, cert_key_passphrase=None, signature_hash=None, key_len=2048, dns_names=None, organization_units=None, server_platform=None, custom_expiration_date=None, comments=None, disable_renewal_notifications=False, product_type_hint=None, renewal_of_order_id=None*)

Order a certificate. Requires that an Organization has been created inside Digicert's CertCentral.

See here for API documentation: https://www.digicert.com/services/v2/documentation/order/order-ssl-determinator

CLI Example:

```
salt-run digicert.order_certificate my_minionid my.domain.com 10           3
↪signature_hash=sha256           dns_names=['this.domain.com', 'that.domain.com
↪']           organization_units='My Domain Org Unit'           comments=
↪'Comment goes here for the approver'
```

This runner can also be used to renew a certificate by passing *renewal_of_order_id*. Previous order details can be retrieved with digicertapi.list_orders.

`salt.runners.digicertapi.`**`show_csrs`**`()`

Show certificate requests for this API key

CLI Example:

```
salt-run digicert.show_csrs
```

`salt.runners.digicertapi.`**`show_organization`**`(`*domain*`)`

Show organization information, especially the company id

CLI Example:

```
salt-run digicert.show_company example.com
```

`salt.runners.digicertapi.`**`show_rsa`**`(`*minion_id, dns_name*`)`

Show a private RSA key

CLI Example:

```
salt-run digicert.show_rsa myminion domain.example.com
```

21.16.8 salt.runners.doc

A runner module to collect and display the inline documentation from the various module types

`salt.runners.doc.`**`execution`**`()`

Collect all the sys.doc output from each minion and return the aggregate

CLI Example:

```
salt-run doc.execution
```

`salt.runners.doc.`**`runner`**`()`

Return all inline documentation for runner modules

CLI Example:

```
salt-run doc.runner
```

`salt.runners.doc.`**`wheel`**`()`

Return all inline documentation for wheel modules

CLI Example:

```
salt-run doc.wheel
```

21.16.9 salt.runners.drac

Manage Dell DRAC from the Master

The login credentials need to be configured in the Salt master configuration file.

```
drac:
  username: admin
  password: secret
```

`salt.runners.drac.`**`poweroff`**(*hostname, timeout=20, username=None, password=None*)

Power server off

CLI Example:

```
salt-run drac.poweroff example.com
```

`salt.runners.drac.`**`poweron`**(*hostname, timeout=20, username=None, password=None*)

Power server on

CLI Example:

```
salt-run drac.poweron example.com
```

`salt.runners.drac.`**`pxe`**(*hostname, timeout=20, username=None, password=None*)

Connect to the Dell DRAC and have the boot order set to PXE and power cycle the system to PXE boot

CLI Example:

```
salt-run drac.pxe example.com
```

`salt.runners.drac.`**`reboot`**(*hostname, timeout=20, username=None, password=None*)

Reboot a server using the Dell DRAC

CLI Example:

```
salt-run drac.reboot example.com
```

`salt.runners.drac.`**`version`**(*hostname, timeout=20, username=None, password=None*)

Display the version of DRAC

CLI Example:

```
salt-run drac.version example.com
```

21.16.10 salt.runners.error

Error generator to enable integration testing of salt runner error handling

`salt.runners.error.`**`error`**(*name=None, message=''*)

If name is None Then return empty dict

Otherwise raise an exception with __name__ from name, message from message

CLI Example:

```
salt-run error
salt-run error.error name="Exception" message="This is an error."
```

21.16.11 salt.runners.event

Module for sending events using the runner system.

New in version 2016.11.0.

`salt.runners.event.`**`send`**(*tag, data=None*)

Send an event with the given tag and data.

This is useful for sending events directly to the master from the shell with salt-run. It is also quite useful for sending events in orchestration states where the `fire_event` requisite isn't sufficient because it does not support sending custom data with the event.

Note that event tags will *not* be namespaced like events sent with the `fire_event` requisite! Whereas events produced from `fire_event` are prefixed with `salt/state_result/<jid>/<minion_id>/<name>`, events sent using this runner module will have no such prefix. Make sure your reactors don't expect a prefix!

Parameters

- **`tag`** – the tag to send with the event
- **`data`** – an optional dictionary of data to send with the event

CLI Example:

```
salt-run event.send my/custom/event '{"foo": "bar"}'
```

Orchestration Example:

```
# orch/command.sls

run_a_command:
  salt.function:
    - name: cmd.run
    - tgt: my_minion
    - arg:
      - exit {{ pillar['exit_code'] }}

send_success_event:
  salt.runner:
    - name: event.send
    - tag: my_event/success
    - data:
        foo: bar
    - require:
      - salt: run_a_command

send_failure_event:
  salt.runner:
    - name: event.send
    - tag: my_event/failure
    - data:
        baz: qux
    - onfail:
      - salt: run_a_command
```

```
salt-run state.orchestrate orch.command pillar='{"exit_code": 0}'
salt-run state.orchestrate orch.command pillar='{"exit_code": 1}'
```

21.16.12 salt.runners.f5

Runner to provide F5 Load Balancer functionality

depends

- pycontrol Python module

configuration In order to connect to a F5 Load Balancer, you must specify in the Salt master configuration the currently available load balancers

```
load_balancers:
  bigip1.example.com:
    username: admin
    password: secret
  bigip2.example.com:
    username: admin
    password: secret
```

class `salt.runners.f5.`**`F5Mgmt`**(*lb, username, password*)

`add_pool_member`(*name, port, pool_name*)
Add a node to a pool

`check_member_pool`(*member, pool_name*)
Check a pool member exists in a specific pool

`check_pool`(*name*)
Check to see if a pool exists

`check_virtualserver`(*name*)
Check to see if a virtual server exists

`create_pool`(*name, method='ROUND_ROBIN'*)
Create a pool on the F5 load balancer

`create_vs`(*name, ip, port, protocol, profile, pool_name*)
Create a virtual server

`lbmethods`()
List all the load balancer methods

`salt.runners.f5.`**`add_pool_member`**(*lb, name, port, pool_name*)
Add a node to a pool

CLI Examples:

```
salt-run f5.add_pool_member load_balancer 10.0.0.1 80 my_pool
```

`salt.runners.f5.`**`check_member_pool`**(*lb, member, pool_name*)
Check a pool member exists in a specific pool

CLI Examples:

```
salt-run f5.check_member_pool load_balancer 10.0.0.1 my_pool
```

`salt.runners.f5.`**`check_pool`**(*lb, name*)
Check to see if a pool exists

CLI Examples:

```
salt-run f5.check_pool load_balancer pool_name
```

`salt.runners.f5.`**`check_virtualserver`**(*lb, name*)
Check to see if a virtual server exists

CLI Examples:

```
salt-run f5.check_virtualserver load_balancer virtual_server
```

`salt.runners.f5.`**`create_pool`**(*lb*, *name*, *method='ROUND_ROBIN'*)

Create a pool on the F5 load balancer

CLI Examples:

```
salt-run f5.create_pool load_balancer pool_name loadbalance_method
salt-run f5.create_pool load_balancer my_pool ROUND_ROBIN
```

`salt.runners.f5.`**`create_vs`**(*lb*, *name*, *ip*, *port*, *protocol*, *profile*, *pool_name*)

Create a virtual server

CLI Examples:

```
salt-run f5.create_vs lbalancer vs_name 10.0.0.1 80 tcp http poolname
```

21.16.13 salt.runners.fileserver

Directly manage the Salt fileserver plugins

`salt.runners.fileserver.`**`clear_cache`**(*backend=None*)

New in version 2015.5.0.

Clear the fileserver cache from VCS fileserver backends (*`git`*, *`hg`*, *`svn`*). Executing this runner with no arguments will clear the cache for all enabled VCS fileserver backends, but this can be narrowed using the `backend` argument.

backend Only clear the update lock for the specified backend(s). If all passed backends start with a minus sign (`-`), then these backends will be excluded from the enabled backends. However, if there is a mix of backends with and without a minus sign (ex: `backend=-roots,git`) then the ones starting with a minus sign will be disregarded.

CLI Example:

```
salt-run fileserver.clear_cache
salt-run fileserver.clear_cache backend=git,hg
salt-run fileserver.clear_cache hg
salt-run fileserver.clear_cache -roots
```

`salt.runners.fileserver.`**`clear_file_list_cache`**(*saltenv=None*, *backend=None*)

New in version 2016.11.0.

The Salt fileserver caches the files/directories/symlinks for each fileserver backend and environment as they are requested. This is done to help the fileserver scale better. Without this caching, when hundreds/thousands of minions simultaneously ask the master what files are available, this would cause the master's CPU load to spike as it obtains the same information separately for each minion.

saltenv By default, this runner will clear the file list caches for all environments. This argument allows for a list of environments to be passed, to clear more selectively. This list can be passed either as a comma-separated string, or a Python list.

backend Similar to the `saltenv` parameter, this argument will restrict the cache clearing to specific fileserver backends (the default behavior is to clear from all enabled fileserver backends). This list can be passed either as a comma-separated string, or a Python list.

Since the ability to clear these caches is often required by users writing custom runners which add/remove files, this runner can easily be called from within a custom runner using any of the following examples:

```
# Clear all file list caches
__salt__['fileserver.clear_file_list_cache']()
# Clear just the 'base' saltenv file list caches
__salt__['fileserver.clear_file_list_cache'](saltenv='base')
```

(continues on next page)

(continued from previous page)

```
# Clear just the 'base' saltenv file list caches from just the 'roots'
# fileserver backend
__salt__['fileserver.clear_file_list_cache'](saltenv='base', backend='roots')
# Clear all file list caches from the 'roots' fileserver backend
__salt__['fileserver.clear_file_list_cache'](backend='roots')
```

Note: In runners, the `__salt__` dictionary will likely be renamed to `__runner__` in a future Salt release to distinguish runner functions from remote execution functions. See this GitHub issue for discussion/updates on this.

If using Salt's Python API (not a runner), the following examples are equivalent to the ones above:

```
import salt.config
import salt.runner

opts = salt.config.master_config('/etc/salt/master')
opts['fun'] = 'fileserver.clear_file_list_cache'

# Clear all file list_caches
opts['arg'] = []  # No arguments
runner = salt.runner.Runner(opts)
cleared = runner.run()

# Clear just the 'base' saltenv file list caches
opts['arg'] = ['base', None]
runner = salt.runner.Runner(opts)
cleared = runner.run()

# Clear just the 'base' saltenv file list caches from just the 'roots'
# fileserver backend
opts['arg'] = ['base', 'roots']
runner = salt.runner.Runner(opts)
cleared = runner.run()

# Clear all file list caches from the 'roots' fileserver backend
opts['arg'] = [None, 'roots']
runner = salt.runner.Runner(opts)
cleared = runner.run()
```

This function will return a dictionary showing a list of environments which were cleared for each backend. An empty return dictionary means that no changes were made.

CLI Examples:

```
# Clear all file list caches
salt-run fileserver.clear_file_list_cache
# Clear just the 'base' saltenv file list caches
salt-run fileserver.clear_file_list_cache saltenv=base
# Clear just the 'base' saltenv file list caches from just the 'roots'
# fileserver backend
salt-run fileserver.clear_file_list_cache saltenv=base backend=roots
# Clear all file list caches from the 'roots' fileserver backend
salt-run fileserver.clear_file_list_cache backend=roots
```

`salt.runners.fileserver.`**`clear_lock`**(*backend=None, remote=None*)

New in version 2015.5.0.

Clear the fileserver update lock from VCS fileserver backends (*git*, *hg*, *svn*). This should only need to be done if a fileserver update was interrupted and a remote is not updating (generating a warning in the Master's log file). Executing this runner with no arguments will remove all update locks from all enabled VCS fileserver backends, but this can be narrowed by using the following arguments:

backend Only clear the update lock for the specified backend(s).

remote If specified, then any remotes which contain the passed string will have their lock cleared. For example, a `remote` value of **github** will remove the lock from all github.com remotes.

CLI Example:

```
salt-run fileserver.clear_lock
salt-run fileserver.clear_lock backend=git,hg
salt-run fileserver.clear_lock backend=git remote=github
salt-run fileserver.clear_lock remote=bitbucket
```

`salt.runners.fileserver.`**`dir_list`**`(`*saltenv='base', backend=None*`)`

Return a list of directories in the given environment

saltenv [base] The salt fileserver environment to be listed

backend Narrow fileserver backends to a subset of the enabled ones. If all passed backends start with a minus sign (`-`), then these backends will be excluded from the enabled backends. However, if there is a mix of backends with and without a minus sign (ex: `backend=-roots,git`) then the ones starting with a minus sign will be disregarded.

New in version 2015.5.0.

CLI Example:

```
salt-run fileserver.dir_list
salt-run fileserver.dir_list saltenv=prod
salt-run fileserver.dir_list saltenv=dev backend=git
salt-run fileserver.dir_list base hg,roots
salt-run fileserver.dir_list -git
```

`salt.runners.fileserver.`**`empty_dir_list`**`(`*saltenv='base', backend=None*`)`

New in version 2015.5.0.

Return a list of empty directories in the given environment

saltenv [base] The salt fileserver environment to be listed

backend Narrow fileserver backends to a subset of the enabled ones. If all passed backends start with a minus sign (`-`), then these backends will be excluded from the enabled backends. However, if there is a mix of backends with and without a minus sign (ex: `backend=-roots,git`) then the ones starting with a minus sign will be disregarded.

Note: Some backends (such as *git* and *hg*) do not support empty directories. So, passing `backend=git` or `backend=hg` will result in an empty list being returned.

CLI Example:

```
salt-run fileserver.empty_dir_list
salt-run fileserver.empty_dir_list saltenv=prod
salt-run fileserver.empty_dir_list backend=roots
```

`salt.runners.fileserver.`**`envs`**`(`*backend=None, sources=False*`)`

Return the available fileserver environments. If no backend is provided, then the environments for all configured backends will be returned.

backend Narrow fileserver backends to a subset of the enabled ones.

Changed in version 2015.5.0: If all passed backends start with a minus sign (`-`), then these backends will be excluded from the enabled backends. However, if there is a mix of backends with and without a minus sign (ex: `backend=-roots,git`) then the ones starting with a minus sign will be disregarded.

Additionally, fileserver backends can now be passed as a comma-separated list. In earlier versions, they needed to be passed as a python list (ex: `backend="['roots', 'git']"`)

CLI Example:

```
salt-run fileserver.envs
salt-run fileserver.envs backend=roots,git
salt-run fileserver.envs git
```

`salt.runners.fileserver.`**`file_list`**`(`*saltenv='base', backend=None*`)`

Return a list of files from the salt fileserver

saltenv [base] The salt fileserver environment to be listed

backend Narrow fileserver backends to a subset of the enabled ones. If all passed backends start with a minus sign (`-`), then these backends will be excluded from the enabled backends. However, if there is a mix of backends with and without a minus sign (ex: `backend=-roots,git`) then the ones starting with a minus sign will be disregarded.

New in version 2015.5.0.

CLI Examples:

```
salt-run fileserver.file_list
salt-run fileserver.file_list saltenv=prod
salt-run fileserver.file_list saltenv=dev backend=git
salt-run fileserver.file_list base hg,roots
salt-run fileserver.file_list -git
```

`salt.runners.fileserver.`**`lock`**`(`*backend=None, remote=None*`)`

New in version 2015.5.0.

Set a fileserver update lock for VCS fileserver backends (*git*, *hg*, *svn*).

Note: This will only operate on enabled backends (those configured in *fileserver_backend*).

---

backend Only set the update lock for the specified backend(s).

remote If not None, then any remotes which contain the passed string will have their lock cleared. For example, a `remote` value of `*github.com*` will remove the lock from all github.com remotes.

CLI Example:

```
salt-run fileserver.lock
salt-run fileserver.lock backend=git,hg
salt-run fileserver.lock backend=git remote='*github.com*'
salt-run fileserver.lock remote=bitbucket
```

`salt.runners.fileserver.`**`symlink_list`**`(`*saltenv='base', backend=None*`)`

Return a list of symlinked files and dirs

saltenv [base] The salt fileserver environment to be listed

backend Narrow fileserver backends to a subset of the enabled ones. If all passed backends start with a minus sign (`-`), then these backends will be excluded from the enabled backends. However, if there is a mix of backends with and without a minus sign (ex: `backend=-roots,git`) then the ones starting with a minus sign will be disregarded.

New in version 2015.5.0.

CLI Example:

```
salt-run fileserver.symlink_list
salt-run fileserver.symlink_list saltenv=prod
salt-run fileserver.symlink_list saltenv=dev backend=git
salt-run fileserver.symlink_list base hg,roots
salt-run fileserver.symlink_list -git
```

`salt.runners.fileserver.`**`update`**`(`*backend=None*`)`

Update the fileserver cache. If no backend is provided, then the cache for all configured backends will be updated.

backend Narrow fileserver backends to a subset of the enabled ones.

> Changed in version 2015.5.0: If all passed backends start with a minus sign (`-`), then these backends will be excluded from the enabled backends. However, if there is a mix of backends with and without a minus sign (ex: `backend=-roots,git`) then the ones starting with a minus sign will be disregarded.
>
> Additionally, fileserver backends can now be passed as a comma-separated list. In earlier versions, they needed to be passed as a python list (ex: `backend="['roots', 'git']"`)

CLI Example:

```
salt-run fileserver.update
salt-run fileserver.update backend=roots,git
```

21.16.14 salt.runners.git_pillar

Runner module to directly manage the git external pillar

`salt.runners.git_pillar.`**`update`**`(`*branch=None*, *repo=None*`)`

New in version 2014.1.0.

Changed in version 2015.8.4: This runner function now supports the *git_pillar configuration schema* introduced in 2015.8.0. Additionally, the branch and repo can now be omitted to update all git_pillar remotes. The return data has also changed to a dictionary. The values will be `True` only if new commits were fetched, and `False` if there were errors or no new commits were fetched.

Changed in version 2018.3.0: The return for a given git_pillar remote will now be `None` when no changes were fetched. `False` now is reserved only for instances in which there were errors.

Fetch one or all configured git_pillar remotes.

Note: This will *not* fast-forward the git_pillar cachedir on the master. All it does is perform a `git fetch`. If this runner is executed with `-l debug`, you may see a log message that says that the repo is up-to-date. Keep in mind that Salt automatically fetches git_pillar repos roughly every 60 seconds (or whatever *`loop_interval`* is set to). So, it is possible that the repo was fetched automatically in the time between when changes were pushed to the repo, and when this runner was executed. When in doubt, simply refresh pillar data using *`saltutil.refresh_pillar`* and then use *`pillar.item`* to check if the pillar data has changed as expected.

CLI Example:

```
# Update specific branch and repo
salt-run git_pillar.update branch='branch' repo='https://foo.com/bar.git'
# Update all repos
salt-run git_pillar.update
# Run with debug logging
salt-run git_pillar.update -l debug
```

21.16.15 salt.runners.http

Module for making various web calls. Primarily designed for webhooks and the like, but also useful for basic http testing.

New in version 2015.5.0.

`salt.runners.http.`**`query`**(*url*, *output=True*, ***kwargs*)

Query a resource, and decode the return data

Passes through all the parameters described in the *`utils.http.query function`*:

CLI Example:

```
salt-run http.query http://somelink.com/
salt-run http.query http://somelink.com/ method=POST             params=
↪'key1=val1&key2=val2'
salt-run http.query http://somelink.com/ method=POST             data='<xml>
↪somecontent</xml>'
```

`salt.runners.http.`**`update_ca_bundle`**(*target=None*, *source=None*, *merge_files=None*)

Update the local CA bundle file from a URL

New in version 2015.5.0.

CLI Example:

```
salt-run http.update_ca_bundle
salt-run http.update_ca_bundle target=/path/to/cacerts.pem
salt-run http.update_ca_bundle source=https://example.com/cacerts.pem
```

If the `target` is not specified, it will be pulled from the `ca_cert` configuration variable available to the master. If it cannot be found there, it will be placed at `<<FILE_ROOTS>>/cacerts.pem`.

If the `source` is not specified, it will be pulled from the `ca_cert_url` configuration variable available to the master. If it cannot be found, it will be downloaded from the cURL website, using an http (not https) URL. USING THE DEFAULT URL SHOULD BE AVOIDED!

`merge_files` may also be specified, which includes a string or list of strings representing a file or files to be appended to the end of the CA bundle, once it is downloaded.

CLI Example:

```
salt-run http.update_ca_bundle merge_files=/path/to/mycert.pem
```

21.16.16 salt.runners.jobs

A convenience system to manage jobs, both active and already run

`salt.runners.jobs.`**`active`**(*display_progress=False*)

Return a report on all actively running jobs from a job id centric perspective

CLI Example:

```
salt-run jobs.active
```

`salt.runners.jobs.`**`exit_success`**(*jid*, *ext_source=None*)

Check if a job has been executed and exit successfully

jid The jid to look up.

ext_source The external job cache to use. Default: *None*.

CLI Example:

```
salt-run jobs.exit_success 20160520145827701627
```

salt.runners.jobs.**last_run**(*ext_source=None*, *outputter=None*, *metadata=None*, *function=None*, *target=None*, *display_progress=False*)

New in version 2015.8.0.

List all detectable jobs and associated functions

CLI Example:

```
salt-run jobs.last_run
salt-run jobs.last_run target=nodename
salt-run jobs.last_run function='cmd.run'
salt-run jobs.last_run metadata="{'foo': 'bar'}"
```

salt.runners.jobs.**list_job**(*jid*, *ext_source=None*, *display_progress=False*)

List a specific job given by its jid

ext_source If provided, specifies which external job cache to use.

display_progress [False] If `True`, fire progress events.

New in version 2015.8.8.

CLI Example:

```
salt-run jobs.list_job 20130916125524463507
salt-run jobs.list_job 20130916125524463507 --out=pprint
```

salt.runners.jobs.**list_jobs**(*ext_source=None*, *outputter=None*, *search_metadata=None*, *search_function=None*, *search_target=None*, *start_time=None*, *end_time=None*, *display_progress=False*)

List all detectable jobs and associated functions

ext_source If provided, specifies which external job cache to use.

FILTER OPTIONS

Note: If more than one of the below options are used, only jobs which match *all* of the filters will be returned.

search_metadata Specify a dictionary to match to the job's metadata. If any of the key-value pairs in this dictionary match, the job will be returned. Example:

```
salt-run jobs.list_jobs search_metadata='{"foo": "bar", "baz": "qux"}'
```

search_function Can be passed as a string or a list. Returns jobs which match the specified function. Globbing is allowed. Example:

```
salt-run jobs.list_jobs search_function='test.*'
salt-run jobs.list_jobs search_function='["test.*", "pkg.install"]'
```

Changed in version 2015.8.8: Multiple targets can now also be passed as a comma-separated list. For example:

```
salt-run jobs.list_jobs search_function='test.*,pkg.install'
```

search_target Can be passed as a string or a list. Returns jobs which match the specified minion name. Globbing is allowed. Example:

```
salt-run jobs.list_jobs search_target='*.mydomain.tld'
salt-run jobs.list_jobs search_target='["db*", "myminion"]'
```

Changed in version 2015.8.8: Multiple targets can now also be passed as a comma-separated list. For example:

```
salt-run jobs.list_jobs search_target='db*,myminion'
```

start_time Accepts any timestamp supported by the dateutil Python module (if this module is not installed, this argument will be ignored). Returns jobs which started after this timestamp.

end_time Accepts any timestamp supported by the dateutil Python module (if this module is not installed, this argument will be ignored). Returns jobs which started before this timestamp.

CLI Example:

```
salt-run jobs.list_jobs
salt-run jobs.list_jobs search_function='test.*' search_target='localhost' search_
↪metadata='{"bar": "foo"}'
salt-run jobs.list_jobs start_time='2015, Mar 16 19:00' end_time='2015, Mar 18
↪22:00'
```

`salt.runners.jobs.`**`list_jobs_filter`**(*count, filter_find_job=True, ext_source=None, outputter=None, display_progress=False*)

List all detectable jobs and associated functions

ext_source The external job cache to use. Default: *None*.

CLI Example:

```
salt-run jobs.list_jobs_filter 50
salt-run jobs.list_jobs_filter 100 filter_find_job=False
```

`salt.runners.jobs.`**`lookup_jid`**(*jid, ext_source=None, returned=True, missing=False, display_progress=False*)

Return the printout from a previously executed job

jid The jid to look up.

ext_source The external job cache to use. Default: *None*.

returned [True] If `True`, include the minions that did return from the command.

New in version 2015.8.0.

missing [False] If `True`, include the minions that did *not* return from the command.

display_progress [False] If `True`, fire progress events.

New in version 2015.5.0.

CLI Example:

```
salt-run jobs.lookup_jid 20130916125524463507
salt-run jobs.lookup_jid 20130916125524463507 --out=highstate
```

`salt.runners.jobs.`**`print_job`**(*jid, ext_source=None*)

Print a specific job's detail given by it's jid, including the return data.

CLI Example:

```
salt-run jobs.print_job 20130916125524463507
```

21.16.17 salt.runners.launchd

Manage launchd plist files

`salt.runners.launchd.`**`write_launchd_plist`**(*program*)

Write a launchd plist for managing salt-master or salt-minion

CLI Example:

```
salt-run launchd.write_launchd_plist salt-master
```

21.16.18 salt.runners.lxc

Control Linux Containers via Salt

depends lxc execution module

`salt.runners.lxc.`**`cloud_init`**`(`*names, host=None, quiet=False, **kwargs*`)`

Wrapper for using lxc.init in saltcloud compatibility mode

names Name of the containers, supports a single name or a comma delimited list of names.

host Minion to start the container on. Required.

path path to the container parent default: /var/lib/lxc (system default)

New in version 2015.8.0.

saltcloud_mode init the container with the saltcloud opts format instead

`salt.runners.lxc.`**`find_guest`**`(`*name, quiet=False, path=None*`)`

Returns the host for a container.

path path to the container parent default: /var/lib/lxc (system default)

New in version 2015.8.0.

```
salt-run lxc.find_guest name
```

`salt.runners.lxc.`**`find_guests`**`(`*names, path=None*`)`

Return a dict of hosts and named guests

path path to the container parent default: /var/lib/lxc (system default)

New in version 2015.8.0.

`salt.runners.lxc.`**`freeze`**`(`*name, quiet=False, path=None*`)`

Freeze the named container

path path to the container parent default: /var/lib/lxc (system default)

New in version 2015.8.0.

```
salt-run lxc.freeze name
```

`salt.runners.lxc.`**`info`**`(`*name, quiet=False, path=None*`)`

Returns information about a container.

path path to the container parent default: /var/lib/lxc (system default)

New in version 2015.8.0.

```
salt-run lxc.info name
```

`salt.runners.lxc.`**`init`**`(`*names, host=None, saltcloud_mode=False, quiet=False, **kwargs*`)`

Initialize a new container

```
salt-run lxc.init name host=minion_id [cpuset=cgroups_cpuset] \
        [cpushare=cgroups_cpushare] [memory=cgroups_memory] \
        [template=lxc_template_name] [clone=original name] \
        [profile=lxc_profile] [network_proflile=network_profile] \
        [nic=network_profile] [nic_opts=nic_opts] \
        [start=(true|false)] [seed=(true|false)] \
        [install=(true|false)] [config=minion_config] \
        [snapshot=(true|false)]
```

names Name of the containers, supports a single name or a comma delimited list of names.

host Minion on which to initialize the container **(required)**

path path to the container parent default: /var/lib/lxc (system default)

New in version 2015.8.0.

saltcloud_mode init the container with the saltcloud opts format instead See lxc.init_interface module documentation

cpuset cgroups cpuset.

cpushare cgroups cpu shares.

memory cgroups memory limit, in MB

Changed in version 2015.5.0: If no value is passed, no limit is set. In earlier Salt versions, not passing this value causes a 1024MB memory limit to be set, and it was necessary to pass `memory=0` to set no limit.

template Name of LXC template on which to base this container

clone Clone this container from an existing container

profile A LXC profile (defined in config or pillar).

network_profile Network profile to use for the container

New in version 2015.5.2.

nic Deprecated since version 2015.5.0: Use `network_profile` instead

nic_opts Extra options for network interfaces. E.g.:

`{"eth0": {"mac": "aa:bb:cc:dd:ee:ff", "ipv4": "10.1.1.1", "ipv6": "2001:db8::ff00:42:8329"}}`

start Start the newly created container.

seed Seed the container with the minion config and autosign its key. Default: true

install If salt-minion is not already installed, install it. Default: true

config Optional config parameters. By default, the id is set to the name of the container.

`salt.runners.lxc.`**`list_`**(*host=None*, *quiet=False*, *path=None*)

List defined containers (running, stopped, and frozen) for the named (or all) host(s).

path path to the container parent default: /var/lib/lxc (system default)

New in version 2015.8.0.

```
salt-run lxc.list [host=minion_id]
```

`salt.runners.lxc.`**`purge`**(*name*, *delete_key=True*, *quiet=False*, *path=None*)

Purge the named container and delete its minion key if present. WARNING: Destroys all data associated with the container.

path path to the container parent default: /var/lib/lxc (system default)

New in version 2015.8.0.

```
salt-run lxc.purge name
```

`salt.runners.lxc.`**`start`**(*name*, *quiet=False*, *path=None*)

Start the named container.

path path to the container parent default: /var/lib/lxc (system default)

New in version 2015.8.0.

```
salt-run lxc.start name
```

`salt.runners.lxc.`**`stop`**(*name*, *quiet=False*, *path=None*)

Stop the named container.

path path to the container parent default: /var/lib/lxc (system default)

New in version 2015.8.0.

```
salt-run lxc.stop name
```

`salt.runners.lxc.`**`unfreeze`**`(`*name, quiet=False, path=None*`)`

Unfreeze the named container

path path to the container parent default: /var/lib/lxc (system default)

New in version 2015.8.0.

```
salt-run lxc.unfreeze name
```

21.16.19 salt.runners.manage

General management functions for salt, tools like seeing what hosts are up and what hosts are down

`salt.runners.manage.`**`alived`**`(`*subset=None, show_ip=False, show_ipv4=None*`)`

New in version 2015.8.0.

Changed in version 2019.2.0: The 'show_ipv4' argument has been renamed to 'show_ip' as it now includes IPv6 addresses for IPv6-connected minions.

Print a list of all minions that are up according to Salt's presence detection (no commands will be sent to minions)

subset [None] Pass in a CIDR range to filter minions by IP address.

show_ip [False] Also show the IP address each minion is connecting from.

CLI Example:

```
salt-run manage.alived
```

`salt.runners.manage.`**`allowed`**`(`*subset=None, show_ip=False, show_ipv4=None*`)`

New in version 2015.8.0.

Changed in version 2019.2.0: The 'show_ipv4' argument has been renamed to 'show_ip' as it now includes IPv6 addresses for IPv6-connected minions.

Print a list of all minions that are up according to Salt's presence detection (no commands will be sent to minions)

subset [None] Pass in a CIDR range to filter minions by IP address.

show_ip [False] Also show the IP address each minion is connecting from.

CLI Example:

```
salt-run manage.allowed
```

`salt.runners.manage.`**`bootstrap`**`(`*version='develop', script=None, hosts='', script_args='', roster='flat', ssh_user=None, ssh_password=None, ssh_priv_key=None, tmp_dir='/tmp/.bootstrap', http_backend='tornado'*`)`

Bootstrap minions with salt-bootstrap

version [develop] Git tag of version to install

script [https://bootstrap.saltstack.com] URL containing the script to execute

hosts Comma-separated hosts [example: hosts='host1.local,host2.local']. These hosts need to exist in the specified roster.

script_args Any additional arguments that you want to pass to the script.

New in version 2016.11.0.

roster [flat] The roster to use for Salt SSH. More information about roster files can be found in *Salt's Roster Documentation*.

A full list of roster types, see the *builtin roster modules* documentation.

New in version 2016.11.0.

ssh_user If `user` isn't found in the `roster`, a default SSH user can be set here. Keep in mind that `ssh_user` will not override the roster `user` value if it is already defined.

New in version 2016.11.0.

ssh_password If `passwd` isn't found in the `roster`, a default SSH password can be set here. Keep in mind that `ssh_password` will not override the roster `passwd` value if it is already defined.

New in version 2016.11.0.

ssh_privkey If `priv` isn't found in the `roster`, a default SSH private key can be set here. Keep in mind that `ssh_password` will not override the roster `passwd` value if it is already defined.

New in version 2016.11.0.

tmp_dir [/tmp/.bootstrap] The temporary directory to download the bootstrap script in. This directory will have `-<uuid4>` appended to it. For example: `/tmp/.bootstrap-a19a728e-d40a-4801-aba9-d00655c143a7/`

New in version 2016.11.0.

http_backend [tornado] The backend library to use to download the script. If you need to use a `file:///` URL, then you should set this to `urllib2`.

New in version 2016.11.0.

CLI Example:

```
salt-run manage.bootstrap hosts='host1,host2'
salt-run manage.bootstrap hosts='host1,host2' version='v0.17'
salt-run manage.bootstrap hosts='host1,host2' version='v0.17'          script=
↪'https://bootstrap.saltstack.com/develop'
```

`salt.runners.manage.`**`bootstrap_psexec`**(*hosts='', master=None, version=None, arch='win32', installer_url=None, username=None, password=None*)

Bootstrap Windows minions via PsExec.

hosts Comma separated list of hosts to deploy the Windows Salt minion.

master Address of the Salt master passed as an argument to the installer.

version Point release of installer to download. Defaults to the most recent.

arch Architecture of installer to download. Defaults to win32.

installer_url URL of minion installer executable. Defaults to the latest version from https://repo.saltstack.com/windows/

username Optional user name for login on remote computer.

password Password for optional username. If omitted, PsExec will prompt for one to be entered for each host.

CLI Example:

```
salt-run manage.bootstrap_psexec hosts='host1,host2'
salt-run manage.bootstrap_psexec hosts='host1,host2' version='0.17' username=
↪'DOMAIN\Administrator'
salt-run manage.bootstrap_psexec hosts='host1,host2' installer_url='http://
↪exampledomain/salt-installer.exe'
```

`salt.runners.manage.`**`down`**(*removekeys=False, tgt='*', tgt_type='glob', timeout=None, gather_job_timeout=None*)

Changed in version 2017.7.0: The `expr_form` argument has been renamed to `tgt_type`, earlier releases must use `expr_form`.

Print a list of all the down or unresponsive salt minions Optionally remove keys of down minions

CLI Example:

```
salt-run manage.down
salt-run manage.down removekeys=True
salt-run manage.down tgt="webservers" tgt_type="nodegroup"
```

`salt.runners.manage.`**`joined`**`(`*subset=None*, *show_ip=False*, *show_ipv4=None*`)`

New in version 2015.8.0.

Changed in version 2019.2.0: The 'show_ipv4' argument has been renamed to 'show_ip' as it now includes IPv6 addresses for IPv6-connected minions.

Print a list of all minions that are up according to Salt's presence detection (no commands will be sent to minions)

subset [None] Pass in a CIDR range to filter minions by IP address.

show_ip [False] Also show the IP address each minion is connecting from.

CLI Example:

```
salt-run manage.joined
```

`salt.runners.manage.`**`key_regen`**`()`

This routine is used to regenerate all keys in an environment. This is invasive! ALL KEYS IN THE SALT ENVIRONMENT WILL BE REGENERATED!!

The key_regen routine sends a command out to minions to revoke the master key and remove all minion keys, it then removes all keys from the master and prompts the user to restart the master. The minions will all reconnect and keys will be placed in pending.

After the master is restarted and minion keys are in the pending directory execute a salt-key -A command to accept the regenerated minion keys.

The master *must* be restarted within 60 seconds of running this command or the minions will think there is something wrong with the keys and abort.

Only Execute this runner after upgrading minions and master to 0.15.1 or higher!

CLI Example:

```
salt-run manage.key_regen
```

`salt.runners.manage.`**`list_not_state`**`(`*subset=None*, *show_ip=False*, *show_ipv4=None*`)`

New in version 2015.8.0.

Changed in version 2019.2.0: The 'show_ipv4' argument has been renamed to 'show_ip' as it now includes IPv6 addresses for IPv6-connected minions.

Print a list of all minions that are NOT up according to Salt's presence detection (no commands will be sent to minions)

subset [None] Pass in a CIDR range to filter minions by IP address.

show_ip [False] Also show the IP address each minion is connecting from.

CLI Example:

```
salt-run manage.list_not_state
```

`salt.runners.manage.`**`list_state`**`(`*subset=None*, *show_ip=False*, *show_ipv4=None*`)`

New in version 2015.8.0.

Changed in version 2019.2.0: The 'show_ipv4' argument has been renamed to 'show_ip' as it now includes IPv6 addresses for IPv6-connected minions.

Print a list of all minions that are up according to Salt's presence detection (no commands will be sent to minions)

subset [None] Pass in a CIDR range to filter minions by IP address.
show_ip [False] Also show the IP address each minion is connecting from.
CLI Example:

```
salt-run manage.list_state
```

`salt.runners.manage.`**`not_alived`**`(`*subset=None*, *show_ip=False*, *show_ipv4=None*`)`
New in version 2015.8.0.

Changed in version 2019.2.0: The 'show_ipv4' argument has been renamed to 'show_ip' as it now includes IPv6 addresses for IPv6-connected minions.

Print a list of all minions that are NOT up according to Salt's presence detection (no commands will be sent)
subset [None] Pass in a CIDR range to filter minions by IP address.
show_ip [False] Also show the IP address each minion is connecting from.
CLI Example:

```
salt-run manage.not_alived
```

`salt.runners.manage.`**`not_allowed`**`(`*subset=None*, *show_ip=False*, *show_ipv4=None*`)`
New in version 2015.8.0.

Changed in version 2019.2.0: The 'show_ipv4' argument has been renamed to 'show_ip' as it now includes IPv6 addresses for IPv6-connected minions.

Print a list of all minions that are NOT up according to Salt's presence detection (no commands will be sent)
subset [None] Pass in a CIDR range to filter minions by IP address.
show_ip [False] Also show the IP address each minion is connecting from.
CLI Example:

```
salt-run manage.not_allowed
```

`salt.runners.manage.`**`not_joined`**`(`*subset=None*, *show_ip=False*, *show_ipv4=None*`)`
New in version 2015.8.0.

Changed in version 2019.2.0: The 'show_ipv4' argument has been renamed to 'show_ip' as it now includes IPv6 addresses for IPv6-connected minions.

Print a list of all minions that are NOT up according to Salt's presence detection (no commands will be sent)
subset [None] Pass in a CIDR range to filter minions by IP address.
show_ip [False] Also show the IP address each minion is connecting from.
CLI Example:

```
salt-run manage.not_joined
```

`salt.runners.manage.`**`not_present`**`(`*subset=None*, *show_ip=False*, *show_ipv4=None*`)`
New in version 2015.5.0.

Changed in version 2019.2.0: The 'show_ipv4' argument has been renamed to 'show_ip' as it now includes IPv6 addresses for IPv6-connected minions.

Print a list of all minions that are NOT up according to Salt's presence detection (no commands will be sent)
subset [None] Pass in a CIDR range to filter minions by IP address.
show_ip [False] Also show the IP address each minion is connecting from.
CLI Example:

```
salt-run manage.not_present
```

`salt.runners.manage.`**`not_reaped`**(*subset=None*, *show_ip=False*, *show_ipv4=None*)

New in version 2015.8.0.

Changed in version 2019.2.0: The 'show_ipv4' argument has been renamed to 'show_ip' as it now includes IPv6 addresses for IPv6-connected minions.

Print a list of all minions that are NOT up according to Salt's presence detection (no commands will be sent)

subset [None] Pass in a CIDR range to filter minions by IP address.

show_ip [False] Also show the IP address each minion is connecting from.

CLI Example:

```
salt-run manage.not_reaped
```

`salt.runners.manage.`**`present`**(*subset=None*, *show_ip=False*, *show_ipv4=None*)

Changed in version 2019.2.0: The 'show_ipv4' argument has been renamed to 'show_ip' as it now includes IPv6 addresses for IPv6-connected minions.

Print a list of all minions that are up according to Salt's presence detection (no commands will be sent to minions)

subset [None] Pass in a CIDR range to filter minions by IP address.

show_ip [False] Also show the IP address each minion is connecting from.

CLI Example:

```
salt-run manage.present
```

`salt.runners.manage.`**`reaped`**(*subset=None*, *show_ip=False*, *show_ipv4=None*)

New in version 2015.8.0.

Changed in version 2019.2.0: The 'show_ipv4' argument has been renamed to 'show_ip' as it now includes IPv6 addresses for IPv6-connected minions.

Print a list of all minions that are up according to Salt's presence detection (no commands will be sent to minions)

subset [None] Pass in a CIDR range to filter minions by IP address.

show_ip [False] Also show the IP address each minion is connecting from.

CLI Example:

```
salt-run manage.reaped
```

`salt.runners.manage.`**`safe_accept`**(*target*, *tgt_type='glob'*)

Changed in version 2017.7.0: The `expr_form` argument has been renamed to `tgt_type`, earlier releases must use `expr_form`.

Accept a minion's public key after checking the fingerprint over salt-ssh

CLI Example:

```
salt-run manage.safe_accept my_minion
salt-run manage.safe_accept minion1,minion2 tgt_type=list
```

`salt.runners.manage.`**`status`**(*output=True*, *tgt='*'*, *tgt_type='glob'*, *timeout=None*, *gather_job_timeout=None*)

Changed in version 2017.7.0: The `expr_form` argument has been renamed to `tgt_type`, earlier releases must use `expr_form`.

Print the status of all known salt minions

CLI Example:

```
salt-run manage.status
salt-run manage.status tgt="webservers" tgt_type="nodegroup"
salt-run manage.status timeout=5 gather_job_timeout=10
```

`salt.runners.manage.`**`up`**`(`*tgt='*', tgt_type='glob', timeout=None, gather_job_timeout=None*`)`

Changed in version 2017.7.0: The `expr_form` argument has been renamed to `tgt_type`, earlier releases must use `expr_form`.

Print a list of all of the minions that are up

CLI Example:

```
salt-run manage.up
salt-run manage.up tgt="webservers" tgt_type="nodegroup"
salt-run manage.up timeout=5 gather_job_timeout=10
```

`salt.runners.manage.`**`versions`**`()`

Check the version of active minions

CLI Example:

```
salt-run manage.versions
```

21.16.20 salt.runners.mattermost

Note for 2017.7 releases!

Due to the salt.runners.config module not being available in this release series, importing the salt.runners.config module from the master branch is required to make this module work.

Ref: Mattermost runner failing to retrieve config values due to unavailable config runner #43479

Module for sending messages to Mattermost

New in version 2017.7.0.

configuration This module can be used by either passing an api_url and hook directly or by specifying both in a configuration profile in the salt master/minion config. For example:

```
mattermost:
  hook: peWcBiMOS9HrZG15peWcBiMOS9HrZG15
  api_url: https://example.com
```

`salt.runners.mattermost.`**`post_event`**`(`*event, channel=None, username=None, api_url=None, hook=None*`)`

Send an event to a Mattermost channel. :param channel: The channel name, either will work. :param username: The username of the poster. :param event: The event to send to the Mattermost channel. :param api_url: The Mattermost api url, if not specified in the configuration. :param hook: The Mattermost hook, if not specified in the configuration. :return: Boolean if message was sent successfully.

`salt.runners.mattermost.`**`post_message`**`(`*message, channel=None, username=None, api_url=None, hook=None*`)`

Send a message to a Mattermost channel. :param channel: The channel name, either will work. :param username: The username of the poster. :param message: The message to send to the Mattermost channel. :param api_url: The Mattermost api url, if not specified in the configuration. :param hook: The Mattermost hook, if not specified in the configuration. :return: Boolean if message was sent successfully.

CLI Example:

```
salt-run mattermost.post_message message='Build is done'
```

21.16.21 salt.runners.mine

A runner to access data from the salt mine

`salt.runners.mine.`**`get`**`(`*tgt, fun, tgt_type='glob'*`)`

Gathers the data from the specified minions' mine, pass in the target, function to look up and the target type

CLI Example:

```
salt-run mine.get '*' network.interfaces
```

`salt.runners.mine.`**`update`**`(`*tgt, tgt_type='glob', clear=False, mine_functions=None*`)`

New in version 2017.7.0.

Update the mine data on a certain group of minions.

tgt Which minions to target for the execution.

tgt_type: **`glob`** The type of `tgt`.

clear: **`False`** Boolean flag specifying whether updating will clear the existing mines, or will update. Default: `False` (update).

mine_functions Update the mine data on certain functions only. This feature can be used when updating the mine for functions that require refresh at different intervals than the rest of the functions specified under `mine_functions` in the minion/master config or pillar.

CLI Example:

```
salt-run mine.update '*'
salt-run mine.update 'juniper-edges' tgt_type='nodegroup'
```

21.16.22 salt.runners.nacl

This module helps include encrypted passwords in pillars, grains and salt state files.

depends libnacl, https://github.com/saltstack/libnacl

This is often useful if you wish to store your pillars in source control or share your pillar data with others that you trust. I don't advise making your pillars public regardless if they are encrypted or not.

configuration The following configuration defaults can be define (pillar or config files) Avoid storing private keys in pillars! Ensure master does not have *pillar_opts=True*:

```
# cat /etc/salt/master.d/nacl.conf
nacl.config:
    # NOTE: `key` and `key_file` have been renamed to `sk`, `sk_file`
    # also `box_type` default changed from secretbox to sealedbox.
    box_type: sealedbox                     (default)
    sk_file: /etc/salt/pki/master/nacl      (default)
    pk_file: /etc/salt/pki/master/nacl.pub  (default)
    sk: None
    pk: None
```

Usage can override the config defaults:

```
salt-run nacl.enc sk_file=/etc/salt/pki/master/nacl pk_file=/etc/salt/
↪pki/master/nacl.pub
```

The nacl lib uses 32byte keys, these keys are base64 encoded to make your life more simple. To generate your *sk_file* and *pk_file* use:

```
salt-run nacl.keygen sk_file=/etc/salt/pki/master/nacl
# or if you want to work without files.
salt-run nacl.keygen
local:
    ----------
    pk:
        /kfGX7PbWeu099702PBbKWLpG/9p06IQRswkdWHCDk0=
    sk:
        SVWut5SqNpuPeNzb1b9y6b2eXg2PLIog43GBzp48Sow=
```

Now with your keypair, you can encrypt data:

You have two option, *sealedbox* or *secretbox*.

SecretBox is data encrypted using private key *pk*. Sealedbox is encrypted using public key *pk*.

Recommend using Sealedbox because the one way encryption permits developers to encrypt data for source control but not decrypt. Sealedbox only has one key that is for both encryption and decryption.

```
salt-run nacl.enc asecretpass pk=/kfGX7PbWeu099702PBbKWLpG/9p06IQRswkdWHCDk0=
tqXzeIJnTAM9Xf0mdLcpEdklMbfBGPj2oTKmlgrm3S1DTVVHNnh9h8mU1GKllGq/+cYsk6m5WhGdk58=
```

To decrypt the data:

```
salt-run nacl.dec data=
↪'tqXzeIJnTAM9Xf0mdLcpEdklMbfBGPj2oTKmlgrm3S1DTVVHNnh9h8mU1GKllGq/+cYsk6m5WhGdk58='
↪        sk='SVWut5SqNpuPeNzb1b9y6b2eXg2PLIog43GBzp48Sow='
```

When the keys are defined in the master config you can use them from the nacl runner without extra parameters:

```
# cat /etc/salt/master.d/nacl.conf
nacl.config:
    sk_file: /etc/salt/pki/master/nacl
    pk: 'cTIqXwnUiD1ulg4kXsbeCE7/NoeKEzd4nLeYcCFpd9k='
```

```
salt-run nacl.enc 'asecretpass'
salt-run nacl.dec 'tqXzeIJnTAM9Xf0mdLcpEdklMbfBGPj2oTKmlgrm3S1DTVVHNnh9h8mU1GKllGq/
↪+cYsk6m5WhGdk58='
```

```
# a salt developers minion could have pillar data that includes a nacl public key
nacl.config:
    pk: '/kfGX7PbWeu099702PBbKWLpG/9p06IQRswkdWHCDk0='
```

The developer can then use a less-secure system to encrypt data.

```
salt-run nacl.enc apassword
```

Pillar files can include protected data that the salt master decrypts:

```
pillarexample:
    user: root
    password1: {{salt.nacl.dec('DRB7Q6/X5gGSRCTpZyxS6hlbWj0llUA+uaVyvou3vJ4=')|json}}
    cert_key: {{salt.nacl.dec_file('/srv/salt/certs/example.com/key.nacl')|json}}
    cert_key2: {{salt.nacl.dec_file('salt:///certs/example.com/key.nacl')|json}}
```

Larger files like certificates can be encrypted with:

```
salt-run nacl.enc_file /tmp/cert.crt out=/tmp/cert.nacl
```

`salt.runners.nacl.`**`dec`**(*data*, ***kwargs*)

Alias to *{box_type}_decrypt*

box_type: secretbox, sealedbox(default)

`salt.runners.nacl.`**`dec_file`**(*name*, *out=None*, ***kwargs*)

This is a helper function to decrypt a file and return its contents.

You can provide an optional output file using *out*

name can be a local file or when not using *salt-run* can be a url like *salt://*, *https://* etc.

CLI Examples:

```
salt-run nacl.dec_file name=/tmp/id_rsa.nacl
salt-run nacl.dec_file name=/tmp/id_rsa.nacl box_type=secretbox             sk_
↪file=/etc/salt/pki/master/nacl.pub
```

`salt.runners.nacl.`**`enc`**(*data*, ***kwargs*)

Alias to *{box_type}_encrypt*

box_type: secretbox, sealedbox(default)

`salt.runners.nacl.`**`enc_file`**(*name*, *out=None*, ***kwargs*)

This is a helper function to encrypt a file and return its contents.

You can provide an optional output file using *out*

name can be a local file or when not using *salt-run* can be a url like *salt://*, *https://* etc.

CLI Examples:

```
salt-run nacl.enc_file name=/tmp/id_rsa
salt-run nacl.enc_file name=/tmp/id_rsa box_type=secretbox             sk_file=/
↪etc/salt/pki/master/nacl.pub
```

`salt.runners.nacl.`**`keygen`**(*sk_file=None*, *pk_file=None*, ***kwargs*)

Use libnacl to generate a keypair.

If no *sk_file* is defined return a keypair.

If only the *sk_file* is defined *pk_file* will use the same name with a postfix *.pub*.

When the *sk_file* is already existing, but *pk_file* is not. The *pk_file* will be generated using the *sk_file*.

CLI Examples:

```
salt-run nacl.keygen
salt-run nacl.keygen sk_file=/etc/salt/pki/master/nacl
salt-run nacl.keygen sk_file=/etc/salt/pki/master/nacl pk_file=/etc/salt/pki/
↪master/nacl.pub
salt-run nacl.keygen
```

`salt.runners.nacl.`**`sealedbox_decrypt`**(*data*, ***kwargs*)

Decrypt data using a secret key that was encrypted using a public key with *nacl.sealedbox_encrypt*.

CLI Examples:

```
salt-run nacl.sealedbox_decrypt pEXHQM6cuaF7A=
salt-run nacl.sealedbox_decrypt data='pEXHQM6cuaF7A=' sk_file=/etc/salt/pki/
↪master/nacl
salt-run nacl.sealedbox_decrypt data='pEXHQM6cuaF7A=' sk='YmFkcGFzcwo='
```

`salt.runners.nacl.`**`sealedbox_encrypt`**(*data*, ***kwargs*)
Encrypt data using a public key generated from *nacl.keygen*. The encryptd data can be decrypted using *nacl.sealedbox_decrypt* only with the secret key.

CLI Examples:

```
salt-run nacl.sealedbox_encrypt datatoenc
```

`salt.runners.nacl.`**`secretbox_decrypt`**(*data*, ***kwargs*)
Decrypt data that was encrypted using *nacl.secretbox_encrypt* using the secret key that was generated from *nacl.keygen*.

CLI Examples:

```
salt-run nacl.secretbox_decrypt pEXHQM6cuaF7A=
salt-run nacl.secretbox_decrypt data='pEXHQM6cuaF7A=' sk_file=/etc/salt/pki/
↪master/nacl
salt-run nacl.secretbox_decrypt data='pEXHQM6cuaF7A=' sk='YmFkcGFzcwo='
```

`salt.runners.nacl.`**`secretbox_encrypt`**(*data*, ***kwargs*)
Encrypt data using a secret key generated from *nacl.keygen*. The same secret key can be used to decrypt the data using *nacl.secretbox_decrypt*.

CLI Examples:

```
salt-run nacl.secretbox_encrypt datatoenc
salt-run nacl.secretbox_encrypt datatoenc sk_file=/etc/salt/pki/master/nacl
salt-run nacl.secretbox_encrypt datatoenc sk='YmFkcGFzcwo='
```

21.16.23 salt.runners.net

NET Finder

New in version 2017.7.0.

A runner to find network details easily and fast. It's smart enough to know what you are looking for.

Configuration

- Minion (proxy) config

  To have the complete features, one needs to add the following mine configuration in the minion (proxy) config file:

  ```
  mine_functions:
    net.ipaddrs: []
    net.lldp: []
    net.mac: []
    net.arp: []
    net.interfaces: []
  ```

Which instructs Salt to cache the data returned by the NAPALM-functions. While they are not mandatory, the less functions configured, the less details will be found by the runner.

How often the mines are refreshed, can be specified using:

```
mine_interval: <X minutes>
```

- Master config

  By default the following options can be configured on the master. They are not necessary, but available in case the user has different requirements.

  target: `*` From what minions will collect the mine data. Default: `*` (collect from all minions).

  expr_form: `glob` Minion matching expression form. Default: `glob`.

  ignore_interfaces A list of interfaces name to ignore. By default will consider all interfaces.

  display: `True` Display on the screen or return structured object? Default: `True` (return on the CLI).

  outputter: `table` Specify the outputter name when displaying on the CLI. Default: `table`.

  Configuration example:

```
runners:
  net.find:
    target: 'edge*'
    expr_form: 'glob'
    ignore_interfaces:
      - lo0
      - em1
      - jsrv
      - fxp0
    outputter: yaml
```

`salt.runners.net.find`(*addr*, *best=True*, *display=True*)

Search in all possible entities (Interfaces, MAC tables, ARP tables, LLDP neighbors), using the following mine functions:

- net.mac
- net.arp
- net.lldp
- net.ipaddrs
- net.interfaces

This function has the advantage that it knows where to look, but the output might become quite long as returns all possible matches.

Optional arguments:

best: `True` Return only the best match with the interfaces IP networks when the saerching pattern is a valid IP Address or Network.

display: `True` Display on the screen or return structured object? Default: `True` (return on the CLI).

CLI Example:

```
$ sudo salt-run net.find 10.10.10.7
```

Output Example:

```
Details for all interfaces that include network 10.10.10.7/32 - only best match
↪returned
_________________________________________________________________________________
↪______________________________________
```

(continues on next page)

(continued from previous page)

```
|    Device    | Interface | Interface Description |  UP  | Enabled | Speed↩
↪[Mbps] |    MAC Address    |  IP Addresses |
----------------------------------------------------------------------------------
↪--------------------------------------
| edge01.flw01 |    irb    |                       | True |   True  |      -1    ↩
↪ | 5C:5E:AB:AC:52:B4 | 10.10.10.1/22 |
----------------------------------------------------------------------------------
↪--------------------------------------

ARP Entries for IP 10.10.10.7
_____________________________________________________________________________
|    Device    |      Interface      |        MAC        |     IP     |  Age  |
_____________________________________________________________________________
| edge01.flw01 | irb.349 [ae0.349] | 2C:60:0C:2A:4C:0A | 10.10.10.7 | 832.0 |
_____________________________________________________________________________
```

salt.runners.net.**findarp**(*device=None, interface=None, mac=None, ip=None, display=True*)

Search for entries in the ARP tables using the following mine functions:

- net.arp

Optional arguments:

device Return interface data from a certain device only.

interface Return data selecting by interface name.

mac Search using a specific MAC Address.

ip Search using a specific IP Address.

display: **`True`** Display on the screen or return structured object? Default: `True`, will return on the CLI.

CLI Example:

```
$ sudo salt-run net.findarp mac=8C:60:0F:78:EC:41
```

Output Example:

```
ARP Entries for MAC 8C:60:0F:78:EC:41
________________________________________________________________________________
|    Device    |      Interface      |        MAC        |       IP      |  Age  |
________________________________________________________________________________
| edge01.bjm01 | irb.171 [ae0.171] | 8C:60:0F:78:EC:41 | 172.172.17.19 | 956.0 |
________________________________________________________________________________
```

salt.runners.net.**findmac**(*device=None, mac=None, interface=None, vlan=None, display=True*)

Search in the MAC Address tables, using the following mine functions:

- net.mac

Optional arguments:

device Return interface data from a certain device only.

interface Return data selecting by interface name.

mac Search using a specific MAC Address.

vlan Search using a VLAN ID.

display: **`True`** Display on the screen or return structured object? Default: `True`, will return on the CLI.

CLI Example:

```
$ sudo salt-run net.findmac mac=8C:60:0F:78:EC:41
```

Output Example:

```
MAC Address(es)
__________________________________________________________________________________
↪___________
```

(continues on next page)

(continued from previous page)

```
|    Device     | Interface |         MAC         | VLAN | Static | Active | Moves |☒
↪Last move |
----------------------------------------------------------------------------------
↪___________
| edge01.bjm01 |  ae0.171  | 8C:60:0F:78:EC:41 | 171  | False  |  True  |   0   |☒
↪   0.0    |
----------------------------------------------------------------------------------
↪___________
```

salt.runners.net.**interfaces**(*device=None*, *interface=None*, *title=None*, *pattern=None*, *ipnet=None*, *best=True*, *display=True*)

Search for interfaces details in the following mine functions:

- net.interfaces
- net.ipaddrs

Optional arguments:

device Return interface data from a certain device only.

interface Return data selecting by interface name.

pattern Return interfaces that contain a certain pattern in their description.

ipnet Return interfaces whose IP networks associated include this IP network.

best: `True` When `ipnet` is specified, this argument says if the runner should return only the best match (the output will contain at most one row). Default: `True` (return only the best match).

display: **True** Display on the screen or return structured object? Default: `True` (return on the CLI).

title Display a custom title for the table.

CLI Example:

```
$ sudo salt-run net.interfaces interface=vt-0/0/10
```

Output Example:

```
Details for interface xe-0/0/0
----------------------------------------------------------------------------------
↪_______________________________
|    Device     | Interface | Interface Description |  UP  | Enabled | Speed☒
↪[Mbps] | MAC Address | IP Addresses |
----------------------------------------------------------------------------------
↪_______________________________
| edge01.bjm01 | vt-0/0/10 |                       | True |   True  |     1000    ☒
↪ |             |              |
----------------------------------------------------------------------------------
↪_______________________________
| edge01.flw01 | vt-0/0/10 |                       | True |   True  |     1000    ☒
↪ |             |              |
----------------------------------------------------------------------------------
↪_______________________________
| edge01.pos01 | vt-0/0/10 |                       | True |   True  |     1000    ☒
↪ |             |              |
----------------------------------------------------------------------------------
↪_______________________________
| edge01.oua01 | vt-0/0/10 |                       | True |   True  |     1000    ☒
↪ |             |              |
----------------------------------------------------------------------------------
↪_______________________________
```

salt.runners.net.**lldp**(*device=None*, *interface=None*, *title=None*, *pattern=None*, *chassis=None*, *display=True*)

Search in the LLDP neighbors, using the following mine functions:

- net.lldp

Optional arguments:

device Return interface data from a certain device only.

interface Return data selecting by interface name.

pattern Return LLDP neighbors that have contain this pattern in one of the following fields:

- Remote Port ID
- Remote Port Description
- Remote System Name
- Remote System Description

chassis Search using a specific Chassis ID.

display: `True` Display on the screen or return structured object? Default: `True` (return on the CLI).

display: `True` Display on the screen or return structured object? Default: `True` (return on the CLI).

title Display a custom title for the table.

CLI Example:

```
$ sudo salt-run net.lldp pattern=Ethernet1/48
```

Output Example:

```
Pattern "Ethernet1/48" found in one of the following LLDP details
---------------------------------------------------------------------------------
↪-------------------------------------------------------------------------------
↪-----------------------
|    Device    | Interface | Parent Interface | Remote Chassis ID | Remote Port↩
↪ID | Remote Port Description |   Remote System Name   |            Remote↩
↪System Description            |
---------------------------------------------------------------------------------
↪-------------------------------------------------------------------------------
↪-----------------------
| edge01.bjm01 |  xe-2/3/4 |       ae0        | 8C:60:4F:3B:52:19 |             ↩
↪ |       Ethernet1/48      | edge05.bjm01.dummy.net |   Cisco NX-OS(tm) n6000,↩
↪Software (n6000-uk9),  |
|              |           |                  |                   |             ↩
↪ |                         |                        | Version 7.3(0)N7(5),↩
↪RELEASE SOFTWARE Copyright |
|              |           |                  |                   |             ↩
↪ |                         |                        |  (c) 2002-2012 by Cisco↩
↪Systems, Inc. Compiled  |
|              |           |                  |                   |             ↩
↪ |                         |                        |                2/17/2016↩
↪22:00:00               |
---------------------------------------------------------------------------------
↪-------------------------------------------------------------------------------
↪-----------------------
| edge01.flw01 |  xe-1/2/3 |       ae0        | 8C:60:4F:1A:B4:22 |             ↩
↪ |       Ethernet1/48      | edge05.flw01.dummy.net |   Cisco NX-OS(tm) n6000,↩
↪Software (n6000-uk9),  |
|              |           |                  |                   |             ↩
↪ |                         |                        | Version 7.3(0)N7(5),↩
↪RELEASE SOFTWARE Copyright |
|              |           |                  |                   |             ↩
↪ |                         |                        |  (c) 2002-2012 by Cisco↩
↪Systems, Inc. Compiled  |
|              |           |                  |                   |             ↩
↪ |                         |                        |                2/17/2016↩
↪22:00:00               |
---------------------------------------------------------------------------------
↪-------------------------------------------------------------------------------
↪-----------------------
```

(continues on next page)

(continued from previous page)

```
| edge01.oua01 |  xe-0/1/2 |       ae1        | 8C:60:4F:51:A4:22 |
↪ |        Ethernet1/48      | edge05.oua01.dummy.net |   Cisco NX-OS(tm) n6000,
↪Software (n6000-uk9),  |
|              |           |                  |                   |
↪ |                         |                        | Version 7.3(0)N7(5),
↪RELEASE SOFTWARE Copyright |
|              |           |                  |                   |
↪ |                         |                        |  (c) 2002-2012 by Cisco
↪Systems, Inc. Compiled   |
|              |           |                  |                   |
↪ |                         |                        |              2/17/2016
↪22:00:00               |
----------------------------------------------------------------------------------
↪--------------------------------------------------------------------------------
↪-----------------------
```

salt.runners.net.**multi_find**(**patterns*, ***kwargs*)

Execute multiple search tasks. This function is based on the *find* function. Depending on the search items, some information might overlap.

Optional arguments:

best: `True` Return only the best match with the interfaces IP networks when the saerching pattern is a valid IP Address or Network.

display: `True` Display on the screen or return structured object? Default: *True* (return on the CLI).

CLI Example:

```
$ sudo salt-run net.multi_find Ethernet1/49 xe-0/1/2
```

Output Example:

```
Pattern "Ethernet1/49" found in one of the following LLDP details

    ------------------------------------------------------------------------------
↪--------------------------------------------------------------------------------
↪-----------------------
    |    Device     | Interface | Parent Interface | Remote Chassis ID | Remote
↪Port Description | Remote Port ID |          Remote System Description
↪|   Remote System Name   |
    ------------------------------------------------------------------------------
↪--------------------------------------------------------------------------------
↪-----------------------
    | edge01.oua04 |  xe-0/1/2 |       ae1        | DE:AD:BE:EF:DE:AD |
↪Ethernet1/49      |                | Cisco NX-OS(tm) n6000, Software (n6000-
↪uk9) | edge07.oua04.dummy.net |
    ------------------------------------------------------------------------------
↪--------------------------------------------------------------------------------
↪-----------------------

Details for interface xe-0/1/2

    ------------------------------------------------------------------------------
↪-----------------------------------------
    |    Device     | Interface | Interface Description | IP Addresses | Enabled |
↪ UP  |    MAC Address    | Speed [Mbps] |
    ------------------------------------------------------------------------------
↪-----------------------------------------
```

(continues on next page)

(continued from previous page)

```
    | edge01.oua04 |  xe-0/1/2 |     ae1 sw01.oua04    |               |   True   |⊠
↪True | BE:EF:DE:AD:BE:EF |    10000     |
    ------------------------------------------------------------------------------
↪-----------------------------------------

LLDP Neighbors for interface xe-0/1/2

    ------------------------------------------------------------------------------
↪--------------------------------------------------------------------------------
↪-----------------------
    |    Device    | Interface | Parent Interface | Remote Chassis ID | Remote⊠
↪Port Description | Remote Port ID |          Remote System Description         ⊠
↪|   Remote System Name   |
    ------------------------------------------------------------------------------
↪--------------------------------------------------------------------------------
↪-----------------------
    | edge01.oua04 |  xe-0/1/2 |       ae1        | DE:AD:BE:EF:DE:AD |       ⊠
↪Ethernet1/49      |                | Cisco NX-OS(tm) n6000, Software (n6000-
↪uk9) | edge07.oua04.dummy.net |
    ------------------------------------------------------------------------------
↪--------------------------------------------------------------------------------
↪-----------------------
```

21.16.24 salt.runners.network

Network tools to run from the Master

`salt.runners.network.`**`wol`**(*mac, bcast='255.255.255.255', destport=9*)

Send a "Magic Packet" to wake up a Minion

CLI Example:

```
salt-run network.wol 08-00-27-13-69-77
salt-run network.wol 080027136977 255.255.255.255 7
salt-run network.wol 08:00:27:13:69:77 255.255.255.255 7
```

`salt.runners.network.`**`wollist`**(*maclist, bcast='255.255.255.255', destport=9*)

Send a "Magic Packet" to wake up a list of Minions. This list must contain one MAC hardware address per line

CLI Example:

```
salt-run network.wollist '/path/to/maclist'
salt-run network.wollist '/path/to/maclist' 255.255.255.255 7
salt-run network.wollist '/path/to/maclist' 255.255.255.255 7
```

`salt.runners.network.`**`wolmatch`**(*tgt, tgt_type='glob', bcast='255.255.255.255', destport=9*)

Send a "Magic Packet" to wake up Minions that are matched in the grains cache

CLI Example:

```
salt-run network.wolmatch minion_id
salt-run network.wolmatch 192.168.0.0/16 tgt_type='ipcidr' bcast=255.255.255.255⊠
↪destport=7
```

21.16.25 salt.runners.pagerduty

Runner Module for Firing Events via PagerDuty

New in version 2014.1.0.

configuration This module can be used by specifying the name of a configuration profile in the master config.

For example:

```
my-pagerduty-account:
    pagerduty.api_key: F3Rbyjbve43rfFWf2214
    pagerduty.subdomain: mysubdomain
```

`salt.runners.pagerduty.`**`create_event`**(*service_key=None*, *description=None*, *details=None*, *incident_key=None*, *profile=None*)

Create an event in PagerDuty. Designed for use in states.

CLI Example:

```
salt-run pagerduty.create_event <service_key> <description> <details>          ⏎
↪profile=my-pagerduty-account
```

The following parameters are required:
service_key This key can be found by using pagerduty.list_services.
description This is a short description of the event.
details This can be a more detailed description of the event.
profile This refers to the configuration profile to use to connect to the PagerDuty service.

`salt.runners.pagerduty.`**`list_escalation_policies`**(*profile=None*, *api_key=None*)

This function is an alias of `list_policies`.

List escalation policies belonging to this account

CLI Example:

salt-run pagerduty.list_policies my-pagerduty-account salt-run pagerduty.list_escalation_policies my-pagerduty-account

`salt.runners.pagerduty.`**`list_incidents`**(*profile=None*, *api_key=None*)

List incidents belonging to this account

CLI Example:

salt-run pagerduty.list_incidents my-pagerduty-account

`salt.runners.pagerduty.`**`list_maintenance_windows`**(*profile=None*, *api_key=None*)

This function is an alias of `list_windows`.

List maintenance windows belonging to this account

CLI Example:

salt-run pagerduty.list_windows my-pagerduty-account salt-run pagerduty.list_maintenance_windows my-pagerduty-account

`salt.runners.pagerduty.`**`list_policies`**(*profile=None*, *api_key=None*)

List escalation policies belonging to this account

CLI Example:

salt-run pagerduty.list_policies my-pagerduty-account salt-run pagerduty.list_escalation_policies my-pagerduty-account

`salt.runners.pagerduty.`**`list_schedules`**(*profile=None*, *api_key=None*)

List schedules belonging to this account

CLI Example:

salt-run pagerduty.list_schedules my-pagerduty-account

`salt.runners.pagerduty.`**`list_services`**(*profile=None*, *api_key=None*)

List services belonging to this account

CLI Example:

salt-run pagerduty.list_services my-pagerduty-account

`salt.runners.pagerduty.`**`list_users`**(*profile=None*, *api_key=None*)

List users belonging to this account

CLI Example:

salt-run pagerduty.list_users my-pagerduty-account

`salt.runners.pagerduty.`**`list_windows`**(*profile=None*, *api_key=None*)

List maintenance windows belonging to this account

CLI Example:

salt-run pagerduty.list_windows my-pagerduty-account salt-run pagerduty.list_maintenance_windows my-pagerduty-account

21.16.26 salt.runners.pillar

Functions to interact with the pillar compiler on the master

`salt.runners.pillar.`**`show_pillar`**(*minion='*'*, ***kwargs*)

Returns the compiled pillar either of a specific minion or just the global available pillars. This function assumes that no minion has the id `*`. Function also accepts pillarenv as attribute in order to limit to a specific pillar branch of git

CLI Example:

shows minion specific pillar:

```
salt-run pillar.show_pillar 'www.example.com'
```

shows global pillar:

```
salt-run pillar.show_pillar
```

shows global pillar for 'dev' pillar environment: (note that not specifying pillarenv will merge all pillar environments using the master config option pillar_source_merging_strategy.)

```
salt-run pillar.show_pillar 'pillarenv=dev'
```

shows global pillar for 'dev' pillar environment and specific pillarenv = dev:

```
salt-run pillar.show_pillar 'saltenv=dev' 'pillarenv=dev'
```

API Example:

```
import salt.config
import salt.runner
opts = salt.config.master_config('/etc/salt/master')
runner = salt.runner.RunnerClient(opts)
pillar = runner.cmd('pillar.show_pillar', [])
print(pillar)
```

`salt.runners.pillar.`**`show_top`**(*minion=None*, *saltenv='base'*)

Returns the compiled top data for pillar for a specific minion. If no minion is specified, we use the first minion we find.

CLI Example:

```
salt-run pillar.show_top
```

21.16.27 salt.runners.pkg

Package helper functions using `salt.modules.pkg`

New in version 2015.8.0.

`salt.runners.pkg.`**`list_upgrades`**(*jid*, *style='group'*, *outputter='nested'*, *ext_source=None*)

Show list of available pkg upgrades using a specified format style

CLI Example:

```
salt-run pkg.list_upgrades jid=20141120114114417719 style=group
```

21.16.28 salt.runners.queue

General management and processing of queues.

This runner facilitates interacting with various queue backends such as the included sqlite3 queue or the planned AWS SQS and Redis queues

The queue functions such as *insert*, *delete*, and *pop* can be used for typical management of the queue.

The *process_queue* function pops the requested number of items from the queue and creates a Salt Event that can then be processed by a Reactor. The *process_queue* function can be called manually, or can be configured to run on a schedule with the Salt Scheduler or regular system cron. It is also possible to use the peer system to allow a minion to call the runner.

This runner, as well as the Queues system, is not api stable at this time.

There are many things that could potentially be done with queues within Salt. For the time being the focus will be on queueing infrastructure actions on specific minions. The queues generally will be populated with minion IDs. When the *process_queue* runner function is called events are created on the Salt Event bus that indicate the queue and a list of one or more minion IDs. The reactor is set up to match on event tags for a specific queue and then take infrastructure actions on those minion IDs. These actions might be to delete the minion's key from the master, use salt-cloud to destroy the vm, or some other custom action.

Queued runners

Using the Salt Queues, references to the commandline arguments of other runners can be saved to be processed later. The queue runners require a queue backend that can store json data (default: `pgjsonb`).

Once the queue is setup, the *runner_queue* will need to be configured.

```
runner_queue:
  queue: runners
  backend: pgjsonb
```

Note: only the queue is required, this defaults to using pgjsonb

---

Once this is set, then the following can be added to the scheduler on the master and it will run the specified amount of commands per time period.

```
schedule:
  runner queue:
    schedule:
      function: queue.process_runner
      minutes: 1
      kwargs:
        quantity: 2
```

The above configuration will pop 2 runner jobs off the runner queue, and then run them. And it will do this every minute, unless there are any jobs that are still running from the last time the process_runner task was executed.

`salt.runners.queue.`**`delete`**(*queue, items, backend='sqlite'*)

Delete an item or items from a queue

CLI Example:

```
salt-run queue.delete myqueue myitem
salt-run queue.delete myqueue myitem backend=sqlite
salt-run queue.delete myqueue "['item1', 'item2', 'item3']"
```

`salt.runners.queue.`**`insert`**(*queue, items, backend='sqlite'*)

Add an item or items to a queue

CLI Example:

```
salt-run queue.insert myqueue myitem
salt-run queue.insert myqueue "['item1', 'item2', 'item3']"
salt-run queue.insert myqueue myitem backend=sqlite
salt-run queue.insert myqueue "['item1', 'item2', 'item3']" backend=sqlite
```

`salt.runners.queue.`**`insert_runner`**(*fun, args=None, kwargs=None, queue=None, backend=None*)

Insert a reference to a runner into the queue so that it can be run later.

fun The runner function that is going to be run

args list or comma-seperated string of args to send to fun

kwargs dictionary of keyword arguments to send to fun

queue queue to insert the runner reference into

backend backend that to use for the queue

CLI Example:

```
salt-run queue.insert_runner test.stdout_print
salt-run queue.insert_runner event.send test_insert_runner kwargs='{"data": {"foo
↪": "bar"}}'
```

`salt.runners.queue.`**`list_items`**(*queue, backend='sqlite'*)

List contents of a queue

CLI Example:

```
salt-run queue.list_items myqueue
salt-run queue.list_items myqueue backend=sqlite
```

`salt.runners.queue.`**`list_length`**(*queue*, *backend='sqlite'*)

Provide the number of items in a queue

CLI Example:

```
salt-run queue.list_length myqueue
salt-run queue.list_length myqueue backend=sqlite
```

`salt.runners.queue.`**`list_queues`**(*backend='sqlite'*)

Return a list of Salt Queues on the backend

CLI Example:

```
salt-run queue.list_queues
salt-run queue.list_queues backend=sqlite
```

`salt.runners.queue.`**`pop`**(*queue*, *quantity=1*, *backend='sqlite'*, *is_runner=False*)

Pop one or more or all items from a queue

CLI Example:

```
salt-run queue.pop myqueue
salt-run queue.pop myqueue 6
salt-run queue.pop myqueue all
salt-run queue.pop myqueue 6 backend=sqlite
salt-run queue.pop myqueue all backend=sqlite
```

`salt.runners.queue.`**`process_queue`**(*queue*, *quantity=1*, *backend='sqlite'*, *is_runner=False*)

Pop items off a queue and create an event on the Salt event bus to be processed by a Reactor.

CLI Example:

```
salt-run queue.process_queue myqueue
salt-run queue.process_queue myqueue 6
salt-run queue.process_queue myqueue all backend=sqlite
```

`salt.runners.queue.`**`process_runner`**(*quantity=1*, *queue=None*, *backend=None*)

Process queued runners

quantity number of runners to process

queue queue to insert the runner reference into

backend backend that to use for the queue

CLI Example:

```
salt-run queue.process_runner
salt-run queue.process_runner 5
```

21.16.29 salt.runners.reactor

A convenience system to manage reactors

Beginning in the 2017.7 release, the reactor runner requires that the reactor system is running. This is accomplished one of two ways, either by having reactors configured or by including `reactor` in the engine configuration for the Salt master.

engines:

- reactor

`salt.runners.reactor.`**`add`**`(`*event, reactors, saltenv='base', test=None*`)`

Add a new reactor

CLI Example:

```
salt-run reactor.add 'salt/cloud/*/destroyed' reactors='/srv/reactor/destroy/*.sls
↪'
```

`salt.runners.reactor.`**`delete`**`(`*event, saltenv='base', test=None*`)`

Delete a reactor

CLI Example:

```
salt-run reactor.delete 'salt/cloud/*/destroyed'
```

`salt.runners.reactor.`**`list_`**`(`*saltenv='base', test=None*`)`

List currently configured reactors

CLI Example:

```
salt-run reactor.list
```

21.16.30 salt.runners.salt

This runner makes Salt's execution modules available on the salt master.

New in version 2016.11.0. Salt's execution modules are normally available on the salt minion. Use this runner to call execution modules on the salt master. Salt *execution modules* are the functions called by the `salt` command.

Execution modules can be called with `salt-run`:

```
salt-run salt.cmd test.ping
# call functions with arguments and keyword arguments
salt-run salt.cmd test.arg 1 2 3 key=value a=1
```

Execution modules are also available to salt runners:

```
__salt__['salt.cmd'](fun=fun, args=args, kwargs=kwargs)
```

`salt.runners.salt.`**`cmd`**`(`*fun, *args, **kwargs*`)`

Changed in version 2018.3.0: Added `with_pillar` argument

Execute `fun` with the given `args` and `kwargs`. Parameter `fun` should be the string *name* of the execution module to call.

Note: Execution modules will be loaded *every time* this function is called. Additionally, keep in mind that since runners execute on the master, custom execution modules will need to be synced to the master using *`salt-run saltutil.sync_modules`*, otherwise they will not be available.

---

with_pillar [False] If `True`, pillar data will be compiled for the master

> **Note:** To target the master in the pillar top file, keep in mind that the default `id` for the master is `<hostname>_master`. This can be overridden by setting an `id` configuration parameter in the master config file.

CLI example:

```
salt-run salt.cmd test.ping
# call functions with arguments and keyword arguments
salt-run salt.cmd test.arg 1 2 3 a=1
salt-run salt.cmd mymod.myfunc with_pillar=True
```

`salt.runners.salt.`**`execute`**(*tgt, fun, arg=(), timeout=None, tgt_type='glob', ret='', jid='', kwarg=None, **kwargs*)

New in version 2017.7.0.

Execute `fun` on all minions matched by `tgt` and `tgt_type`. Parameter `fun` is the name of execution module function to call.

This function should mainly be used as a helper for runner modules, in order to avoid redundant code. For example, when inside a runner one needs to execute a certain function on arbitrary groups of minions, only has to:

```
ret1 = __salt__['salt.execute']('*', 'mod.fun')
ret2 = __salt__['salt.execute']('my_nodegroup', 'mod2.fun2', tgt_type='nodegroup')
```

It can also be used to schedule jobs directly on the master, for example:

```
schedule:
    collect_bgp_stats:
        function: salt.execute
        args:
            - edge-routers
            - bgp.neighbors
        kwargs:
            tgt_type: nodegroup
        days: 1
        returner: redis
```

21.16.31 salt.runners.saltutil

The Saltutil runner is used to sync custom types to the Master. See the *saltutil module* for documentation on managing updates to minions.

New in version 2016.3.0.

`salt.runners.saltutil.`**`sync_all`**(*saltenv='base', extmod_whitelist=None, extmod_blacklist=None*)

Sync all custom types

saltenv [base] The fileserver environment from which to sync. To sync from more than one environment, pass a comma-separated list.

extmod_whitelist [None] dictionary of modules to sync based on type

extmod_blacklist [None] dictionary of modules to blacklist based on type

CLI Example:

```
salt-run saltutil.sync_all
salt-run saltutil.sync_all extmod_whitelist={'runners': ['custom_runner'], 'grains
↪': []}
```

`salt.runners.saltutil.`**`sync_cache`**(*saltenv='base', extmod_whitelist=None, extmod_blacklist=None*)

New in version 2017.7.0.

Sync cache modules from `salt://_cache` to the master

saltenv [base] The fileserver environment from which to sync. To sync from more than one environment, pass a comma-separated list.

extmod_whitelist [None] comma-seperated list of modules to sync

extmod_blacklist [None] comma-seperated list of modules to blacklist based on type

CLI Example:

```
salt-run saltutil.sync_cache
```

`salt.runners.saltutil.`**`sync_clouds`**(*saltenv='base'*, *extmod_whitelist=None*, *extmod_blacklist=None*)

New in version 2017.7.0.

Sync cloud modules from `salt://_clouds` to the master

saltenv [base] The fileserver environment from which to sync. To sync from more than one environment, pass a comma-separated list.

extmod_whitelist [None] comma-seperated list of modules to sync

extmod_blacklist [None] comma-seperated list of modules to blacklist based on type

CLI Example:

```
salt-run saltutil.sync_clouds
```

`salt.runners.saltutil.`**`sync_eauth_tokens`**(*saltenv='base'*, *extmod_whitelist=None*, *extmod_blacklist=None*)

New in version 2018.3.0.

Sync eauth token modules from `salt://_tokens` to the master

saltenv [base] The fileserver environment from which to sync. To sync from more than one environment, pass a comma-separated list.

extmod_whitelist [None] comma-seperated list of modules to sync

extmod_blacklist [None] comma-seperated list of modules to blacklist based on type

CLI Example:

```
salt-run saltutil.sync_eauth_tokens
```

`salt.runners.saltutil.`**`sync_engines`**(*saltenv='base'*, *extmod_whitelist=None*, *extmod_blacklist=None*)

Sync engines from `salt://_engines` to the master

saltenv [base] The fileserver environment from which to sync. To sync from more than one environment, pass a comma-separated list.

extmod_whitelist [None] comma-seperated list of modules to sync

extmod_blacklist [None] comma-seperated list of modules to blacklist based on type

CLI Example:

```
salt-run saltutil.sync_engines
```

`salt.runners.saltutil.`**`sync_executors`**(*saltenv='base'*, *extmod_whitelist=None*, *extmod_blacklist=None*)

New in version 3000.

Sync executor modules from `salt://_executors` to the master

saltenv [base] The fileserver environment from which to sync. To sync from more than one environment, pass a comma-separated list.

extmod_whitelist [None] comma-seperated list of modules to sync

extmod_blacklist [None] comma-seperated list of modules to blacklist based on type

CLI Example:

```
salt-run saltutil.sync_executors
```

`salt.runners.saltutil.`**`sync_fileserver`**(*saltenv='base', extmod_whitelist=None, extmod_blacklist=None*)

New in version 2018.3.0.

Sync fileserver modules from `salt://_fileserver` to the master

saltenv [base] The fileserver environment from which to sync. To sync from more than one environment, pass a comma-separated list.

extmod_whitelist [None] comma-seperated list of modules to sync

extmod_blacklist [None] comma-seperated list of modules to blacklist based on type

CLI Example:

```
salt-run saltutil.sync_fileserver
```

`salt.runners.saltutil.`**`sync_grains`**(*saltenv='base', extmod_whitelist=None, extmod_blacklist=None*)

Sync grains modules from `salt://_grains` to the master

saltenv [base] The fileserver environment from which to sync. To sync from more than one environment, pass a comma-separated list.

extmod_whitelist [None] comma-seperated list of modules to sync

extmod_blacklist [None] comma-seperated list of modules to blacklist based on type

CLI Example:

```
salt-run saltutil.sync_grains
```

`salt.runners.saltutil.`**`sync_modules`**(*saltenv='base', extmod_whitelist=None, extmod_blacklist=None*)

Sync execution modules from `salt://_modules` to the master

saltenv [base] The fileserver environment from which to sync. To sync from more than one environment, pass a comma-separated list.

extmod_whitelist [None] comma-seperated list of modules to sync

extmod_blacklist [None] comma-seperated list of modules to blacklist based on type

CLI Example:

```
salt-run saltutil.sync_modules
```

`salt.runners.saltutil.`**`sync_output`**(*saltenv='base', extmod_whitelist=None, extmod_blacklist=None*)

Sync output modules from `salt://_output` to the master

saltenv [base] The fileserver environment from which to sync. To sync from more than one environment, pass a comma-separated list.

extmod_whitelist [None] comma-seperated list of modules to sync

extmod_blacklist [None] comma-seperated list of modules to blacklist based on type

CLI Example:

```
salt-run saltutil.sync_output
```

`salt.runners.saltutil.`**`sync_pillar`**(*saltenv='base', extmod_whitelist=None, extmod_blacklist=None*)

Sync pillar modules from `salt://_pillar` to the master

saltenv [base] The fileserver environment from which to sync. To sync from more than one environment, pass a comma-separated list.

extmod_whitelist [None] comma-seperated list of modules to sync

extmod_blacklist [None] comma-seperated list of modules to blacklist based on type

CLI Example:

```
salt-run saltutil.sync_pillar
```

`salt.runners.saltutil.`**`sync_proxymodules`**`(`*saltenv='base', extmod_whitelist=None, extmod_blacklist=None*`)`

Sync proxy modules from `salt://_proxy` to the master

saltenv [base] The fileserver environment from which to sync. To sync from more than one environment, pass a comma-separated list.

extmod_whitelist [None] comma-seperated list of modules to sync

extmod_blacklist [None] comma-seperated list of modules to blacklist based on type

CLI Example:

```
salt-run saltutil.sync_proxymodules
```

`salt.runners.saltutil.`**`sync_queues`**`(`*saltenv='base', extmod_whitelist=None, extmod_blacklist=None*`)`

Sync queue modules from `salt://_queues` to the master

saltenv [base] The fileserver environment from which to sync. To sync from more than one environment, pass a comma-separated list.

extmod_whitelist [None] comma-seperated list of modules to sync

extmod_blacklist [None] comma-seperated list of modules to blacklist based on type

CLI Example:

```
salt-run saltutil.sync_queues
```

`salt.runners.saltutil.`**`sync_renderers`**`(`*saltenv='base', extmod_whitelist=None, extmod_blacklist=None*`)`

Sync renderer modules from from `salt://_renderers` to the master

saltenv [base] The fileserver environment from which to sync. To sync from more than one environment, pass a comma-separated list.

extmod_whitelist [None] comma-seperated list of modules to sync

extmod_blacklist [None] comma-seperated list of modules to blacklist based on type

CLI Example:

```
salt-run saltutil.sync_renderers
```

`salt.runners.saltutil.`**`sync_returners`**`(`*saltenv='base', extmod_whitelist=None, extmod_blacklist=None*`)`

Sync returner modules from `salt://_returners` to the master

saltenv [base] The fileserver environment from which to sync. To sync from more than one environment, pass a comma-separated list.

extmod_whitelist [None] comma-seperated list of modules to sync

extmod_blacklist [None] comma-seperated list of modules to blacklist based on type

CLI Example:

```
salt-run saltutil.sync_returners
```

`salt.runners.saltutil.`**`sync_roster`**`(`*saltenv='base', extmod_whitelist=None, extmod_blacklist=None*`)`

New in version 2017.7.0.

Sync roster modules from `salt://_roster` to the master

saltenv [base] The fileserver environment from which to sync. To sync from more than one environment, pass a comma-separated list.

extmod_whitelist [None] comma-seperated list of modules to sync

extmod_blacklist [None] comma-seperated list of modules to blacklist based on type

CLI Example:

```
salt-run saltutil.sync_roster
```

`salt.runners.saltutil.`**`sync_runners`**(*saltenv='base'*, *extmod_whitelist=None*, *extmod_blacklist=None*)

Sync runners from `salt://_runners` to the master

saltenv [base] The fileserver environment from which to sync. To sync from more than one environment, pass a comma-separated list.

extmod_whitelist [None] comma-seperated list of modules to sync

extmod_blacklist [None] comma-seperated list of modules to blacklist based on type

CLI Example:

```
salt-run saltutil.sync_runners
```

`salt.runners.saltutil.`**`sync_sdb`**(*saltenv='base'*, *extmod_whitelist=None*, *extmod_blacklist=None*)

New in version 2017.7.0.

Sync sdb modules from `salt://_sdb` to the master

saltenv [base] The fileserver environment from which to sync. To sync from more than one environment, pass a comma-separated list.

extmod_whitelist [None] comma-seperated list of modules to sync

extmod_blacklist [None] comma-seperated list of modules to blacklist based on type

CLI Example:

```
salt-run saltutil.sync_sdb
```

`salt.runners.saltutil.`**`sync_serializers`**(*saltenv='base'*, *extmod_whitelist=None*, *extmod_blacklist=None*)

New in version 2019.2.0.

Sync serializer modules from `salt://_serializers` to the master

saltenv [base] The fileserver environment from which to sync. To sync from more than one environment, pass a comma-separated list.

extmod_whitelist [None] comma-seperated list of modules to sync

extmod_blacklist [None] comma-seperated list of modules to blacklist based on type

CLI Example:

```
salt-run saltutil.sync_utils
```

`salt.runners.saltutil.`**`sync_states`**(*saltenv='base'*, *extmod_whitelist=None*, *extmod_blacklist=None*)

Sync state modules from `salt://_states` to the master

saltenv [base] The fileserver environment from which to sync. To sync from more than one environment, pass a comma-separated list.

extmod_whitelist [None] comma-seperated list of modules to sync

extmod_blacklist [None] comma-seperated list of modules to blacklist based on type

CLI Example:

```
salt-run saltutil.sync_states
```

`salt.runners.saltutil.`**`sync_thorium`**(*saltenv='base'*, *extmod_whitelist=None*, *extmod_blacklist=None*)

New in version 2018.3.0.

Sync Thorium from `salt://_thorium` to the master

saltenv: **`base`** The fileserver environment from which to sync. To sync from more than one environment, pass a comma-separated list.

extmod_whitelist comma-seperated list of modules to sync

extmod_blacklist comma-seperated list of modules to blacklist based on type

CLI Example:

```
salt-run saltutil.sync_thorium
```

`salt.runners.saltutil.`**`sync_tops`**`(`*saltenv='base', extmod_whitelist=None, extmod_blacklist=None*`)`

New in version 2016.3.7,2016.11.4,2017.7.0.

Sync master_tops modules from `salt://_tops` to the master

saltenv [base] The fileserver environment from which to sync. To sync from more than one environment, pass a comma-separated list.

CLI Example:

```
salt-run saltutil.sync_tops
```

`salt.runners.saltutil.`**`sync_utils`**`(`*saltenv='base', extmod_whitelist=None, extmod_blacklist=None*`)`

New in version 2016.11.0.

Sync utils modules from `salt://_utils` to the master

saltenv [base] The fileserver environment from which to sync. To sync from more than one environment, pass a comma-separated list.

extmod_whitelist [None] comma-seperated list of modules to sync

extmod_blacklist [None] comma-seperated list of modules to blacklist based on type

CLI Example:

```
salt-run saltutil.sync_utils
```

`salt.runners.saltutil.`**`sync_wheel`**`(`*saltenv='base', extmod_whitelist=None, extmod_blacklist=None*`)`

Sync wheel modules from `salt://_wheel` to the master

saltenv [base] The fileserver environment from which to sync. To sync from more than one environment, pass a comma-separated list.

extmod_whitelist [None] comma-seperated list of modules to sync

extmod_blacklist [None] comma-seperated list of modules to blacklist based on type

CLI Example:

```
salt-run saltutil.sync_wheel
```

21.16.32 salt.runners.sdb

Runner for setting and querying data via the sdb API on the master

`salt.runners.sdb.`**`delete`**`(`*uri*`)`

Delete a value from a db, using a uri in the form of `sdb://<profile>/<key>`. If the uri provided does not start with `sdb://` or the value is not successfully deleted, return `False`.

CLI Example:

```
salt-run sdb.delete sdb://mymemcached/foo
```

`salt.runners.sdb.`**`get`**`(`*uri*`)`

Get a value from a db, using a uri in the form of sdb://<profile>/<key>. If the uri provided does not start with sdb://, then it will be returned as-is.

CLI Example:

```
salt-run sdb.get sdb://mymemcached/foo
```

`salt.runners.sdb.`**`get_or_set_hash`**(*uri*, *length=8*, *chars='abcdefghijklmnopqrstuvwxyz0123456789!@#$%^&*(-_=+)'*)

Perform a one-time generation of a hash and write it to sdb. If that value has already been set return the value instead.

This is useful for generating passwords or keys that are specific to multiple minions that need to be stored somewhere centrally.

CLI Example:

```
salt-run sdb.get_or_set_hash 'SECRET_KEY' 50
```

Warning: This function could return strings which may contain characters which are reserved as directives by the YAML parser, such as strings beginning with %. To avoid issues when using the output of this function in an SLS file containing YAML+Jinja, surround the call with single quotes.

`salt.runners.sdb.`**`set_`**(*uri*, *value*)

Set a value in a db, using a uri in the form of `sdb://<profile>/<key>`. If the uri provided does not start with `sdb://` or the value is not successfully set, return `False`.

CLI Example:

```
salt-run sdb.set sdb://mymemcached/foo bar
```

21.16.33 salt.runners.smartos_vmadm

Runner for SmartOS minions control vmadm

`salt.runners.smartos_vmadm.`**`get`**(*search*, *one=True*)

Return information for vms

search [string] filter vms, see the execution module.

one [boolean] return only one vm

Note: If the search parameter does not contain an equal (=) symbol it will be assumed it will be tried as uuid, hostname, and alias.

CLI Example:

```
salt-run vmadm.get 91244bba-1146-e4ec-c07e-e825e0223aa9
salt-run vmadm.get search='alias=saskia'
```

`salt.runners.smartos_vmadm.`**`is_running`**(*search*)

Return true if vm is running

search [string] filter vms, see the execution module.

Note: If the search parameter does not contain an equal (=) symbol it will be assumed it will be tried as uuid, hostname, and alias.

Note: If multiple vms are matched, the result will be true of ALL vms are running

CLI Example:

```
salt-run vmadm.is_running 91244bba-1146-e4ec-c07e-e825e0223aa9
salt-run vmadm.is_running search='alias=julia'
```

`salt.runners.smartos_vmadm.`**`list_vms`**(*search=None*, *verbose=False*)

List all vms

search [string] filter vms, see the execution module

verbose [boolean] print additional information about the vm

CLI Example:

```
salt-run vmadm.list
salt-run vmadm.list search='type=KVM'
salt-run vmadm.list verbose=True
```

`salt.runners.smartos_vmadm.`**`nodes`**(*verbose=False*)

List all compute nodes

verbose [boolean] print additional information about the node e.g. platform version, hvm capable, ...

CLI Example:

```
salt-run vmadm.nodes
salt-run vmadm.nodes verbose=True
```

`salt.runners.smartos_vmadm.`**`reboot`**(*search*, *one=True*, *force=False*)

Reboot one or more vms

search [string] filter vms, see the execution module.

one [boolean] reboot only one vm

force [boolean] force reboot, faster but no graceful shutdown

Note: If the search parameter does not contain an equal (=) symbol it will be assumed it will be tried as uuid, hostname, and alias.

CLI Example:

```
salt-run vmadm.reboot 91244bba-1146-e4ec-c07e-e825e0223aa9
salt-run vmadm.reboot search='alias=marije'
salt-run vmadm.reboot search='type=KVM' one=False
```

`salt.runners.smartos_vmadm.`**`start`**(*search*, *one=True*)

Start one or more vms

search [string] filter vms, see the execution module.

one [boolean] start only one vm

Note: If the search parameter does not contain an equal (=) symbol it will be assumed it will be tried as uuid, hostname, and alias.

CLI Example:

```
salt-run vmadm.start 91244bba-1146-e4ec-c07e-e825e0223aa9
salt-run vmadm.start search='alias=jiska'
salt-run vmadm.start search='type=KVM' one=False
```

`salt.runners.smartos_vmadm.`**`stop`**(*search*, *one=True*)

Stop one or more vms

search [string] filter vms, see the execution module.

one [boolean] stop only one vm

Note: If the search parameter does not contain an equal (=) symbol it will be assumed it will be tried as uuid, hostname, and alias.

CLI Example:

```
salt-run vmadm.stop 91244bba-1146-e4ec-c07e-e825e0223aa9
salt-run vmadm.stop search='alias=jody'
salt-run vmadm.stop search='type=KVM' one=False
```

21.16.34 salt.runners.spacewalk

Spacewalk Runner

New in version 2016.3.0.

Runner to interact with Spacewalk using Spacewalk API

codeauthor Nitin Madhok <nmadhok@clemson.edu>, Joachim Werner <joe@suse.com>, Benedikt Werner <1benediktwerner@gmail.com>

maintainer Benedikt Werner <1benediktwerner@gmail.com>

To use this runner, set up the Spacewalk URL, username and password in the master configuration at `/etc/salt/master` or `/etc/salt/master.d/spacewalk.conf`:

```
spacewalk:
  spacewalk01.domain.com:
    username: 'testuser'
    password: 'verybadpass'
  spacewalk02.domain.com:
    username: 'testuser'
    password: 'verybadpass'
```

Note: Optionally, `protocol` can be specified if the spacewalk server is not using the defaults. Default is `protocol: https`.

`salt.runners.spacewalk.`**`addGroupsToKey`**(*server, activation_key, groups*)

Add server groups to a activation key

CLI Example:

```
salt-run spacewalk.addGroupsToKey spacewalk01.domain.com 1-my-key '[group1,
↪group2]'
```

`salt.runners.spacewalk.`**`api`**(*server, command, *args, **kwargs*)

Call the Spacewalk xmlrpc api.

CLI Example:

```
salt-run spacewalk.api spacewalk01.domain.com systemgroup.create MyGroup
↪Description
salt-run spacewalk.api spacewalk01.domain.com systemgroup.create arguments='[
↪"MyGroup", "Description"]'
```

State Example:

```
create_group:
  salt.runner:
    - name: spacewalk.api
    - server: spacewalk01.domain.com
    - command: systemgroup.create
    - arguments:
      - MyGroup
      - Description
```

`salt.runners.spacewalk.`**`deleteAllActivationKeys`**`(`*server*`)`

Delete all activation keys from Spacewalk

CLI Example:

```
salt-run spacewalk.deleteAllActivationKeys spacewalk01.domain.com
```

`salt.runners.spacewalk.`**`deleteAllGroups`**`(`*server*`)`

Delete all server groups from Spacewalk

`salt.runners.spacewalk.`**`deleteAllSystems`**`(`*server*`)`

Delete all systems from Spacewalk

CLI Example:

```
salt-run spacewalk.deleteAllSystems spacewalk01.domain.com
```

`salt.runners.spacewalk.`**`unregister`**`(`*name, server_url*`)`

Unregister specified server from Spacewalk

CLI Example:

```
salt-run spacewalk.unregister my-test-vm spacewalk01.domain.com
```

21.16.35 salt.runners.ssh

A Runner module interface on top of the salt-ssh Python API.

This allows for programmatic use from salt-api, the Reactor, Orchestrate, etc.

`salt.runners.ssh.`**`cmd`**`(`*tgt, fun, arg=(), timeout=None, tgt_type='glob', kwarg=None*`)`

New in version 2015.5.0.

Changed in version 2017.7.0: The `expr_form` argument has been renamed to `tgt_type`, earlier releases must use `expr_form`.

Execute a single command via the salt-ssh subsystem and return all routines at once

A wrapper around the *`SSHClient.cmd`* method.

21.16.36 salt.runners.state

Execute orchestration functions

`salt.runners.state.`**`event`**`(`*tagmatch='*', count=-1, quiet=False, sock_dir=None, pretty=False, node='master'*`)`

Watch Salt's event bus and block until the given tag is matched

New in version 2014.7.0.

Changed in version 2019.2.0: `tagmatch` can now be either a glob or regular expression.

This is useful for utilizing Salt's event bus from shell scripts or for taking simple actions directly from the CLI.

Enable debug logging to see ignored events.

Parameters

- **`tagmatch`** – the event is written to stdout for each tag that matches this glob or regular expression.
- **`count`** – this number is decremented for each event that matches the `tagmatch` parameter; pass `-1` to listen forever.
- **`quiet`** – do not print to stdout; just block
- **`sock_dir`** – path to the Salt master's event socket file.
- **`pretty`** – Output the JSON all on a single line if `False` (useful for shell tools); pretty-print the JSON output if `True`.
- **`node`** – Watch the minion-side or master-side event bus. .. versionadded:: 2016.3.0

CLI Examples:

```
# Reboot a minion and run highstate when it comes back online
salt 'jerry' system.reboot && \\
    salt-run state.event 'salt/minion/jerry/start' count=1 quiet=True && \\
    salt 'jerry' state.highstate

# Reboot multiple minions and run highstate when all are back online
salt -L 'kevin,stewart,dave' system.reboot && \\
    salt-run state.event 'salt/minion/*/start' count=3 quiet=True && \\
    salt -L 'kevin,stewart,dave' state.highstate

# Watch the event bus forever in a shell while-loop.
salt-run state.event | while read -r tag data; do
    echo $tag
    echo $data | jq --color-output .
done
```

See also:

See https://github.com/saltstack/salt/blob/master/tests/eventlisten.sh for an example of usage within a shell script.

`salt.runners.state.`**`orchestrate`**(*mods, saltenv='base', test=None, exclude=None, pillar=None, pillarenv=None, pillar_enc=None, orchestration_jid=None*)

New in version 0.17.0.

Execute a state run from the master, used as a powerful orchestration system.

See also:

More Orchestrate documentation

- *Full Orchestrate Tutorial*
- *`Docs for the master-side state module`*

CLI Examples:

```
salt-run state.orchestrate webserver
salt-run state.orchestrate webserver saltenv=dev test=True
salt-run state.orchestrate webserver saltenv=dev pillarenv=aws
```

Changed in version 2014.1.1: Runner renamed from `state.sls` to `state.orchestrate`

Changed in version 2014.7.0: Runner uses the pillar variable

Changed in version develop: Runner uses the pillar_enc variable that allows renderers to render the pillar. This is usable when supplying the contents of a file as pillar, and the file contains gpg-encrypted entries.

See also:

GPG renderer documentation

CLI Examples:

```
salt-run state.orchestrate webserver pillar_enc=gpg pillar="$(cat somefile.json)"
```

`salt.runners.state.`**`orchestrate_high`**(*data*, *test=None*, *queue=False*, *pillar=None*, ***kwargs*)

Execute a single state orchestration routine

New in version 2015.5.0.

CLI Example:

```
salt-run state.orchestrate_high '{
    stage_one:
        {salt.state: [{tgt: "db*"}, {sls: postgres_setup}]},
    stage_two:
        {salt.state: [{tgt: "web*"}, {sls: apache_setup}, {
            require: [{salt: stage_one}],
        }]},
    }'
```

`salt.runners.state.`**`orchestrate_show_sls`**(*mods*, *saltenv='base'*, *test=None*, *queue=False*, *pillar=None*, *pillarenv=None*, *pillar_enc=None*)

Display the state data from a specific sls, or list of sls files, after being render using the master minion.

Note, the master minion adds a "_master" suffix to it's minion id.

See also:

The state.show_sls module function

CLI Example: .. code-block:: bash
salt-run state.orch_show_sls my-orch-formula.my-orch-state 'pillar={ nodegroup: ng1 }'

`salt.runners.state.`**`orchestrate_single`**(*fun*, *name*, *test=None*, *queue=False*, *pillar=None*, ***kwargs*)

Execute a single state orchestration routine

New in version 2015.5.0.

CLI Example:

```
salt-run state.orchestrate_single fun=salt.wheel name=key.list_all
```

`salt.runners.state.`**`pause`**(*jid*, *state_id=None*, *duration=None*)

Set up a state id pause, this instructs a running state to pause at a given state id. This needs to pass in the jid of the running state and can optionally pass in a duration in seconds.

`salt.runners.state.`**`resume`**(*jid*, *state_id=None*)

Remove a pause from a jid, allowing it to continue

`salt.runners.state.`**`rm_pause`**(*jid*, *state_id=None*)

This function is an alias of `resume`.

Remove a pause from a jid, allowing it to continue

`salt.runners.state.`**`set_pause`**(*jid*, *state_id=None*, *duration=None*)

This function is an alias of `pause`.

Set up a state id pause, this instructs a running state to pause at a given state id. This needs to pass in the jid of the running state and can optionally pass in a duration in seconds.

`salt.runners.state.`**`soft_kill`**(*jid*, *state_id=None*)

Set up a state run to die before executing the given state id, this instructs a running state to safely exit at a given state id. This needs to pass in the jid of the running state. If a state_id is not passed then the jid referenced will be safely exited at the beginning of the next state run.

21.16.37 salt.runners.survey

A general map/reduce style salt runner for aggregating results returned by several different minions.

New in version 2014.7.0.

Aggregated results are sorted by the size of the minion pools which returned matching results.

Useful for playing the game: *"some of these things are not like the others..."* when identifying discrepancies in a large infrastructure managed by salt.

`salt.runners.survey.`**`diff`**(**args*, ***kwargs*)

Return the DIFFERENCE of the result sets returned by each matching minion pool

New in version 2014.7.0.

These pools are determined from the aggregated and sorted results of a salt command.

This command displays the "diffs" as a series of 2-way differences – namely the difference between the FIRST displayed minion pool (according to sort order) and EACH SUBSEQUENT minion pool result set.

Differences are displayed according to the Python `difflib.unified_diff()` as in the case of the salt execution module `file.get_diff`.

This command is submitted via a salt runner using the general form:

```
salt-run survey.diff [survey_sort=up/down] <target>
             <salt-execution-module> <salt-execution-module parameters>
```

Optionally accept a `survey_sort=` parameter. Default: `survey_sort=down`

CLI Example #1: (Example to display the "differences of files")

```
salt-run survey.diff survey_sort=up "*" cp.get_file_str file:///etc/hosts
```

`salt.runners.survey.`**`hash`**(**args*, ***kwargs*)

Return the MATCHING minion pools from the aggregated and sorted results of a salt command

New in version 2014.7.0.

This command is submitted via a salt runner using the general form:

```
salt-run survey.hash [survey_sort=up/down] <target>
          <salt-execution-module> <salt-execution-module parameters>
```

Optionally accept a `survey_sort=` parameter. Default: `survey_sort=down`

CLI Example #1: (functionally equivalent to `salt-run manage.up`)

```
salt-run survey.hash "*" test.ping
```

CLI Example #2: (find an "outlier" minion config file)

```
salt-run survey.hash "*" file.get_hash /etc/salt/minion survey_sort=up
```

21.16.38 salt.runners.test

This runner is used only for test purposes and servers no production purpose

`salt.runners.test.`**`arg`**`(`**args, **kwargs*`)`
Output the given args and kwargs

Kwargs will be filtered for 'private' keynames.

`salt.runners.test.`**`get_opts`**`()`
New in version 2018.3.0.

Return the configuration options of the master.

CLI Example:

```
salt-run test.get_opts
```

`salt.runners.test.`**`metasyntactic`**`(`*locality='us'*`)`
Return common metasyntactic variables for the given locality

`salt.runners.test.`**`raw_arg`**`(`**args, **kwargs*`)`
Output the given args and kwargs

`salt.runners.test.`**`sleep`**`(`*s_time=10*`)`
Sleep t seconds, then return True

`salt.runners.test.`**`stdout_print`**`()`
Print 'foo' and return 'bar'

`salt.runners.test.`**`stream`**`()`
Return True

21.16.39 salt.runners.thin

The thin runner is used to manage the salt thin systems.

Salt Thin is a transport-less version of Salt that can be used to run routines in a standalone way. This runner has tools which generate the standalone salt system for easy consumption.

`salt.runners.thin.`**`generate`**`(`*extra_mods='', overwrite=False, so_mods='', python2_bin='python2', python3_bin='python3', absonly=True, compress='gzip'*`)`
Generate the salt-thin tarball and print the location of the tarball Optional additional mods to include (e.g. mako) can be supplied as a comma delimited string. Permits forcing an overwrite of the output file as well.

CLI Example:

```
salt-run thin.generate
salt-run thin.generate mako
salt-run thin.generate mako,wempy 1
salt-run thin.generate overwrite=1
```

`salt.runners.thin.`**`generate_min`**`(`*extra_mods='', overwrite=False, so_mods='', python2_bin='python2', python3_bin='python3'*`)`
Generate the salt-thin tarball and print the location of the tarball Optional additional mods to include (e.g. mako) can be supplied as a comma delimited string. Permits forcing an overwrite of the output file as well.

CLI Example:

```
salt-run thin.generate_min
```

21.16.40 salt.runners.vault

maintainer SaltStack

maturity new

platform all

Runner functions supporting the Vault modules. Configuration instructions are documented in the execution module docs.

`salt.runners.vault.`**`generate_token`**`(`*minion_id*, *signature*, *impersonated_by_master=False*`)`

Generate a Vault token for minion minion_id

minion_id The id of the minion that requests a token

signature Cryptographic signature which validates that the request is indeed sent by the minion (or the master, see impersonated_by_master).

impersonated_by_master If the master needs to create a token on behalf of the minion, this is True. This happens when the master generates minion pillars.

`salt.runners.vault.`**`show_policies`**`(`*minion_id*`)`

Show the Vault policies that are applied to tokens for the given minion

minion_id The minions id

CLI Example:

```
salt-run vault.show_policies myminion
```

`salt.runners.vault.`**`unseal`**`()`

Unseal Vault server

This function uses the 'keys' from the 'vault' configuration to unseal vault server

vault:

keys:

- n63/TbrQuL3xaIW7ZZpuXj/tIfnK1/MbVxO4vT3wYD2A
- S9OwCvMRhErEA4NVVELYBs6w/Me6+urgUr24xGK44Uy3
- F1j4b7JKq850NS6Kboiy5laJ0xY8dWJvB3fcwA+SraYl
- 1cYtvjKJNDVam9c7HNqJUfINk4PYyAXIpjkpN/sIuzPv
- 3pPK5X6vGtwLhNOFv1U2elahECz3HpRUfNXJFYLw6lid

CLI Examples:

```
salt-run vault.unseal
```

21.16.41 salt.runners.venafiapi

Support for Venafi

depends

- vcert Python module

configuration In order to connect to Venafi services you need to specify it in Salt master configuration. Example for Venafi Cloud (using env variables):

venafi: api_key: "sdb://osenv/CLOUDAPIKEY"

Example for Venafi Platform (using env variables):

venafi: base_url: "https://tpp.example.com/" tpp_user: admin tpp_password: "sdb://osenv/TPP_PASSWORD" trust_bundle: "/opt/venafi/bundle.pem"

`salt.runners.venafiapi.`**`del_cached_domain`**`(`*domains*`)`

Delete cached domains from the master

CLI Example:

```
salt-run venafi.del_cached_domain domain1.example.com,domain2.example.com
```

`salt.runners.venafiapi.`**`list_domain_cache`**`()`

List domains that have been cached

CLI Example:

```
salt-run venafi.list_domain_cache
```

`salt.runners.venafiapi.`**`renew`**`(`*minion_id, dns_name=None, zone=None, country=None, state=None, loc=None, org=None, org_unit=None, key_password=None, csr_path=None, pkey_path=None*`)`

Request a new certificate

CLI Example:

```
salt-run venafi.request <minion_id> <dns_name>
```

`salt.runners.venafiapi.`**`request`**`(`*minion_id, dns_name=None, zone=None, country=None, state=None, loc=None, org=None, org_unit=None, key_password=None, csr_path=None, pkey_path=None*`)`

Request a new certificate

CLI Example:

```
salt-run venafi.request <minion_id> <dns_name>
```

`salt.runners.venafiapi.`**`show_cert`**`(`*dns_name*`)`

Show issued certificate for domain

CLI Example:

```
salt-run venafi.show_cert example.com
```

21.16.42 salt.runners.virt

Control virtual machines via Salt

`salt.runners.virt.`**`force_off`**`(`*name*`)`

Force power down the named virtual machine

`salt.runners.virt.`**`host_info`**`(`*host=None*`)`

Return information about the host connected to this master `salt.runners.virt.`**`init`**(*name*, *cpu*, *mem*, *image*, *hypervisor='kvm'*, *host=None*, *seed=True*, *nic='default'*, *install=True*, *start=True*, *disk='default'*, *saltenv='base'*, *enable_vnc=False*, *seed_cmd='seed.apply'*, *enable_qcow=False*)

This routine is used to create a new virtual machine. This routines takes a number of options to determine what the newly created virtual machine will look like.

name The mandatory name of the new virtual machine. The name option is also the minion id, all minions must have an id.

cpu The number of cpus to allocate to this new virtual machine.

mem The amount of memory to allocate to this virtual machine. The number is interpreted in megabytes.

image The network location of the virtual machine image, commonly a location on the salt fileserver, but http, https and ftp can also be used.

hypervisor The hypervisor to use for the new virtual machine. Default is *kvm*.

host The host to use for the new virtual machine, if this is omitted Salt will automatically detect what host to use.

seed Set to *False* to prevent Salt from seeding the new virtual machine.

nic The nic profile to use, defaults to the "default" nic profile which assumes a single network interface per VM associated with the "br0" bridge on the master.

install Set to False to prevent Salt from installing a minion on the new VM before it spins up.

disk The disk profile to use

saltenv The Salt environment to use

enable_vnc Whether a VNC screen is attached to resulting VM. Default is *False*.

seed_cmd If seed is *True*, use this execution module function to seed new VM. Default is *seed.apply*.

enable_qcow Clone disk image as a copy-on-write qcow2 image, using downloaded *image* as backing file.

`salt.runners.virt.`**`list`**(*host=None*, *quiet=False*, *hyper=None*)

List the virtual machines on each host, this is a simplified query, showing only the virtual machine names belonging to each host. A single host can be passed in to specify an individual host to list.

`salt.runners.virt.`**`migrate`**(*name*, *target=''*)

Migrate a VM from one host to another. This routine will just start the migration and display information on how to look up the progress.

`salt.runners.virt.`**`next_host`**()

Return the host to use for the next autodeployed VM. This queries the available host and executes some math the determine the most "available" next host.

`salt.runners.virt.`**`pause`**(*name*)

Pause the named VM

`salt.runners.virt.`**`purge`**(*name*, *delete_key=True*)

Destroy the named VM

`salt.runners.virt.`**`query`**(*host=None*, *quiet=False*)

Query the virtual machines. When called without options all hosts are detected and a full query is returned. A single host can be passed in to specify an individual host to query.

`salt.runners.virt.`**`reset`**(*name*)

Force power down and restart an existing VM

`salt.runners.virt.`**`resume`**(*name*)

Resume a paused VM

`salt.runners.virt.`**`start`**(*name*)

Start a named virtual machine

`salt.runners.virt.`**`vm_info`**(*name*, *quiet=False*)

Return the information on the named VM

21.16.43 salt.runners.vistara

Vistara Runner

Runner to interact with the Vistara (http://www.vistarait.com/) REST API

codeauthor Brad Thurber <brad.thurber@gmail.com>

To use this runner, the Vistara client_id and Vistara oauth2 client_key and client_secret must be set in the master config.

For example `/etc/salt/master.d/_vistara.conf`:

```
vistara:
  client_id: client_012345
  client_key: N0tReallyaR3alKeyButShouldB12345
  client_secret: ThisI5AreallyLongsecretKeyIwonderwhyTheyMakethemSoBigTheseDays00
```

`salt.runners.vistara.`**`delete_device`**(*name*, *safety_on=True*)

Deletes a device from Vistara based on DNS name or partial name. By default, delete_device will only perform the delete if a single host is returned. Set safety_on=False to delete all matches (up to default API search page size)

CLI Example:

```
salt-run vistara.delete_device 'hostname-101.mycompany.com'
salt-run vistara.delete_device 'hostname-101'
salt-run vistara.delete_device 'hostname-1' safety_on=False
```

21.16.44 salt.runners.winrepo

Runner to manage Windows software repo

`salt.runners.winrepo.`**`genrepo`**(*opts=None*, *fire_event=True*)

Generate winrepo_cachefile based on sls files in the winrepo_dir

opts Specify an alternate opts dict. Should not be used unless this function is imported into an execution module.

fire_event [True] Fire an event on failure. Only supported on the master.

CLI Example:

```
salt-run winrepo.genrepo
```

`salt.runners.winrepo.`**`update_git_repos`**(*opts=None*, *clean=False*, *masterless=False*)

Checkout git repos containing Windows Software Package Definitions

opts Specify an alternate opts dict. Should not be used unless this function is imported into an execution module.

clean [False] Clean repo cachedirs which are not configured under *`winrepo_remotes`*.

> **Warning:** This argument should not be set to `True` if a mix of git and non-git repo definitions are being used, as it will result in the non-git repo definitions being removed.

New in version 2015.8.0.

CLI Examples:

```
salt-run winrepo.update_git_repos
salt-run winrepo.update_git_repos clean=True
```

21.17 sdb modules

cache	cache Module
confidant	An SDB module for getting credentials from confidant.
consul	Consul sdb Module
couchdb	CouchDB sdb Module
env	Environment sdb Module
etcd_db	etcd Database Module
keyring_db	Keyring Database Module
memcached	Memcached sdb Module
redis_sdb	Redis SDB module
rest	Generic REST API SDB Module
sqlite3	SQLite sdb Module
tism	tISM - the Immutable Secrets Manager SDB Module
vault	Vault SDB Module
yaml	Pull sdb values from a YAML file

21.17.1 salt.sdb.cache module

cache Module

maintainer SaltStack

maturity New

platform all

New in version 2017.7.0.

This module provides access to Salt's cache subsystem.

Like all sdb modules, the cache module requires a configuration profile to be configured in either the minion or master configuration file. This profile requires very little. In the example:

```
mastercloudcache:
  driver: cache
  bank: cloud/active/ec2/my-ec2-conf/saltmaster
  cachedir: /var/cache/salt
```

The `driver` refers to the cache module, `bank` refers to the cache bank that contains the data and `cachedir` (optional), if used, points to an alternate directory for cache data storage.

```
master_ip: sdb://mastercloudcache/public_ips
```

It is also possible to override both the `bank` and `cachedir` options inside the SDB URI:

```
master_ip: sdb://mastercloudcache/public_ips?cachedir=/var/cache/salt
```

For this reason, both the `bank` and the `cachedir` options can be omitted from the SDB profile. However, if the `bank` option is omitted, it must be specified in the URI:

```
master_ip: sdb://mastercloudcache/public_ips?bank=cloud/active/ec2/my-ec2-conf/
↪saltmaster
```

`salt.sdb.cache.`**`delete`**(*key*, *service=None*, *profile=None*)

Get a value from the cache service

`salt.sdb.cache.`**`get`**(*key*, *service=None*, *profile=None*)

Get a value from the cache service

`salt.sdb.cache.`**`set_`**(*key*, *value*, *service=None*, *profile=None*)

Set a key/value pair in the cache service

21.17.2 salt.sdb.confidant

An SDB module for getting credentials from confidant.

Configuring the Confidant module

The module can be configured via sdb in the minion config:

```
confidant:
  driver: confidant
  # The URL of the confidant web service
  url: 'https://confidant-production.example.com'
  # The context to use for KMS authentication
  auth_context:
    from: example-production-iad
    to: confidant-production-iad
    user_type: service
  # The KMS master key to use for authentication
  auth_key: "alias/authnz"
  # Cache file for KMS auth token
  token_cache_file: /run/confidant/confidant_token
  # The duration of the validity of a token, in minutes
  token_duration: 60
  # key, keyid and region can be defined in the profile, but it's generally
  # best to use IAM roles or environment variables for AWS auth.
  keyid: 98nh9h9h908h09kjjk
  key: jhf908gyeghehe0he0g8h9u0j0n0n09hj09h0
  region: us-east-1
```

depends confidant-common, confidant-client

Module Documentation

`salt.sdb.confidant.`**`get`**(*key*, *profile=None*)

Read pillar data from Confidant via its API.

CLI Example:

salt myminion sdb.get 'sdb://confidant/credentials'

Valid keys are: credentials, credentials_metadata, result. credentials returns a dict of joined credential_pairs, credentials_metadata returns a dict of metadata relevant to the credentials mapped to the confidant service, and result returns a bool that can be used to determine if the sdb call succeeded or failed to fetch credentials from confidant (or from local cache). If result is false, the data in credentials or credentials_metadata can't be trusted.

21.17.3 salt.sdb.consul module

Consul sdb Module

maintainer SaltStack

maturity New

platform all

This module allows access to Consul using an `sdb://` URI

Like all sdb modules, the Consul module requires a configuration profile to be configured in either the minion or master configuration file. This profile requires very little. For example:

```
myconsul:
  driver: consul
  host: 127.0.0.1
  port: 8500
  token: b6376760-a8bb-edd5-fcda-33bc13bfc556
  scheme: http
  consistency: default
  dc: dev
  verify: True
```

The `driver` refers to the Consul module, all other options are optional. For option details see: https://python-consul.readthedocs.io/en/latest/#consul

`salt.sdb.consul.`**`get`**`(`*key*, *profile=None*`)`

`salt.sdb.consul.`**`get_conn`**`(`*profile*`)`
Return a client object for accessing consul

`salt.sdb.consul.`**`set_`**`(`*key*, *value*, *profile=None*`)`

21.17.4 salt.sdb.couchdb

CouchDB sdb Module

maintainer SaltStack

maturity New

depends python2-couchdb

platform all

This allow interaction between Salt and a CouchDB [couchdb.apache.org] database. It uses salt's *sdb* system to allow for inserts and retrevals using the *sdb://* prefix in salt configuration files.

To use the couchbase sdb module, it must first be configured in the salt master or minion config. The following arguments are required:

```
couchdb_sdb:
  driver: couchdb
  host: localhost
  port: 5984
  database: salt_sdb
```

One could then query the CouchDB instance via an *sdb://* URI such as the following:

```
password: sdb://couchdb_sdb/mykey
```

To use this interface, you must track IDs on your own or have another source to do the map-reduce logic necessary to calculate the ID you wish to fetch.

Additional contributions to build true map-reduce functionality into this module would be welcome.

`salt.sdb.couchdb.`**`get`**`(`*key, profile=None*`)`
Get a value from couchdb by id

`salt.sdb.couchdb.`**`set_`**`(`*key, value, profile=None*`)`
Set a key/value pair in couchdb

21.17.5 salt.sdb.env module

Environment sdb Module

maintainer SaltStack

maturity New

depends None

platform all

This module allows access to environment variables using an `sdb://` URI.

Example configuration for this module:

```
osenv:
  driver: env
```

WARNING:

OS environment variables will be available to read via SDB. Please make sure you don't have any sensitive data in your environment variables!!

Example usage of sdb env module:

```
set some env var:
  cmd.run:
    - name: echo {{ salt['sdb.set']('sdb://osenv/foo', 'bar') }}
    - order: 1

{% if salt['sdb.get']('sdb://osenv/foo') == 'bar' %}
always-changes-and-succeeds:
  test.succeed_with_changes:
    - name: foo
{% else %}
always-changes-and-fails:
  test.fail_with_changes:
    - name: foo
{% endif  %}
```

The above example will return success.

The `env` sdb module can also be used with salt cloud. Assuming you have exported the environment variable named `compute` (and have `osenv` defined). The example below will look for the salt cloud config key `compute_name` in the environment:

```
my-openstack-config:
  compute_name: sdb://osenv/compute
  ..snip
```

`salt.sdb.env.`**`get`**(*key*, *profile=None*)
Get a value

`salt.sdb.env.`**`set_`**(*key*, *value*, *profile=None*)
Set a key/value pair

21.17.6 salt.sdb.etcd_db

etcd Database Module

maintainer SaltStack

maturity New

depends python-etcd

platform all

New in version 2015.5.0.

This module allows access to the etcd database using an `sdb://` URI. This package is located at `https://pypi.python.org/pypi/python-etcd`.

Like all sdb modules, the etcd module requires a configuration profile to be configured in either the minion or master configuration file. This profile requires very little. In the example:

```
myetcd:
  driver: etcd
  etcd.host: 127.0.0.1
  etcd.port: 2379
```

The `driver` refers to the etcd module, `etcd.host` refers to the host that is hosting the etcd database and `etcd.port` refers to the port on that host.

```
password: sdb://myetcd/mypassword
```

`salt.sdb.etcd_db.`**`delete`**(*key*, *service=None*, *profile=None*)
Get a value from the etcd service

`salt.sdb.etcd_db.`**`get`**(*key*, *service=None*, *profile=None*)
Get a value from the etcd service

`salt.sdb.etcd_db.`**`set_`**(*key*, *value*, *service=None*, *profile=None*)
Set a key/value pair in the etcd service

21.17.7 salt.sdb.keyring_db

Keyring Database Module

maintainer SaltStack

maturity New

depends keyring

platform all

This module allows access to the keyring package using an `sdb://` URI. This package is located at `https://pypi.python.org/pypi/keyring`.

Care must be taken when using keyring. Not all keyend backends are supported on all operating systems. Also, many backends require an agent to be running in order to work. For instance, the "Secret Service" backend requires a compatible agent such as `gnome-keyring-daemon` or `kwallet` to be running. The keyczar backend does not seem to enjoy the benefits of an agent, and so using it will require either that the password is typed in manually (which is unreasonable for the salt-minion and salt-master daemons, especially in production) or an agent is written for it.

Like all sdb modules, the keyring module requires a configuration profile to be configured in either the minion or master configuration file. This profile requires very little. In the example:

```
mykeyring:
  driver: keyring
  service: system
```

The `driver` refers to the keyring module, `service` refers to the service that will be used inside of keyring (which may be likened unto a database table) and `mykeyring` refers to the name that will appear in the URI:

```
password: sdb://mykeyring/mypassword
```

The underlying backend configuration must be configured via keyring itself. For examples and documentation, see keyring:

https://pypi.python.org/pypi/keyring

New in version 2014.1.4.

`salt.sdb.keyring_db.`**`get`**`(`*key*, *service=None*, *profile=None*`)`
Get a value from a keyring service

`salt.sdb.keyring_db.`**`set_`**`(`*key*, *value*, *service=None*, *profile=None*`)`
Set a key/value pair in a keyring service

21.17.8 salt.sdb.memcached

Memcached sdb Module

maintainer SaltStack

maturity New

depends python-memcached

platform all

This module allows access to memcached using an `sdb://` URI. This package is located at `https://pypi.python.org/pypi/python-memcached`.

Like all sdb modules, the memcached module requires a configuration profile to be configured in either the minion or master configuration file. This profile requires very little. In the example:

```
mymemcache:
  driver: memcached
  memcached.host: localhost
  memcached.port: 11211
```

The `driver` refers to the memcached module, `host` and `port` the memcached server to connect to (defaults to `localhost` and `11211`, and `mymemcached` refers to the name that will appear in the URI:

```
password: sdb://mymemcached/mykey
```

`salt.sdb.memcached.`**`get`**`(`*key, profile=None*`)`
Get a value from memcached

`salt.sdb.memcached.`**`set_`**`(`*key, value, profile=None*`)`
Set a key/value pair in memcached

21.17.9 salt.sdb.redis_sdb module

Redis SDB module

New in version 2019.2.0.

This module allows access to Redis using an `sdb://` URI.

Like all SDB modules, the Redis module requires a configuration profile to be configured in either the minion or master configuration file. This profile requires very little. For example:

```
sdb_redis:
  driver: redis
  host: 127.0.0.1
  port: 6379
  password: pass
  db: 1
```

The `driver` refers to the Redis module, all other options are optional. For option details see: https://redis-py.readthedocs.io/en/latest/.

`salt.sdb.redis_sdb.`**`delete`**`(`*key, profile=None*`)`
Delete a key from the Redis SDB.

`salt.sdb.redis_sdb.`**`get`**`(`*key, profile=None*`)`
Get a value from the Redis SDB.

`salt.sdb.redis_sdb.`**`set_`**`(`*key, value, profile=None*`)`
Set a value into the Redis SDB.

21.17.10 salt.sdb.rest module

Generic REST API SDB Module

maintainer SaltStack

maturity New

platform all

New in version 2015.8.0.

This module allows access to a REST interface using an `sdb://` URI.

Like all REST modules, the REST module requires a configuration profile to be configured in either the minion or master configuration file. This profile requires very little. In the example:

```
my-rest-api:
  driver: rest
  urls:
    url: https://api.github.com/
  keys:
    url: https://api.github.com/users/{{user}}/keys
    backend: requests
```

The `driver` refers to the REST module, and must be set to `rest` in order to use this driver. Each of the other items inside this block refers to a separate set of HTTP items, including a URL and any options associated with it. The options used here should match the options available in `salt.utils.http.query()`.

In order to call the `urls` item in the example, the following reference can be made inside a configuration file:

```
github_urls: sdb://my-rest-api/urls
```

Key/Value pairs may also be used with this driver, and merged into the URL using the configured renderer (`jinja`, by default). For instance, in order to use the `keys` item in the example, the following reference can be made:

```
github_urls: sdb://my-rest-api/keys?user=myuser
```

This will cause the following URL to actually be called:

```
https://api.github.com/users/myuser/keys
```

Key/Value pairs will NOT be passed through as GET data. If GET data needs to be sent to the URL, then it should be configured in the SDB configuration block. For instance:

```
another-rest-api:
  driver: rest
  user_data:
    url: https://api.example.com/users/
    params:
      user: myuser
```

`salt.sdb.rest.`**`get`**(*key*, *service=None*, *profile=None*)
  Get a value from the REST interface

`salt.sdb.rest.`**`query`**(*key*, *value=None*, *service=None*, *profile=None*)
  Get a value from the REST interface

`salt.sdb.rest.`**`set_`**(*key*, *value*, *service=None*, *profile=None*)
  Set a key/value pair in the REST interface

21.17.11 salt.sdb.sqlite3

SQLite sdb Module

maintainer SaltStack

maturity New

platform all

This module allows access to sqlite3 using an `sdb://` URI

Like all sdb modules, the sqlite3 module requires a configuration profile to be configured in either the minion or master configuration file. This profile requires very little. For example:

```
mysqlite:
  driver: sqlite3
  database: /tmp/sdb.sqlite
  table: sdb
  create_table: True
```

The `driver` refers to the sqlite3 module, `database` refers to the sqlite3 database file. `table` is the table within the db that will hold keys and values (defaults to `sdb`). The database and table will be created if they do not exist.

Advanced Usage:

Instead of a table name, it is possible to provide custom SQL statements to create the table(s) and get and set values.

```
myadvanced
  driver: sqlite3
  database: /tmp/sdb-advanced.sqlite
create_statements:
  - "CREATE TABLE advanced (a text, b text, c blob, d blob)"
  - "CREATE INDEX myidx ON advanced (a)"
get_query: "SELECT d FROM advanced WHERE a=:key"
set_query: "INSERT OR REPLACE INTO advanced (a, d) VALUES (:key, :value)"
```

`salt.sdb.sqlite3.`**`get`**(*key*, *profile=None*)
: Get a value from sqlite3

`salt.sdb.sqlite3.`**`set_`**(*key*, *value*, *profile=None*)
: Set a key/value pair in sqlite3

21.17.12 salt.sdb.tism module

tISM - the Immutable Secrets Manager SDB Module

maintainer tISM

maturity New

platform all

New in version 2017.7.0.

This module will decrypt PGP encrypted secrets against a tISM server.

```
sdb://<profile>/<encrypted secret>

sdb://tism/
↪hQEMAzJ+GfdAB3KqAQf9E3cyvrPEWR1sf1tMvH0nrJ0bZa9kDFLPxvtwAOqlRiNp0F7IpiiVRF+h+sW5Mb4ffB1TElMzQ+/
↪G5ptd6CjmgBfBsuGeajWmvLEi4lC6/
↪9v1rYGjjLeOCCcN4Dl5AHlxUUaSrxB8akTDvSAnPvGhtRTZqDlltl5UEHsyYXM8RaeCrBw5Or1yvC9Ctx2saVp3xmALQvyhzkU
↪H30L/r50+CBkuI/u4M2pXDcMYsvvt4ajCbJn91qaQ7BDI=
```

A profile must be setup in the minion configuration or pillar. If you want to use sdb in a runner or pillar you must also place a profile in the master configuration.

```
tism:
  driver: tism
  url: https://my.tismd:8080/decrypt
  token: eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
↪eyJhZG1pbiI6MSwiZXhwIjoxNTg1MTExNDYwLCJqdGkiOiI3NnA5cWNiMWdtdmw4Iiwia2V5cyI6WyJBTEwiXX0.
↪RtAhG6Uorf5xnSf4Ya_GwJnoHkCsql4r1_hiOeDSLzo
```

(continues on next page)

(continued from previous page)

```
```

`salt.sdb.tism.`**`get`**(*key*, *service=None*, *profile=None*)
Get a decrypted secret from the tISMd API

21.17.13 salt.sdb.vault module

Vault SDB Module

maintainer SaltStack

maturity New

platform all

New in version 2016.11.0.

This module allows access to Hashicorp Vault using an `sdb://` URI.

Base configuration instructions are documented in the execution module docs. Below are noted extra configuration required for the sdb module, but the base configuration must also be completed.

Like all sdb modules, the vault module requires a configuration profile to be configured in either the minion configuration file or a pillar. This profile requires only setting the `driver` parameter to `vault`:

```
myvault:
  driver: vault
```

Once configured you can access data using a URL such as:

```
password: sdb://myvault/secret/passwords/mypassword
```

In this URL, `myvault` refers to the configuration profile, `secret/passwords` is the path where the data resides, and `mypassword` is the key of the data to return.

The above URI is analogous to running the following vault command:

```
$ vault read -field=mypassword secret/passwords
```

`salt.sdb.vault.`**`get`**(*key*, *profile=None*)
Get a value from the vault service

`salt.sdb.vault.`**`set_`**(*key*, *value*, *profile=None*)
Set a key/value pair in the vault service

21.17.14 salt.sdb.yaml module

Pull sdb values from a YAML file

maintainer SaltStack

maturity New

platform all

New in version 2017.7.0.

Configuration:

```
my-yaml-file:
  driver: yaml
  files:
    - /path/to/foo.yaml
    - /path/to/bar.yaml
```

The files are merged together and the result is searched using the same mechanism Salt uses for searching Grains and Pillar data structures.

Optional configuration:

```
my-yaml-file:
  driver: yaml
  files:
    - /path/to/foo.yaml
    - /path/to/bar.yaml
  merge:
    strategy: smart
    merge_list: false
  gpg: true
```

Setting the `gpg` option to `true` (default is `false`) will decrypt embedded GPG-encrypted data using the *GPG renderer*.

`salt.sdb.yaml.`**`get`**`(`*key*, *profile=None*`)`

Get a value from the dictionary

`salt.sdb.yaml.`**`set_`**`(`**args*, ***kwargs*`)`

Setting a value is not supported; edit the YAML files directly

21.18 serializer modules

configparser	salt.serializers.configparser
json	salt.serializers.json
msgpack	salt.serializers.msgpack
python	salt.serializers.python
toml	salt.serializers.toml
yaml	salt.serializers.yaml
yamlex	salt.serializers.yamlex

21.18.1 salt.serializers.configparser module

salt.serializers.configparser

New in version 2016.3.0.

Implements a configparser serializer.

`salt.serializers.configparser.`**`deserialize`**`(`*stream_or_string*, ***options*`)`

Deserialize any string or stream like object into a Python data structure.

Parameters

- **`stream_or_string`** – stream or string to deserialize.
- **`options`** – options given to lower configparser module.

`salt.serializers.configparser.`**`serialize`**`(`*obj*, ***options*`)`

Serialize Python data to a configparser formatted string or file.

Parameters

- **`obj`** – the data structure to serialize
- **`options`** – options given to lower configparser module.

21.18.2 salt.serializers.json

salt.serializers.json

Implements JSON serializer.

It's just a wrapper around json (or simplejson if available).

`salt.serializers.json.`**`deserialize`**`(`*stream_or_string*, ***options*`)`

Deserialize any string or stream like object into a Python data structure.

Parameters

- **`stream_or_string`** – stream or string to deserialize.
- **`options`** – options given to lower json/simplejson module.

`salt.serializers.json.`**`serialize`**`(`*obj*, ***options*`)`

Serialize Python data to JSON.

Parameters

- **`obj`** – the data structure to serialize
- **`options`** – options given to lower json/simplejson module.

21.18.3 salt.serializers.msgpack

salt.serializers.msgpack

Implements MsgPack serializer.

`salt.serializers.msgpack.`**`deserialize`**`(`*stream_or_string*, ***options*`)`

Deserialize any string of stream like object into a Python data structure.

Parameters

- **`stream_or_string`** – stream or string to deserialize.
- **`options`** – options given to lower msgpack module.

`salt.serializers.msgpack.`**`serialize`**`(`*obj*, ***options*`)`

Serialize Python data to MsgPack.

Parameters

- **`obj`** – the data structure to serialize
- **`options`** – options given to lower msgpack module.

21.18.4 salt.serializers.python module

salt.serializers.python

New in version 2016.3.0.

Implements a Python serializer (via pprint.format)

`salt.serializers.python.`**`serialize`**`(`*obj*, ***options*`)`

Serialize Python data to a Python string representation (via pprint.format)

Parameters
- **`obj`** – the data structure to serialize
- **`options`** – options given to pprint.format

21.18.5 salt.serializers.toml

salt.serializers.toml

Implements TOML serializer.

It's just a wrapper around pytoml module.

`salt.serializers.toml.`**`deserialize`**`(`*stream_or_string*, ***options*`)`

Deserialize from TOML into Python data structure.

Parameters
- **`stream_or_string`** – toml stream or string to deserialize.
- **`options`** – options given to lower pytoml module.

`salt.serializers.toml.`**`serialize`**`(`*obj*, ***options*`)`

Serialize Python data to TOML.

Parameters
- **`obj`** – the data structure to serialize.
- **`options`** – options given to lower pytoml module.

21.18.6 salt.serializers.yaml

salt.serializers.yaml

Implements YAML serializer.

Underneath, it is based on pyyaml and use the safe dumper and loader. It also use C bindings if they are available.

`salt.serializers.yaml.`**`deserialize`**`(`*stream_or_string*, ***options*`)`

Deserialize any string of stream like object into a Python data structure.

Parameters
- **`stream_or_string`** – stream or string to deserialize.
- **`options`** – options given to lower yaml module.

`salt.serializers.yaml.`**`serialize`**`(`*obj*, ***options*`)`

Serialize Python data to YAML.

Parameters
- **`obj`** – the data structure to serialize
- **`options`** – options given to lower yaml module.

21.18.7 salt.serializers.yamlex

salt.serializers.yamlex

YAMLEX is a format that allows for things like sls files to be more intuitive.

It's an extension of YAML that implements all the salt magic: - it implies omap for any dict like. - it implies that string like data are str, not unicode - …

For example, the file *states.sls* has this contents:

```
foo:
  bar: 42
  baz: [1, 2, 3]
```

The file can be parsed into Python like this

```
from salt.serializers import yamlex

with open('state.sls', 'r') as stream:
    obj = yamlex.deserialize(stream)
```

Check that `obj` is an OrderedDict

```
from salt.utils.odict import OrderedDict

assert isinstance(obj, dict)
assert isinstance(obj, OrderedDict)
```

yamlex *__repr__* and *__str__* objects' methods render YAML understandable string. It means that they are template friendly.

```
print '{0}'.format(obj)
```

returns:

```
{foo: {bar: 42, baz: [1, 2, 3]}}
```

and they are still valid YAML:

```
from salt.serializers import yaml
yml_obj = yaml.deserialize(str(obj))
assert yml_obj == obj
```

yamlex implements also custom tags:

!aggregate

this tag allows structures aggregation.

For example:

```
placeholder: !aggregate foo
placeholder: !aggregate bar
placeholder: !aggregate baz
```

is rendered as

```
placeholder: [foo, bar, baz]
```

!reset

this tag flushes the computing value.

```
placeholder: {!aggregate foo: {foo: 42}}
placeholder: {!aggregate foo: {bar: null}}
!reset placeholder: {!aggregate foo: {baz: inga}}
```

is roughly equivalent to

```
placeholder: {!aggregate foo: {baz: inga}}
```

Document is defacto an aggregate mapping.

`salt.serializers.yamlex.`**`deserialize`**`(`*stream_or_string, **options*`)`

Deserialize any string of stream like object into a Python data structure.

Parameters

- **`stream_or_string`** – stream or string to deserialize.
- **`options`** – options given to lower yaml module.

`salt.serializers.yamlex.`**`serialize`**`(`*obj, **options*`)`

Serialize Python data to YAML.

Parameters

- **`obj`** – the data structure to serialize
- **`options`** – options given to lower yaml module.

21.19 state modules

acme	ACME / Let's Encrypt certificate management state
alias	Configuration of email aliases
alternatives	Configuration of the alternatives system
ansiblegate	Execution of Ansible modules from within states
apache	Apache state
apache_conf	Manage Apache Confs
apache_module	Manage Apache Modules
apache_site	Manage Apache Sites
aptpkg	Package management operations specific to APT- and DEB-based systems
archive	Extract an archive
artifactory	This state downloads artifacts from artifactory.
at	Configuration disposable regularly scheduled tasks for at.
augeas	Configuration management using Augeas
aws_sqs	Manage SQS Queues
azurearm_compute	Azure (ARM) Compute State Module
azurearm_network	Azure (ARM) Network State Module
azurearm_resource	Azure (ARM) Resource State Module
beacon	Management of the Salt beacons
bigip	A state module designed to enforce load-balancing configurations for F5 Big-IP entities.
blockdev	Management of Block Devices
boto3_elasticache	Manage Elasticache with boto3
boto3_elasticsearch	Manage Elasticsearch Service
boto3_route53	Manage Route53 records with Boto 3
boto_apigateway	Manage Apigateway Rest APIs
boto_asg	Manage Autoscale Groups
boto_cfn	Connection module for Amazon Cloud Formation
boto_cloudfront	Manage CloudFront distributions
boto_cloudtrail	Manage CloudTrail Objects
boto_cloudwatch_alarm	Manage Cloudwatch alarms

Continued on next page

Table 19 – continued from previous page

`boto_cloudwatch_event`	Manage CloudTrail Objects
`boto_cognitoidentity`	Manage CognitoIdentity Functions
`boto_datapipeline`	Manage Data Pipelines
`boto_dynamodb`	Manage DynamoDB Tables
`boto_ec2`	Manage EC2
`boto_elasticache`	Manage Elasticache
`boto_elasticsearch_domain`	Manage Elasticsearch Domains
`boto_elb`	Manage ELBs
`boto_elbv2`	Manage AWS Application Load Balancer
`boto_iam`	Manage IAM objects
`boto_iam_role`	Manage IAM roles
`boto_iot`	Manage IoT Objects
`boto_kinesis`	Manage Kinesis Streams
`boto_kms`	Manage KMS keys, key policies and grants.
`boto_lambda`	Manage Lambda Functions
`boto_lc`	Manage Launch Configurations
`boto_rds`	Manage RDSs
`boto_route53`	Manage Route53 records
`boto_s3_bucket`	Manage S3 Buckets
`boto_secgroup`	Manage Security Groups
`boto_sns`	Manage SNS Topics
`boto_sqs`	Manage SQS Queues
`boto_vpc`	Manage VPCs
`bower`	Installation of Bower Packages
`cabal`	Installation of Cabal Packages
`ceph`	Manage ceph with salt.
`chef`	Execute Chef client runs
`chocolatey`	Manage Chocolatey package installs ..
`chronos_job`	Configure Chronos jobs via a salt proxy.
`cimc`	A state module to manage Cisco UCS chassis devices.
`cisconso`	State module for Cisco NSO Proxy minions
`cloud`	Using states instead of maps to deploy clouds
`cmd`	Execution of arbitrary commands
`composer`	Installation of Composer Packages
`cron`	Management of cron, the Unix command scheduler
`csf`	CSF Ip tables management
`cyg`	Installation of Cygwin packages.
`ddns`	Dynamic DNS updates
`debconfmod`	Management of debconf selections
`dellchassis`	Manage chassis via Salt Proxies.
`disk`	Disk monitoring state
`docker_container`	Management of Docker containers
`docker_image`	Management of Docker images
`docker_network`	Management of Docker networks
`docker_volume`	Management of Docker volumes
`drac`	Management of Dell DRAC
`elasticsearch`	State module to manage Elasticsearch.
`elasticsearch_index`	State module to manage Elasticsearch indices
`elasticsearch_index_template`	State module to manage Elasticsearch index templates

Continued on next page

Table 19 – continued from previous page

`environ`	Support for getting and setting the environment variables of the current salt process.
`eselect`	Management of Gentoo configuration using eselect
`etcd_mod`	Manage etcd Keys
`ethtool`	Configuration of network device
`esxi`	Manage VMware ESXi Hosts.
`event`	Send events through Salt's event system during state runs
`file`	Operations on regular files, special files, directories, and symlinks
`firewall`	State to check firewall configurations
`firewalld`	Management of firewalld
`gem`	Installation of Ruby modules packaged as gems
`git`	States to manage git repositories and git configuration
`github`	Github User State Module
`glance`	Managing Images in OpenStack Glance
`glance_image`	Management of OpenStack Glance Images
`glusterfs`	Manage GlusterFS pool.
`gnomedesktop`	Configuration of the GNOME desktop
`gpg`	Management of the GPG keychains
`grafana`	Manage Grafana Dashboards
`grafana4_dashboard`	Manage Grafana v4.0 Dashboards
`grafana4_datasource`	Manage Grafana v4.0 data sources
`grafana4_org`	Manage Grafana v4.0 orgs
`grafana4_user`	Manage Grafana v4.0 users
`grafana_dashboard`	Manage Grafana v2.0 Dashboards
`grafana_datasource`	Manage Grafana v2.0 data sources
`grains`	Manage grains on the minion
`group`	Management of user groups
`heat`	Management of Heat
`hg`	Interaction with Mercurial repositories
`host`	Management of addresses and names in hosts file
`http`	HTTP monitoring states
`icinga2`	Icinga2 state
`ifttt`	Trigger an event in IFTTT
`incron`	Management of incron, the inotify cron
`influxdb08_database`	Management of Influxdb 0.8 databases
`influxdb08_user`	Management of InfluxDB 0.8 users
`influxdb_continuous_query`	Management of Influxdb continuous queries
`influxdb_database`	Management of Influxdb databases
`influxdb_retention_policy`	Management of Influxdb retention policies
`influxdb_user`	Management of InfluxDB users
`infoblox_a`	Infoblox A record managment.
`infoblox_cname`	Infoblox CNAME managment.
`infoblox_host_record`	Infoblox host record managment.
`infoblox_range`	Infoblox host record managment.
`ini_manage`	Manage ini files
`ipmi`	Manage IPMI devices over LAN
`ipset`	Management of ipsets
`iptables`	Management of iptables

Continued on next page

Table 19 – continued from previous page

`jboss7`	Manage JBoss 7 Application Server via CLI interface
`jenkins`	Management of Jenkins
`junos`	State modules to interact with Junos devices.
`kapacitor`	Kapacitor state module.
`kernelpkg`	Manage kernel packages and active kernel version
`keyboard`	Management of keyboard layouts
`keystone`	Management of Keystone users
`keystone_domain`	Management of OpenStack Keystone Domains
`keystone_endpoint`	Management of OpenStack Keystone Endpoints
`keystone_group`	Management of OpenStack Keystone Groups
`keystone_project`	Management of OpenStack Keystone Projects
`keystone_role`	Management of OpenStack Keystone Roles
`keystone_role_grant`	Management of OpenStack Keystone Role Grants
`keystone_service`	Management of OpenStack Keystone Services
`keystone_user`	Management of OpenStack Keystone Users
`keystore`	State management of a java keystore
`kmod`	Loading and unloading of kernel modules
`kubernetes`	Manage kubernetes resources as salt states
`layman`	Management of Gentoo Overlays using layman
`ldap`	Manage entries in an LDAP database
`libcloud_dns`	Manage DNS records and zones using libcloud
`libcloud_loadbalancer`	Apache Libcloud Load Balancer State
`libcloud_storage`	Apache Libcloud Storage State
`linux_acl`	Linux File Access Control Lists
`locale`	Management of languages/locales
`logadm`	Management of logs using Solaris logadm.
`logrotate`	Module for managing logrotate.
`loop`	Loop state
`lvm`	Management of Linux logical volumes
`lvs_server`	Management of LVS (Linux Virtual Server) Real Server
`lvs_service`	Management of LVS (Linux Virtual Server) Service
`lxc`	Manage Linux Containers
`mac_assistive`	Allows you to manage assistive access on macOS minions with 10.9+
`macdefaults`	Writing/reading defaults from a macOS minion
`mac_keychain`	Installing of certificates to the keychain
`macpackage`	Installing of mac pkg files
`mac_xattr`	Allows you to manage extended attributes on files or directories
`makeconf`	Management of Gentoo make.conf
`marathon_app`	Configure Marathon apps via a salt proxy.
`mdadm_raid`	Managing software RAID with mdadm
`memcached`	States for Management of Memcached Keys
`modjk`	State to control Apache modjk
`modjk_worker`	Manage modjk workers
`module`	Execution of Salt modules from within states
`mongodb_database`	Management of MongoDB Databases
`mongodb_user`	Management of MongoDB Users
`monit`	Monit state
`mount`	Mounting of filesystems

Continued on next page

Table 19 – continued from previous page

msteams	Send a message card to Microsoft Teams
mysql_database	Management of MySQL databases (schemas)
mysql_grants	Management of MySQL grants (user permissions)
mysql_query	Execution of MySQL queries
mysql_user	Management of MySQL users
netacl	Network ACL
netconfig	Network Config
netntp	Network NTP
netsnmp	Network SNMP
netusers	Network Users
network	Configuration of network interfaces
net_napalm_yang	NAPALM YANG state
neutron_network	Management of OpenStack Neutron Networks
neutron_secgroup	Management of OpenStack Neutron Security Groups
neutron_secgroup_rule	Management of OpenStack Neutron Security Group Rules
neutron_subnet	Management of OpenStack Neutron Subnets
nfs_export	Management of NFS exports
nftables	Management of nftables
npm	Installation of NPM Packages
ntp	Management of NTP servers
nxos	State module for Cisco NX OS Switches Proxy minions
openstack_config	Manage OpenStack configuration file settings.
openvswitch_bridge	Management of Open vSwitch bridges.
openvswitch_port	Management of Open vSwitch ports.
opsgenie	Create/Close an alert in OpsGenie
pagerduty	Create an Event in PagerDuty
pagerduty_escalation_policy	Manage PagerDuty escalation policies.
pagerduty_schedule	Manage PagerDuty schedules.
pagerduty_service	Manage PagerDuty services
pagerduty_user	Manage PagerDuty users.
panos	A state module to manage Palo Alto network devices.
pcs	Management of Pacemaker/Corosync clusters with PCS
pecl	Installation of PHP Extensions Using pecl
pdbedit	Manage accounts in Samba's passdb using pdbedit
pip_state	Installation of Python Packages Using pip
pkg	Installation of packages using OS package managers such as yum or apt-get
pkgbuild	The pkgbuild state is the front of Salt package building backend.
pkgng	Manage package remote repo using FreeBSD pkgng
pkgrepo	Management of APT/DNF/YUM/Zypper package repos
portage_config	Management of Portage package configuration on Gentoo
ports	Manage software from FreeBSD ports
postgres_cluster	Management of PostgreSQL clusters
postgres_database	Management of PostgreSQL databases
postgres_extension	Management of PostgreSQL extensions
postgres_group	Management of PostgreSQL groups (roles)

Continued on next page

Table 19 – continued from previous page

`postgres_initdb`	Initialization of PostgreSQL data directory
`postgres_language`	Management of PostgreSQL languages
`postgres_privileges`	Management of PostgreSQL Privileges
`postgres_schema`	Management of PostgreSQL schemas
`postgres_tablespace`	Management of PostgreSQL tablespace
`postgres_user`	Management of PostgreSQL users (roles)
`powerpath`	Powerpath configuration support
`probes`	Network Probes
`process`	Process Management
`proxy`	Allows you to manage proxy settings on minions
`pushover`	Send a message to PushOver
`pyenv`	Managing python installations with pyenv
`pyrax_queues`	Manage Rackspace Queues
`quota`	Management of POSIX Quotas
`rabbitmq_cluster`	Manage RabbitMQ Clusters
`rabbitmq_plugin`	Manage RabbitMQ Plugins
`rabbitmq_policy`	Manage RabbitMQ Policies
`rabbitmq_upstream`	Manage RabbitMQ Upstreams
`rabbitmq_user`	Manage RabbitMQ Users
`rabbitmq_vhost`	Manage RabbitMQ Virtual Hosts
`rbac_solaris`	Management of Solaris RBAC
`rbenv`	Managing Ruby installations with rbenv
`rdp`	Manage RDP Service on Windows servers
`redismod`	Management of Redis server
`reg`	Manage the Windows registry
`rsync`	State to synchronize files and directories with rsync.
`rvm`	Managing Ruby installations and gemsets with Ruby Version Manager (RVM)
`salt_proxy`	Salt proxy state
`saltmod`	Control the Salt command interface
`saltutil`	Saltutil State
`schedule`	Management of the Salt scheduler
`selinux`	Management of SELinux rules
`serverdensity_device`	Monitor Server with Server Density
`service`	Starting or restarting of services and daemons
`slack`	Send a message to Slack
`smartos`	Management of SmartOS Standalone Compute Nodes
`smtp`	Sending Messages via SMTP
`snapper`	Managing implicit state and baselines using snapshots
`solrcloud`	States for solrcloud alias and collection configuration
`splunk`	Splunk User State Module
`splunk_search`	Splunk Search State Module
`sqlite3`	Management of SQLite3 databases
`ssh_auth`	Control of entries in SSH authorized_key files
`ssh_known_hosts`	Control of SSH known_hosts entries
`stateconf`	Stateconf System
`status`	Minion status monitoring
`statuspage`	StatusPage
`supervisord`	Interaction with the Supervisor daemon
`svn`	Manage SVN repositories

Continued on next page

Table 19 – continued from previous page

`sysctl`	Configuration of the kernel using sysctl
`syslog_ng`	State module for syslog_ng
`sysrc`	State to work with sysrc
`telemetry_alert`	Manage Telemetry alert configurations
`test`	Test States
`testinframod`	
`timezone`	Management of timezones
`tls`	Enforce state for SSL/TLS
`tomcat`	Manage Apache Tomcat web applications
`trafficserver`	Control Apache Traffic Server
`tuned`	Interface to Red Hat tuned-adm module
`uptime`	Monitor Web Server with Uptime
`user`	Management of user accounts
`vagrant`	Manage Vagrant VMs
`vault`	States for managing Hashicorp Vault.
`vbox_guest`	VirtualBox Guest Additions installer state
`victorops`	Create an Event in VictorOps
`virt`	Manage virt
`virtualenv_mod`	Setup of Python virtualenv sandboxes.
`webutil`	Support for htpasswd module.
`win_certutil`	Installing of certificates to the Windows Certificate Manager
`win_dacl`	Windows Object Access Control Lists
`win_dism`	Installing of Windows features using DISM
`win_dns_client`	Module for configuring DNS Client on Windows systems
`win_firewall`	State for configuring Windows Firewall
`win_iis`	Microsoft IIS site management
`win_lgpo`	Manage Windows Local Group Policy
`win_license`	Installation and activation of windows licenses
`win_network`	Configuration of network interfaces on Windows hosts
`win_path`	Manage the Windows System PATH
`win_pki`	Microsoft certificate management via the Pki PowerShell module.
`win_powercfg`	This module allows you to control the power settings of a windows minion via powercfg.
`win_servermanager`	Manage Windows features via the ServerManager powershell module.
`win_smtp_server`	Module for managing IIS SMTP server configuration on Windows servers.
`win_snmp`	Module for managing SNMP service settings on Windows servers.
`win_system`	Management of Windows system information
`win_wua`	Installation of Windows Updates using the Windows Update Agent
`win_wusa`	Microsoft Updates (KB) Management
`winrepo`	Manage Windows Package Repository
`x509`	Manage X509 Certificates
`xml`	XML Manager
`xmpp`	Sending Messages over XMPP

Continued on next page

Table 19 – continued from previous page

zabbix_action	New in version 2017.7.
zabbix_host	Management of Zabbix hosts.
zabbix_hostgroup	Management of Zabbix host groups.
zabbix_mediatype	Management of Zabbix mediatypes.
zabbix_template	New in version 2017.7.
zabbix_user	Management of Zabbix users.
zabbix_usergroup	Management of Zabbix user groups.
zabbix_valuemap	New in version 2017.7.
zcbuildout	Management of zc.buildout
zenoss	State to manage monitoring in Zenoss.
zk_concurrency	Control concurrency of steps within state execution using zookeeper
zfs	States for managing zfs datasets
zone	Management of Solaris Zones
zookeeper	**depends** kazoo
zpool	States for managing zpools

21.19.1 salt.states.acme module

ACME / Let's Encrypt certificate management state

See also the module documentation

```
reload-gitlab:
  cmd.run:
    - name: gitlab-ctl hup

dev.example.com:
  acme.cert:
    - aliases:
      - gitlab.example.com
    - email: acmemaster@example.com
    - webroot: /opt/gitlab/embedded/service/gitlab-rails/public
    - renew: 14
    - fire_event: acme/dev.example.com
    - onchanges_in:
      - cmd: reload-gitlab
```

`salt.states.acme.`**`cert`**(*name, aliases=None, email=None, webroot=None, test_cert=False, renew=None, keysize=None, server=None, owner='root', group='root', mode='0640', certname=None, preferred_challenges=None, tls_sni_01_port=None, tls_sni_01_address=None, http_01_port=None, http_01_address=None, dns_plugin=None, dns_plugin_credentials=None*)

Obtain/renew a certificate from an ACME CA, probably Let's Encrypt.

Parameters

- **`name`** – Common Name of the certificate (DNS name of certificate)
- **`aliases`** – subjectAltNames (Additional DNS names on certificate)
- **`email`** – e-mail address for interaction with ACME provider

- **webroot** – True or a full path to webroot. Otherwise use standalone mode
- **test_cert** – Request a certificate from the Happy Hacker Fake CA (mutually exclusive with 'server')
- **renew** – True/'force' to force a renewal, or a window of renewal before expiry in days
- **keysize** – RSA key bits
- **server** – API endpoint to talk to
- **owner** – owner of the private key file
- **group** – group of the private key file
- **mode** – mode of the private key file
- **certname** – Name of the certificate to save
- **preferred_challenges** – A sorted, comma delimited list of the preferred challenge to use during authorization with the most preferred challenge listed first.
- **tls_sni_01_port** – Port used during tls-sni-01 challenge. This only affects the port Certbot listens on. A conforming ACME server will still attempt to connect on port 443.
- **tls_sni_01_address** – The address the server listens to during tls-sni-01 challenge.
- **http_01_port** – Port used in the http-01 challenge. This only affects the port Certbot listens on. A conforming ACME server will still attempt to connect on port 80.
- **https_01_address** – The address the server listens to during http-01 challenge.
- **dns_plugin** – Name of a DNS plugin to use (currently only 'cloudflare')
- **dns_plugin_credentials** – Path to the credentials file if required by the specified DNS plugin

21.19.2 salt.states.alias

Configuration of email aliases

The mail aliases file can be managed to contain definitions for specific email aliases:

```
username:
  alias.present:
    - target: user@example.com
```

```
thomas:
  alias.present:
    - target: thomas@example.com
```

The default alias file is set to `/etc/aliases`, as defined in Salt's *config execution module*. To change the alias file from the default location, set the following in your minion config:

```
aliases.file: /my/alias/file
```

`salt.states.alias.`**`absent`**(*name*)

Ensure that the named alias is absent

name The alias to remove

`salt.states.alias.`**`present`**(*name, target*)

Ensures that the named alias is present with the given target or list of targets. If the alias exists but the target differs from the previous entry, the target(s) will be overwritten. If the alias does not exist, the alias will be created.

name The local user/address to assign an alias to

target The forwarding address

21.19.3 salt.states.alternatives

Configuration of the alternatives system

Control the alternatives system

```
{% set my_hadoop_conf = '/opt/hadoop/conf' %}

{{ my_hadoop_conf }}:
  file.directory

hadoop-0.20-conf:
  alternatives.install:
    - name: hadoop-0.20-conf
    - link: /etc/hadoop-0.20/conf
    - path: {{ my_hadoop_conf }}
    - priority: 30
    - require:
      - file: {{ my_hadoop_conf }}

hadoop-0.20-conf:
  alternatives.remove:
    - name: hadoop-0.20-conf
    - path: {{ my_hadoop_conf }}
```

`salt.states.alternatives.`**`auto`**`(`*name*`)`

New in version 0.17.0.

Instruct alternatives to use the highest priority path for <name>
name is the master name for this link group (e.g. pager)

`salt.states.alternatives.`**`install`**`(`*name, link, path, priority*`)`

Install new alternative for defined <name>
name is the master name for this link group (e.g. pager)
link is the symlink pointing to /etc/alternatives/<name>. (e.g. /usr/bin/pager)
path is the location of the new alternative target. NB: This file / directory must already exist. (e.g. /usr/bin/less)
priority is an integer; options with higher numbers have higher priority in automatic mode.

`salt.states.alternatives.`**`remove`**`(`*name, path*`)`

Removes installed alternative for defined <name> and <path> or fallback to default alternative, if some defined before.
name is the master name for this link group (e.g. pager)
path is the location of one of the alternative target files. (e.g. /usr/bin/less)

`salt.states.alternatives.`**`set_`**`(`*name, path*`)`

New in version 0.17.0.

Sets alternative for <name> to <path>, if <path> is defined as an alternative for <name>.
name is the master name for this link group (e.g. pager)
path is the location of one of the alternative target files. (e.g. /usr/bin/less)

```
foo:
  alternatives.set:
    - path: /usr/bin/foo-2.0
```

21.19.4 salt.states.ansiblegate

Execution of Ansible modules from within states

With *ansible.call* these states allow individual Ansible module calls to be made via states. To call an Ansible module function use a `module.run` state:

```
some_set_of_tasks:
  ansible:
    - system.ping
    - packaging.os.zypper
      - name: emacs
      - state: installed
```

class `salt.states.ansiblegate.`**`AnsibleState`**

Ansible state caller.

`get_args`(*argset*)

Get args and kwargs from the argset.

Parameters **`argset`** –

Returns

`salt.states.ansiblegate.`**`playbooks`**(*name*, *rundir=None*, *git_repo=None*, *git_kwargs=None*, *ansible_kwargs=None*)

Run Ansible Playbooks

Parameters

- **`name`** – path to playbook. This can be relative to rundir or the git repo
- **`rundir`** – location to run ansible-playbook from.
- **`git_repo`** – git repository to clone for ansible playbooks. This is cloned using the *git.latest* state, and is cloned to the *rundir* if specified, otherwise it is clone to the *cache_dir*
- **`git_kwargs`** – extra kwargs to pass to *git.latest* state module besides the *name* and *target*
- **`ansible_kwargs`** – extra kwargs to pass to *ansible.playbooks* execution module besides the *name* and *target*

Returns Ansible playbook output.

```
run nginx install:
  ansible.playbooks:
    - name: install.yml
    - git_repo: git://github.com/gituser/playbook.git
    - git_kwargs:
        rev: master
```

21.19.5 salt.states.apache

Apache state

New in version 2014.7.0.

Allows for inputting a yaml dictionary into a file for apache configuration files.

The variable `this` is special and signifies what should be included with the above word between angle brackets (<>).

```
/etc/httpd/conf.d/website.com.conf:
  apache.configfile:
    - config:
      - VirtualHost:
```

(continues on next page)

(continued from previous page)

```
          this: '*:80'
          ServerName:
            - website.com
          ServerAlias:
            - www.website.com
            - dev.website.com
          ErrorLog: logs/website.com-error_log
          CustomLog: logs/website.com-access_log combined
          DocumentRoot: /var/www/vhosts/website.com
          Directory:
            this: /var/www/vhosts/website.com
            Order: Deny,Allow
            Deny from: all
            Allow from:
              - 127.0.0.1
              - 192.168.100.0/24
            Options:
              - Indexes
              - FollowSymlinks
            AllowOverride: All
```

Changed in version 2018.3.

Allows having the same section container multiple times (e.g. <Directory /path/to/dir>).

YAML structure stays the same only replace dictionary with a list.

When a section container does not have mandatory attribute, such as <Else>, it still needs keyword `this` with empty string (or "" if nicer output is required - without space).

```
/etc/httpd/conf.d/website.com.conf:
  apache.configfile:
    - config:
      - VirtualHost:
        - this: '*:80'
        - ServerName:
          - website.com
        - DocumentRoot: /var/www/vhosts/website.com
        - Directory:
            this: /var/www/vhosts/website.com
            Order: Deny,Allow
            Deny from: all
            Allow from:
              - 127.0.0.1
              - 192.168.100.0/24
            Options:
              - Indexes
              - FollowSymlinks
            AllowOverride: All
        - Directory:
          - this: /var/www/vhosts/website.com/private
          - Order: Deny,Allow
          - Deny from: all
          - Allow from:
            - 127.0.0.1
            - 192.168.100.0/24
          - If:
```

(continues on next page)

(continued from previous page)

```
                this: some condition
                do: something
            - Else:
                this:
                do: something else
            - Else:
                this: ""
                do: another thing
```

salt.states.apache.**configfile**(*name*, *config*)

21.19.6 salt.states.apache_conf module

Manage Apache Confs

New in version 2016.3.0.

Enable and disable apache confs.

```
Enable security conf:
  apache_conf.enabled:
    - name: security

Disable security conf:
  apache_conf.disabled:
    - name: security
```

salt.states.apache_conf.**disabled**(*name*)
Ensure an Apache conf is disabled.
name Name of the Apache conf

salt.states.apache_conf.**enabled**(*name*)
Ensure an Apache conf is enabled.
name Name of the Apache conf

21.19.7 salt.states.apache_module

Manage Apache Modules

New in version 2014.7.0.

Enable and disable apache modules.

```
Enable cgi module:
  apache_module.enabled:
    - name: cgi

Disable cgi module:
  apache_module.disabled:
    - name: cgi
```

salt.states.apache_module.**disabled**(*name*)
Ensure an Apache module is disabled.

New in version 2016.3.0.
name Name of the Apache module

`salt.states.apache_module.`**`enabled`**`(`*name*`)`

Ensure an Apache module is enabled.

New in version 2016.3.0.

name Name of the Apache module

21.19.8 salt.states.apache_site module

Manage Apache Sites

New in version 2016.3.0.

Enable and disable apache sites.

```
Enable default site:
  apache_site.enabled:
    - name: default

Disable default site:
  apache_site.disabled:
    - name: default
```

`salt.states.apache_site.`**`disabled`**`(`*name*`)`

Ensure an Apache site is disabled.

name Name of the Apache site

`salt.states.apache_site.`**`enabled`**`(`*name*`)`

Ensure an Apache site is enabled.

name Name of the Apache site

21.19.9 salt.states.aptpkg

Package management operations specific to APT- and DEB-based systems

`salt.states.aptpkg.`**`held`**`(`*name*`)`

Set package in 'hold' state, meaning it will not be upgraded.

name The name of the package, e.g., 'tmux'

21.19.10 salt.states.archive

Extract an archive

New in version 2014.1.0.

`salt.states.archive.`**`extracted`**`(`*name, source, source_hash=None, source_hash_name=None, source_hash_update=False, skip_files_list_verify=False, skip_verify=False, password=None, options=None, list_options=None, force=False, overwrite=False, clean=False, clean_parent=False, user=None, group=None, if_missing=None, trim_output=False, use_cmd_unzip=None, extract_perms=True, enforce_toplevel=True, enforce_ownership_on=None, archive_format=None, **kwargs*`)`

New in version 2014.1.0.

Changed in version 2016.11.0: This state has been rewritten. Some arguments are new to this release and will not be available in the 2016.3 release cycle (and earlier). Additionally, the **ZIP Archive Handling** section below applies specifically to the 2016.11.0 release (and newer).

Ensure that an archive is extracted to a specific directory.

Important: Changes for 2016.11.0

In earlier releases, this state would rely on the `if_missing` argument to determine whether or not the archive needed to be extracted. When this argument was not passed, then the state would just assume `if_missing` is the same as the `name` argument (i.e. the parent directory into which the archive would be extracted).

This caused a number of annoyances. One such annoyance was the need to know beforehand a path that would result from the extraction of the archive, and setting `if_missing` to that directory, like so:

```
extract_myapp:
  archive.extracted:
    - name: /var/www
    - source: salt://apps/src/myapp-16.2.4.tar.gz
    - user: www
    - group: www
    - if_missing: /var/www/myapp-16.2.4
```

If `/var/www` already existed, this would effectively make `if_missing` a required argument, just to get Salt to extract the archive.

Some users worked around this by adding the top-level directory of the archive to the end of the `name` argument, and then used `--strip` or `--strip-components` to remove that top-level dir when extracting:

```
extract_myapp:
  archive.extracted:
    - name: /var/www/myapp-16.2.4
    - source: salt://apps/src/myapp-16.2.4.tar.gz
    - user: www
    - group: www
```

With the rewrite for 2016.11.0, these workarounds are no longer necessary. `if_missing` is still a supported argument, but it is no longer required. The equivalent SLS in 2016.11.0 would be:

```
extract_myapp:
  archive.extracted:
    - name: /var/www
    - source: salt://apps/src/myapp-16.2.4.tar.gz
    - user: www
    - group: www
```

Salt now uses a function called `archive.list` to get a list of files/directories in the archive. Using this information, the state can now check the minion to see if any paths are missing, and know whether or not the archive needs to be extracted. This makes the `if_missing` argument unnecessary in most use cases.

Important: ZIP Archive Handling

Note: this information applies to 2016.11.0 and later.

Salt has two different functions for extracting ZIP archives:

1. `archive.unzip`, which uses Python's zipfile module to extract ZIP files.

2. *archive.cmd_unzip*, which uses the `unzip` CLI command to extract ZIP files.

Salt will prefer the use of *archive.cmd_unzip* when CLI options are specified (via the `options` argument), and will otherwise prefer the *archive.unzip* function. Use of *archive.cmd_unzip* can be forced however by setting the `use_cmd_unzip` argument to `True`. By contrast, setting this argument to `False` will force usage of *archive.unzip*. For example:

```
/var/www:
  archive.extracted:
    - source: salt://foo/bar/myapp.zip
    - use_cmd_unzip: True
```

When `use_cmd_unzip` is omitted, Salt will choose which extraction function to use based on the source archive and the arguments passed to the state. When in doubt, simply do not set this argument; it is provided as a means of overriding the logic Salt uses to decide which function to use.

There are differences in the features available in both extraction functions. These are detailed below.

- *Command-line options* (only supported by *archive.cmd_unzip*) - When the `options` argument is used, *archive.cmd_unzip* is the only function that can be used to extract the archive. Therefore, if `use_cmd_unzip` is specified and set to `False`, and `options` is also set, the state will not proceed.
- *Permissions* - Due to an upstream bug in Python, permissions are not preserved when the zipfile module is used to extract an archive. As of the 2016.11.0 release, *archive.unzip* (as well as this state) has an `extract_perms` argument which, when set to `True` (the default), will attempt to match the permissions of the extracted files/directories to those defined within the archive. To disable this functionality and have the state not attempt to preserve the permissions from the ZIP archive, set `extract_perms` to `False`:

```
/var/www:
  archive.extracted:
    - source: salt://foo/bar/myapp.zip
    - extract_perms: False
```

---

name Directory into which the archive should be extracted

source Archive to be extracted

Note: This argument uses the same syntax as its counterpart in the *file.managed* state.

source_hash Hash of source file, or file with list of hash-to-file mappings

Note: This argument uses the same syntax as its counterpart in the *file.managed* state.

Changed in version 2016.11.0: If this argument specifies the hash itself, instead of a URI to a file containing hashes, the hash type can now be omitted and Salt will determine the hash type based on the length of the hash. For example, both of the below states are now valid, while before only the second one would be:

```
foo_app:
  archive.extracted:
    - name: /var/www
    - source: https://mydomain.tld/foo.tar.gz
    - source_hash: 3360db35e682f1c5f9c58aa307de16d41361618c

bar_app:
  archive.extracted:
```

(continues on next page)

(continued from previous page)

```
    - name: /var/www
    - source: https://mydomain.tld/bar.tar.gz
    - source_hash: sha1=5edb7d584b82ddcbf76e311601f5d4442974aaa5
```

source_hash_name When `source_hash` refers to a hash file, Salt will try to find the correct hash by matching the filename part of the `source` URI. When managing a file with a `source` of `salt://files/foo.tar.gz`, then the following line in a hash file would match:

```
acbd18db4cc2f85cedef654fccc4a4d8    foo.tar.gz
```

This line would also match:

```
acbd18db4cc2f85cedef654fccc4a4d8    ./dir1/foo.tar.gz
```

However, sometimes a hash file will include multiple similar paths:

```
37b51d194a7513e45b56f6524f2d51f2    ./dir1/foo.txt
acbd18db4cc2f85cedef654fccc4a4d8    ./dir2/foo.txt
73feffa4b7f6bb68e44cf984c85f6e88    ./dir3/foo.txt
```

In cases like this, Salt may match the incorrect hash. This argument can be used to tell Salt which filename to match, to ensure that the correct hash is identified. For example:

```
/var/www:
  archive.extracted:
    - source: https://mydomain.tld/dir2/foo.tar.gz
    - source_hash: https://mydomain.tld/hashes
    - source_hash_name: ./dir2/foo.tar.gz
```

Note: This argument must contain the full filename entry from the checksum file, as this argument is meant to disambiguate matches for multiple files that have the same basename. So, in the example above, simply using `foo.txt` would not match.

New in version 2016.11.0.

source_hash_update [False] Set this to `True` if archive should be extracted if source_hash has changed and there is a difference between the archive and the local files. This would extract regardless of the `if_missing` parameter.

Note that this is only checked if the `source` value has not changed. If it has (e.g. to increment a version number in the path) then the archive will not be extracted even if the hash has changed.

Note: Setting this to `True` along with `keep_source` set to `False` will result the `source` re-download to do a archive file list check. If it's not desirable please consider the `skip_files_list_verify` argument.

New in version 2016.3.0.

skip_files_list_verify [False] Set this to `True` if archive should be extracted if source_hash has changed but only checksums of the archive will be checked to determine if the extraction is required.

Note: The current limitation of this logic is that you have to set minions `hash_type` config option to the same one that you're going to pass via `source_hash` argument.

New in version 3000.

skip_verify [False] If `True`, hash verification of remote file sources (`http://`, `https://`, `ftp://`) will be skipped, and the `source_hash` argument will be ignored.

New in version 2016.3.4.

keep_source [True] For `source` archives not local to the minion (i.e. from the Salt fileserver or a remote source such as `http(s)` or `ftp`), Salt will need to download the archive to the minion cache before they can be extracted. To remove the downloaded archive after extraction, set this argument to `False`.

New in version 2017.7.3.

keep [True] Same as `keep_source`, kept for backward-compatibility.

Note: If both `keep_source` and `keep` are used, `keep` will be ignored.

---

password **For ZIP archives only**. Password used for extraction.

New in version 2016.3.0.

Changed in version 2016.11.0: The newly-added *`archive.is_encrypted`* function will be used to determine if the archive is password-protected. If it is, then the `password` argument will be required for the state to proceed.

options **For tar and zip archives only**. This option can be used to specify a string of additional arguments to pass to the tar/zip command.

If this argument is not used, then the minion will attempt to use Python's native tarfile/zipfile support to extract it. For zip archives, this argument is mostly used to overwrite existing files with `o`.

Using this argument means that the `tar` or `unzip` command will be used, which is less platform-independent, so keep this in mind when using this option; the CLI options must be valid options for the `tar`/`unzip` implementation on the minion's OS.

New in version 2016.11.0.

Changed in version 2015.8.11,2016.3.2: XZ-compressed tar archives no longer require `J` to manually be set in the `options`, they are now detected automatically and decompressed using the xz CLI command and extracted using `tar xvf`. This is a more platform-independent solution, as not all tar implementations support the `J` argument for extracting archives.

Note: For tar archives, main operators like `-x`, `--extract`, `--get`, `-c` and `-f`/`--file` should *not* be used here.

---

list_options **For tar archives only**. This state uses *`archive.list`* to discover the contents of the source archive so that it knows which file paths should exist on the minion if the archive has already been extracted. For the vast majority of tar archives, *`archive.list`* "just works". Archives compressed using gzip, bzip2, and xz/lzma (with the help of the xz CLI command) are supported automatically. However, for archives compressed using other compression types, CLI options must be passed to *`archive.list`*.

This argument will be passed through to *`archive.list`* as its `options` argument, to allow it to successfully list the archive's contents. For the vast majority of archives, this argument should not need to be used, it should only be needed in cases where the state fails with an error stating that the archive's contents could not be listed.

New in version 2016.11.0.

force [False] If a path that should be occupied by a file in the extracted result is instead a directory (or vice-versa), the state will fail. Set this argument to `True` to force these paths to be removed in order to allow the archive to be extracted.

Warning: Use this option *very* carefully.

New in version 2016.11.0.

overwrite [False] Set this to `True` to force the archive to be extracted. This is useful for cases where the filenames/directories have not changed, but the content of the files have.

New in version 2016.11.1.

clean [False] Set this to `True` to remove any top-level files and recursively remove any top-level directory paths before extracting.

Note: Files will only be cleaned first if extracting the archive is deemed necessary, either by paths missing on the minion, or if `overwrite` is set to `True`.

New in version 2016.11.1.

clean_parent [False] If `True`, and the archive is extracted, delete the parent directory (i.e. the directory into which the archive is extracted), and then re-create that directory before extracting. Note that `clean` and `clean_parent` are mutually exclusive.

New in version 3000.

user The user to own each extracted file. Not available on Windows.

New in version 2015.8.0.

Changed in version 2016.3.0: When used in combination with `if_missing`, ownership will only be enforced if `if_missing` is a directory.

Changed in version 2016.11.0: Ownership will be enforced only on the file/directory paths found by running *`archive.list`* on the source archive. An alternative root directory on which to enforce ownership can be specified using the `enforce_ownership_on` argument.

group The group to own each extracted file. Not available on Windows.

New in version 2015.8.0.

Changed in version 2016.3.0: When used in combination with `if_missing`, ownership will only be enforced if `if_missing` is a directory.

Changed in version 2016.11.0: Ownership will be enforced only on the file/directory paths found by running *`archive.list`* on the source archive. An alternative root directory on which to enforce ownership can be specified using the `enforce_ownership_on` argument.

if_missing If specified, this path will be checked, and if it exists then the archive will not be extracted. This path can be either a directory or a file, so this option can also be used to check for a semaphore file and conditionally skip extraction.

Changed in version 2016.3.0: When used in combination with either `user` or `group`, ownership will only be enforced when `if_missing` is a directory.

Changed in version 2016.11.0: Ownership enforcement is no longer tied to this argument, it is simply checked for existence and extraction will be skipped if if is present.

trim_output [False] Useful for archives with many files in them. This can either be set to `True` (in which case only the first 100 files extracted will be in the state results), or it can be set to an integer for more exact control over the max number of files to include in the state results.

New in version 2016.3.0.

use_cmd_unzip [False] Set to `True` for zip files to force usage of the *`archive.cmd_unzip`* function to extract.

New in version 2016.11.0.

extract_perms [True] **For ZIP archives only.** When using *archive.unzip* to extract ZIP archives, Salt works around an upstream bug in Python to set the permissions on extracted files/directories to match those encoded into the ZIP archive. Set this argument to `False` to skip this workaround.

New in version 2016.11.0.

enforce_toplevel [True] This option will enforce a single directory at the top level of the source archive, to prevent extracting a 'tar-bomb'. Set this argument to `False` to allow archives with files (or multiple directories) at the top level to be extracted.

New in version 2016.11.0.

enforce_ownership_on When `user` or `group` is specified, Salt will default to enforcing permissions on the file/directory paths detected by running *archive.list* on the source archive. Use this argument to specify an alternate directory on which ownership should be enforced.

Note: This path must be within the path specified by the `name` argument.

New in version 2016.11.0.

archive_format One of `tar`, `zip`, or `rar`.

Changed in version 2016.11.0: If omitted, the archive format will be guessed based on the value of the `source` argument. If the minion is running a release older than 2016.11.0, this option is required.

Examples

1. tar with lmza (i.e. xz) compression:

```
graylog2-server:
  archive.extracted:
    - name: /opt/
    - source: https://github.com/downloads/Graylog2/graylog2-server/graylog2-
↪server-0.9.6p1.tar.lzma
    - source_hash: md5=499ae16dcae71eeb7c3a30c75ea7a1a6
```

2. tar archive with flag for verbose output, and enforcement of user/group ownership:

```
graylog2-server:
  archive.extracted:
    - name: /opt/
    - source: https://github.com/downloads/Graylog2/graylog2-server/graylog2-
↪server-0.9.6p1.tar.gz
    - source_hash: md5=499ae16dcae71eeb7c3a30c75ea7a1a6
    - options: v
    - user: foo
    - group: foo
```

3. tar archive, with `source_hash_update` set to `True` to prevent state from attempting extraction unless the `source_hash` differs from the previous time the archive was extracted:

```
graylog2-server:
  archive.extracted:
    - name: /opt/
    - source: https://github.com/downloads/Graylog2/graylog2-server/graylog2-
↪server-0.9.6p1.tar.lzma
    - source_hash: md5=499ae16dcae71eeb7c3a30c75ea7a1a6
    - source_hash_update: True
```

21.19.11 salt.states.artifactory

This state downloads artifacts from artifactory.

`salt.states.artifactory.`**`downloaded`**(*name, artifact, target_dir='/tmp', target_file=None, use_literal_group_id=False*)

Ensures that the artifact from artifactory exists at given location. If it doesn't exist, then it will be downloaded. If it already exists then the checksum of existing file is checked against checksum in artifactory. If it is different then the step will fail.

artifact Details of the artifact to be downloaded from artifactory. Various options are:

- artifactory_url: URL of the artifactory instance
- repository: Repository in artifactory
- artifact_id: Artifact ID
- group_id: Group ID
- packaging: Packaging
- classifier: Classifier .. versionadded:: 2015.8.0
- **version: Version** One of the following: - Version to download - `latest` - Download the latest release of this artifact - `latest_snapshot` - Download the latest snapshot for this artifact
- username: Artifactory username .. versionadded:: 2015.8.0
- password: Artifactory password .. versionadded:: 2015.8.0

target_dir Directory where the artifact should be downloaded. By default it is downloaded to /tmp directory.

target_file Target file to download artifact to. By default file name is resolved by artifactory.

An example to download an artifact to a specific file:

```
jboss_module_downloaded:
  artifactory.downloaded:
   - artifact:
       artifactory_url: http://artifactory.intranet.example.com/artifactory
       repository: 'libs-release-local'
       artifact_id: 'module'
       group_id: 'com.company.module'
       packaging: 'jar'
       classifier: 'sources'
       version: '1.0'
   - target_file: /opt/jboss7/modules/com/company/lib/module.jar
```

Download artifact to the folder (automatically resolves file name):

```
jboss_module_downloaded:
  artifactory.downloaded:
   - artifact:
        artifactory_url: http://artifactory.intranet.example.com/artifactory
        repository: 'libs-release-local'
        artifact_id: 'module'
        group_id: 'com.company.module'
        packaging: 'jar'
        classifier: 'sources'
        version: '1.0'
   - target_dir: /opt/jboss7/modules/com/company/lib
```

21.19.12 salt.states.at

Configuration disposable regularly scheduled tasks for at.

The at state can be add disposable regularly scheduled tasks for your system.

`salt.states.at.`**`absent`**(*name*, *jobid=None*, ***kwargs*)

Changed in version 2017.7.0.

Remove a job from queue

jobid: **string|int** Specific jobid to remove

tag [string] Job's tag

runas [string] Runs user-specified jobs

kwargs Addition kwargs can be provided to filter jobs. See output of *at.jobcheck* for more.

```
example1:
  at.absent:
```

Warning: this will remove all jobs!

```
example2:
  at.absent:
    - year: 13
```

```
example3:
  at.absent:
    - tag: rose
```

```
example4:
  at.absent:
    - tag: rose
    - day: 13
    - hour: 16
```

```
example5:
  at.absent:
    - jobid: 4
```

`salt.states.at.`**`mod_watch`**(*name*, ***kwargs*)

The at watcher, called to invoke the watch command.

Note: This state exists to support special handling of the `watch` *requisite*. It should not be called directly.

Parameters for this function should be set by the state being triggered.

name The name of the atjob

`salt.states.at.`**`present`**(*name*, *timespec*, *tag=None*, *user=None*, *job=None*, *unique_tag=False*)

Changed in version 2017.7.0.

Add a job to queue.

job [string] Command to run.

timespec [string] The 'timespec' follows the format documented in the at(1) manpage.

tag [string] Make a tag for the job.

user [string] The user to run the at job .. versionadded:: 2014.1.4

unique_tag [boolean] If set to True job will not be added if a job with the tag exists. .. versionadded:: 2017.7.0

```
rose:
  at.present:
```

(continues on next page)

(continued from previous page)

```
    - job: 'echo "I love saltstack" > love'
    - timespec: '9:09 11/09/13'
    - tag: love
    - user: jam
```

`salt.states.at.`**`watch`**(*name*, *timespec*, *tag=None*, *user=None*, *job=None*, *unique_tag=False*)

New in version 2017.7.0.

Add an at job if trigger by watch
job [string] Command to run.
timespec [string] The 'timespec' follows the format documented in the at(1) manpage.
tag [string] Make a tag for the job.
user [string] The user to run the at job .. versionadded:: 2014.1.4
unique_tag [boolean] If set to True job will not be added if a job with the tag exists. .. versionadded:: 2017.7.0

```
minion_restart:
  at.watch:
    - job: 'salt-call --local service.restart salt-minion'
    - timespec: 'now +1 min'
    - tag: minion_restart
    - unique_tag: trye
    - watch:
      - file: /etc/salt/minion
```

21.19.13 salt.states.augeas

Configuration management using Augeas

New in version 0.17.0.

This state requires the `augeas` Python module.

Augeas can be used to manage configuration files.

Warning: Minimal installations of Debian and Ubuntu have been seen to have packaging bugs with python-augeas, causing the augeas module to fail to import. If the minion has the augeas module installed, and the state fails with a comment saying that the state is unavailable, first restart the salt-minion service. If the problem persists past that, the following command can be run from the master to determine what is causing the import to fail:

```
salt minion-id cmd.run 'python -c "from augeas import Augeas"'
```

For affected Debian/Ubuntu hosts, installing `libpython2.7` has been known to resolve the issue.

`salt.states.augeas.`**`change`**(*name*, *context=None*, *changes=None*, *lens=None*, *load_path=None*, ***kwargs*)

New in version 2014.7.0.

This state replaces `setvalue()`.

Issue changes to Augeas, optionally for a specific context, with a specific lens.
name State name
context A file path, prefixed by `/files`. Should resolve to an actual file (not an arbitrary augeas path). This is used to avoid duplicating the file name for each item in the changes list (for example, `set bind 0.`

`0.0.0` in the example below operates on the file specified by `context`). If `context` is not specified, a file path prefixed by `/files` should be included with the `set` command.

The file path is examined to determine if the specified changes are already present.

```
redis-conf:
  augeas.change:
    - context: /files/etc/redis/redis.conf
    - changes:
      - set bind 0.0.0.0
      - set maxmemory 1G
```

changes List of changes that are issued to Augeas. Available commands are `set`, `setm`, `mv`/`move`, `ins`/`insert`, and `rm`/`remove`.

lens The lens to use, needs to be suffixed with *.lns*, e.g.: *Nginx.lns*. See the list of stock lenses shipped with Augeas.

New in version 2016.3.0.

load_path A list of directories that modules should be searched in. This is in addition to the standard load path and the directories in AUGEAS_LENS_LIB.

Usage examples:

Set the `bind` parameter in `/etc/redis/redis.conf`:

```
redis-conf:
  augeas.change:
    - changes:
      - set /files/etc/redis/redis.conf/bind 0.0.0.0
```

Note: Use the `context` parameter to specify the file you want to manipulate. This way you don't have to include this in the changes every time:

```
redis-conf:
  augeas.change:
    - context: /files/etc/redis/redis.conf
    - changes:
      - set bind 0.0.0.0
      - set databases 4
      - set maxmemory 1G
```

Augeas is aware of a lot of common configuration files and their syntax. It knows the difference between for example ini and yaml files, but also files with very specific syntax, like the hosts file. This is done with *lenses*, which provide mappings between the Augeas tree and the file.

There are many preconfigured lenses that come with Augeas by default, and they specify the common locations for configuration files. So most of the time Augeas will know how to manipulate a file. In the event that you need to manipulate a file that Augeas doesn't know about, you can specify the lens to use like this:

```
redis-conf:
  augeas.change:
    - lens: redis.lns
    - context: /files/etc/redis/redis.conf
    - changes:
      - set bind 0.0.0.0
```

Note: Even though Augeas knows that `/etc/redis/redis.conf` is a Redis configuration file and knows how to parse it, it is recommended to specify the lens anyway. This is because by default, Augeas loads all known lenses and their associated file paths. All these files are parsed when Augeas is loaded, which can take some time. When specifying a lens, Augeas is loaded with only that lens, which speeds things up quite a bit.

A more complex example, this adds an entry to the services file for Zabbix, and removes an obsolete service:

```
zabbix-service:
  augeas.change:
    - lens: services.lns
    - context: /files/etc/services
    - changes:
      - ins service-name after service-name[last()]
      - set service-name[last()] "zabbix-agent"
      - set "service-name[. = 'zabbix-agent']/port" 10050
      - set "service-name[. = 'zabbix-agent']/protocol" tcp
      - set "service-name[. = 'zabbix-agent']/#comment" "Zabbix Agent service"
      - rm "service-name[. = 'im-obsolete']"
    - unless: grep "zabbix-agent" /etc/services
```

Warning: Don't forget the `unless` here, otherwise it will fail on next runs because the service is already defined. Additionally you have to quote lines containing `service-name[. = 'zabbix-agent']` otherwise *`augeas_cfg`* execute will fail because it will receive more parameters than expected.

Note: Order is important when defining a service with Augeas, in this case it's `port`, `protocol` and `#comment`. For more info about the lens check *services lens documentation*.

---

http://augeas.net/docs/references/lenses/files/services-aug.html#Services.record

21.19.14 salt.states.aws_sqs

Manage SQS Queues

Create and destroy SQS queues. Be aware that this interacts with Amazon's services, and so may incur charges.

This module uses the awscli tool provided by Amazon. This can be downloaded from pip. Also check the documentation for awscli for configuration information.

```
myqueue:
    aws_sqs.exists:
        - region: eu-west-1
```

`salt.states.aws_sqs.`**`absent`**(*name, region, user=None, opts=False*)

Remove the named SQS queue if it exists.

name Name of the SQS queue.

region Region to remove the queue from

user Name of the user performing the SQS operations

opts Include additional arguments and options to the aws command line

`salt.states.aws_sqs.`**`exists`**(*name, region, user=None, opts=False*)

Ensure the SQS queue exists.

name Name of the SQS queue.

region Region to create the queue

user Name of the user performing the SQS operations

opts Include additional arguments and options to the aws command line

21.19.15 salt.states.azurearm_compute

Azure (ARM) Compute State Module

New in version 2019.2.0.

maintainer <devops@decisionlab.io>

maturity new

depends

- azure >= 2.0.0
- azure-common >= 1.1.8
- azure-mgmt >= 1.0.0
- azure-mgmt-compute >= 1.0.0
- azure-mgmt-network >= 1.7.1
- azure-mgmt-resource >= 1.1.0
- azure-mgmt-storage >= 1.0.0
- azure-mgmt-web >= 0.32.0
- azure-storage >= 0.34.3
- msrestazure >= 0.4.21

platform linux

configuration This module requires Azure Resource Manager credentials to be passed as a dictionary of keyword arguments to the `connection_auth` parameter in order to work properly. Since the authentication parameters are sensitive, it's recommended to pass them to the states via pillar.

Required provider parameters:

if using username and password:

- `subscription_id`
- `username`
- `password`

if using a service principal:

- `subscription_id`
- `tenant`
- `client_id`
- `secret`

Optional provider parameters:

cloud_environment: Used to point the cloud driver to different API endpoints, such as Azure GovCloud. Possible val

- `AZURE_PUBLIC_CLOUD` (default)
- `AZURE_CHINA_CLOUD`

- `AZURE_US_GOV_CLOUD`
- `AZURE_GERMAN_CLOUD`

Example Pillar for Azure Resource Manager authentication:

```
azurearm:
    user_pass_auth:
        subscription_id: 3287abc8-f98a-c678-3bde-326766fd3617
        username: fletch
        password: 123pass
    mysubscription:
        subscription_id: 3287abc8-f98a-c678-3bde-326766fd3617
        tenant: ABCDEFAB-1234-ABCD-1234-ABCDEFABCDEF
        client_id: ABCDEFAB-1234-ABCD-1234-ABCDEFABCDEF
        secret: XXXXXXXXXXXXXXXXXXXXXXXX
        cloud_environment: AZURE_PUBLIC_CLOUD
```

Example states using Azure Resource Manager authentication:

```
{% set profile = salt['pillar.get']('azurearm:mysubscription') %}
Ensure availability set exists:
    azurearm_compute.availability_set_present:
        - name: my_avail_set
        - resource_group: my_rg
        - virtual_machines:
            - my_vm1
            - my_vm2
        - tags:
            how_awesome: very
            contact_name: Elmer Fudd Gantry
        - connection_auth: {{ profile }}

Ensure availability set is absent:
    azurearm_compute.availability_set_absent:
        - name: other_avail_set
        - resource_group: my_rg
        - connection_auth: {{ profile }}
```

`salt.states.azurearm_compute.`**`availability_set_absent`**(*name*, *resource_group*, *connection_auth=None*)

New in version 2019.2.0.

Ensure an availability set does not exist in a resource group.

Parameters

- **name** – Name of the availability set.
- **resource_group** – Name of the resource group containing the availability set.
- **connection_auth** – A dict with subscription and authentication parameters to be used in connecting to the Azure Resource Manager API.

`salt.states.azurearm_compute.`**`availability_set_present`**(*name*, *resource_group*, *tags=None*, *platform_update_domain_count=None*, *platform_fault_domain_count=None*, *virtual_machines=None*, *sku=None*, *connection_auth=None*, ***kwargs*)

New in version 2019.2.0.

Ensure an availability set exists.

Parameters

- **`name`** – Name of the availability set.
- **`resource_group`** – The resource group assigned to the availability set.
- **`tags`** – A dictionary of strings can be passed as tag metadata to the availability set object.
- **`platform_update_domain_count`** – An optional parameter which indicates groups of virtual machines and underlying physical hardware that can be rebooted at the same time.
- **`platform_fault_domain_count`** – An optional parameter which defines the group of virtual machines that share a common power source and network switch.
- **`virtual_machines`** – A list of names of existing virtual machines to be included in the availability set.
- **`sku`** – The availability set SKU, which specifies whether the availability set is managed or not. Possible values are 'Aligned' or 'Classic'. An 'Aligned' availability set is managed, 'Classic' is not.
- **`connection_auth`** – A dict with subscription and authentication parameters to be used in connecting to the Azure Resource Manager API.

Example usage:

```
Ensure availability set exists:
    azurearm_compute.availability_set_present:
        - name: aset1
        - resource_group: group1
        - platform_update_domain_count: 5
        - platform_fault_domain_count: 3
        - sku: aligned
        - tags:
            contact_name: Elmer Fudd Gantry
        - connection_auth: {{ profile }}
        - require:
          - azurearm_resource: Ensure resource group exists
```

21.19.16 salt.states.azurearm_network

Azure (ARM) Network State Module

New in version 2019.2.0.

maintainer <devops@decisionlab.io>

maturity new

depends

- azure >= 2.0.0
- azure-common >= 1.1.8
- azure-mgmt >= 1.0.0
- azure-mgmt-compute >= 1.0.0
- azure-mgmt-network >= 1.7.1
- azure-mgmt-resource >= 1.1.0
- azure-mgmt-storage >= 1.0.0
- azure-mgmt-web >= 0.32.0

- azure-storage >= 0.34.3
- msrestazure >= 0.4.21

platform linux

configuration This module requires Azure Resource Manager credentials to be passed as a dictionary of keyword arguments to the `connection_auth` parameter in order to work properly. Since the authentication parameters are sensitive, it's recommended to pass them to the states via pillar.

Required provider parameters:

if using username and password:

- `subscription_id`
- `username`
- `password`

if using a service principal:

- `subscription_id`
- `tenant`
- `client_id`
- `secret`

Optional provider parameters:

cloud_environment: Used to point the cloud driver to different API endpoints, such as Azure GovCloud. Possible val

- `AZURE_PUBLIC_CLOUD` (default)
- `AZURE_CHINA_CLOUD`
- `AZURE_US_GOV_CLOUD`
- `AZURE_GERMAN_CLOUD`

Example Pillar for Azure Resource Manager authentication:

```
azurearm:
    user_pass_auth:
        subscription_id: 3287abc8-f98a-c678-3bde-326766fd3617
        username: fletch
        password: 123pass
    mysubscription:
        subscription_id: 3287abc8-f98a-c678-3bde-326766fd3617
        tenant: ABCDEFAB-1234-ABCD-1234-ABCDEFABCDEF
        client_id: ABCDEFAB-1234-ABCD-1234-ABCDEFABCDEF
        secret: XXXXXXXXXXXXXXXXXXXXXXXX
        cloud_environment: AZURE_PUBLIC_CLOUD
```

Example states using Azure Resource Manager authentication:

```
{% set profile = salt['pillar.get']('azurearm:mysubscription') %}
Ensure virtual network exists:
    azurearm_network.virtual_network_present:
        - name: my_vnet
        - resource_group: my_rg
        - address_prefixes:
```

(continues on next page)

(continued from previous page)

```
            - '10.0.0.0/8'
            - '192.168.0.0/16'
        - dns_servers:
            - '8.8.8.8'
        - tags:
            how_awesome: very
            contact_name: Elmer Fudd Gantry
        - connection_auth: {{ profile }}

Ensure virtual network is absent:
    azurearm_network.virtual_network_absent:
        - name: other_vnet
        - resource_group: my_rg
        - connection_auth: {{ profile }}
```

`salt.states.azurearm_network.`**`load_balancer_absent`**(*name*, *resource_group*, *connection_auth=None*)

New in version 2019.2.0.

Ensure a load balancer does not exist in the resource group.

Parameters

- **name** – Name of the load balancer.
- **resource_group** – The resource group assigned to the load balancer.
- **connection_auth** – A dict with subscription and authentication parameters to be used in connecting to the Azure Resource Manager API.

`salt.states.azurearm_network.`**`load_balancer_present`**(*name*, *resource_group*, *sku=None*, *frontend_ip_configurations=None*, *backend_address_pools=None*, *load_balancing_rules=None*, *probes=None*, *inbound_nat_rules=None*, *inbound_nat_pools=None*, *outbound_nat_rules=None*, *tags=None*, *connection_auth=None*, ***kwargs*)

New in version 2019.2.0.

Ensure a load balancer exists.

Parameters

- **name** – Name of the load balancer.
- **resource_group** – The resource group assigned to the load balancer.
- **sku** – The load balancer SKU, which can be 'Basic' or 'Standard'.
- **tags** – A dictionary of strings can be passed as tag metadata to the load balancer object.
- **frontend_ip_configurations** – An optional list of dictionaries representing valid FrontendIPConfiguration objects. A frontend IP configuration can be either private (using private IP address and subnet parameters) or public (using a reference to a public IP address object). Valid parameters are:
  - `name`: The name of the resource that is unique within a resource group.
  - `private_ip_address`: The private IP address of the IP configuration. Required if 'private_ip_allocation_method' is 'Static'.
  - `private_ip_allocation_method`: The Private IP allocation method. Possible values are: 'Static' and 'Dynamic'.
  - `subnet`: Name of an existing subnet inside of which the frontend IP will reside.
  - `public_ip_address`: Name of an existing public IP address which will be

assigned to the frontend IP object.

- **backend_address_pools** – An optional list of dictionaries representing valid BackendAddressPool objects. Only the 'name' parameter is valid for a BackendAddressPool dictionary. All other parameters are read-only references from other objects linking to the backend address pool. Inbound traffic is randomly load balanced across IPs in the backend IPs.
- **probes** – An optional list of dictionaries representing valid Probe objects. Valid parameters are:
  - `name`: The name of the resource that is unique within a resource group.
  - `protocol`: The protocol of the endpoint. Possible values are 'Http' or 'Tcp'. If 'Tcp' is specified, a received ACK is required for the probe to be successful. If 'Http' is specified, a 200 OK response from the specified URI is required for the probe to be successful.
  - `port`: The port for communicating the probe. Possible values range from 1 to 65535, inclusive.
  - `interval_in_seconds`: The interval, in seconds, for how frequently to probe the endpoint for health status. Typically, the interval is slightly less than half the allocated timeout period (in seconds) which allows two full probes before taking the instance out of rotation. The default value is 15, the minimum value is 5.
  - `number_of_probes`: The number of probes where if no response, will result in stopping further traffic from being delivered to the endpoint. This values allows endpoints to be taken out of rotation faster or slower than the typical times used in Azure.
  - `request_path`: The URI used for requesting health status from the VM. Path is required if a protocol is set to 'Http'. Otherwise, it is not allowed. There is no default value.
- **load_balancing_rules** – An optional list of dictionaries representing valid LoadBalancingRule objects. Valid parameters are:
  - `name`: The name of the resource that is unique within a resource group.
  - `load_distribution`: The load distribution policy for this rule. Possible values are 'Default', 'SourceIP', and 'SourceIPProtocol'.
  - `frontend_port`: The port for the external endpoint. Port numbers for each rule must be unique within the Load Balancer. Acceptable values are between 0 and 65534. Note that value 0 enables 'Any Port'.
  - `backend_port`: The port used for internal connections on the endpoint. Acceptable values are between 0 and 65535. Note that value 0 enables 'Any Port'.
  - `idle_timeout_in_minutes`: The timeout for the TCP idle connection. The value can be set between 4 and 30 minutes. The default value is 4 minutes. This element is only used when the protocol is set to TCP.
  - `enable_floating_ip`: Configures a virtual machine's endpoint for the floating IP capability required to configure a SQL AlwaysOn Availability Group. This setting is required when using the SQL AlwaysOn Availability Groups in SQL server. This setting can't be changed after you create the endpoint.
  - `disable_outbound_snat`: Configures SNAT for the VMs in the backend pool to use the public IP address specified in the frontend of the load balancing rule.
  - `frontend_ip_configuration`: Name of the frontend IP configuration object used by the load balancing rule object.
  - `backend_address_pool`: Name of the backend address pool object used by the load balancing rule object. Inbound traffic is randomly load balanced across IPs in the backend IPs.
  - `probe`: Name of the probe object used by the load balancing rule object.
- **inbound_nat_rules** – An optional list of dictionaries representing valid In-

boundNatRule objects. Defining inbound NAT rules on your load balancer is mutually exclusive with defining an inbound NAT pool. Inbound NAT pools are referenced from virtual machine scale sets. NICs that are associated with individual virtual machines cannot reference an Inbound NAT pool. They have to reference individual inbound NAT rules. Valid parameters are:
  - `name`: The name of the resource that is unique within a resource group.
  - `frontend_ip_configuration`: Name of the frontend IP configuration object used by the inbound NAT rule object.
  - `protocol`: Possible values include 'Udp', 'Tcp', or 'All'.
  - `frontend_port`: The port for the external endpoint. Port numbers for each rule must be unique within the Load Balancer. Acceptable values range from 1 to 65534.
  - `backend_port`: The port used for the internal endpoint. Acceptable values range from 1 to 65535.
  - `idle_timeout_in_minutes`: The timeout for the TCP idle connection. The value can be set between 4 and 30 minutes. The default value is 4 minutes. This element is only used when the protocol is set to TCP.
  - `enable_floating_ip`: Configures a virtual machine's endpoint for the floating IP capability required to configure a SQL AlwaysOn Availability Group. This setting is required when using the SQL AlwaysOn Availability Groups in SQL server. This setting can't be changed after you create the endpoint.
- **`inbound_nat_pools`** – An optional list of dictionaries representing valid InboundNatPool objects. They define an external port range for inbound NAT to a single backend port on NICs associated with a load balancer. Inbound NAT rules are created automatically for each NIC associated with the Load Balancer using an external port from this range. Defining an Inbound NAT pool on your Load Balancer is mutually exclusive with defining inbound NAT rules. Inbound NAT pools are referenced from virtual machine scale sets. NICs that are associated with individual virtual machines cannot reference an inbound NAT pool. They have to reference individual inbound NAT rules. Valid parameters are:
  - `name`: The name of the resource that is unique within a resource group.
  - `frontend_ip_configuration`: Name of the frontend IP configuration object used by the inbound NAT pool object.
  - `protocol`: Possible values include 'Udp', 'Tcp', or 'All'.
  - `frontend_port_range_start`: The first port number in the range of external ports that will be used to provide Inbound NAT to NICs associated with a load balancer. Acceptable values range between 1 and 65534.
  - `frontend_port_range_end`: The last port number in the range of external ports that will be used to provide Inbound NAT to NICs associated with a load balancer. Acceptable values range between 1 and 65535.
  - `backend_port`: The port used for internal connections to the endpoint. Acceptable values are between 1 and 65535.
- **`outbound_nat_rules`** – An optional list of dictionaries representing valid OutboundNatRule objects. Valid parameters are:
  - `name`: The name of the resource that is unique within a resource group.
  - `frontend_ip_configuration`: Name of the frontend IP configuration object used by the outbound NAT rule object.
  - `backend_address_pool`: Name of the backend address pool object used by the outbound NAT rule object. Outbound traffic is randomly load balanced across IPs in the backend IPs.
  - `allocated_outbound_ports`: The number of outbound ports to be used for NAT.
- **`connection_auth`** – A dict with subscription and authentication parameters to be used in connecting to the Azure Resource Manager API.

Example usage:

```
Ensure load balancer exists:
    azurearm_network.load_balancer_present:
        - name: lb1
        - resource_group: group1
        - location: eastus
        - frontend_ip_configurations:
          - name: lb1_feip1
            public_ip_address: pub_ip1
        - backend_address_pools:
          - name: lb1_bepool1
        - probes:
          - name: lb1_webprobe1
            protocol: tcp
            port: 80
            interval_in_seconds: 5
            number_of_probes: 2
        - load_balancing_rules:
          - name: lb1_webprobe1
            protocol: tcp
            frontend_port: 80
            backend_port: 80
            idle_timeout_in_minutes: 4
            frontend_ip_configuration: lb1_feip1
            backend_address_pool: lb1_bepool1
            probe: lb1_webprobe1
        - tags:
            contact_name: Elmer Fudd Gantry
        - connection_auth: {{ profile }}
        - require:
          - azurearm_resource: Ensure resource group exists
          - azurearm_network: Ensure public IP exists
```

`salt.states.azurearm_network.`**`network_interface_absent`**(*name*, *resource_group*, *connection_auth=None*)

New in version 2019.2.0.

Ensure a network interface does not exist in the resource group.

Parameters

- **name** – Name of the network interface.
- **resource_group** – The resource group assigned to the network interface.
- **connection_auth** – A dict with subscription and authentication parameters to be used in connecting to the Azure Resource Manager API.

`salt.states.azurearm_network.`**`network_interface_present`**(*name*, *ip_configurations*, *subnet*, *virtual_network*, *resource_group*, *tags=None*, *virtual_machine=None*, *network_security_group=None*, *dns_settings=None*, *mac_address=None*, *primary=None*, *enable_accelerated_networking=None*, *enable_ip_forwarding=None*, *connection_auth=None*, ***kwargs*)

New in version 2019.2.0.

Ensure a network interface exists.

Parameters

- **name** – Name of the network interface.
- **ip_configurations** – A list of dictionaries representing valid NetworkInterfaceIPConfiguration objects. The 'name' key is required at minimum. At least one IP Configuration must be present.
- **subnet** – Name of the existing subnet assigned to the network interface.
- **virtual_network** – Name of the existing virtual network containing the subnet.
- **resource_group** – The resource group assigned to the virtual network.
- **tags** – A dictionary of strings can be passed as tag metadata to the network interface object.
- **network_security_group** – The name of the existing network security group to assign to the network interface.
- **virtual_machine** – The name of the existing virtual machine to assign to the network interface.
- **dns_settings** – An optional dictionary representing a valid NetworkInterfaceDnsSettings object. Valid parameters are:
  - `dns_servers`: List of DNS server IP addresses. Use 'AzureProvidedDNS' to switch to Azure provided DNS resolution. 'AzureProvidedDNS' value cannot be combined with other IPs, it must be the only value in dns_servers collection.
  - `internal_dns_name_label`: Relative DNS name for this NIC used for internal communications between VMs in the same virtual network.
  - `internal_fqdn`: Fully qualified DNS name supporting internal communications between VMs in the same virtual network.
  - `internal_domain_name_suffix`: Even if internal_dns_name_label is not specified, a DNS entry is created for the primary NIC of the VM. This DNS name can be constructed by concatenating the VM name with the value of internal_domain_name_suffix.
- **mac_address** – Optional string containing the MAC address of the network interface.
- **primary** – Optional boolean allowing the interface to be set as the primary network interface on a virtual machine with multiple interfaces attached.
- **enable_accelerated_networking** – Optional boolean indicating whether accelerated networking should be enabled for the interface.
- **enable_ip_forwarding** – Optional boolean indicating whether IP forwarding should be enabled for the interface.
- **connection_auth** – A dict with subscription and authentication parameters to be used in connecting to the Azure Resource Manager API.

Example usage:

```
Ensure network interface exists:
    azurearm_network.network_interface_present:
        - name: iface1
        - subnet: vnet1_sn1
        - virtual_network: vnet1
        - resource_group: group1
        - ip_configurations:
          - name: iface1_ipc1
            public_ip_address: pub_ip2
        - dns_settings:
            internal_dns_name_label: decisionlab-int-test-label
        - primary: True
        - enable_accelerated_networking: True
```

(continues on next page)

(continued from previous page)

```
        - enable_ip_forwarding: False
        - network_security_group: nsg1
        - connection_auth: {{ profile }}
        - require:
          - azurearm_network: Ensure subnet exists
          - azurearm_network: Ensure network security group exists
          - azurearm_network: Ensure another public IP exists
```

`salt.states.azurearm_network.`**`network_security_group_absent`**`(`*name, resource_group, connection_auth=None*`)`

New in version 2019.2.0.

Ensure a network security group does not exist in the resource group.

Parameters

- **name** – Name of the network security group.
- **resource_group** – The resource group assigned to the network security group.
- **connection_auth** – A dict with subscription and authentication parameters to be used in connecting to the Azure Resource Manager API.

`salt.states.azurearm_network.`**`network_security_group_present`**`(`*name, resource_group, tags=None, security_rules=None, connection_auth=None, **kwargs*`)`

New in version 2019.2.0.

Ensure a network security group exists.

Parameters

- **name** – Name of the network security group.
- **resource_group** – The resource group assigned to the network security group.
- **tags** – A dictionary of strings can be passed as tag metadata to the network security group object.
- **security_rules** – An optional list of dictionaries representing valid SecurityRule objects. See the documentation for the security_rule_present state or security_rule_create_or_update execution module for more information on required and optional parameters for security rules. The rules are only managed if this parameter is present. When this parameter is absent, implemented rules will not be removed, and will merely become unmanaged.
- **connection_auth** – A dict with subscription and authentication parameters to be used in connecting to the Azure Resource Manager API.

Example usage:

```
Ensure network security group exists:
    azurearm_network.network_security_group_present:
        - name: nsg1
        - resource_group: group1
        - security_rules:
          - name: nsg1_rule1
            priority: 100
            protocol: tcp
            access: allow
            direction: outbound
```

(continues on next page)

(continued from previous page)

```
            source_address_prefix: virtualnetwork
            destination_address_prefix: internet
            source_port_range: '*'
            destination_port_range: '*'
          - name: nsg1_rule2
            priority: 101
            protocol: tcp
            access: allow
            direction: inbound
            source_address_prefix: internet
            destination_address_prefix: virtualnetwork
            source_port_range: '*'
            destination_port_ranges:
              - '80'
              - '443'
        - tags:
            contact_name: Elmer Fudd Gantry
        - connection_auth: {{ profile }}
        - require:
          - azurearm_resource: Ensure resource group exists
```

`salt.states.azurearm_network.`**`public_ip_address_absent`**(*name*, *resource_group*, *connection_auth=None*)

New in version 2019.2.0.

Ensure a public IP address does not exist in the resource group.

Parameters

- **`name`** – Name of the public IP address.
- **`resource_group`** – The resource group assigned to the public IP address.
- **`connection_auth`** – A dict with subscription and authentication parameters to be used in connecting to the Azure Resource Manager API.

`salt.states.azurearm_network.`**`public_ip_address_present`**(*name*, *resource_group*, *tags=None*, *sku=None*, *public_ip_allocation_method=None*, *public_ip_address_version=None*, *dns_settings=None*, *idle_timeout_in_minutes=None*, *connection_auth=None*, ***kwargs*)

New in version 2019.2.0.

Ensure a public IP address exists.

Parameters

- **`name`** – Name of the public IP address.
- **`resource_group`** – The resource group assigned to the public IP address.
- **`dns_settings`** – An optional dictionary representing a valid PublicIPAddressDnsSettings object. Parameters include 'domain_name_label' and 'reverse_fqdn', which accept strings. The 'domain_name_label' parameter is concatenated with the regionalized DNS zone make up the fully qualified domain name associated with the public IP address. If a domain name label is specified, an A DNS record is created for the public IP in the Microsoft Azure DNS system. The 'reverse_fqdn' parameter is a user-visible, fully qualified domain name that resolves to this public IP address. If the reverse FQDN is specified, then a PTR DNS record is created pointing from the IP address in the in-addr.arpa domain to the reverse FQDN.

- **sku** – The public IP address SKU, which can be 'Basic' or 'Standard'.
- **public_ip_allocation_method** – The public IP allocation method. Possible values are: 'Static' and 'Dynamic'.
- **public_ip_address_version** – The public IP address version. Possible values are: 'IPv4' and 'IPv6'.
- **idle_timeout_in_minutes** – An integer representing the idle timeout of the public IP address.
- **tags** – A dictionary of strings can be passed as tag metadata to the public IP address object.
- **connection_auth** – A dict with subscription and authentication parameters to be used in connecting to the Azure Resource Manager API.

Example usage:

```
Ensure public IP exists:
    azurearm_network.public_ip_address_present:
        - name: pub_ip1
        - resource_group: group1
        - dns_settings:
            domain_name_label: decisionlab-ext-test-label
        - sku: basic
        - public_ip_allocation_method: static
        - public_ip_address_version: ipv4
        - idle_timeout_in_minutes: 4
        - tags:
            contact_name: Elmer Fudd Gantry
        - connection_auth: {{ profile }}
        - require:
          - azurearm_resource: Ensure resource group exists
```

`salt.states.azurearm_network.`**`route_absent`**(*name*, *route_table*, *resource_group*, *connection_auth=None*)

New in version 2019.2.0.

Ensure a route table does not exist in the resource group.

Parameters

- **name** – Name of the route table.
- **route_table** – The name of the existing route table containing the route.
- **resource_group** – The resource group assigned to the route table.
- **connection_auth** – A dict with subscription and authentication parameters to be used in connecting to the Azure Resource Manager API.

`salt.states.azurearm_network.`**`route_present`**(*name*, *address_prefix*, *next_hop_type*, *route_table*, *resource_group*, *next_hop_ip_address=None*, *connection_auth=None*, ***kwargs*)

New in version 2019.2.0.

Ensure a route exists within a route table.

Parameters

- **name** – Name of the route.
- **address_prefix** – The destination CIDR to which the route applies.
- **next_hop_type** – The type of Azure hop the packet should be sent to. Possible values are: 'VirtualNetworkGateway', 'VnetLocal', 'Internet', 'VirtualAppliance', and 'None'.
- **next_hop_ip_address** – The IP address packets should be forwarded to. Next hop values are only allowed in routes where the next hop type is 'VirtualAppliance'.
- **route_table** – The name of the existing route table which will contain the route.

- **resource_group** – The resource group assigned to the route table.
- **connection_auth** – A dict with subscription and authentication parameters to be used in connecting to the Azure Resource Manager API.

Example usage:

```
Ensure route exists:
    azurearm_network.route_present:
        - name: rt1_route2
        - route_table: rt1
        - resource_group: group1
        - address_prefix: '192.168.0.0/16'
        - next_hop_type: vnetlocal
        - connection_auth: {{ profile }}
        - require:
          - azurearm_network: Ensure route table exists
```

`salt.states.azurearm_network.`**`route_table_absent`**(*name*, *resource_group*, *connection_auth=None*)

New in version 2019.2.0.

Ensure a route table does not exist in the resource group.

Parameters

- **name** – Name of the route table.
- **resource_group** – The resource group assigned to the route table.
- **connection_auth** – A dict with subscription and authentication parameters to be used in connecting to the Azure Resource Manager API.

`salt.states.azurearm_network.`**`route_table_present`**(*name*, *resource_group*, *tags=None*, *routes=None*, *disable_bgp_route_propagation=None*, *connection_auth=None*, ***kwargs*)

New in version 2019.2.0.

Ensure a route table exists.

Parameters

- **name** – Name of the route table.
- **resource_group** – The resource group assigned to the route table.
- **routes** – An optional list of dictionaries representing valid Route objects contained within a route table. See the documentation for the route_present state or route_create_or_update execution module for more information on required and optional parameters for routes. The routes are only managed if this parameter is present. When this parameter is absent, implemented routes will not be removed, and will merely become unmanaged.
- **disable_bgp_route_propagation** – An optional boolean parameter setting whether to disable the routes learned by BGP on the route table.
- **tags** – A dictionary of strings can be passed as tag metadata to the route table object.
- **connection_auth** – A dict with subscription and authentication parameters to be used in connecting to the Azure Resource Manager API.

Example usage:

```
Ensure route table exists:
    azurearm_network.route_table_present:
        - name: rt1
        - resource_group: group1
        - routes:
          - name: rt1_route1
            address_prefix: '0.0.0.0/0'
```

(continues on next page)

(continued from previous page)

```
            next_hop_type: internet
          - name: rt1_route2
            address_prefix: '192.168.0.0/16'
            next_hop_type: vnetlocal
        - tags:
            contact_name: Elmer Fudd Gantry
        - connection_auth: {{ profile }}
        - require:
          - azurearm_resource: Ensure resource group exists
```

salt.states.azurearm_network.**security_rule_absent**(*name*, *security_group*, *resource_group*, *connection_auth=None*)

New in version 2019.2.0.

Ensure a security rule does not exist in the network security group.

Parameters

- **name** – Name of the security rule.
- **security_group** – The network security group containing the security rule.
- **resource_group** – The resource group assigned to the network security group.
- **connection_auth** – A dict with subscription and authentication parameters to be used in connecting to the Azure Resource Manager API.

salt.states.azurearm_network.**security_rule_present**(*name*, *access*, *direction*, *priority*, *protocol*, *security_group*, *resource_group*, *destination_address_prefix=None*, *destination_port_range=None*, *source_address_prefix=None*, *source_port_range=None*, *description=None*, *destination_address_prefixes=None*, *destination_port_ranges=None*, *source_address_prefixes=None*, *source_port_ranges=None*, *connection_auth=None*, ***kwargs*)

New in version 2019.2.0.

Ensure a security rule exists.

Parameters

- **name** – Name of the security rule.
- **access** – ’allow’ or ’deny’
- **direction** – ’inbound’ or ’outbound’
- **priority** – Integer between 100 and 4096 used for ordering rule application.
- **protocol** – ’tcp’, ’udp’, or ’*’
- **security_group** – The name of the existing network security group to contain the security rule.
- **resource_group** – The resource group assigned to the network security group.
- **description** – Optional description of the security rule.
- **destination_address_prefix** – The CIDR or destination IP range. Asterix ’*’ can also be used to match all destination IPs. Default tags such as ’VirtualNetwork’, ’AzureLoadBalancer’ and ’Internet’ can also be used. If this is an ingress rule, specifies where network traffic originates from.
- **destination_port_range** – The destination port or range. Integer or range between 0 and 65535. Asterix ’*’ can also be used to match all ports.

- **source_address_prefix** – The CIDR or source IP range. Asterix '*' can also be used to match all source IPs. Default tags such as 'VirtualNetwork', 'AzureLoadBalancer' and 'Internet' can also be used. If this is an ingress rule, specifies where network traffic originates from.
- **source_port_range** – The source port or range. Integer or range between 0 and 65535. Asterix '*' can also be used to match all ports.
- **destination_address_prefixes** – A list of destination_address_prefix values. This parameter overrides destination_address_prefix and will cause any value entered there to be ignored.
- **destination_port_ranges** – A list of destination_port_range values. This parameter overrides destination_port_range and will cause any value entered there to be ignored.
- **source_address_prefixes** – A list of source_address_prefix values. This parameter overrides source_address_prefix and will cause any value entered there to be ignored.
- **source_port_ranges** – A list of source_port_range values. This parameter overrides source_port_range and will cause any value entered there to be ignored.
- **connection_auth** – A dict with subscription and authentication parameters to be used in connecting to the Azure Resource Manager API.

Example usage:

```
Ensure security rule exists:
    azurearm_network.security_rule_present:
        - name: nsg1_rule2
        - security_group: nsg1
        - resource_group: group1
        - priority: 101
        - protocol: tcp
        - access: allow
        - direction: inbound
        - source_address_prefix: internet
        - destination_address_prefix: virtualnetwork
        - source_port_range: '*'
        - destination_port_ranges:
          - '80'
          - '443'
        - connection_auth: {{ profile }}
        - require:
          - azurearm_network: Ensure network security group exists
```

`salt.states.azurearm_network.`**`subnet_absent`**(*name*, *virtual_network*, *resource_group*, *connection_auth=None*)

New in version 2019.2.0.

Ensure a virtual network does not exist in the virtual network.

Parameters

- **name** – Name of the subnet.
- **virtual_network** – Name of the existing virtual network containing the subnet.
- **resource_group** – The resource group assigned to the virtual network.
- **connection_auth** – A dict with subscription and authentication parameters to be used in connecting to the Azure Resource Manager API.

`salt.states.azurearm_network.`**`subnet_present`**(*name*, *address_prefix*, *virtual_network*, *resource_group*, *security_group=None*, *route_table=None*, *connection_auth=None*, ***kwargs*)

New in version 2019.2.0.

Ensure a subnet exists.

Parameters

- **`name`** – Name of the subnet.
- **`address_prefix`** – A CIDR block used by the subnet within the virtual network.
- **`virtual_network`** – Name of the existing virtual network to contain the subnet.
- **`resource_group`** – The resource group assigned to the virtual network.
- **`security_group`** – The name of the existing network security group to assign to the subnet.
- **`route_table`** – The name of the existing route table to assign to the subnet.
- **`connection_auth`** – A dict with subscription and authentication parameters to be used in connecting to the Azure Resource Manager API.

Example usage:

```
Ensure subnet exists:
    azurearm_network.subnet_present:
        - name: vnet1_sn1
        - virtual_network: vnet1
        - resource_group: group1
        - address_prefix: '192.168.1.0/24'
        - security_group: nsg1
        - route_table: rt1
        - connection_auth: {{ profile }}
        - require:
          - azurearm_network: Ensure virtual network exists
          - azurearm_network: Ensure network security group exists
          - azurearm_network: Ensure route table exists
```

`salt.states.azurearm_network.`**`virtual_network_absent`**(*name*, *resource_group*, *connection_auth=None*)

New in version 2019.2.0.

Ensure a virtual network does not exist in the resource group.

Parameters

- **`name`** – Name of the virtual network.
- **`resource_group`** – The resource group assigned to the virtual network.
- **`connection_auth`** – A dict with subscription and authentication parameters to be used in connecting to the Azure Resource Manager API.

`salt.states.azurearm_network.`**`virtual_network_present`**(*name*, *address_prefixes*, *resource_group*, *dns_servers=None*, *tags=None*, *connection_auth=None*, ***kwargs*)

New in version 2019.2.0.

Ensure a virtual network exists.

Parameters

- **`name`** – Name of the virtual network.
- **`resource_group`** – The resource group assigned to the virtual network.
- **`address_prefixes`** – A list of CIDR blocks which can be used by subnets within the virtual network.
- **`dns_servers`** – A list of DNS server addresses.
- **`tags`** – A dictionary of strings can be passed as tag metadata to the virtual network object.
- **`connection_auth`** – A dict with subscription and authentication parameters to be used in connecting to the Azure Resource Manager API.

Example usage:

```
Ensure virtual network exists:
    azurearm_network.virtual_network_present:
        - name: vnet1
        - resource_group: group1
        - address_prefixes:
            - '10.0.0.0/8'
            - '192.168.0.0/16'
        - dns_servers:
            - '8.8.8.8'
        - tags:
            contact_name: Elmer Fudd Gantry
        - connection_auth: {{ profile }}
        - require:
          - azurearm_resource: Ensure resource group exists
```

21.19.17 salt.states.azurearm_resource

Azure (ARM) Resource State Module

New in version 2019.2.0.

maintainer <devops@decisionlab.io>

maturity new

depends

- azure >= 2.0.0
- azure-common >= 1.1.8
- azure-mgmt >= 1.0.0
- azure-mgmt-compute >= 1.0.0
- azure-mgmt-network >= 1.7.1
- azure-mgmt-resource >= 1.1.0
- azure-mgmt-storage >= 1.0.0
- azure-mgmt-web >= 0.32.0
- azure-storage >= 0.34.3
- msrestazure >= 0.4.21

platform linux

configuration This module requires Azure Resource Manager credentials to be passed as a dictionary of keyword arguments to the `connection_auth` parameter in order to work properly. Since the authentication parameters are sensitive, it's recommended to pass them to the states via pillar.

Required provider parameters:

if using username and password:

- `subscription_id`
- `username`
- `password`

if using a service principal:

- `subscription_id`
- `tenant`
- `client_id`
- `secret`

Optional provider parameters:

cloud_environment: Used to point the cloud driver to different API endpoints, such as Azure GovCloud. Possible val

- `AZURE_PUBLIC_CLOUD` (default)
- `AZURE_CHINA_CLOUD`
- `AZURE_US_GOV_CLOUD`
- `AZURE_GERMAN_CLOUD`

Example Pillar for Azure Resource Manager authentication:

```
azurearm:
    user_pass_auth:
        subscription_id: 3287abc8-f98a-c678-3bde-326766fd3617
        username: fletch
        password: 123pass
    mysubscription:
        subscription_id: 3287abc8-f98a-c678-3bde-326766fd3617
        tenant: ABCDEFAB-1234-ABCD-1234-ABCDEFABCDEF
        client_id: ABCDEFAB-1234-ABCD-1234-ABCDEFABCDEF
        secret: XXXXXXXXXXXXXXXXXXXXXXXX
        cloud_environment: AZURE_PUBLIC_CLOUD
```

Example states using Azure Resource Manager authentication:

```
{% set profile = salt['pillar.get']('azurearm:mysubscription') %}
Ensure resource group exists:
    azurearm_resource.resource_group_present:
        - name: my_rg
        - location: westus
        - tags:
            how_awesome: very
            contact_name: Elmer Fudd Gantry
        - connection_auth: {{ profile }}

Ensure resource group is absent:
    azurearm_resource.resource_group_absent:
        - name: other_rg
        - connection_auth: {{ profile }}
```

`salt.states.azurearm_resource.`**`policy_assignment_absent`**`(`*name, scope, connection_auth=None*`)`

New in version 2019.2.0.

Ensure a policy assignment does not exist in the provided scope.

Parameters

- **`name`** – Name of the policy assignment.
- **`scope`** – The scope of the policy assignment.

connection_auth A dict with subscription and authentication parameters to be used in connecting to the Azure Resource Manager API.

`salt.states.azurearm_resource.`**`policy_assignment_present`**`(`*name, scope, definition_name, display_name=None, description=None, assignment_type=None, parameters=None, connection_auth=None, **kwargs*`)`

New in version 2019.2.0.

Ensure a security policy assignment exists.

Parameters

- **name** – Name of the policy assignment.
- **scope** – The scope of the policy assignment.
- **definition_name** – The name of the policy definition to assign.
- **display_name** – The display name of the policy assignment.
- **description** – The policy assignment description.
- **assignment_type** – The type of policy assignment.
- **parameters** – Required dictionary if a parameter is used in the policy rule.
- **connection_auth** – A dict with subscription and authentication parameters to be used in connecting to the Azure Resource Manager API.

Example usage:

```
Ensure policy assignment exists:
    azurearm_resource.policy_assignment_present:
        - name: testassign
        - scope: /subscriptions/bc75htn-a0fhsi-349b-56gh-4fghti-f84852
        - definition_name: testpolicy
        - display_name: Test Assignment
        - description: Test assignment for testing assignments.
        - connection_auth: {{ profile }}
```

`salt.states.azurearm_resource.`**`policy_definition_absent`**`(`*name, connection_auth=None*`)`

New in version 2019.2.0.

Ensure a policy definition does not exist in the current subscription.

Parameters

- **name** – Name of the policy definition.
- **connection_auth** – A dict with subscription and authentication parameters to be used in connecting to the Azure Resource Manager API.

`salt.states.azurearm_resource.`**`policy_definition_present`**`(`*name, policy_rule=None, policy_type=None, mode=None, display_name=None, description=None, metadata=None, parameters=None, policy_rule_json=None, policy_rule_file=None, template='jinja', source_hash=None, source_hash_name=None, skip_verify=False, connection_auth=None, **kwargs*`)`

New in version 2019.2.0.

Ensure a security policy definition exists.

Parameters

- **name** – Name of the policy definition.
- **policy_rule** – A YAML dictionary defining the policy rule. See Azure Policy Definition documentation for details on the structure. One of `policy_rule`, `policy_rule_json`, or `policy_rule_file` is required, in that order of precedence for use if multiple parameters are used.
- **policy_rule_json** – A text field defining the entirety of a policy definition in JSON. See Azure Policy Definition documentation for details on the structure. One of `policy_rule`, `policy_rule_json`, or `policy_rule_file` is required, in that order of precedence for use if multiple parameters are used. Note that the *name* field in the JSON will override the `name` parameter in the state.
- **policy_rule_file** – The source of a JSON file defining the entirety of a policy definition. See Azure Policy Definition documentation for details on the structure. One of `policy_rule`, `policy_rule_json`, or `policy_rule_file` is required, in that order of precedence for use if multiple parameters are used. Note that the *name* field in the JSON will override the `name` parameter in the state.
- **skip_verify** – Used for the `policy_rule_file` parameter. If `True`, hash verification of remote file sources (`http://`, `https://`, `ftp://`) will be skipped, and the `source_hash` argument will be ignored.
- **source_hash** – This can be a source hash string or the URI of a file that contains source hash strings.
- **source_hash_name** – When `source_hash` refers to a hash file, Salt will try to find the correct hash by matching the filename/URI associated with that hash.
- **policy_type** – The type of policy definition. Possible values are NotSpecified, BuiltIn, and Custom. Only used with the `policy_rule` parameter.
- **mode** – The policy definition mode. Possible values are NotSpecified, Indexed, and All. Only used with the `policy_rule` parameter.
- **display_name** – The display name of the policy definition. Only used with the `policy_rule` parameter.
- **description** – The policy definition description. Only used with the `policy_rule` parameter.
- **metadata** – The policy definition metadata defined as a dictionary. Only used with the `policy_rule` parameter.
- **parameters** – Required dictionary if a parameter is used in the policy rule. Only used with the `policy_rule` parameter.
- **connection_auth** – A dict with subscription and authentication parameters to be used in connecting to the Azure Resource Manager API.

Example usage:

```
Ensure policy definition exists:
    azurearm_resource.policy_definition_present:
        - name: testpolicy
        - display_name: Test Policy
        - description: Test policy for testing policies.
        - policy_rule:
            if:
              allOf:
                - equals: Microsoft.Compute/virtualMachines/write
                  source: action
                - field: location
                  in:
                    - eastus
```

(continues on next page)

(continued from previous page)

```
                    - eastus2
                    - centralus
            then:
              effect: deny
        - connection_auth: {{ profile }}
```

salt.states.azurearm_resource.**resource_group_absent**(*name*, *connection_auth=None*)

New in version 2019.2.0.

Ensure a resource group does not exist in the current subscription.

Parameters

- **name** – Name of the resource group.
- **connection_auth** – A dict with subscription and authentication parameters to be used in connecting to the Azure Resource Manager API.

salt.states.azurearm_resource.**resource_group_present**(*name*, *location*, *managed_by=None*, *tags=None*, *connection_auth=None*, ***kwargs*)

New in version 2019.2.0.

Ensure a resource group exists.

Parameters

- **name** – Name of the resource group.
- **location** – The Azure location in which to create the resource group. This value cannot be updated once the resource group is created.
- **managed_by** – The ID of the resource that manages this resource group. This value cannot be updated once the resource group is created.
- **tags** – A dictionary of strings can be passed as tag metadata to the resource group object.
- **connection_auth** – A dict with subscription and authentication parameters to be used in connecting to the Azure Resource Manager API.

Example usage:

```
Ensure resource group exists:
    azurearm_resource.resource_group_present:
        - name: group1
        - location: eastus
        - tags:
            contact_name: Elmer Fudd Gantry
        - connection_auth: {{ profile }}
```

21.19.18 salt.states.beacon

Management of the Salt beacons

New in version 2015.8.0.

```
ps:
  beacon.present:
    - save: True
    - enable: False
    - services:
        salt-master: running
        apache2: stopped
```

(continues on next page)

(continued from previous page)

```
sh:
  beacon.present: []

load:
  beacon.present:
    - averages:
        1m:
          - 0.0
          - 2.0
        5m:
          - 0.0
          - 1.5
        15m:
          - 0.1
          - 1.0

.. versionadded:: 3000

Beginning in the 3000 release, multiple copies of a beacon can be configured
using the ``beacon_module`` parameter.

inotify_infs:
  beacon.present:
    - save: True
    - enable: True
    - files:
       /etc/infs.conf:
         mask:
           - create
           - delete
           - modify
         recurse: True
         auto_add: True
    - interval: 10
    - beacon_module: inotify
    - disable_during_state_run: True

inotify_ntp:
  beacon.present:
    - save: True
    - enable: True
    - files:
       /etc/ntp.conf:
         mask:
           - create
           - delete
           - modify
         recurse: True
         auto_add: True
    - interval: 10
    - beacon_module: inotify
    - disable_during_state_run: True
```

`salt.states.beacon.`**`absent`**(*name*, *save=False*, ***kwargs*)

Ensure beacon is absent.

name The name of the beacon ensured absent.

save True/False, if True the beacons.conf file be updated too. Default is False.

`salt.states.beacon.`**`disabled`**`(`*name*, ***kwargs*`)`
Disable a beacon.
name The name of the beacon to disable.

`salt.states.beacon.`**`enabled`**`(`*name*, ***kwargs*`)`
Enable a beacon.
name The name of the beacon to enable.

`salt.states.beacon.`**`present`**`(`*name*, *save=False*, ***kwargs*`)`
Ensure beacon is configured with the included beacon data.
name The name of the beacon ensure is configured.
save True/False, if True the beacons.conf file be updated too. Default is False.

21.19.19 salt.states.bigip

A state module designed to enforce load-balancing configurations for F5 Big-IP entities.

maturity develop

platform f5_bigip_11.6

`salt.states.bigip.`**`add_pool_member`**`(`*hostname*, *username*, *password*, *name*, *member*`)`
A function to connect to a bigip device and add a new member to an existing pool.
hostname The host/address of the bigip device
username The iControl REST username
password The iControl REST password
name The name of the pool to modify
member The member to add to the pool

`salt.states.bigip.`**`create_monitor`**`(`*hostname*, *username*, *password*, *monitor_type*, *name*, ***kwargs*`)`
A function to connect to a bigip device and create a monitor.
hostname The host/address of the bigip device
username The iControl REST username
password The iControl REST password
monitor_type The type of monitor to create
name The name of the monitor to create
kwargs [arg=val] ...

Consult F5 BIGIP user guide for specific options for each monitor type. Typically, tmsh arg names are used.

`salt.states.bigip.`**`create_node`**`(`*hostname*, *username*, *password*, *name*, *address*`)`
Create a new node if it does not already exist.
hostname The host/address of the bigip device
username The iControl REST username
password The iControl REST password
name The name of the node to create
address The address of the node

`salt.states.bigip.`**`create_pool`**(*hostname*, *username*, *password*, *name*, *members=None*, *allow_nat=None*, *allow_snat=None*, *description=None*, *gateway_failsafe_device=None*, *ignore_persisted_weight=None*, *ip_tos_to_client=None*, *ip_tos_to_server=None*, *link_qos_to_client=None*, *link_qos_to_server=None*, *load_balancing_mode=None*, *min_active_members=None*, *min_up_members=None*, *min_up_members_action=None*, *min_up_members_checking=None*, *monitor=None*, *profiles=None*, *queue_depth_limit=None*, *queue_on_connection_limit=None*, *queue_time_limit=None*, *reselect_tries=None*, *service_down_action=None*, *slow_ramp_time=None*)

Create a new node if it does not already exist.

hostname The host/address of the bigip device

username The iControl REST username

password The iControl REST password

name The name of the pool to create

members List of members to be added to the pool

allow_nat [yes | no]

allow_snat [yes | no]

description [string]

gateway_failsafe_device [string]

ignore_persisted_weight [enabled | disabled]

ip_tos_to_client [pass-through | [integer]]

ip_tos_to_server [pass-through | [integer]]

link_qos_to_client [pass-through | [integer]]

link_qos_to_server [pass-through | [integer]]

load_balancing_mode [dynamic-ratio-member | dynamic-ratio-node | fastest-app-response | fastest-node | least-connections-members | least-connections-node | least-sessions | observed-member | observed-node | predictive-member | predictive-node | ratio-least-connections-member | ratio-least-connections-node | ratio-member | ratio-node | ratio-session | round-robin | weighted-least-connections-member | weighted-least-connections-node]

min_active_members [integer]

min_up_members [integer]

min_up_members_action [failover | reboot | restart-all]

min_up_members_checking [enabled | disabled]

monitor [name]

profiles [none | profile_name]

queue_depth_limit [integer]

queue_on_connection_limit [enabled | disabled]

queue_time_limit [integer]

reselect_tries [integer]

service_down_action [drop | none | reselect | reset]

slow_ramp_time [integer]

`salt.states.bigip.`**`create_profile`**(*hostname*, *username*, *password*, *profile_type*, *name*, ***kwargs*)

A function to connect to a bigip device and create a profile.

hostname The host/address of the bigip device

username The iControl REST username

password The iControl REST password

profile_type The type of profile to create

name The name of the profile to create

kwargs [arg=val] ...

> Consult F5 BIGIP user guide for specific options for each profile type. Typically, tmsh arg names are used.

Special Characters `|`, `,` and `:` must be escaped using `\` when used within strings.

`salt.states.bigip.`**`create_virtual`**(*hostname, username, password, name, destination, pool=None, address_status=None, auto_lasthop=None, bwc_policy=None, cmp_enabled=None, connection_limit=None, dhcp_relay=None, description=None, fallback_persistence=None, flow_eviction_policy=None, gtm_score=None, ip_forward=None, ip_protocol=None, internal=None, twelve_forward=None, last_hop_pool=None, mask=None, mirror=None, nat64=None, persist=None, profiles=None, policies=None, rate_class=None, rate_limit=None, rate_limit_mode=None, rate_limit_dst=None, rate_limit_src=None, rules=None, related_rules=None, reject=None, source=None, source_address_translation=None, source_port=None, virtual_state=None, traffic_classes=None, translate_address=None, translate_port=None, vlans=None*)

A function to connect to a bigip device and create a virtual server if it does not already exists.

hostname The host/address of the bigip device
username The iControl REST username
password The iControl REST password
name The name of the virtual to create
destination [[virtual_address_name:port] | [ipv4:port] | [ipv6.port]]
pool [[pool_name] | none]
address_status [yes | no]
auto_lasthop [default | enabled | disabled]
bwc_policy [none] | string]
cmp_enabled [yes | no]
dhcp_relay [yes | no}
connection_limit [integer]
description [string]
state [disabled | enabled]
fallback_persistence [none | [profile name]]
flow_eviction_policy [none | [eviction policy name]]
gtm_score [integer]
ip_forward [yes | no]
ip_protocol [any | protocol]
internal [yes | no]
twelve_forward(12-forward) [yes | no]
last_hop-pool [[pool_name] | none]
mask { [ipv4] | [ipv6] }
mirror { [disabled | enabled | none] }
nat64 [enabled | disabled]
persist [list]
profiles [none | default | list]
policies [none | default | list]
rate_class [name]
rate_limit [integer]
rate_limit-mode [destination | object | object-destination | object-source | object-source-destination | source | source-destination]
rate_limit-dst [integer]
rate_limit-src [integer]
rules [none | list]
related_rules [none | list]
reject [yes | no]
source { [ipv4[/prefixlen]] | [ipv6[/prefixlen]] }

source_address_translation [none | snat:pool_name | lsn | automap | dictionary]
source_port [change | preserve | preserve-strict]
state [enabled | disabled]
traffic_classes [none | default | list]
translate_address [enabled | disabled]
translate_port [enabled | disabled]
vlans [none | default | dictionary]
  vlan_ids [list]
  enabled [true | false]

`salt.states.bigip.`**`delete_monitor`**`(`*hostname*, *username*, *password*, *monitor_type*, *name*`)`

Modify an existing monitor. If it does exists, only the parameters specified will be enforced.

hostname The host/address of the bigip device
username The iControl REST username
password The iControl REST password
monitor_type The type of monitor to create
name The name of the monitor to create
kwargs [arg=val] ...

> Consult F5 BIGIP user guide for specific options for each monitor type. Typically, tmsh arg names are used.

`salt.states.bigip.`**`delete_node`**`(`*hostname*, *username*, *password*, *name*`)`

Delete an existing node.

hostname The host/address of the bigip device
username The iControl REST username
password The iControl REST password
name The name of the node which will be deleted.

`salt.states.bigip.`**`delete_pool`**`(`*hostname*, *username*, *password*, *name*`)`

Delete an existing pool.

hostname The host/address of the bigip device
username The iControl REST username
password The iControl REST password
name The name of the pool which will be deleted

`salt.states.bigip.`**`delete_pool_member`**`(`*hostname*, *username*, *password*, *name*, *member*`)`

Delete an existing pool member.

hostname The host/address of the bigip device
username The iControl REST username
password The iControl REST password
name The name of the pool to be modified
member The name of the member to delete from the pool

`salt.states.bigip.`**`delete_profile`**`(`*hostname*, *username*, *password*, *profile_type*, *name*`)`

Modify an existing profile. If it does exists, only the parameters specified will be enforced.

hostname The host/address of the bigip device
username The iControl REST username
password The iControl REST password
profile_type The type of profile to create
name The name of the profile to create
kwargs [arg=val] ...

> Consult F5 BIGIP user guide for specific options for each profile type. Typically, tmsh arg names are used.

`salt.states.bigip.`**`delete_virtual`**`(`*hostname*, *username*, *password*, *name*`)`

Delete an existing virtual.

hostname The host/address of the bigip device
username The iControl REST username
password The iControl REST password
name The name of the virtual which will be deleted

`salt.states.bigip.`**`list_monitor`**(*hostname*, *username*, *password*, *monitor_type*, *name*)

A function to list an existing monitor.

hostname The host/address of the bigip device
username The iControl REST username
password The iControl REST password
monitor_type The type of monitor to list
name The name of the monitor to list

`salt.states.bigip.`**`list_node`**(*hostname*, *username*, *password*, *name*)

A function to connect to a bigip device and list a specific node.

hostname The host/address of the bigip device
username The iControl REST username
password The iControl REST password
name The name of the node to list.

`salt.states.bigip.`**`list_pool`**(*hostname*, *username*, *password*, *name*)

A function to connect to a bigip device and list a specific pool.

hostname The host/address of the bigip device
username The iControl REST username
password The iControl REST password
name The name of the pool to list.

`salt.states.bigip.`**`list_profile`**(*hostname*, *username*, *password*, *profile_type*, *name*)

A function to list an existing profile.

hostname The host/address of the bigip device
username The iControl REST username
password The iControl REST password
profile_type The type of profile to list
name The name of the profile to list

`salt.states.bigip.`**`list_virtual`**(*hostname*, *username*, *password*, *name*)

A function to list a specific virtual.

hostname The host/address of the bigip device
username The iControl REST username
password The iControl REST password
name The name of the virtual to list

`salt.states.bigip.`**`manage_monitor`**(*hostname*, *username*, *password*, *monitor_type*, *name*, ***kwargs*)

Create a new monitor if a monitor of this type and name does not already exists. If it does exists, only the parameters specified will be enforced.

hostname The host/address of the bigip device
username The iControl REST username
password The iControl REST password
monitor_type The type of monitor to create
name The name of the monitor to create
kwargs [arg=val] ...

> Consult F5 BIGIP user guide for specific options for each monitor type. Typically, tmsh arg names are used.

`salt.states.bigip.`**`manage_node`**`(`*hostname*, *username*, *password*, *name*, *address*, *connection_limit=None*, *description=None*, *dynamic_ratio=None*, *logging=None*, *monitor=None*, *rate_limit=None*, *ratio=None*, *session=None*, *node_state=None*`)`

Manages a node of a given bigip device. If the node does not exist it will be created, otherwise, only the properties which are different than the existing will be updated.

hostname The host/address of the bigip device
username The iControl REST username
password The iControl REST password
name The name of the node to manage.
address The address of the node
connection_limit [integer]
description [string]
dynam c_ratio: [integer]
logging [enabled | disabled]
monitor [[name] | none | default]
rate_limit [integer]
ratio [integer]
session [user-enabled | user-disabled]
node_state (state) [user-down | user-up]

`salt.states.bigip.`**`manage_pool`**`(`*hostname*, *username*, *password*, *name*, *allow_nat=None*, *allow_snat=None*, *description=None*, *gateway_failsafe_device=None*, *ignore_persisted_weight=None*, *ip_tos_to_client=None*, *ip_tos_to_server=None*, *link_qos_to_client=None*, *link_qos_to_server=None*, *load_balancing_mode=None*, *min_active_members=None*, *min_up_members=None*, *min_up_members_action=None*, *min_up_members_checking=None*, *monitor=None*, *profiles=None*, *queue_depth_limit=None*, *queue_on_connection_limit=None*, *queue_time_limit=None*, *reselect_tries=None*, *service_down_action=None*, *slow_ramp_time=None*`)`

Create a new pool if it does not already exist. Pool members are managed separately. Only the parameters specified are enforced.

hostname The host/address of the bigip device
username The iControl REST username
password The iControl REST password
name The name of the pool to create
allow_nat [yes | no]
allow_snat [yes | no]
description [string]
gateway_failsafe_device [string]
ignore_persisted_weight [enabled | disabled]
ip_tos_to_client [pass-through | [integer]]
ip_tos_to_server [pass-through | [integer]]
link_qos_to_client [pass-through | [integer]]
link_qos_to_server [pass-through | [integer]]
load_balancing_mode [dynamic-ratio-member | dynamic-ratio-node | fastest-app-response | fastest-node | least-connections-members | least-connections-node | least-sessions | observed-member | observed-node | predictive-member | predictive-node | ratio-least-connections-member | ratio-least-connections-node | ratio-member | ratio-node | ratio-session | round-robin | weighted-least-connections-member | weighted-least-connections-node]
min_active_members [integer]
min_up_members [integer]
min_up_members_action [failover | reboot | restart-all]

min_up_members_checking [enabled | disabled]
monitor [name]
profiles [none | profile_name]
queue_depth_limit [integer]
queue_on_connection_limit [enabled | disabled]
queue_time_limit [integer]
reselect_tries [integer]
service_down_action [drop | none | reselect | reset]
slow_ramp_time [integer]

`salt.states.bigip.`**`manage_pool_members`**(*hostname*, *username*, *password*, *name*, *members*)

Manage the members of an existing pool. This function replaces all current pool members. Only the parameters specified are enforced.

hostname The host/address of the bigip device
username The iControl REST username
password The iControl REST password
name The name of the pool to modify
members list of pool members to manage.

`salt.states.bigip.`**`manage_profile`**(*hostname*, *username*, *password*, *profile_type*, *name*, ***kwargs*)

Create a new profile if a monitor of this type and name does not already exists. If it does exists, only the parameters specified will be enforced.

hostname The host/address of the bigip device
username The iControl REST username
password The iControl REST password
profile_type The type of profile to create
name The name of the profile to create
kwargs [arg=val] ...

> Consult F5 BIGIP user guide for specific options for each profile type. Typically, tmsh arg names are used.

`salt.states.bigip.`**`manage_virtual`**(*hostname*, *username*, *password*, *name*, *destination*, *pool=None*, *address_status=None*, *auto_lasthop=None*, *bwc_policy=None*, *cmp_enabled=None*, *connection_limit=None*, *dhcp_relay=None*, *description=None*, *fallback_persistence=None*, *flow_eviction_policy=None*, *gtm_score=None*, *ip_forward=None*, *ip_protocol=None*, *internal=None*, *twelve_forward=None*, *last_hop_pool=None*, *mask=None*, *mirror=None*, *nat64=None*, *persist=None*, *profiles=None*, *policies=None*, *rate_class=None*, *rate_limit=None*, *rate_limit_mode=None*, *rate_limit_dst=None*, *rate_limit_src=None*, *rules=None*, *related_rules=None*, *reject=None*, *source=None*, *source_address_translation=None*, *source_port=None*, *virtual_state=None*, *traffic_classes=None*, *translate_address=None*, *translate_port=None*, *vlans=None*)

Manage a virtual server. If a virtual does not exists it will be created, otherwise only the parameters specified will be enforced.

hostname The host/address of the bigip device
username The iControl REST username
password The iControl REST password
name The name of the virtual to create
destination [[virtual_address_name:port] | [ipv4:port] | [ipv6.port]]
pool [[pool_name] | none]
address_status [yes | no]
auto_lasthop [default | enabled | disabled]

bwc_policy [none] | string]
cmp_enabled [yes | no]
dhcp_relay [yes | no}
connection_limit [integer]
description [string]
state [disabled | enabled]
fallback_persistence [none | [profile name]]
flow_eviction_policy [none | [eviction policy name]]
gtm_score [integer]
ip_forward [yes | no]
ip_protocol [any | protocol]
internal [yes | no]
twelve_forward(12-forward) [yes | no]
last_hop-pool [[pool_name] | none]
mask { [ipv4] | [ipv6] }
mirror { [disabled | enabled | none] }
nat64 [enabled | disabled]
persist [list]
profiles [none | default | list]
policies [none | default | list]
rate_class [name]
rate_limit [integer]
rate_limit-mode [destination | object | object-destination | object-source | object-source-destination | source | source-destination]
rate_limit-dst [integer]
rate_limit-src [integer]
rules [none | list]
related_rules [none | list]
reject [yes | no]
source { [ipv4[/prefixlen]] | [ipv6[/prefixlen]] }
source_address_translation [none | snat:pool_name | lsn | automap | dictionary]
source_port [change | preserve | preserve-strict]
state [enabled | disabled]
traffic_classes [none | default | list]
translate_address [enabled | disabled]
translate_port [enabled | disabled]
vlans [none | default | dictionary]
  vlan_ids [list]
  enabled [true | false]

`salt.states.bigip.`**`modify_monitor`**(*hostname*, *username*, *password*, *monitor_type*, *name*, ***kwargs*)

Modify an existing monitor. If it does exists, only the parameters specified will be enforced.
hostname The host/address of the bigip device
username The iControl REST username
password The iControl REST password
monitor_type The type of monitor to create
name The name of the monitor to create
kwargs [arg=val] ...

> Consult F5 BIGIP user guide for specific options for each monitor type. Typically, tmsh arg names are used.

`salt.states.bigip.`**`modify_node`**`(`*hostname, username, password, name, connection_limit=None, description=None, dynamic_ratio=None, logging=None, monitor=None, rate_limit=None, ratio=None, session=None, node_state=None*`)`

Modify an existing node. Only a node which already exists will be modified and only the parameters specified will be enforced.

hostname The host/address of the bigip device
username The iControl REST username
password The iControl REST password
name The name of the node to modify
connection_limit [integer]
description [string]
dynamic_ratio [integer]
logging [enabled | disabled]
monitor [[name] | none | default]
rate_limit [integer]
ratio [integer]
session [user-enabled | user-disabled]
node_state (state) [user-down | user-up]

`salt.states.bigip.`**`modify_pool`**`(`*hostname, username, password, name, allow_nat=None, allow_snat=None, description=None, gateway_failsafe_device=None, ignore_persisted_weight=None, ip_tos_to_client=None, ip_tos_to_server=None, link_qos_to_client=None, link_qos_to_server=None, load_balancing_mode=None, min_active_members=None, min_up_members=None, min_up_members_action=None, min_up_members_checking=None, monitor=None, profiles=None, queue_depth_limit=None, queue_on_connection_limit=None, queue_time_limit=None, reselect_tries=None, service_down_action=None, slow_ramp_time=None*`)`

Modify an existing pool. Pool members are managed separately. Only the parameters specified are enforced.

hostname The host/address of the bigip device
username The iControl REST username
password The iControl REST password
name The name of the pool to create
allow_nat [yes | no]
allow_snat [yes | no]
description [string]
gateway_failsafe_device [string]
ignore_persisted_weight [enabled | disabled]
ip_tos_to_client [pass-through | [integer]]
ip_tos_to_server [pass-through | [integer]]
link_qos_to_client [pass-through | [integer]]
link_qos_to_server [pass-through | [integer]]
load_balancing_mode [dynamic-ratio-member | dynamic-ratio-node | fastest-app-response | fastest-node | least-connections-members | least-connections-node | least-sessions | observed-member | observed-node | predictive-member | predictive-node | ratio-least-connections-member | ratio-least-connections-node | ratio-member | ratio-node | ratio-session | round-robin | weighted-least-connections-member | weighted-least-connections-node]
min_active_members [integer]
min_up_members [integer]
min_up_members_action [failover | reboot | restart-all]
min_up_members_checking [enabled | disabled]
monitor [name]

profiles [none | profile_name]
queue_depth_limit [integer]
queue_on_connection_limit [enabled | disabled]
queue_time_limit [integer]
reselect_tries [integer]
service_down_action [drop | none | reselect | reset]
slow_ramp_time [integer]

`salt.states.bigip.`**`modify_pool_member`**`(`*hostname, username, password, name, member, connection_limit=None, description=None, dynamic_ratio=None, inherit_profile=None, logging=None, monitor=None, priority_group=None, profiles=None, rate_limit=None, ratio=None, session=None, member_state=None*`)`

A function to connect to a bigip device and modify a member of an existing pool.
hostname The host/address of the bigip device
username The iControl REST username
password The iControl REST password
name The name of the pool to modify
member The member modify
connection_limit [integer]
description [string]
dynamic_ratio [integer]
inherit_profile [enabled | disabled]
logging [enabled | disabled]
monitor [name]
priority_group [integer]
profiles [none | profile_name]
rate_limit [integer]
ratio [integer]
session [user-enabled | user-disabled]
member_state (state) [user-up | user-down]

`salt.states.bigip.`**`modify_profile`**`(`*hostname, username, password, profile_type, name, **kwargs*`)`

Modify an existing profile. If it does exists, only the parameters specified will be enforced.
hostname The host/address of the bigip device
username The iControl REST username
password The iControl REST password
profile_type The type of profile to create
name The name of the profile to create
kwargs [arg=val] ...

> Consult F5 BIGIP user guide for specific options for each monitor type. Typically, tmsh arg names are used.

`salt.states.bigip.`**`modify_virtual`**`(`*hostname, username, password, name, destination, pool=None, address_status=None, auto_lasthop=None, bwc_policy=None, cmp_enabled=None, connection_limit=None, dhcp_relay=None, description=None, fallback_persistence=None, flow_eviction_policy=None, gtm_score=None, ip_forward=None, ip_protocol=None, internal=None, twelve_forward=None, last_hop_pool=None, mask=None, mirror=None, nat64=None, persist=None, profiles=None, policies=None, rate_class=None, rate_limit=None, rate_limit_mode=None, rate_limit_dst=None, rate_limit_src=None, rules=None, related_rules=None, reject=None, source=None, source_address_translation=None, source_port=None, virtual_state=None, traffic_classes=None, translate_address=None, translate_port=None, vlans=None*`)`

Modify an virtual server. modify an existing virtual. Only parameters specified will be enforced.

hostname The host/address of the bigip device
username The iControl REST username
password The iControl REST password
name The name of the virtual to create
destination [[virtual_address_name:port] | [ipv4:port] | [ipv6.port]]
pool [[pool_name] | none]
address_status [yes | no]
auto_lasthop [default | enabled | disabled]
bwc_policy [none] | string]
cmp_enabled [yes | no]
dhcp_relay [yes | no}
connection_limit [integer]
description [string]
state [disabled | enabled]
fallback_persistence [none | [profile name]]
flow_eviction_policy [none | [eviction policy name]]
gtm_score [integer]
ip_forward [yes | no]
ip_protocol [any | protocol]
internal [yes | no]
twelve_forward(12-forward) [yes | no]
last_hop-pool [[pool_name] | none]
mask { [ipv4] | [ipv6] }
mirror { [disabled | enabled | none] }
nat64 [enabled | disabled]
persist [list]
profiles [none | default | list]
policies [none | default | list]
rate_class [name]
rate_limit [integer]
rate_limit-mode [destination | object | object-destination | object-source | object-source-destination | source | source-destination]
rate_limit_dst [integer]
rate_limit_src [integer]
rules [none | list]
related_rules [none | list]
reject [yes | no]
source { [ipv4[/prefixlen]] | [ipv6[/prefixlen]] }
source_address_translation [none | snat:pool_name | lsn | automap | dictionary]

source_port [change | preserve | preserve-strict]
state [enabled | disabled]
traffic_classes [none | default | list]
translate_address [enabled | disabled]
translate_port [enabled | disabled]
vlans [none | default | dictionary]
  vlan_ids [list]
  enabled [true | false]

21.19.20 salt.states.blockdev

Management of Block Devices

A state module to manage blockdevices

```
/dev/sda:
  blockdev.tuned:
    - read-only: True

master-data:
  blockdev.tuned:
    - name: /dev/vg/master-data
    - read-only: True
    - read-ahead: 1024
```

New in version 2014.7.0.

`salt.states.blockdev.`**`formatted`**(*name*, *fs_type='ext4'*, *force=False*, ***kwargs*)

Manage filesystems of partitions.

name The name of the block device

fs_type The filesystem it should be formatted as

force Force mke2fs to create a filesystem, even if the specified device is not a partition on a block special device. This option is only enabled for ext and xfs filesystems

This option is dangerous, use it with caution.

New in version 2016.11.0.

`salt.states.blockdev.`**`tuned`**(*name*, ***kwargs*)

Manage options of block device

name The name of the block device

opts:

- **read-ahead** Read-ahead buffer size
- **filesystem-read-ahead** Filesystem Read-ahead buffer size
- **read-only** Set Read-Only
- **read-write** Set Read-Write

21.19.21 salt.states.boto3_elasticache module

Manage Elasticache with boto3

New in version 2017.7.0.

Create, destroy and update Elasticache clusters. Be aware that this interacts with Amazon's services, and so may incur charges.

This module uses boto3 behind the scenes - as a result it inherits any limitations it boto3's implementation of the AWS API. It is also designed to as directly as possible leverage boto3's parameter naming and semantics. This allows one to use http://boto3.readthedocs.io/en/latest/reference/services/elasticache.html as an excellent source for details too involved to reiterate here.

Note: This module is designed to be transparent ("intentionally ignorant" is the phrase I used to describe it to my boss) to new AWS / boto options - since all AWS API params are passed directly through both the state and executions modules, any new args to existing functions which become available after this documentation is written should work immediately.

Brand new API calls, of course, would still require new functions to be added :)

This module accepts explicit elasticache credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More information is available here.

If IAM roles are not used you need to specify them either in a pillar file or in the minion's config file:

```
elasticache.keyid: GKTADJGHEIQSXMKKRBJ08H
elasticache.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

It's also possible to specify `key`, `keyid` and `region` via a profile, either passed in as a dict, or as a string to pull from pillars or minion config:

```
myprofile:
  keyid: GKTADJGHEIQSXMKKRBJ08H
  key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
    region: us-east-1
```

```
Ensure myelasticache exists:
  boto3_elasticache.present:
    - name: myelasticache
    - engine: redis
    - cache_node_type: cache.t1.micro
    - num_cache_nodes: 1
    - notification_topic_arn: arn:aws:sns:us-east-1:879879:my-sns-topic
    - region: us-east-1
    - keyid: GKTADJGHEIQSXMKKRBJ08H
    - key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

```
# Using a profile from pillars
Ensure myelasticache exists:
  boto3_elasticache.present:
    - name: myelasticache
    - engine: redis
    - cache_node_type: cache.t1.micro
    - num_cache_nodes: 1
    - notification_topic_arn: arn:aws:sns:us-east-1:879879:my-sns-topic
    - region: us-east-1
    - profile: myprofile
```

```
# Passing in a profile
Ensure myelasticache exists:
  boto3_elasticache.present:
    - name: myelasticache
    - engine: redis
    - cache_node_type: cache.t1.micro
```

(continues on next page)

(continued from previous page)

```
    - num_cache_nodes: 1
    - notification_topic_arn: arn:aws:sns:us-east-1:879879:my-sns-topic
    - region: us-east-1
    - profile:
        keyid: GKTADJGHEIQSXMKKRBJ08H
        key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

`salt.states.boto3_elasticache.`**`cache_cluster_absent`**`(`*name, wait=600, region=None, key=None, keyid=None, profile=None, **args*`)`

Ensure a given cache cluster is deleted.

name Name of the cache cluster.

wait Integer describing how long, in seconds, to wait for confirmation from AWS that the resource is in the desired state. Zero meaning to return success or failure immediately of course. Note that waiting for the cluster to become available is generally the better course, as failure to do so will often lead to subsequent failures when managing dependent resources.

CacheClusterId The node group (shard) identifier. Note: In general this parameter is not needed, as 'name' is used if it's not provided.

FinalSnapshotIdentifier The user-supplied name of a final cache cluster snapshot. This is the unique name that identifies the snapshot. ElastiCache creates the snapshot, and then deletes the cache cluster immediately afterward.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto3_elasticache.`**`cache_cluster_present`**`(`*name, wait=900, security_groups=None, region=None, key=None, keyid=None, profile=None, **args*`)`

Ensure a given cache cluster exists.

name Name of the cache cluster (cache cluster id).

wait Integer describing how long, in seconds, to wait for confirmation from AWS that the resource is in the desired state. Zero meaning to return success or failure immediately of course. Note that waiting for the cluster to become available is generally the better course, as failure to do so will often lead to subsequent failures when managing dependent resources.

security_groups One or more VPC security groups (names and/or IDs) associated with the cache cluster.

Note: This is additive with any sec groups provided via the SecurityGroupIds parameter below. Use this parameter ONLY when you are creating a cluster in a VPC.

CacheClusterId The node group (shard) identifier. This parameter is stored as a lowercase string.

Constraints:

- A name must contain from 1 to 20 alphanumeric characters or hyphens.
- The first character must be a letter.
- A name cannot end with a hyphen or contain two consecutive hyphens.

Note: In general this parameter is not needed, as 'name' is used if it's not provided.

ReplicationGroupId The ID of the replication group to which this cache cluster should belong. If this parameter is specified, the cache cluster is added to the specified replication group as a read replica; otherwise, the cache cluster is a standalone primary that is not part of any replication group. If the specified replication group is Multi-AZ enabled and the Availability Zone is not specified, the cache cluster is created in Availability Zones that provide the best spread of read replicas across Availability Zones.

AZMode Specifies whether the nodes in this Memcached cluster are created in a single Availability Zone or created across multiple Availability Zones in the cluster's region. If the AZMode and PreferredAvailabilityZones are not specified, ElastiCache assumes single-az mode.

Note: This parameter is ONLY supported for Memcached cache clusters.

PreferredAvailabilityZone The EC2 Availability Zone in which the cache cluster is created. All nodes belonging to this Memcached cache cluster are placed in the preferred Availability Zone. If you want to create your nodes across multiple Availability Zones, use PreferredAvailabilityZones.

Default: System chosen Availability Zone.

PreferredAvailabilityZones A list of the Availability Zones in which cache nodes are created. The order of the zones in the list is not important. The number of Availability Zones listed must equal the value of NumCacheNodes. If you want all the nodes in the same Availability Zone, use PreferredAvailabilityZone instead, or repeat the Availability Zone multiple times in the list.

Default: System chosen Availability Zones.

Note: This option is ONLY supported on Memcached.

If you are creating your cache cluster in an Amazon VPC (recommended) you can only locate nodes in Availability Zones that are associated with the subnets in the selected subnet group.

NumCacheNodes The initial (integer) number of cache nodes that the cache cluster has.

Note: For clusters running Redis, this value must be 1.

For clusters running Memcached, this value must be between 1 and 20.

CacheNodeType The compute and memory capacity of the nodes in the node group (shard). Valid node types (and pricing for them) are exhaustively described at https://aws.amazon.com/elasticache/pricing/

Note:

All T2 instances must be created in a VPC

Redis backup/restore is not supported for Redis (cluster mode disabled) T1 and T2 instances. Backup/restore is supported on Redis (cluster mode enabled) T2 instances.

Redis Append-only files (AOF) functionality is not supported for T1 or T2 instances.

Engine The name of the cache engine to be used for this cache cluster. Valid values for this parameter are: memcached | redis

EngineVersion The version number of the cache engine to be used for this cache cluster. To view the supported cache engine versions, use the DescribeCacheEngineVersions operation.

Note: You can upgrade to a newer engine version but you cannot downgrade to an earlier engine version. If you want to use an earlier engine version, you must delete the existing cache cluster or replication group and create it anew with the earlier engine version.

CacheParameterGroupName The name of the parameter group to associate with this cache cluster. If this argument is omitted, the default parameter group for the specified engine is used. You cannot use any parameter group which has cluster-enabled='yes' when creating a cluster.

CacheSubnetGroupName The name of the Cache Subnet Group to be used for the cache cluster. Use this parameter ONLY when you are creating a cache cluster within a VPC.

Note: If you're going to launch your cluster in an Amazon VPC, you need to create a subnet group before you start creating a cluster.

CacheSecurityGroupNames A list of Cache Security Group names to associate with this cache cluster. Use this parameter ONLY when you are creating a cache cluster outside of a VPC.

SecurityGroupIds One or more VPC security groups associated with the cache cluster. Use this parameter ONLY when you are creating a cache cluster within a VPC.

Tags A list of tags to be added to this resource. Note that due to shortcomings in the AWS API for Elasticache, these can only be set during resource creation - later modification is not (currently) supported.

SnapshotArns A single-element string list containing an Amazon Resource Name (ARN) that uniquely identifies a Redis RDB snapshot file stored in Amazon S3. The snapshot file is used to populate the node group (shard). The Amazon S3 object name in the ARN cannot contain any commas.

Note: This parameter is ONLY valid if the Engine parameter is redis.

SnapshotName The name of a Redis snapshot from which to restore data into the new node group (shard). The snapshot status changes to restoring while the new node group (shard) is being created.

Note: This parameter is ONLY valid if the Engine parameter is redis.

PreferredMaintenanceWindow Specifies the weekly time range during which maintenance on the cache cluster is permitted. It is specified as a range in the format ddd:hh24:mi-ddd:hh24:mi (24H Clock UTC). The minimum maintenance window is a 60 minute period. Valid values for ddd are: sun, mon, tue, wed, thu, fri, sat

Example: sun:23:00-mon:01:30

Port The port number on which each of the cache nodes accepts connections.

Default: 6379

NotificationTopicArn The Amazon Resource Name (ARN) of the Amazon Simple Notification Service (SNS) topic to which notifications are sent.

Note: The Amazon SNS topic owner must be the same as the cache cluster owner.

AutoMinorVersionUpgrade This (boolean) parameter is currently disabled.

SnapshotRetentionLimit The number of days for which ElastiCache retains automatic snapshots before deleting them.

Default: 0 (i.e., automatic backups are disabled for this cache cluster).

Note: This parameter is ONLY valid if the Engine parameter is redis.

SnapshotWindow The daily time range (in UTC) during which ElastiCache begins taking a daily snapshot of your node group (shard). If you do not specify this parameter, ElastiCache automatically chooses an appropriate time range.

Example: 05:00-09:00

Note: This parameter is ONLY valid if the Engine parameter is redis.

AuthToken The password used to access a password protected server.

Password constraints:

- Must be only printable ASCII characters.
- Must be at least 16 characters and no more than 128 characters in length.
- Cannot contain any of the following characters: '/', '"', or "@".

CacheNodeIdsToRemove A list of cache node IDs to be removed. A node ID is a numeric identifier (0001, 0002, etc.). This parameter is only valid when NumCacheNodes is less than the existing number of cache nodes. The number of cache node IDs supplied in this parameter must match the difference between the existing number of cache nodes in the cluster or pending cache nodes, whichever is greater, and the value of NumCacheNodes in the request.

NewAvailabilityZones The list of Availability Zones where the new Memcached cache nodes are created. This parameter is only valid when NumCacheNodes in the request is greater than the sum of the number of active cache nodes and the number of cache nodes pending creation (which may be zero). The number of Availability Zones supplied in this list must match the cache nodes being added in this request. Note: This option is only supported on Memcached clusters.

NotificationTopicStatus The status of the SNS notification topic. Notifications are sent only if the status is active.

Valid values: active | inactive

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto3_elasticache.`**`cache_subnet_group_absent`**`(`*name*, *region=None*, *key=None*, *keyid=None*, *profile=None*, ***args*`)`

Ensure a given cache subnet group is deleted.

name Name of the cache subnet group.

CacheSubnetGroupName A name for the cache subnet group. Note: In general this parameter is not needed, as 'name' is used if it's not provided.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto3_elasticache.`**`cache_subnet_group_present`**`(`*name*, *subnets=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*, ***args*`)`

Ensure cache subnet group exists.

name A name for the cache subnet group. This value is stored as a lowercase string. Constraints: Must contain no more than 255 alphanumeric characters or hyphens.

subnets A list of VPC subnets (IDs, Names, or a mix) for the cache subnet group.

CacheSubnetGroupName A name for the cache subnet group. This value is stored as a lowercase string. Constraints: Must contain no more than 255 alphanumeric characters or hyphens. Note: In general this parameter is not needed, as 'name' is used if it's not provided.

CacheSubnetGroupDescription A description for the cache subnet group.

SubnetIds A list of VPC subnet IDs for the cache subnet group. This is ADDITIVE with 'subnets' above.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto3_elasticache.`**`replication_group_absent`**`(`*name*, *wait=600*, *region=None*, *key=None*, *keyid=None*, *profile=None*, ***args*`)`

Ensure a given replication group is deleted.

name Name of the replication group.

wait Integer describing how long, in seconds, to wait for confirmation from AWS that the resource is in the desired state. Zero meaning to return success or failure immediately of course. Note that waiting for the cluster to become available is generally the better course, as failure to do so will often lead to subsequent failures when managing dependent resources.

ReplicationGroupId The replication group identifier. Note: In general this parameter is not needed, as 'name' is used if it's not provided.

RetainPrimaryCluster If set to true, all of the read replicas are deleted, but the primary node is retained.

FinalSnapshotIdentifier The name of a final node group (shard) snapshot. ElastiCache creates the snapshot from the primary node in the cluster, rather than one of the replicas; this is to ensure that it captures the freshest data. After the final snapshot is taken, the replication group is immediately deleted.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto3_elasticache.`**`replication_group_present`**`(`*name*, *wait=900*, *security_groups=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*, ***args*`)`

Ensure a replication group exists and is in the given state.

name Name of replication group

wait Integer describing how long, in seconds, to wait for confirmation from AWS that the resource is in the desired state. Zero meaning to return success or failure immediately of course. Note that waiting for the cluster to become available is generally the better course, as failure to do so will often lead to subsequent failures when managing dependent resources.

security_groups One or more VPC security groups (names and/or IDs) associated with the cache cluster.

Note: This is additive with any sec groups provided via the SecurityGroupIds parameter below. Use this parameter ONLY when you are creating a cluster in a VPC.

ReplicationGroupId The replication group identifier. This parameter is stored as a lowercase string.

Constraints:

- A name must contain from 1 to 20 alphanumeric characters or hyphens.
- The first character must be a letter.
- A name cannot end with a hyphen or contain two consecutive hyphens.

Note: In general this parameter is not needed, as 'name' is used if it's not provided.

ReplicationGroupDescription A user-created description for the replication group.

PrimaryClusterId The identifier of the cache cluster that serves as the primary for this replication group. This cache cluster must already exist and have a status of available. This parameter is not required if NumCacheClusters, NumNodeGroups, or ReplicasPerNodeGroup is specified.

AutomaticFailoverEnabled Specifies whether a read-only replica is automatically promoted to read/write primary if the existing primary fails. If true, Multi-AZ is enabled for this replication group. If false, Multi-AZ is disabled for this replication group.

Default: False

Note: AutomaticFailoverEnabled must be enabled for Redis (cluster mode enabled) replication groups.

ElastiCache Multi-AZ replication groups is not supported on:

- Redis versions earlier than 2.8.6.
- Redis (cluster mode disabled): T1 and T2 node types.
- Redis (cluster mode enabled): T2 node types.

NumCacheClusters The number of clusters this replication group initially has. This parameter is not used if there is more than one node group (shard). You should use ReplicasPerNodeGroup instead. If Multi-AZ is enabled , the value of this parameter must be at least 2. The maximum permitted value for NumCacheClusters is 6 (primary plus 5 replicas).

PreferredCacheClusterAZs A list of EC2 Availability Zones in which the replication group's cache clusters are created. The order of the Availability Zones in the list is the order in which clusters are allocated. The primary cluster is created in the first AZ in the list. This parameter is not used if there is more than one node group (shard). You should use NodeGroupConfiguration instead. The number of Availability Zones listed must equal the value of NumCacheClusters.

Default: System chosen Availability Zones.

Note: If you are creating your replication group in an Amazon VPC (recommended), you can only locate cache clusters in Availability Zones associated with the subnets in the selected subnet group.

NumNodeGroups An optional parameter that specifies the number of node groups (shards) for this Redis (cluster mode enabled) replication group. For Redis (cluster mode disabled) either omit this parameter or set it to 1.

Default: 1

ReplicasPerNodeGroup An optional parameter that specifies the number of replica nodes in each node group (shard). Valid values are: 0 to 5

NodeGroupConfiguration A list of node group (shard) configuration options. Each node group (shard) configuration has the following: Slots, PrimaryAvailabilityZone, ReplicaAvailabilityZones, ReplicaCount. If you're creating a Redis (cluster mode disabled) or a Redis (cluster mode enabled) replication group, you can use this parameter to configure one node group (shard) or you can omit this parameter. For fiddly details of the expected data layout of this param, see http://boto3.readthedocs.io/en/latest/reference/services/elasticache.html?#ElastiCache.Client.create_replication_group

CacheNodeType The compute and memory capacity of the nodes in the node group (shard). See https://aws.amazon.com/elasticache/pricing/ for current sizing, prices, and constraints.

Engine The name of the cache engine to be used for the cache clusters in this replication group.

EngineVersion The version number of the cache engine to be used for the cache clusters in this replication group. To view the supported cache engine versions, use the DescribeCacheEngineVersions operation.

Note: You can upgrade to a newer engine version but you cannot downgrade to an earlier engine version. If you want to use an earlier engine version, you must delete the existing cache cluster or replication group and create it anew with the earlier engine version.

CacheParameterGroupName The name of the parameter group to associate with this replication group. If this argument is omitted, the default cache parameter group for the specified engine is used.

Note: If you are running Redis version 3.2.4 or later, only one node group (shard), and want to use a default parameter group, we recommend that you specify the parameter group by name.

To create a Redis (cluster mode disabled) replication group, use CacheParameterGroupName=default.redis3.2

To create a Redis (cluster mode enabled) replication group, use CacheParameterGroupName=default.redis3.2.cluster.on

---

CacheSubnetGroupName The name of the cache subnet group to be used for the replication group.

Note: If you're going to launch your cluster in an Amazon VPC, you need to create a s group before you start creating a cluster. For more information, see Subnets and Subnet Groups.

CacheSecurityGroupNames A list of cache security group names to associate with this replication group.

SecurityGroupIds One or more Amazon VPC security groups associated with this replication group. Use this parameter only when you are creating a replication group in an VPC.

Tags A list of tags to be added to this resource. Note that due to shortcomings in the AWS API for Elasticache, these can only be set during resource creation - later modification is not (currently) supported.

SnapshotArns A list of ARNs that uniquely identify the Redis RDB snapshot files stored in Amazon S3. These snapshot files are used to populate the replication group. The Amazon S3 object name in the ARN cannot contain any commas. The list must match the number of node groups (shards) in the replication group, which means you cannot repartition.

Note: This parameter is only valid if the Engine parameter is redis.

SnapshotName The name of a snapshot from which to restore data into the new replication group. The snapshot status changes to restoring while the new replication group is being created. Note: This parameter is only valid if the Engine parameter is redis.

PreferredMaintenanceWindow Specifies the weekly time range during which maintenance on the cluster is performed. It is specified as a range in the format ddd:hh24:mi-ddd:hh24:mi (24H Clock UTC). The minimum maintenance window is a 60 minute period. Valid values for ddd are: sun, mon, tue, wed, thu, fri, sat

Example: sun:23:00-mon:01:30

Port The port number on which each member of the replication group accepts connections.

NotificationTopicArn The ARN of an SNS topic to which notifications are sent.

Note: The SNS topic owner must be the same as the cache cluster owner.

AutoMinorVersionUpgrade This parameter is currently disabled.

SnapshotRetentionLimit The number of days for which ElastiCache will retain automatic snapshots before deleting them.

Default: 0 (that is, automatic backups are disabled for this cache cluster).

Note: This parameter is only valid if the Engine parameter is redis.

SnapshotWindow The daily time range (in UTC) during which ElastiCache begins taking a daily snapshot of your node group (shard). If you do not specify this parameter, ElastiCache automatically chooses an appropriate time range.

Example: 05:00-09:00

Note: This parameter is only valid if the Engine parameter is redis.

AuthToken The password used to access a password protected server. Password constraints:
- Must be only printable ASCII characters.
- Must be at least 16 characters and no more than 128 characters in length.
- Cannot contain any of the following characters: '/', '"', or "@".

SnapshottingClusterId The cache cluster ID that is used as the daily snapshot source for the replication group.

NotificationTopicStatus The status of the SNS notification topic. Notifications are sent only if the status is active. Valid values: active | inactive

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

21.19.22 salt.states.boto3_elasticsearch module

Manage Elasticsearch Service

New in version Natrium.

configuration This module accepts explicit AWS credentials but can also utilize IAM roles assigned to the instance trough Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More Information available at:

```
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-
↪ec2.html
```

If IAM roles are not used you need to specify them either in a pillar or in the minion's config file:

```
es.keyid: GKTADJGHEIQSXMKKRBJ08H
es.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

A region may also be specified in the configuration:

```
es.region: us-east-1
```

If a region is not specified, the default is us-east-1.

It's also possible to specify key, keyid and region via a profile, either as a passed in dict, or as a string to pull from pillars or minion config:

```
myprofile:
    keyid: GKTADJGHEIQSXMKKRBJ08H
    key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
    region: us-east-1
```

codeauthor Herbert Buurman <herbert.buurman@ogd.nl>

depends boto3

`salt.states.boto3_elasticsearch.`**`absent`**(*name, blocking=True, region=None, keyid=None, key=None, profile=None*)

Ensure the Elasticsearch Domain specified does not exist.

Parameters
- **name** (*str*) – The name of the Elasticsearch domain to be made absent.
- **blocking** (*bool*) – Whether or not the state should wait for the deletion to be completed. Default: `True`

New in version Natrium.

Example:

```
Remove Elasticsearch Domain:
  boto3_elasticsearch.absent:
  - name: my_domain
  - region: eu-west-1
```

`salt.states.boto3_elasticsearch.`**`latest`**(*name*, *minor_only=True*, *region=None*, *keyid=None*, *key=None*, *profile=None*)

Ensures the Elasticsearch domain specifies runs on the latest compatible version of elasticsearch, upgrading it if it is not.

Note that this operation is blocking until the upgrade is complete.

Parameters

- **name** (*str*) – The name of the Elasticsearch domain to upgrade.
- **minor_only** (*bool*) – Only upgrade to the latest minor version.

New in version Natrium.

Example:

The following example will ensure the elasticsearch domain `my_domain` is upgraded to the latest minor version. So if it is currently 5.1 it will be upgraded to 5.6.

```
Upgrade Elasticsearch Domain:
  boto3_elasticsearch.latest:
  - name: my_domain
  - minor_only: True
  - region: eu-west-1
```

`salt.states.boto3_elasticsearch.`**`present`**(*name*, *elasticsearch_version=None*, *elasticsearch_cluster_config=None*, *ebs_options=None*, *access_policies=None*, *snapshot_options=None*, *vpc_options=None*, *cognito_options=None*, *encryption_at_rest_options=None*, *node_to_node_encryption_options=None*, *advanced_options=None*, *log_publishing_options=None*, *blocking=True*, *tags=None*, *region=None*, *keyid=None*, *key=None*, *profile=None*)

Ensure an Elasticsearch Domain exists.

Parameters

- **name** (*str*) – The name of the Elasticsearch domain that you are creating. Domain names are unique across the domains owned by an account within an AWS region. Domain names must start with a letter or number and can contain the following characters: a-z (lowercase), 0-9, and - (hyphen).
- **elasticsearch_version** (*str*) – String of format X.Y to specify version for the Elasticsearch domain eg. ”1.5” or ”2.3”.
- **elasticsearch_cluster_config** (*dict*) – Dict specifying the configuration options for an Elasticsearch domain. Keys (case sensitive) in here are:
  - InstanceType (str): The instance type for an Elasticsearch cluster.
  - InstanceCount (int): The instance type for an Elasticsearch cluster.
  - DedicatedMasterEnabled (bool): Indicate whether a dedicated master node is enabled.
  - ZoneAwarenessEnabled (bool): Indicate whether zone awareness is enabled.

    - ZoneAwarenessConfig (dict): Specifies the zone awareness configuration for a domain when zone awareness is enabled. Keys (case sensitive) in here are:
      - AvailabilityZoneCount (int): An integer value to indicate the number of availability zones for a domain when zone awareness is enabled. This should be equal to number of subnets if VPC endpoints is enabled.
    - DedicatedMasterType (str): The instance type for a dedicated master node.
    - DedicatedMasterCount (int): Total number of dedicated master nodes, active and on standby, for the cluster.
- **ebs_options** (*dict*) – Dict specifying the options to enable or disable and specifying the type and size of EBS storage volumes. Keys (case sensitive) in here are:
  - EBSEnabled (bool): Specifies whether EBS-based storage is enabled.
  - VolumeType (str): Specifies the volume type for EBS-based storage.
  - VolumeSize (int): Integer to specify the size of an EBS volume.
  - Iops (int): Specifies the IOPD for a Provisioned IOPS EBS volume (SSD).
- **access_policies** (*str or dict*) – Dict or JSON string with the IAM access policy.
- **snapshot_options** (*dict*) – Dict specifying the snapshot options. Keys (case senstive) in here are:
  - AutomatedSnapshotStartHour (int): Specifies the time, in UTC format, when the service takes a daily automated snapshot of the specified Elasticsearch domain. Default value is 0 hours.
- **vpc_options** (*dict*) – Dict with the options to specify the subnets and security groups for the VPC endpoint. Keys (case sensitive) in here are:
  - SubnetIds (list): The list of subnets for the VPC endpoint.
  - SecurityGroupIds (list): The list of security groups for the VPC endpoint.
- **cognito_options** (*dict*) – Dict with options to specify the cognito user and identity pools for Kibana authentication. Keys (case senstive) in here are:
  - Enabled (bool): Specifies the option to enable Cognito for Kibana authentication.
  - UserPoolId (str): Specifies the Cognito user pool ID for Kibana authentication.
  - IdentityPoolId (str): Specifies the Cognito identity pool ID for Kibana authentication.
  - RoleArn (str): Specifies the role ARN that provides Elasticsearch permissions for accessing Cognito resources.
- **encryption_at_rest_options** (*dict*) – Dict specifying the encryption at rest options. This option can only be used for the creation of a new Elasticsearch domain. Keys (case sensitive) in here are:
  - Enabled (bool): Specifies the option to enable Encryption At Rest.
  - KmsKeyId (str): Specifies the KMS Key ID for Encryption At Rest options.
- **node_to_node_encryption_options** (*dict*) – Dict specifying the node to node encryption options. This option can only be used for the creation of a new Elasticsearch domain. Keys (case sensitive) in here are:
  - Enabled (bool): Specify True to enable node-to-node encryption.
- **advanced_options** (*dict*) – Dict with option to allow references to indices in an HTTP request body. Must be False when configuring access to individual sub-resources. By default, the value is True. See http://docs.aws.amazon.com/elasticsearch-service/latest/developerguide /es-createupdatedomains.html#es-createdomain-configure-advanced-options for more information.
- **log_publishing_options** (*dict*) – Dict with options for various type of logs. The keys denote the type of log file and can be one of the following:
  - INDEX_SLOW_LOGS
  - SEARCH_SLOW_LOGS
  - ES_APPLICATION_LOGS

The value assigned to each key is a dict with the following case sensitive keys:
  - CloudWatchLogsLogGroupArn (str): The ARN of the Cloudwatch log group to which the log needs to be published.
  - Enabled (bool): Specifies whether given log publishing option is enabled or not.
- **blocking** (*bool*) – Whether or not the state should wait for all operations (create/update/upgrade) to be completed. Default: `True`
- **tags** (*dict*) – Dict of tags to ensure are present on the Elasticsearch domain.

New in version Natrium.

Example:

This will create an elasticsearch domain consisting of a single t2.small instance in the eu-west-1 region (Ireland) and will wait until the instance is available before returning from the state.

```
Create new domain:
  boto3_elasticsearch.present:
  - name: my_domain
  - elasticsearch_version: '5.1'
  - elasticsearch_cluster_config:
      InstanceType: t2.small.elasticsearch
      InstanceCount: 1
      DedicatedMasterEnabled: False
      ZoneAwarenessEnabled: False
  - ebs_options:
      EBSEnabled: True
      VolumeType: gp2
      VolumeSize: 10
  - snapshot_options:
      AutomatedSnapshotStartHour: 3
  - vpc_options:
      SubnetIds:
      - subnet-12345678
      SecurityGroupIds:
      - sg-12345678
  - node_to_node_encryption_options:
      Enabled: False
  - region: eu-west-1
  - tags:
      foo: bar
      baz: qux
```

`salt.states.boto3_elasticsearch.`**`tagged`**(*name, tags=None, replace=False, region=None, keyid=None, key=None, profile=None*)

Ensures the Elasticsearch domain has the tags provided. Adds tags to the domain unless `replace` is set to `True`, in which case all existing tags will be replaced with the tags provided in `tags`. (This will remove all tags if `replace` is `True` and `tags` is empty).

Parameters

- **name** (*str*) – The Elasticsearch domain to work with.
- **tags** (*dict*) – The tags to add to/replace on the Elasticsearch domain.
- **replace** (*bool*) – Whether or not to replace (`True`) all existing tags on the Elasticsearch domain, or add (`False`) tags to the ES domain.

New in version Natrium.

`salt.states.boto3_elasticsearch.`**`upgraded`**(*name, elasticsearch_version, blocking=True, region=None, keyid=None, key=None, profile=None*)

Ensures the Elasticsearch domain specified runs on the specified version of elasticsearch. Only upgrades are possible as downgrades require a manual snapshot and an S3 bucket to store them in.

Note that this operation is blocking until the upgrade is complete.

Parameters

- **name** (*str*) – The name of the Elasticsearch domain to upgrade.
- **elasticsearch_version** (*str*) – String of format X.Y to specify version for the Elasticsearch domain eg. "1.5" or "2.3".

New in version Natrium.

Example:

```
Upgrade Elasticsearch Domain:
  boto3_elasticsearch.upgraded:
  - name: my_domain
  - elasticsearch_version: '7.2'
  - region: eu-west-1
```

21.19.23 salt.states.boto3_route53 module

Manage Route53 records with Boto 3

New in version 2017.7.0.

Create and delete Route53 records. Be aware that this interacts with Amazon's services, and so may incur charges.

This module uses `boto3`, which can be installed via package, or pip.

This module accepts explicit route53 credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More information available here.

If IAM roles are not used you need to specify them either in a pillar file or in the minion's config file:

```
route53.keyid: GKTADJGHEIQSXMKKRBJ08H
route53.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

It's also possible to specify `key`, `keyid` and `region` via a profile, either passed in as a dict, or as a string to pull from pillars or minion config:

```
myprofile:
  keyid: GKTADJGHEIQSXMKKRBJ08H
  key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
  region: us-east-1
```

```
An exciting new AWS Route 53 Hosted Zone:
  boto_route53.hosted_zone_present:
    - Name: example.com.
    - PrivateZone: true
    - VPCs:
      - VPCName: MyLittleVPC
        VPCRegion: us-east-1
      - VPCId: vpc-12345678
    - region: us-east-1
    - keyid: GKTADJGHEIQSXMKKRBJ08H
    - key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs

mycnamerecord:
  boto_route53.rr_present:
    - Name: test.example.com.
    - ResourceRecords:
```

(continues on next page)

(continued from previous page)

```
      - my-elb.us-east-1.elb.amazonaws.com.
    - DomainName: example.com.
    - TTL: 60
    - Type: CNAME
    - region: us-east-1
    - keyid: GKTADJGHEIQSXMKKRBJ08H
    - key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

`salt.states.boto3_route53.`**`hosted_zone_absent`**`(`*name, Name=None, PrivateZone=False, region=None, key=None, keyid=None, profile=None*`)`

Ensure the Route53 Hostes Zone described is absent

name The name of the state definition.

Name The name of the domain. This should be a fully-specified domain, and should terminate with a period. If not provided, the value of name will be used.

PrivateZone Set True if deleting a private hosted zone.

`salt.states.boto3_route53.`**`hosted_zone_present`**`(`*name, Name=None, PrivateZone=False, CallerReference=None, Comment=None, VPCs=None, region=None, key=None, keyid=None, profile=None*`)`

Ensure a hosted zone exists with the given attributes.

name The name of the state definition.

Name The name of the domain. This should be a fully-specified domain, and should terminate with a period. This is the name you have registered with your DNS registrar. It is also the name you will delegate from your registrar to the Amazon Route 53 delegation servers returned in response to this request. If not provided, the value of name will be used.

PrivateZone Set True if creating a private hosted zone. If true, then 'VPCs' is also required.

Comment Any comments you want to include about the hosted zone.

CallerReference A unique string that identifies the request and that allows create_hosted_zone() calls to be retried without the risk of executing the operation twice. This helps ensure idempotency across state calls, but can cause issues if a zone is deleted and then an attempt is made to recreate it with the same CallerReference. If not provided, a unique UUID will be generated at each state run, which can potentially lead to duplicate zones being created if the state is run again while the previous zone creation is still in PENDING status (which can occasionally take several minutes to clear). Maximum length of 128.

VPCs A list of dicts, each dict composed of a VPCRegion, and either a VPCId or a VPCName. Note that this param is ONLY used if PrivateZone == True

- **VPCId** When creating a private hosted zone, either the VPC ID or VPC Name to associate with is required. Exclusive with VPCName.
- **VPCName** When creating a private hosted zone, either the VPC ID or VPC Name to associate with is required. Exclusive with VPCId.
- **VPCRegion** When creating a private hosted zone, the region of the associated VPC is required. If not provided, an effort will be made to determine it from VPCId or VPCName, if possible. This will fail if a given VPCName exists in multiple regions visible to the bound account, in which case you'll need to provide an explicit value for VPCRegion.

`salt.states.boto3_route53.`**`rr_absent`**`(`*name, HostedZoneId=None, DomainName=None, PrivateZone=False, Name=None, Type=None, SetIdentifier=None, region=None, key=None, keyid=None, profile=None*`)`

Ensure the Route53 record is deleted.

name The name of the state definition. This will be used for Name if the latter is not provided.

HostedZoneId The ID of the zone to delete the record from. Exclusive with DomainName.

DomainName The domain name of the zone to delete the record from. Exclusive with HostedZoneId.

PrivateZone Set to True if the RR to be removed is in a private zone, False if public.

Name Name of the resource record.

Type The record type (A, NS, MX, TXT, etc.)

SetIdentifier Valid for Weighted, Latency, Geolocation, and Failover resource record sets only. An identifier that differentiates among multiple resource record sets that have the same combination of DNS name and type. The value of SetIdentifier must be unique for each resource record set that has the same combination of DNS name and type. Omit SetIdentifier for any other types of record sets.

region The region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile Dict, or pillar key pointing to a dict, containing AWS region/key/keyid.

`salt.states.boto3_route53.`**`rr_present`**`(`*name*, *HostedZoneId=None*, *DomainName=None*, *PrivateZone=False*, *Name=None*, *Type=None*, *SetIdentifier=None*, *Weight=None*, *Region=None*, *GeoLocation=None*, *Failover=None*, *TTL=None*, *ResourceRecords=None*, *AliasTarget=None*, *HealthCheckId=None*, *TrafficPolicyInstanceId=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*`)`

Ensure the Route53 record is present.

name The name of the state definition. This will be used for Name if the latter is not provided.

HostedZoneId The ID of a zone to create the record in. Exclusive with DomainName.

DomainName The domain name of a zone to create the record in. Exclusive with HostedZoneId.

PrivateZone Set to True if the resource record should be in a private zone, False if public.

Name Name of the Route 53 resource record being managed.

Type The record type (A, NS, MX, TXT, etc.)

SetIdentifier Valid for Weighted, Latency, Geolocation, and Failover resource record sets only. An identifier that differentiates among multiple resource record sets that have the same combination of DNS name and type. The value of SetIdentifier must be unique for each resource record set that has the same combination of DNS name and type. Omit SetIdentifier for any other types of record sets.

Weight Valid for Weighted resource record sets only. Among resource record sets that have the same combination of DNS name and type, a value that determines the proportion of DNS queries that Amazon Route 53 responds to using the current resource record set. Amazon Route 53 calculates the sum of the weights for the resource record sets that have the same combination of DNS name and type. Amazon Route 53 then responds to queries based on the ratio of a resource's weight to the total.

Note the following:

- You must specify a value for the Weight element for every weighted resource record set.
- You can only specify one ResourceRecord per weighted resource record set.
- You can't create latency, failover, or geolocation resource record sets that have the same values for the Name and Type elements as weighted resource record sets.
- You can create a maximum of 100 weighted resource record sets that have the same values for the Name and Type elements.
- For weighted (but not weighted alias) resource record sets, if you set Weight to 0 for a resource record set, Amazon Route 53 never responds to queries with the applicable value for that resource record set. However, if you set Weight to 0 for all resource record sets that have the same combination of DNS name and type, traffic is routed to all resources with equal probability. The effect of setting Weight to 0 is different when you associate health checks with weighted resource record sets. For more information, see Options for Configuring Amazon Route 53 Active-Active and Active-Passive Failover in the Amazon Route 53 Developer Guide.

Region Valid for Latency-based resource record sets only. The Amazon EC2 Region where the resource that is specified in this resource record set resides. The resource typically is an AWS resource, such as an EC2 instance or an ELB load balancer, and is referred to by an IP address or a DNS domain name, depending on the record type.

GeoLocation Geo location resource record sets only. A dict that lets you control how Route 53 responds to DNS queries based on the geographic origin of the query. For example, if you want all queries from Africa to be routed to a web server with an IP address of 192.0.2.111, create a resource record set with a

Type of A and a ContinentCode of AF.

```
ContinentCode
    The two-letter code for the continent.
    Valid values: AF | AN | AS | EU | OC | NA | SA
    Constraint: Specifying ContinentCode with either CountryCode or␣
↪SubdivisionCode
                returns an InvalidInput error.
CountryCode
    The two-letter code for the country.
SubdivisionCode
    The code for the subdivision, for example, a state in the United States␣
↪or a
    province in Canada.
```

Notes

- Creating geolocation and geolocation alias resource record sets in private hosted zones is not supported.
- If you create separate resource record sets for overlapping geographic regions (for example, one resource record set for a continent and one for a country on the same continent), priority goes to the smallest geographic region. This allows you to route most queries for a continent to one resource and to route queries for a country on that continent to a different resource.
- You can't create two geolocation resource record sets that specify the same geographic location.
- The value * in the CountryCode element matches all geographic locations that aren't specified in other geolocation resource record sets that have the same values for the Name and Type elements.
- Geolocation works by mapping IP addresses to locations. However, some IP addresses aren't mapped to geographic locations, so even if you create geolocation resource record sets that cover all seven continents, Amazon Route 53 will receive some DNS queries from locations that it can't identify. We recommend that you create a resource record set for which the value of CountryCode is *, which handles both queries that come from locations for which you haven't created geolocation resource record sets and queries from IP addresses that aren't mapped to a location. If you don't create a * resource record set, Amazon Route 53 returns a "no answer" response for queries from those locations.
- You can't create non-geolocation resource record sets that have the same values for the Name and Type elements as geolocation resource record sets.

TTL The resource record cache time to live (TTL), in seconds. Note the following:

- If you're creating an alias resource record set, omit TTL. Amazon Route 53 uses the value of TTL for the alias target.
- If you're associating this resource record set with a health check (if you're adding a HealthCheckId element), we recommend that you specify a TTL of 60 seconds or less so clients respond quickly to changes in health status.
- All of the resource record sets in a group of weighted, latency, geolocation, or failover resource record sets must have the same value for TTL.
- If a group of weighted resource record sets includes one or more weighted alias resource record sets for which the alias target is an ELB load balancer, we recommend that you specify a TTL of 60 seconds for all of the non-alias weighted resource record sets that have the same name and type. Values other than 60 seconds (the TTL for load balancers) will change the effect of the values that you specify for Weight.

ResourceRecords A list, containing one or more values for the resource record. No single value can exceed 4,000 characters. For details on how to format values for different record types, see Supported DNS Resource Record Types in the Amazon Route 53 Developer Guide.

Note: You can specify more than one value for all record types except CNAME and SOA.

It is also possible to pass "magic" strings as resource record values. This functionality can easily be extended, but for the moment supports the following:

'magic:ec2_instance_tag:some_tag_name:some_string:some_instance_attr'

This tells salt to lookup an EC2 instance with a tag 'some_tag_name' which has the value 'some_string' and substitute the 'some_instance_attr' attribute of that instance as the resource record value being evaluated.

This should work generally for any EC2 instance tags, as long as the instance attribute being fetched is available to getattr(instance, 'attribute') as seen in the code below. Anything else will most likely require this function to be extended to handle it.

The canonical use-case for this (at least at our site) is to query the Name tag (which we always populate with the host's FQDN) to lookup the public or private IPs bound to the instance, so we can then automgically create Route 53 records for them.

AliasTarget The rules governing how to define an AliasTarget for the various supported use-cases are obtuse beyond reason and attempting to paraphrase them (or even worse, cut-and-paste them in their entirety) would be silly and counterproductive. If you need this feature, then Read The Fine Materials at the Boto 3 Route 53 page and/or the AWS Route 53 docs and suss them for yourself - I sure won't claim to understand them partcularly well.

region The region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile Dict, or pillar key pointing to a dict, containing AWS region/key/keyid.

21.19.24 salt.states.boto_apigateway module

Manage Apigateway Rest APIs

New in version 2016.11.0.

depends

- boto >= 2.8.0
- boto3 >= 1.2.1
- botocore >= 1.4.49

Create and destroy rest apis depending on a swagger version 2 definition file. Be aware that this interacts with Amazon's services, and so may incur charges.

This module uses `boto3`, which can be installed via package, or pip.

This module accepts explicit vpc credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More information available here.

If IAM roles are not used you need to specify them either in a pillar file or in the minion's config file:

```
vpc.keyid: GKTADJGHEIQSXMKKRBJ08H
vpc.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

It's also possible to specify `key`, `keyid` and `region` via a profile, either passed in as a dict, or as a string to pull from pillars or minion config:

```
myprofile:
  keyid: GKTADJGHEIQSXMKKRBJ08H
  key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
  region: us-east-1
```

```
Ensure Apigateway API exists:
  boto_apigateway.present:
    - name: myfunction
    - region: us-east-1
    - keyid: GKTADJGHEIQSXMKKRBJ08H
    - key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

`salt.states.boto_apigateway.`**`absent`**(*name*, *api_name*, *stage_name*, *nuke_api=False*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Ensure the stage_name associated with the given api_name deployed by boto_apigateway's present state is removed. If the currently associated deployment to the given stage_name has no other stages associated with it, the deployment will also be removed.

name Name of the swagger file in YAML format

api_name Name of the rest api on AWS ApiGateway to ensure is absent.

stage_name Name of the stage to be removed irrespective of the swagger file content. If the current deployment associated with the stage_name has no other stages associated with it, the deployment will also be removed.

nuke_api If True, removes the API itself only if there are no other stages associated with any other deployments once the given stage_name is removed.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_apigateway.`**`present`**(*name*, *api_name*, *swagger_file*, *stage_name*, *api_key_required*, *lambda_integration_role*, *lambda_region=None*, *stage_variables=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*, *lambda_funcname_format='{stage}_{api}_{resource}_{method}'*, *authorization_type='NONE'*, *error_response_template=None*, *response_template=None*)

Ensure the spcified api_name with the corresponding swaggerfile is deployed to the given stage_name in AWS ApiGateway.

this state currently only supports ApiGateway integration with AWS Lambda, and CORS support is handled through a Mock integration.

There may be multiple deployments for the API object, each deployment is tagged with a description (i.e. unique label) in pretty printed json format consisting of the following key/values.

```
{
    "api_name": api_name,
    "swagger_file": basename_of_swagger_file
    "swagger_file_md5sum": md5sum_of_swagger_file,
    "swagger_info_object": info_object_content_in_swagger_file
}
```

Please note that the name of the lambda function to be integrated will be derived via the provided lambda_funcname_format parameters:

- the default lambda_funcname_format is a string with the following substitutable keys: "{stage}_{api}_{resource}_{method}". The user can choose to reorder the known keys.
- the stage key corresponds to the stage_name passed in.
- the api key corresponds to the api_name passed in.
- the resource corresponds to the resource path defined in the passed swagger file.
- the method corresponds to the method for a resource path defined in the passed swagger file.

For the default lambda_funcname_format, given the following input:

```
api_name = '  Test    Service'
stage_name = 'alpha'
basePath = '/api'
path = '/a/{b}/c'
method = 'POST'
```

We will end up with the following Lambda Function Name that will be looked up: 'test_service_alpha_a_b_c_post'

The canconicalization of these input parameters is done in the following order:

1. lambda_funcname_format is formatted with the input parameters as passed,
2. resulting string is stripped for leading/trailing spaces,
3. path parameter's curly braces are removed from the resource path,
4. consecutive spaces and forward slashes in the paths are replaced with '_'
5. consecutive '_' are replaced with '_'

Please note that for error response handling, the swagger file must have an error response model with the following schema. The lambda functions should throw exceptions for any non successful responses. An optional pattern field can be specified in errorMessage field to aid the response mapping from Lambda to the proper error return status codes.

```
Error:
  type: object
  properties:
    stackTrace:
      type: array
      items:
        type: array
        items:
          type: string
      description: call stack
  errorType:
    type: string
    description: error type
  errorMessage:
    type: string
    description: |
      Error message, will be matched based on pattern.
      If no pattern is specified, the default pattern used for response mapping
↪will be +*.
```

name The name of the state definition

api_name The name of the rest api that we want to ensure exists in AWS API Gateway

swagger_file Name of the location of the swagger rest api definition file in YAML format.

stage_name Name of the stage we want to be associated with the given api_name and swagger_file definition

api_key_required True or False - whether the API Key is required to call API methods

lambda_integration_role The name or ARN of the IAM role that the AWS ApiGateway assumes when it executes your lambda function to handle incoming requests

lambda_region The region where we expect to find the lambda functions. This is used to determine the region where we should look for the Lambda Function for integration purposes. The region determination is based on the following priority:

1. lambda_region as passed in (is not None)
2. if lambda_region is None, use the region as if a boto_lambda function were executed without explicitly specifying lambda region.
3. if region determined in (2) is different than the region used by boto_apigateway functions, a final lookup will be attempted using the boto_apigateway region.

stage_variables A dict with variables and their values, or a pillar key (string) that contains a dict with variables and their values. key and values in the dict must be strings. {'string': 'string'}

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

lambda_funcname_format Please review the earlier example for the usage. The only substituable keys in the funcname format are {stage}, {api}, {resource}, {method}. Any other keys or positional subsitution parameters will be flagged as an invalid input.

authorization_type This field can be either 'NONE', or 'AWS_IAM'. This will be applied to all methods in the given swagger spec file. Default is set to 'NONE'

error_response_template String value that defines the response template mapping that should be applied in cases error occurs. Refer to AWS documentation for details: http://docs.aws.amazon.com/apigateway/latest/developerguide/api-gateway-mapping-template-reference.html

If set to None, the following default value is used:

```
'#set($inputRoot = $input.path('$'))\n'
'{\n'
'  "errorMessage" : "$inputRoot.errorMessage",\n'
'  "errorType" : "$inputRoot.errorType",\n'
'  "stackTrace" : [\n'
'#foreach($stackTrace in $inputRoot.stackTrace)\n'
'    [\n'
'#foreach($elem in $stackTrace)\n'
'      "$elem"\n'
'#if($foreach.hasNext),#end\n'
'#end\n'
'    ]\n'
'#if($foreach.hasNext),#end\n'
'#end\n'
'  ]\n'
```

New in version 2017.7.0.

response_template String value that defines the response template mapping applied in case of success (including OPTIONS method) If set to None, empty ({}) template is assumed, which will transfer response from the lambda function as is.

New in version 2017.7.0.

`salt.states.boto_apigateway.`**`usage_plan_absent`**`(`*name, plan_name, region=None, key=None, keyid=None, profile=None*`)`

Ensures usage plan identified by name is no longer present

New in version 2017.7.0.

name name of the state

plan_name name of the plan to remove

```yaml
usage plan absent:
  boto_apigateway.usage_plan_absent:
    - plan_name: my_usage_plan
    - profile: my_profile
```

`salt.states.boto_apigateway.`**`usage_plan_association_absent`**`(`*name, plan_name, api_stages, region=None, key=None, keyid=None, profile=None*`)`

Ensures usage plan identified by name is removed from provided api_stages If a plan is associated to stages not listed in api_stages parameter, those associations remain intact.

New in version 2017.7.0.

name name of the state

plan_name name of the plan to use

api_stages list of dictionaries, where each dictionary consists of the following keys:

  apiId apiId of the api to detach usage plan from

  stage stage name of the api to detach usage plan from

```
UsagePlanAssociationAbsent:
  boto_apigateway.usage_plan_association_absent:
    - plan_name: my_plan
    - api_stages:
      - apiId: 9kb0404ec0
        stage: my_stage
      - apiId: l9v7o2aj90
        stage: my_stage
    - profile: my_profile
```

`salt.states.boto_apigateway.`**`usage_plan_association_present`**`(`*name, plan_name, api_stages, region=None, key=None, keyid=None, profile=None*`)`

Ensures usage plan identified by name is added to provided api_stages

New in version 2017.7.0.

name name of the state

plan_name name of the plan to use

api_stages list of dictionaries, where each dictionary consists of the following keys:

  apiId apiId of the api to attach usage plan to

  stage stage name of the api to attach usage plan to

```
UsagePlanAssociationPresent:
  boto_apigateway.usage_plan_association_present:
    - plan_name: my_plan
    - api_stages:
      - apiId: 9kb0404ec0
        stage: my_stage
      - apiId: l9v7o2aj90
        stage: my_stage
    - profile: my_profile
```

`salt.states.boto_apigateway.`**`usage_plan_present`**`(`*name, plan_name, description=None, throttle=None, quota=None, region=None, key=None, keyid=None, profile=None*`)`

Ensure the spcifieda usage plan with the corresponding metrics is deployed

New in version 2017.7.0.

name name of the state

plan_name [Required] name of the usage plan

throttle [Optional] throttling parameters expressed as a dictionary. If provided, at least one of the throttling parameters must be present

  rateLimit rate per second at which capacity bucket is populated

  burstLimit maximum rate allowed

quota [Optional] quota on the number of api calls permitted by the plan. If provided, limit and period must be present

  limit [Required] number of calls permitted per quota period

  offset [Optional] number of calls to be subtracted from the limit at the beginning of the period

  period [Required] period to which quota applies. Must be DAY, WEEK or MONTH

```
UsagePlanPresent:
  boto_apigateway.usage_plan_present:
    - plan_name: my_usage_plan
    - throttle:
        rateLimit: 70
        burstLimit: 100
    - quota:
        limit: 1000
        offset: 0
        period: DAY
    - profile: my_profile
```

21.19.25 salt.states.boto_asg

Manage Autoscale Groups

New in version 2014.7.0.

Create and destroy autoscale groups. Be aware that this interacts with Amazon's services, and so may incur charges.

This module uses boto, which can be installed via package, or pip.

This module accepts explicit autoscale credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More Information available at:

```
http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-ec2.html
```

If IAM roles are not used you need to specify them either in a pillar or in the minion's config file:

```
asg.keyid: GKTADJGHEIQSXMKKRBJ08H
asg.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

It's also possible to specify key, keyid and region via a profile, either as a passed in dict, or as a string to pull from pillars or minion config:

```
myprofile:
    keyid: GKTADJGHEIQSXMKKRBJ08H
    key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
    region: us-east-1
```

```
Ensure myasg exists:
  boto_asg.present:
    - name: myasg
    - launch_config_name: mylc
```

(continues on next page)

(continued from previous page)

```
    - availability_zones:
      - us-east-1a
      - us-east-1b
    - min_size: 1
    - max_size: 1
    - desired_capacity: 1
    - load_balancers:
      - myelb
    - suspended_processes:
        - AddToLoadBalancer
        - AlarmNotification
    - scaling_policies
        - adjustment_type: ChangeInCapacity
        - as_name: api-production-iad
        - cooldown: 1800
        - min_adjustment_step: None
        - name: ScaleDown
        - scaling_adjustment: -1
    - region: us-east-1
    - keyid: GKTADJGHEIQSXMKKRBJ08H
    - key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs

# Using a profile from pillars.
Ensure myasg exists:
  boto_asg.present:
    - name: myasg
    - launch_config_name: mylc
    - availability_zones:
      - us-east-1a
      - us-east-1b
    - min_size: 1
    - max_size: 1
    - desired_capacity: 1
    - load_balancers:
      - myelb
    - profile: myprofile

# Passing in a profile.
Ensure myasg exists:
  boto_asg.present:
    - name: myasg
    - launch_config_name: mylc
    - availability_zones:
      - us-east-1a
      - us-east-1b
    - min_size: 1
    - max_size: 1
    - desired_capacity: 1
    - load_balancers:
      - myelb
    - profile:
        keyid: GKTADJGHEIQSXMKKRBJ08H
        key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
        region: us-east-1

# Deleting an autoscale group with running instances.
```

(continues on next page)

(continued from previous page)

```
Ensure myasg is deleted:
  boto_asg.absent:
    - name: myasg
    # If instances exist, we must force the deletion of the asg.
    - force: True
```

It's possible to specify cloudwatch alarms that will be setup along with the ASG. Note the alarm name will be the name attribute defined, plus the ASG resource name.

```
Ensure myasg exists:
  boto_asg.present:
    - name: myasg
    - launch_config_name: mylc
    - availability_zones:
      - us-east-1a
      - us-east-1b
    - min_size: 1
    - max_size: 1
    - desired_capacity: 1
    - load_balancers:
      - myelb
    - profile: myprofile
    - alarms:
        CPU:
          name: 'ASG CPU **MANAGED BY SALT**'
          attributes:
            metric: CPUUtilization
            namespace: AWS/EC2
            statistic: Average
            comparison: '>='
            threshold: 65.0
            period: 60
            evaluation_periods: 30
            unit: null
            description: 'ASG CPU'
            alarm_actions: [ 'arn:aws:sns:us-east-1:12345:myalarm' ]
            insufficient_data_actions: []
            ok_actions: [ 'arn:aws:sns:us-east-1:12345:myalarm' ]
```

You can also use alarms from pillars, and override values from the pillar alarms by setting overrides on the resource. Note that 'boto_asg_alarms' will be used as a default value for all resources, if defined and can be used to ensure alarms are always set for an ASG resource.

Setting the alarms in a pillar:

```
my_asg_alarm:
  CPU:
    name: 'ASG CPU **MANAGED BY SALT**'
    attributes:
      metric: CPUUtilization
      namespace: AWS/EC2
      statistic: Average
      comparison: '>='
      threshold: 65.0
      period: 60
      evaluation_periods: 30
      unit: null
```

(continues on next page)

(continued from previous page)

```
      description: 'ASG CPU'
      alarm_actions: [ 'arn:aws:sns:us-east-1:12345:myalarm' ]
      insufficient_data_actions: []
      ok_actions: [ 'arn:aws:sns:us-east-1:12345:myalarm' ]
```

Overriding the alarm values on the resource:

```
Ensure myasg exists:
  boto_asg.present:
    - name: myasg
    - launch_config_name: mylc
    - availability_zones:
      - us-east-1a
      - us-east-1b
    - min_size: 1
    - max_size: 1
    - desired_capacity: 1
    - load_balancers:
      - myelb
    - profile: myprofile
    - alarms_from_pillar: my_asg_alarm
    # override CPU:attributes:threshold
    - alarms:
        CPU:
          attributes:
            threshold: 50.0
```

`salt.states.boto_asg.`**`absent`**(*name, force=False, region=None, key=None, keyid=None, profile=None, remove_lc=False*)

Ensure the named autoscale group is deleted.

name Name of the autoscale group.

force Force deletion of autoscale group.

remove_lc Delete the launch config as well.

region The region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_asg.`**`present`**(*name, launch_config_name, availability_zones, min_size, max_size, launch_config=None, desired_capacity=None, load_balancers=None, default_cooldown=None, health_check_type=None, health_check_period=None, placement_group=None, vpc_zone_identifier=None, subnet_names=None, tags=None, termination_policies=None, termination_policies_from_pillar='boto_asg_termination_policies', suspended_processes=None, scaling_policies=None, scaling_policies_from_pillar='boto_asg_scaling_policies', scheduled_actions=None, scheduled_actions_from_pillar='boto_asg_scheduled_actions', alarms=None, alarms_from_pillar='boto_asg_alarms', region=None, key=None, keyid=None, profile=None, notification_arn=None, notification_arn_from_pillar='boto_asg_notification_arn', notification_types=None, notification_types_from_pillar='boto_asg_notification_types'*)

Ensure the autoscale group exists.

name Name of the autoscale group.

launch_config_name Name of the launch config to use for the group. Or, if `launch_config` is specified, this will be the launch config name's prefix. (see below)

launch_config A dictionary of launch config attributes. If specified, a launch config will be used or created, matching this set of attributes, and the autoscale group will be set to use that launch config. The launch config name will be the `launch_config_name` followed by a hyphen followed by a hash of the `launch_config` dict contents. Example:

```
my_asg:
  boto_asg.present:
  - launch_config:
    - ebs_optimized: false
    - instance_profile_name: my_iam_profile
    - kernel_id: ''
    - ramdisk_id: ''
    - key_name: my_ssh_key
    - image_name: aws2015091-hvm
    - instance_type: c3.xlarge
    - instance_monitoring: false
    - security_groups:
      - my_sec_group_01
      - my_sec_group_02
```

availability_zones List of availability zones for the group.

min_size Minimum size of the group.

max_size Maximum size of the group.

desired_capacity The desired capacity of the group.

load_balancers List of load balancers for the group. Once set this can not be updated (Amazon restriction).

default_cooldown Number of seconds after a Scaling Activity completes before any further scaling activities can start.

health_check_type The service you want the health status from, Amazon EC2 or Elastic Load Balancer (EC2 or ELB).

health_check_period Length of time in seconds after a new EC2 instance comes into service that Auto Scaling starts checking its health.

placement_group Physical location of your cluster placement group created in Amazon EC2. Once set this can not be updated (Amazon restriction).

vpc_zone_identifier A list of the subnet identifiers of the Virtual Private Cloud.

subnet_names For VPC, a list of subnet names (NOT subnet IDs) to deploy into. Exclusive with vpc_zone_identifier.

tags A list of tags. Example:

```
- key: 'key'
  value: 'value'
  propagate_at_launch: true
```

termination_policies A list of termination policies. Valid values are:

- `OldestInstance`
- `NewestInstance`
- `OldestLaunchConfiguration`
- `ClosestToNextInstanceHour`
- `Default`

If no value is specified, the `Default` value is used.

termination_policies_from_pillar: name of pillar dict that contains termination policy settings. Termination policies defined for this specific state will override those from pillar.

suspended_processes List of processes to be suspended. see http://docs.aws.amazon.com/AutoScaling/latest/DeveloperGuide/US_SuspendResume.html

scaling_policies List of scaling policies. Each policy is a dict of key-values described by https://boto.readthedocs.io/en/latest/ref/autoscale.html#boto.ec2.autoscale.policy.ScalingPolicy

scaling_policies_from_pillar: name of pillar dict that contains scaling policy settings. Scaling policies defined for this specific state will override those from pillar.

scheduled_actions: a dictionary of scheduled actions. Each key is the name of scheduled action and each value is dictionary of options. For example:

```
- scheduled_actions:
    scale_up_at_10:
        desired_capacity: 4
        min_size: 3
        max_size: 5
        recurrence: "0 9 * * 1-5"
    scale_down_at_7:
        desired_capacity: 1
        min_size: 1
        max_size: 1
        recurrence: "0 19 * * 1-5"
```

scheduled_actions_from_pillar: name of pillar dict that contains scheduled_actions settings. Scheduled actions for this specific state will override those from pillar.

alarms: a dictionary of name->boto_cloudwatch_alarm sections to be associated with this ASG. All attributes should be specified except for dimension which will be automatically set to this ASG.

See the `salt.states.boto_cloudwatch_alarm` state for information about these attributes.

If any alarm actions include ":self:" this will be replaced with the asg name. For example, alarm_actions reading "['scaling_policy:self:ScaleUp']" will map to the arn for this asg's scaling policy named "ScaleUp". In addition, any alarms that have only scaling_policy as actions will be ignored if min_size is equal to max_size for this ASG.

alarms_from_pillar: name of pillar dict that contains alarm settings. Alarms defined for this specific state will override those from pillar.

region The region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

notification_arn The AWS arn that notifications will be sent to

notification_arn_from_pillar name of the pillar dict that contains `notifcation_arn` settings. A `notification_arn` defined for this specific state will override the one from pillar.

notification_types A list of event names that will trigger a notification. The list of valid notification types is:

- `autoscaling:EC2_INSTANCE_LAUNCH`
- `autoscaling:EC2_INSTANCE_LAUNCH_ERROR`
- `autoscaling:EC2_INSTANCE_TERMINATE`
- `autoscaling:EC2_INSTANCE_TERMINATE_ERROR`
- `autoscaling:TEST_NOTIFICATION`

notification_types_from_pillar name of the pillar dict that contains `notifcation_types` settings. `notification_types` defined for this specific state will override those from the pillar.

21.19.26 salt.states.boto_cfn

Connection module for Amazon Cloud Formation

New in version 2015.8.0.

depends boto

configuration This module accepts explicit AWS credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More Information available at http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-ec2.html

If IAM roles are not used you need to specify them either in a pillar or in the minion's config file:

```
keyid: GKTADJGHEIQSXMKKRBJ08H
key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

```
stack-present:
  boto_cfn.present:
    - name: mystack
    - template_body: salt://base/mytemplate.json
    - disable_rollback: true
    - region: eu-west-1
    - keyid: 'AKIAJHTMIQ2ASDFLASDF'
    - key: 'fdkjsafkljsASSADFalkfjasdf'
```

```
stack-absent:
  boto_cfn.absent:
    - name: mystack
```

`salt.states.boto_cfn.`**`absent`**(*name, region=None, key=None, keyid=None, profile=None*)

Ensure cloud formation stack is absent.

name (string) – The name of the stack to delete.

region (string) - Region to connect to.

key (string) - Secret key to be used.

keyid (string) - Access key to be used.

profile (dict) - A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_cfn.`**`present`**(*name, template_body=None, template_url=None, parameters=None, notification_arns=None, disable_rollback=None, timeout_in_minutes=None, capabilities=None, tags=None, on_failure=None, stack_policy_body=None, stack_policy_url=None, use_previous_template=None, stack_policy_during_update_body=None, stack_policy_during_update_url=None, region=None, key=None, keyid=None, profile=None*)

Ensure cloud formation stack is present.

name (string) - Name of the stack.

template_body (string) – Structure containing the template body. Can also be loaded from a file by using salt://.

template_url (string) – Location of file containing the template body. The URL must point to a template located in an S3 bucket in the same region as the stack.

parameters (list) – A list of key/value tuples that specify input parameters for the stack. A 3-tuple (key, value, bool) may be used to specify the UsePreviousValue option.

notification_arns (list) – The Simple Notification Service (SNS) topic ARNs to publish stack related events. You can find your SNS topic ARNs using the SNS_console or your Command Line Interface (CLI).

disable_rollback (bool) – Indicates whether or not to rollback on failure.

timeout_in_minutes (integer) – The amount of time that can pass before the stack status becomes CREATE_FAILED; if DisableRollback is not set or is set to False, the stack will be rolled back.

capabilities (list) – The list of capabilities you want to allow in the stack. Currently, the only valid capability is 'CAPABILITY_IAM'.

tags (dict) – A set of user-defined Tags to associate with this stack, represented by key/value pairs. Tags defined for the stack are propagated to EC2 resources that are created as part of the stack. A maximum number of 10 tags can be specified.

on_failure (string) – Determines what action will be taken if stack creation fails. This must be one of: DO_NOTHING, ROLLBACK, or DELETE. You can specify either OnFailure or DisableRollback, but not both.

stack_policy_body (string) – Structure containing the stack policy body. Can also be loaded from a file by using salt://.

stack_policy_url (string) – Location of a file containing the stack policy. The URL must point to a policy (max size: 16KB) located in an S3 bucket in the same region as the stack.If you pass StackPolicyBody and StackPolicyURL, only StackPolicyBody is used.

use_previous_template (boolean) – Used only when templates are not the same. Set to True to use the previous template instead of uploading a new one via TemplateBody or TemplateURL.

stack_policy_during_update_body (string) – Used only when templates are not the same. Structure containing the temporary overriding stack policy body. If you pass StackPolicyDuringUpdateBody and StackPolicyDuringUpdateURL, only StackPolicyDuringUpdateBody is used. Can also be loaded from a file by using salt://.

stack_policy_during_update_url (string) – Used only when templates are not the same. Location of a file containing the temporary overriding stack policy. The URL must point to a policy (max size: 16KB) located in an S3 bucket in the same region as the stack. If you pass StackPolicyDuringUpdateBody and StackPolicyDuringUpdateURL, only StackPolicyDuringUpdateBody is used.

region (string) - Region to connect to.

key (string) - Secret key to be used.

keyid (string) - Access key to be used.

profile (dict) - A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

21.19.27 salt.states.boto_cloudfront

Manage CloudFront distributions

New in version 2018.3.0.

Create, update and destroy CloudFront distributions.

This module accepts explicit AWS credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More information available here.

If IAM roles are not used you need to specify them, either in a pillar file or in the minion's config file:

```
cloudfront.keyid: GKTADJGHEIQSXMKKRBJ08H
cloudfront.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

It's also possible to specify `key`, `keyid`, and `region` via a profile, either passed in as a dict, or a string to pull from pillars or minion config:

```
myprofile:
    keyid: GKTADJGHEIQSXMKKRBJ08H
    key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
    region: us-east-1
```

```
aws:
    region:
        us-east-1:
            profile:
                keyid: GKTADJGHEIQSXMKKRBJ08H
                key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
                region: us-east-1
```

depends boto3

`salt.states.boto_cloudfront.`**`present`**(*name, config, tags, region=None, key=None, keyid=None, profile=None*)

Ensure the CloudFront distribution is present.

name (string) Name of the CloudFront distribution

config (dict) Configuration for the distribution

tags (dict) Tags to associate with the distribution

region (string) Region to connect to

key (string) Secret key to use

keyid (string) Access key to use

profile (dict or string) A dict with region, key, and keyid, or a pillar key (string) that contains such a dict.

Example:

```
Manage my_distribution CloudFront distribution:
    boto_cloudfront.present:
      - name: my_distribution
      - config:
          Comment: 'partial config shown, most parameters elided'
          Enabled: True
      - tags:
          testing_key: testing_value
```

21.19.28 salt.states.boto_cloudtrail module

Manage CloudTrail Objects

New in version 2016.3.0.

Create and destroy CloudTrail objects. Be aware that this interacts with Amazon's services, and so may incur charges.

depends

- boto
- boto3

The dependencies listed above can be installed via package or pip.

This module accepts explicit vpc credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More information available here.

If IAM roles are not used you need to specify them either in a pillar file or in the minion's config file:

```
vpc.keyid: GKTADJGHEIQSXMKKRBJ08H
vpc.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

It's also possible to specify `key`, `keyid` and `region` via a profile, either passed in as a dict, or as a string to pull from pillars or minion config:

```
myprofile:
    keyid: GKTADJGHEIQSXMKKRBJ08H
    key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
        region: us-east-1
```

```
Ensure trail exists:
    boto_cloudtrail.present:
        - Name: mytrail
        - S3BucketName: mybucket
        - S3KeyPrefix: prefix
        - region: us-east-1
        - keyid: GKTADJGHEIQSXMKKRBJ08H
        - key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

`salt.states.boto_cloudtrail.`**`absent`**(*name, Name, region=None, key=None, keyid=None, profile=None*)

Ensure trail with passed properties is absent.

name The name of the state definition.

Name Name of the trail.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_cloudtrail.`**`present`**(*name, Name, S3BucketName, S3KeyPrefix=None, SnsTopicName=None, IncludeGlobalServiceEvents=True, IsMultiRegionTrail=None, EnableLogFileValidation=False, CloudWatchLogsLogGroupArn=None, CloudWatchLogsRoleArn=None, KmsKeyId=None, LoggingEnabled=True, Tags=None, region=None, key=None, keyid=None, profile=None*)

Ensure trail exists.

name The name of the state definition

Name Name of the trail.

S3BucketName Specifies the name of the Amazon S3 bucket designated for publishing log files.

S3KeyPrefix Specifies the Amazon S3 key prefix that comes after the name of the bucket you have designated for log file delivery.

SnsTopicName Specifies the name of the Amazon SNS topic defined for notification of log file delivery. The maximum length is 256 characters.

IncludeGlobalServiceEvents Specifies whether the trail is publishing events from global services such as IAM to the log files.

EnableLogFileValidation Specifies whether log file integrity validation is enabled. The default is false.

CloudWatchLogsLogGroupArn Specifies a log group name using an Amazon Resource Name (ARN), a unique identifier that represents the log group to which CloudTrail logs will be delivered. Not required unless you specify CloudWatchLogsRoleArn.

CloudWatchLogsRoleArn Specifies the role for the CloudWatch Logs endpoint to assume to write to a user's log group.

KmsKeyId Specifies the KMS key ID to use to encrypt the logs delivered by CloudTrail. The value can be a an alias name prefixed by "alias/", a fully specified ARN to an alias, a fully specified ARN to a key, or a globally unique identifier.

LoggingEnabled Whether logging should be enabled for the trail

Tags A dictionary of tags that should be set on the trail

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

21.19.29 salt.states.boto_cloudwatch_alarm

Manage Cloudwatch alarms

New in version 2014.7.0.

Create and destroy cloudwatch alarms. Be aware that this interacts with Amazon's services, and so may incur charges.

This module uses boto, which can be installed via package, or pip.

This module accepts explicit credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More Information available at:

http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-ec2.html

If IAM roles are not used you need to specify them either in a pillar or in the minion's config file:

```
cloudwatch.keyid: GKTADJGHEIQSXMKKRBJ08H
cloudwatch.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

It's also possible to specify key, keyid and region via a profile, either as a passed in dict, or as a string to pull from pillars or minion config:

```
myprofile:
  keyid: GKTADJGHEIQSXMKKRBJ08H
  key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
  region: us-east-1
```

```
my test alarm:
  boto_cloudwatch_alarm.present:
    - name: my test alarm
    - attributes:
        metric: ApproximateNumberOfMessagesVisible
        namespace: AWS/SQS
        statistic: Average
        comparison: ">="
        threshold: 20000.0
        period: 60
        evaluation_periods: 1
        description: test alarm via salt
        dimensions:
          QueueName:
            - the-sqs-queue-name
        alarm_actions:
          - arn:aws:sns:us-east-1:1111111:myalerting-action
```

`salt.states.boto_cloudwatch_alarm.`**`absent`**`(`*name, region=None, key=None, keyid=None, profile=None*`)`

Ensure the named cloudwatch alarm is deleted.

name Name of the alarm.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_cloudwatch_alarm.`**`present`**`(`*name, attributes, region=None, key=None, keyid=None, profile=None*`)`

Ensure the cloudwatch alarm exists.

name Name of the alarm

attributes A dict of key/value cloudwatch alarm attributes.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

21.19.30 salt.states.boto_cloudwatch_event module

Manage CloudTrail Objects

New in version 2016.11.0.

Create and destroy CloudWatch event rules. Be aware that this interacts with Amazon's services, and so may incur charges.

This module uses `boto3`, which can be installed via package, or pip.

This module accepts explicit vpc credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More information available here.

If IAM roles are not used you need to specify them either in a pillar file or in the minion's config file:

```
cloudwatch_event.keyid: GKTADJGHEIQSXMKKRBJ08H
cloudwatch_event.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

It's also possible to specify `key`, `keyid` and `region` via a profile, either passed in as a dict, or as a string to pull from pillars or minion config:

```
myprofile:
    keyid: GKTADJGHEIQSXMKKRBJ08H
    key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
        region: us-east-1
```

```
Ensure event rule exists:
    boto_cloudwatch_event.present:
        - Name: mytrail
        - ScheduleExpression: 'rate(120 minutes)'
        - State: 'DISABLED'
        - Targets:
          - Id: "target1"
```

(continues on next page)

(continued from previous page)

```
          Arn: "arn:aws:lambda:us-west-1:124456715622:function:my_function"
          Input: '{"arbitrary": "json"}'
        - region: us-east-1
        - keyid: GKTADJGHEIQSXMKKRBJ08H
        - key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

`salt.states.boto_cloudwatch_event.`**`absent`**`(`*name, Name=None, region=None, key=None, keyid=None, profile=None*`)`

Ensure CloudWatch event rule with passed properties is absent.

name The name of the state definition.

Name Name of the event rule. Defaults to the value of the 'name' param if not provided.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_cloudwatch_event.`**`present`**`(`*name, Name=None, ScheduleExpression=None, EventPattern=None, Description=None, RoleArn=None, State=None, Targets=None, region=None, key=None, keyid=None, profile=None*`)`

Ensure trail exists.

name The name of the state definition

Name Name of the event rule. Defaults to the value of the 'name' param if not provided.

ScheduleExpression The scheduling expression. For example, `cron(0 20 * * ? *)`, "rate(5 minutes)"

EventPattern The event pattern.

Description A description of the rule

State Indicates whether the rule is ENABLED or DISABLED.

RoleArn The Amazon Resource Name (ARN) of the IAM role associated with the rule.

Targets A list of rresources to be invoked when the rule is triggered.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

21.19.31 salt.states.boto_cognitoidentity module

Manage CognitoIdentity Functions

New in version 2016.11.0.

Create and destroy CognitoIdentity identity pools. Be aware that this interacts with Amazon's services, and so may incur charges.

This module uses `boto3`, which can be installed via package, or pip.

This module accepts explicit vpc credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More information available here.

If IAM roles are not used you need to specify them either in a pillar file or in the minion's config file:

```
vpc.keyid: GKTADJGHEIQSXMKKRBJ08H
vpc.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

It's also possible to specify `key`, `keyid` and `region` via a profile, either passed in as a dict, or as a string to pull from pillars or minion config:

```
myprofile:
    keyid: GKTADJGHEIQSXMKKRBJ08H
    key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
        region: us-east-1
```

```
Ensure function exists:
    boto_cognitoidentity.pool_present:
        - PoolName: my_identity_pool
        - region: us-east-1
        - keyid: GKTADJGHEIQSXMKKRBJ08H
        - key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

`salt.states.boto_cognitoidentity.`**`pool_absent`**`(`*name, IdentityPoolName, RemoveAllMatched=False, region=None, key=None, keyid=None, profile=None*`)`

Ensure cognito identity pool with passed properties is absent.

name The name of the state definition.

IdentityPoolName Name of the Cognito Identity Pool. Please note that this may match multiple pools with the same given name, in which case, all will be removed.

RemoveAllMatched If True, all identity pools with the matching IdentityPoolName will be removed. If False and there are more than one identity pool with the matching IdentityPoolName, no action will be taken. If False and there is only one identity pool with the matching IdentityPoolName, the identity pool will be removed.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_cognitoidentity.`**`pool_present`**`(`*name, IdentityPoolName, AuthenticatedRole, AllowUnauthenticatedIdentities=False, UnauthenticatedRole=None, SupportedLoginProviders=None, DeveloperProviderName=None, OpenIdConnectProviderARNs=None, region=None, key=None, keyid=None, profile=None*`)`

Ensure Cognito Identity Pool exists.

name The name of the state definition

IdentityPoolName Name of the Cognito Identity Pool

AuthenticatedRole An IAM role name or ARN that will be associated with temporary AWS credentials for an authenticated cognito identity.

AllowUnauthenticatedIdentities Whether to allow anonymous user identities

UnauthenticatedRole An IAM role name or ARN that will be associated with anonymous user identities

SupportedLoginProviders A dictionary or pillar that contains key:value pairs mapping provider names to provider app IDs.

DeveloperProviderName A string which is the domain by which Cognito will refer to your users. This name acts as a placeholder that allows your backend and the Cognito service to communicate about the developer provider. Once you have set a developer provider name, you cannot change it. Please take care in setting this parameter.

OpenIdConnectProviderARNs A list or pillar name that contains a list of OpenID Connect provider ARNs.
region Region to connect to.
key Secret key to be used.
keyid Access key to be used.
profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

21.19.32 salt.states.boto_datapipeline module

Manage Data Pipelines

New in version 2016.3.0.

Be aware that this interacts with Amazon's services, and so may incur charges.

This module uses `boto3`, which can be installed via package, or pip.

This module accepts explicit AWS credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More information available here.

If IAM roles are not used you need to specify them either in a pillar file or in the minion's config file:

```
datapipeline.keyid: GKTADJGHEIQSXMKKRBJ08H
datapipeline.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

It's also possible to specify `key`, `keyid` and `region` via a profile, either passed in as a dict, or as a string to pull from pillars or minion config:

```
myprofile:
  keyid: GKTADJGHEIQSXMKKRBJ08H
  key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
  region: us-east-1
```

```
Ensure daily data pipeline exists:
  boto_datapipeline.present:
    - name: my-datapipeline
    - pipeline_objects:
        DefaultSchedule:
          name: Every 1 day
          fields:
            period: 1 Day
            type: Schedule
            startAt: FIRST_ACTIVATION_DATE_TIME
    - parameter_values:
        myDDBTableName: my-dynamo-table
```

`salt.states.boto_datapipeline.`**`absent`**`(`*name, region=None, key=None, keyid=None, profile=None*`)`

Ensure a pipeline with the service_name does not exist
name Name of the service to ensure a data pipeline does not exist for.
region Region to connect to.
key Secret key to be used.
keyid Access key to be used.
profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_datapipeline.present`(*name*, *pipeline_objects=None*, *pipeline_objects_from_pillars='boto_datapipeline_pipeline_objects'*, *parameter_objects=None*, *parameter_objects_from_pillars='boto_datapipeline_parameter_objects'*, *parameter_values=None*, *parameter_values_from_pillars='boto_datapipeline_parameter_values'*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Ensure the data pipeline exists with matching definition.

name Name of the service to ensure a data pipeline exists for.

pipeline_objects Pipeline objects to use. Will override objects read from pillars.

pipeline_objects_from_pillars The pillar key to use for lookup.

parameter_objects Parameter objects to use. Will override objects read from pillars.

parameter_objects_from_pillars The pillar key to use for lookup.

parameter_values Parameter values to use. Will override values read from pillars.

parameter_values_from_pillars The pillar key to use for lookup.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

21.19.33 salt.states.boto_dynamodb

Manage DynamoDB Tables

New in version 2015.5.0.

Create and destroy DynamoDB tables. Be aware that this interacts with Amazon's services, and so may incur charges.

This module uses `boto`, which can be installed via package, or pip.

This module accepts explicit DynamoDB credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More information available here.

If IAM roles are not used you need to specify them either in a pillar file or in the minion's config file:

```
keyid: GKTADJGHEIQSXMKKRBJ08H
key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
region: us-east-1
```

It's also possible to specify `key`, `keyid` and `region` via a profile, either passed in as a dict, or as a string to pull from pillars or minion config:

```
myprofile:
    keyid: GKTADJGHEIQSXMKKRBJ08H
    key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
        region: us-east-1
```

```
Ensure DynamoDB table does not exist:
  boto_dynamodb.absent:
    - table_name: new_table
    - keyid: GKTADJGHEIQSXMKKRBJ08H
    - key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
    - region: us-east-1
```

(continues on next page)

(continued from previous page)

```
Ensure DynamoDB table exists:
  boto_dynamodb.present:
    - table_name: new_table
    - read_capacity_units: 1
    - write_capacity_units: 2
    - hash_key: primary_id
    - hash_key_data_type: N
    - range_key: start_timestamp
    - range_key_data_type: N
    - keyid: GKTADJGHEIQSXMKKRBJ08H
    - key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
    - region: us-east-1
    - local_indexes:
        - index:
            - name: "primary_id_end_timestamp_index"
            - hash_key: primary_id
            - hash_key_data_type: N
            - range_key: end_timestamp
            - range_key_data_type: N
    - global_indexes:
        - index:
            - name: "name_end_timestamp_index"
            - hash_key: name
            - hash_key_data_type: S
            - range_key: end_timestamp
            - range_key_data_type: N
            - read_capacity_units: 3
            - write_capacity_units: 4
```

It's possible to specify cloudwatch alarms that will be setup along with the DynamoDB table. Note the alarm name will be defined by the name attribute provided, plus the DynamoDB resource name.

```
Ensure DynamoDB table exists:
  boto_dynamodb.present:
    - name: new_table
    - read_capacity_units: 1
    - write_capacity_units: 2
    - hash_key: primary_id
    - hash_key_data_type: N
    - range_key: start_timestamp
    - range_key_data_type: N
    - alarms:
         ConsumedWriteCapacityUnits:
            name: 'DynamoDB ConsumedWriteCapacityUnits **MANAGED BY SALT**'
            attributes:
              metric: ConsumedWriteCapacityUnits
              namespace: AWS/DynamoDB
              statistic: Sum
              comparison: '>='
              # threshold_percent is used to calculate the actual threshold
              # based on the provisioned capacity for the table.
              threshold_percent: 0.75
              period: 300
              evaluation_periods: 2
              unit: Count
```

(continues on next page)

(continued from previous page)

```
              description: 'DynamoDB ConsumedWriteCapacityUnits'
              alarm_actions: [ 'arn:aws:sns:us-east-1:1234:my-alarm' ]
              insufficient_data_actions: []
              ok_actions: [ 'arn:aws:sns:us-east-1:1234:my-alarm' ]
    - keyid: GKTADJGHEIQSXMKKRBJ08H
    - key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
    - region: us-east-1
```

You can also use alarms from pillars, and override values from the pillar alarms by setting overrides on the resource. Note that 'boto_dynamodb_alarms' will be used as a default value for all resources, if defined and can be used to ensure alarms are always set for a resource.

Setting the alarms in a pillar:

```
boto_dynamodb_alarms:
  ConsumedWriteCapacityUnits:
    name: 'DynamoDB ConsumedWriteCapacityUnits **MANAGED BY SALT**'
    attributes:
      metric: ConsumedWriteCapacityUnits
      namespace: AWS/DynamoDB
      statistic: Sum
      comparison: '>='
      # threshold_percent is used to calculate the actual threshold
      # based on the provisioned capacity for the table.
      threshold_percent: 0.75
      period: 300
      evaluation_periods: 2
      unit: Count
      description: 'DynamoDB ConsumedWriteCapacityUnits'
      alarm_actions: [ 'arn:aws:sns:us-east-1:1234:my-alarm' ]
      insufficient_data_actions: []
      ok_actions: [ 'arn:aws:sns:us-east-1:1234:my-alarm' ]

Ensure DynamoDB table exists:
  boto_dynamodb.present:
    - name: new_table
    - read_capacity_units: 1
    - write_capacity_units: 2
    - hash_key: primary_id
    - hash_key_data_type: N
    - range_key: start_timestamp
    - range_key_data_type: N
    - alarms:
         ConsumedWriteCapacityUnits:
            attributes:
              threshold_percent: 0.90
              period: 900
```

exception `salt.states.boto_dynamodb.`**`GsiNotUpdatableError`**

Raised when a global secondary index cannot be updated.

`salt.states.boto_dynamodb.`**`absent`**(*name, region=None, key=None, keyid=None, profile=None*)

Ensure the DynamoDB table does not exist.

name Name of the DynamoDB table.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_dynamodb.`**`present`**`(`*name=None, table_name=None, region=None, key=None, keyid=None, profile=None, read_capacity_units=None, write_capacity_units=None, alarms=None, alarms_from_pillar='boto_dynamodb_alarms', hash_key=None, hash_key_data_type=None, range_key=None, range_key_data_type=None, local_indexes=None, global_indexes=None, backup_configs_from_pillars='boto_dynamodb_backup_configs'*`)`

Ensure the DynamoDB table exists. Table throughput can be updated after table creation.

Global secondary indexes (GSIs) are managed with some exceptions:

- If a GSI deletion is detected, a failure will occur (deletes should be done manually in the AWS console).
- If multiple GSIs are added in a single Salt call, a failure will occur (boto supports one creation at a time). Note that this only applies after table creation; multiple GSIs can be created during table creation.
- Updates to existing GSIs are limited to read/write capacity only (DynamoDB limitation).

name Name of the DynamoDB table

table_name Name of the DynamoDB table (deprecated)

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

read_capacity_units The read throughput for this table

write_capacity_units The write throughput for this table

hash_key The name of the attribute that will be used as the hash key for this table

hash_key_data_type The DynamoDB datatype of the hash key

range_key The name of the attribute that will be used as the range key for this table

range_key_data_type The DynamoDB datatype of the range key

local_indexes The local indexes you would like to create

global_indexes The global indexes you would like to create

backup_configs_from_pillars Pillars to use to configure DataPipeline backups

21.19.34 salt.states.boto_ec2

Manage EC2

New in version 2015.8.0.

This module provides an interface to the Elastic Compute Cloud (EC2) service from AWS.

The below code creates a key pair:

```
create-key-pair:
  boto_ec2.key_present:
    - name: mykeypair
    - save_private: /root/
    - region: eu-west-1
    - keyid: GKTADJGHEIQSXMKKRBJ08H
    - key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

```
import-key-pair:
   boto_ec2.key_present:
```

(continues on next page)

(continued from previous page)

```
    - name: mykeypair
    - upload_public: 'ssh-rsa AAAA'
    - keyid: GKTADJGHEIQSXMKKRBJ08H
    - key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

You can also use salt:// in order to define the public key.

```
import-key-pair:
   boto_ec2.key_present:
    - name: mykeypair
    - upload_public: salt://mybase/public_key.pub
    - keyid: GKTADJGHEIQSXMKKRBJ08H
    - key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

The below code deletes a key pair:

```
delete-key-pair:
  boto_ec2.key_absent:
    - name: mykeypair
    - region: eu-west-1
    - keyid: GKTADJGHEIQSXMKKRBJ08H
    - key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

`salt.states.boto_ec2.`**`eni_absent`**(*name*, *release_eip=False*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Ensure the EC2 ENI is absent.

New in version 2016.3.0.

name Name tag associated with the ENI.
release_eip True/False - release any EIP associated with the ENI
region Region to connect to.
key Secret key to be used.
keyid Access key to be used.
profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_ec2.`**`eni_present`**(*name*, *subnet_id=None*, *subnet_name=None*, *private_ip_address=None*, *description=None*, *groups=None*, *source_dest_check=True*, *allocate_eip=None*, *arecords=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Ensure the EC2 ENI exists.

New in version 2016.3.0.

name Name tag associated with the ENI.
subnet_id The VPC subnet ID the ENI will exist within.
subnet_name The VPC subnet name the ENI will exist within.
private_ip_address The private ip address to use for this ENI. If this is not specified AWS will automatically assign a private IP address to the ENI. Must be specified at creation time; will be ignored afterward.
description Description of the key.
groups A list of security groups to apply to the ENI.
source_dest_check Boolean specifying whether source/destination checking is enabled on the ENI.
allocate_eip allocate and associate an EIP to the ENI. Could be 'standard' to allocate Elastic IP to EC2 region or 'vpc' to get it for a particular VPC

Changed in version 2016.11.0.

arecords A list of arecord dicts with attributes needed for the DNS add_record state. By default the boto_route53.add_record state will be used, which requires: name, zone, ttl, and identifier. See the boto_route53 state for information about these attributes. Other DNS modules can be called by specifying the provider keyword. By default, the private ENI IP address will be used, set 'public: True' in the arecord dict to use the ENI's public IP address

New in version 2016.3.0.

region Region to connect to.
key Secret key to be used.
keyid Access key to be used.
profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_ec2.`**`instance_absent`**`(`*name, instance_name=None, instance_id=None, release_eip=False, region=None, key=None, keyid=None, profile=None, filters=None*`)`

Ensure an EC2 instance does not exist (is stopped and removed).

Changed in version 2016.11.0.
name (string) - The name of the state definition.
instance_name (string) - The name of the instance.
instance_id (string) - The ID of the instance.
release_eip (bool) - Release any associated EIPs during termination.
region (string) - Region to connect to.
key (string) - Secret key to be used.
keyid (string) - Access key to be used.
profile (variable) - A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.
filters (dict) - A dict of additional filters to use in matching the instance to delete.
YAML example fragment:

```
- filters:
    vpc-id: vpc-abcdef12
```

`salt.states.boto_ec2.`**`instance_present`**`(`*name, instance_name=None, instance_id=None, image_id=None, image_name=None, tags=None, key_name=None, security_groups=None, user_data=None, instance_type=None, placement=None, kernel_id=None, ramdisk_id=None, vpc_id=None, vpc_name=None, monitoring_enabled=None, subnet_id=None, subnet_name=None, private_ip_address=None, block_device_map=None, disable_api_termination=None, instance_initiated_shutdown_behavior=None, placement_group=None, client_token=None, security_group_ids=None, security_group_names=None, additional_info=None, tenancy=None, instance_profile_arn=None, instance_profile_name=None, ebs_optimized=None, network_interfaces=None, network_interface_name=None, network_interface_id=None, attributes=None, target_state=None, public_ip=None, allocation_id=None, allocate_eip=False, region=None, key=None, keyid=None, profile=None*`)`

Ensure an EC2 instance is running with the given attributes and state.
name (string) - The name of the state definition. Recommended that this match the instance_name attribute

(generally the FQDN of the instance).

instance_name (string) - The name of the instance, generally its FQDN. Exclusive with 'instance_id'.

instance_id (string) - The ID of the instance (if known). Exclusive with 'instance_name'.

image_id (string) – The ID of the AMI image to run.

image_name (string) – The name of the AMI image to run.

tags (dict) - Tags to apply to the instance.

key_name (string) – The name of the key pair with which to launch instances.

security_groups (list of strings) – The names of the EC2 classic security groups with which to associate instances

user_data (string) – The Base64-encoded MIME user data to be made available to the instance(s) in this reservation.

instance_type (string) – The EC2 instance size/type. Note that only certain types are compatible with HVM based AMIs.

placement (string) – The Availability Zone to launch the instance into.

kernel_id (string) – The ID of the kernel with which to launch the instances.

ramdisk_id (string) – The ID of the RAM disk with which to launch the instances.

vpc_id (string) - The ID of a VPC to attach the instance to.

vpc_name (string) - The name of a VPC to attach the instance to.

monitoring_enabled (bool) – Enable detailed CloudWatch monitoring on the instance.

subnet_id (string) – The ID of the subnet within which to launch the instances for VPC.

subnet_name (string) – The name of the subnet within which to launch the instances for VPC.

private_ip_address (string) – If you're using VPC, you can optionally use this parameter to assign the instance a specific available IP address from the subnet (e.g., 10.0.0.25).

block_device_map (boto.ec2.blockdevicemapping.BlockDeviceMapping) – A BlockDeviceMapping data structure describing the EBS volumes associated with the Image.

disable_api_termination (bool) – If True, the instances will be locked and will not be able to be terminated via the API.

instance_initiated_shutdown_behavior (string) – Specifies whether the instance stops or terminates on instance-initiated shutdown. Valid values are:

- 'stop'
- 'terminate'

placement_group (string) – If specified, this is the name of the placement group in which the instance(s) will be launched.

client_token (string) – Unique, case-sensitive identifier you provide to ensure idempotency of the request. Maximum 64 ASCII characters.

security_group_ids (list of strings) – The IDs of the VPC security groups with which to associate instances.

security_group_names (list of strings) – The names of the VPC security groups with which to associate instances.

additional_info (string) – Specifies additional information to make available to the instance(s).

tenancy (string) – The tenancy of the instance you want to launch. An instance with a tenancy of 'dedicated' runs on single-tenant hardware and can only be launched into a VPC. Valid values are:"default" or "dedicated". NOTE: To use dedicated tenancy you MUST specify a VPC subnet-ID as well.

instance_profile_arn (string) – The Amazon resource name (ARN) of the IAM Instance Profile (IIP) to associate with the instances.

instance_profile_name (string) – The name of the IAM Instance Profile (IIP) to associate with the instances.

ebs_optimized (bool) – Whether the instance is optimized for EBS I/O. This optimization provides dedicated throughput to Amazon EBS and a tuned configuration stack to provide optimal EBS I/O performance. This optimization isn't available with all instance types.

network_interfaces (boto.ec2.networkinterface.NetworkInterfaceCollection) – A NetworkInterfaceCollection data structure containing the ENI specifications for the instance.

network_interface_name

(string) - The name of Elastic Network Interface to attach

New in version 2016.11.0.

network_interface_id

(string) - The id of Elastic Network Interface to attach

New in version 2016.11.0.

attributes (dict) - Instance attributes and value to be applied to the instance. Available options are:

- instanceType - A valid instance type (m1.small)
- kernel - Kernel ID (None)
- ramdisk - Ramdisk ID (None)
- userData - Base64 encoded String (None)
- disableApiTermination - Boolean (true)
- instanceInitiatedShutdownBehavior - stop|terminate
- blockDeviceMapping - List of strings - ie: ['/dev/sda=false']
- sourceDestCheck - Boolean (true)
- groupSet - Set of Security Groups or IDs
- ebsOptimized - Boolean (false)
- sriovNetSupport - String - ie: 'simple'

target_state (string) - The desired target state of the instance. Available options are:

- running
- stopped

Note that this option is currently UNIMPLEMENTED.

public_ip: (string) - The IP of a previously allocated EIP address, which will be attached to the instance. EC2 Classic instances ONLY - for VCP pass in an allocation_id instead.

allocation_id: (string) - The ID of a previously allocated EIP address, which will be attached to the instance. VPC instances ONLY - for Classic pass in a public_ip instead.

allocate_eip: (bool) - Allocate and attach an EIP on-the-fly for this instance. Note you'll want to releaase this address when terminating the instance, either manually or via the 'release_eip' flag to 'instance_absent'.

region (string) - Region to connect to.

key (string) - Secret key to be used.

keyid (string) - Access key to be used.

profile (variable) - A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

New in version 2016.3.0.

`salt.states.boto_ec2.`**`key_absent`**(*name*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Deletes a key pair

`salt.states.boto_ec2.`**`key_present`**(*name*, *save_private=None*, *upload_public=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Ensure key pair is present.

`salt.states.boto_ec2.`**`private_ips_absent`**(*name*, *network_interface_name=None*, *network_interface_id=None*, *private_ip_addresses=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Ensure an ENI does not have secondary private ip addresses associated with it

name (String) - State definition name

network_interface_id (String) - The EC2 network interface id, example eni-123456789

private_ip_addresses (List or String) - The secondary private ip address(es) that should be absent on the ENI.

region (string) - Region to connect to.

key (string) - Secret key to be used.

keyid (string) - Access key to be used.

profile (variable) - A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_ec2.`**`private_ips_present`**`(`*name*, *network_interface_name=None*, *network_interface_id=None*, *private_ip_addresses=None*, *allow_reassignment=False*, *region=None*, *key=None*, *keyid=None*, *profile=None*`)`

Ensure an ENI has secondary private ip addresses associated with it

name (String) - State definition name

network_interface_id (String) - The EC2 network interface id, example eni-123456789

private_ip_addresses (List or String) - The secondary private ip address(es) that should be present on the ENI.

allow_reassignment (Boolean) - If true, will reassign a secondary private ip address associated with another ENI. If false, state will fail if the secondary private ip address is associated with another ENI.

region (string) - Region to connect to.

key (string) - Secret key to be used.

keyid (string) - Access key to be used.

profile (variable) - A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_ec2.`**`snapshot_created`**`(`*name*, *ami_name*, *instance_name*, *wait_until_available=True*, *wait_timeout_seconds=300*, ***kwargs*`)`

Create a snapshot from the given instance

New in version 2016.3.0.

`salt.states.boto_ec2.`**`volume_absent`**`(`*name*, *volume_name=None*, *volume_id=None*, *instance_name=None*, *instance_id=None*, *device=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*`)`

Ensure the EC2 volume is detached and absent.

New in version 2016.11.0.

name State definition name.

volume_name Name tag associated with the volume. For safety, if this matches more than one volume, the state will refuse to apply.

volume_id Resource ID of the volume.

instance_name Only remove volume if it is attached to instance with this Name tag. Exclusive with 'instance_id'. Requires 'device'.

instance_id Only remove volume if it is attached to this instance. Exclusive with 'instance_name'. Requires 'device'.

device Match by device rather than ID. Requires one of 'instance_name' or 'instance_id'.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_ec2.`**`volume_present`**`(`*name*, *volume_name=None*, *volume_id=None*, *instance_name=None*, *instance_id=None*, *device=None*, *size=None*, *snapshot_id=None*, *volume_type=None*, *iops=None*, *encrypted=False*, *kms_key_id=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*`)`

Ensure the EC2 volume is present and attached.

name State definition name.

volume_name The Name tag value for the volume. If no volume with that matching name tag is found, a new volume will be created. If multiple volumes are matched, the state will fail.

volume_id Resource ID of the volume. Exclusive with 'volume_name'.

instance_name Attach volume to instance with this Name tag. Exclusive with 'instance_id'.

instance_id Attach volume to instance with this ID. Exclusive with 'instance_name'.

device The device on the instance through which the volume is exposed (e.g. /dev/sdh)

size The size of the new volume, in GiB. If you're creating the volume from a snapshot and don't specify a volume size, the default is the snapshot size. Optionally specified at volume creation time; will be ignored afterward. Requires 'volume_name'.

snapshot_id The snapshot ID from which the new Volume will be created. Optionally specified at volume creation time; will be ignored afterward. Requires 'volume_name'.

volume_type The type of the volume. Optionally specified at volume creation time; will be ignored afterward. Requires 'volume_name'. Valid volume types for AWS can be found here: http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSVolumeTypes.html

iops The provisioned IOPS you want to associate with this volume. Optionally specified at volume creation time; will be ignored afterward. Requires 'volume_name'.

encrypted Specifies whether the volume should be encrypted. Optionally specified at volume creation time; will be ignored afterward. Requires 'volume_name'.

kms_key_id If encrypted is True, this KMS Key ID may be specified to encrypt volume with this key. Optionally specified at volume creation time; will be ignored afterward. Requires 'volume_name'. e.g.: arn:aws:kms:us-east-1:012345678910:key/abcd1234-a123-456a-a12b-a123b4cd56ef

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_ec2.`**`volumes_tagged`**(*name, tag_maps, authoritative=False, region=None, key=None, keyid=None, profile=None*)

Ensure EC2 volume(s) matching the given filters have the defined tags.

New in version 2016.11.0.

name State definition name.

tag_maps List of dicts of filters and tags, where 'filters' is a dict suitable for passing to the 'filters' argument of boto_ec2.get_all_volumes(), and 'tags' is a dict of tags to be set on volumes as matched by the given filters. The filter syntax is extended to permit passing either a list of volume_ids or an instance_name (with instance_name being the Name tag of the instance to which the desired volumes are mapped). Each mapping in the list is applied separately, so multiple sets of volumes can be all tagged differently with one call to this function.

YAML example fragment:

```
- filters:
    attachment.instance_id: i-abcdef12
  tags:
    Name: dev-int-abcdef12.aws-foo.com
- filters:
    attachment.device: /dev/sdf
  tags:
    ManagedSnapshots: true
    BillingGroup: bubba.hotep@aws-foo.com
- filters:
    instance_name: prd-foo-01.aws-foo.com
  tags:
    Name: prd-foo-01.aws-foo.com
    BillingGroup: infra-team@aws-foo.com
- filters:
    volume_ids: [ vol-12345689, vol-abcdef12 ]
  tags:
    BillingGroup: infra-team@aws-foo.com
```

authoritative Should un-declared tags currently set on matched volumes be deleted? Boolean.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

21.19.35 salt.states.boto_elasticache

Manage Elasticache

New in version 2014.7.0.

Create, destroy and update Elasticache clusters. Be aware that this interacts with Amazon's services, and so may incur charges.

Note: This module currently only supports creation and deletion of elasticache resources and will not modify clusters when their configuration changes in your state files.

This module uses `boto`, which can be installed via package, or pip.

This module accepts explicit elasticache credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More information available here.

If IAM roles are not used you need to specify them either in a pillar file or in the minion's config file:

```
elasticache.keyid: GKTADJGHEIQSXMKKRBJ08H
elasticache.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

It's also possible to specify `key`, `keyid` and `region` via a profile, either passed in as a dict, or as a string to pull from pillars or minion config:

```
myprofile:
  keyid: GKTADJGHEIQSXMKKRBJ08H
  key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
    region: us-east-1
```

```
Ensure myelasticache exists:
  boto_elasticache.present:
    - name: myelasticache
    - engine: redis
    - cache_node_type: cache.t1.micro
    - num_cache_nodes: 1
    - notification_topic_arn: arn:aws:sns:us-east-1:879879:my-sns-topic
    - region: us-east-1
    - keyid: GKTADJGHEIQSXMKKRBJ08H
    - key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs

# Using a profile from pillars
Ensure myelasticache exists:
  boto_elasticache.present:
    - name: myelasticache
    - engine: redis
    - cache_node_type: cache.t1.micro
    - num_cache_nodes: 1
    - notification_topic_arn: arn:aws:sns:us-east-1:879879:my-sns-topic
    - region: us-east-1
    - profile: myprofile
```

(continues on next page)

(continued from previous page)

```
# Passing in a profile
Ensure myelasticache exists:
  boto_elasticache.present:
    - name: myelasticache
    - engine: redis
    - cache_node_type: cache.t1.micro
    - num_cache_nodes: 1
    - notification_topic_arn: arn:aws:sns:us-east-1:879879:my-sns-topic
    - region: us-east-1
    - profile:
        keyid: GKTADJGHEIQSXMKKRBJ08H
        key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

`salt.states.boto_elasticache.`**`absent`**(*name*, *wait=True*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Ensure the named elasticache cluster is deleted.

name Name of the cache cluster.

wait Boolean. Wait for confirmation from boto that the cluster is in the deleting state.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_elasticache.`**`cache_cluster_absent`**(**args*, ***kwargs*)

`salt.states.boto_elasticache.`**`cache_cluster_present`**(**args*, ***kwargs*)

`salt.states.boto_elasticache.`**`creategroup`**(*name*, *primary_cluster_id*, *replication_group_description*, *wait=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Ensure the a replication group is create.

name Name of replication group

wait Waits for the group to be available

primary_cluster_id Name of the master cache node

replication_group_description Description for the group

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_elasticache.`**`present`**(*name*, *engine=None*, *cache_node_type=None*, *num_cache_nodes=None*, *preferred_availability_zone=None*, *port=None*, *cache_parameter_group_name=None*, *cache_security_group_names=None*, *replication_group_id=None*, *auto_minor_version_upgrade=True*, *security_group_ids=None*, *cache_subnet_group_name=None*, *engine_version=None*, *notification_topic_arn=None*, *preferred_maintenance_window=None*, *wait=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Ensure the cache cluster exists.

name Name of the cache cluster (cache cluster id).

engine The name of the cache engine to be used for this cache cluster. Valid values are memcached or redis.

cache_node_type The compute and memory capacity of the nodes in the cache cluster. cache.t1.micro, cache.m1.small, etc. See: https://boto.readthedocs.io/en/latest/ref/elasticache.html#boto.elasticache.layer1.ElastiCacheConnection.create_cache_cluster

num_cache_nodes The number of cache nodes that the cache cluster will have.

preferred_availability_zone The EC2 Availability Zone in which the cache cluster will be created. All cache nodes belonging to a cache cluster are placed in the preferred availability zone.

port The port number on which each of the cache nodes will accept connections.

cache_parameter_group_name The name of the cache parameter group to associate with this cache cluster. If this argument is omitted, the default cache parameter group for the specified engine will be used.

cache_security_group_names A list of cache security group names to associate with this cache cluster. Use this parameter only when you are creating a cluster outside of a VPC.

replication_group_id The replication group to which this cache cluster should belong. If this parameter is specified, the cache cluster will be added to the specified replication group as a read replica; otherwise, the cache cluster will be a standalone primary that is not part of any replication group.

auto_minor_version_upgrade Determines whether minor engine upgrades will be applied automatically to the cache cluster during the maintenance window. A value of True allows these upgrades to occur; False disables automatic upgrades.

security_group_ids One or more VPC security groups associated with the cache cluster. Use this parameter only when you are creating a cluster in a VPC.

cache_subnet_group_name The name of the cache subnet group to be used for the cache cluster. Use this parameter only when you are creating a cluster in a VPC.

engine_version The version number of the cache engine to be used for this cluster.

notification_topic_arn The Amazon Resource Name (ARN) of the Amazon Simple Notification Service (SNS) topic to which notifications will be sent. The Amazon SNS topic owner must be the same as the cache cluster owner.

preferred_maintenance_window The weekly time range (in UTC) during which system maintenance can occur. Example: sun:05:00-sun:09:00

wait Boolean. Wait for confirmation from boto that the cluster is in the available state.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_elasticache.`**`replication_group_absent`**`(`*name, tags=None, region=None, key=None, keyid=None, profile=None*`)`

`salt.states.boto_elasticache.`**`replication_group_present`**`(`**args, **kwargs*`)`

`salt.states.boto_elasticache.`**`subnet_group_absent`**`(`*name, tags=None, region=None, key=None, keyid=None, profile=None*`)`

`salt.states.boto_elasticache.`**`subnet_group_present`**`(`*name, subnet_ids=None, subnet_names=None, description=None, tags=None, region=None, key=None, keyid=None, profile=None*`)`

Ensure ElastiCache subnet group exists.

New in version 2015.8.0.

name The name for the ElastiCache subnet group. This value is stored as a lowercase string.

subnet_ids A list of VPC subnet IDs for the cache subnet group. Exclusive with subnet_names.

subnet_names A list of VPC subnet names for the cache subnet group. Exclusive with subnet_ids.

description Subnet group description.

tags A list of tags.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

21.19.36 salt.states.boto_elasticsearch_domain module

Manage Elasticsearch Domains

New in version 2016.11.0.

Create and destroy Elasticsearch domains. Be aware that this interacts with Amazon's services, and so may incur charges.

This module uses `boto3`, which can be installed via package, or pip.

This module accepts explicit vpc credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More information available here.

If IAM roles are not used you need to specify them either in a pillar file or in the minion's config file:

```
vpc.keyid: GKTADJGHEIQSXMKKRBJ08H
vpc.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

It's also possible to specify `key`, `keyid` and `region` via a profile, either passed in as a dict, or as a string to pull from pillars or minion config:

```
myprofile:
  keyid: GKTADJGHEIQSXMKKRBJ08H
  key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
  region: us-east-1
```

```
Ensure domain exists:
    boto_elasticsearch_domain.present:
        - DomainName: mydomain
        - profile='user-credentials'
        - ElasticsearchVersion: "2.3"
        - ElasticsearchClusterConfig:
            InstanceType": "t2.micro.elasticsearch"
            InstanceCount: 1
            DedicatedMasterEnabled: False
            ZoneAwarenessEnabled: False
        - EBSOptions:
            EBSEnabled: True
            VolumeType: "gp2"
            VolumeSize: 10
            Iops: 0
        - AccessPolicies:
            Version: "2012-10-17"
            Statement:
              - Effect: "Allow"
              - Principal:
                  AWS: "*"
              - Action:
                - "es:*"
```

(continues on next page)

(continued from previous page)

```
              - Resource: "arn:aws:es:*:111111111111:domain/mydomain/*"
              - Condition:
                  IpAddress:
                    "aws:SourceIp":
                      - "127.0.0.1"
                      - "127.0.0.2"
        - SnapshotOptions:
            AutomatedSnapshotStartHour: 0
        - AdvancedOptions:
            rest.action.multi.allow_explicit_index": "true"
        - Tags:
            a: "b"
        - region: us-east-1
        - keyid: GKTADJGHEIQSXMKKRBJ08H
        - key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

`salt.states.boto_elasticsearch_domain.absent`(*name*, *DomainName*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Ensure domain with passed properties is absent.

name The name of the state definition.

DomainName Name of the domain.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_elasticsearch_domain.present`(*name*, *DomainName*, *ElasticsearchClusterConfig=None*, *EBSOptions=None*, *AccessPolicies=None*, *SnapshotOptions=None*, *AdvancedOptions=None*, *Tags=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*, *ElasticsearchVersion='1.5'*)

Ensure domain exists.

name The name of the state definition

DomainName Name of the domain.

ElasticsearchClusterConfig Configuration options for an Elasticsearch domain. Specifies the instance type and number of instances in the domain cluster.

InstanceType (string) – The instance type for an Elasticsearch cluster.

InstanceCount (integer) – The number of instances in the specified domain cluster.

DedicatedMasterEnabled (boolean) – A boolean value to indicate whether a dedicated master node is enabled. See About Dedicated Master Nodes for more information.

ZoneAwarenessEnabled (boolean) – A boolean value to indicate whether zone awareness is enabled. See About Zone Awareness for more information.

DedicatedMasterType (string) – The instance type for a dedicated master node.

DedicatedMasterCount (integer) – Total number of dedicated master nodes, active and on standby, for the cluster.

EBSOptions Options to enable, disable and specify the type and size of EBS storage volumes.

EBSEnabled (boolean) – Specifies whether EBS-based storage is enabled.

VolumeType (string) – Specifies the volume type for EBS-based storage.

VolumeSize (integer) – Integer to specify the size of an EBS volume.

Iops (integer) – Specifies the IOPD for a Provisioned IOPS EBS volume (SSD).

AccessPolicies IAM access policy

SnapshotOptions Option to set time, in UTC format, of the daily automated snapshot. Default value is 0 hours.

AutomatedSnapshotStartHour (integer) – Specifies the time, in UTC format, when the service takes a daily automated snapshot of the specified Elasticsearch domain. Default value is 0 hours.

AdvancedOptions Option to allow references to indices in an HTTP request body. Must be false when configuring access to individual sub-resources. By default, the value is true .

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

ElasticsearchVersion String of format X.Y to specify version for the Elasticsearch domain eg. "1.5" or "2.3".

21.19.37 salt.states.boto_elb

Manage ELBs

New in version 2014.7.0.

Create and destroy ELBs. Be aware that this interacts with Amazon's services, and so may incur charges.

This module uses `boto`, which can be installed via package, or pip.

This module accepts explicit elb credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More information available here.

If IAM roles are not used you need to specify them either in a pillar file or in the minion's config file:

```
elb.keyid: GKTADJGHEIQSXMKKRBJ08H
elb.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

It's also possible to specify `key`, `keyid` and `region` via a profile, either passed in as a dict, or as a string to pull from pillars or minion config:

```
myprofile:
    keyid: GKTADJGHEIQSXMKKRBJ08H
    key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
    region: us-east-1
```

```
Ensure myelb ELB exists:
    boto_elb.present:
        - name: myelb
        - region: us-east-1
        - availability_zones:
            - us-east-1a
            - us-east-1c
            - us-east-1d
        - keyid: GKTADJGHEIQSXMKKRBJ08H
        - key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
        - listeners:
            - elb_port: 443
```

(continues on next page)

(continued from previous page)

```
              instance_port: 80
              elb_protocol: HTTPS
              instance_protocol: HTTP
              certificate: 'arn:aws:iam::1111111:server-certificate/mycert'
              policies:
                  - my-ssl-policy
                  - cookie-policy
            - elb_port: 8210
              instance_port: 8210
              elb_protocol: TCP
        - backends:
            - instance_port: 80
              policies:
                  - enable-proxy-protocol
        - health_check:
            target: 'HTTP:80/'
        - attributes:
            cross_zone_load_balancing:
              enabled: true
            access_log:
              enabled: true
              s3_bucket_name: 'mybucket'
              s3_bucket_prefix: 'my-logs'
              emit_interval: 5
            connecting_settings:
              idle_timeout: 60
        - cnames:
            - name: mycname.example.com.
              zone: example.com.
              ttl: 60
            - name: myothercname.example.com.
              zone: example.com.
        - security_groups:
            - my-security-group
        - policies:
            - policy_name: my-ssl-policy
              policy_type: SSLNegotiationPolicyType
              policy:
                Protocol-TLSv1.2: true
                Protocol-SSLv3: false
                Server-Defined-Cipher-Order: true
                ECDHE-ECDSA-AES128-GCM-SHA256: true
            - policy_name: cookie-policy
              policy_type: LBCookieStickinessPolicyType
              policy: {}  # no policy means this is a session cookie
            - policy_name: enable-proxy-protocol
              policy_type: ProxyProtocolPolicyType
              policy:
                ProxyProtocol: true

# Using a profile from pillars
Ensure myelb ELB exists:
    boto_elb.present:
        - name: myelb
        - region: us-east-1
        - profile: myelbprofile
```

(continues on next page)

(continued from previous page)

```
# Passing in a profile
Ensure myelb ELB exists:
    boto_elb.present:
        - name: myelb
        - region: us-east-1
        - profile:
            keyid: GKTADJGHEIQSXMKKRBJ08H
            key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

It's possible to specify attributes from pillars by specifying a pillar. You can override the values defined in the pillard by setting the attributes on the resource. The module will use the default pillar key 'boto_elb_attributes', which allows you to set default attributes for all ELB resources.

Setting the attributes pillar:

```
my_elb_attributes:
  cross_zone_load_balancing:
    enabled: true
  connection_draining:
    enabled: true
    timeout: 20
  access_log:
    enabled: true
    s3_bucket_name: 'mybucket'
    s3_bucket_prefix: 'my-logs'
    emit_interval: 5
```

Overriding the attribute values on the resource:

```
Ensure myelb ELB exists:
    boto_elb.present:
        - name: myelb
        - region: us-east-1
        - attributes_from_pillar: my_elb_attributes
        # override cross_zone_load_balancing:enabled
        - attributes:
            cross_zone_load_balancing:
              enabled: false
        - profile: myelbprofile
```

It's possible to specify cloudwatch alarms that will be setup along with the ELB. Note the alarm name will be defined by the name attribute provided, plus the ELB resource name.

```
Ensure myelb ELB exists:
    boto_elb.present:
        - name: myelb
        - region: us-east-1
        - profile: myelbprofile
        - alarms:
            UnHealthyHostCount:
              name: 'ELB UnHealthyHostCount **MANAGED BY SALT**'
              attributes:
                metric: UnHealthyHostCount
                namespace: AWS/ELB
                statistic: Average
                comparison: '>='
```

(continues on next page)

(continued from previous page)

```
                threshold: 1.0
                period: 600
                evaluation_periods: 6
                unit: null
                description: ELB UnHealthyHostCount
                alarm_actions: ['arn:aws:sns:us-east-1:12345:myalarm']
                insufficient_data_actions: []
                ok_actions: ['arn:aws:sns:us-east-1:12345:myalarm']
```

You can also use alarms from pillars, and override values from the pillar alarms by setting overrides on the resource. Note that 'boto_elb_alarms' will be used as a default value for all resources, if defined and can be used to ensure alarms are always set for a resource.

Setting the alarms in a pillar:

```
my_elb_alarm:
  UnHealthyHostCount:
    name: 'ELB UnHealthyHostCount **MANAGED BY SALT**'
    attributes:
      metric: UnHealthyHostCount
      namespace: AWS/ELB
      statistic: Average
      comparison: '>='
      threshold: 1.0
      period: 600
      evaluation_periods: 6
      unit: null
      description: ELB UnHealthyHostCount
      alarm_actions: ['arn:aws:sns:us-east-1:12345:myalarm']
      insufficient_data_actions: []
      ok_actions: ['arn:aws:sns:us-east-1:12345:myalarm']
```

Overriding the alarm values on the resource:

```
Ensure myelb ELB exists:
    boto_elb.present:
        - name: myelb
        - region: us-east-1
        - profile: myelbprofile
        - alarms_from_pillar: my_elb_alarm
        # override UnHealthyHostCount:attributes:threshold
        - alarms:
            UnHealthyHostCount:
              attributes:
                threshold: 2.0
```

Tags can also be set:

New in version 2016.3.0.

```
Ensure myelb ELB exists:
    boto_elb.present:
        - name: myelb
        - region: us-east-1
        - profile: myelbprofile
        - tags:
```

(continues on next page)

(continued from previous page)

```
            MyTag: 'My Tag Value'
            OtherTag: 'My Other Value'
```

`salt.states.boto_elb.`**`absent`**(*name, region=None, key=None, keyid=None, profile=None*)

Ensure an ELB does not exist

name name of the ELB

`salt.states.boto_elb.`**`present`**(*name, listeners, availability_zones=None, subnets=None, subnet_names=None, security_groups=None, scheme='internet-facing', health_check=None, attributes=None, attributes_from_pillar='boto_elb_attributes', cnames=None, alarms=None, alarms_from_pillar='boto_elb_alarms', policies=None, policies_from_pillar='boto_elb_policies', backends=None, region=None, key=None, keyid=None, profile=None, wait_for_sync=True, tags=None, instance_ids=None, instance_names=None*)

Ensure the ELB exists.

name Name of the ELB.

availability_zones A list of availability zones for this ELB.

listeners A list of listener lists; example:

```
[
    ['443', 'HTTPS', 'arn:aws:iam::1111111:server-certificate/mycert'],
    ['8443', '80', 'HTTPS', 'HTTP', 'arn:aws:iam::1111111:server-certificate/
↪mycert']
]
```

subnets A list of subnet IDs in your VPC to attach to your LoadBalancer.

subnet_names A list of subnet names in your VPC to attach to your LoadBalancer.

security_groups The security groups assigned to your LoadBalancer within your VPC. Must be passed either as a list or a comma-separated string.

For example, a list:

```
- security_groups:
  - secgroup-one
  - secgroup-two
```

Or as a comma-separated string:

```
- security_groups: secgroup-one,secgroup-two
```

scheme The type of a LoadBalancer, `internet-facing` or `internal`. Once set, can not be modified.

health_check A dict defining the health check for this ELB.

attributes A dict defining the attributes to set on this ELB. Unknown keys will be silently ignored.

See the *`salt.modules.boto_elb.set_attributes`* function for recognized attributes.

attributes_from_pillar name of pillar dict that contains attributes. Attributes defined for this specific state will override those from pillar.

cnames A list of cname dicts with attributes needed for the DNS add_record state. By default the boto_route53.add_record state will be used, which requires: name, zone, ttl, and identifier. See the boto_route53 state for information about these attributes. Other DNS modules can be called by specifying the provider keyword. the cnames dict will be passed to the state as kwargs.

See the *`salt.states.boto_route53`* state for information about these attributes.

alarms: a dictionary of name->boto_cloudwatch_alarm sections to be associated with this ELB. All attributes should be specified except for dimension which will be automatically set to this ELB.

See the *`salt.states.boto_cloudwatch_alarm`* state for information about these attributes.

alarms_from_pillar: name of pillar dict that contains alarm settings. Alarms defined for this specific state will override those from pillar.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

wait_for_sync Wait for an INSYNC change status from Route53.

tags dict of tags

instance_ids list of instance ids. The state will ensure that these, and ONLY these, instances are registered with the ELB. This is additive with instance_names.

instance_names list of instance names. The state will ensure that these, and ONLY these, instances are registered with the ELB. This is additive with instance_ids.

`salt.states.boto_elb.`**`register_instances`**(*name*, *instances*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Add EC2 instance(s) to an Elastic Load Balancer. Removing an instance from the `instances` list does not remove it from the ELB.

name The name of the Elastic Load Balancer to add EC2 instances to.

instances A list of EC2 instance IDs that this Elastic Load Balancer should distribute traffic to. This state will only ever append new instances to the ELB. EC2 instances already associated with this ELB will not be removed if they are not in the `instances` list.

New in version 2015.8.0.

```
add-instances:
  boto_elb.register_instances:
    - name: myloadbalancer
    - instances:
      - instance-id1
      - instance-id2
```

21.19.38 salt.states.boto_elbv2 module

Manage AWS Application Load Balancer

New in version 2017.7.0.

Add and remove targets from an ALB target group.

This module uses `boto3`, which can be installed via package, or pip.

This module accepts explicit alb credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More information available here.

If IAM roles are not used you need to specify them either in a pillar file or in the minion's config file:

```
elbv2.keyid: GKTADJGHEIQSXMKKRBJ08H
elbv2.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
elbv2.region: us-west-2
```

It's also possible to specify `key`, `keyid` and `region` via a profile, either passed in as a dict, or as a string to pull from pillars or minion config:

```yaml
myprofile:
    keyid: GKTADJGHEIQSXMKKRBJ08H
    key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
    region: us-east-1
```

`salt.states.boto_elbv2.`**`create_target_group`**(*name*, *protocol*, *port*, *vpc_id*, *region=None*, *key=None*, *keyid=None*, *profile=None*, *health_check_protocol='HTTP'*, *health_check_port='traffic-port'*, *health_check_path='/'*, *health_check_interval_seconds=30*, *health_check_timeout_seconds=5*, *healthy_threshold_count=5*, *unhealthy_threshold_count=2*, ***kwargs*)

New in version 2017.11.0.

Create target group if not present.
name (string) - The name of the target group.
protocol (string) - The protocol to use for routing traffic to the targets
port (int) - The port on which the targets receive traffic. This port is used unless you specify a port override when registering the traffic.
vpc_id (string) - The identifier of the virtual private cloud (VPC).
health_check_protocol (string) - The protocol the load balancer uses when performing health check on targets. The default is the HTTP protocol.
health_check_port (string) - The port the load balancer uses when performing health checks on targets. The default is 'traffic-port', which indicates the port on which each target receives traffic from the load balancer.
health_check_path (string) - The ping path that is the destination on the targets for health checks. The default is /.
health_check_interval_seconds (integer) - The approximate amount of time, in seconds, between health checks of an individual target. The default is 30 seconds.
health_check_timeout_seconds (integer) - The amount of time, in seconds, during which no response from a target means a failed health check. The default is 5 seconds.
healthy_threshold_count (integer) - The number of consecutive health checks successes required before considering an unhealthy target healthy. The default is 5.
unhealthy_threshold_count (integer) - The number of consecutive health check failures required before considering a target unhealthy. The default is 2.
returns (bool) - True on success, False on failure.
CLI example: .. code-block:: yaml

create-target:

  boto_elb2.create_targets_group:

- name: myALB
- protocol: https
- port: 443
- vpc_id: myVPC

`salt.states.boto_elbv2.`**`delete_target_group`**(*name*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Delete target group.
name (string) - The Amazon Resource Name (ARN) of the resource.
returns (bool) - True on success, False on failure.
CLI example:

```
check-target:
  boto_elb2.delete_targets_group:
```

(continues on next page)

(continued from previous page)

```
    - name: myALB
    - protocol: https
    - port: 443
    - vpc_id: myVPC
```

salt.states.boto_elbv2.**targets_deregistered**(*name*, *targets*, *region=None*, *key=None*, *keyid=None*, *profile=None*, ***kwargs*)

Remove targets to an Application Load Balancer target group.

name The ARN of the Application Load Balancer Target Group to remove targets from.

targets A list of target IDs or a string of a single target registered to the target group to be removed

New in version Unknown.

```
remove-targets:
  boto_elb.targets_deregistered:
    - name: arn:myloadbalancer
    - targets:
      - instance-id1
      - instance-id2
```

salt.states.boto_elbv2.**targets_registered**(*name*, *targets*, *region=None*, *key=None*, *keyid=None*, *profile=None*, ***kwargs*)

New in version 2017.7.0.

Add targets to an Application Load Balancer target group. This state will not remove targets.

name The ARN of the Application Load Balancer Target Group to add targets to.

targets A list of target IDs or a string of a single target that this target group should distribute traffic to.

```
add-targets:
  boto_elb.targets_registered:
    - name: arn:myloadbalancer
    - targets:
      - instance-id1
      - instance-id2
```

21.19.39 salt.states.boto_iam

Manage IAM objects

New in version 2015.8.0.

This module uses `boto`, which can be installed via package, or pip.

This module accepts explicit IAM credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More information available here.

It's also possible to specify `key`, `keyid` and `region` via a profile, either passed in as a dict, or as a string to pull from pillars or minion config:

```
delete-user:
  boto_iam.user_absent:
    - name: myuser
    - delete_keys: true
```

```
delete-keys:
  boto_iam.keys_absent:
    - access_keys:
      - 'AKIAJHTMIQ2ASDFLASDF'
      - 'PQIAJHTMIQ2ASRTLASFR'
    - user_name: myuser
```

```
create-user:
  boto_iam.user_present:
    - name: myuser
    - policies:
        mypolicy: |
            {
                "Version": "2012-10-17",
                "Statement": [{
                "Effect": "Allow",
                "Action": "*",
                "Resource": "*"}]
            }
    - password: NewPassword$$1
    - region: eu-west-1
    - keyid: 'AKIAJHTMIQ2ASDFLASDF'
    - key: 'fdkjsafkljsASSADFalkfjasdf'
```

```
create-group:
  boto_iam.group_present:
    - name: mygroup
    - users:
      - myuser
      - myuser1
    - policies:
        mypolicy: |
            {
                "Version": "2012-10-17",
                "Statement": [{
                "Effect": "Allow",
                "Action": "*",
                "Resource": "*"}]
            }
    - region: eu-west-1
    - keyid: 'AKIAJHTMIQ2ASDFLASDF'
    - key: 'safsdfsal;fdkjsafkljsASSADFalkfj'
```

```
change-policy:
  boto_iam.account_policy:
    - change_password: True
    - region: eu-west-1
    - keyid: 'AKIAJHTMIQ2ASDFLASDF'
    - key: 'safsdfsal;fdkjsafkljsASSADFalkfj'
```

```
create server certificate:
  boto_iam.server_cert_present:
    - name: mycert
    - public_key: salt://base/mycert.crt
    - private_key: salt://base/mycert.key
    - cert_chain: salt://base/mycert_chain.crt
```

(continues on next page)

(continued from previous page)

```
    - region: eu-west-1
    - keyid: 'AKIAJHTMIQ2ASDFLASDF'
    - key: 'fdkjsafkljsASSADFalkfjasdf'
```

```
delete server certificate:
  boto_iam.server_cert_absent:
    - name: mycert
```

```
create keys for user:
  boto_iam.keys_present:
    - name: myusername
    - number: 2
    - save_dir: /root
    - region: eu-west-1
    - keyid: 'AKIAJHTMIQ2ASDFLASDF'
    - key: 'fdkjsafkljsASSADFalkfjasdf'
```

```
create policy:
  boto_iam.policy_present:
    - name: myname
    - policy_document: '{"MyPolicy": "Statement": [{"Action": ["sqs:*"], "Effect":
↪"Allow", "Resource": ["arn:aws:sqs:*:*:*"], "Sid": "MyPolicySqs1"}]}'
    - region: eu-west-1
    - keyid: 'AKIAJHTMIQ2ASDFLASDF'
    - key: 'fdkjsafkljsASSADFalkfjasdf'
```

```
add-saml-provider:
  boto_iam.saml_provider_present:
    - name: my_saml_provider
    - saml_metadata_document: salt://base/files/provider.xml
    - keyid: 'AKIAJHTMIQ2ASDFLASDF'
    - key: 'safsdfsal;fdkjsafkljsASSADFalkfj'
```

`salt.states.boto_iam.`**`account_policy`**`(`*name=None, allow_users_to_change_password=None, hard_expiry=None, max_password_age=None, minimum_password_length=None, password_reuse_prevention=None, require_lowercase_characters=None, require_numbers=None, require_symbols=None, require_uppercase_characters=None, region=None, key=None, keyid=None, profile=None*`)`

Change account policy.

New in version 2015.8.0.

name (string) The name of the account policy

allow_users_to_change_password (bool) Allows all IAM users in your account to use the AWS Management Console to change their own passwords.

hard_expiry (bool) Prevents IAM users from setting a new password after their password has expired.

max_password_age (int) The number of days that an IAM user password is valid.

minimum_password_length (int) The minimum number of characters allowed in an IAM user password.

password_reuse_prevention (int) Specifies the number of previous passwords that IAM users are prevented from reusing.

require_lowercase_characters (bool) Specifies whether IAM user passwords must contain at least one lowercase character from the ISO basic Latin alphabet (a to z).

require_numbers (bool) Specifies whether IAM user passwords must contain at least one numeric character (0 to 9).

require_symbols (bool) Specifies whether IAM user passwords must contain at least one of the following non-alphanumeric characters: ! @ # $ % ^ & * ( ) _ + - = [ ] { } | '

require_uppercase_characters (bool) Specifies whether IAM user passwords must contain at least one uppercase character from the ISO basic Latin alphabet (A to Z).

region (string) Region to connect to.

key (string) Secret key to be used.

keyid (string) Access key to be used.

profile (dict) A dict with region, key and keyid, or a pillar key (string)

`salt.states.boto_iam.`**`group_absent`**`(`*name*, *region=None*, *key=None*, *keyid=None*, *profile=None*`)`

New in version 2015.8.0.

Ensure the IAM group is absent.

name (string) The name of the group.

region (string) Region to connect to.

key (string) Secret key to be used.

keyid (string) Access key to be used.

profile (dict) A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_iam.`**`group_present`**`(`*name*, *policies=None*, *policies_from_pillars=None*, *managed_policies=None*, *users=None*, *path='/'*, *region=None*, *key=None*, *keyid=None*, *profile=None*, *delete_policies=True*`)`

New in version 2015.8.0.

Ensure the IAM group is present

name (string) The name of the new group.

path (string) The path for the group, defaults to '/'

policies (dict) A dict of IAM group policy documents.

policies_from_pillars (list) A list of pillars that contain role policy dicts. Policies in the pillars will be merged in the order defined in the list and key conflicts will be handled by later defined keys overriding earlier defined keys. The policies defined here will be merged with the policies defined in the policies argument. If keys conflict, the keys in the policies argument will override the keys defined in policies_from_pillars.

managed_policies (list) A list of policy names or ARNs that should be attached to this group.

users (list) A list of users to be added to the group.

region (string) Region to connect to.

key (string) Secret key to be used.

keyid (string) Access key to be used.

profile (dict) A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

delete_policies (boolean) Delete or detach existing policies that are not in the given list of policies. Default value is `True`. If `False` is specified, existing policies will not be deleted or detached allowing manual modifications on the IAM group to be persistent.

`salt.states.boto_iam.`**`keys_absent`**`(`*access_keys*, *user_name*, *region=None*, *key=None*, *keyid=None*, *profile=None*`)`

New in version 2015.8.0.

Ensure the IAM user access_key_id is absent.

access_key_id (list) A list of access key ids

user_name (string) The username of the user

region (string) Region to connect to.

key (string) Secret key to be used.

keyid (string) Access key to be used.

profile (dict) A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_iam.`**`keys_present`**`(`*name, number, save_dir, region=None, key=None, keyid=None, profile=None, save_format='{2}\n{0}\n{3}\n{1}\n'*`)`

New in version 2015.8.0.

Ensure the IAM access keys are present.

name (string) The name of the new user.

number (int) Number of keys that user should have.

save_dir (string) The directory that the key/keys will be saved. Keys are saved to a file named according to the username privided.

region (string) Region to connect to.

key (string) Secret key to be used.

keyid (string) Access key to be used.

profile (dict) A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

save_format (dict) Save format is repeated for each key. Default format is "{2}n{0}n{3}n{1}n", where {0} and {1} are placeholders for new key_id and key respectively, whereas {2} and {3} are "key_id-{number}" and 'key-{number}' strings kept for compatibility.

`salt.states.boto_iam.`**`policy_absent`**`(`*name, region=None, key=None, keyid=None, profile=None*`)`

New in version 2015.8.0.

Ensure the IAM managed policy with the specified name is absent

name (string) The name of the new policy.

region (string) Region to connect to.

key (string) Secret key to be used.

keyid (string) Access key to be used.

profile (dict) A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_iam.`**`policy_present`**`(`*name, policy_document, path=None, description=None, region=None, key=None, keyid=None, profile=None*`)`

New in version 2015.8.0.

Ensure the IAM managed policy is present

name (string) The name of the new policy.

policy_document (dict) The document of the new policy

path (string) The path in which the policy will be created. Default is '/'.

description (string) Description

region (string) Region to connect to.

key (string) Secret key to be used.

keyid (string) Access key to be used.

profile (dict) A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_iam.`**`saml_provider_absent`**`(`*name, region=None, key=None, keyid=None, profile=None*`)`

New in version 2016.11.0.

Ensure the SAML provider with the specified name is absent.

name (string) The name of the SAML provider.

saml_metadata_document (string) The xml document of the SAML provider.

region (string) Region to connect to.

key (string) Secret key to be used.

keyid (string) Access key to be used.

profile (dict) A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_iam.`**`saml_provider_present`**(*name*, *saml_metadata_document*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

New in version 2016.11.0.

Ensure the SAML provider with the specified name is present.

name (string) The name of the SAML provider.

saml_metadata_document (string) The xml document of the SAML provider.

region (string) Region to connect to.

key (string) Secret key to be used.

keyid (string) Access key to be used.

profile (dict) A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_iam.`**`server_cert_absent`**(*name*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Deletes a server certificate.

New in version 2015.8.0.

name (string) The name for the server certificate. Do not include the path in this value.

region (string) The name of the region to connect to.

key (string) The key to be used in order to connect

keyid (string) The keyid to be used in order to connect

profile (string) The profile that contains a dict of region, key, keyid

`salt.states.boto_iam.`**`server_cert_present`**(*name*, *public_key*, *private_key*, *cert_chain=None*, *path=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Crete server certificate.

New in version 2015.8.0.

name (string) The name for the server certificate. Do not include the path in this value.

public_key (string) The contents of the public key certificate in PEM-encoded format.

private_key (string) The contents of the private key in PEM-encoded format.

cert_chain (string) The contents of the certificate chain. This is typically a concatenation of the PEM-encoded public key certificates of the chain.

path (string) The path for the server certificate.

region (string) The name of the region to connect to.

key (string) The key to be used in order to connect

keyid (string) The keyid to be used in order to connect

profile (string) The profile that contains a dict of region, key, keyid

`salt.states.boto_iam.`**`user_absent`**(*name*, *delete_keys=True*, *delete_mfa_devices=True*, *delete_profile=True*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

New in version 2015.8.0.

Ensure the IAM user is absent. User cannot be deleted if it has keys.

name (string) The name of the new user.

delete_keys (bool) Delete all keys from user.

delete_mfa_devices (bool) Delete all mfa devices from user.

New in version 2016.3.0.

delete_profile (bool) Delete profile from user.

New in version 2016.3.0.

region (string) Region to connect to.

key (string) Secret key to be used.

keyid (string) Access key to be used.

profile (dict) A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_iam.`**`user_present`**`(`*name, policies=None, policies_from_pillars=None, managed_policies=None, password=None, path=None, region=None, key=None, keyid=None, profile=None*`)`

New in version 2015.8.0.

Ensure the IAM user is present

name (string) The name of the new user.

policies (dict) A dict of IAM group policy documents.

policies_from_pillars (list) A list of pillars that contain role policy dicts. Policies in the pillars will be merged in the order defined in the list and key conflicts will be handled by later defined keys overriding earlier defined keys. The policies defined here will be merged with the policies defined in the policies argument. If keys conflict, the keys in the policies argument will override the keys defined in policies_from_pillars.

managed_policies (list) A list of managed policy names or ARNs that should be attached to this user.

password (string) The password for the new user. Must comply with account policy.

path (string) The path of the user. Default is '/'.

New in version 2015.8.2.

region (string) Region to connect to.

key (string) Secret key to be used.

keyid (string) Access key to be used.

profile (dict) A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

21.19.40 salt.states.boto_iam_role

Manage IAM roles

New in version 2014.7.0.

This module uses `boto`, which can be installed via package, or pip.

This module accepts explicit IAM credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More information available here.

If IAM roles are not used you need to specify them either in a pillar file or in the minion's config file:

```
iam.keyid: GKTADJGHEIQSXMKKRBJ08H
iam.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

It's also possible to specify `key`, `keyid` and `region` via a profile, either passed in as a dict, or as a string to pull from pillars or minion config:

```
myprofile:
    keyid: GKTADJGHEIQSXMKKRBJ08H
    key: askjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
    region: us-east-1
```

Creating a role will automatically create an instance profile and associate it with the role. This is the default behavior of the AWS console.

```yaml
myrole:
    boto_iam_role.present:
        - region: us-east-1
        - key: GKTADJGHEIQSXMKKRBJ08H
        - keyid: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
        - policies_from_pillars:
            - shared_iam_bootstrap_policy
        - policies:
            MySQSPolicy:
                Statement:
                  - Action:
                        - sqs:*
                    Effect: Allow
                    Resource:
                        - arn:aws:sqs:*:*:*
                    Sid: MyPolicySQS1
            MyS3Policy:
                Statement:
                  - Action:
                        - s3:GetObject
                    Effect: Allow
                    Resource:
                        - arn:aws:s3:*:*:mybucket/*

# Using a credentials profile from pillars
myrole:
    boto_iam_role.present:
        - profile: myiamprofile

# Passing in a credentials profile
myrole:
    boto_iam_role.present:
        - profile:
            key: GKTADJGHEIQSXMKKRBJ08H
            keyid: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
            region: us-east-1
```

If `delete_policies: False` is specified, existing policies that are not in the given list of policies will not be deleted. This allows manual modifications on the IAM role to be persistent. This functionality was added in 2015.8.0.

Note: When using the `profile` parameter and `region` is set outside of the profile group, region is ignored and a default region will be used.

If `region` is missing from the `profile` data set, `us-east-1` will be used as the default region.

`salt.states.boto_iam_role.`**`absent`**`(`*name*, *region=None*, *key=None*, *keyid=None*, *profile=None*`)`

Ensure the IAM role is deleted.

name Name of the IAM role.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_iam_role.`**`present`**(*name*, *policy_document=None*, *policy_document_from_pillars=None*, *path=None*, *policies=None*, *policies_from_pillars=None*, *managed_policies=None*, *create_instance_profile=True*, *region=None*, *key=None*, *keyid=None*, *profile=None*, *delete_policies=True*)

Ensure the IAM role exists.

name Name of the IAM role.

policy_document The policy that grants an entity permission to assume the role. (See https://boto.readthedocs.io/en/latest/ref/iam.html#boto.iam.connection.IAMConnection.create_role)

policy_document_from_pillars A pillar key that contains a role policy document. The statements defined here will be appended with the policy document statements defined in the policy_document argument.

New in version 2017.7.0.

path The path to the role/instance profile. (See https://boto.readthedocs.io/en/latest/ref/iam.html#boto.iam.connection.IAMConnection.create_role)

policies A dict of IAM role policies.

policies_from_pillars A list of pillars that contain role policy dicts. Policies in the pillars will be merged in the order defined in the list and key conflicts will be handled by later defined keys overriding earlier defined keys. The policies defined here will be merged with the policies defined in the policies argument. If keys conflict, the keys in the policies argument will override the keys defined in policies_from_pillars.

managed_policies A list of (AWS or Customer) managed policies to be attached to the role.

create_instance_profile A boolean of whether or not to create an instance profile and associate it with this role.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

delete_policies Deletes existing policies that are not in the given list of policies. Default value is `True`. If `False` is specified, existing policies will not be deleted allowing manual modifications on the IAM role to be persistent.

New in version 2015.8.0.

21.19.41 salt.states.boto_iot module

Manage IoT Objects

New in version 2016.3.0.

Create and destroy IoT objects. Be aware that this interacts with Amazon's services, and so may incur charges.

depends

- boto
- boto3

The dependencies listed above can be installed via package or pip.

This module accepts explicit vpc credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More information available here.

If IAM roles are not used you need to specify them either in a pillar file or in the minion's config file:

```
vpc.keyid: GKTADJGHEIQSXMKKRBJ08H
vpc.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

It's also possible to specify `key`, `keyid` and `region` via a profile, either passed in as a dict, or as a string to pull from pillars or minion config:

```
myprofile:
    keyid: GKTADJGHEIQSXMKKRBJ08H
    key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
        region: us-east-1
```

```
Ensure policy exists:
    boto_iot.policy_present:
        - policyName: mypolicy
        - policyDocument:
            Version: "2012-10-17"
            Statement:
              Action:
                - iot:Publish
              Resource:
                - "*"
              Effect: "Allow"
        - region: us-east-1
        - keyid: GKTADJGHEIQSXMKKRBJ08H
        - key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs

Ensure topic rule exists:
    boto_iot.topic_rule_present:
        - ruleName: myrule
        - sql: "SELECT * FROM 'iot/test'"
        - description: 'test rule'
        - ruleDisabled: false
        - actions:
          - lambda:
              functionArn: "arn:aws:us-east-1:1234:function/functionname"
        - region: us-east-1
        - keyid: GKTADJGHEIQSXMKKRBJ08H
        - key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

`salt.states.boto_iot.`**`policy_absent`**(*name*, *policyName*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Ensure policy with passed properties is absent.

name The name of the state definition.

policyName Name of the policy.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_iot.`**`policy_attached`**(*name*, *policyName*, *principal*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Ensure policy is attached to the given principal.

name The name of the state definition

policyName Name of the policy.

principal The principal which can be a certificate ARN or a Cognito ID.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_iot.`**`policy_detached`**`(`*name, policyName, principal, region=None, key=None, keyid=None, profile=None*`)`

Ensure policy is attached to the given principal.

name The name of the state definition.

policyName Name of the policy.

principal The principal which can be a certificate ARN or a Cognito ID.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_iot.`**`policy_present`**`(`*name, policyName, policyDocument, region=None, key=None, keyid=None, profile=None*`)`

Ensure policy exists.

name The name of the state definition

policyName Name of the policy.

policyDocument The JSON document that describes the policy. The length of the policyDocument must be a minimum length of 1, with a maximum length of 2048, excluding whitespace.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_iot.`**`thing_type_absent`**`(`*name, thingTypeName, region=None, key=None, keyid=None, profile=None*`)`

Ensure thing type with passed properties is absent.

New in version 2016.11.0.

name The name of the state definition.

thingTypeName Name of the thing type.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_iot.`**`thing_type_present`**`(`*name, thingTypeName, thingTypeDescription, searchableAttributesList, region=None, key=None, keyid=None, profile=None*`)`

Ensure thing type exists.

New in version 2016.11.0.

name The name of the state definition

thingTypeName Name of the thing type

thingTypeDescription Description of the thing type

searchableAttributesList List of string attributes that are searchable for the thing type

region Region to connect to.

key Secret key to be used.

keyid Access key to be used

profile A dict with region, key, keyid, or a pillar key (string) that contains a dict with region, key, and keyid `salt.states.boto_iot.`**`topic_rule_absent`**`(`*name, ruleName, region=None, key=None, keyid=None, profile=None*`)`

Ensure topic rule with passed properties is absent.

name The name of the state definition.

ruleName Name of the policy.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_iot.`**`topic_rule_present`**`(`*name, ruleName, sql, actions, description=", ruleDisabled=False, region=None, key=None, keyid=None, profile=None*`)`

Ensure topic rule exists.

name The name of the state definition

ruleName Name of the rule.

sql The SQL statement used to query the topic.

actions The actions associated with the rule.

description The description of the rule.

ruleDisable Specifies whether the rule is disabled.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

21.19.42 salt.states.boto_kinesis module

Manage Kinesis Streams

New in version 2017.7.0.

Create and destroy Kinesis streams. Be aware that this interacts with Amazon's services, and so may incur charges.

This module uses `boto3`, which can be installed via package, or pip.

This module accepts explicit Kinesis credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More information available here.

If IAM roles are not used you need to specify them either in a pillar file or in the minion's config file:

```
keyid: GKTADJGHEIQSXMKKRBJ08H
key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
region: us-east-1
```

It's also possible to specify `key`, `keyid` and `region` via a profile, either passed in as a dict, or as a string to pull from pillars or minion config:

```
myprofile:
    keyid: GKTADJGHEIQSXMKKRBJ08H
    key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
        region: us-east-1
```

```
Ensure Kinesis stream does not exist:
  boto_kinesis.absent:
    - name: new_stream
    - keyid: GKTADJGHEIQSXMKKRBJ08H
    - key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
    - region: us-east-1

Ensure Kinesis stream exists:
  boto_kinesis.present:
    - name: new_stream
    - retention_hours: 168
    - enhanced_monitoring: ['ALL']
    - num_shards: 2
    - keyid: GKTADJGHEIQSXMKKRBJ08H
    - key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
    - region: us-east-1
```

`salt.states.boto_kinesis.`**`absent`**(*name, region=None, key=None, keyid=None, profile=None*)

Delete the kinesis stream, if it exists.

name (string) Stream name

region (string) Region to connect to.

key (string) Secret key to be used.

keyid (string) Access key to be used.

profile (dict) A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_kinesis.`**`present`**(*name, retention_hours=None, enhanced_monitoring=None, num_shards=None, do_reshard=True, region=None, key=None, keyid=None, profile=None*)

Ensure the kinesis stream is properly configured and scaled.

name (string) Stream name

retention_hours (int) Retain data for this many hours. AWS allows minimum 24 hours, maximum 168 hours.

enhanced_monitoring (list of string) Turn on enhanced monitoring for the specified shard-level metrics. Pass in ['ALL'] or True for all metrics, [] or False for no metrics. Turn on individual metrics by passing in a list: ['IncomingBytes', 'OutgoingBytes'] Note that if only some metrics are supplied, the remaining metrics will be turned off.

num_shards (int) Reshard stream (if necessary) to this number of shards !!!!! Resharding is expensive! Each split or merge can take up to 30 seconds, and the reshard method balances the partition space evenly. Resharding from N to N+1 can require 2N operations. Resharding is much faster with powers of 2 (e.g. 2^N to 2^N+1) !!!!!

do_reshard (boolean) If set to False, this script will NEVER reshard the stream, regardless of other input. Useful for testing.

region (string) Region to connect to.

key (string) Secret key to be used.

keyid (string) Access key to be used.

profile (dict) A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

21.19.43 salt.states.boto_kms

Manage KMS keys, key policies and grants.

New in version 2015.8.0.

Be aware that this interacts with Amazon's services, and so may incur charges.

This module uses `boto`, which can be installed via package, or pip.

This module accepts explicit kms credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More information available here.

If IAM roles are not used you need to specify them either in a pillar file or in the minion's config file:

```
elb.keyid: GKTADJGHEIQSXMKKRBJ08H
elb.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

It's also possible to specify `key`, `keyid` and `region` via a profile, either passed in as a dict, or as a string to pull from pillars or minion config:

```
myprofile:
    keyid: GKTADJGHEIQSXMKKRBJ08H
    key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
    region: us-east-1
```

```
Ensure mykey key exists:
  boto_kms.key_present:
    - name: mykey
    - region: us-east-1

# Using a profile from pillars
Ensure mykey key exists:
  boto_kms.key_present:
    - name: mykey
    - region: us-east-1
    - profile: myprofile

# Passing in a profile
Ensure mykey key exists:
  boto_key.key_present:
    - name: mykey
    - region: us-east-1
    - profile:
        keyid: GKTADJGHEIQSXMKKRBJ08H
        key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

`salt.states.boto_kms.`**`key_present`**(*name, policy, description=None, key_usage=None, grants=None, manage_grants=False, key_rotation=False, enabled=True, region=None, key=None, keyid=None, profile=None*)

Ensure the KMS key exists. KMS keys can not be deleted, so this function must be used to ensure the key is enabled or disabled.

name Name of the key.

policy Key usage policy.

description Description of the key.

key_usage Specifies the intended use of the key. Can only be set on creation, defaults to ENCRYPT_DECRYPT, which is also the only supported option.

grants A list of grants to apply to the key. Not currently implemented.

manage_grants Whether or not to manage grants. False by default, which will not manage any grants.

key_rotation Whether or not key rotation is enabled for the key. False by default.

enabled Whether or not the key is enabled. True by default.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

21.19.44 salt.states.boto_lambda module

Manage Lambda Functions

New in version 2016.3.0.

Create and destroy Lambda Functions. Be aware that this interacts with Amazon's services, and so may incur charges.

depends

- boto
- boto3

The dependencies listed above can be installed via package or pip.

This module accepts explicit vpc credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More information available here.

If IAM roles are not used you need to specify them either in a pillar file or in the minion's config file:

```
vpc.keyid: GKTADJGHEIQSXMKKRBJ08H
vpc.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

It's also possible to specify `key`, `keyid` and `region` via a profile, either passed in as a dict, or as a string to pull from pillars or minion config:

```
myprofile:
    keyid: GKTADJGHEIQSXMKKRBJ08H
    key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
        region: us-east-1
```

```
Ensure function exists:
    boto_lambda.function_present:
        - FunctionName: myfunction
        - Runtime: python2.7
        - Role: iam_role_name
        - Handler: entry_function
        - ZipFile: code.zip
        - S3Bucket: bucketname
        - S3Key: keyname
        - S3ObjectVersion: version
        - Description: "My Lambda Function"
        - Timeout: 3
        - MemorySize: 128
        - region: us-east-1
        - keyid: GKTADJGHEIQSXMKKRBJ08H
        - key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

`salt.states.boto_lambda.`**`alias_absent`**(*name, FunctionName, Name, region=None, key=None, keyid=None, profile=None*)

Ensure alias with passed properties is absent.

name The name of the state definition.

FunctionName Name of the function.

Name Name of the alias.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_lambda.`**`alias_present`**(*name*, *FunctionName*, *Name*, *FunctionVersion*, *Description=''*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Ensure alias exists.

name The name of the state definition.

FunctionName Name of the function for which you want to create an alias.

Name The name of the alias to be created.

FunctionVersion Function version for which you are creating the alias.

Description A short, user-defined function description. Lambda does not use this value. Assign a meaningful description as you see fit.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_lambda.`**`event_source_mapping_absent`**(*name*, *EventSourceArn*, *FunctionName*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Ensure event source mapping with passed properties is absent.

name The name of the state definition.

EventSourceArn ARN of the event source.

FunctionName Name of the lambda function.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_lambda.`**`event_source_mapping_present`**(*name*, *EventSourceArn*, *FunctionName*, *StartingPosition*, *Enabled=True*, *BatchSize=100*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Ensure event source mapping exists.

name The name of the state definition.

EventSourceArn The Amazon Resource Name (ARN) of the Amazon Kinesis or the Amazon DynamoDB stream that is the event source.

FunctionName The Lambda function to invoke when AWS Lambda detects an event on the stream.

You can specify an unqualified function name (for example, "Thumbnail") or you can specify Amazon Resource Name (ARN) of the function (for example, "arn:aws:lambda:us-west-2:account-id:function:ThumbNail"). AWS Lambda also allows you to specify only the account ID qualifier (for example, "account-id:Thumbnail"). Note that the length constraint applies only to the ARN. If you specify only the function name, it is limited to 64 character in length.

StartingPosition The position in the stream where AWS Lambda should start reading. (TRIM_HORIZON | LATEST)

Enabled Indicates whether AWS Lambda should begin polling the event source. By default, Enabled is true.

BatchSize The largest number of records that AWS Lambda will retrieve from your event source at the time of invoking your function. Your function receives an event with all the retrieved records. The default is 100 records.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_lambda.`**`function_absent`**`(`*name, FunctionName, region=None, key=None, keyid=None, profile=None*`)`

Ensure function with passed properties is absent.

name The name of the state definition.

FunctionName Name of the function.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_lambda.`**`function_present`**`(`*name, FunctionName, Runtime, Role, Handler, ZipFile=None, S3Bucket=None, S3Key=None, S3ObjectVersion=None, Description=", Timeout=3, MemorySize=128, Permissions=None, RoleRetries=5, region=None, key=None, keyid=None, profile=None, VpcConfig=None, Environment=None*`)`

Ensure function exists.

name The name of the state definition

FunctionName Name of the Function.

Runtime The Runtime environment for the function. One of 'nodejs', 'java8', or 'python2.7'

Role The name or ARN of the IAM role that the function assumes when it executes your function to access any other AWS resources.

Handler The function within your code that Lambda calls to begin execution. For Node.js it is the module-name.*export* value in your function. For Java, it can be package.classname::handler or package.classname.

ZipFile A path to a .zip file containing your deployment package. If this is specified, S3Bucket and S3Key must not be specified.

S3Bucket Amazon S3 bucket name where the .zip file containing your package is stored. If this is specified, S3Key must be specified and ZipFile must NOT be specified.

S3Key The Amazon S3 object (the deployment package) key name you want to upload. If this is specified, S3Key must be specified and ZipFile must NOT be specified.

S3ObjectVersion The version of S3 object to use. Optional, should only be specified if S3Bucket and S3Key are specified.

Description A short, user-defined function description. Lambda does not use this value. Assign a meaningful description as you see fit.

Timeout The function execution time at which Lambda should terminate this function. Because the execution time has cost implications, we recommend you set this value based on your expected execution time. The default is 3 seconds.

MemorySize The amount of memory, in MB, your function is given. Lambda uses this memory size to infer the amount of CPU and memory allocated to your function. Your function use-case determines your CPU and memory requirements. For example, a database operation might need less memory compared to an image processing function. The default value is 128 MB. The value must be a multiple of 64 MB.

VpcConfig If your Lambda function accesses resources in a VPC, you must provide this parameter identifying the list of security group IDs/Names and subnet IDs/Name. These must all belong to the same VPC. This is a dict of the form:

```
VpcConfig:
  SecurityGroupNames:
    - mysecgroup1
    - mysecgroup2
  SecurityGroupIds:
    - sg-abcdef1234
  SubnetNames:
    - mysubnet1
  SubnetIds:
    - subnet-1234abcd
    - subnet-abcd1234
```

If VpcConfig is provided at all, you MUST pass at least one security group and one subnet.

Permissions A list of permission definitions to be added to the function's policy

RoleRetries IAM Roles may take some time to propagate to all regions once created. During that time function creation may fail; this state will atuomatically retry this number of times. The default is 5.

Environment The parent object that contains your environment's configuration settings. This is a dictionary of the form:

```
{
    'Variables': {
        'VariableName': 'VariableValue'
    }
}
```

New in version 2017.7.0.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

21.19.45 salt.states.boto_lc

Manage Launch Configurations

New in version 2014.7.0.

Create and destroy Launch Configurations. Be aware that this interacts with Amazon's services, and so may incur charges.

A limitation of this module is that you can not modify launch configurations once they have been created. If a launch configuration with the specified name exists, this module will always report success, even if the specified configuration doesn't match. This is due to a limitation in Amazon's launch configuration API, as it only allows launch configurations to be created and deleted.

Also note that a launch configuration that's in use by an autoscale group can not be deleted until the autoscale group is no longer using it. This may affect the way in which you want to order your states.

This module uses `boto`, which can be installed via package, or pip.

This module accepts explicit autoscale credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More information available here.

If IAM roles are not used you need to specify them either in a pillar file or in the minion's config file:

```
asg.keyid: GKTADJGHEIQSXMKKRBJ08H
asg.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

It's also possible to specify `key`, `keyid` and `region` via a profile, either passed in as a dict, or as a string to pull from pillars or minion config:

```
myprofile:
    keyid: GKTADJGHEIQSXMKKRBJ08H
    key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
    region: us-east-1
```

Credential information is shared with autoscale groups as launch configurations and autoscale groups are completely dependent on each other.

```
Ensure mylc exists:
  boto_lc.present:
    - name: mylc
    - image_id: ami-0b9c9f62
    - key_name: mykey
    - security_groups:
        - mygroup
    - instance_type: m1.small
    - instance_monitoring: true
    - block_device_mappings:
        - '/dev/sda1':
            size: 20
            volume_type: 'io1'
            iops: 220
            delete_on_termination: true
    - cloud_init:
        boothooks:
          'disable-master.sh': |
            #!/bin/bash
            echo "manual" > /etc/init/salt-master.override
        scripts:
          'run_salt.sh': |
            #!/bin/bash

            add-apt-repository -y ppa:saltstack/salt
            apt-get update
            apt-get install -y salt-minion
            salt-call state.highstate
    - region: us-east-1
    - keyid: GKTADJGHEIQSXMKKRBJ08H
    - key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs

# Using a profile from pillars.
Ensure mylc exists:
  boto_lc.present:
    - name: mylc
    - image_id: ami-0b9c9f62
    - profile: myprofile

# Passing in a profile.
Ensure mylc exists:
  boto_lc.present:
    - name: mylc
```

(continues on next page)

(continued from previous page)

```
    - image_id: ami-0b9c9f62
    - profile:
        keyid: GKTADJGHEIQSXMKKRBJ08H
        key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
        region: us-east-1
```

`salt.states.boto_lc.absent`(*name*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Ensure the named launch configuration is deleted.

name Name of the launch configuration.

region The region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_lc.present`(*name*, *image_id*, *key_name=None*, *vpc_id=None*, *vpc_name=None*, *security_groups=None*, *user_data=None*, *cloud_init=None*, *instance_type='m1.small'*, *kernel_id=None*, *ramdisk_id=None*, *block_device_mappings=None*, *delete_on_termination=None*, *instance_monitoring=False*, *spot_price=None*, *instance_profile_name=None*, *ebs_optimized=False*, *associate_public_ip_address=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Ensure the launch configuration exists.

name Name of the launch configuration.

image_id AMI to use for instances. AMI must exist or creation of the launch configuration will fail.

key_name Name of the EC2 key pair to use for instances. Key must exist or creation of the launch configuration will fail.

vpc_id The VPC id where the security groups are defined. Only necessary when using named security groups that exist outside of the default VPC. Mutually exclusive with vpc_name.

vpc_name Name of the VPC where the security groups are defined. Only Necessary when using named security groups that exist outside of the default VPC. Mutually exclusive with vpc_id.

security_groups List of Names or security group id's of the security groups with which to associate the EC2 instances or VPC instances, respectively. Security groups must exist, or creation of the launch configuration will fail.

user_data The user data available to launched EC2 instances.

cloud_init A dict of cloud_init configuration. Currently supported keys: boothooks, scripts and cloud-config. Mutually exclusive with user_data.

instance_type The instance type. ex: m1.small.

kernel_id The kernel id for the instance.

ramdisk_id The RAM disk ID for the instance.

block_device_mappings A dict of block device mappings that contains a dict with volume_type, delete_on_termination, iops, size, encrypted, snapshot_id.

- **volume_type** Indicates what volume type to use. Valid values are standard, io1, gp2. Default is standard.
- **delete_on_termination** Whether the volume should be explicitly marked for deletion when its instance is terminated (True), or left around (False). If not provided, or None is explicitly passed, the default AWS behaviour is used, which is True for ROOT volumes of instances, and False for all others.
- **iops** For Provisioned IOPS (SSD) volumes only. The number of I/O operations per second (IOPS) to provision for the volume.
- **size** Desired volume size (in GiB).
- **encrypted** Indicates whether the volume should be encrypted. Encrypted EBS volumes must be attached to instances that support Amazon EBS encryption. Volumes that are created from encrypted snapshots are automatically encrypted. There is no way to create an encrypted volume from an unencrypted snapshot or an unencrypted volume from an encrypted snapshot.

instance_monitoring Whether instances in group are launched with detailed monitoring.

spot_price The spot price you are bidding. Only applies if you are building an autoscaling group with spot instances.

instance_profile_name The name or the Amazon Resource Name (ARN) of the instance profile associated with the IAM role for the instance. Instance profile must exist or the creation of the launch configuration will fail.

ebs_optimized Specifies whether the instance is optimized for EBS I/O (true) or not (false).

associate_public_ip_address Used for Auto Scaling groups that launch instances into an Amazon Virtual Private Cloud. Specifies whether to assign a public IP address to each instance launched in a Amazon VPC.

region The region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

21.19.46 salt.states.boto_rds

Manage RDSs

New in version 2015.8.0.

Create and destroy RDS instances. Be aware that this interacts with Amazon's services, and so may incur charges.

This module uses `boto`, which can be installed via package, or pip.

This module accepts explicit rds credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More information available here.

If IAM roles are not used you need to specify them either in a pillar file or in the minion's config file:

```
rds.keyid: GKTADJGHEIQSXMKKRBJ08H
rds.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

It's also possible to specify `key`, `keyid` and `region` via a profile, either passed in as a dict, or as a string to pull from pillars or minion config:

```
myprofile:
    keyid: GKTADJGHEIQSXMKKRBJ08H
    key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
        region: us-east-1
```

```
Ensure myrds RDS exists:
  boto_rds.present:
    - name: myrds
    - allocated_storage: 5
    - storage_type: standard
    - db_instance_class: db.t2.micro
    - engine: MySQL
    - master_username: myuser
    - master_user_password: mypass
    - region: us-east-1
    - keyid: GKTADJGHEIQSXMKKRBJ08H
    - key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

(continues on next page)

(continued from previous page)

```
    - tags:
        key: value
```

```
Ensure parameter group exists:
    create-parameter-group:
      boto_rds.parameter_present:
        - name: myparametergroup
        - db_parameter_group_family: mysql5.6
        - description: "parameter group family"
        - parameters:
          - binlog_cache_size: 32768
          - binlog_checksum: CRC32
        - region: eu-west-1
```

depends boto3

`salt.states.boto_rds.`**`absent`**(*name, skip_final_snapshot=None, final_db_snapshot_identifier=None, tags=None, wait_for_deletion=True, timeout=180, region=None, key=None, keyid=None, profile=None*)

Ensure RDS instance is absent.

name Name of the RDS instance.

skip_final_snapshot Whether a final db snapshot is created before the instance is deleted. If True, no snapshot is created. If False, a snapshot is created before deleting the instance.

final_db_snapshot_identifier If a final snapshot is requested, this is the identifier used for that snapshot.

tags A dict of tags.

wait_for_deletion (bool) Wait for the RDS instance to be deleted completely before finishing the state.

timeout (in seconds) The amount of time that can pass before raising an Exception.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_rds.`**`parameter_present`**(*name, db_parameter_group_family, description, parameters=None, apply_method='pending-reboot', tags=None, region=None, key=None, keyid=None, profile=None*)

Ensure DB parameter group exists and update parameters.

name The name for the parameter group.

db_parameter_group_family The DB parameter group family name. A DB parameter group can be associated with one and only one DB parameter group family, and can be applied only to a DB instance running a database engine and engine version compatible with that DB parameter group family.

description Parameter group description.

parameters The DB parameters that need to be changed of type dictionary.

apply_method The *apply-immediate* method can be used only for dynamic parameters; the *pending-reboot* method can be used with MySQL and Oracle DB instances for either dynamic or static parameters. For Microsoft SQL Server DB instances, the *pending-reboot* method can be used only for static parameters.

tags A dict of tags.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_rds.`**`present`**(*name, allocated_storage, db_instance_class, engine, master_username, master_user_password, db_name=None, storage_type=None, db_security_groups=None, vpc_security_group_ids=None, vpc_security_groups=None, availability_zone=None, db_subnet_group_name=None, preferred_maintenance_window=None, db_parameter_group_name=None, db_cluster_identifier=None, tde_credential_arn=None, tde_credential_password=None, storage_encrypted=None, kms_keyid=None, backup_retention_period=None, preferred_backup_window=None, port=None, multi_az=None, engine_version=None, auto_minor_version_upgrade=None, license_model=None, iops=None, option_group_name=None, character_set_name=None, publicly_accessible=None, wait_status=None, tags=None, copy_tags_to_snapshot=None, region=None, domain=None, key=None, keyid=None, monitoring_interval=None, monitoring_role_arn=None, domain_iam_role_name=None, promotion_tier=None, profile=None*)

Ensure RDS instance exists.

name Name of the RDS state definition.

allocated_storage The amount of storage (in gigabytes) to be initially allocated for the database instance.

db_instance_class The compute and memory capacity of the Amazon RDS DB instance.

engine The name of the database engine to be used for this instance. Supported engine types are: MySQL, mariadb, oracle-se1, oracle-se, oracle-ee, sqlserver-ee, sqlserver-se, sqlserver-ex, sqlserver-web, postgres and aurora. For more information, please see the `engine` argument in the Boto3 RDS create_db_instance documentation.

master_username The name of master user for the client DB instance.

master_user_password The password for the master database user. Can be any printable ASCII character except ”/”, ’”’, or ”@”.

db_name The meaning of this parameter differs according to the database engine you use. See the Boto3 RDS documentation to determine the appropriate value for your configuration. https://boto3.readthedocs.io/en/latest/reference/services/rds.html#RDS.Client.create_db_instance

storage_type Specifies the storage type to be associated with the DB instance. Options are standard, gp2 and io1. If you specify io1, you must also include a value for the Iops parameter.

db_security_groups A list of DB security groups to associate with this DB instance.

vpc_security_group_ids A list of EC2 VPC security group IDs to associate with this DB instance.

vpc_security_groups A list of EC2 VPC security groups (IDs or Name tags) to associate with this DB instance.

availability_zone The EC2 Availability Zone that the database instance will be created in.

db_subnet_group_name A DB subnet group to associate with this DB instance.

preferred_maintenance_window The weekly time range (in UTC) during which system maintenance can occur.

db_parameter_group_name A DB parameter group to associate with this DB instance.

db_cluster_identifier If the DB instance is a member of a DB cluster, contains the name of the DB cluster that the DB instance is a member of.

tde_credential_arn The ARN from the Key Store with which the instance is associated for TDE encryption.

tde_credential_password The password to use for TDE encryption if an encryption key is not used.

storage_encrypted Specifies whether the DB instance is encrypted.

kms_keyid If storage_encrypted is true, the KMS key identifier for the encrypted DB instance.

backup_retention_period The number of days for which automated backups are retained.

preferred_backup_window The daily time range during which automated backups are created if automated backups are enabled.

port The port number on which the database accepts connections.

multi_az Specifies if the DB instance is a Multi-AZ deployment. You cannot set the AvailabilityZone parameter if the MultiAZ parameter is set to true.

engine_version The version number of the database engine to use.

auto_minor_version_upgrade Indicates that minor engine upgrades will be applied automatically to the DB instance during the maintenance window.

license_model License model information for this DB instance.

iops The amount of Provisioned IOPS (input/output operations per second) to be initially allocated for the DB instance.

option_group_name Indicates that the DB instance should be associated with the specified option group.

character_set_name For supported engines, indicates that the DB instance should be associated with the specified CharacterSet.

publicly_accessible Specifies the accessibility options for the DB instance. A value of true specifies an Internet-facing instance with a publicly resolvable DNS name, which resolves to a public IP address. A value of false specifies an internal instance with a DNS name that resolves to a private IP address.

wait_status Wait for the RDS instance to reach a desired status before finishing the state. Available states: available, modifying, backing-up

tags A dict of tags.

copy_tags_to_snapshot Specifies whether tags are copied from the DB instance to snapshots of the DB instance.

region Region to connect to.

domain The identifier of the Active Directory Domain.

key AWS secret key to be used.

keyid AWS access key to be used.

monitoring_interval The interval, in seconds, between points when Enhanced Monitoring metrics are collected for the DB instance.

monitoring_role_arn The ARN for the IAM role that permits RDS to send Enhanced Monitoring metrics to CloudWatch Logs.

domain_iam_role_name Specify the name of the IAM role to be used when making API calls to the Directory Service.

promotion_tier A value that specifies the order in which an Aurora Replica is promoted to the primary instance after a failure of the existing primary instance. For more information, see Fault Tolerance for an Aurora DB Cluster .

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_rds.`**`replica_present`**(*name, source, db_instance_class=None, availability_zone=None, port=None, auto_minor_version_upgrade=None, iops=None, option_group_name=None, publicly_accessible=None, tags=None, region=None, key=None, keyid=None, profile=None, db_parameter_group_name=None*)

Ensure RDS replica exists.

```
Ensure myrds replica RDS exists:
  boto_rds.create_replica:
    - name: myreplica
    - source: mydb
```

`salt.states.boto_rds.`**`subnet_group_absent`**(*name, tags=None, region=None, key=None, keyid=None, profile=None*)

`salt.states.boto_rds.`**`subnet_group_present`**(*name, description, subnet_ids=None, subnet_names=None, tags=None, region=None, key=None, keyid=None, profile=None*)

Ensure DB subnet group exists.

name The name for the DB subnet group. This value is stored as a lowercase string.

subnet_ids A list of the EC2 Subnet IDs for the DB subnet group. Either subnet_ids or subnet_names must be provided.

subnet_names A list of The EC2 Subnet names for the DB subnet group. Either subnet_ids or subnet_names must be provided.

description Subnet group description.

tags A dict of tags.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

21.19.47 salt.states.boto_route53

Manage Route53 records

New in version 2014.7.0.

Create and delete Route53 records. Be aware that this interacts with Amazon's services, and so may incur charges.

This module uses `boto`, which can be installed via package, or pip.

This module accepts explicit route53 credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More information available here.

If IAM roles are not used you need to specify them either in a pillar file or in the minion's config file:

```
route53.keyid: GKTADJGHEIQSXMKKRBJ08H
route53.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

It's also possible to specify `key`, `keyid` and `region` via a profile, either passed in as a dict, or as a string to pull from pillars or minion config:

```
myprofile:
  keyid: GKTADJGHEIQSXMKKRBJ08H
  key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
  region: us-east-1
```

```
mycnamerecord:
  boto_route53.present:
    - name: test.example.com.
    - value: my-elb.us-east-1.elb.amazonaws.com.
    - zone: example.com.
    - ttl: 60
    - record_type: CNAME
    - region: us-east-1
    - keyid: GKTADJGHEIQSXMKKRBJ08H
    - key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs

# Using a profile from pillars
myarecord:
  boto_route53.present:
    - name: test.example.com.
    - value: 1.1.1.1
    - zone: example.com.
    - ttl: 60
    - record_type: A
    - region: us-east-1
```

(continues on next page)

(continued from previous page)

```
    - profile: myprofile

# Passing in a profile
myarecord:
  boto_route53.present:
    - name: test.example.com.
    - value: 1.1.1.1
    - zone: example.com.
    - ttl: 60
    - record_type: A
    - region: us-east-1
    - profile:
        keyid: GKTADJGHEIQSXMKKRBJ08H
        key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

`salt.states.boto_route53.`**`absent`**`(`*name, zone, record_type, identifier=None, region=None, key=None, keyid=None, profile=None, wait_for_sync=True, split_dns=False, private_zone=False*`)`

Ensure the Route53 record is deleted.

name Name of the record.

zone The zone to delete the record from.

record_type The record type (A, NS, MX, TXT, etc.)

identifier An identifier to match for deletion.

region The region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

wait_for_sync Wait for an INSYNC change status from Route53.

split_dns Route53 supports a public and private DNS zone with the same names.

private_zone If using split_dns, specify if this is the private zone.

`salt.states.boto_route53.`**`hosted_zone_absent`**`(`*name, domain_name=None, region=None, key=None, keyid=None, profile=None*`)`

Ensure the Route53 Hostes Zone described is absent

name The name of the state definition.

domain_name The FQDN (including final period) of the zone you wish absent. If not provided, the value of name will be used.

`salt.states.boto_route53.`**`hosted_zone_present`**`(`*name, domain_name=None, private_zone=False, caller_ref=None, comment=”, vpc_id=None, vpc_name=None, vpc_region=None, region=None, key=None, keyid=None, profile=None*`)`

Ensure a hosted zone exists with the given attributes. Note that most things cannot be modified once a zone is created - it must be deleted and re-spun to update these attributes:

- private_zone (AWS API limitation).
- comment (the appropriate call exists in the AWS API and in boto3, but has not, as of this writing, been added to boto2).
- vpc_id (same story - we really need to rewrite this module with boto3)
- vpc_name (really just a pointer to vpc_id anyway).
- vpc_region (again, supported in boto3 but not boto2).

If you need the ability to update these attributes, please use the newer boto3_route53 module instead.

name The name of the state definition.

domain_name The name of the domain. This must be fully-qualified, terminating with a period. This is the name you have registered with your domain registrar. It is also the name you will delegate from your registrar to the Amazon Route 53 delegation servers returned in response to this request. Defaults to the value of name if not provided.

private_zone Set True if creating a private hosted zone.

caller_ref A unique string that identifies the request and that allows create_hosted_zone() calls to be retried without the risk of executing the operation twice. This helps ensure idempotency across state calls, but can cause issues if a zone is deleted and then an attempt is made to recreate it with the same caller_ref. If not provided, a unique UUID will be generated at each state run, which avoids the risk of the above (transient) error. This option is generally not needed. Maximum length of 128.

comment Any comments you want to include about the hosted zone.

vpc_id When creating a private hosted zone, either the VPC ID or VPC Name to associate with is required. Exclusive with vpe_name. Ignored when creating a non-private zone.

vpc_name When creating a private hosted zone, either the VPC ID or VPC Name to associate with is required. Exclusive with vpe_id. Ignored when creating a non-private zone.

vpc_region When creating a private hosted zone, the region of the associated VPC is required. If not provided, an effort will be made to determine it from vpc_id or vpc_name, where possible. If this fails, you'll need to provide an explicit value for this option. Ignored when creating a non-private zone.

`salt.states.boto_route53.`**`present`**`(`*name*, *value*, *zone*, *record_type*, *ttl=None*, *identifier=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*, *wait_for_sync=True*, *split_dns=False*, *private_zone=False*`)`

Ensure the Route53 record is present.

name Name of the record.

value

Value of the record. As a special case, you can pass in: *private:<Name tag>* to have the function autodetermine the private IP *public:<Name tag>* to have the function autodetermine the public IP

zone The zone to create the record in.

record_type The record type (A, NS, MX, TXT, etc.)

ttl The time to live for the record.

identifier The unique identifier to use for this record.

region The region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

wait_for_sync Wait for an INSYNC change status from Route53 before returning success.

split_dns Route53 supports parallel public and private DNS zones with the same name.

private_zone If using split_dns, specify if this is the private zone.

`salt.states.boto_route53.`**`rr_absent`**`(`**args*, ***kwargs*`)`

`salt.states.boto_route53.`**`rr_present`**`(`**args*, ***kwargs*`)`

21.19.48 salt.states.boto_s3_bucket module

Manage S3 Buckets

New in version 2016.3.0.

Create and destroy S3 buckets. Be aware that this interacts with Amazon's services, and so may incur charges.

depends

- boto
- boto3

The dependencies listed above can be installed via package or pip.

This module accepts explicit vpc credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More information available here.

If IAM roles are not used you need to specify them either in a pillar file or in the minion's config file:

```
vpc.keyid: GKTADJGHEIQSXMKKRBJ08H
vpc.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

It's also possible to specify `key`, `keyid` and `region` via a profile, either passed in as a dict, or as a string to pull from pillars or minion config:

```
myprofile:
    keyid: GKTADJGHEIQSXMKKRBJ08H
    key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
    region: us-east-1
```

```
Ensure bucket exists:
    boto_s3_bucket.present:
        - Bucket: mybucket
        - LocationConstraint: EU
        - ACL:
          - GrantRead: "uri=http://acs.amazonaws.com/groups/global/AllUsers"
        - CORSRules:
          - AllowedHeaders: []
            AllowedMethods: ["GET"]
            AllowedOrigins: ["*"]
            ExposeHeaders: []
            MaxAgeSeconds: 123
        - LifecycleConfiguration:
          - Expiration:
              Days: 123
            ID: "idstring"
            Prefix: "prefixstring"
            Status: "enabled",
            ID: "lc1"
            Transitions:
              - Days: 123
                StorageClass: "GLACIER"
            NoncurrentVersionTransitions:
              - NoncurrentDays: 123
                StorageClass: "GLACIER"
            NoncurrentVersionExpiration:
              NoncurrentDays: 123
        - Logging:
            TargetBucket: log_bucket
            TargetPrefix: prefix
            TargetGrants:
              - Grantee:
                  DisplayName: "string"
                  EmailAddress: "string"
                  ID: "string"
                  Type: "AmazonCustomerByEmail"
                  URI: "string"
                Permission: "READ"
        - NotificationConfiguration:
```

(continues on next page)

(continued from previous page)

```
            LambdaFunctionConfiguration:
              - Id: "string"
                LambdaFunctionArn: "string"
                Events:
                  - "s3:ObjectCreated:*"
                Filter:
                  Key:
                    FilterRules:
                      - Name: "prefix"
                        Value: "string"
        - Policy:
            Version: "2012-10-17"
            Statement:
              - Sid: "String"
                Effect: "Allow"
                Principal:
                  AWS: "arn:aws:iam::133434421342:root"
                Action: "s3:PutObject"
                Resource: "arn:aws:s3:::my-bucket/*"
        - Replication:
            Role: myrole
            Rules:
              - ID: "string"
                Prefix: "string"
                Status: "Enabled"
                Destination:
                  Bucket: "arn:aws:s3:::my-bucket"
        - RequestPayment:
            Payer: Requester
        - Tagging:
            tag_name: tag_value
            tag_name_2: tag_value
        - Versioning:
            Status: "Enabled"
        - Website:
            ErrorDocument:
              Key: "error.html"
            IndexDocument:
              Suffix: "index.html"
            RedirectAllRequestsTo:
              Hostname: "string"
              Protocol: "http"
            RoutingRules:
              - Condition:
                  HttpErrorCodeReturnedEquals: "string"
                  KeyPrefixEquals: "string"
                Redirect:
                  HostName: "string"
                  HttpRedirectCode: "string"
                  Protocol: "http"
                  ReplaceKeyPrefixWith: "string"
                  ReplaceKeyWith: "string"
        - region: us-east-1
        - keyid: GKTADJGHEIQSXMKKRBJ08H
        - key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

`salt.states.boto_s3_bucket.`**`absent`**`(`*name*, *Bucket*, *Force=False*, *region=None*, *key=None*, *keyid=None*, *profile=None*`)`

Ensure bucket with passed properties is absent.

name The name of the state definition.

Bucket Name of the bucket.

Force Empty the bucket first if necessary - Boolean.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_s3_bucket.`**`present`**`(`*name*, *Bucket*, *LocationConstraint=None*, *ACL=None*, *CORSRules=None*, *LifecycleConfiguration=None*, *Logging=None*, *NotificationConfiguration=None*, *Policy=None*, *Replication=None*, *RequestPayment=None*, *Tagging=None*, *Versioning=None*, *Website=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*`)`

Ensure bucket exists.

name The name of the state definition

Bucket Name of the bucket.

LocationConstraint 'EU'|'eu-west-1'|'us-west-1'|'us-west-2'|'ap-southeast-1'|'ap-southeast-2'|'ap-northeast-1'|'sa-east-1'|'cn-north-1'|'eu-central-1'

ACL The permissions on a bucket using access control lists (ACL).

CORSRules The cors configuration for a bucket.

LifecycleConfiguration Lifecycle configuration for your bucket

Logging The logging parameters for a bucket and to specify permissions for who can view and modify the logging parameters.

NotificationConfiguration notifications of specified events for a bucket

Policy Policy on the bucket

Replication Replication rules. You can add as many as 1,000 rules. Total replication configuration size can be up to 2 MB

RequestPayment The request payment configuration for a bucket. By default, the bucket owner pays for downloads from the bucket. This configuration parameter enables the bucket owner (only) to specify that the person requesting the download will be charged for the download

Tagging A dictionary of tags that should be set on the bucket

Versioning The versioning state of the bucket

Website The website configuration of the bucket

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

21.19.49 salt.states.boto_secgroup

Manage Security Groups

New in version 2014.7.0.

Create and destroy Security Groups. Be aware that this interacts with Amazon's services, and so may incur charges.

This module uses `boto`, which can be installed via package, or pip.

This module accepts explicit EC2 credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More information available here.

If IAM roles are not used you need to specify them either in a pillar file or in the minion's config file:

```
secgroup.keyid: GKTADJGHEIQSXMKKRBJ08H
secgroup.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

It's also possible to specify `key`, `keyid` and `region` via a profile, either passed in as a dict, or as a string to pull from pillars or minion config:

```
myprofile:
    keyid: GKTADJGHEIQSXMKKRBJ08H
    key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
    region: us-east-1
```

```
Ensure mysecgroup exists:
    boto_secgroup.present:
        - name: mysecgroup
        - description: My security group
        - vpc_name: myvpc
        - rules:
            - ip_protocol: tcp
              from_port: 80
              to_port: 80
              cidr_ip:
                - 10.0.0.0/8
                - 192.168.0.0/16
            - ip_protocol: tcp
              from_port: 8080
              to_port: 8090
              cidr_ip:
                - 10.0.0.0/8
                - 192.168.0.0/16
            - ip_protocol: icmp
              from_port: -1
              to_port: -1
              source_group_name: mysecgroup
            - ip_protocol: tcp
              from_port: 8080
              to_port: 8080
              source_group_name: MyOtherSecGroup
              source_group_name_vpc: MyPeeredVPC
        - rules_egress:
            - ip_protocol: all
              from_port: -1
              to_port: -1
              cidr_ip:
                - 10.0.0.0/8
                - 192.168.0.0/16
        - tags:
            SomeTag: 'My Tag Value'
            SomeOtherTag: 'Other Tag Value'
        - region: us-east-1
        - keyid: GKTADJGHEIQSXMKKRBJ08H
        - key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs

# Using a profile from pillars
```

(continues on next page)

(continued from previous page)

```
Ensure mysecgroup exists:
    boto_secgroup.present:
        - name: mysecgroup
        - description: My security group
        - profile: myprofile

# Passing in a profile
Ensure mysecgroup exists:
    boto_secgroup.present:
        - name: mysecgroup
        - description: My security group
        - profile:
            keyid: GKTADJGHEIQSXMKKRBJ08H
            key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
            region: us-east-1
```

Note: When using the `profile` parameter and `region` is set outside of the profile group, region is ignored and a default region will be used.

If `region` is missing from the `profile` data set, `us-east-1` will be used as the default region.

`salt.states.boto_secgroup.`**`absent`**`(`*name, vpc_id=None, vpc_name=None, region=None, key=None, keyid=None, profile=None*`)`

Ensure a security group with the specified name does not exist.

name Name of the security group.

vpc_id The ID of the VPC to remove the security group from, if any. Exclusive with vpc_name.

vpc_name The name of the VPC to remove the security group from, if any. Exclusive with vpc_name.

New in version 2016.3.0.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

New in version 2016.3.0.

`salt.states.boto_secgroup.`**`present`**`(`*name, description, vpc_id=None, vpc_name=None, rules=None, rules_egress=None, delete_ingress_rules=True, delete_egress_rules=True, region=None, key=None, keyid=None, profile=None, tags=None*`)`

Ensure the security group exists with the specified rules.

name Name of the security group.

description A description of this security group.

vpc_id The ID of the VPC to create the security group in, if any. Exclusive with vpc_name.

vpc_name The name of the VPC to create the security group in, if any. Exclusive with vpc_id.

New in version 2016.3.0.

New in version 2015.8.2.

rules A list of ingress rule dicts. If not specified, `rules=None`, the ingress rules will be unmanaged. If set to an empty list, `[]`, then all ingress rules will be removed.

rules_egress A list of egress rule dicts. If not specified, `rules_egress=None`, the egress rules will be unmanaged. If set to an empty list, `[]`, then all egress rules will be removed.

delete_ingress_rules Some tools (EMR comes to mind) insist on adding rules on-the-fly, which salt will happily remove on the next run. Set this param to False to avoid deleting rules which were added outside of salt.

delete_egress_rules Some tools (EMR comes to mind) insist on adding rules on-the-fly, which salt will happily remove on the next run. Set this param to False to avoid deleting rules which were added outside of salt.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key, and keyid.

tags List of key:value pairs of tags to set on the security group

New in version 2016.3.0.

21.19.50 salt.states.boto_sns

Manage SNS Topics

Create and destroy SNS topics. Be aware that this interacts with Amazon's services, and so may incur charges.

This module uses `boto`, which can be installed via package, or pip.

This module accepts explicit AWS credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More information available here.

If IAM roles are not used you need to specify them either in a pillar file or in the minion's config file:

```
sns.keyid: GKTADJGHEIQSXMKKRBJ08H
sns.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

It's also possible to specify `key`, `keyid` and `region` via a profile, either passed in as a dict, or as a string to pull from pillars or minion config:

```
myprofile:
    keyid: GKTADJGHEIQSXMKKRBJ08H
    key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
    region: us-east-1
```

```
mytopic:
    boto_sns.present:
        - region: us-east-1
        - keyid: GKTADJGHEIQSXMKKRBJ08H
        - key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs

# Using a profile from pillars
mytopic:
    boto_sns.present:
        - region: us-east-1
        - profile: mysnsprofile

# Passing in a profile
mytopic:
    boto_sns.present:
        - region: us-east-1
        - profile:
            keyid: GKTADJGHEIQSXMKKRBJ08H
            key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

`salt.states.boto_sns.`**`absent`**(*name*, *region=None*, *key=None*, *keyid=None*, *profile=None*, *unsubscribe=False*)

Ensure the named sns topic is deleted.

name Name of the SNS topic.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

unsubscribe If True, unsubscribe all subcriptions to the SNS topic before deleting the SNS topic

New in version 2016.11.0.

`salt.states.boto_sns.`**`present`**(*name*, *subscriptions=None*, *region=None*, *key=None*, *keyid=None*, *profile=None*)

Ensure the SNS topic exists.

name Name of the SNS topic.

subscriptions List of SNS subscriptions.

Each subscription is a dictionary with a protocol and endpoint key:

```
[
{'protocol': 'https', 'endpoint': 'https://www.example.com/sns-endpoint'},
{'protocol': 'sqs', 'endpoint': 'arn:aws:sqs:us-west-2:123456789012:MyQueue'}
]
```

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

21.19.51 salt.states.boto_sqs

Manage SQS Queues

New in version 2014.7.0.

Create and destroy SQS queues. Be aware that this interacts with Amazon's services, and so may incur charges.

This module uses `boto`, which can be installed via package, or pip.

This module accepts explicit SQS credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More information available here.

If IAM roles are not used you need to specify them either in a pillar file or in the minion's config file:

```
sqs.keyid: GKTADJGHEIQSXMKKRBJ08H
sqs.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

It's also possible to specify `key`, `keyid` and `region` via a profile, either passed in as a dict, or as a string to pull from pillars or minion config:

```
myprofile:
    keyid: GKTADJGHEIQSXMKKRBJ08H
    key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
    region: us-east-1
```

```
myqueue:
    boto_sqs.present:
        - region: us-east-1
        - keyid: GKTADJGHEIQSXMKKRBJ08H
        - key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
        - attributes:
            ReceiveMessageWaitTimeSeconds: 20

# Using a profile from pillars
myqueue:
    boto_sqs.present:
        - region: us-east-1
        - profile: mysqsprofile

# Passing in a profile
myqueue:
    boto_sqs.present:
        - region: us-east-1
        - profile:
            keyid: GKTADJGHEIQSXMKKRBJ08H
            key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

`salt.states.boto_sqs.`**`absent`**(*name, region=None, key=None, keyid=None, profile=None*)

Ensure the named sqs queue is deleted.

name Name of the SQS queue.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_sqs.`**`present`**(*name, attributes=None, region=None, key=None, keyid=None, profile=None*)

Ensure the SQS queue exists.

name Name of the SQS queue.

attributes A dict of key/value SQS attributes.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

21.19.52 salt.states.boto_vpc

Manage VPCs

New in version 2015.8.0.

depends

- boto >= 2.8.0
- boto3 >= 1.2.6

Create and destroy VPCs. Be aware that this interacts with Amazon's services, and so may incur charges.

This module accepts explicit vpc credentials but can also utilize IAM roles assigned to the instance through Instance Profiles. Dynamic credentials are then automatically obtained from AWS API and no further configuration is necessary. More information available here.

If IAM roles are not used you need to specify them either in a pillar file or in the minion's config file:

```
vpc.keyid: GKTADJGHEIQSXMKKRBJ08H
vpc.key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

It's also possible to specify `key`, `keyid` and `region` via a profile, either passed in as a dict, or as a string to pull from pillars or minion config:

```
myprofile:
  keyid: GKTADJGHEIQSXMKKRBJ08H
  key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
  region: us-east-1
```

```
aws:
  region:
    us-east-1:
      profile:
        keyid: GKTADJGHEIQSXMKKRBJ08H
        key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
        region: us-east-1
```

```
Ensure VPC exists:
  boto_vpc.present:
    - name: myvpc
    - cidr_block: 10.10.11.0/24
    - dns_hostnames: True
    - region: us-east-1
    - keyid: GKTADJGHEIQSXMKKRBJ08H
    - key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs

Ensure subnet exists:
  boto_vpc.subnet_present:
    - name: mysubnet
    - vpc_id: vpc-123456
    - cidr_block: 10.0.0.0/16
    - region: us-east-1
    - profile: myprofile

{% set profile = salt['pillar.get']('aws:region:us-east-1:profile' ) %}
Ensure internet gateway exists:
  boto_vpc.internet_gateway_present:
    - name: myigw
    - vpc_name: myvpc
    - profile: {{ profile }}

Ensure route table exists:
  boto_vpc.route_table_present:
    - name: my_route_table
    - vpc_id: vpc-123456
    - routes:
      - destination_cidr_block: 0.0.0.0/0
        instance_id: i-123456
      - subnet_names:
```

(continues on next page)

(continued from previous page)

```
        - subnet1
        - subnet2
      - region: us-east-1
      - profile:
        keyid: GKTADJGHEIQSXMKKRBJ08H
        key: askdjghsdfjkghWupUjasdflkdfklgjsdfjajkghs
```

New in version 2016.11.0.

Request, accept and delete VPC peering connections. VPC peering connections can be named allowing the name to be used throughout the state file. Following example shows how to request and accept a VPC peering connection.

```
accept the vpc peering connection:
  boto_vpc.accept_vpc_peering_connection:
    - conn_name: salt_vpc_peering
    - region: us-west-2
    - require:
      - boto_vpc: request a vpc peering connection

request a vpc peering connection:
  boto_vpc.request_vpc_peering_connection:
    - requester_vpc_id: vpc-4a3d522e
    - peer_vpc_id: vpc-ae81e9ca
    - region: us-west-2
    - conn_name: salt_vpc_peering
```

VPC peering connections need not be named. In this case the VPC peering connection ID should be used in the state file.

```
accept the vpc peering connection:
  boto_vpc.accept_vpc_peering_connection:
    - conn_id: pcx-1873c371
    - region: us-west-2
```

VPC peering connections can be deleted, as shown below.

```
delete a named vpc peering connection:
  boto_vpc.delete_vpc_peering_connection:
    - conn_name: salt_vpc_peering
```

Delete also accepts a VPC peering connection id.

```
delete a vpc peering connection by id:
  boto_vpc.delete_vpc_peering_connection:
    - conn_id: pcx-1873c371
```

`salt.states.boto_vpc.`**`absent`**`(`*name, tags=None, region=None, key=None, keyid=None, profile=None*`)`

Ensure VPC with passed properties is absent.

name Name of the VPC.

tags A list of tags. All tags must match.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_vpc.`**`accept_vpc_peering_connection`**`(`*name=None, conn_id=None, conn_name=None, region=None, key=None, keyid=None, profile=None*`)`

Accept a VPC pending requested peering connection between two VPCs.

name Name of this state

conn_id The connection ID to accept. Exclusive with conn_name. String type.

conn_name The name of the VPC peering connection to accept. Exclusive with conn_id. String type.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

New in version 2016.11.0.

Example:

```
boto_vpc.accept_vpc_peering_connection:
  - conn_name: salt_peering_connection

# usage with vpc peering connection id and region
boto_vpc.accept_vpc_peering_connection:
  - conn_id: pbx-1873d472
  - region: us-west-2
```

`salt.states.boto_vpc.`**`delete_vpc_peering_connection`**`(`*name, conn_id=None, conn_name=None, region=None, key=None, keyid=None, profile=None*`)`

name Name of the state

conn_id ID of the peering connection to delete. Exclusive with conn_name.

conn_name The name of the peering connection to delete. Exclusive with conn_id.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

New in version 2016.11.0.

Example:

```
delete a vpc peering connection:
  boto_vpc.delete_vpc_peering_connection:
    - region: us-west-2
    - conn_id: pcx-4613b12e
```

Connection name can be specified (instead of ID). Specifying both conn_name and conn_id will result in an error.

```
delete a vpc peering connection:
  boto_vpc.delete_vpc_peering_connection:
    - conn_name: salt_vpc_peering
```

`salt.states.boto_vpc.`**`dhcp_options_absent`**`(`*name=None, dhcp_options_id=None, region=None, key=None, keyid=None, profile=None*`)`

Ensure a set of DHCP options with the given settings exist.

name (string) Name of the DHCP options set.

dhcp_options_id (string) Id of the DHCP options set.

region (string) Region to connect to.

key (string) Secret key to be used.

keyid (string) Access key to be used.

profile (various) A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

New in version 2016.3.0.

`salt.states.boto_vpc.`**`dhcp_options_present`**`(`*name, dhcp_options_id=None, vpc_name=None, vpc_id=None, domain_name=None, domain_name_servers=None, ntp_servers=None, netbios_name_servers=None, netbios_node_type=None, tags=None, region=None, key=None, keyid=None, profile=None*`)`

Ensure a set of DHCP options with the given settings exist. Note that the current implementation only SETS values during option set creation. It is unable to update option sets in place, and thus merely verifies the set exists via the given name and/or dhcp_options_id param.

name (string) Name of the DHCP options.

vpc_name (string) Name of a VPC to which the options should be associated. Either vpc_name or vpc_id must be provided.

vpc_id (string) Id of a VPC to which the options should be associated. Either vpc_name or vpc_id must be provided.

domain_name (string) Domain name to be assiciated with this option set.

domain_name_servers (list of strings) The IP address(es) of up to four domain name servers.

ntp_servers (list of strings) The IP address(es) of up to four desired NTP servers.

netbios_name_servers (list of strings) The IP address(es) of up to four NetBIOS name servers.

netbios_node_type (string) The NetBIOS node type (1, 2, 4, or 8). For more information about the allowed values, see RFC 2132. The recommended is 2 at this time (broadcast and multicast are currently not supported).

tags (dict of key:value pairs) A set of tags to be added.

region (string) Region to connect to.

key (string) Secret key to be used.

keyid (string) Access key to be used.

profile (various) A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

New in version 2016.3.0.

`salt.states.boto_vpc.`**`internet_gateway_absent`**`(`*name, detach=False, region=None, key=None, keyid=None, profile=None*`)`

Ensure the named internet gateway is absent.

name Name of the internet gateway.

detach First detach the internet gateway from a VPC, if attached.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_vpc.`**`internet_gateway_present`**`(`*name, vpc_name=None, vpc_id=None, tags=None, region=None, key=None, keyid=None, profile=None*`)`

Ensure an internet gateway exists.

name Name of the internet gateway.

vpc_name Name of the VPC to which the internet gateway should be attached.

vpc_id Id of the VPC to which the internet_gateway should be attached. Only one of vpc_name or vpc_id may be provided.

tags A list of tags.
region Region to connect to.
key Secret key to be used.
keyid Access key to be used.
profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_vpc.`**`nat_gateway_absent`**(*name=None, subnet_name=None, subnet_id=None, region=None, key=None, keyid=None, profile=None, wait_for_delete_retries=0*)

Ensure the nat gateway in the named subnet is absent.

This function requires boto3.

New in version 2016.11.0.
name Name of the state.
subnet_name Name of the subnet within which the nat gateway should exist
subnet_id Id of the subnet within which the nat gateway should exist. Either subnet_name or subnet_id must be provided.
region Region to connect to.
key Secret key to be used.
keyid Access key to be used.
profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.
wait_for_delete_retries NAT gateway may take some time to be go into deleted or failed state. During the deletion process, subsequent release of elastic IPs may fail; this state will automatically retry this number of times to ensure the NAT gateway is in deleted or failed state before proceeding. Default is set to 0 for backward compatibility.

`salt.states.boto_vpc.`**`nat_gateway_present`**(*name, subnet_name=None, subnet_id=None, region=None, key=None, keyid=None, profile=None, allocation_id=None*)

Ensure a nat gateway exists within the specified subnet

This function requires boto3.

New in version 2016.11.0.

Example:

```
boto_vpc.nat_gateway_present:
  - subnet_name: my-subnet
```

name Name of the state
subnet_name Name of the subnet within which the nat gateway should exist
subnet_id Id of the subnet within which the nat gateway should exist. Either subnet_name or subnet_id must be provided.
allocation_id If specified, the elastic IP address referenced by the ID is associated with the gateway. Otherwise, a new allocation_id is created and used.
region Region to connect to.
key Secret key to be used.
keyid Access key to be used.
profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_vpc.`**`present`**(*name, cidr_block, instance_tenancy=None, dns_support=None, dns_hostnames=None, tags=None, region=None, key=None, keyid=None, profile=None*)

Ensure VPC exists.

name Name of the VPC.
cidr_block The range of IPs in CIDR format, for example: 10.0.0.0/24. Block size must be between /16 and /28 netmask.
instance_tenancy Instances launched in this VPC will be ingle-tenant or dedicated hardware.
dns_support Indicates whether the DNS resolution is supported for the VPC.
dns_hostnames Indicates whether the instances launched in the VPC get DNS hostnames.
tags A list of tags.
region Region to connect to.
key Secret key to be used.
keyid Access key to be used.
profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_vpc.`**`request_vpc_peering_connection`**(*name, requester_vpc_id=None, requester_vpc_name=None, peer_vpc_id=None, peer_vpc_name=None, conn_name=None, peer_owner_id=None, region=None, key=None, keyid=None, profile=None*)

name Name of the state
requester_vpc_id ID of the requesting VPC. Exclusive with requester_vpc_name. String type.
requester_vpc_name Name tag of the requesting VPC. Exclusive with requester_vpc_id. String type.
peer_vpc_id ID of the VPC tp crete VPC peering connection with. This can be a VPC in another account. Exclusive with peer_vpc_name. String type.
peer_vpc_name Name tag of the VPC tp crete VPC peering connection with. This can only be a VPC the same account. Exclusive with peer_vpc_id. String type.
conn_name The (optional) name to use for this VPC peering connection. String type.
peer_owner_id ID of the owner of the peer VPC. String type. If this isn't supplied AWS uses your account ID. Required if peering to a different account.
region Region to connect to.
key Secret key to be used.
keyid Access key to be used.
profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

New in version 2016.11.0.

Example:

```
request a vpc peering connection:
  boto_vpc.request_vpc_peering_connection:
    - requester_vpc_id: vpc-4b3522e
    - peer_vpc_id: vpc-ae83f9ca
    - conn_name: salt_peering_connection
```

`salt.states.boto_vpc.`**`route_table_absent`**(*name, region=None, key=None, keyid=None, profile=None*)

Ensure the named route table is absent.

name Name of the route table.
region Region to connect to.
key Secret key to be used.
keyid Access key to be used.
profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_vpc.`**`route_table_present`**(*name, vpc_name=None, vpc_id=None, routes=None, subnet_ids=None, subnet_names=None, tags=None, region=None, key=None, keyid=None, profile=None*)

Ensure route table with routes exists and is associated to a VPC.

This function requires boto3 to be installed if nat gatewyas are specified.

Example:

```
boto_vpc.route_table_present:
  - name: my_route_table
  - vpc_id: vpc-123456
  - routes:
    - destination_cidr_block: 0.0.0.0/0
      internet_gateway_name: InternetGateway
    - destination_cidr_block: 10.10.11.0/24
      instance_id: i-123456
    - destination_cidr_block: 10.10.12.0/24
      interface_id: eni-123456
    - destination_cidr_block: 10.10.13.0/24
      instance_name: mygatewayserver
  - subnet_names:
    - subnet1
    - subnet2
```

name Name of the route table.
vpc_name Name of the VPC with which the route table should be associated.
vpc_id Id of the VPC with which the route table should be associated. Either vpc_name or vpc_id must be provided.
routes A list of routes. Each route has a cidr and a target.
subnet_ids A list of subnet ids to associate
subnet_names A list of subnet names to associate
tags A list of tags.
region Region to connect to.
key Secret key to be used.
keyid Access key to be used.
profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_vpc.`**`subnet_absent`**(*name=None, subnet_id=None, region=None, key=None, keyid=None, profile=None*)

Ensure subnet with passed properties is absent.
name Name of the subnet.
region Region to connect to.
key Secret key to be used.
keyid Access key to be used.
profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_vpc.`**`subnet_present`**(*name, cidr_block, vpc_name=None, vpc_id=None, availability_zone=None, tags=None, region=None, key=None, keyid=None, profile=None, route_table_id=None, route_table_name=None, auto_assign_public_ipv4=False*)

Ensure a subnet exists.
name Name of the subnet.

cidr_block The range if IPs for the subnet, in CIDR format. For example: 10.0.0.0/24. Block size must be between /16 and /28 netmask.

vpc_name Name of the VPC in which the subnet should be placed. Either vpc_name or vpc_id must be provided.

vpc_id Id of the VPC in which the subnet should be placed. Either vpc_name or vpc_id must be provided.

availability_zone AZ in which the subnet should be placed.

tags A list of tags.

route_table_id A route table ID to explicitly associate the subnet with. If both route_table_id and route_table_name are specified, route_table_id will take precedence.

New in version 2016.11.0.

route_table_name A route table name to explicitly associate the subnet with. If both route_table_id and route_table_name are specified, route_table_id will take precedence.

New in version 2016.11.0.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

`salt.states.boto_vpc.`**`vpc_peering_connection_absent`**`(`*name, conn_id=None, conn_name=None, region=None, key=None, keyid=None, profile=None*`)`

`salt.states.boto_vpc.`**`vpc_peering_connection_present`**`(`*name, requester_vpc_id=None, requester_vpc_name=None, peer_vpc_id=None, peer_vpc_name=None, conn_name=None, peer_owner_id=None, region=None, key=None, keyid=None, profile=None*`)`

name Name of the state

requester_vpc_id ID of the requesting VPC. Exclusive with requester_vpc_name.

requester_vpc_name Name tag of the requesting VPC. Exclusive with requester_vpc_id.

peer_vpc_id ID of the VPC tp crete VPC peering connection with. This can be a VPC in another account. Exclusive with peer_vpc_name.

peer_vpc_name Name tag of the VPC tp crete VPC peering connection with. This can only be a VPC in the same account, else resolving it into a vpc ID will fail. Exclusive with peer_vpc_id.

conn_name The name to use for this VPC peering connection.

peer_owner_id ID of the owner of the peer VPC. Defaults to your account ID, so a value is required if peering with a VPC in a different account.

region Region to connect to.

key Secret key to be used.

keyid Access key to be used.

profile A dict with region, key and keyid, or a pillar key (string) that contains a dict with region, key and keyid.

New in version 2016.11.0.

Example:

```
ensure peering twixt local vpc and the other guys:
  boto_vpc.vpc_peering_connection_present:
    - requester_vpc_name: my_local_vpc
```

(continues on next page)

(continued from previous page)

```
    - peer_vpc_name: some_other_guys_vpc
    - conn_name: peering_from_here_to_there
    - peer_owner_id: 012345654321
```

21.19.53 salt.states.bower

Installation of Bower Packages

These states manage the installed packages using Bower. Note that npm, git and bower must be installed for these states to be available, so bower states should include requisites to pkg.installed states for the packages which provide npm and git (simply `npm` and `git` in most cases), and npm.installed state for the package which provides bower.

Example:

```
npm:
  pkg.installed
git:
  pkg.installed
bower:
  npm.installed
  require:
    - pkg: npm
    - pkg: git

underscore:
  bower.installed:
    - dir: /path/to/project
    - require:
      - npm: bower
```

`salt.states.bower.`**`bootstrap`**`(`*name*, *user=None*`)`

Bootstraps a frontend distribution.

Will execute 'bower install' on the specified directory.
user The user to run Bower with

`salt.states.bower.`**`installed`**`(`*name*, *dir*, *pkgs=None*, *user=None*, *env=None*`)`

Verify that the given package is installed and is at the correct version (if specified).

```
underscore:
  bower.installed:
    - dir: /path/to/project
    - user: someuser

jquery#2.0:
  bower.installed:
    - dir: /path/to/project
```

name The package to install
dir The target directory in which to install the package
pkgs A list of packages to install with a single Bower invocation; specifying this argument will ignore the `name` argument
user The user to run Bower with
env A list of environment variables to be set prior to execution. The format is the same as the *`cmd.run`*. state function.

`salt.states.bower.`**`pruned`**`(`*name*, *user=None*, *env=None*`)`

New in version 2017.7.0.

Cleans up local bower_components directory.

Will execute 'bower prune' on the specified directory (param: name)

user The user to run Bower with

`salt.states.bower.`**`removed`**`(`*name*, *dir*, *user=None*`)`

Verify that the given package is not installed.

dir The target directory in which to install the package

user The user to run Bower with

21.19.54 salt.states.cabal

Installation of Cabal Packages

New in version 2015.8.0.

These states manage the installed packages for Haskell using cabal. Note that cabal-install must be installed for these states to be available, so cabal states should include a requisite to a pkg.installed state for the package which provides cabal (`cabal-install` in case of Debian based distributions). Example:

```
.. code-block:: yaml
```

cabal-install: pkg.installed

ShellCheck:

cabal.installed:

- require: - pkg: cabal-install

`salt.states.cabal.`**`installed`**`(`*name*, *pkgs=None*, *user=None*, *install_global=False*, *env=None*`)`

Verify that the given package is installed and is at the correct version (if specified).

```yaml
ShellCheck-0.3.5:
  cabal:
    - installed:
```

name The package to install

user The user to run cabal install with

install_global Install package globally instead of locally

env A list of environment variables to be set prior to execution. The format is the same as the `cmd.run`. state function.

`salt.states.cabal.`**`removed`**`(`*name*, *user=None*, *env=None*`)`

Verify that given package is not installed.

21.19.55 salt.states.ceph module

Manage ceph with salt.

New in version 2016.11.0.

`salt.states.ceph.`**`quorum`**`(`*name*, ***kwargs*`)`

Quorum state

This state checks the mon daemons are in quorum. It does not alter the cluster but can be used in formula as a dependency for many cluster operations.

Example usage in sls file:

```
quorum:
  sesceph.quorum:
    - require:
      - sesceph: mon_running
```

21.19.56 salt.states.chef

Execute Chef client runs

Run chef-client or chef-solo

```
my-chef-run:
  chef.client:
    - override-runlist: 'demo1,demo2'
    - server: 'https://chef.domain.com'

default-chef-run:
  chef.client: []

my-solo-run:
  chef.solo:
    - environment: dev
```

`salt.states.chef.`**`client`**`(`*name*, ***kwargs*`)`

name Unique identifier for the state. Does not affect the Chef run.
server The chef server URL
client_key Set the client key file location
config The configuration file to use
config-file-jail Directory under which config files are allowed to be loaded (no client.rb or knife.rb outside this path will be loaded).
environment Set the Chef Environment on the node
group Group to set privilege to
json-attributes Load attributes from a JSON file or URL
localmode Point chef-client at local repository if True
log_level Set the log level (debug, info, warn, error, fatal)
logfile Set the log file location
node-name The node name for this client
override-runlist Replace current run list with specified items for a single run
pid Set the PID file location, defaults to /tmp/chef-client.pid
run-lock-timeout Set maximum duration to wait for another client run to finish, default is indefinitely.
runlist Permanently replace current run list with specified items
user User to set privilege to
validation_key Set the validation key file location, used for registering new clients

`salt.states.chef.`**`solo`**`(`*name*, ***kwargs*`)`

name Unique identifier for the state. Does not affect the Chef run.
config The configuration file to use
environment Set the Chef Environment on the node
group Group to set privilege to
json-attributes Load attributes from a JSON file or URL

log_level Set the log level (debug, info, warn, error, fatal)
logfile Set the log file location
node-name The node name for this client
override-runlist Replace current run list with specified items for a single run
recipe-url Pull down a remote gzipped tarball of recipes and untar it to the cookbook cache
run-lock-timeout Set maximum duration to wait for another client run to finish, default is indefinitely.
user User to set privilege to

21.19.57 salt.states.chocolatey module

Manage Chocolatey package installs .. versionadded:: 2016.3.0

Note: Chocolatey pulls data from the Chocolatey internet database to determine current versions, find available versions, etc. This is normally a slow operation and may be optimized by specifying a local, smaller chocolatey repo.

`salt.states.chocolatey.`**`installed`**(*name, version=None, source=None, force=False, pre_versions=False, install_args=None, override_args=False, force_x86=False, package_args=None, allow_multiple=False*)

Installs a package if not already installed

Parameters

- **`name`** (*`str`*) – The name of the package to be installed. Required.
- **`version`** (*`str`*) – Install a specific version of the package. Defaults to latest version. If the version is different to the one installed then the specified version will be installed. Default is None.
- **`source`** (*`str`*) – Chocolatey repository (directory, share or remote URL, feed). Defaults to the official Chocolatey feed. Default is None.
- **`force`** (*`bool`*) – Reinstall the current version of an existing package. Do not use with `allow_multiple`. Default is False.
- **`pre_versions`** (*`bool`*) – Include pre-release packages. Default is False.
- **`install_args`** (*`str`*) – Install arguments you want to pass to the installation process, i.e product key or feature list. Default is None.
- **`override_args`** (*`bool`*) – Set to True if you want to override the original install arguments (for the native installer) in the package and use your own. When this is set to False install_args will be appended to the end of the default arguments. Default is False.
- **`force_x86`** (*`bool`*) – Force x86 (32bit) installation on 64 bit systems. Default is False.
- **`package_args`** (*`str`*) – Arguments you want to pass to the package. Default is None.
- **`allow_multiple`** (*`bool`*) – Allow mulitiple versions of the package to be installed. Do not use with `force`. Does not work with all packages. Default is False.

  New in version 2017.7.0.

```
Installsomepackage:
  chocolatey.installed:
    - name: packagename
    - version: '12.04'
    - source: 'mychocolatey/source'
    - force: True
```

`salt.states.chocolatey.`**`uninstalled`**(*name, version=None, uninstall_args=None, override_args=False*)

Uninstalls a package

name The name of the package to be uninstalled

version Uninstalls a specific version of the package. Defaults to latest version installed.

uninstall_args A list of uninstall arguments you want to pass to the uninstallation process i.e product key or feature list

override_args Set to true if you want to override the original uninstall arguments (for the native uninstaller)in the package and use your own. When this is set to False uninstall_args will be appended to the end of the default arguments

```
Removemypackage:
  chocolatey.uninstalled:
    - name: mypackage
    - version: '21.5'
```

`salt.states.chocolatey.`**`upgraded`**(*name*, *version=None*, *source=None*, *force=False*, *pre_versions=False*, *install_args=None*, *override_args=False*, *force_x86=False*, *package_args=None*)

Upgrades a package. Will install the package if not installed.

New in version 2018.3.0.

Parameters

- **name** (*str*) – The name of the package to be installed. Required.
- **version** (*str*) – Install a specific version of the package. Defaults to latest version. If the version is greater than the one installed then the specified version will be installed. Default is `None`.
- **source** (*str*) – Chocolatey repository (directory, share or remote URL, feed). Defaults to the official Chocolatey feed. Default is `None`.
- **force** (*bool*) – `True` will reinstall an existing package with the same version. Default is `False`.
- **pre_versions** (*bool*) – `True` will nclude pre-release packages. Default is `False`.
- **install_args** (*str*) – Install arguments you want to pass to the installation process, i.e product key or feature list. Default is `None`.
- **override_args** (*bool*) – `True` will override the original install arguments (for the native installer) in the package and use those specified in `install_args`. `False` will append install_args to the end of the default arguments. Default is `False`.
- **force_x86** (*bool*) – `True` forces 32bit installation on 64 bit systems. Default is `False`.
- **package_args** (*str*) – Arguments you want to pass to the package. Default is `None`.

```
upgrade_some_package:
  chocolatey.upgraded:
    - name: packagename
    - version: '12.04'
    - source: 'mychocolatey/source'
```

21.19.58 salt.states.chronos_job module

Configure Chronos jobs via a salt proxy.

```
my_job:
  chronos_job.config:
    - config:
```

(continues on next page)

(continued from previous page)

```
        schedule: "R//PT2S"
        command: "echo 'hi'"
        owner: "me@example.com"
```

New in version 2015.8.2.

`salt.states.chronos_job.`**`absent`**(*name*)

Ensure that the chronos job with the given name is not present.

Parameters **`name`** – The app name

Returns A standard Salt changes dictionary

`salt.states.chronos_job.`**`config`**(*name*, *config*)

Ensure that the chronos job with the given name is present and is configured to match the given config values.

Parameters

- **`name`** – The job name
- **`config`** – The configuration to apply (dict)

Returns A standard Salt changes dictionary

21.19.59 salt.states.cimc

A state module to manage Cisco UCS chassis devices.

codeauthor `Spencer Ervin <spencer_ervin@hotmail.com>`

maturity new

depends none

platform unix

About

This state module was designed to handle connections to a Cisco Unified Computing System (UCS) chassis. This module relies on the CIMC proxy module to interface with the device.

See also:

`CIMC Proxy Module`

`salt.states.cimc.`**`hostname`**(*name*, *hostname=None*)

Ensures that the hostname is set to the specified value.

New in version 2019.2.0.

name: The name of the module function to execute.

hostname(str): The hostname of the server.

SLS Example:

```
set_name:
  cimc.hostname:
    - hostname: foobar
```

`salt.states.cimc.`**`logging_levels`**(*name*, *remote=None*, *local=None*)

Ensures that the logging levels are set on the device. The logging levels must match the following options: emergency, alert, critical, error, warning, notice, informational, debug.

New in version 2019.2.0.

name: The name of the module function to execute.

remote(str): The logging level for SYSLOG logs.

local(str): The logging level for the local device.

SLS Example:

```
logging_levels:
  cimc.logging_levels:
    - remote: informational
    - local: notice
```

`salt.states.cimc.`**`ntp`**`(`*name*, *servers*`)`

Ensures that the NTP servers are configured. Servers are provided as an individual string or list format. Only four NTP servers will be reviewed. Any entries past four will be ignored.

name: The name of the module function to execute.

servers(str, list): The IP address or FQDN of the NTP servers.

SLS Example:

```
ntp_configuration_list:
  cimc.ntp:
    - servers:
      - foo.bar.com
      - 10.10.10.10

ntp_configuration_str:
  cimc.ntp:
    - servers: foo.bar.com
```

`salt.states.cimc.`**`power_configuration`**`(`*name*, *policy=None*, *delayType=None*, *delayValue=None*`)`

Ensures that the power configuration is configured on the system. This is only available on some C-Series servers.

New in version 2019.2.0.

name: The name of the module function to execute.

policy(str): The action to be taken when chassis power is restored after an unexpected power loss. This can be one of the following:

> reset: The server is allowed to boot up normally when power is restored. The server can restart immediately or, optionally, after a fixed or random delay.
>
> stay-off: The server remains off until it is manually restarted.
>
> last-state: The server restarts and the system attempts to restore any processes that were running before power was lost.

delayType(str): If the selected policy is reset, the restart can be delayed with this option. This can be one of the following:

> fixed: The server restarts after a fixed delay.
>
> random: The server restarts after a random delay.

delayValue(int): If a fixed delay is selected, once chassis power is restored and the Cisco IMC has finished rebooting, the system waits for the specified number of seconds before restarting the server. Enter an integer between 0 and 240.

SLS Example:

```
reset_power:
  cimc.power_configuration:
    - policy: reset
    - delayType: fixed
    - delayValue: 0

power_off:
  cimc.power_configuration:
    - policy: stay-off
```

`salt.states.cimc.`**`syslog`**`(`*name, primary=None, secondary=None*`)`

Ensures that the syslog servers are set to the specified values. A value of None will be ignored.

name: The name of the module function to execute.

primary(str): The IP address or FQDN of the primary syslog server.

secondary(str): The IP address or FQDN of the secondary syslog server.

SLS Example:

```
syslog_configuration:
  cimc.syslog:
    - primary: 10.10.10.10
    - secondary: foo.bar.com
```

`salt.states.cimc.`**`user`**`(`*name, id=", user=", priv=", password=", status='active'*`)`

Ensures that a user is configured on the device. Due to being unable to verify the user password. This is a forced operation.

New in version 2019.2.0.

name: The name of the module function to execute.

id(int): The user ID slot on the device.

user(str): The username of the user.

priv(str): The privilege level of the user.

password(str): The password of the user.

status(str): The status of the user. Can be either active or inactive.

SLS Example:

```
user_configuration:
  cimc.user:
    - id: 11
    - user: foo
    - priv: admin
    - password: mypassword
    - status: active
```

21.19.60 salt.states.cisconso

State module for Cisco NSO Proxy minions

For documentation on setting up the cisconso proxy minion look in the documentation for *`salt.proxy.cisconso`*.

`salt.states.cisconso.`**`value_present`**`(`*name*, *datastore*, *path*, *config*`)`

Ensure a specific value exists at a given path

Parameters

- **name** (`str`) – The name for this rule
- **datastore** (`DatastoreType` (`str` enum).) – The datastore, e.g. running, operational. One of the NETCONF store IETF types
- **path** (`list`, `str` OR `tuple`) – The device path to set the value at, a list of element names in order, / separated
- **config** (`dict`) – The new value at the given path

Examples:

```
enable pap auth:
  cisconso.config_present:
    - name: enable_pap_auth
    - datastore: running
    - path: devices/device/ex0/config/sys/interfaces/serial/ppp0/authentication
    - config:
        authentication:
            method: pap
            "list-name": foobar
```

21.19.61 salt.states.cloud

Using states instead of maps to deploy clouds

New in version 2014.1.0.

Use this minion to spin up a cloud instance:

```
my-ec2-instance:
  cloud.profile:
    my-ec2-config
```

`salt.states.cloud.`**`absent`**`(`*name*, *onlyif=None*, *unless=None*`)`

Ensure that no instances with the specified names exist.

CAUTION: This is a destructive state, which will search all configured cloud providers for the named instance, and destroy it.

name The name of the instance to destroy
onlyif Do run the state only if is unless succeed
unless Do not run the state at least unless succeed

`salt.states.cloud.`**`present`**`(`*name*, *cloud_provider*, *onlyif=None*, *unless=None*, *opts=None*, ***kwargs*`)`

Spin up a single instance on a cloud provider, using salt-cloud. This state does not take a profile argument; rather, it takes the arguments that would normally be configured as part of the state.

Note that while this function does take any configuration argument that would normally be used to create an instance, it will not verify the state of any of those arguments on an existing instance. Stateful properties of an instance should be configured using their own individual state (i.e., cloud.tagged, cloud.untagged, etc).

name The name of the instance to create
cloud_provider The name of the cloud provider to use
onlyif Do run the state only if is unless succeed
unless Do not run the state at least unless succeed
opts Any extra opts that need to be used

`salt.states.cloud.`**`profile`**`(`*name*, *profile*, *onlyif=None*, *unless=None*, *opts=None*, ***kwargs*`)`

Create a single instance on a cloud provider, using a salt-cloud profile.

Note that while profiles used this function do take any configuration argument that would normally be used to create an instance using a profile, this state will not verify the state of any of those arguments on an existing instance. Stateful properties of an instance should be configured using their own individual state (i.e., cloud.tagged, cloud.untagged, etc).

name The name of the instance to create
profile The name of the cloud profile to use
onlyif Do run the state only if is unless succeed
unless Do not run the state at least unless succeed
kwargs Any profile override or addition
opts Any extra opts that need to be used

`salt.states.cloud.`**`volume_absent`**`(`*name*, *provider=None*, ***kwargs*`)`

Check that a block volume exists.

`salt.states.cloud.`**`volume_attached`**`(`*name*, *server_name*, *provider=None*, ***kwargs*`)`

Check if a block volume is attached.

`salt.states.cloud.`**`volume_detached`**`(`*name*, *server_name=None*, *provider=None*, ***kwargs*`)`

Check if a block volume is attached.

Returns True if server or Volume do not exist.

`salt.states.cloud.`**`volume_present`**`(`*name*, *provider=None*, ***kwargs*`)`

Check that a block volume exists.

21.19.62 salt.states.cmd

Execution of arbitrary commands

The cmd state module manages the enforcement of executed commands, this state can tell a command to run under certain circumstances.

A simple example to execute a command:

```
# Store the current date in a file
'date > /tmp/salt-run':
  cmd.run
```

Only run if another execution failed, in this case truncate syslog if there is no disk space:

```
'> /var/log/messages/:
  cmd.run:
    - unless: echo 'foo' > /tmp/.test && rm -f /tmp/.test
```

Only run if the file specified by `creates` does not exist, in this case touch /tmp/foo if it does not exist:

```
touch /tmp/foo:
  cmd.run:
    - creates: /tmp/foo
```

`creates` also accepts a list of files, in which case this state will run if **any** of the files do not exist:

```
"echo 'foo' | tee /tmp/bar > /tmp/baz":
  cmd.run:
```

(continues on next page)

(continued from previous page)

```
    - creates:
      - /tmp/bar
      - /tmp/baz
```

Note: The `creates` option was added to version 2014.7.0

Sometimes when running a command that starts up a daemon, the init script doesn't return properly which causes Salt to wait indefinitely for a response. In situations like this try the following:

```
run_installer:
  cmd.run:
    - name: /tmp/installer.bin > /dev/null 2>&1
```

Salt determines whether the `cmd` state is successfully enforced based on the exit code returned by the command. If the command returns a zero exit code, then salt determines that the state was successfully enforced. If the script returns a non-zero exit code, then salt determines that it failed to successfully enforce the state. If a command returns a non-zero exit code but you wish to treat this as a success, then you must place the command in a script and explicitly set the exit code of the script to zero.

Please note that the success or failure of the state is not affected by whether a state change occurred nor the stateful argument.

When executing a command or script, the state (i.e., changed or not) of the command is unknown to Salt's state system. Therefore, by default, the `cmd` state assumes that any command execution results in a changed state.

This means that if a `cmd` state is watched by another state then the state that's watching will always be executed due to the *changed* state in the `cmd` state.

Using the "Stateful" Argument

Many state functions in this module now also accept a `stateful` argument. If `stateful` is specified to be true then it is assumed that the command or script will determine its own state and communicate it back by following a simple protocol described below:

1. **If there's nothing in the stdout of the command, then assume no changes**. Otherwise, the stdout must be either in JSON or its *last* non-empty line must be a string of key=value pairs delimited by spaces (no spaces on either side of =).
2. **If it's JSON then it must be a JSON object (e.g., {})**. If it's key=value pairs then quoting may be used to include spaces. (Python's shlex module is used to parse the key=value string)

   Two special keys or attributes are recognized in the output:

   ```
   changed: bool (i.e., 'yes', 'no', 'true', 'false', case-insensitive)
   comment: str  (i.e., any string)
   ```

   So, only if `changed` is `True` then assume the command execution has changed the state, and any other key values or attributes in the output will be set as part of the changes.
3. **If there's a comment then it will be used as the comment of the state**.

   Here's an example of how one might write a shell script for use with a stateful command:

```
#!/bin/bash
#
echo "Working hard..."

# writing the state line
echo  # an empty line here so the next line will be the last.
echo "changed=yes comment='something has changed' whatever=123"
```

And an example SLS file using this module:

```
Run myscript:
  cmd.run:
    - name: /path/to/myscript
    - cwd: /
    - stateful: True

Run only if myscript changed something:
  cmd.run:
    - name: echo hello
    - cwd: /
    - onchanges:
        - cmd: Run myscript
```

Note that if the second `cmd.run` state also specifies `stateful: True` it can then be watched by some other states as well.

4. **The stateful argument can optionally include a test_name parameter.**

   This is used to specify a command to run in test mode. This command should return stateful data for changes that would be made by the command in the name parameter.

   New in version 2015.2.0.

```
Run myscript:
  cmd.run:
    - name: /path/to/myscript
    - cwd: /
    - stateful:
      - test_name: /path/to/myscript test

Run masterscript:
  cmd.script:
    - name: masterscript
    - source: salt://path/to/masterscript
    - cwd: /
    - stateful:
      - test_name: masterscript test
```

Should I use `cmd.run` or `cmd.wait`?

Note: Use `cmd.run` together with *onchanges* instead of `cmd.wait`.

These two states are often confused. The important thing to remember about them is that `cmd.run` states are run each time the SLS file that contains them is applied. If it is more desirable to have a command that only runs after some other state changes, then *cmd.wait* does just that. *cmd.wait* is designed to *watch* other states, and is executed when the state it is watching changes. Example:

```
/usr/local/bin/postinstall.sh:
  cmd.wait:
    - watch:
      - pkg: mycustompkg
  file.managed:
    - source: salt://utils/scripts/postinstall.sh

mycustompkg:
  pkg.installed:
    - require:
      - file: /usr/local/bin/postinstall.sh
```

`cmd.wait` itself do not do anything; all functionality is inside its `mod_watch` function, which is called by `watch` on changes.

The preferred format is using the *onchanges Requisite*, which works on `cmd.run` as well as on any other state. The example would then look as follows:

```
/usr/local/bin/postinstall.sh:
  cmd.run:
    - onchanges:
      - pkg: mycustompkg
  file.managed:
    - source: salt://utils/scripts/postinstall.sh

mycustompkg:
  pkg.installed:
    - require:
      - file: /usr/local/bin/postinstall.sh
```

How do I create an environment from a pillar map?

The map that comes from a pillar can be directly consumed by the env option! To use it, one may pass it like this. Example:

```
printenv:
  cmd.run:
    - env: {{ salt['pillar.get']('example:key', {}) }}
```

`salt.states.cmd.`**`call`**(*name, func, args=(), kws=None, onlyif=None, unless=None, creates=None, output_loglevel='debug', hide_output=False, use_vt=False, **kwargs*)

Invoke a pre-defined Python function with arguments specified in the state declaration. This function is mainly used by the *salt.renderers.pydsl* renderer.

The interpretation of `onlyif` and `unless` arguments are identical to those of *cmd.run*, and all other arguments(`cwd`, `runas`, ...) allowed by *cmd.run* are allowed here, except that their effects apply only to the commands specified in *onlyif* and *unless* rather than to the function to be invoked.

In addition, the `stateful` argument has no effects here.

The return value of the invoked function will be interpreted as follows.

If it's a dictionary then it will be passed through to the state system, which expects it to have the usual structure returned by any salt state function.

Otherwise, the return value (denoted as `result` in the code below) is expected to be a JSON serializable object, and this dictionary is returned:

```
{
    'name': name
    'changes': {'retval': result},
    'result': True if result is None else bool(result),
    'comment': result if isinstance(result, six.string_types) else ''
}
```

`salt.states.cmd.`**`mod_run_check`**(*cmd_kwargs*, *onlyif*, *unless*, *creates*)

Execute the onlyif and unless logic. Return a result dict if: * onlyif failed (onlyif != 0) * unless succeeded (unless == 0) else return True

`salt.states.cmd.`**`mod_watch`**(*name*, ***kwargs*)

Execute a cmd function based on a watch call

Note: This state exists to support special handling of the `watch` *requisite*. It should not be called directly.

Parameters for this function should be set by the state being triggered.

`salt.states.cmd.`**`run`**(*name*, *onlyif=None*, *unless=None*, *creates=None*, *cwd=None*, *root=None*, *runas=None*, *shell=None*, *env=None*, *prepend_path=None*, *stateful=False*, *umask=None*, *output_loglevel='debug'*, *hide_output=False*, *timeout=None*, *ignore_timeout=False*, *use_vt=False*, *success_retcodes=None*, ***kwargs*)

Run a command if certain circumstances are met. Use `cmd.wait` if you want to use the `watch` requisite.

name The command to execute, remember that the command will execute with the path and permissions of the salt-minion.

onlyif A command to run as a check, run the named command only if the command passed to the `onlyif` option returns a zero exit status

unless A command to run as a check, only run the named command if the command passed to the `unless` option returns a non-zero exit status

cwd The current working directory to execute the command in, defaults to /root

root Path to the root of the jail to use. If this parameter is set, the command will run inside a chroot

runas The user name to run the command as

shell The shell to use for execution, defaults to the shell grain

env A list of environment variables to be set prior to execution. Example:

```
script-foo:
  cmd.run:
    - env:
      - BATCH: 'yes'
```

Warning: The above illustrates a common PyYAML pitfall, that **yes**, **no**, **on**, **off**, **true**, and **false** are all loaded as boolean `True` and `False` values, and must be enclosed in quotes to be used as strings. More info on this (and other) PyYAML idiosyncrasies can be found *here*.

Variables as values are not evaluated. So $PATH in the following example is a literal '$PATH':

```
script-bar:
  cmd.run:
    - env: "PATH=/some/path:$PATH"
```

One can still use the existing $PATH by using a bit of Jinja:

```
{% set current_path = salt['environ.get']('PATH', '/bin:/usr/bin') %}

mycommand:
  cmd.run:
    - name: ls -l /
    - env:
      - PATH: {{ [current_path, '/my/special/bin']|join(':') }}
```

prepend_path $PATH segment to prepend (trailing ':' not necessary) to $PATH. This is an easier alternative to the Jinja workaround.

New in version 2018.3.0.

stateful The command being executed is expected to return data about executing a state. For more information, see the *Using the "Stateful" Argument* section.

umask The umask (in octal) to use when running the command.

output_loglevel [debug] Control the loglevel at which the output from the command is logged to the minion log.

Note: The command being run will still be logged at the `debug` loglevel regardless, unless `quiet` is used for this value.

hide_output [False] Suppress stdout and stderr in the state's results.

Note: This is separate from `output_loglevel`, which only handles how Salt logs to the minion log.

New in version 2018.3.0.

timeout If the command has not terminated after timeout seconds, send the subprocess sigterm, and if sigterm is ignored, follow up with sigkill

ignore_timeout Ignore the timeout of commands, which is useful for running nohup processes.

New in version 2015.8.0.

creates Only run if the file specified by `creates` do not exist. If you specify a list of files then this state will only run if **any** of the files do not exist.

New in version 2014.7.0.

use_vt [False] Use VT utils (saltstack) to stream the command output more interactively to the console and the logs. This is experimental.

bg [False] If `True`, run command in background and do not await or deliver its results.

New in version 2016.3.6.

success_retcodes: This parameter will be allow a list of
non-zero return codes that should be considered a success. If the return code returned from the run matches any in the provided list, the return code will be overridden with zero.

New in version 2019.2.0.

Note: cmd.run supports the usage of `reload_modules`. This functionality allows you to force Salt to reload all modules. You should only use `reload_modules` if your cmd.run does some sort of installation (such as `pip`), if you do not reload the modules future items in your state which rely on the software being installed will fail.

```
getpip:
  cmd.run:
    - name: /usr/bin/python /usr/local/sbin/get-pip.py
    - unless: which pip
    - require:
```

(continues on next page)

(continued from previous page)

```
      - pkg: python
      - file: /usr/local/sbin/get-pip.py
    - reload_modules: True
```

`salt.states.cmd.script`(*name, source=None, template=None, onlyif=None, unless=None, creates=None, cwd=None, runas=None, shell=None, env=None, stateful=False, umask=None, timeout=None, use_vt=False, output_loglevel='debug', hide_output=False, defaults=None, context=None, success_retcodes=None, **kwargs*)

Download a script and execute it with specified arguments.

source The location of the script to download. If the file is located on the master in the directory named spam, and is called eggs, the source string is salt://spam/eggs

template If this setting is applied then the named templating engine will be used to render the downloaded file. Currently jinja, mako, and wempy are supported

name Either "cmd arg1 arg2 arg3..." (cmd is not used) or a source "salt://...".

onlyif Run the named command only if the command passed to the `onlyif` option returns true

unless Run the named command only if the command passed to the `unless` option returns false

cwd The current working directory to execute the command in, defaults to /root

runas The name of the user to run the command as

shell The shell to use for execution. The default is set in grains['shell']

env A list of environment variables to be set prior to execution. Example:

```
salt://scripts/foo.sh:
  cmd.script:
    - env:
      - BATCH: 'yes'
```

> **Warning:** The above illustrates a common PyYAML pitfall, that **yes**, **no**, **on**, **off**, **true**, and **false** are all loaded as boolean `True` and `False` values, and must be enclosed in quotes to be used as strings. More info on this (and other) PyYAML idiosyncrasies can be found *here*.

Variables as values are not evaluated. So $PATH in the following example is a literal '$PATH':

```
salt://scripts/bar.sh:
  cmd.script:
    - env: "PATH=/some/path:$PATH"
```

One can still use the existing $PATH by using a bit of Jinja:

```
{% set current_path = salt['environ.get']('PATH', '/bin:/usr/bin') %}

mycommand:
  cmd.run:
    - name: ls -l /
    - env:
      - PATH: {{ [current_path, '/my/special/bin']|join(':') }}
```

saltenv `[base]` The Salt environment to use

umask The umask (in octal) to use when running the command.

stateful The command being executed is expected to return data about executing a state. For more information, see the *Using the "Stateful" Argument* section.

timeout If the command has not terminated after timeout seconds, send the subprocess sigterm, and if sigterm is ignored, follow up with sigkill

args String of command line args to pass to the script. Only used if no args are specified as part of the *name* argument. To pass a string containing spaces in YAML, you will need to doubly-quote it: "arg1 'arg two' arg3"

creates Only run if the file specified by `creates` do not exist. If you specify a list of files then this state will only run if **any** of the files do not exist.

New in version 2014.7.0.

use_vt Use VT utils (saltstack) to stream the command output more interactively to the console and the logs. This is experimental.

context New in version 2016.3.0.

Overrides default context variables passed to the template.

defaults New in version 2016.3.0.

Default context passed to the template.

output_loglevel [debug] Control the loglevel at which the output from the command is logged to the minion log.

> **Note:** The command being run will still be logged at the `debug` loglevel regardless, unless `quiet` is used for this value.

hide_output [False] Suppress stdout and stderr in the state's results.

> **Note:** This is separate from `output_loglevel`, which only handles how Salt logs to the minion log.

New in version 2018.3.0.

success_retcodes: This parameter will be allow a list of

non-zero return codes that should be considered a success. If the return code returned from the run matches any in the provided list, the return code will be overridden with zero.

New in version 2019.2.0.

`salt.states.cmd.wait`(*name*, *onlyif=None*, *unless=None*, *creates=None*, *cwd=None*, *root=None*, *runas=None*, *shell=None*, *env=()*, *stateful=False*, *umask=None*, *output_loglevel='debug'*, *hide_output=False*, *use_vt=False*, *success_retcodes=None*, ***kwargs*)

Run the given command only if the watch statement calls it.

> **Note:** Use *`cmd.run`* together with `onchanges` instead of *`cmd.wait`*.

name The command to execute, remember that the command will execute with the path and permissions of the salt-minion.

onlyif A command to run as a check, run the named command only if the command passed to the `onlyif` option returns true

unless A command to run as a check, only run the named command if the command passed to the `unless` option returns false

cwd The current working directory to execute the command in, defaults to /root

root Path to the root of the jail to use. If this parameter is set, the command will run inside a chroot

runas The user name to run the command as

shell The shell to use for execution, defaults to /bin/sh

env A list of environment variables to be set prior to execution. Example:

```
script-foo:
  cmd.wait:
```

(continues on next page)

(continued from previous page)

```
    - env:
      - BATCH: 'yes'
```

Warning: The above illustrates a common PyYAML pitfall, that **yes**, **no**, **on**, **off**, **true**, and **false** are all loaded as boolean `True` and `False` values, and must be enclosed in quotes to be used as strings. More info on this (and other) PyYAML idiosyncrasies can be found *here*.

Variables as values are not evaluated. So $PATH in the following example is a literal '$PATH':

```
script-bar:
  cmd.wait:
    - env: "PATH=/some/path:$PATH"
```

One can still use the existing $PATH by using a bit of Jinja:

```
{% set current_path = salt['environ.get']('PATH', '/bin:/usr/bin') %}

mycommand:
  cmd.run:
    - name: ls -l /
    - env:
      - PATH: {{ [current_path, '/my/special/bin']|join(':') }}
```

umask The umask (in octal) to use when running the command.

stateful The command being executed is expected to return data about executing a state. For more information, see the *Using the "Stateful" Argument* section.

creates Only run if the file specified by `creates` do not exist. If you specify a list of files then this state will only run if **any** of the files do not exist.

New in version 2014.7.0.

output_loglevel [debug] Control the loglevel at which the output from the command is logged to the minion log.

Note: The command being run will still be logged at the `debug` loglevel regardless, unless `quiet` is used for this value.

hide_output [False] Suppress stdout and stderr in the state's results.

Note: This is separate from `output_loglevel`, which only handles how Salt logs to the minion log.

New in version 2018.3.0.

use_vt Use VT utils (saltstack) to stream the command output more interactively to the console and the logs. This is experimental.

success_retcodes: This parameter will be allow a list of

non-zero return codes that should be considered a success. If the return code returned from the run matches any in the provided list, the return code will be overridden with zero.

New in version 2019.2.0.

`salt.states.cmd.`**`wait_call`**(*name, func, args=(), kws=None, onlyif=None, unless=None, creates=None, stateful=False, use_vt=False, output_loglevel='debug', hide_output=False, **kwargs*) `salt.states.cmd.`**`wait_script`**`(`*name, source=None, template=None, onlyif=None, unless=None, cwd=None, runas=None, shell=None, env=None, stateful=False, umask=None, use_vt=False, output_loglevel='debug', hide_output=False, **kwargs*`)`

Download a script from a remote source and execute it only if a watch statement calls it.

source The source script being downloaded to the minion, this source script is hosted on the salt master server. If the file is located on the master in the directory named spam, and is called eggs, the source string is salt://spam/eggs

template If this setting is applied then the named templating engine will be used to render the downloaded file, currently jinja, mako, and wempy are supported

name The command to execute, remember that the command will execute with the path and permissions of the salt-minion.

onlyif A command to run as a check, run the named command only if the command passed to the `onlyif` option returns true

unless A command to run as a check, only run the named command if the command passed to the `unless` option returns false

cwd The current working directory to execute the command in, defaults to /root

runas The user name to run the command as

shell The shell to use for execution, defaults to the shell grain

env A list of environment variables to be set prior to execution. Example:

```
salt://scripts/foo.sh:
  cmd.wait_script:
    - env:
      - BATCH: 'yes'
```

> **Warning:** The above illustrates a common PyYAML pitfall, that **yes**, **no**, **on**, **off**, **true**, and **false** are all loaded as boolean `True` and `False` values, and must be enclosed in quotes to be used as strings. More info on this (and other) PyYAML idiosyncrasies can be found *here*.

Variables as values are not evaluated. So $PATH in the following example is a literal '$PATH':

```
salt://scripts/bar.sh:
  cmd.wait_script:
    - env: "PATH=/some/path:$PATH"
```

One can still use the existing $PATH by using a bit of Jinja:

```
{% set current_path = salt['environ.get']('PATH', '/bin:/usr/bin') %}

mycommand:
  cmd.run:
    - name: ls -l /
    - env:
      - PATH: {{ [current_path, '/my/special/bin']|join(':') }}
```

umask The umask (in octal) to use when running the command.

stateful The command being executed is expected to return data about executing a state. For more information, see the *Using the "Stateful" Argument* section.

use_vt Use VT utils (saltstack) to stream the command output more interactively to the console and the logs. This is experimental.

output_loglevel [debug] Control the loglevel at which the output from the command is logged to the minion log.

Note: The command being run will still be logged at the `debug` loglevel regardless, unless `quiet` is used for this value.

---

hide_output [False] Suppress stdout and stderr in the state's results.

Note: This is separate from `output_loglevel`, which only handles how Salt logs to the minion log.

New in version 2018.3.0.

success_retcodes: This parameter will be allow a list of
non-zero return codes that should be considered a success. If the return code returned from the run matches any in the provided list, the return code will be overridden with zero.

New in version 2019.2.0.

21.19.63 salt.states.composer

Installation of Composer Packages

These states manage the installed packages for composer for PHP. Note that either composer is installed and accessible via a bin directory or you can pass the location of composer in the state.

```
get-composer:
  cmd.run:
    - name: 'CURL=`which curl`; $CURL -sS https://getcomposer.org/installer | php'
    - unless: test -f /usr/local/bin/composer
    - cwd: /root/

install-composer:
  cmd.wait:
    - name: mv /root/composer.phar /usr/local/bin/composer
    - cwd: /root/
    - watch:
      - cmd: get-composer

/path/to/project:
  composer.installed:
    - no_dev: true
    - require:
      - cmd: install-composer


# Without composer installed in your PATH
# Note: composer.phar must be executable for state to work properly
/path/to/project:
  composer.installed:
    - composer: /path/to/composer.phar
    - php: /usr/local/bin/php
    - no_dev: true
```

`salt.states.composer.`**`installed`**(*name, composer=None, php=None, user=None, prefer_source=None, prefer_dist=None, no_scripts=None, no_plugins=None, optimize=None, no_dev=None, quiet=False, composer_home='/root', always_check=True, env=None*)

Verify that the correct versions of composer dependencies are present.

name Directory location of the `composer.json` file.

composer Location of the `composer.phar` file. If not set composer will just execute `composer` as if it is installed globally. (i.e. `/path/to/composer.phar`)

php Location of the php executable to use with composer. (i.e. `/usr/bin/php`)

user Which system user to run composer as.

New in version 2014.1.4.

prefer_source `--prefer-source` option of composer.

prefer_dist `--prefer-dist` option of composer.

no_scripts `--no-scripts` option of composer.

no_plugins `--no-plugins` option of composer.

optimize `--optimize-autoloader` option of composer. Recommended for production.

no_dev `--no-dev` option for composer. Recommended for production.

quiet `--quiet` option for composer. Whether or not to return output from composer.

composer_home `$COMPOSER_HOME` environment variable

always_check If `True`, *always* run `composer install` in the directory. This is the default behavior. If `False`, only run `composer install` if there is no vendor directory present.

env A list of environment variables to be set prior to execution.

`salt.states.composer.`**`update`**`(`*name, composer=None, php=None, user=None, prefer_source=None, prefer_dist=None, no_scripts=None, no_plugins=None, optimize=None, no_dev=None, quiet=False, composer_home='/root', env=None*`)`

Composer update the directory to ensure we have the latest versions of all project dependencies.

name Directory location of the `composer.json` file.

composer Location of the `composer.phar` file. If not set composer will just execute `composer` as if it is installed globally. (i.e. /path/to/composer.phar)

php Location of the php executable to use with composer. (i.e. `/usr/bin/php`)

user Which system user to run composer as.

New in version 2014.1.4.

prefer_source `--prefer-source` option of composer.

prefer_dist `--prefer-dist` option of composer.

no_scripts `--no-scripts` option of composer.

no_plugins `--no-plugins` option of composer.

optimize `--optimize-autoloader` option of composer. Recommended for production.

no_dev `--no-dev` option for composer. Recommended for production.

quiet `--quiet` option for composer. Whether or not to return output from composer.

composer_home `$COMPOSER_HOME` environment variable

env A list of environment variables to be set prior to execution.

21.19.64 salt.states.cron

Management of cron, the Unix command scheduler

Cron declarations require a number of parameters. The following are the parameters used by Salt to define the various timing values for a cron job:

- `minute`
- `hour`
- `daymonth`
- `month`
- `dayweek` (0 to 6 are Sunday through Saturday, 7 can also be used for Sunday)

Warning: Any timing arguments not specified take a value of `*`. This means that setting `hour` to `5`, while not defining the `minute` param, will result in Salt adding a job that will execute every minute between 5 and 6 A.M.!

Additionally, the default user for these states is `root`. Therefore, if the cron job is for another user, it is necessary to specify that user with the `user` parameter.

A long time ago (before 2014.2), when making changes to an existing cron job, the name declaration is the parameter used to uniquely identify the job, so if an existing cron that looks like this:

```
date > /tmp/crontest:
  cron.present:
    - user: root
    - minute: 5
```

Is changed to this:

```
date > /tmp/crontest:
  cron.present:
    - user: root
    - minute: 7
    - hour: 2
```

Then the existing cron will be updated, but if the cron command is changed, then a new cron job will be added to the user's crontab.

The current behavior is still relying on that mechanism, but you can also specify an identifier to identify your crontabs:

```
date > /tmp/crontest:
  cron.present:
    - identifier: SUPERCRON
    - user: root
    - minute: 7
    - hour: 2
```

New in version 2014.1.2.

And, some months later, you modify it:

```
superscript > /tmp/crontest:
  cron.present:
    - identifier: SUPERCRON
    - user: root
    - minute: 3
    - hour: 4
```

New in version 2014.1.2.

The old **date > /tmp/crontest** will be replaced by **superscript > /tmp/crontest**.

Additionally, Salt also supports running a cron every `x minutes` very similarly to the Unix convention of using `*/5` to have a job run every five minutes. In Salt, this looks like:

```
date > /tmp/crontest:
  cron.present:
    - user: root
    - minute: '*/5'
```

The job will now run every 5 minutes.

Additionally, the temporal parameters (minute, hour, etc.) can be randomized by using `random` instead of using a specific value. For example, by using the `random` keyword in the `minute` parameter of a cron state, the same cron job can be pushed to hundreds or thousands of hosts, and they would each use a randomly-generated minute. This can be helpful when the cron job accesses a network resource, and it is not desirable for all hosts to run the job concurrently.

```
/path/to/cron/script:
  cron.present:
    - user: root
    - minute: random
    - hour: 2
```

New in version 0.16.0.

Since Salt assumes a value of `*` for unspecified temporal parameters, adding a parameter to the state and setting it to `random` will change that value from `*` to a randomized numeric value. However, if that field in the cron entry on the minion already contains a numeric value, then using the `random` keyword will not modify it.

Added the opportunity to set a job with a special keyword like '@reboot' or '@hourly'. Quotes must be used, otherwise PyYAML will strip the '@' sign.

```
/path/to/cron/script:
  cron.present:
    - user: root
    - special: '@hourly'
```

The script will be executed every reboot if cron daemon support this option.

```
/path/to/cron/otherscript:
  cron.absent:
    - user: root
    - special: '@daily'
```

This counter part definition will ensure than a job with a special keyword is not set.

`salt.states.cron.`**`absent`**(*name*, *user='root'*, *identifier=False*, *special=None*, ***kwargs*)

Verifies that the specified cron job is absent for the specified user; only the name is matched when removing a cron job.

name The command that should be absent in the user crontab.

user The name of the user whose crontab needs to be modified, defaults to the root user

identifier Custom-defined identifier for tracking the cron line for future crontab edits. This defaults to the state name

special The special keyword used in the job (eg. @reboot, @hourly…). Quotes must be used, otherwise PyYAML will strip the '@' sign.

`salt.states.cron.`**`env_absent`**(*name*, *user='root'*)

Verifies that the specified environment variable is absent from the crontab for the specified user

name The name of the environment variable to remove from the user crontab

user The name of the user whose crontab needs to be modified, defaults to the root user

`salt.states.cron.`**`env_present`**(*name*, *value=None*, *user='root'*)

Verifies that the specified environment variable is present in the crontab for the specified user.

name The name of the environment variable to set in the user crontab

user The name of the user whose crontab needs to be modified, defaults to the root user

value The value to set for the given environment variable `salt.states.cron.`**`file`**(*name*, *source_hash=''*, *source_hash_name=None*, *user='root'*, *template=None*, *context=None*, *replace=True*, *defaults=None*, *backup=''*, ***kwargs*)

Provides file.managed-like functionality (templating, etc.) for a pre-made crontab file, to be assigned to a given user.

name The source file to be used as the crontab. This source file can be hosted on either the salt master server, or on an HTTP or FTP server. For files hosted on the salt file server, if the file is located on the master in the directory named spam, and is called eggs, the source string is `salt://spam/eggs`

If the file is hosted on a HTTP or FTP server then the source_hash argument is also required

source_hash This can be either a file which contains a source hash string for the source, or a source hash string. The source hash string is the hash algorithm followed by the hash of the file: `md5=e138491e9d5b97023cea823fe17bac22`

source_hash_name When `source_hash` refers to a hash file, Salt will try to find the correct hash by matching the filename/URI associated with that hash. By default, Salt will look for the filename being managed. When managing a file at path `/tmp/foo.txt`, then the following line in a hash file would match:

```
acbd18db4cc2f85cedef654fccc4a4d8    foo.txt
```

However, sometimes a hash file will include multiple similar paths:

```
37b51d194a7513e45b56f6524f2d51f2    ./dir1/foo.txt
acbd18db4cc2f85cedef654fccc4a4d8    ./dir2/foo.txt
73feffa4b7f6bb68e44cf984c85f6e88    ./dir3/foo.txt
```

In cases like this, Salt may match the incorrect hash. This argument can be used to tell Salt which filename to match, to ensure that the correct hash is identified. For example:

```yaml
foo_crontab:
  cron.file:
    - name: https://mydomain.tld/dir2/foo.txt
    - source_hash: https://mydomain.tld/hashes
    - source_hash_name: ./dir2/foo.txt
```

Note: This argument must contain the full filename entry from the checksum file, as this argument is meant to disambiguate matches for multiple files that have the same basename. So, in the example above, simply using `foo.txt` would not match.

New in version 2016.3.5.

user The user to whom the crontab should be assigned. This defaults to root.

template If this setting is applied then the named templating engine will be used to render the downloaded file. Currently, jinja and mako are supported.

context Overrides default context variables passed to the template.

replace If the crontab should be replaced, if False then this command will be ignored if a crontab exists for the specified user. Default is True.

defaults Default context passed to the template.

backup Overrides the default backup mode for the user's crontab.

`salt.states.cron.`**`present`**(*name*, *user='root'*, *minute='*'*, *hour='*'*, *daymonth='*'*, *month='*'*, *dayweek='*'*, *comment=None*, *commented=False*, *identifier=False*, *special=None*)

Verifies that the specified cron job is present for the specified user. It is recommended to use *identifier*. Otherwise the cron job is installed twice if you change the name. For more advanced information about what exactly can be set in the cron timing parameters, check your cron system's documentation. Most Unix-like systems' cron documentation can be found via the crontab man page: `man 5 crontab`.

name The command that should be executed by the cron job.

user The name of the user whose crontab needs to be modified, defaults to the root user

minute The information to be set into the minute section, this can be any string supported by your cron system's the minute field. Default is `*`

hour The information to be set in the hour section. Default is `*`

daymonth The information to be set in the day of month section. Default is `*`

month The information to be set in the month section. Default is `*`

dayweek The information to be set in the day of week section. Default is `*`

comment User comment to be added on line previous the cron job

commented The cron job is set commented (prefixed with `#DISABLED#`). Defaults to False.

New in version 2016.3.0.

identifier Custom-defined identifier for tracking the cron line for future crontab edits. This defaults to the state name

special A special keyword to specify periodicity (eg. @reboot, @hourly…). Quotes must be used, otherwise PyYAML will strip the '@' sign.

New in version 2016.3.0.

21.19.65 salt.states.csf module

CSF Ip tables management

depends

- csf utility

configuration See http://download.configserver.com/csf/install.txt for setup instructions.

```
Simply allow/deny rules:
  csf.rule_present:
    ip: 1.2.3.4
    method: allow
```

`salt.states.csf.`**`nics_skip`**(*name*, *nics*, *ipv6*)

Alias for *`csf.nics_skipped`*

`salt.states.csf.`**`nics_skipped`**(*name*, *nics*, *ipv6=False*)

name Meaningless arg, but required for state.

nics A list of nics to skip.

ipv6 Boolean. Set to true if you want to skip the ipv6 interface. Default false (ipv4).

`salt.states.csf.`**`option_present`**(*name*, *value*, *reload=False*)

Ensure the state of a particular option/setting in csf.

name The option name in csf.conf

value The value it should be set to.

reload Boolean. If set to true, csf will be reloaded after.

`salt.states.csf.`**`ports_open`**(*name*, *ports*, *proto='tcp'*, *direction='in'*)

Ensure ports are open for a protocol, in a direction. e.g. - proto='tcp', direction='in' would set the values for TCP_IN in the csf.conf file.

ports A list of ports that should be open.

proto The protocol. May be one of 'tcp', 'udp', 'tcp6', or 'udp6'.

direction Choose 'in', 'out', or both to indicate the port should be opened for inbound traffic, outbound traffic, or both.

`salt.states.csf.`**`rule_absent`**(*name*, *method*, *port=None*, *proto='tcp'*, *direction='in'*, *port_origin='d'*, *ip_origin='s'*, *ttl=None*, *reload=False*)

Ensure iptable is not present.

name The ip address or CIDR for the rule.
method The type of rule. Either 'allow' or 'deny'.
port Optional port to be open or closed for the iptables rule.
proto The protocol. Either 'tcp', 'udp'. Only applicable if port is specified.
direction The diretion of traffic to apply the rule to. Either 'in', or 'out'. Only applicable if port is specified.
port_origin Specifies either the source or destination port is relevant for this rule. Only applicable if port is specified. Either 's', or 'd'.
ip_origin Specifies whether the ip in this rule refers to the source or destination ip. Either 's', or 'd'. Only applicable if port is specified.
ttl How long the rule should exist. If supplied, *csf.tempallow()* or csf.tempdeny()‘ are used.
reload Reload the csf service after applying this rule. Default false.

`salt.states.csf.`**`rule_present`**(*name*, *method*, *port=None*, *proto='tcp'*, *direction='in'*, *port_origin='d'*, *ip_origin='s'*, *ttl=None*, *comment=''*, *reload=False*)

Ensure iptable rule exists.
name The ip address or CIDR for the rule.
method The type of rule. Either 'allow' or 'deny'.
port Optional port to be open or closed for the iptables rule.
proto The protocol. Either 'tcp', or 'udp'. Only applicable if port is specified.
direction The diretion of traffic to apply the rule to. Either 'in', or 'out'. Only applicable if port is specified.
port_origin Specifies either the source or destination port is relevant for this rule. Only applicable if port is specified. Either 's', or 'd'.
ip_origin Specifies whether the ip in this rule refers to the source or destination ip. Either 's', or 'd'. Only applicable if port is specified.
ttl How long the rule should exist. If supplied, *csf.tempallow()* or csf.tempdeny()‘ are used.
comment An optional comment to appear after the rule as a #comment .
reload Reload the csf service after applying this rule. Default false.

`salt.states.csf.`**`testing_off`**(*name*, *reload=False*)

Ensure testing mode is enabled in csf.
reload Reload CSF after changing the testing status. Default false.

`salt.states.csf.`**`testing_on`**(*name*, *reload=False*)

Ensure testing mode is enabled in csf.
reload Reload CSF after changing the testing status. Default false.

21.19.66 salt.states.cyg

Installation of Cygwin packages.

A state module to manage cygwin packages. Packages can be installed or removed.

```
dos2unix:
  cyg.installed
```

class `salt.states.cyg.`**`DictDiffer`**(*current_dict*, *past_dict*)

Calculate the difference between two dictionaries.

(1) items added
(2) items removed
(3) keys same in both but changed values
(4) keys same in both and unchanged values

`added`()

Return a set of additions to past_dict.

`changed`()

Return a set of the keys with changed values.

removed()
Return a set of things removed from past_dict.

same()
True if the two dicts are the same.

unchanged()
Return a set of the keys with unchanged values.

`salt.states.cyg.`**installed**(*name*, *cyg_arch='x86_64'*, *mirrors=None*)
Make sure that a package is installed.

name The name of the package to install

cyg_arch [x86_64] The cygwin architecture to install the package into. Current options are x86 and x86_64

mirrors [None] List of mirrors to check. None will use a default mirror (kernel.org)

CLI Example:

```
rsync:
  cyg.installed:
    - mirrors:
      - http://mirror/without/public/key: ""
      - http://mirror/with/public/key: http://url/of/public/key
```

`salt.states.cyg.`**removed**(*name*, *cyg_arch='x86_64'*, *mirrors=None*)
Make sure that a package is not installed.

name The name of the package to uninstall

cyg_arch [x86_64] The cygwin architecture to remove the package from. Current options are x86 and x86_64

mirrors [None] List of mirrors to check. None will use a default mirror (kernel.org)

CLI Example:

```
rsync:
  cyg.removed:
    - mirrors:
      - http://mirror/without/public/key: ""
      - http://mirror/with/public/key: http://url/of/public/key
```

`salt.states.cyg.`**updated**(*name=None*, *cyg_arch='x86_64'*, *mirrors=None*)
Make sure all packages are up to date.

name [None] No affect, salt fails poorly without the arg available

cyg_arch [x86_64] The cygwin architecture to update. Current options are x86 and x86_64

mirrors [None] List of mirrors to check. None will use a default mirror (kernel.org)

CLI Example:

```
rsync:
  cyg.updated:
    - mirrors:
      - http://mirror/without/public/key: ""
      - http://mirror/with/public/key: http://url/of/public/key
```

21.19.67 salt.states.ddns

Dynamic DNS updates

Ensure a DNS record is present or absent utilizing RFC 2136 type dynamic updates.

depends

- dnspython

Note: The `dnspython` module is required when managing DDNS using a TSIG key. If you are not using a TSIG key, DDNS is allowed by ACLs based on IP address and the `dnspython` module is not required.

Example:

```
webserver:
  ddns.present:
    - zone: example.com
    - ttl: 60
    - data: 111.222.333.444
    - nameserver: 123.234.345.456
    - keyfile: /srv/salt/dnspy_tsig_key.txt
```

`salt.states.ddns.`**`absent`**(*name*, *zone*, *data=None*, *rdtype=None*, ***kwargs*)

Ensures that the named DNS record is absent.

name The host portion of the DNS record, e.g., 'webserver'. Name and zone are concatenated when the entry is created unless name includes a trailing dot, so make sure that information is not duplicated in these two arguments.

zone The zone to check

data Data for the DNS record. E.g., the IP address for an A record. If omitted, all records matching name (and rdtype, if provided) will be purged.

rdtype DNS resource type. If omitted, all types will be purged.

`**kwargs` Additional arguments the ddns.update function may need (e.g. nameserver, keyfile, keyname). Note that the nsupdate key file can't be reused by this function, the keyfile and other arguments must follow the dnspython spec.

`salt.states.ddns.`**`present`**(*name*, *zone*, *ttl*, *data*, *rdtype='A'*, ***kwargs*)

Ensures that the named DNS record is present with the given ttl.

name The host portion of the DNS record, e.g., 'webserver'. Name and zone are concatenated when the entry is created unless name includes a trailing dot, so make sure that information is not duplicated in these two arguments.

zone The zone to check/update

ttl TTL for the record

data Data for the DNS record. E.g., the IP address for an A record.

rdtype DNS resource type. Default 'A'.

`**kwargs` Additional arguments the ddns.update function may need (e.g. nameserver, keyfile, keyname). Note that the nsupdate key file can't be reused by this function, the keyfile and other arguments must follow the dnspython spec.

21.19.68 salt.states.debconfmod

Management of debconf selections

depends

- debconf-utils package

The debconfmod state module manages the enforcement of debconf selections, this state can set those selections prior to package installation.

Available Functions

The debconfmod state has two functions, the `set` and `set_file` functions

set Set debconf selections from the state itself

set_file Set debconf selections from a file

```
nullmailer-debconf:
  debconf.set:
    - name: nullmailer
    - data:
        'shared/mailname': {'type': 'string', 'value': 'server.domain.tld'}
        'nullmailer/relayhost': {'type': 'string', 'value': 'mail.domain.tld'}
ferm-debconf:
  debconf.set:
    - name: ferm
    - data:
        'ferm/enable': {'type': 'boolean', 'value': True}
```

Note: Due to how PyYAML imports nested dicts (see *here*), the values in the `data` dict must be indented four spaces instead of two.

If you're setting debconf values that requires *dpkg-reconfigure*, you can use the `onchanges` requisite to reconfigure your package:

```
set-default-shell:
  debconf.set:
    - name: dash
    - data:
        'dash/sh': {'type': 'boolean', 'value': false}

reconfigure-dash:
  cmd.run:
    - name: dpkg-reconfigure -f noninteractive dash
    - onchanges:
      - debconf: set-default-shell
```

Every time the `set-default-shell` state changes, the `reconfigure-dash` state will also run.

Note: For boolean types, the value should be `true` or `false`, not `'true'` or `'false'`.

`salt.states.debconfmod.`**`set`**`(`*name*, *data*, ***kwargs*`)`

Set debconf selections

```
<state_id>:
  debconf.set:
    - name: <name>
    - data:
        <question>: {'type': <type>, 'value': <value>}
        <question>: {'type': <type>, 'value': <value>}

<state_id>:
  debconf.set:
    - name: <name>
    - data:
        <question>: {'type': <type>, 'value': <value>}
        <question>: {'type': <type>, 'value': <value>}
```

name: The package name to set answers for.

data: A set of questions/answers for debconf. Note that everything under this must be indented twice.
question: The question the is being pre-answered
type: The type of question that is being asked (string, boolean, select, etc.)
value: The answer to the question

`salt.states.debconfmod.`**`set_file`**`(`*name*, *source*, *template=None*, *context=None*, *defaults=None*, ***kwargs*`)`

Set debconf selections from a file or a template

```
<state_id>:
  debconf.set_file:
    - source: salt://pathto/pkg.selections

<state_id>:
  debconf.set_file:
    - source: salt://pathto/pkg.selections?saltenv=myenvironment

<state_id>:
  debconf.set_file:
    - source: salt://pathto/pkg.selections.jinja2
    - template: jinja
    - context:
        some_value: "false"
```

source: The location of the file containing the package selections
template If this setting is applied then the named templating engine will be used to render the package selections file, currently jinja, mako, and wempy are supported
context Overrides default context variables passed to the template.
defaults Default context passed to the template.

21.19.69 salt.states.dellchassis

Manage chassis via Salt Proxies.

New in version 2015.8.2.

Below is an example state that sets basic parameters:

```
my-dell-chassis:
  dellchassis.chassis:
    - chassis_name: my-dell-chassis
    - datacenter: dc-1-us
    - location: my-location
    - mode: 2
    - idrac_launch: 1
    - slot_names:
      - server-1: my-slot-name
      - server-2: my-other-slot-name
    - blade_power_states:
      - server-1: on
      - server-2: off
      - server-3: powercycle
```

However, it is possible to place the entire set of chassis configuration data in pillar. Here's an example pillar structure:

```
proxy:
  host: 10.27.20.18
```

(continues on next page)

(continued from previous page)

```
admin_username: root
fallback_admin_username: root
passwords:
  - super-secret
  - old-secret
proxytype: fx2

chassis:
  name: fx2-1
  username: root
  password: saltstack1
  datacenter: london
  location: rack-1-shelf-3
  management_mode: 2
  idrac_launch: 0
  slot_names:
    - 'server-1': blade1
    - 'server-2': blade2

  servers:
    server-1:
      idrac_password: saltstack1
      ipmi_over_lan: True
      ip: 172.17.17.132
      netmask: 255.255.0.0
      gateway: 172.17.17.1
    server-2:
      idrac_password: saltstack1
      ipmi_over_lan: True
      ip: 172.17.17.2
      netmask: 255.255.0.0
      gateway: 172.17.17.1
    server-3:
      idrac_password: saltstack1
      ipmi_over_lan: True
      ip: 172.17.17.20
      netmask: 255.255.0.0
      gateway: 172.17.17.1
    server-4:
      idrac_password: saltstack1
      ipmi_over_lan: True
      ip: 172.17.17.2
      netmask: 255.255.0.0
      gateway: 172.17.17.1

  switches:
    switch-1:
      ip: 192.168.1.2
      netmask: 255.255.255.0
      gateway: 192.168.1.1
      snmp: nonpublic
      password: saltstack1
    switch-2:
      ip: 192.168.1.3
      netmask: 255.255.255.0
      gateway: 192.168.1.1
```

(continues on next page)

(continued from previous page)

```
        snmp: nonpublic
        password: saltstack1
```

And to go with it, here's an example state that pulls the data from the pillar stated above:

```
{% set details = pillar.get('proxy:chassis', {}) %}
standup-step1:
  dellchassis.chassis:
    - name: {{ details['name'] }}
    - location: {{ details['location'] }}
    - mode: {{ details['management_mode'] }}
    - idrac_launch: {{ details['idrac_launch'] }}
    - slot_names:
      {% for entry details['slot_names'] %}
        - {{ entry.keys()[0] }}: {{ entry[entry.keys()[0]]  }}
      {% endfor %}

blade_powercycle:
  dellchassis.chassis:
    - blade_power_states:
      - server-1: powercycle
      - server-2: powercycle
      - server-3: powercycle
      - server-4: powercycle

# Set idrac_passwords for blades.  racadm needs them to be called 'server-x'
{% for k, v in details['servers'].iteritems() %}
{{ k }}:
  dellchassis.blade_idrac:
    - idrac_password: {{ v['idrac_password'] }}
{% endfor %}

# Set management ip addresses, passwords, and snmp strings for switches
{% for k, v in details['switches'].iteritems() %}
{{ k }}-switch-setup:
  dellchassis.switch:
    - name: {{ k }}
    - ip: {{ v['ip'] }}
    - netmask: {{ v['netmask'] }}
    - gateway: {{ v['gateway'] }}
    - password: {{ v['password'] }}
    - snmp: {{ v['snmp'] }}
{% endfor %}
```

Note: This state module relies on the dracr.py execution module, which runs racadm commands on the chassis, blades, etc. The racadm command runs very slowly and, depending on your state, the proxy minion return might timeout before the racadm commands have completed. If you are repeatedly seeing minions timeout after state calls, please use the `-t` CLI argument to increase the timeout variable.

For example:

```
salt '*' state.sls my-dell-chasis-state-name -t 60
```

Note: The Dell CMC units perform adequately but many iDRACs are **excruciatingly** slow. Some functions can take minutes to execute.

`salt.states.dellchassis.`**`blade_idrac`**`(`*name, idrac_password=None, idrac_ipmi=None, idrac_ip=None, idrac_netmask=None, idrac_gateway=None, idrac_dnsname=None, idrac_dhcp=None*`)`

Set parameters for iDRAC in a blade.

Parameters

- **`idrac_password`** – Password to use to connect to the iDRACs directly (idrac_ipmi and idrac_dnsname must be set directly on the iDRAC. They can't be set through the CMC. If this password is present, use it instead of the CMC password)
- **`idrac_ipmi`** – Enable/Disable IPMI over LAN
- **`idrac_ip`** – Set IP address for iDRAC
- **`idrac_netmask`** – Set netmask for iDRAC
- **`idrac_gateway`** – Set gateway for iDRAC
- **`idrac_dhcp`** – Turn on DHCP for iDRAC (True turns on, False does nothing becaause setting a static IP will disable DHCP).

Returns A standard Salt changes dictionary

NOTE: If any of the IP address settings is configured, all of ip, netmask, and gateway must be present

`salt.states.dellchassis.`**`chassis`**`(`*name, chassis_name=None, password=None, datacenter=None, location=None, mode=None, idrac_launch=None, slot_names=None, blade_power_states=None*`)`

Manage a Dell Chassis.

chassis_name The name of the chassis.

datacenter The datacenter in which the chassis is located

location The location of the chassis.

password Password for the chassis. Note: If this password is set for the chassis, the current implementation of this state will set this password both on the chassis and the iDrac passwords on any configured blades. If the password for the blades should be distinct, they should be set separately with the blade_idrac function.

mode The management mode of the chassis. Viable options are:

- 0: None
- 1: Monitor
- 2: Manage and Monitor

idrac_launch The iDRAC launch method of the chassis. Viable options are:

- 0: Disabled (launch iDRAC using IP address)
- 1: Enabled (launch iDRAC using DNS name)

slot_names The names of the slots, provided as a list identified by their slot numbers.

blade_power_states The power states of a blade server, provided as a list and identified by their server numbers. Viable options are:

- on: Ensure the blade server is powered on.
- off: Ensure the blade server is powered off.
- powercycle: Power cycle the blade server.

Example:

```yaml
my-dell-chassis:
  dellchassis.chassis:
    - chassis_name: my-dell-chassis
    - location: my-location
    - datacenter: london
    - mode: 2
    - idrac_launch: 1
    - slot_names:
```

(continues on next page)

(continued from previous page)

```
    - 1: my-slot-name
    - 2: my-other-slot-name
  - blade_power_states:
    - server-1: on
    - server-2: off
    - server-3: powercycle
```

`salt.states.dellchassis.`**`firmware_update`**(*hosts=None, directory=''*)

State to update the firmware on host using the `racadm` command

firmwarefile filename (string) starting with `salt://`

host string representing the hostname supplied to the `racadm` command

directory Directory name where firmwarefile will be downloaded

```
dell-chassis-firmware-update:
  dellchassis.firmware_update:
    hosts:
      cmc:
        salt://firmware_cmc.exe
      server-1:
        salt://firmware.exe
    directory: /opt/firmwares
```

`salt.states.dellchassis.`**`switch`**(*name, ip=None, netmask=None, gateway=None, dhcp=None, password=None, snmp=None*)

Manage switches in a Dell Chassis.

name The switch designation (e.g. switch-1, switch-2)

ip The Static IP Address of the switch

netmask The netmask for the static IP

gateway The gateway for the static IP

dhcp True: Enable DHCP False: Do not change DHCP setup (disabling DHCP is automatic when a static IP is set)

password The access (root) password for the switch

snmp The SNMP community string for the switch

Example:

```
my-dell-chassis:
  dellchassis.switch:
    - switch: switch-1
    - ip: 192.168.1.1
    - netmask: 255.255.255.0
    - gateway: 192.168.1.254
    - dhcp: True
    - password: secret
    - snmp: public
```

21.19.70 salt.states.disk

Disk monitoring state

Monitor the state of disk resources.

The `disk.status` function can be used to report that the used space of a filesystem is within the specified limits.

```
used_space:
  disk.status:
    - name: /dev/xda1
    - maximum: 79%
    - minimum: 11%
```

It can be used with an `onfail` requisite, for example, to take additional action in response to or in preparation for other states.

```
storage_threshold:
  disk.status:
    - name: /dev/xda1
    - maximum: 97%

clear_cache:
  cmd.run:
    - name: rm -r /var/cache/app
    - onfail:
      - disk: storage_threshold
```

To use kilobytes (KB) for `minimum` and `maximum` rather than percents, specify the `absolute` flag:

```
used_space:
  disk.status:
    - name: /dev/xda1
    - minimum: 1024 KB
    - maximum: 1048576 KB
    - absolute: True
```

`salt.states.disk.`**`status`**`(`*name, maximum=None, minimum=None, absolute=False, free=False*`)`

Return the current disk usage stats for the named mount point

name Disk mount or directory for which to check used space

maximum The maximum disk utilization

minimum The minimum disk utilization

absolute By default, the utilization is measured in percentage. Set the *absolute* flag to use kilobytes.

New in version 2016.11.0.

free By default, *minimum* & *maximum* refer to the amount of used space. Set to *True* to evaluate the free space instead.

21.19.71 salt.states.docker_container

Management of Docker containers

New in version 2017.7.0.

depends docker Python module

Note: Older releases of the Python bindings for Docker were called docker-py in PyPI. All releases of docker, and releases of docker-py >= 1.6.0 are supported. These python bindings can easily be installed using *`pip.install`*:

```
salt myminion pip.install docker
```

To upgrade from docker-py to docker, you must first uninstall docker-py, and then install docker:

```
salt myminion pip.uninstall docker-py
salt myminion pip.install docker
```

These states were moved from the `docker` state module (formerly called **dockerng**) in the 2017.7.0 release. When running the *`docker_container.running`* state for the first time after upgrading to 2017.7.0, your container(s) may be replaced. The changes may show diffs for certain parameters which say that the old value was an empty string, and the new value is `None`. This is due to the fact that in prior releases Salt was passing empty strings for these values when creating the container if they were undefined in the SLS file, where now Salt simply does not pass any arguments not explicitly defined in the SLS file. Subsequent runs of the state should not replace the container if the configuration remains unchanged.

Note: To pull from a Docker registry, authentication must be configured. See *here* for more information on how to configure access to docker registries in *Pillar* data.

`salt.states.docker_container.`**`absent`**`(`*name*, *force=False*`)`

Ensure that a container is absent

name Name of the container

force [False] Set to `True` to remove the container even if it is running

Usage Examples:

```
mycontainer:
  docker_container.absent

multiple_containers:
  docker_container.absent:
    - names:
      - foo
      - bar
      - baz
```

`salt.states.docker_container.`**`mod_run_check`**`(`*onlyif*, *unless*, *creates*`)`

Execute the onlyif/unless/creates logic. Returns a result dict if any of the checks fail, otherwise returns True

`salt.states.docker_container.`**`mod_watch`**`(`*name*, *sfun=None*, ***kwargs*`)`

The docker_container watcher, called to invoke the watch command.

Note: This state exists to support special handling of the `watch` *requisite*. It should not be called directly.

Parameters for this function should be set by the state being triggered.

`salt.states.docker_container.`**`run`**`(`*name*, *image=None*, *onlyif=None*, *unless=None*, *creates=None*, *bg=False*, *failhard=True*, *replace=False*, *force=False*, *skip_translate=None*, *ignore_collisions=False*, *validate_ip_addrs=True*, *client_timeout=60*, ***kwargs*`)`

New in version 2018.3.0.

Note: If no tag is specified in the image name, and nothing matching the specified image is pulled on the minion, the `docker pull` that retrieves the image will pull *all tags* for the image. A tag of `latest` is not implicit for the pull. For this reason, it is recommended to specify the image in `repo:tag` notation.

Like the *`cmd.run`* state, only for Docker. Does the equivalent of a `docker run` and returns information about the container that was created, as well as its output.

This state accepts the same arguments as *docker_container.running*, with the exception of `watch_action`, `start`, and `shutdown_timeout` (though the `force` argument has a different meaning in this state).

In addition, this state accepts the arguments from *docker.logs*, with the exception of `follow`, to control how logs are returned.

Additionally, the following arguments are supported:

onlyif A command or list of commands to run as a check. The container will only run if any of the specified commands returns a zero exit status.

unless A command or list of commands to run as a check. The container will only run if any of the specified commands returns a non-zero exit status.

creates A path or list of paths. Only run if one or more of the specified paths do not exist on the minion.

bg [False] If `True`, run container in background and do not await or deliver its results.

> **Note:** This may not be useful in cases where other states depend on the results of this state. Also, the logs will be inaccessible once the container exits if `auto_remove` is set to `True`, so keep this in mind.

failhard [True] If `True`, the state will return a `False` result if the exit code of the container is non-zero. When this argument is set to `False`, the state will return a `True` result regardless of the container's exit code.

> **Note:** This has no effect if `bg` is set to `True`.

replace [False] If `True`, and if the named container already exists, this will remove the existing container. The default behavior is to return a `False` result when the container already exists.

force [False] If `True`, and the named container already exists, *and* `replace` is also set to `True`, then the container will be forcibly removed. Otherwise, the state will not proceed and will return a `False` result.

CLI Examples:

```
salt myminion docker.run_container myuser/myimage command=/usr/local/bin/myscript.
↪sh
```

USAGE EXAMPLE

```
{% set pkg_version = salt.pillar.get('pkg_version', '1.0-1') %}
build_package:
  docker_container.run:
    - image: myuser/builder:latest
    - binds: /home/myuser/builds:/build_dir
    - command: /scripts/build.sh {{ pkg_version }}
    - creates: /home/myuser/builds/myapp-{{ pkg_version }}.noarch.rpm
    - replace: True
    - networks:
      - mynet
    - require:
      - docker_network: mynet
```

`salt.states.docker_container.`**`running`**(*name, image=None, skip_translate=None, ignore_collisions=False, validate_ip_addrs=True, force=False, watch_action='force', start=True, shutdown_timeout=None, client_timeout=60, networks=None, **kwargs*)

Ensure that a container with a specific configuration is present and running

name Name of the container

image Image to use for the container

Note: This state will pull the image if it is not present. However, if the image needs to be built from a Dockerfile or loaded from a saved image, or if you would like to use requisites to trigger a replacement of the container when the image is updated, then the `docker_image.present` state should be used to manage the image.

Changed in version 2018.3.0: If no tag is specified in the image name, and nothing matching the specified image is pulled on the minion, the `docker pull` that retrieves the image will pull *all tags* for the image. A tag of `latest` is no longer implicit for the pull. For this reason, it is recommended to specify the image in `repo:tag` notation.

skip_translate This function translates Salt CLI or SLS input into the format which docker-py expects. However, in the event that Salt's translation logic fails (due to potential changes in the Docker Remote API, or to bugs in the translation code), this argument can be used to exert granular control over which arguments are translated and which are not.

Pass this argument as a comma-separated list (or Python list) of arguments, and translation for each passed argument name will be skipped. Alternatively, pass `True` and *all* translation will be skipped.

Skipping tranlsation allows for arguments to be formatted directly in the format which docker-py expects. This allows for API changes and other issues to be more easily worked around. An example of using this option to skip translation would be:

For example, imagine that there is an issue with processing the `port_bindings` argument, and the following configuration no longer works as expected:

```
mycontainer:
  docker_container.running:
    - image: 7.3.1611
    - port_bindings:
      - 10.2.9.10:8080:80
```

By using `skip_translate`, you can forego the input translation and configure the port binding in the format docker-py needs:

```
mycontainer:
  docker_container.running:
    - image: 7.3.1611
    - skip_translate: port_bindings
    - port_bindings: {8080: [('10.2.9.10', 80)], '4193/udp': 9314}
```

See the following links for more information:

- docker-py Low-level API
- Docker Engine API

ignore_collisions [False] Since many of docker-py's arguments differ in name from their CLI counterparts (with which most Docker users are more familiar), Salt detects usage of these and aliases them to the docker-py version of that argument so that both CLI and API versions of a given argument are supported. However, if both the alias and the docker-py version of the same argument (e.g. `env` and `environment`) are used, an error will be raised. Set this argument to `True` to suppress these errors and keep the docker-py version of the argument.

validate_ip_addrs [True] For parameters which accept IP addresses as input, IP address validation will be performed. To disable, set this to `False`

force [False] Set this parameter to `True` to force Salt to re-create the container irrespective of whether or not it is configured as desired.

watch_action [force] Control what type of action is taken when this state *watches* another state that has changes. The default action is `force`, which runs the state with `force` set to `True`, triggering a rebuild of the container.

If any other value is passed, it will be assumed to be a kill signal. If the container matches the specified configuration, and is running, then the action will be to send that signal to the container. Kill signals can be either strings or numbers, and are defined in the **Standard Signals** section of the `signal(7)` manpage. Run `man 7 signal` on a Linux host to browse this manpage. For example:

```
mycontainer:
  docker_container.running:
    - image: busybox
    - watch_action: SIGHUP
    - watch:
      - file: some_file
```

Note: If the container differs from the specified configuration, or is not running, then instead of sending a signal to the container, the container will be re-created/started and no signal will be sent.

start [True] Set to `False` to suppress starting of the container if it exists, matches the desired configuration, but is not running. This is useful for data-only containers, or for non-daemonized container processes, such as the Django `migrate` and `collectstatic` commands. In instances such as this, the container only needs to be started the first time.

shutdown_timeout If the container needs to be replaced, the container will be stopped using *`docker.stop`*. If a `shutdown_timout` is not set, and the container was created using `stop_timeout`, that timeout will be used. If neither of these values were set, then a timeout of 10 seconds will be used.

Changed in version 2017.7.0: This option was renamed from `stop_timeout` to `shutdown_timeout` to accommodate the `stop_timeout` container configuration setting.

client_timeout [60] Timeout in seconds for the Docker client. This is not a timeout for this function, but for receiving a response from the API.

Note: This is only used if Salt needs to pull the requested image.

NETWORK MANAGEMENT

New in version 2018.3.0.

Changed in version 2019.2.0: If the `networks` option is used, any networks (including the default `bridge` network) which are not specified will be disconnected.

The `networks` argument can be used to ensure that a container is attached to one or more networks. Optionally, arguments can be passed to the networks. In the example below, `net1` is being configured with arguments, while `net2` and `bridge` are being configured *without* arguments:

```
foo:
  docker_container.running:
    - image: myuser/myimage:foo
    - networks:
      - net1:
        - aliases:
          - bar
          - baz
        - ipv4_address: 10.0.20.50
      - net2
      - bridge
    - require:
      - docker_network: net1
      - docker_network: net2
```

The supported arguments are the ones from the docker-py's connect_container_to_network function (other than `container` and `net_id`).

Important: Unlike with the arguments described in the **CONTAINER CONFIGURATION PARAMETERS** section below, these network configuration parameters are not translated at all. Consult the connect_container_to_network documentation for the correct type/format of data to pass.

To start a container with no network connectivity (only possible in 2019.2.0 and later) pass this option as an empty list. For example:

```yaml
foo:
  docker_container.running:
    - image: myuser/myimage:foo
    - networks: []
```

CONTAINER CONFIGURATION PARAMETERS

auto_remove (or *rm*) [False] Enable auto-removal of the container on daemon side when the container's process exits (analogous to running a docker container with `--rm` on the CLI).

```yaml
foo:
  docker_container.running:
    - image: bar/baz:latest
    - auto_remove: True
```

binds Files/directories to bind mount. Each bind mount should be passed in one of the following formats:

- `<host_path>:<container_path>` - `host_path` is mounted within the container as `container_path` with read-write access.
- `<host_path>:<container_path>:<selinux_context>` - `host_path` is mounted within the container as `container_path` with read-write access. Additionally, the specified selinux context will be set within the container.
- `<host_path>:<container_path>:<read_only>` - `host_path` is mounted within the container as `container_path`, with the read-only or read-write setting explicitly defined.
- `<host_path>:<container_path>:<read_only>,<selinux_context>` - `host_path` is mounted within the container as `container_path`, with the read-only or read-write setting explicitly defined. Additionally, the specified selinux context will be set within the container.

`<read_only>` can be either `rw` for read-write access, or `ro` for read-only access. When omitted, it is assumed to be read-write.

`<selinux_context>` can be `z` if the volume is shared between multiple containers, or `Z` if the volume should be private.

Note: When both `<read_only>` and `<selinux_context>` are specified, there must be a comma before `<selinux_context>`.

Binds can be expressed as a comma-separated list or a YAML list. The below two examples are equivalent:

```yaml
foo:
  docker_container.running:
    - image: bar/baz:latest
    - binds: /srv/www:/var/www:ro,/etc/foo.conf:/usr/local/etc/foo.conf:rw
```

```yaml
foo:
  docker_container.running:
```

(continues on next page)

(continued from previous page)

```
    - image: bar/baz:latest
    - binds:
      - /srv/www:/var/www:ro
      - /home/myuser/conf/foo.conf:/etc/foo.conf:rw
```

However, in cases where both ro/rw and an selinux context are combined, the only option is to use a YAML list, like so:

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - binds:
      - /srv/www:/var/www:ro,Z
      - /home/myuser/conf/foo.conf:/etc/foo.conf:rw,Z
```

Since the second bind in the previous example is mounted read-write, the `rw` and comma can be dropped. For example:

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - binds:
      - /srv/www:/var/www:ro,Z
      - /home/myuser/conf/foo.conf:/etc/foo.conf:Z
```

blkio_weight Block IO weight (relative weight), accepts a weight value between 10 and 1000.

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - blkio_weight: 100
```

blkio_weight_device Block IO weight (relative device weight), specified as a list of expressions in the format `PATH:RATE`

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - blkio_weight_device: /dev/sda:100
```

cap_add List of capabilities to add within the container. Can be expressed as a comma-separated list or a Python list. The below two examples are equivalent:

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - cap_add: SYS_ADMIN,MKNOD
```

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - cap_add:
      - SYS_ADMIN
      - MKNOD
```

Note: This option requires Docker 1.2.0 or newer.

cap_drop List of capabilities to drop within the container. Can be expressed as a comma-separated list or a Python list. The below two examples are equivalent:

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - cap_drop: SYS_ADMIN,MKNOD
```

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - cap_drop:
      - SYS_ADMIN
      - MKNOD
```

Note: This option requires Docker 1.2.0 or newer.

---

command (or *cmd*) Command to run in the container

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - command: bash
```

cpuset_cpus (or *cpuset*) CPUs on which which to allow execution, specified as a string containing a range (e.g. `0-3`) or a comma-separated list of CPUs (e.g. `0,1`).

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - cpuset_cpus: "0,1"
```

cpuset_mems Memory nodes on which which to allow execution, specified as a string containing a range (e.g. `0-3`) or a comma-separated list of MEMs (e.g. `0,1`). Only effective on NUMA systems.

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - cpuset_mems: "0,1"
```

cpu_group The length of a CPU period in microseconds

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - cpu_group: 100000
```

cpu_period Microseconds of CPU time that the container can get in a CPU period

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - cpu_period: 50000
```

cpu_shares CPU shares (relative weight), specified as an integer between 2 and 1024.

```
foo:
  docker_container.running:
```

(continues on next page)

(continued from previous page)

```
    - image: bar/baz:latest
    - cpu_shares: 512
```

detach [False] If `True`, run the container's command in the background (daemon mode)

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - detach: True
```

devices List of host devices to expose within the container. Can be expressed as a comma-separated list or a YAML list. The below two examples are equivalent:

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - devices: /dev/net/tun,/dev/xvda1:/dev/xvda1,/dev/xvdb1:/dev/xvdb1:r
```

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - devices:
      - /dev/net/tun
      - /dev/xvda1:/dev/xvda1
      - /dev/xvdb1:/dev/xvdb1:r
```

device_read_bps Limit read rate (bytes per second) from a device, specified as a list of expressions in the format `PATH:RATE`, where `RATE` is either an integer number of bytes, or a string ending in `kb`, `mb`, or `gb`. Can be expressed as a comma-separated list or a YAML list. The below two examples are equivalent:

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - devices_read_bps: /dev/sda:1mb,/dev/sdb:5mb
```

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - devices_read_bps:
      - /dev/sda:1mb
      - /dev/sdb:5mb
```

device_read_iops Limit read rate (I/O per second) from a device, specified as a list of expressions in the format `PATH:RATE`, where `RATE` is a number of I/O operations. Can be expressed as a comma-separated list or a YAML list. The below two examples are equivalent:

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - devices_read_iops: /dev/sda:1000,/dev/sdb:500
```

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - devices_read_iops:
      - /dev/sda:1000
      - /dev/sdb:500
```

device_write_bps Limit write rate (bytes per second) from a device, specified as a list of expressions in the format `PATH:RATE`, where `RATE` is either an integer number of bytes, or a string ending in `kb`, `mb`, or `gb`. Can be expressed as a comma-separated list or a YAML list. The below two examples are equivalent:

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - devices_write_bps: /dev/sda:1mb,/dev/sdb:5mb
```

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - devices_write_bps:
      - /dev/sda:1mb
      - /dev/sdb:5mb
```

device_read_iops Limit write rate (I/O per second) from a device, specified as a list of expressions in the format `PATH:RATE`, where `RATE` is a number of I/O operations. Can be expressed as a comma-separated list or a YAML list. The below two examples are equivalent:

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - devices_read_iops: /dev/sda:1000,/dev/sdb:500
```

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - devices_read_iops:
      - /dev/sda:1000
      - /dev/sdb:500
```

dns List of DNS nameservers. Can be expressed as a comma-separated list or a YAML list. The below two examples are equivalent:

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - dns: 8.8.8.8,8.8.4.4
```

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - dns:
      - 8.8.8.8
      - 8.8.4.4
```

Note: To skip IP address validation, use `validate_ip_addrs=False`

dns_opt Additional options to be added to the container's `resolv.conf` file. Can be expressed as a comma-separated list or a YAML list. The below two examples are equivalent:

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - dns_opt: ndots:9
```

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - dns_opt:
      - ndots:9
```

dns_search List of DNS search domains. Can be expressed as a comma-separated list or a YAML list. The below two examples are equivalent:

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - dns_search: foo1.domain.tld,foo2.domain.tld
```

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - dns_search:
      - foo1.domain.tld
      - foo2.domain.tld
```

domainname The domain name to use for the container

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - dommainname: domain.tld
```

entrypoint Entrypoint for the container

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - entrypoint: "mycmd --arg1 --arg2"
```

This argument can also be specified as a list:

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - entrypoint:
      - mycmd
      - --arg1
      - --arg2
```

environment Either a list of variable/value mappings, or a list of strings in the format `VARNAME=value`. The below three examples are equivalent:

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - environment:
      - VAR1: value
      - VAR2: value
```

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - environment: 'VAR1=value,VAR2=value'
```

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - environment:
      - VAR1=value
      - VAR2=value
```

extra_hosts Additional hosts to add to the container's /etc/hosts file. Can be expressed as a comma-separated list or a Python list. The below two examples are equivalent:

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - extra_hosts: web1:10.9.8.7,web2:10.9.8.8
```

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - extra_hosts:
      - web1:10.9.8.7
      - web2:10.9.8.8
```

Note: To skip IP address validation, use `validate_ip_addrs=False`

Note: This option requires Docker 1.3.0 or newer.

group_add List of additional group names and/or IDs that the container process will run as. Can be expressed as a comma-separated list or a YAML list. The below two examples are equivalent:

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - group_add: web,network
```

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - group_add:
      - web
      - network
```

hostname Hostname of the container. If not provided, the value passed as the container's``name`` will be used for the hostname.

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - hostname: web1
```

Warning: `hostname` cannot be set if `network_mode` is set to `host`. The below example will result in an error:

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - hostname: web1
    - network_mode: host
```

interactive (or *stdin_open*) [False] Leave stdin open, even if not attached

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - interactive: True
```

ipc_mode (or *ipc*) Set the IPC mode for the container. The default behavior is to create a private IPC namespace for the container, but this option can be used to change that behavior:

- `container:<container_name_or_id>` reuses another container shared memory, semaphores and message queues
- `host`: use the host's shared memory, semaphores and message queues

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - ipc_mode: container:foo
```

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - ipc_mode: host
```

> **Warning:** Using `host` gives the container full access to local shared memory and is therefore considered insecure.

isolation Specifies the type of isolation technology used by containers

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - isolation: hyperv
```

> **Note:** The default value on Windows server is `process`, while the default value on Windows client is `hyperv`. On Linux, only `default` is supported.

labels Add metadata to the container. Labels can be set both with and without values, and labels with values can be passed either as `key=value` or `key: value` pairs. For example, while the below would be very confusing to read, it is technically valid, and demonstrates the different ways in which labels can be passed:

```
mynet:
  docker_network.present:
    - labels:
      - foo
      - bar=baz
      - hello: world
```

The labels can also simply be passed as a YAML dictionary, though this can be error-prone due to some *idiosyncrasies* with how PyYAML loads nested data structures:

```
foo:
  docker_network.present:
    - labels:
        foo: ''
        bar: baz
        hello: world
```

Changed in version 2018.3.0: Methods for specifying labels can now be mixed. Earlier releases required either labels with or without values.

links Link this container to another. Links can be specified as a list of mappings or a comma-separated or Python list of expressions in the format `<container_name_or_id>:<link_alias>`. The below three examples are equivalent:

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - links:
      - web1: link1
      - web2: link2
```

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - links: web1:link1,web2:link2
```

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - links:
      - web1:link1
      - web2:link2
```

log_driver and log_opt Set container's logging driver and options to configure that driver. Requires Docker 1.6 or newer.

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - log_driver: syslog
    - log_opt:
      - syslog-address: tcp://192.168.0.42
      - syslog-facility: daemon
```

The `log_opt` can also be expressed as a comma-separated or YAML list of `key=value` pairs. The below two examples are equivalent to the above one:

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - log_driver: syslog
    - log_opt: "syslog-address=tcp://192.168.0.42,syslog-facility=daemon"
```

```
foo:
  docker_container.running:
```

(continues on next page)

(continued from previous page)

```
    - image: bar/baz:latest
    - log_driver: syslog
    - log_opt:
      - syslog-address=tcp://192.168.0.42
      - syslog-facility=daemon
```

Note: The logging driver feature was improved in Docker 1.13 introducing option name changes. Please see Docker's Configure logging drivers documentation for more information.

---

lxc_conf Additional LXC configuration parameters to set before starting the container. Either a list of variable/value mappings, or a list of strings in the format `VARNAME=value`. The below three examples are equivalent:

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - lxc_conf:
      - lxc.utsname: docker
      - lxc.arch: x86_64
```

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - lxc_conf: lxc.utsname=docker,lxc.arch=x86_64
```

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - lxc_conf:
      - lxc.utsname=docker
      - lxc.arch=x86_64
```

Note: These LXC configuration parameters will only have the desired effect if the container is using the LXC execution driver, which has been deprecated for some time.

---

mac_address MAC address to use for the container. If not specified, a random MAC address will be used.

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - mac_address: 01:23:45:67:89:0a
```

mem_limit (or *memory*) [0] Memory limit. Can be specified in bytes or using single-letter units (i.e. `512M`, `2G`, etc.). A value of `0` (the default) means no memory limit.

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - mem_limit: 512M
```

mem_swappiness Tune a container's memory swappiness behavior. Accepts an integer between 0 and 100.

```
foo:
  docker_container.running:
```

(continues on next page)

(continued from previous page)

```
    - image: bar/baz:latest
    - mem_swappiness: 60
```

memswap_limit (or *memory_swap*) [-1] Total memory limit (memory plus swap). Set to `-1` to disable swap. A value of `0` means no swap limit.

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - memswap_limit: 1G
```

network_disabled [False] If `True`, networking will be disabled within the container

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - network_disabled: True
```

network_mode [bridge] One of the following:

- `bridge` - Creates a new network stack for the container on the docker bridge
- `none` - No networking (equivalent of the Docker CLI argument `--net=none`). Not to be confused with Python's `None`.
- `container:<name_or_id>` - Reuses another container's network stack
- `host` - Use the host's network stack inside the container

> **Warning:** Using `host` mode gives the container full access to the hosts system's services (such as D-bus), and is therefore considered insecure.

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - network_mode: "none"
```

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - network_mode: container:web1
```

oom_kill_disable Whether to disable OOM killer

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - oom_kill_disable: False
```

oom_score_adj An integer value containing the score given to the container in order to tune OOM killer preferences

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - oom_score_adj: 500
```

pid_mode Set to `host` to use the host container's PID namespace within the container. Requires Docker 1.5.0 or newer.

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - pid_mode: host
```

Note: This option requires Docker 1.5.0 or newer.

---

pids_limit Set the container's PID limit. Set to `-1` for unlimited.

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - pids_limit: 2000
```

port_bindings (or *publish*) Bind exposed ports. Port bindings should be passed in the same way as the `--publish` argument to the `docker run` CLI command:

- `ip:hostPort:containerPort` - Bind a specific IP and port on the host to a specific port within the container.
- `ip::containerPort` - Bind a specific IP and an ephemeral port to a specific port within the container.
- `hostPort:containerPort` - Bind a specific port on all of the host's interfaces to a specific port within the container.
- `containerPort` - Bind an ephemeral port on all of the host's interfaces to a specific port within the container.

Multiple bindings can be separated by commas, or expressed as a YAML list, and port ranges can be defined using dashes. The below two examples are equivalent:

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - port_bindings: "4505-4506:14505-14506,2123:2123/udp,8080"
```

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - port_bindings:
      - 4505-4506:14505-14506
      - 2123:2123/udp
      - 8080
```

Note: When specifying a protocol, it must be passed in the `containerPort` value, as seen in the examples above.

---

ports A list of ports to expose on the container. Can either be a comma-separated list or a YAML list. If the protocol is omitted, the port will be assumed to be a TCP port. The below two examples are equivalent:

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - ports: 1111,2222/udp
```

```
foo:
  docker_container.running:
```

(continues on next page)

(continued from previous page)

```
    - image: bar/baz:latest
    - ports:
      - 1111
      - 2222/udp
```

privileged [False] If `True`, runs the exec process with extended privileges

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - privileged: True
```

publish_all_ports (or *publish_all*) [False] Publish all ports to the host

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - ports: 8080
    - publish_all_ports: True
```

read_only [False] If `True`, mount the container's root filesystem as read only

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - read_only: True
```

restart_policy (or *restart*) Set a restart policy for the container. Must be passed as a string in the format `policy[:retry_count]` where `policy` is one of `always`, `unless-stopped`, or `on-failure`, and `retry_count` is an optional limit to the number of retries. The retry count is ignored when using the `always` or `unless-stopped` restart policy.

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - restart_policy: on-failure:5

bar:
  docker_container.running:
    - image: bar/baz:latest
    - restart_policy: always
```

security_opt (or *security_opts*): Security configuration for MLS systems such as SELinux and AppArmor. Can be expressed as a comma-separated list or a YAML list. The below two examples are equivalent:

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - security_opt: apparmor:unconfined
```

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - security_opt:
      - apparmor:unconfined
```

Important: Some security options can contain commas. In these cases, this argument *must* be passed as a Python list, as splitting by comma will result in an invalid configuration.

Note: See the documentation for security_opt at https://docs.docker.com/engine/reference/run/#security-configuration

shm_size Size of /dev/shm

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - shm_size: 128M
```

stop_signal Specify the signal docker will send to the container when stopping. Useful when running systemd as PID 1 inside the container.

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - stop_signal: SIGRTMIN+3
```

Note: This option requires Docker 1.9.0 or newer and docker-py 1.7.0 or newer.

New in version 2016.11.0.

stop_timeout Timeout to stop the container, in seconds

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - stop_timeout: 5
```

Note: In releases prior to 2017.7.0, this option was not set in the container configuration, but rather this timeout was enforced only when shutting down an existing container to replace it. To remove the ambiguity, and to allow for the container to have a stop timeout set for it, the old `stop_timeout` argument has been renamed to `shutdown_timeout`, while `stop_timeout` now refer's to the container's configured stop timeout.

storage_opt Storage driver options for the container. Can be either a list of strings in the format `option=value`, or a list of mappings between option and value. The below three examples are equivalent:

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - storage_opt:
      - dm.basesize: 40G
```

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - storage_opt: dm.basesize=40G
```

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - storage_opt:
      - dm.basesize=40G
```

sysctls (or *sysctl*) Set sysctl options for the container. Can be either a list of strings in the format `option=value`, or a list of mappings between option and value. The below three examples are equivalent:

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - sysctls:
      - fs.nr_open: 1048576
      - kernel.pid_max: 32768
```

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - sysctls: fs.nr_open=1048576,kernel.pid_max=32768
```

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - sysctls:
      - fs.nr_open=1048576
      - kernel.pid_max=32768
```

tmpfs A map of container directories which should be replaced by tmpfs mounts and their corresponding mount options.

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - tmpfs:
      - /run: rw,noexec,nosuid,size=65536k
```

tty [False] Attach TTYs

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - tty: True
```

ulimits List of ulimits. These limits should be passed in the format `<ulimit_name>:<soft_limit>:<hard_limit>`, with the hard limit being optional. Can be expressed as a comma-separated list or a YAML list. The below two examples are equivalent:

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - ulimits: nofile=1024:1024,nproc=60
```

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - ulimits:
```

(continues on next page)

(continued from previous page)

```
      - nofile=1024:1024
      - nproc=60
```

user User under which to run exec process

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - user: foo
```

userns_mode (or *user_ns_mode*) Sets the user namsepace mode, when the user namespace remapping option is enabled

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - userns_mode: host
```

volumes (or *volume*) List of directories to expose as volumes. Can be expressed as a comma-separated list or a YAML list. The below two examples are equivalent:

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - volumes: /mnt/vol1,/mnt/vol2
```

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - volumes:
      - /mnt/vol1
      - /mnt/vol2
```

volumes_from Container names or IDs from which the container will get volumes. Can be expressed as a comma-separated list or a YAML list. The below two examples are equivalent:

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - volumes_from: foo
```

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - volumes_from:
      - foo
```

volume_driver sets the container's volume driver

```
foo:
  docker_container.running:
    - image: bar/baz:latest
    - volume_driver: foobar
```

working_dir (or *workdir*) Working directory inside the container

```
foo:
  docker_container.running:
```

(continues on next page)

(continued from previous page)

```
    - image: bar/baz:latest
    - working_dir: /var/log/nginx
```

`salt.states.docker_container.`**`stopped`**`(`*name=None, containers=None, shutdown_timeout=None, unpause=False, error_on_absent=True, **kwargs*`)`

Ensure that a container (or containers) is stopped

name Name or ID of the container

containers Run this state on more than one container at a time. The following two examples accomplish the same thing:

```
stopped_containers:
  docker_container.stopped:
    - names:
      - foo
      - bar
      - baz
```

```
stopped_containers:
  docker_container.stopped:
    - containers:
      - foo
      - bar
      - baz
```

However, the second example will be a bit quicker since Salt will stop all specified containers in a single run, rather than executing the state separately on each image (as it would in the first example).

shutdown_timeout Timeout for graceful shutdown of the container. If this timeout is exceeded, the container will be killed. If this value is not passed, then the container's configured `stop_timeout` will be observed. If `stop_timeout` was also unset on the container, then a timeout of 10 seconds will be used.

unpause [False] Set to `True` to unpause any paused containers before stopping. If unset, then an error will be raised for any container that was paused.

error_on_absent [True] By default, this state will return an error if any of the specified containers are absent. Set this to `False` to suppress that error.

21.19.72 salt.states.docker_image

Management of Docker images

New in version 2017.7.0.

depends docker Python module

Note: Older releases of the Python bindings for Docker were called docker-py in PyPI. All releases of docker, and releases of docker-py >= 1.6.0 are supported. These python bindings can easily be installed using *`pip.install`*:

```
salt myminion pip.install docker
```

To upgrade from docker-py to docker, you must first uninstall docker-py, and then install docker:

```
salt myminion pip.uninstall docker-py
salt myminion pip.install docker
```

These states were moved from the `docker` state module (formerly called **dockerng**) in the 2017.7.0 release.

Note: To pull from a Docker registry, authentication must be configured. See *here* for more information on how to configure access to docker registries in *Pillar* data.

`salt.states.docker_image.`**`absent`**`(`*name=None, images=None, force=False*`)`

Ensure that an image is absent from the Minion. Image names can be specified either using `repo:tag` notation, or just the repo name (in which case a tag of `latest` is assumed).

images Run this state on more than one image at a time. The following two examples accomplish the same thing:

```
remove_images:
  docker_image.absent:
    - names:
      - busybox
      - centos:6
      - nginx
```

```
remove_images:
  docker_image.absent:
    - images:
      - busybox
      - centos:6
      - nginx
```

However, the second example will be a bit quicker since Salt will do all the deletions in a single run, rather than executing the state separately on each image (as it would in the first example).

force [False] Salt will fail to remove any images currently in use by a container. Set this option to true to remove the image even if it is already present.

Note: This option can also be overridden by Pillar data. If the Minion has a pillar variable named `docker.running.force` which is set to `True`, it will turn on this option. This pillar variable can even be set at runtime. For example:

```
salt myminion state.sls docker_stuff pillar="{docker.force: True}"
```

If this pillar variable is present and set to `False`, then it will turn off this option.

For more granular control, setting a pillar variable named `docker.force.image_name` will affect only the named image.

`salt.states.docker_image.`**`mod_watch`**`(`*name, sfun=None, **kwargs*`)`

The docker_image watcher, called to invoke the watch command.

Note: This state exists to support special handling of the `watch` *requisite*. It should not be called directly.

Parameters for this function should be set by the state being triggered.

`salt.states.docker_image.`**`present`**`(`*name, tag=None, build=None, load=None, force=False, insecure_registry=False, client_timeout=60, dockerfile=None, sls=None, base='opensuse/python', saltenv='base', pillarenv=None, pillar=None, **kwargs*`)`

Changed in version 2018.3.0: The `tag` argument has been added. It is now required unless pulling from a registry.

Ensure that an image is present. The image can either be pulled from a Docker registry, built from a Dockerfile, loaded from a saved image, or built by running SLS files against a base image.

If none of the `build`, `load`, or `sls` arguments are used, then Salt will pull from the *configured registries*. If the specified image already exists, it will not be pulled unless `force` is set to `True`. Here is an example of a state that will pull an image from the Docker Hub:

```
myuser/myimage:
  docker_image.present:
    - tag: mytag
```

tag Tag name for the image. Required when using `build`, `load`, or `sls` to create the image, but optional if pulling from a repository.

New in version 2018.3.0.

build Path to directory on the Minion containing a Dockerfile

```
myuser/myimage:
  docker_image.present:
    - build: /home/myuser/docker/myimage
    - tag: mytag

myuser/myimage:
  docker_image.present:
    - build: /home/myuser/docker/myimage
    - tag: mytag
    - dockerfile: Dockerfile.alternative
```

The image will be built using *`docker.build`* and the specified image name and tag will be applied to it.

New in version 2016.11.0.

Changed in version 2018.3.0: The `tag` must be manually specified using the `tag` argument.

load Loads a tar archive created with *`docker.load`* (or the `docker load` Docker CLI command), and assigns it the specified repo and tag.

```
myuser/myimage:
  docker_image.present:
    - load: salt://path/to/image.tar
    - tag: mytag
```

Changed in version 2018.3.0: The `tag` must be manually specified using the `tag` argument.

force [False] Set this parameter to `True` to force Salt to pull/build/load the image even if it is already present.

client_timeout Timeout in seconds for the Docker client. This is not a timeout for the state, but for receiving a response from the API.

dockerfile Allows for an alternative Dockerfile to be specified. Path to alternative Dockefile is relative to the build path for the Docker container.

New in version 2016.11.0.

sls Allow for building of image with *`docker.sls_build`* by specifying the SLS files with which to build. This can be a list or comma-seperated string.

```
myuser/myimage:
  docker_image.present:
    - tag: latest
```

(continues on next page)

(continued from previous page)

```
    - sls:
        - webapp1
        - webapp2
    - base: centos
    - saltenv: base
```

Changed in version 2018.3.0: The `tag` must be manually specified using the `tag` argument.

base Base image with which to start *docker.sls_build*

New in version 2017.7.0.

saltenv Specify the environment from which to retrieve the SLS indicated by the *mods* parameter.

New in version 2017.7.0.

Changed in version 2018.3.0: Now uses the effective saltenv if not explicitly passed. In earlier versions, `base` was assumed as a default.

pillarenv Specify a Pillar environment to be used when applying states. This can also be set in the minion config file using the *pillarenv* option. When neither the *pillarenv* minion config option nor this CLI argument is used, all Pillar environments will be merged together.

New in version 2018.3.0.

pillar Custom Pillar values, passed as a dictionary of key-value pairs

Note: Values passed this way will override Pillar values set via `pillar_roots` or an external Pillar source.

New in version 2018.3.0.

21.19.73 salt.states.docker_network

Management of Docker networks

New in version 2017.7.0.

depends docker Python module

Note: Older releases of the Python bindings for Docker were called docker-py in PyPI. All releases of docker, and releases of docker-py >= 1.6.0 are supported. These python bindings can easily be installed using *pip.install*:

```
salt myminion pip.install docker
```

To upgrade from docker-py to docker, you must first uninstall docker-py, and then install docker:

```
salt myminion pip.uninstall docker-py
salt myminion pip.install docker
```

These states were moved from the `docker` state module (formerly called **dockerng**) in the 2017.7.0 release.

`salt.states.docker_network.`**`absent`**`(`*name*`)`

Ensure that a network is absent.

name Name of the network

Usage Example:

```
network_foo:
  docker_network.absent
```

`salt.states.docker_network.`**`present`**(*name*, *skip_translate=None*, *ignore_collisions=False*, *validate_ip_addrs=True*, *containers=None*, *reconnect=True*, ***kwargs*)

Changed in version 2018.3.0: Support added for network configuration options other than `driver` and `driver_opts`, as well as IPAM configuration.

Ensure that a network is present

Note: This state supports all arguments for network and IPAM pool configuration which are available for the release of docker-py installed on the minion. For that reason, the arguments described below in the *NETWORK CONFIGURATION* and *IP ADDRESS MANAGEMENT (IPAM)* sections may not accurately reflect what is available on the minion. The `docker.get_client_args` function can be used to check the available arguments for the installed version of docker-py (they are found in the `network_config` and `ipam_config` sections of the return data), but Salt will not prevent a user from attempting to use an argument which is unsupported in the release of Docker which is installed. In those cases, network creation be attempted but will fail.

---

name Network name

skip_translate This function translates Salt SLS input into the format which docker-py expects. However, in the event that Salt's translation logic fails (due to potential changes in the Docker Remote API, or to bugs in the translation code), this argument can be used to exert granular control over which arguments are translated and which are not.

Pass this argument as a comma-separated list (or Python list) of arguments, and translation for each passed argument name will be skipped. Alternatively, pass `True` and *all* translation will be skipped.

Skipping tranlsation allows for arguments to be formatted directly in the format which docker-py expects. This allows for API changes and other issues to be more easily worked around. See the following links for more information:

- docker-py Low-level API
- Docker Engine API

New in version 2018.3.0.

ignore_collisions [False] Since many of docker-py's arguments differ in name from their CLI counterparts (with which most Docker users are more familiar), Salt detects usage of these and aliases them to the docker-py version of that argument. However, if both the alias and the docker-py version of the same argument (e.g. `options` and `driver_opts`) are used, an error will be raised. Set this argument to `True` to suppress these errors and keep the docker-py version of the argument.

New in version 2018.3.0.

validate_ip_addrs [True] For parameters which accept IP addresses/subnets as input, validation will be performed. To disable, set this to `False`.

New in version 2018.3.0.

containers A list of containers which should be connected to this network.

Note: As of the 2018.3.0 release, this is not the recommended way of managing a container's membership in a network, for a couple reasons:

1. It does not support setting static IPs, aliases, or links in the container's IP configuration.
2. If a `docker_container.running` state replaces a container, it will not be reconnected to the network until the `docker_network.present` state is run again. Since containers often have `require` requisites to ensure that the network is present, this means that

the `docker_network.present` state ends up being run *before* the `docker_container.running`, leaving the container unattached at the end of the Salt run.

For these reasons, it is recommended to use *docker_container.running's network management support*.

---

reconnect [True] If `containers` is not used, and the network is replaced, then Salt will keep track of the containers which were connected to the network and reconnect them to the network after it is replaced. Salt will first attempt to reconnect using the same IP the container had before the network was replaced. If that fails (for instance, if the network was replaced because the subnet was modified), then the container will be reconnected without an explicit IP address, and its IP will be assigned by Docker.

Set this option to `False` to keep Salt from trying to reconnect containers. This can be useful in some cases when *managing static IPs in docker_container.running*. For instance, if a network's subnet is modified, it is likely that the static IP will need to be updated in the `docker_container.running` state as well. When the network is replaced, the initial reconnect attempt would fail, and the container would be reconnected with an automatically-assigned IP address. Then, when the `docker_container.running` state executes, it would disconnect the network *again* and reconnect using the new static IP. Disabling the reconnect behavior in these cases would prevent the unnecessary extra reconnection.

New in version 2018.3.0.

NETWORK CONFIGURATION ARGUMENTS

driver Network driver

```
mynet:
  docker_network.present:
    - driver: macvlan
```

driver_opts (or *driver_opt*, or *options*) Options for the network driver. Either a dictionary of option names and values or a Python list of strings in the format `varname=value`. The below three examples are equivalent:

```
mynet:
  docker_network.present:
    - driver: macvlan
    - driver_opts: macvlan_mode=bridge,parent=eth0
```

```
mynet:
  docker_network.present:
    - driver: macvlan
    - driver_opts:
      - macvlan_mode=bridge
      - parent=eth0
```

```
mynet:
  docker_network.present:
    - driver: macvlan
    - driver_opts:
      - macvlan_mode: bridge
      - parent: eth0
```

The options can also simply be passed as a dictionary, though this can be error-prone due to some *idiosyncrasies* with how PyYAML loads nested data structures:

```
mynet:
  docker_network.present:
    - driver: macvlan
```

(continues on next page)

(continued from previous page)

```
    - driver_opts:
        macvlan_mode: bridge
        parent: eth0
```

check_duplicate [True] If `True`, checks for networks with duplicate names. Since networks are primarily keyed based on a random ID and not on the name, and network name is strictly a user-friendly alias to the network which is uniquely identified using ID, there is no guaranteed way to check for duplicates. This option providess a best effort, checking for any networks which have the same name, but it is not guaranteed to catch all name collisions.

```
mynet:
  docker_network.present:
    - check_duplicate: False
```

internal [False] If `True`, restricts external access to the network

```
mynet:
  docker_network.present:
    - internal: True
```

labels Add metadata to the network. Labels can be set both with and without values, and labels with values can be passed either as `key=value` or `key: value` pairs. For example, while the below would be very confusing to read, it is technically valid, and demonstrates the different ways in which labels can be passed:

```
mynet:
  docker_network.present:
    - labels:
      - foo
      - bar=baz
      - hello: world
```

The labels can also simply be passed as a YAML dictionary, though this can be error-prone due to some *idiosyncrasies* with how PyYAML loads nested data structures:

```
foo:
  docker_network.present:
    - labels:
        foo: ''
        bar: baz
        hello: world
```

Changed in version 2018.3.0: Methods for specifying labels can now be mixed. Earlier releases required either labels with or without values.

enable_ipv6 (or *ipv6*) [False] Enable IPv6 on the network

```
mynet:
  docker_network.present:
    - enable_ipv6: True
```

Note: While it should go without saying, this argument must be set to `True` to *configure an IPv6 subnet*. Also, if this option is turned on without an IPv6 subnet explicitly configured, you will get an error unless you have set up a fixed IPv6 subnet. Consult the Docker IPv6 docs for information on how to do this.

attachable [False] If `True`, and the network is in the global scope, non-service containers on worker nodes will be able to connect to the network.

```
mynet:
  docker_network.present:
    - attachable: True
```

Note: This option cannot be reliably managed on CentOS 7. This is because while support for this option was added in API version 1.24, its value was not added to the inpsect results until API version 1.26. The version of Docker which is available for CentOS 7 runs API version 1.24, meaning that while Salt can pass this argument to the API, it has no way of knowing the value of this config option in an existing Docker network.

---

scope Specify the network's scope (`local`, `global` or `swarm`)

```
mynet:
  docker_network.present:
    - scope: local
```

ingress [False] If `True`, create an ingress network which provides the routing-mesh in swarm mode

```
mynet:
  docker_network.present:
    - ingress: True
```

IP ADDRESS MANAGEMENT (IPAM)

This state supports networks with either IPv4, or both IPv4 and IPv6. If configuring IPv4, then you can pass the *IPAM pool arguments* below as individual arguments. However, if configuring IPv4 and IPv6, the arguments must be passed as a list of dictionaries, in the `ipam_pools` argument (click *here* for some examples). These docs also have more information on these arguments.

IPAM ARGUMENTS

ipam_driver IPAM driver to use, if different from the default one

```
mynet:
  docker_network.present:
    - ipam_driver: foo
```

ipam_opts Options for the IPAM driver. Either a dictionary of option names and values or a Python list of strings in the format `varname=value`. The below three examples are equivalent:

```
mynet:
  docker_network.present:
    - ipam_driver: foo
    - ipam_opts: foo=bar,baz=qux
```

```
mynet:
  docker_network.present:
    - ipam_driver: foo
    - ipam_opts:
      - foo=bar
      - baz=qux
```

```
mynet:
  docker_network.present:
    - ipam_driver: foo
    - ipam_opts:
      - foo: bar
      - baz: qux
```

The options can also simply be passed as a dictionary, though this can be error-prone due to some *idiosyncrasies* with how PyYAML loads nested data structures:

```
mynet:
  docker_network.present:
    - ipam_driver: macvlan
    - ipam_opts:
        foo: bar
        baz: qux
```

IPAM POOL ARGUMENTS

subnet Subnet in CIDR format that represents a network segment

iprange (or *ip_range*) Allocate container IP from a sub-range within the subnet

Subnet in CIDR format that represents a network segment

gateway IPv4 or IPv6 gateway for the master subnet

aux_addresses (or *aux_address*) A dictionary of mapping container names to IP addresses which should be allocated for them should they connect to the network. Either a dictionary of option names and values or a Python list of strings in the format `host=ipaddr`.

IPAM CONFIGURATION EXAMPLES

Below is an example of an IPv4-only network (keep in mind that `subnet` is the only required argument).

```
mynet:
  docker_network.present:
    - subnet: 10.0.20.0/24
    - iprange: 10.0.20.128/25
    - gateway: 10.0.20.254
    - aux_addresses:
      - foo.bar.tld: 10.0.20.50
      - hello.world.tld: 10.0.20.51
```

Note: The `aux_addresses` can be passed differently, in the same way that `driver_opts` and `ipam_opts` can.

This same network could also be configured this way:

```
mynet:
  docker_network.present:
    - ipam_pools:
      - subnet: 10.0.20.0/24
        iprange: 10.0.20.128/25
        gateway: 10.0.20.254
        aux_addresses:
          foo.bar.tld: 10.0.20.50
          hello.world.tld: 10.0.20.51
```

Here is an example of a mixed IPv4/IPv6 subnet.

```
mynet:
  docker_network.present:
    - ipam_pools:
      - subnet: 10.0.20.0/24
        gateway: 10.0.20.1
      - subnet: fe3f:2180:26:1::/123
        gateway: fe3f:2180:26:1::1
```

21.19.74 salt.states.docker_volume

Management of Docker volumes

New in version 2017.7.0.

depends docker Python module

Note: Older releases of the Python bindings for Docker were called docker-py in PyPI. All releases of docker, and releases of docker-py >= 1.6.0 are supported. These python bindings can easily be installed using *`pip.install`*:

```
salt myminion pip.install docker
```

To upgrade from docker-py to docker, you must first uninstall docker-py, and then install docker:

```
salt myminion pip.uninstall docker-py
salt myminion pip.install docker
```

These states were moved from the `docker` state module (formerly called **dockerng**) in the 2017.7.0 release.

`salt.states.docker_volume.`**`absent`**(*name*, *driver=None*)

Ensure that a volume is absent.

New in version 2015.8.4.

Changed in version 2017.7.0: This state was renamed from **docker.volume_absent** to **docker_volume.absent**

name Name of the volume

Usage Examples:

```
volume_foo:
  docker_volume.absent
```

`salt.states.docker_volume.`**`present`**(*name*, *driver=None*, *driver_opts=None*, *force=False*)

Ensure that a volume is present.

New in version 2015.8.4.

Changed in version 2015.8.6: This state no longer deletes and re-creates a volume if the existing volume's driver does not match the `driver` parameter (unless the `force` parameter is set to `True`).

Changed in version 2017.7.0: This state was renamed from **docker.volume_present** to **docker_volume.present**

name Name of the volume

driver Type of driver for that volume. If `None` and the volume does not yet exist, the volume will be created using Docker's default driver. If `None` and the volume does exist, this function does nothing, even if the existing volume's driver is not the Docker default driver. (To ensure that an existing volume's driver matches the Docker default, you must explicitly name Docker's default driver here.)

driver_opts Options for the volume driver

force [False] If the volume already exists but the existing volume's driver does not match the driver specified by the `driver` parameter, this parameter controls whether the function errors out (if `False`) or deletes and re-creates the volume (if `True`).

New in version 2015.8.6.

Usage Examples:

```
volume_foo:
  docker_volume.present
```

```
volume_bar:
  docker_volume.present
    - name: bar
    - driver: local
    - driver_opts:
        foo: bar
```

```
volume_bar:
  docker_volume.present
    - name: bar
    - driver: local
    - driver_opts:
        - foo: bar
        - option: value
```

21.19.75 salt.states.drac

Management of Dell DRAC

The DRAC module is used to create and manage DRAC cards on Dell servers

Ensure the user damian is present

```
damian:
  drac.present:
    - name: damian
    - password: secret
    - permission: login,test_alerts,clear_logs
```

Ensure the user damian does not exist

```
damian:
  drac.absent:
    - name: damian
```

Ensure DRAC network is in a consistent state

```
my_network:
  drac.network:
    - ip: 10.225.108.29
    - netmask: 255.255.255.224
    - gateway: 10.225.108.1
```

`salt.states.drac.`**`absent`**`(`*name*`)`

Ensure a user does not exist on the Dell DRAC

name: The users username

`salt.states.drac.`**`network`**`(`*ip, netmask, gateway*`)`

Ensure the DRAC network settings are consistent

`salt.states.drac.`**`present`**`(`*name, password, permission*`)`

Ensure the user exists on the Dell DRAC

name: The users username

password The password used to authenticate

permission The permissions that should be assigned to a user

21.19.76 salt.states.elasticsearch

State module to manage Elasticsearch.

New in version 2017.7.0.

`salt.states.elasticsearch.`**`alias_absent`**`(`*name, index*`)`

Ensure that the index alias is absent.

name Name of the index alias to remove

index Name of the index for the alias

`salt.states.elasticsearch.`**`alias_present`**`(`*name, index, definition=None*`)`

Ensure that the named index alias is present.

name Name of the alias

index Name of the index

definition Optional dict for filters as per https://www.elastic.co/guide/en/elasticsearch/reference/current/indices-aliases.html

Example:

```
mytestalias:
  elasticsearch.alias_present:
    - index: testindex
    - definition:
        filter:
          term:
            user: kimchy
```

`salt.states.elasticsearch.`**`index_absent`**`(`*name*`)`

Ensure that the named index is absent.

name Name of the index to remove

`salt.states.elasticsearch.`**`index_present`**`(`*name, definition=None*`)`

Ensure that the named index is present.

name Name of the index to add

definition Optional dict for creation parameters as per https://www.elastic.co/guide/en/elasticsearch/reference/current/indices-create-index.html

Example:

```
# Default settings
mytestindex:
  elasticsearch_index.present

# Extra settings
mytestindex2:
  elasticsearch_index.present:
    - definition:
        settings:
          index:
            number_of_shards: 10
```

`salt.states.elasticsearch.`**`index_template_absent`**`(`*name*`)`

Ensure that the named index template is absent.

name Name of the index to remove

`salt.states.elasticsearch.`**`index_template_present`**`(`*name, definition, check_definition=False*`)`

Ensure that the named index template is present.

name Name of the index to add

definition Required dict for creation parameters as per https://www.elastic.co/guide/en/elasticsearch/reference/current/indices-templates.html

check_definition If the template already exists and the definition is up to date

Example:

```
mytestindex2_template:
  elasticsearch.index_template_present:
    - definition:
        template: logstash-*
        order: 1
        settings:
          number_of_shards: 1
```

`salt.states.elasticsearch.`**`pipeline_absent`**`(`*name*`)`

Ensure that the named pipeline is absent

name Name of the pipeline to remove

`salt.states.elasticsearch.`**`pipeline_present`**`(`*name*, *definition*`)`

Ensure that the named pipeline is present.

name Name of the index to add

definition Required dict for creation parameters as per https://www.elastic.co/guide/en/elasticsearch/reference/master/pipeline.html

Example:

```
test_pipeline:
  elasticsearch.pipeline_present:
    - definition:
        description: example pipeline
        processors:
          - set:
              field: collector_timestamp_millis
              value: '{{ '{{' }}_ingest.timestamp{{ '}}' }}'
```

`salt.states.elasticsearch.`**`search_template_absent`**`(`*name*`)`

Ensure that the search template is absent

name Name of the search template to remove

`salt.states.elasticsearch.`**`search_template_present`**`(`*name*, *definition*`)`

Ensure that the named search template is present.

name Name of the search template to add

definition Required dict for creation parameters as per http://www.elastic.co/guide/en/elasticsearch/reference/current/search-template.html

Example:

```
test_pipeline:
  elasticsearch.search_template_present:
    - definition:
        inline:
          size: 10
```

21.19.77 salt.states.elasticsearch_index

State module to manage Elasticsearch indices

New in version 2015.8.0.

Deprecated since version 2017.7.0: Use elasticsearch state instead `salt.states.elasticsearch_index.`**`absent`**`(`*name*`)`

Ensure that the named index is absent.

name Name of the index to remove

`salt.states.elasticsearch_index.`**`present`**`(`*name, definition=None*`)`

New in version 2015.8.0.

Changed in version 2017.3.0: Marked `definition` as optional.

Ensure that the named index is present.

name Name of the index to add

definition Optional dict for creation parameters as per https://www.elastic.co/guide/en/elasticsearch/reference/current/indices-create-index.html

Example:

```
# Default settings
mytestindex:
  elasticsearch_index.present

# Extra settings
mytestindex2:
  elasticsearch_index.present:
    - definition:
        settings:
          index:
            number_of_shards: 10
```

21.19.78 salt.states.elasticsearch_index_template

State module to manage Elasticsearch index templates

New in version 2015.8.0.

Deprecated since version 2017.7.0: Use elasticsearch state instead

`salt.states.elasticsearch_index_template.`**`absent`**`(`*name*`)`

Ensure that the named index template is absent.

name Name of the index to remove

`salt.states.elasticsearch_index_template.`**`present`**`(`*name, definition*`)`

New in version 2015.8.0.

Changed in version 2017.3.0: Marked `definition` as required.

Ensure that the named index templat eis present.

name Name of the index to add

definition Required dict for creation parameters as per https://www.elastic.co/guide/en/elasticsearch/reference/current/indices-templates.html

Example:

```
mytestindex2_template:
  elasticsearch_index_template.present:
    - definition:
        template: logstash-*
        order: 1
        settings:
          number_of_shards: 1
```

21.19.79 salt.states.environ

Support for getting and setting the environment variables of the current salt process.

`salt.states.environ.`**`setenv`**(*name*, *value*, *false_unsets=False*, *clear_all=False*, *update_minion=False*, *permanent=False*)

Set the salt process environment variables.

name The environment key to set. Must be a string.

value Either a string or dict. When string, it will be the value set for the environment key of 'name' above. When a dict, each key/value pair represents an environment variable to set.

false_unsets If a key's value is False and false_unsets is True, then the key will be removed from the salt processes environment dict entirely. If a key's value is False and false_unsets is not True, then the key's value will be set to an empty string. Default: False

clear_all USE WITH CAUTION! This option can unset environment variables needed for salt to function properly. If clear_all is True, then any environment variables not defined in the environ dict will be deleted. Default: False

update_minion If True, apply these environ changes to the main salt-minion process. If False, the environ changes will only affect the current salt subprocess. Default: False

permanent On Windows minions this will set the environment variable in the registry so that it is always added as a environment variable when applications open. If you want to set the variable to HKLM instead of HKCU just pass in "HKLM" for this parameter. On all other minion types this will be ignored. Note: This will only take affect on applications opened after this has been set.

Example:

```
a_string_env:
   environ.setenv:
     - name: foo
     - value: bar
     - update_minion: True

a_dict_env:
   environ.setenv:
     - name: does_not_matter
     - value:
         foo: bar
         baz: quux
```

21.19.80 salt.states.eselect

Management of Gentoo configuration using eselect

A state module to manage Gentoo configuration via eselect

`salt.states.eselect.`**`set_`**(*name*, *target*, *module_parameter=None*, *action_parameter=None*)

Verify that the given module is set to the given target

name The name of the module

target The target to be set for this module

module_parameter additional params passed to the defined module

action_parameter additional params passed to the defined action

```
profile:
  eselect.set:
    - target: hardened/linux/amd64
```

21.19.81 salt.states.etcd_mod

Manage etcd Keys

New in version 2015.8.0.

depends

- python-etcd

This state module supports setting and removing keys from etcd.

Configuration

To work with an etcd server you must configure an etcd profile. The etcd config can be set in either the Salt Minion configuration file or in pillar:

```
my_etd_config:
  etcd.host: 127.0.0.1
  etcd.port: 4001
```

It is technically possible to configure etcd without using a profile, but this is not considered to be a best practice, especially when multiple etcd servers or clusters are available.

```
etcd.host: 127.0.0.1
etcd.port: 4001
```

Note: The etcd configuration can also be set in the Salt Master config file, but in order to use any etcd configurations defined in the Salt Master config, the *`pillar_opts`* must be set to `True`.

Be aware that setting `pillar_opts` to `True` has security implications as this makes all master configuration settings available in all minion's pillars.

Etcd profile configuration can be overridden using following arguments: `host`, `port`, `username`, `password`, `ca`, `client_key` and `client_cert`.

```
my-value:
  etcd.set:
    - name: /path/to/key
    - value: value
    - host: 127.0.0.1
    - port: 2379
    - username: user
    - password: pass
```

Available Functions

- `set`

  This will set a value to a key in etcd. Changes will be returned if the key has been created or the value of the key has been updated. This means you can watch these states for changes.

```
/foo/bar/baz:
  etcd.set:
    - value: foo
    - profile: my_etcd_config
```

- `wait_set`

  Performs the same functionality as `set` but only if a watch requisite is `True`.

```
/some/file.txt:
  file.managed:
    - source: salt://file.txt

/foo/bar/baz:
  etcd.wait_set:
    - value: foo
    - profile: my_etcd_config
    - watch:
      - file: /some/file.txt
```

- `rm`

  This will delete a key from etcd. If the key exists then changes will be returned and thus you can watch for changes on the state, if the key does not exist then no changes will occur.

```
/foo/bar/baz:
  etcd.rm:
    - profile: my_etcd_config
```

- `wait_rm`

  Performs the same functionality as `rm` but only if a watch requisite is `True`.

```
/some/file.txt:
  file.managed:
    - source: salt://file.txt

/foo/bar/baz:
  etcd.wait_rm:
    - profile: my_etcd_config
    - watch:
      - file: /some/file.txt
```

`salt.states.etcd_mod.`**`directory`**(*name*, *profile=None*, ***kwargs*)

Create a directory in etcd.

name The etcd directory name, for example: `/foo/bar/baz`.

profile Optional, defaults to `None`. Sets the etcd profile to use which has been defined in the Salt Master config.

```
my_etd_config:
  etcd.host: 127.0.0.1
  etcd.port: 4001
```

`salt.states.etcd_mod.`**`mod_watch`**(*name*, ***kwargs*)

The etcd watcher, called to invoke the watch command. When called, execute a etcd function based on a watch call requisite.

Note: This state exists to support special handling of the `watch` *requisite*. It should not be called directly.

Parameters for this function should be set by the state being triggered.

`salt.states.etcd_mod.`**`rm`**(*name*, *recurse=False*, *profile=None*, ***kwargs*)

Deletes a key from etcd

name The etcd key name to remove, for example `/foo/bar/baz`.

recurse Optional, defaults to `False`. If `True` performs a recursive delete.

profile Optional, defaults to `None`. Sets the etcd profile to use which has been defined in the Salt Master config.

```
my_etd_config:
  etcd.host: 127.0.0.1
  etcd.port: 4001
```

`salt.states.etcd_mod.`**`set_`**(*name*, *value*, *profile=None*, ***kwargs*)

Set a key in etcd

name The etcd key name, for example: `/foo/bar/baz`.

value The value the key should contain.

profile Optional, defaults to `None`. Sets the etcd profile to use which has been defined in the Salt Master config.

```
my_etd_config:
  etcd.host: 127.0.0.1
  etcd.port: 4001
```

`salt.states.etcd_mod.`**`wait_rm`**(*name*, *recurse=False*, *profile=None*, ***kwargs*)

Deletes a key from etcd only if the watch statement calls it. This function is also aliased as `wait_rm`.

name The etcd key name to remove, for example `/foo/bar/baz`.

recurse Optional, defaults to `False`. If `True` performs a recursive delete, see: https://python-etcd.readthedocs.io/en/latest/#delete-a-key.

profile Optional, defaults to `None`. Sets the etcd profile to use which has been defined in the Salt Master config.

```
my_etd_config:
  etcd.host: 127.0.0.1
  etcd.port: 4001
```

`salt.states.etcd_mod.`**`wait_set`**(*name*, *value*, *profile=None*, ***kwargs*)

Set a key in etcd only if the watch statement calls it. This function is also aliased as `wait_set`.

name The etcd key name, for example: `/foo/bar/baz`.

value The value the key should contain.

profile The etcd profile to use that has been configured on the Salt Master, this is optional and defaults to `None`.

```
my_etd_config:
  etcd.host: 127.0.0.1
  etcd.port: 4001
```

21.19.82 salt.states.ethtool module

Configuration of network device

New in version 2016.11.0.

codeauthor Krzysztof Pawlowski <msciciel@msciciel.eu>

maturity new

depends python-ethtool

platform linux

```
eth0:
  ethtool.coalesce:
    - name: eth0
    - rx_usecs: 24
    - tx_usecs: 48

eth0:
  ethtool.ring:
    - name: eth0
    - rx: 1024
    - tx: 1024

eth0:
  ethtool.offload:
    - name: eth0
    - tcp_segmentation_offload: on
```

`salt.states.ethtool.`**`coalesce`**(*name*, ***kwargs*)

Manage coalescing settings of network device

name Interface name to apply coalescing settings

```
eth0:
  ethtool.coalesce:
    - name: eth0
    - adaptive_rx: on
    - adaptive_tx: on
    - rx_usecs: 24
    - rx_frame: 0
    - rx_usecs_irq: 0
    - rx_frames_irq: 0
    - tx_usecs: 48
    - tx_frames: 0
    - tx_usecs_irq: 0
    - tx_frames_irq: 0
    - stats_block_usecs: 0
    - pkt_rate_low: 0
    - rx_usecs_low: 0
    - rx_frames_low: 0
    - tx_usecs_low: 0
    - tx_frames_low: 0
    - pkt_rate_high: 0
    - rx_usecs_high: 0
    - rx_frames_high: 0
    - tx_usecs_high: 0
    - tx_frames_high: 0
    - sample_interval: 0
```

`salt.states.ethtool.`**`offload`**(*name*, ***kwargs*)

Manage protocol offload and other features of network device

name Interface name to apply coalescing settings

```
eth0:
```

(continues on next page)

(continued from previous page)

```
  ethtool.offload:
    - name: eth0
    - tcp_segmentation_offload: on
```

`salt.states.ethtool.ring(name, **kwargs)`

Manage rx/tx ring parameters of network device

Use 'max' word to set with factory maximum

name Interface name to apply ring parameters

```
eth0:
  ethtool.ring:
    - name: eth0
    - rx: 1024
    - rx_mini: 0
    - rx_jumbo: 0
    - tx: max
```

21.19.83 salt.states.esxi

Manage VMware ESXi Hosts.

New in version 2015.8.4.

Dependencies

- pyVmomi Python Module
- ESXCLI

pyVmomi

PyVmomi can be installed via pip:

```
pip install pyVmomi
```

Note: Version 6.0 of pyVmomi has some problems with SSL error handling on certain versions of Python. If using version 6.0 of pyVmomi, Python 2.6, Python 2.7.9, or newer must be present. This is due to an upstream dependency in pyVmomi 6.0 that is not supported in Python versions 2.7 to 2.7.8. If the version of Python is not in the supported range, you will need to install an earlier version of pyVmomi. See Issue #29537 for more information.

Based on the note above, to install an earlier version of pyVmomi than the version currently listed in PyPi, run the following:

```
pip install pyVmomi==5.5.0.2014.1.1
```

The 5.5.0.2014.1.1 is a known stable version that this original ESXi State Module was developed against.

ESXCLI

Currently, about a third of the functions used in the vSphere Execution Module require the ESXCLI package be installed on the machine running the Proxy Minion process.

The ESXCLI package is also referred to as the VMware vSphere CLI, or vCLI. VMware provides vCLI package installation instructions for vSphere 5.5 and vSphere 6.0.

Once all of the required dependencies are in place and the vCLI package is installed, you can check to see if you can connect to your ESXi host or vCenter server by running the following command:

```
esxcli -s <host-location> -u <username> -p <password> system syslog config get
```

If the connection was successful, ESXCLI was successfully installed on your system. You should see output related to the ESXi host's syslog configuration.

Note: Be aware that some functionality in this state module may depend on the type of license attached to the ESXi host.

For example, certain services are only available to manipulate service state or policies with a VMware vSphere Enterprise or Enterprise Plus license, while others are available with a Standard license. The `ntpd` service is restricted to an Enterprise Plus license, while `ssh` is available via the Standard license.

Please see the vSphere Comparison page for more information.

About

This state module was written to be used in conjunction with Salt's *`ESXi Proxy Minion`*. For a tutorial on how to use Salt's ESXi Proxy Minion, please refer to the *ESXi Proxy Minion Tutorial* for configuration examples, dependency installation instructions, how to run remote execution functions against ESXi hosts via a Salt Proxy Minion, and a larger state example.

`salt.states.esxi.`**`coredump_configured`**(*name, enabled, dump_ip, host_vnic='vmk0', dump_port=6500*)

Ensures a host's core dump configuration.

name Name of the state.

enabled Sets whether or not ESXi core dump collection should be enabled. This is a boolean value set to `True` or `False` to enable or disable core dumps.

Note that ESXi requires that the core dump must be enabled before any other parameters may be set. This also affects the `changes` results in the state return dictionary. If `enabled` is `False`, we can't obtain any previous settings to compare other state variables, resulting in many `old` references returning `None`.

Once `enabled` is `True` the `changes` dictionary comparisons will be more accurate. This is due to the way the system coredemp network configuration command returns data.

dump_ip The IP address of host that will accept the dump.

host_vnic Host VNic port through which to communicate. Defaults to `vmk0`.

dump_port TCP port to use for the dump. Defaults to `6500`.

Example:

```
configure-host-coredump:
  esxi.coredump_configured:
    - enabled: True
    - dump_ip: 'my-coredump-ip.example.com'
```

`salt.states.esxi.`**`diskgroups_configured`**`(`*name*, *diskgroups*, *erase_disks=False*`)`

Configures the disk groups to use for vsan.

This function will do the following:

1. Check whether or not all disks in the diskgroup spec exist, and raises and errors if they do not.
2. Create diskgroups with the correct disk configurations if diskgroup (identified by the cache disk canonical name) doesn't exist
3. Adds extra capacity disks to the existing diskgroup

Example:

```
{
    'cache_scsi_addr': 'vmhba1:C0:T0:L0',
    'capacity_scsi_addrs': [
        'vmhba2:C0:T0:L0',
        'vmhba3:C0:T0:L0',
        'vmhba4:C0:T0:L0',
    ]
}
```

name Mandatory state name

diskgroups Disk group representation containing scsi disk addresses. Scsi addresses are expected for disks in the diskgroup:

erase_disks Specifies whether to erase all partitions on all disks member of the disk group before the disk group is created. Default value is False.

`salt.states.esxi.`**`host_cache_configured`**`(`*name*, *enabled*, *datastore*, *swap_size='100%'*, *dedicated_backing_disk=False*, *erase_backing_disk=False*`)`

Configures the host cache used for swapping.

It will do the following:

1. Checks if backing disk exists
2. Creates the VMFS datastore if doesn't exist (datastore partition will be created and use the entire disk)
3. Raises an error if `dedicated_backing_disk` is `True` and partitions already exist on the backing disk
4. Configures host_cache to use a portion of the datastore for caching (either a specific size or a percentage of the datastore)

Examples

Percentage swap size (can't be 100%)

```
{
    'enabled': true,
    'datastore': {
        'backing_disk_scsi_addr': 'vmhba0:C0:T0:L0',
        'vmfs_version': 5,
        'name': 'hostcache'
        }
    'dedicated_backing_disk': false
    'swap_size': '98%',
}
```

Fixed sized swap size

```
{
    'enabled': true,
    'datastore': {
        'backing_disk_scsi_addr': 'vmhba0:C0:T0:L0',
```

(continues on next page)

(continued from previous page)

```
        'vmfs_version': 5,
        'name': 'hostcache'
        }
    'dedicated_backing_disk': true
    'swap_size': '10GiB',
}
```

name Mandatory state name.

enabled Specifies whether the host cache is enabled.

datastore Specifies the host cache datastore.

swap_size Specifies the size of the host cache swap. Can be a percentage or a value in GiB. Default value is `100%`.

dedicated_backing_disk Specifies whether the backing disk is dedicated to the host cache which means it must have no other partitions. Default is False

erase_backing_disk Specifies whether to erase all partitions on the backing disk before the datastore is created. Default value is False.

`salt.states.esxi.`**`ntp_configured`**(*name*, *service_running*, *ntp_servers=None*, *service_policy=None*, *service_restart=False*, *update_datetime=False*)

Ensures a host's NTP server configuration such as setting NTP servers, ensuring the NTP daemon is running or stopped, or restarting the NTP daemon for the ESXi host.

name Name of the state.

service_running Ensures the running state of the ntp daemon for the host. Boolean value where `True` indicates that ntpd should be running and `False` indicates that it should be stopped.

ntp_servers A list of servers that should be added to the ESXi host's NTP configuration.

service_policy The policy to set for the NTP service.

Note: When setting the service policy to `off` or `on`, you *must* quote the setting. If you don't, the yaml parser will set the string to a boolean, which will cause trouble checking for stateful changes and will error when trying to set the policy on the ESXi host.

service_restart If set to `True`, the ntp daemon will be restarted, regardless of its previous running state. Default is `False`.

update_datetime If set to `True`, the date/time on the given host will be updated to UTC. Default setting is `False`. This option should be used with caution since network delays and execution delays can result in time skews.

Example:

```
configure-host-ntp:
  esxi.ntp_configured:
    - service_running: True
    - ntp_servers:
      - 192.174.1.100
      - 192.174.1.200
    - service_policy: 'on'
    - service_restart: True
```

`salt.states.esxi.`**`password_present`**(*name*, *password*)

Ensures the given password is set on the ESXi host. Passwords cannot be obtained from host, so if a password is set in this state, the `vsphere.update_host_password` function will always run (except when using test=True functionality) and the state's changes dictionary will always be populated.

The username for which the password will change is the same username that is used to authenticate against the ESXi host via the Proxy Minion. For example, if the pillar definition for the proxy username is defined as `root`, then the username that the password will be updated for via this state is `root`.

name Name of the state.

password The new password to change on the host.

Example:

```
configure-host-password:
  esxi.password_present:
    - password: 'new-bad-password'
```

`salt.states.esxi.`**`ssh_configured`**`(`*name, service_running, ssh_key=None, ssh_key_file=None, service_policy=None, service_restart=False, certificate_verify=False*`)`

Manage the SSH configuration for a host including whether or not SSH is running or the presence of a given SSH key. Note: Only one ssh key can be uploaded for root. Uploading a second key will replace any existing key.

name Name of the state.

service_running Ensures whether or not the SSH service should be running on a host. Represented as a boolean value where `True` indicates that SSH should be running and `False` indicates that SSH should stopped.

In order to update SSH keys, the SSH service must be running.

ssh_key Public SSH key to added to the authorized_keys file on the ESXi host. You can use `ssh_key` or `ssh_key_file`, but not both.

ssh_key_file File containing the public SSH key to be added to the authorized_keys file on the ESXi host. You can use `ssh_key_file` or `ssh_key`, but not both.

service_policy The policy to set for the NTP service.

Note: When setting the service policy to `off` or `on`, you *must* quote the setting. If you don't, the yaml parser will set the string to a boolean, which will cause trouble checking for stateful changes and will error when trying to set the policy on the ESXi host.

service_restart If set to `True`, the SSH service will be restarted, regardless of its previous running state. Default is `False`.

certificate_verify If set to `True`, the SSL connection must present a valid certificate. Default is `False`.

Example:

```
configure-host-ssh:
  esxi.ssh_configured:
    - service_running: True
    - ssh_key_file: /etc/salt/ssh_keys/my_key.pub
    - service_policy: 'on'
    - service_restart: True
    - certificate_verify: True
```

`salt.states.esxi.`**`syslog_configured`**`(`*name, syslog_configs, firewall=True, reset_service=True, reset_syslog_config=False, reset_configs=None*`)`

Ensures the specified syslog configuration parameters. By default, this state will reset the syslog service after any new or changed parameters are set successfully.

name Name of the state.

syslog_configs Name of parameter to set (corresponds to the command line switch for esxcli without the double dashes (–))

Valid syslog_config values are `logdir`, `loghost`, `logdir-unique`, `default-rotate`, `default-size`, and `default-timeout`.

Each syslog_config option also needs a configuration value to set. For example, `loghost` requires URLs or IP addresses to use for logging. Multiple log servers can be specified by listing them, comma-separated, but without spaces before or after commas

(reference: https://blogs.vmware.com/vsphere/2012/04/configuring-multiple-syslog-servers-for-esxi-5.html)

firewall Enable the firewall rule set for syslog. Defaults to `True`.

reset_service After a successful parameter set, reset the service. Defaults to `True`.

reset_syslog_config Resets the syslog service to it's default settings. Defaults to `False`. If set to `True`, default settings defined by the list of syslog configs in `reset_configs` will be reset before running any other syslog settings.

reset_configs A comma-delimited list of parameters to reset. Only runs if `reset_syslog_config` is set to `True`. If `reset_syslog_config` is set to `True`, but no syslog configs are listed in `reset_configs`, then `reset_configs` will be set to `all` by default.

See `syslog_configs` parameter above for a list of valid options.

Example:

```
configure-host-syslog:
  esxi.syslog_configured:
    - syslog_configs:
        loghost: ssl://localhost:5432,tcp://10.1.0.1:1514
        default-timeout: 120
    - firewall: True
    - reset_service: True
    - reset_syslog_config: True
    - reset_configs: loghost,default-timeout
```

`salt.states.esxi.`**`vmotion_configured`**(*name*, *enabled*, *device='vmk0'*)

Configures a host's VMotion properties such as enabling VMotion and setting the device VirtualNic that VMotion will use.

name Name of the state.

enabled Ensures whether or not VMotion should be enabled on a host as a boolean value where `True` indicates that VMotion should be enabled and `False` indicates that VMotion should be disabled.

device The device that uniquely identifies the VirtualNic that will be used for VMotion for the host. Defaults to `vmk0`.

Example:

```
configure-vmotion:
  esxi.vmotion_configured:
    - enabled: True
    - device: sample-device
```

`salt.states.esxi.`**`vsan_configured`**(*name*, *enabled*, *add_disks_to_vsan=False*)

Configures a host's VSAN properties such as enabling or disabling VSAN, or adding VSAN-eligible disks to the VSAN system for the host.

name Name of the state.

enabled Ensures whether or not VSAN should be enabled on a host as a boolean value where `True` indicates that VSAN should be enabled and `False` indicates that VSAN should be disabled.

add_disks_to_vsan If set to `True`, any VSAN-eligible disks for the given host will be added to the host's VSAN system. Default is `False`.

Example:

```
configure-host-vsan:
  esxi.vsan_configured:
    - enabled: True
    - add_disks_to_vsan: True
```

21.19.84 salt.states.event

Send events through Salt's event system during state runs

`salt.states.event.`**`send`**`(`*name, data=None, preload=None, with_env=False, with_grains=False, with_pillar=False, show_changed=True, **kwargs*`)`

Send an event to the Salt Master

New in version 2014.7.0.

Accepts the same arguments as the *`event.send`* execution module of the same name, with the additional argument:

Parameters **`show_changed`** – If `True`, state will show as changed with the data argument as the change value. If `False`, shows as unchanged.

Example:

```
# ...snip bunch of states above

mycompany/mystaterun/status/update:
  event.send:
    - data:
        status: "Half-way through the state run!"

# ...snip bunch of states below
```

`salt.states.event.`**`wait`**`(`*name, sfun=None*`)`

Fire an event on the Salt master event bus if called from a watch statement

New in version 2014.7.0.

Example:

```
# Stand up a new web server.
apache:
  pkg:
    - installed
    - name: httpd
  service:
    - running
    - enable: True
    - name: httpd

# Notify the load balancer to update the pool once Apache is running.
refresh_pool:
  event:
    - wait
    - name: mycompany/loadbalancer/pool/update
    - data:
        new_web_server_ip: {{ grains['ipv4'] | first() }}
    - watch:
      - pkg: apache
```

21.19.85 salt.states.file

Operations on regular files, special files, directories, and symlinks

Salt States can aggressively manipulate files on a system. There are a number of ways in which files can be managed.

Regular files can be enforced with the *file.managed* state. This state downloads files from the salt master and places them on the target system. Managed files can be rendered as a jinja, mako, or wempy template, adding a dynamic component to file management. An example of *file.managed* which makes use of the jinja templating system would look like this:

```
/etc/http/conf/http.conf:
  file.managed:
    - source: salt://apache/http.conf
    - user: root
    - group: root
    - mode: 644
    - attrs: ai
    - template: jinja
    - defaults:
        custom_var: "default value"
        other_var: 123
{% if grains['os'] == 'Ubuntu' %}
    - context:
        custom_var: "override"
{% endif %}
```

It is also possible to use the *py renderer* as a templating option. The template would be a Python script which would need to contain a function called `run()`, which returns a string. All arguments to the state will be made available to the Python script as globals. The returned string will be the contents of the managed file. For example:

```
def run():
    lines = ['foo', 'bar', 'baz']
    lines.extend([source, name, user, context])  # Arguments as globals
    return '\n\n'.join(lines)
```

Note: The `defaults` and `context` arguments require extra indentation (four spaces instead of the normal two) in order to create a nested dictionary. *More information*.

If using a template, any user-defined template variables in the file defined in `source` must be passed in using the `defaults` and/or `context` arguments. The general best practice is to place default values in `defaults`, with conditional overrides going into `context`, as seen above.

The template will receive a variable `custom_var`, which would be accessed in the template using `{{ custom_var }}`. If the operating system is Ubuntu, the value of the variable `custom_var` would be *override*, otherwise it is the default *default value*

The `source` parameter can be specified as a list. If this is done, then the first file to be matched will be the one that is used. This allows you to have a default file on which to fall back if the desired file does not exist on the salt fileserver. Here's an example:

```
/etc/foo.conf:
  file.managed:
    - source:
      - salt://foo.conf.{{ grains['fqdn'] }}
      - salt://foo.conf.fallback
    - user: foo
    - group: users
    - mode: 644
    - attrs: i
    - backup: minion
```

Note: Salt supports backing up managed files via the backup option. For more details on this functionality please review the *backup_mode documentation*.

The `source` parameter can also specify a file in another Salt environment. In this example `foo.conf` in the `dev` environment will be used instead.

```
/etc/foo.conf:
  file.managed:
    - source:
      - 'salt://foo.conf?saltenv=dev'
    - user: foo
    - group: users
    - mode: '0644'
    - attrs: i
```

Warning: When using a mode that includes a leading zero you must wrap the value in single quotes. If the value is not wrapped in quotes it will be read by YAML as an integer and evaluated as an octal.

The `names` parameter, which is part of the state compiler, can be used to expand the contents of a single state declaration into multiple, single state declarations. Each item in the `names` list receives its own individual state `name` and is converted into its own low-data structure. This is a convenient way to manage several files with similar attributes.

```
salt_master_conf:
  file.managed:
    - user: root
    - group: root
    - mode: '0644'
    - names:
      - /etc/salt/master.d/master.conf:
        - source: salt://saltmaster/master.conf
      - /etc/salt/minion.d/minion-99.conf:
        - source: salt://saltmaster/minion.conf
```

Note: There is more documentation about this feature in the *Names declaration* section of the *Highstate docs*.

Special files can be managed via the `mknod` function. This function will create and enforce the permissions on a special file. The function supports the creation of character devices, block devices, and FIFO pipes. The function will create the directory structure up to the special file if it is needed on the minion. The function will not overwrite or operate on (change major/minor numbers) existing special files with the exception of user, group, and permissions. In most cases the creation of some special files require root permissions on the minion. This would require that the minion to be run as the root user. Here is an example of a character device:

```
/var/named/chroot/dev/random:
  file.mknod:
    - ntype: c
    - major: 1
    - minor: 8
    - user: named
    - group: named
    - mode: 660
```

Here is an example of a block device:

```
/var/named/chroot/dev/loop0:
  file.mknod:
    - ntype: b
    - major: 7
    - minor: 0
    - user: named
    - group: named
    - mode: 660
```

Here is an example of a fifo pipe:

```
/var/named/chroot/var/log/logfifo:
  file.mknod:
    - ntype: p
    - user: named
    - group: named
    - mode: 660
```

Directories can be managed via the `directory` function. This function can create and enforce the permissions on a directory. A directory statement will look like this:

```
/srv/stuff/substuf:
  file.directory:
    - user: fred
    - group: users
    - mode: 755
    - makedirs: True
```

If you need to enforce user and/or group ownership or permissions recursively on the directory's contents, you can do so by adding a `recurse` directive:

```
/srv/stuff/substuf:
  file.directory:
    - user: fred
    - group: users
    - mode: 755
    - makedirs: True
    - recurse:
      - user
      - group
      - mode
```

As a default, `mode` will resolve to `dir_mode` and `file_mode`, to specify both directory and file permissions, use this form:

```
/srv/stuff/substuf:
  file.directory:
    - user: fred
    - group: users
    - file_mode: 744
    - dir_mode: 755
    - makedirs: True
    - recurse:
      - user
      - group
      - mode
```

Symlinks can be easily created; the symlink function is very simple and only takes a few arguments:

```
/etc/grub.conf:
  file.symlink:
    - target: /boot/grub/grub.conf
```

Recursive directory management can also be set via the `recurse` function. Recursive directory management allows for a directory on the salt master to be recursively copied down to the minion. This is a great tool for deploying large code and configuration systems. A state using `recurse` would look something like this:

```
/opt/code/flask:
  file.recurse:
    - source: salt://code/flask
    - include_empty: True
```

A more complex `recurse` example:

```
{% set site_user = 'testuser' %}
{% set site_name = 'test_site' %}
{% set project_name = 'test_proj' %}
{% set sites_dir = 'test_dir' %}

django-project:
  file.recurse:
    - name: {{ sites_dir }}/{{ site_name }}/{{ project_name }}
    - user: {{ site_user }}
    - dir_mode: 2775
    - file_mode: '0644'
    - template: jinja
    - source: salt://project/templates_dir
    - include_empty: True
```

Retention scheduling can be applied to manage contents of backup directories. For example:

```
/var/backups/example_directory:
  file.retention_schedule:
    - strptime_format: example_name_%Y%m%dT%H%M%S.tar.bz2
    - retain:
        most_recent: 5
        first_of_hour: 4
        first_of_day: 14
        first_of_week: 6
        first_of_month: 6
        first_of_year: all
```

`salt.states.file.`**`absent`**(*name*, ***kwargs*)

Make sure that the named file or directory is absent. If it exists, it will be deleted. This will work to reverse any of the functions in the file state module. If a directory is supplied, it will be recursively deleted.

name The path which should be deleted

`salt.states.file.`**`accumulated`**(*name*, *filename*, *text*, ***kwargs*)

Prepare accumulator which can be used in template in file.managed state. Accumulator dictionary becomes available in template. It can also be used in file.blockreplace.

name Accumulator name

filename Filename which would receive this accumulator (see file.managed state documentation about `name`)

text String or list for adding in accumulator

require_in / watch_in One of them required for sure we fill up accumulator before we manage the file. Probably the same as filename

Example:

Given the following:

```
animals_doing_things:
  file.accumulated:
    - filename: /tmp/animal_file.txt
    - text: ' jumps over the lazy dog.'
    - require_in:
      - file: animal_file

animal_file:
  file.managed:
    - name: /tmp/animal_file.txt
    - source: salt://animal_file.txt
    - template: jinja
```

One might write a template for `animal_file.txt` like the following:

```
The quick brown fox{% for animal in accumulator['animals_doing_things'] %}{{
↪animal }}{% endfor %}
```

Collectively, the above states and template file will produce:

```
The quick brown fox jumps over the lazy dog.
```

Multiple accumulators can be "chained" together.

Note: The 'accumulator' data structure is a Python dictionary. Do not expect any loop over the keys in a deterministic order!

`salt.states.file.`**`append`**`(`*name, text=None, makedirs=False, source=None, source_hash=None, template='jinja', sources=None, source_hashes=None, defaults=None, context=None, ignore_whitespace=True*`)`

Ensure that some text appears at the end of a file.

The text will not be appended if it already exists in the file. A single string of text or a list of strings may be appended.

name The location of the file to append to.

text The text to be appended, which can be a single string or a list of strings.

makedirs If the file is located in a path without a parent directory, then the state will fail. If makedirs is set to True, then the parent directories will be created to facilitate the creation of the named file. Defaults to False.

source A single source file to append. This source file can be hosted on either the salt master server, or on an HTTP or FTP server. Both HTTPS and HTTP are supported as well as downloading directly from Amazon S3 compatible URLs with both pre-configured and automatic IAM credentials (see s3.get state documentation). File retrieval from Openstack Swift object storage is supported via swift://container/object_path URLs (see swift.get documentation).

For files hosted on the salt file server, if the file is located on the master in the directory named spam, and is called eggs, the source string is salt://spam/eggs.

If the file is hosted on an HTTP or FTP server, the source_hash argument is also required.

source_hash

This can be one of the following:

1. a source hash string
2. the URI of a file that contains source hash strings

The function accepts the first encountered long unbroken alphanumeric string of correct length as a valid hash, in order from most secure to least secure:

```
Type    Length
======  ======
sha512     128
sha384      96
sha256      64
sha224      56
sha1        40
md5         32
```

See the `source_hash` parameter description for *`file.managed`* function for more details and examples.

template The named templating engine will be used to render the appended-to file. Defaults to `jinja`. The following templates are supported:

- *`cheetah`*
- *`genshi`*
- *`jinja`*
- *`mako`*
- *`py`*
- *`wempy`*

sources A list of source files to append. If the files are hosted on an HTTP or FTP server, the source_hashes argument is also required.

source_hashes A list of source_hashes corresponding to the sources list specified in the sources argument.

defaults Default context passed to the template.

context Overrides default context variables passed to the template.

ignore_whitespace New in version 2015.8.4.

Spaces and Tabs in text are ignored by default, when searching for the appending content, one space or multiple tabs are the same for salt. Set this option to `False` if you want to change this behavior.

Multi-line example:

```
/etc/motd:
  file.append:
    - text: |
        Thou hadst better eat salt with the Philosophers of Greece,
        than sugar with the Courtiers of Italy.
        - Benjamin Franklin
```

Multiple lines of text:

```
/etc/motd:
  file.append:
    - text:
      - Trust no one unless you have eaten much salt with him.
      - "Salt is born of the purest of parents: the sun and the sea."
```

Gather text from multiple template files:

```
/etc/motd:
  file:
    - append
    - template: jinja
    - sources:
      - salt://motd/devops-messages.tmpl
```

(continues on next page)

(continued from previous page)

```
      - salt://motd/hr-messages.tmpl
      - salt://motd/general-messages.tmpl
```

New in version 0.9.5.

`salt.states.file.`**`blockreplace`**`(`*name, marker_start='#– start managed zone –', marker_end='#– end managed zone –', source=None, source_hash=None, template='jinja', sources=None, source_hashes=None, defaults=None, context=None, content='', append_if_not_found=False, prepend_if_not_found=False, backup='.bak', show_changes=True, append_newline=None*`)`

Maintain an edit in a file in a zone delimited by two line markers

New in version 2014.1.0.

Changed in version 2017.7.5,2018.3.1: `append_newline` argument added. Additionally, to improve idempotence, if the string represented by `marker_end` is found in the middle of the line, the content preceding the marker will be removed when the block is replaced. This allows one to remove `append_newline: False` from the SLS and have the block properly replaced if the end of the content block is immediately followed by the `marker_end` (i.e. no newline before the marker).

A block of content delimited by comments can help you manage several lines entries without worrying about old entries removal. This can help you maintaining an un-managed file containing manual edits.

Note: This function will store two copies of the file in-memory (the original version and the edited version) in order to detect changes and only edit the targeted file if necessary.

Additionally, you can use *`file.accumulated`* and target this state. All accumulated data dictionaries' content will be added in the content block.

---

name Filesystem path to the file to be edited

marker_start The line content identifying a line as the start of the content block. Note that the whole line containing this marker will be considered, so whitespace or extra content before or after the marker is included in final output

marker_end The line content identifying the end of the content block. As of versions 2017.7.5 and 2018.3.1, everything up to the text matching the marker will be replaced, so it's important to ensure that your marker includes the beginning of the text you wish to replace.

content The content to be used between the two lines identified by `marker_start` and `marker_end`

source The source file to download to the minion, this source file can be hosted on either the salt master server, or on an HTTP or FTP server. Both HTTPS and HTTP are supported as well as downloading directly from Amazon S3 compatible URLs with both pre-configured and automatic IAM credentials. (see s3.get state documentation) File retrieval from Openstack Swift object storage is supported via swift://container/object_path URLs, see swift.get documentation. For files hosted on the salt file server, if the file is located on the master in the directory named spam, and is called eggs, the source string is salt://spam/eggs. If source is left blank or None (use ~ in YAML), the file will be created as an empty file and the content will not be managed. This is also the case when a file already exists and the source is undefined; the contents of the file will not be changed or managed.

If the file is hosted on a HTTP or FTP server then the source_hash argument is also required.

A list of sources can also be passed in to provide a default source and a set of fallbacks. The first source in the list that is found to exist will be used and subsequent entries in the list will be ignored.

```
file_override_example:
  file.blockreplace:
```

(continues on next page)

(continued from previous page)

```
    - name: /etc/example.conf
    - source:
      - salt://file_that_does_not_exist
      - salt://file_that_exists
```

source_hash

This can be one of the following:

1. a source hash string
2. the URI of a file that contains source hash strings

The function accepts the first encountered long unbroken alphanumeric string of correct length as a valid hash, in order from most secure to least secure:

```
Type    Length
======  ======
sha512     128
sha384      96
sha256      64
sha224      56
sha1        40
md5         32
```

See the `source_hash` parameter description for *`file.managed`* function for more details and examples.

template [jinja] Templating engine to be used to render the downloaded file. The following engines are supported:

- *`cheetah`*
- *`genshi`*
- *`jinja`*
- *`mako`*
- *`py`*
- *`wempy`*

context Overrides default context variables passed to the template

defaults Default context passed to the template

append_if_not_found [False] If markers are not found and this option is set to `True`, the content block will be appended to the file.

prepend_if_not_found [False] If markers are not found and this option is set to `True`, the content block will be prepended to the file.

backup The file extension to use for a backup of the file if any edit is made. Set this to `False` to skip making a backup.

dry_run [False] If `True`, do not make any edits to the file and simply return the changes that *would* be made.

show_changes [True] Controls how changes are presented. If `True`, the `Changes` section of the state return will contain a unified diff of the changes made. If False, then it will contain a boolean (`True` if any changes were made, otherwise `False`).

append_newline Controls whether or not a newline is appended to the content block. If the value of this argument is `True` then a newline will be added to the content block. If it is `False`, then a newline will *not* be added to the content block. If it is unspecified, then a newline will only be added to the content block if it does not already end in a newline.

New in version 2017.7.5,2018.3.1.

Example of usage with an accumulator and with a variable:

```
{% set myvar = 42 %}
hosts-config-block-{{ myvar }}:
  file.blockreplace:
```

(continues on next page)

(continued from previous page)

```
    - name: /etc/hosts
    - marker_start: "# START managed zone {{ myvar }} -DO-NOT-EDIT-"
    - marker_end: "# END managed zone {{ myvar }} --"
    - content: 'First line of content'
    - append_if_not_found: True
    - backup: '.bak'
    - show_changes: True

hosts-config-block-{{ myvar }}-accumulated1:
  file.accumulated:
    - filename: /etc/hosts
    - name: my-accumulator-{{ myvar }}
    - text: "text 2"
    - require_in:
      - file: hosts-config-block-{{ myvar }}

hosts-config-block-{{ myvar }}-accumulated2:
  file.accumulated:
    - filename: /etc/hosts
    - name: my-accumulator-{{ myvar }}
    - text: |
         text 3
         text 4
    - require_in:
      - file: hosts-config-block-{{ myvar }}
```

will generate and maintain a block of content in `/etc/hosts`:

```
# START managed zone 42 -DO-NOT-EDIT-
First line of content
text 2
text 3
text 4
# END managed zone 42 --
```

`salt.states.file.`**`cached`**`(`*name, source_hash='', source_hash_name=None, skip_verify=False, saltenv='base'*`)`

New in version 2017.7.3.

Ensures that a file is saved to the minion's cache. This state is primarily invoked by other states to ensure that we do not re-download a source file if we do not need to.

name The URL of the file to be cached. To cache a file from an environment other than `base`, either use the `saltenv` argument or include the saltenv in the URL (e.g. `salt://path/to/file.conf?saltenv=dev`).

> **Note:** A list of URLs is not supported, this must be a single URL. If a local file is passed here, then the state will obviously not try to download anything, but it will compare a hash if one is specified.

source_hash See the documentation for this same argument in the `file.managed` state.

> **Note:** For remote files not originating from the `salt://` fileserver, such as http(s) or ftp servers, this state will not re-download the file if the locally-cached copy matches this hash. This is done to prevent unnecessary downloading on repeated runs of this state. To update the cached copy of a file, it is necessary to update this hash.

source_hash_name See the documentation for this same argument in the *file.managed* state.
skip_verify See the documentation for this same argument in the *file.managed* state.

> **Note:** Setting this to `True` will result in a copy of the file being downloaded from a remote (http(s), ftp, etc.) source each time the state is run.

saltenv Used to specify the environment from which to download a file from the Salt fileserver (i.e. those with `salt://` URL).

This state will in most cases not be useful in SLS files, but it is useful when writing a state or remote-execution module that needs to make sure that a file at a given URL has been downloaded to the cachedir. One example of this is in the `archive.extracted` state:

```
result = __states__['file.cached'](source_match,
                                   source_hash=source_hash,
                                   source_hash_name=source_hash_name,
                                   skip_verify=skip_verify,
                                   saltenv=__env__)
```

This will return a dictionary containing the state's return data, including a `result` key which will state whether or not the state was successful. Note that this will not catch exceptions, so it is best used within a try/except.

Once this state has been run from within another state or remote-execution module, the actual location of the cached file can be obtained using *cp.is_cached*:

```
cached = __salt__['cp.is_cached'](source_match, saltenv=__env__)
```

This function will return the cached path of the file, or an empty string if the file is not present in the minion cache.

`salt.states.file.`**`comment`**(*name, regex, char='#', backup='.bak'*)

Comment out specified lines in a file.

name The full path to the file to be edited

regex A regular expression used to find the lines that are to be commented; this pattern will be wrapped in parenthesis and will move any preceding/trailing `^` or `$` characters outside the parenthesis (e.g., the pattern `^foo$` will be rewritten as `^(foo)$`) Note that you _need_ the leading ^, otherwise each time you run highstate, another comment char will be inserted.

char [#] The character to be inserted at the beginning of a line in order to comment it out

backup [`.bak`] The file will be backed up before edit with this file extension

> **Warning:** This backup will be overwritten each time `sed` / `comment` / `uncomment` is called. Meaning the backup will only be useful after the first invocation.

Set to False/None to not keep a backup.

Usage:

```
/etc/fstab:
  file.comment:
    - regex: ^bind 127.0.0.1
```

New in version 0.9.5.

`salt.states.file.`**`copy_`**(*name, source, force=False, makedirs=False, preserve=False, user=None, group=None, mode=None, subdir=False, **kwargs*)

If the file defined by the `source` option exists on the minion, copy it to the named path. The file will not be overwritten if it already exists, unless the `force` option is set to `True`.

Note: This state only copies files from one location on a minion to another location on the same minion. For copying files from the master, use a *`file.managed`* state.

---

name The location of the file to copy to

source The location of the file to copy to the location specified with name

force If the target location is present then the file will not be moved, specify "force: True" to overwrite the target file

makedirs If the target subdirectories don't exist create them

preserve New in version 2015.5.0.

Set `preserve: True` to preserve user/group ownership and mode after copying. Default is `False`. If `preserve` is set to `True`, then user/group/mode attributes will be ignored.

user New in version 2015.5.0.

The user to own the copied file, this defaults to the user salt is running as on the minion. If `preserve` is set to `True`, then this will be ignored

group New in version 2015.5.0.

The group to own the copied file, this defaults to the group salt is running as on the minion. If `preserve` is set to `True` or on Windows this will be ignored

mode New in version 2015.5.0.

The permissions to set on the copied file, aka 644, '0775', '4664'. If `preserve` is set to `True`, then this will be ignored. Not supported on Windows.

The default mode for new files and directories corresponds umask of salt process. For existing files and directories it's not enforced.

subdir New in version 2015.5.0.

If the name is a directory then place the file inside the named directory

---

Note: The copy function accepts paths that are local to the Salt minion. This function does not support salt://, http://, or the other additional file paths that are supported by *`states.file.managed`* and *`states.file.recurse`*.

`salt.states.file.`**`decode`**`(`*name, encoded_data=None, contents_pillar=None, encoding_type='base64', checksum='md5'*`)`

Decode an encoded file and write it to disk

New in version 2016.3.0.

name Path of the file to be written.

encoded_data The encoded file. Either this option or `contents_pillar` must be specified.

contents_pillar A Pillar path to the encoded file. Uses the same path syntax as *`pillar.get`*. The *`hashutil.base64_encodefile`* function can load encoded content into Pillar. Either this option or `encoded_data` must be specified.

encoding_type [`base64`] The type of encoding.

checksum [`md5`] The hashing algorithm to use to generate checksums. Wraps the *`hashutil.digest`* execution function.

Usage:

```
write_base64_encoded_string_to_a_file:
  file.decode:
    - name: /tmp/new_file
```

(continues on next page)

(continued from previous page)

```
    - encoding_type: base64
    - contents_pillar: mypillar:thefile

# or

write_base64_encoded_string_to_a_file:
  file.decode:
    - name: /tmp/new_file
    - encoding_type: base64
    - encoded_data: |
        Z2V0IHNhbHRlZAo=
```

Be careful with multi-line strings that the YAML indentation is correct. E.g.,

```
write_base64_encoded_string_to_a_file:
  file.decode:
    - name: /tmp/new_file
    - encoding_type: base64
    - encoded_data: |
        {{ salt.pillar.get('path:to:data') | indent(8) }}
```

`salt.states.file.`**`directory`**`(`*name, user=None, group=None, recurse=None, max_depth=None, dir_mode=None, file_mode=None, makedirs=False, clean=False, require=None, exclude_pat=None, follow_symlinks=False, force=False, backupname=None, allow_symlink=True, children_only=False, win_owner=None, win_perms=None, win_deny_perms=None, win_inheritance=True, win_perms_reset=False, **kwargs*`)`

Ensure that a named directory is present and has the right perms

name The location to create or manage a directory, as an absolute path

user The user to own the directory; this defaults to the user salt is running as on the minion

group The group ownership set for the directory; this defaults to the group salt is running as on the minion. On Windows, this is ignored

recurse Enforce user/group ownership and mode of directory recursively. Accepts a list of strings representing what you would like to recurse. If `mode` is defined, will recurse on both `file_mode` and `dir_mode` if they are defined. If `ignore_files` or `ignore_dirs` is included, files or directories will be left unchanged respectively. Example:

```
/var/log/httpd:
  file.directory:
    - user: root
    - group: root
    - dir_mode: 755
    - file_mode: 644
    - recurse:
      - user
      - group
      - mode
```

Leave files or directories unchanged:

```
/var/log/httpd:
  file.directory:
    - user: root
    - group: root
    - dir_mode: 755
```

(continues on next page)

(continued from previous page)

```
    - file_mode: 644
    - recurse:
      - user
      - group
      - mode
      - ignore_dirs
```

New in version 2015.5.0.

max_depth Limit the recursion depth. The default is no limit=None. 'max_depth' and 'clean' are mutually exclusive.

New in version 2016.11.0.

dir_mode / mode The permissions mode to set any directories created. Not supported on Windows.

The default mode for new files and directories corresponds umask of salt process. For existing files and directories it's not enforced.

file_mode The permissions mode to set any files created if 'mode' is run in 'recurse'. This defaults to dir_mode. Not supported on Windows.

The default mode for new files and directories corresponds umask of salt process. For existing files and directories it's not enforced.

makedirs If the directory is located in a path without a parent directory, then the state will fail. If makedirs is set to True, then the parent directories will be created to facilitate the creation of the named file.

clean Make sure that only files that are set up by salt and required by this function are kept. If this option is set then everything in this directory will be deleted unless it is required. 'clean' and 'max_depth' are mutually exclusive.

require Require other resources such as packages or files

exclude_pat When 'clean' is set to True, exclude this pattern from removal list and preserve in the destination.

follow_symlinks [False] If the desired path is a symlink (or `recurse` is defined and a symlink is encountered while recursing), follow it and check the permissions of the directory/file to which the symlink points.

New in version 2014.1.4.

force If the name of the directory exists and is not a directory and force is set to False, the state will fail. If force is set to True, the file in the way of the directory will be deleted to make room for the directory, unless backupname is set, then it will be renamed.

New in version 2014.7.0.

backupname If the name of the directory exists and is not a directory, it will be renamed to the backupname. If the backupname already exists and force is False, the state will fail. Otherwise, the backupname will be removed first.

New in version 2014.7.0.

allow_symlink [True] If allow_symlink is True and the specified path is a symlink, it will be allowed to remain if it points to a directory. If allow_symlink is False then the state will fail, unless force is also set to True, in which case it will be removed or renamed, depending on the value of the backupname argument.

New in version 2014.7.0.

children_only [False] If children_only is True the base of a path is excluded when performing a recursive operation. In case of /path/to/base, base will be ignored while all of /path/to/base/* are still operated on.

win_owner [None] The owner of the directory. If this is not passed, user will be used. If user is not passed, the account under which Salt is running will be used.

New in version 2017.7.0.

win_perms [None] A dictionary containing permissions to grant and their propagation. For example: `{'Administrators': {'perms': 'full_control', 'applies_to': 'this_folder_only'}}` Can be a single basic perm or a list of advanced perms. `perms` must be specified. `applies_to` is optional and defaults to `this_folder_subfolder_files`.

New in version 2017.7.0.

win_deny_perms [None] A dictionary containing permissions to deny and their propagation. For example: `{'Administrators': {'perms': 'full_control', 'applies_to': 'this_folder_only'}}` Can be a single basic perm or a list of advanced perms.

New in version 2017.7.0.

win_inheritance [True] True to inherit permissions from the parent directory, False not to inherit permission.

New in version 2017.7.0.

win_perms_reset [False] If `True` the existing DACL will be cleared and replaced with the settings defined in this function. If `False`, new entries will be appended to the existing DACL. Default is `False`.

New in version 2018.3.0.

Here's an example using the above `win_*` parameters:

```
create_config_dir:
  file.directory:
    - name: 'C:\config\'
    - win_owner: Administrators
    - win_perms:
        # Basic Permissions
        dev_ops:
          perms: full_control
        # List of advanced permissions
        appuser:
          perms:
            - read_attributes
            - read_ea
            - create_folders
            - read_permissions
          applies_to: this_folder_only
        joe_snuffy:
          perms: read
          applies_to: this_folder_files
    - win_deny_perms:
        fred_snuffy:
          perms: full_control
    - win_inheritance: False
```

`salt.states.file.`**`exists`**(*name*, ***kwargs*)

Verify that the named file or directory is present or exists. Ensures pre-requisites outside of Salt's purview (e.g., keytabs, private keys, etc.) have been previously satisfied before deployment.

This function does not create the file if it doesn't exist, it will return an error.

name Absolute path which must exist

`salt.states.file.`**`hardlink`**(*name*, *target*, *force=False*, *makedirs=False*, *user=None*, *group=None*, *dir_mode=None*, ***kwargs*)

Create a hard link If the file already exists and is a hard link pointing to any location other than the specified target, the hard link will be replaced. If the hard link is a regular file or directory then the state will return False. If the regular file is desired to be replaced with a hard link pass force: True

name The location of the hard link to create

target The location that the hard link points to

force If the name of the hard link exists and force is set to False, the state will fail. If force is set to True, the file or directory in the way of the hard link file will be deleted to make room for the hard link, unless backupname is set, when it will be renamed

makedirs If the location of the hard link does not already have a parent directory then the state will fail, setting makedirs to True will allow Salt to create the parent directory

user The user to own any directories made if makedirs is set to true. This defaults to the user salt is running as on the minion

group The group ownership set on any directories made if makedirs is set to true. This defaults to the group salt is running as on the minion. On Windows, this is ignored

dir_mode If directories are to be created, passing this option specifies the permissions for those directories.

`salt.states.file.line`(*name*, *content=None*, *match=None*, *mode=None*, *location=None*, *before=None*, *after=None*, *show_changes=True*, *backup=False*, *quiet=False*, *indent=True*, *create=False*, *user=None*, *group=None*, *file_mode=None*)

Line-based editing of a file.

New in version 2015.8.0.

Parameters

- **`name`** – Filesystem path to the file to be edited.
- **`content`** – Content of the line. Allowed to be empty if mode=delete.
- **`match`** – Match the target line for an action by a fragment of a string or regular expression.

  If neither `before` nor `after` are provided, and `match` is also `None`, match becomes the `content` value.
- **`mode`** – Defines how to edit a line. One of the following options is required:
  - **ensure** If line does not exist, it will be added.
  - **replace** If line already exists, it will be replaced.
  - **delete** Delete the line, once found.
  - **insert** Insert a line.

  Note: If `mode=insert` is used, at least one of the following options must also be defined: `location`, `before`, or `after`. If `location` is used, it takes precedence over the other two options.
- **`location`** – Defines where to place content in the line. Note this option is only used when `mode=insert` is specified. If a location is passed in, it takes precedence over both the `before` and `after` kwargs. Valid locations are:
  - **start** Place the content at the beginning of the file.
  - **end** Place the content at the end of the file.
- **`before`** – Regular expression or an exact case-sensitive fragment of the string. This option is only used when either the `ensure` or `insert` mode is defined.
- **`after`** – Regular expression or an exact case-sensitive fragment of the string. This option is only used when either the `ensure` or `insert` mode is defined.
- **`show_changes`** – Output a unified diff of the old file and the new file. If `False` return a boolean if any changes were made. Default is `True`

  Note: Using this option will store two copies of the file in-memory (the original version and the edited version) in order to generate the diff.
- **`backup`** – Create a backup of the original file with the extension: "Year-Month-Day-Hour-Minutes-Seconds".
- **`quiet`** – Do not raise any exceptions. E.g. ignore the fact that the file that is tried to be edited does not exist and nothing really happened.
- **`indent`** – Keep indentation with the previous line. This option is not considered when the `delete` mode is specified.
- **`create`** – Create an empty file if doesn't exists.

  New in version 2016.11.0.
- **`user`** – The user to own the file, this defaults to the user salt is running as on the minion.

New in version 2016.11.0.

- **group** – The group ownership set for the file, this defaults to the group salt is running as on the minion On Windows, this is ignored.

  New in version 2016.11.0.

- **file_mode** – The permissions to set on this file, aka 644, 0775, 4664. Not supported on Windows.

  New in version 2016.11.0.

If an equal sign (=) appears in an argument to a Salt command, it is interpreted as a keyword argument in the format of `key=val`. That processing can be bypassed in order to pass an equal sign through to the remote shell command by manually specifying the kwarg:

```
update_config:
  file.line:
    - name: /etc/myconfig.conf
    - mode: ensure
    - content: my key = my value
    - before: somekey.*?
```

`salt.states.file.`**`managed`**(*name, source=None, source_hash=”, source_hash_name=None, keep_source=True, user=None, group=None, mode=None, attrs=None, template=None, makedirs=False, dir_mode=None, context=None, replace=True, defaults=None, backup=”, show_changes=True, create=True, contents=None, tmp_dir=”, tmp_ext=”, contents_pillar=None, contents_grains=None, contents_newline=True, contents_delimiter=’:’, encoding=None, encoding_errors=’strict’, allow_empty=True, follow_symlinks=True, check_cmd=None, skip_verify=False, win_owner=None, win_perms=None, win_deny_perms=None, win_inheritance=True, win_perms_reset=False, **kwargs*)

Manage a given file, this function allows for a file to be downloaded from the salt master and potentially run through a templating system.

name The location of the file to manage, as an absolute path.

source The source file to download to the minion, this source file can be hosted on either the salt master server (`salt://`), the salt minion local file system (`/`), or on an HTTP or FTP server (`http(s)://`, `ftp://`).

Both HTTPS and HTTP are supported as well as downloading directly from Amazon S3 compatible URLs with both pre-configured and automatic IAM credentials. (see s3.get state documentation) File retrieval from Openstack Swift object storage is supported via swift://container/object_path URLs, see swift.get documentation. For files hosted on the salt file server, if the file is located on the master in the directory named spam, and is called eggs, the source string is salt://spam/eggs. If source is left blank or None (use ~ in YAML), the file will be created as an empty file and the content will not be managed. This is also the case when a file already exists and the source is undefined; the contents of the file will not be changed or managed. If source is left blank or None, please also set replaced to False to make your intention explicit.

If the file is hosted on a HTTP or FTP server then the source_hash argument is also required.

A list of sources can also be passed in to provide a default source and a set of fallbacks. The first source in the list that is found to exist will be used and subsequent entries in the list will be ignored. Source list functionality only supports local files and remote files hosted on the salt master server or retrievable via HTTP, HTTPS, or FTP.

```
file_override_example:
  file.managed:
    - source:
```

(continues on next page)

(continued from previous page)

```
      - salt://file_that_does_not_exist
      - salt://file_that_exists
```

source_hash

This can be one of the following:

1. a source hash string
2. the URI of a file that contains source hash strings

The function accepts the first encountered long unbroken alphanumeric string of correct length as a valid hash, in order from most secure to least secure:

```
Type    Length
======  ======
sha512   128
sha384    96
sha256    64
sha224    56
sha1      40
md5       32
```

Using a Source Hash File The file can contain several checksums for several files. Each line must contain both the file name and the hash. If no file name is matched, the first hash encountered will be used, otherwise the most secure hash with the correct source file name will be used.

When using a source hash file the source_hash argument needs to be a url, the standard download urls are supported, ftp, http, salt etc:

Example:

```
tomdroid-src-0.7.3.tar.gz:
  file.managed:
    - name: /tmp/tomdroid-src-0.7.3.tar.gz
    - source: https://launchpad.net/tomdroid/beta/0.7.3/+download/
↪tomdroid-src-0.7.3.tar.gz
    - source_hash: https://launchpad.net/tomdroid/beta/0.7.3/+download/
↪tomdroid-src-0.7.3.hash
```

The following lines are all supported formats:

```
/etc/rc.conf ef6e82e4006dee563d98ada2a2a80a27
sha254c8525aee419eb649f0233be91c151178b30f0dff8ebbdcc8de71b1d5c8bcc06a  /
↪etc/resolv.conf
ead48423703509d37c4a90e6a0d53e143b6fc268
```

Debian file type `*.dsc` files are also supported.

Inserting the Source Hash in the SLS Data

The source_hash can be specified as a simple checksum, like so:

```
tomdroid-src-0.7.3.tar.gz:
  file.managed:
    - name: /tmp/tomdroid-src-0.7.3.tar.gz
    - source: https://launchpad.net/tomdroid/beta/0.7.3/+download/tomdroid-
↪src-0.7.3.tar.gz
    - source_hash: 79eef25f9b0b2c642c62b7f737d4f53f
```

Note: Releases prior to 2016.11.0 must also include the hash type, like in the below example:

```
tomdroid-src-0.7.3.tar.gz:
  file.managed:
    - name: /tmp/tomdroid-src-0.7.3.tar.gz
    - source: https://launchpad.net/tomdroid/beta/0.7.3/+download/tomdroid-
↪src-0.7.3.tar.gz
    - source_hash: md5=79eef25f9b0b2c642c62b7f737d4f53f
```

Known issues: If the remote server URL has the hash file as an apparent sub-directory of the source file, the module will discover that it has already cached a directory where a file should be cached. For example:

```
tomdroid-src-0.7.3.tar.gz:
  file.managed:
    - name: /tmp/tomdroid-src-0.7.3.tar.gz
    - source: https://launchpad.net/tomdroid/beta/0.7.3/+download/
↪tomdroid-src-0.7.3.tar.gz
    - source_hash: https://launchpad.net/tomdroid/beta/0.7.3/+download/
↪tomdroid-src-0.7.3.tar.gz/+md5
```

source_hash_name When `source_hash` refers to a hash file, Salt will try to find the correct hash by matching the filename/URI associated with that hash. By default, Salt will look for the filename being managed. When managing a file at path `/tmp/foo.txt`, then the following line in a hash file would match:

```
acbd18db4cc2f85cedef654fccc4a4d8    foo.txt
```

However, sometimes a hash file will include multiple similar paths:

```
37b51d194a7513e45b56f6524f2d51f2    ./dir1/foo.txt
acbd18db4cc2f85cedef654fccc4a4d8    ./dir2/foo.txt
73feffa4b7f6bb68e44cf984c85f6e88    ./dir3/foo.txt
```

In cases like this, Salt may match the incorrect hash. This argument can be used to tell Salt which filename to match, to ensure that the correct hash is identified. For example:

```
/tmp/foo.txt:
  file.managed:
    - source: https://mydomain.tld/dir2/foo.txt
    - source_hash: https://mydomain.tld/hashes
    - source_hash_name: ./dir2/foo.txt
```

Note: This argument must contain the full filename entry from the checksum file, as this argument is meant to disambiguate matches for multiple files that have the same basename. So, in the example above, simply using `foo.txt` would not match.

New in version 2016.3.5.

keep_source [True] Set to `False` to discard the cached copy of the source file once the state completes. This can be useful for larger files to keep them from taking up space in minion cache. However, keep in mind that discarding the source file will result in the state needing to re-download the source file if the state is run again.

New in version 2017.7.3.

user The user to own the file, this defaults to the user salt is running as on the minion

group The group ownership set for the file, this defaults to the group salt is running as on the minion. On Windows, this is ignored

mode The permissions to set on this file, e.g. `644`, `0775`, or `4664`.

The default mode for new files and directories corresponds to the umask of the salt process. The mode of existing files and directories will only be changed if `mode` is specified.

Note: This option is **not** supported on Windows.

Changed in version 2016.11.0: This option can be set to `keep`, and Salt will keep the mode from the Salt fileserver. This is only supported when the `source` URL begins with `salt://`, or for files local to the minion. Because the `source` option cannot be used with any of the `contents` options, setting the `mode` to `keep` is also incompatible with the `contents` options.

Note: keep does not work with salt-ssh.

As a consequence of how the files are transferred to the minion, and the inability to connect back to the master with salt-ssh, salt is unable to stat the file as it exists on the fileserver and thus cannot mirror the mode on the salt-ssh minion

---

attrs The attributes to have on this file, e.g. `a`, `i`. The attributes can be any or a combination of the following characters: `aAcCdDeijPsStTu`.

Note: This option is **not** supported on Windows.

New in version 2018.3.0.

template If this setting is applied, the named templating engine will be used to render the downloaded file. The following templates are supported:

- *`cheetah`*
- *`genshi`*
- *`jinja`*
- *`mako`*
- *`py`*
- *`wempy`*

makedirs [False] If set to `True`, then the parent directories will be created to facilitate the creation of the named file. If `False`, and the parent directory of the destination file doesn't exist, the state will fail.

dir_mode If directories are to be created, passing this option specifies the permissions for those directories. If this is not set, directories will be assigned permissions by adding the execute bit to the mode of the files.

The default mode for new files and directories corresponds umask of salt process. For existing files and directories it's not enforced.

replace [True] If set to `False` and the file already exists, the file will not be modified even if changes would otherwise be made. Permissions and ownership will still be enforced, however.

context Overrides default context variables passed to the template.

defaults Default context passed to the template.

backup Overrides the default backup mode for this specific file. See *backup_mode documentation* for more details.

show_changes Output a unified diff of the old file and the new file. If `False` return a boolean if any changes were made.

create [True] If set to `False`, then the file will only be managed if the file already exists on the system.

contents Specify the contents of the file. Cannot be used in combination with `source`. Ignores hashes and does not use a templating engine.

This value can be either a single string, a multiline YAML string or a list of strings. If a list of strings, then the strings will be joined together with newlines in the resulting file. For example, the below two example states would result in identical file contents:

```
/path/to/file1:
  file.managed:
    - contents:
      - This is line 1
      - This is line 2

/path/to/file2:
  file.managed:
    - contents: |
        This is line 1
        This is line 2
```

contents_pillar New in version 0.17.0.

Changed in version 2016.11.0: contents_pillar can also be a list, and the pillars will be concatenated together to form one file.

Operates like `contents`, but draws from a value stored in pillar, using the pillar path syntax used in *`pillar.get`*. This is useful when the pillar value contains newlines, as referencing a pillar variable using a jinja/mako template can result in YAML formatting issues due to the newlines causing indentation mismatches.

For example, the following could be used to deploy an SSH private key:

```
/home/deployer/.ssh/id_rsa:
  file.managed:
    - user: deployer
    - group: deployer
    - mode: 600
    - attrs: a
    - contents_pillar: userdata:deployer:id_rsa
```

This would populate `/home/deployer/.ssh/id_rsa` with the contents of `pillar['userdata']['deployer']['id_rsa']`. An example of this pillar setup would be like so:

```
userdata:
  deployer:
    id_rsa: |
        -----BEGIN RSA PRIVATE KEY-----
        MIIEowIBAAKCAQEAoQiwO3JhBquPAalQF9qP1lLZNXVjYMIswrMe2HcWUVBgh+vY
        U7sCwx/dH6+VvNwmCoqmNnP+8gTPKGl1vgAObJAnMT623dMXjVKwnEagZPRJIxDy
        B/HaAre9euNiY3LvIzBTWRSeMfT+rWvIKVBpvwlgGrfgz70m0pqxu+UyFbAGLin+
        GpxzZAMaFpZw4sSbIlRuissXZj/sHpQb8p9M5IeO4Z3rjkCP1cxI
        -----END RSA PRIVATE KEY-----
```

Note: The private key above is shortened to keep the example brief, but shows how to do multiline string in YAML. The key is followed by a pipe character, and the multiline string is indented two more spaces.

To avoid the hassle of creating an indented multiline YAML string, the *`file_tree external pillar`* can be used instead. However, this will not work for binary files in Salt releases before 2015.8.4.

---

contents_grains New in version 2014.7.0.

Operates like `contents`, but draws from a value stored in grains, using the grains path syntax used in *`grains.get`*. This functionality works similarly to `contents_pillar`, but with grains.

For example, the following could be used to deploy a "message of the day" file:

```
write_motd:
  file.managed:
    - name: /etc/motd
    - contents_grains: motd
```

This would populate `/etc/motd` file with the contents of the `motd` grain. The `motd` grain is not a default grain, and would need to be set prior to running the state:

```
salt '*' grains.set motd 'Welcome! This system is managed by Salt.'
```

contents_newline [True] New in version 2014.7.0.

Changed in version 2015.8.4: This option is now ignored if the contents being deployed contain binary data.

If `True`, files managed using `contents`, `contents_pillar`, or `contents_grains` will have a newline added to the end of the file if one is not present. Setting this option to `False` will ensure the final line, or entry, does not contain a new line. If the last line, or entry in the file does contain a new line already, this option will not remove it.

contents_delimiter New in version 2015.8.4.

Can be used to specify an alternate delimiter for `contents_pillar` or `contents_grains`. This delimiter will be passed through to *`pillar.get`* or *`grains.get`* when retrieving the contents.

encoding If specified, then the specified encoding will be used. Otherwise, the file will be encoded using the system locale (usually UTF-8). See https://docs.python.org/3/library/codecs.html#standard-encodings for the list of available encodings.

New in version 2017.7.0.

encoding_errors ['strict'] Error encoding scheme. Default is ```'strict'```. See https://docs.python.org/2/library/codecs.html#codec-base-classes for the list of available schemes.

New in version 2017.7.0.

allow_empty [True] New in version 2015.8.4.

If set to `False`, then the state will fail if the contents specified by `contents_pillar` or `contents_grains` are empty.

follow_symlinks [True] New in version 2014.7.0.

If the desired path is a symlink follow it and make changes to the file to which the symlink points.

check_cmd New in version 2014.7.0.

The specified command will be run with an appended argument of a *temporary* file containing the new managed contents. If the command exits with a zero status the new managed contents will be written to the managed destination. If the command exits with a nonzero exit code, the state will fail and no changes will be made to the file.

For example, the following could be used to verify sudoers before making changes:

```
/etc/sudoers:
  file.managed:
    - user: root
    - group: root
    - mode: 0440
    - attrs: i
    - source: salt://sudoers/files/sudoers.jinja
```

(continues on next page)

(continued from previous page)

```
    - template: jinja
    - check_cmd: /usr/sbin/visudo -c -f
```

NOTE: This `check_cmd` functions differently than the requisite `check_cmd`.

tmp_dir Directory for temp file created by `check_cmd`. Useful for checkers dependent on config file location (e.g. daemons restricted to their own config directories by an apparmor profile).

```
/etc/dhcp/dhcpd.conf:
  file.managed:
    - user: root
    - group: root
    - mode: 0755
    - tmp_dir: '/etc/dhcp'
    - contents: "# Managed by Salt"
    - check_cmd: dhcpd -t -cf
```

tmp_ext Suffix for temp file created by `check_cmd`. Useful for checkers dependent on config file extension (e.g. the init-checkconf upstart config checker).

```
/etc/init/test.conf:
  file.managed:
    - user: root
    - group: root
    - mode: 0440
    - tmp_ext: '.conf'
    - contents:
      - 'description "Salt Minion"'
      - 'start on started mountall'
      - 'stop on shutdown'
      - 'respawn'
      - 'exec salt-minion'
    - check_cmd: init-checkconf -f
```

skip_verify [False] If `True`, hash verification of remote file sources (`http://`, `https://`, `ftp://`) will be skipped, and the `source_hash` argument will be ignored.

New in version 2016.3.0.

win_owner [None] The owner of the directory. If this is not passed, user will be used. If user is not passed, the account under which Salt is running will be used.

New in version 2017.7.0.

win_perms [None] A dictionary containing permissions to grant and their propagation. For example: `{'Administrators': {'perms': 'full_control'}}` Can be a single basic perm or a list of advanced perms. `perms` must be specified. `applies_to` does not apply to file objects.

New in version 2017.7.0.

win_deny_perms [None] A dictionary containing permissions to deny and their propagation. For example: `{'Administrators': {'perms': 'full_control'}}` Can be a single basic perm or a list of advanced perms. `perms` must be specified. `applies_to` does not apply to file objects.

New in version 2017.7.0.

win_inheritance [True] True to inherit permissions from the parent directory, False not to inherit permission.

New in version 2017.7.0.

win_perms_reset [False] If `True` the existing DACL will be cleared and replaced with the settings defined in this function. If `False`, new entries will be appended to the existing DACL. Default is `False`.

New in version 2018.3.0.

Here's an example using the above `win_*` parameters:

```
create_config_file:
  file.managed:
    - name: C:\config\settings.cfg
    - source: salt://settings.cfg
    - win_owner: Administrators
    - win_perms:
        # Basic Permissions
        dev_ops:
          perms: full_control
        # List of advanced permissions
        appuser:
          perms:
            - read_attributes
            - read_ea
            - create_folders
            - read_permissions
        joe_snuffy:
          perms: read
    - win_deny_perms:
        fred_snuffy:
          perms: full_control
    - win_inheritance: False
```

`salt.states.file.`**`missing`**(*name*, ***kwargs*)

Verify that the named file or directory is missing, this returns True only if the named file is missing but does not remove the file if it is present.

name Absolute path which must NOT exist

`salt.states.file.`**`mknod`**(*name*, *ntype*, *major=0*, *minor=0*, *user=None*, *group=None*, *mode='0600'*)

Create a special file similar to the 'nix mknod command. The supported device types are `p` (fifo pipe), `c` (character device), and `b` (block device). Provide the major and minor numbers when specifying a character device or block device. A fifo pipe does not require this information. The command will create the necessary dirs if needed. If a file of the same name not of the same type/major/minor exists, it will not be overwritten or unlinked (deleted). This is logically in place as a safety measure because you can really shoot yourself in the foot here and it is the behavior of 'nix `mknod`. It is also important to note that not just anyone can create special devices. Usually this is only done as root. If the state is executed as none other than root on a minion, you may receive a permission error.

name name of the file

ntype node type 'p' (fifo pipe), 'c' (character device), or 'b' (block device)

major major number of the device does not apply to a fifo pipe

minor minor number of the device does not apply to a fifo pipe

user owning user of the device/pipe

group owning group of the device/pipe

mode permissions on the device/pipe

Usage:

```
/dev/chr:
  file.mknod:
    - ntype: c
    - major: 180
    - minor: 31
    - user: root
    - group: root
    - mode: 660

/dev/blk:
```

(continues on next page)

(continued from previous page)

```
  file.mknod:
    - ntype: b
    - major: 8
    - minor: 999
    - user: root
    - group: root
    - mode: 660

/dev/fifo:
  file.mknod:
    - ntype: p
    - user: root
    - group: root
    - mode: 660
```

New in version 0.17.0.

`salt.states.file.`**`mod_run_check_cmd`**(*cmd*, *filename*, ***check_cmd_opts*)

Execute the check_cmd logic.

Return a result dict if `check_cmd` succeeds (check_cmd == 0) otherwise return True

`salt.states.file.`**`not_cached`**(*name*, *saltenv='base'*)

New in version 2017.7.3.

Ensures that a file is saved to the minion's cache. This state is primarily invoked by other states to ensure that we do not re-download a source file if we do not need to.

name The URL of the file to be cached. To cache a file from an environment other than `base`, either use the `saltenv` argument or include the saltenv in the URL (e.g. `salt://path/to/file.conf?saltenv=dev`).

Note: A list of URLs is not supported, this must be a single URL. If a local file is passed here, the state will take no action.

saltenv Used to specify the environment from which to download a file from the Salt fileserver (i.e. those with `salt://` URL).

`salt.states.file.`**`patch`**(*name*, *source=None*, *source_hash=None*, *source_hash_name=None*, *skip_verify=False*, *template=None*, *context=None*, *defaults=None*, *options=''*, *reject_file=None*, *strip=None*, *saltenv=None*, ***kwargs*)

Ensure that a patch has been applied to the specified file or directory

Changed in version 2019.2.0: The `hash` and `dry_run_first` options are now ignored, as the logic which determines whether or not the patch has already been applied no longer requires them. Additionally, this state now supports patch files that modify more than one file. To use these sort of patches, specify a directory (and, if necessary, the `strip` option) instead of a file.

Note: A suitable `patch` executable must be available on the minion. Also, keep in mind that the pre-check this state does to determine whether or not changes need to be made will create a temp file and send all patch output to that file. This means that, in the event that the patch would not have applied cleanly, the comment included in the state results will reference a temp file that will no longer exist once the state finishes running.

name The file or directory to which the patch should be applied

source The patch file to apply

Changed in version 2019.2.0: The source can now be from any file source supported by Salt (`salt://`, `http://`, `https://`, `ftp://`, etc.). Templating is also now supported.

source_hash Works the same way as in *`file.managed`*.

New in version 2019.2.0.

source_hash_name Works the same way as in *`file.managed`*

New in version 2019.2.0.

skip_verify Works the same way as in *`file.managed`*

New in version 2019.2.0.

template Works the same way as in *`file.managed`*

New in version 2019.2.0.

context Works the same way as in *`file.managed`*

New in version 2019.2.0.

defaults Works the same way as in *`file.managed`*

New in version 2019.2.0.

options Extra options to pass to patch. This should not be necessary in most cases.

Note: For best results, short opts should be separate from one another. The `-N` and `-r`, and `-o` options are used internally by this state and cannot be used here. Additionally, instead of using `-pN` or `--strip=N`, use the `strip` option documented below.

reject_file If specified, any rejected hunks will be written to this file. If not specified, then they will be written to a temp file which will be deleted when the state finishes running.

Important: The parent directory must exist. Also, this will overwrite the file if it is already present.

New in version 2019.2.0.

strip Number of directories to strip from paths in the patch file. For example, using the below SLS would instruct Salt to use `-p1` when applying the patch:

```
/etc/myfile.conf:
  file.patch:
    - source: salt://myfile.patch
    - strip: 1
```

New in version 2019.2.0: In previous versions, `-p1` would need to be passed as part of the `options` value.

saltenv Specify the environment from which to retrieve the patch file indicated by the `source` parameter. If not provided, this defaults to the environment from which the state is being executed.

Note: Ignored when the patch file is from a non-`salt://` source.

Usage:

```
# Equivalent to ``patch --forward /opt/myfile.txt myfile.patch``
/opt/myfile.txt:
  file.patch:
    - source: salt://myfile.patch
```

`salt.states.file.`**`prepend`**`(`*name, text=None, makedirs=False, source=None, source_hash=None, template='jinja', sources=None, source_hashes=None, defaults=None, context=None, header=None*`)`

Ensure that some text appears at the beginning of a file

The text will not be prepended again if it already exists in the file. You may specify a single line of text or a list of lines to append.

name The location of the file to append to.

text The text to be appended, which can be a single string or a list of strings.

makedirs If the file is located in a path without a parent directory, then the state will fail. If makedirs is set to True, then the parent directories will be created to facilitate the creation of the named file. Defaults to False.

source A single source file to append. This source file can be hosted on either the salt master server, or on an HTTP or FTP server. Both HTTPS and HTTP are supported as well as downloading directly from Amazon S3 compatible URLs with both pre-configured and automatic IAM credentials (see s3.get state documentation). File retrieval from Openstack Swift object storage is supported via swift://container/object_path URLs (see swift.get documentation).

For files hosted on the salt file server, if the file is located on the master in the directory named spam, and is called eggs, the source string is salt://spam/eggs.

If the file is hosted on an HTTP or FTP server, the source_hash argument is also required.

source_hash

This can be one of the following:

1. a source hash string
2. the URI of a file that contains source hash strings

The function accepts the first encountered long unbroken alphanumeric string of correct length as a valid hash, in order from most secure to least secure:

```
Type    Length
======  ======
sha512     128
sha384      96
sha256      64
sha224      56
sha1        40
md5         32
```

See the `source_hash` parameter description for *`file.managed`* function for more details and examples.

template The named templating engine will be used to render the appended-to file. Defaults to `jinja`. The following templates are supported:

- *`cheetah`*
- *`genshi`*
- *`jinja`*
- *`mako`*
- *`py`*
- *`wempy`*

sources A list of source files to append. If the files are hosted on an HTTP or FTP server, the source_hashes argument is also required.

source_hashes A list of source_hashes corresponding to the sources list specified in the sources argument.

defaults Default context passed to the template.

context Overrides default context variables passed to the template.

ignore_whitespace New in version 2015.8.4.

Spaces and Tabs in text are ignored by default, when searching for the appending content, one space or multiple tabs are the same for salt. Set this option to `False` if you want to change this behavior.

Multi-line example:

```
/etc/motd:
  file.prepend:
    - text: |
        Thou hadst better eat salt with the Philosophers of Greece,
        than sugar with the Courtiers of Italy.
        - Benjamin Franklin
```

Multiple lines of text:

```
/etc/motd:
  file.prepend:
    - text:
      - Trust no one unless you have eaten much salt with him.
      - "Salt is born of the purest of parents: the sun and the sea."
```

Optionally, require the text to appear exactly as specified (order and position). Combine with multi-line or multiple lines of input.

```
/etc/motd:
  file.prepend:
    - header: True
    - text:
      - This will be the very first line in the file.
      - The 2nd line, regardless of duplicates elsewhere in the file.
      - These will be written anew if they do not appear verbatim.
```

Gather text from multiple template files:

```
/etc/motd:
  file:
    - prepend
    - template: jinja
    - sources:
      - salt://motd/devops-messages.tmpl
      - salt://motd/hr-messages.tmpl
      - salt://motd/general-messages.tmpl
```

New in version 2014.7.0.

`salt.states.file.`**`recurse`**(*name, source, keep_source=True, clean=False, require=None, user=None, group=None, dir_mode=None, file_mode=None, sym_mode=None, template=None, context=None, replace=True, defaults=None, include_empty=False, backup=", include_pat=None, exclude_pat=None, maxdepth=None, keep_symlinks=False, force_symlinks=False, win_owner=None, win_perms=None, win_deny_perms=None, win_inheritance=True, **kwargs*)

Recurse through a subdirectory on the master and copy said subdirectory over to the specified path.

name The directory to set the recursion in

source The source directory, this directory is located on the salt master file server and is specified with the salt:// protocol. If the directory is located on the master in the directory named spam, and is called eggs, the source string is salt://spam/eggs

keep_source [True] Set to `False` to discard the cached copy of the source file once the state completes. This can be useful for larger files to keep them from taking up space in minion cache. However, keep in mind that discarding the source file will result in the state needing to re-download the source file if the state is run again.

New in version 2017.7.3.

clean Make sure that only files that are set up by salt and required by this function are kept. If this option is set then everything in this directory will be deleted unless it is required.

require Require other resources such as packages or files

user The user to own the directory. This defaults to the user salt is running as on the minion

group The group ownership set for the directory. This defaults to the group salt is running as on the minion. On Windows, this is ignored

dir_mode The permissions mode to set on any directories created.

The default mode for new files and directories corresponds umask of salt process. For existing files and directories it's not enforced.

Note: This option is **not** supported on Windows.

file_mode The permissions mode to set on any files created.

The default mode for new files and directories corresponds umask of salt process. For existing files and directories it's not enforced.

Note: This option is **not** supported on Windows.

Changed in version 2016.11.0: This option can be set to `keep`, and Salt will keep the mode from the Salt fileserver. This is only supported when the `source` URL begins with `salt://`, or for files local to the minion. Because the `source` option cannot be used with any of the `contents` options, setting the `mode` to `keep` is also incompatible with the `contents` options.

sym_mode The permissions mode to set on any symlink created.

The default mode for new files and directories corresponds umask of salt process. For existing files and directories it's not enforced.

Note: This option is **not** supported on Windows.

template If this setting is applied, the named templating engine will be used to render the downloaded file. The following templates are supported:

- *`cheetah`*
- *`genshi`*
- *`jinja`*
- *`mako`*
- *`py`*
- *`wempy`*

Note: The template option is required when recursively applying templates.

replace [True] If set to `False` and the file already exists, the file will not be modified even if changes would otherwise be made. Permissions and ownership will still be enforced, however.

context Overrides default context variables passed to the template.

defaults Default context passed to the template.

include_empty Set this to True if empty directories should also be created (default is False)

backup Overrides the default backup mode for all replaced files. See *backup_mode documentation* for more details.

include_pat When copying, include only this pattern from the source. Default is glob match; if prefixed with 'E@', then regexp match. Example:

```
- include_pat: hello*       :: glob matches 'hello01', 'hello02'
                               ... but not 'otherhello'
- include_pat: E@hello      :: regexp matches 'otherhello',
                               'hello01' ...
```

exclude_pat Exclude this pattern from the source when copying. If both *include_pat* and *exclude_pat* are supplied, then it will apply conditions cumulatively. i.e. first select based on include_pat, and then within that result apply exclude_pat.

Also, when 'clean=True', exclude this pattern from the removal list and preserve in the destination. Example:

```
- exclude_pat: APPDATA*               :: glob matches APPDATA.01,
                                         APPDATA.02,.. for exclusion
- exclude_pat: E@(APPDATA)|(TEMPDATA) :: regexp matches APPDATA
                                         or TEMPDATA for exclusion
```

maxdepth When copying, only copy paths which are of depth *maxdepth* from the source path. Example:

```
- maxdepth: 0      :: Only include files located in the source
                      directory
- maxdepth: 1      :: Only include files located in the source
                      or immediate subdirectories
```

keep_symlinks Keep symlinks when copying from the source. This option will cause the copy operation to terminate at the symlink. If desire behavior similar to rsync, then set this to True.

force_symlinks Force symlink creation. This option will force the symlink creation. If a file or directory is obstructing symlink creation it will be recursively removed so that symlink creation can proceed. This option is usually not needed except in special circumstances.

win_owner [None] The owner of the symlink and directories if `makedirs` is True. If this is not passed, `user` will be used. If `user` is not passed, the account under which Salt is running will be used.

New in version 2017.7.7.

win_perms [None] A dictionary containing permissions to grant

New in version 2017.7.7.

win_deny_perms [None] A dictionary containing permissions to deny

New in version 2017.7.7.

win_inheritance [None] True to inherit permissions from parent, otherwise False

New in version 2017.7.7.

`salt.states.file.`**`rename`**(*name*, *source*, *force=False*, *makedirs=False*)

If the source file exists on the system, rename it to the named file. The named file will not be overwritten if it already exists unless the force option is set to True.

name The location of the file to rename to

source The location of the file to move to the location specified with name

force If the target location is present then the file will not be moved, specify "force: True" to overwrite the target file

makedirs If the target subdirectories don't exist create them

`salt.states.file.`**`replace`**(*name*, *pattern*, *repl*, *count=0*, *flags=8*, *bufsize=1*, *append_if_not_found=False*, *prepend_if_not_found=False*, *not_found_content=None*, *backup='.bak'*, *show_changes=True*, *ignore_if_missing=False*, *backslash_literal=False*)

Maintain an edit in a file.

New in version 0.17.0.

name Filesystem path to the file to be edited. If a symlink is specified, it will be resolved to its target.

pattern A regular expression, to be matched using Python's `re.search()`.

Note: If you need to match a literal string that contains regex special characters, you may want to use salt's custom Jinja filter, `regex_escape`.

```
{{ 'http://example.com?foo=bar%20baz' | regex_escape }}
```

---

repl The replacement text

count Maximum number of pattern occurrences to be replaced. Defaults to 0. If count is a positive integer n, no more than n occurrences will be replaced, otherwise all occurrences will be replaced.

flags A list of flags defined in the `re` module documentation from the Python standard library. Each list item should be a string that will correlate to the human-friendly flag name. E.g., `['IGNORECASE', 'MULTILINE']`. Optionally, `flags` may be an int, with a value corresponding to the XOR (`|`) of all the desired flags. Defaults to `8` (which equates to `['MULTILINE']`).

Note: `file.replace` reads the entire file as a string to support multiline regex patterns. Therefore, when using anchors such as `^` or `$` in the pattern, those anchors may be relative to the line OR relative to the file. The default for `file.replace` is to treat anchors as relative to the line, which is implemented by setting the default value of `flags` to `['MULTILINE']`. When overriding the default value for `flags`, if `'MULTILINE'` is not present then anchors will be relative to the file. If the desired behavior is for anchors to be relative to the line, then simply add `'MULTILINE'` to the list of flags.

---

bufsize How much of the file to buffer into memory at once. The default value `1` processes one line at a time. The special value `file` may be specified which will read the entire file into memory before processing.

append_if_not_found [False] If set to `True`, and pattern is not found, then the content will be appended to the file.

New in version 2014.7.0.

prepend_if_not_found [False] If set to `True` and pattern is not found, then the content will be prepended to the file.

New in version 2014.7.0.

not_found_content Content to use for append/prepend if not found. If `None` (default), uses `repl`. Useful when `repl` uses references to group in pattern.

New in version 2014.7.0.

backup The file extension to use for a backup of the file before editing. Set to `False` to skip making a backup.

show_changes [True] Output a unified diff of the old file and the new file. If `False` return a boolean if any changes were made. Returns a boolean or a string.

ignore_if_missing [False] New in version 2016.3.4.

Controls what to do if the file is missing. If set to `False`, the state will display an error raised by the execution module. If set to `True`, the state will simply report no changes.

backslash_literal [False] New in version 2016.11.7.

Interpret backslashes as literal backslashes for the repl and not escape characters. This will help when using append/prepend so that the backslashes are not interpreted for the repl on the second run of the state.

For complex regex patterns, it can be useful to avoid the need for complex quoting and escape sequences by making use of YAML's multiline string syntax.

```
complex_search_and_replace:
  file.replace:
    # <...snip...>
```

(continues on next page)

(continued from previous page)

```
    - pattern: |
        CentOS \(2.6.32[^\\n]+\\n\s+root[^\\n]+\\n\)+
```

Note: When using YAML multiline string syntax in `pattern:`, make sure to also use that syntax in the `repl:` part, or you might loose line feeds.

When regex capture groups are used in `pattern:`, their captured value is available for reuse in the `repl:` part as a backreference (ex. `\1`).

```
add_login_group_to_winbind_ssh_access_list:
  file.replace:
    - name: '/etc/security/pam_winbind.conf'
    - pattern: '^(require_membership_of = )(.*)$'
    - repl: '\1\2,append-new-group-to-line'
```

Note: The `file.replace` state uses Python's `re` module. For more advanced options, see https://docs.python.org/2/library/re.html

`salt.states.file.`**`retention_schedule`**`(`*name, retain, strptime_format=None, timezone=None*`)`

Apply retention scheduling to backup storage directory.

New in version 2016.11.0.

Parameters

- **`name`** – The filesystem path to the directory containing backups to be managed.
- **`retain`** – Delete the backups, except for the ones we want to keep. The N below should be an integer but may also be the special value of `all`, which keeps all files matching the criteria. All of the retain options default to None, which means to not keep files based on this criteria.

  most_recent N Keep the most recent N files.

  first_of_hour N For the last N hours from now, keep the first file after the hour.

  first_of_day N For the last N days from now, keep the first file after midnight. See also `timezone`.

  first_of_week N For the last N weeks from now, keep the first file after Sunday midnight.

  first_of_month N For the last N months from now, keep the first file after the start of the month.

  first_of_year N For the last N years from now, keep the first file after the start of the year.
- **`strptime_format`** – A python strptime format string used to first match the filenames of backups and then parse the filename to determine the datetime of the file. https://docs.python.org/2/library/datetime.html#datetime.datetime.strptime Defaults to None, which considers all files in the directory to be backups eligible for deletion and uses `os.path.getmtime()` to determine the datetime.
- **`timezone`** – The timezone to use when determining midnight. This is only used when datetime is pulled from `os.path.getmtime()`. Defaults to `None` which uses the timezone from the locale.

Usage example:

```
/var/backups/example_directory:
```

(continues on next page)

(continued from previous page)

```
file.retention_schedule:
  - retain:
      most_recent: 5
      first_of_hour: 4
      first_of_day: 7
      first_of_week: 6    # NotImplemented yet.
      first_of_month: 6
      first_of_year: all
  - strptime_format: example_name_%Y%m%dT%H%M%S.tar.bz2
  - timezone: None
```

`salt.states.file.`**`serialize`**(*name, dataset=None, dataset_pillar=None, user=None, group=None, mode=None, backup='', makedirs=False, show_changes=True, create=True, merge_if_exists=False, encoding=None, encoding_errors='strict', serializer_opts=None, deserializer_opts=None, **kwargs*)

Serializes dataset and store it into managed file. Useful for sharing simple configuration files.

name The location of the file to create

dataset The dataset that will be serialized

dataset_pillar Operates like `dataset`, but draws from a value stored in pillar, using the pillar path syntax used in *`pillar.get`*. This is useful when the pillar value contains newlines, as referencing a pillar variable using a jinja/mako template can result in YAML formatting issues due to the newlines causing indentation mismatches.

New in version 2015.8.0.

formatter Write the data as this format. See the list of `serializer modules` for supported output formats.

encoding If specified, then the specified encoding will be used. Otherwise, the file will be encoded using the system locale (usually UTF-8). See https://docs.python.org/3/library/codecs.html#standard-encodings for the list of available encodings.

New in version 2017.7.0.

encoding_errors ['strict'] Error encoding scheme. Default is ``` `'strict'` ```. See https://docs.python.org/2/library/codecs.html#codec-base-classes for the list of available schemes.

New in version 2017.7.0.

user The user to own the directory, this defaults to the user salt is running as on the minion

group The group ownership set for the directory, this defaults to the group salt is running as on the minion

mode The permissions to set on this file, e.g. `644`, `0775`, or `4664`.

The default mode for new files and directories corresponds umask of salt process. For existing files and directories it's not enforced.

Note: This option is **not** supported on Windows.

backup Overrides the default backup mode for this specific file.

makedirs Create parent directories for destination file.

New in version 2014.1.3.

show_changes Output a unified diff of the old file and the new file. If `False` return a boolean if any changes were made.

create Default is True, if create is set to False then the file will only be managed if the file already exists on the system.

merge_if_exists Default is False, if merge_if_exists is True then the existing file will be parsed and the dataset passed in will be merged with the existing content

New in version 2014.7.0.

serializer_opts Pass through options to serializer. For example:

```
/etc/dummy/package.yaml
  file.serialize:
    - formatter: yaml
    - serializer_opts:
      - explicit_start: True
      - default_flow_style: True
      - indent: 4
```

The valid opts are the additional opts (i.e. not the data being serialized) for the function used to serialize the data. Documentation for the these functions can be found in the list below:

- For **yaml**: yaml.dump()
- For **json**: json.dumps()
- For **python**: pprint.pformat()

deserializer_opts Like `serializer_opts` above, but only used when merging with an existing file (i.e. when `merge_if_exists` is set to `True`).

The options specified here will be passed to the deserializer to load the existing data, before merging with the specified data and re-serializing.

```
/etc/dummy/package.yaml
  file.serialize:
    - formatter: yaml
    - serializer_opts:
      - explicit_start: True
      - default_flow_style: True
      - indent: 4
    - deserializer_opts:
      - encoding: latin-1
    - merge_if_exists: True
```

The valid opts are the additional opts (i.e. not the data being deserialized) for the function used to deserialize the data. Documentation for the these functions can be found in the list below:

- For **yaml**: yaml.load()
- For **json**: json.loads()

However, note that not all arguments are supported. For example, when deserializing JSON, arguments like `parse_float` and `parse_int` which accept a callable object cannot be handled in an SLS file.

New in version 2019.2.0.

For example, this state:

```
/etc/dummy/package.json:
  file.serialize:
    - dataset:
        name: naive
        description: A package using naive versioning
        author: A confused individual <iam@confused.com>
        dependencies:
          express: '>= 1.2.0'
          optimist: '>= 0.1.0'
        engine: node 0.4.1
    - formatter: json
```

will manage the file `/etc/dummy/package.json`:

```
{
  "author": "A confused individual <iam@confused.com>",
  "dependencies": {
    "express": ">= 1.2.0",
    "optimist": ">= 0.1.0"
  },
  "description": "A package using naive versioning",
  "engine": "node 0.4.1",
  "name": "naive"
}
```

`salt.states.file.`**`shortcut`**(*name*, *target*, *arguments=None*, *working_dir=None*, *description=None*, *icon_location=None*, *force=False*, *backupname=None*, *makedirs=False*, *user=None*, ***kwargs*)

Create a Windows shortcut

If the file already exists and is a shortcut pointing to any location other than the specified target, the shortcut will be replaced. If it is a regular file or directory then the state will return False. If the regular file or directory is desired to be replaced with a shortcut pass force: True, if it is to be renamed, pass a backupname.

name The location of the shortcut to create. Must end with either ".lnk" or ".url"

target The location that the shortcut points to

arguments Any arguments to pass in the shortcut

working_dir Working directory in which to execute target

description Description to set on shortcut

icon_location Location of shortcut's icon

force If the name of the shortcut exists and is not a file and force is set to False, the state will fail. If force is set to True, the link or directory in the way of the shortcut file will be deleted to make room for the shortcut, unless backupname is set, when it will be renamed

backupname If the name of the shortcut exists and is not a file, it will be renamed to the backupname. If the backupname already exists and force is False, the state will fail. Otherwise, the backupname will be removed first.

makedirs If the location of the shortcut does not already have a parent directory then the state will fail, setting makedirs to True will allow Salt to create the parent directory. Setting this to True will also create the parent for backupname if necessary.

user The user to own the file, this defaults to the user salt is running as on the minion

The default mode for new files and directories corresponds umask of salt process. For existing files and directories it's not enforced.

`salt.states.file.`**`symlink`**(*name*, *target*, *force=False*, *backupname=None*, *makedirs=False*, *user=None*, *group=None*, *mode=None*, *win_owner=None*, *win_perms=None*, *win_deny_perms=None*, *win_inheritance=None*, ***kwargs*)

Create a symbolic link (symlink, soft link)

If the file already exists and is a symlink pointing to any location other than the specified target, the symlink will be replaced. If an entry with the same name exists then the state will return False. If the existing entry is desired to be replaced with a symlink pass force: True, if it is to be renamed, pass a backupname.

name The location of the symlink to create

target The location that the symlink points to

force If the name of the symlink exists and is not a symlink and force is set to False, the state will fail. If force is set to True, the existing entry in the way of the symlink file will be deleted to make room for the symlink, unless backupname is set, when it will be renamed

Changed in version 3000: Force will now remove all types of existing file system entries, not just files, directories and symlinks.

backupname If the name of the symlink exists and is not a symlink, it will be renamed to the backupname. If the backupname already exists and force is False, the state will fail. Otherwise, the backupname will be removed first. An absolute path OR a basename file/directory name must be provided. The latter will be placed relative to the symlink destination's parent directory.

makedirs If the location of the symlink does not already have a parent directory then the state will fail, setting makedirs to True will allow Salt to create the parent directory

user The user to own the file, this defaults to the user salt is running as on the minion

group The group ownership set for the file, this defaults to the group salt is running as on the minion. On Windows, this is ignored

mode The permissions to set on this file, aka 644, 0775, 4664. Not supported on Windows.

The default mode for new files and directories corresponds umask of salt process. For existing files and directories it's not enforced.

win_owner [None] The owner of the symlink and directories if `makedirs` is True. If this is not passed, `user` will be used. If `user` is not passed, the account under which Salt is running will be used.

New in version 2017.7.7.

win_perms [None] A dictionary containing permissions to grant

New in version 2017.7.7.

win_deny_perms [None] A dictionary containing permissions to deny

New in version 2017.7.7.

win_inheritance [None] True to inherit permissions from parent, otherwise False

New in version 2017.7.7.

`salt.states.file.`**`tidied`**(*name*, *age=0*, *matches=None*, *rmdirs=False*, *size=0*, ***kwargs*)

Remove unwanted files based on specific criteria. Multiple criteria are OR'd together, so a file that is too large but is not old enough will still get tidied.

If neither age nor size is given all files which match a pattern in matches will be removed.

name The directory tree that should be tidied

age Maximum age in days after which files are considered for removal

matches List of regular expressions to restrict what gets removed. Default: ['.*']

rmdirs Whether or not it's allowed to remove directories

size Maximum allowed file size. Files greater or equal to this size are removed. Doesn't apply to directories or symbolic links

```
cleanup:
  file.tidied:
    - name: /tmp/salt_test
    - rmdirs: True
    - matches:
      - foo
      - b.*r
```

`salt.states.file.`**`touch`**(*name*, *atime=None*, *mtime=None*, *makedirs=False*)

Replicate the 'nix "touch" command to create a new empty file or update the atime and mtime of an existing file.

Note that if you just want to create a file and don't care about atime or mtime, you should use `file.managed` instead, as it is more feature-complete. (Just leave out the `source`/`template`/`contents` arguments, and it will just create the file and/or check its permissions, without messing with contents)

name name of the file

atime atime of the file

mtime mtime of the file

makedirs whether we should create the parent directory/directories in order to touch the file

Usage:

```
/var/log/httpd/logrotate.empty:
  file.touch
```

New in version 0.9.5.

`salt.states.file.`**`uncomment`**`(`*name*, *regex*, *char='#'*, *backup='.bak'*`)`

Uncomment specified commented lines in a file

name The full path to the file to be edited

regex A regular expression used to find the lines that are to be uncommented. This regex should not include the comment character. A leading ^ character will be stripped for convenience (for easily switching between comment() and uncomment()). The regex will be searched for from the beginning of the line, ignoring leading spaces (we prepend '^[t]*')

char [#] The character to remove in order to uncomment a line

backup [`.bak`] The file will be backed up before edit with this file extension;

> **Warning:** This backup will be overwritten each time `sed` / `comment` / `uncomment` is called. Meaning the backup will only be useful after the first invocation.

Set to False/None to not keep a backup.

Usage:

```
/etc/adduser.conf:
  file.uncomment:
    - regex: EXTRA_GROUPS
```

New in version 0.9.5.

21.19.86 salt.states.firewall module

State to check firewall configurations

New in version 2016.3.0.

`salt.states.firewall.`**`check`**`(`*name*, *port=None*, ***kwargs*`)`

Checks if there is an open connection from the minion to the defined host on a specific port.

name host name or ip address to test connection to

port The port to test the connection on

kwargs

Additional parameters, parameters allowed are: proto (tcp or udp) family (ipv4 or ipv6) timeout

```
testgoogle:
  firewall.check:
    - name: 'google.com'
    - port: 80
    - proto: 'tcp'
```

21.19.87 salt.states.firewalld

Management of firewalld

New in version 2015.8.0.

The following example applies changes to the public zone, blocks echo-reply and echo-request packets, does not set the zone to be the default, enables masquerading, and allows ports 22/tcp and 25/tcp. It will be applied permanently and directly before restart/reload.

```
public:
  firewalld.present:
    - name: public
    - block_icmp:
      - echo-reply
      - echo-request
    - default: False
    - masquerade: True
    - ports:
      - 22/tcp
      - 25/tcp
```

The following example applies changes to the public zone, enables masquerading and configures port forwarding TCP traffic from port 22 to 2222, and forwards TCP traffic from port 80 to 443 at 192.168.0.1.

```
my_zone:
  firewalld.present:
    - name: public
    - masquerade: True
    - port_fwd:
      - 22:2222:tcp
      - 80:443:tcp:192.168.0.1
```

The following example binds the public zone to interface eth0 and to all packets coming from the 192.168.1.0/24 subnet. It also removes the zone from all other interfaces or sources.

```
public:
  firewalld.present:
    - name: public
    - interfaces:
      - eth0
    - sources:
      - 192.168.1.0/24
```

Here, we define a new service that encompasses TCP ports 4505 4506:

```
saltmaster:
  firewalld.service:
    - name: saltmaster
    - ports:
      - 4505/tcp
      - 4506/tcp
```

To make this new service available in a zone, the following can be used, which would allow access to the salt master from the 10.0.0.0/8 subnet:

```
saltzone:
  firewalld.present:
    - name: saltzone
    - services:
      - saltmaster
    - sources:
      - 10.0.0.0/8
```

class `salt.states.firewalld.`**`ForwardingMapping`**(*srcport, destport, protocol, destaddr*)

Represents a port forwarding statement mapping a local port to a remote port for a specific protocol (TCP or UDP)

`todict`()

Returns a pretty dictionary meant for command line output.

`salt.states.firewalld.`**`present`**(*name, block_icmp=None, prune_block_icmp=False, default=None, masquerade=False, ports=None, prune_ports=False, port_fwd=None, prune_port_fwd=False, services=None, prune_services=False, interfaces=None, prune_interfaces=False, sources=None, prune_sources=False, rich_rules=None, prune_rich_rules=False*)

Ensure a zone has specific attributes.

name The zone to modify.

default [None] Set this zone as the default zone if `True`.

masquerade [False] Enable or disable masquerade for a zone.

block_icmp [None] List of ICMP types to block in the zone.

prune_block_icmp [False] If `True`, remove all but the specified block_icmp from the zone.

ports [None] List of ports to add to the zone.

prune_ports [False] If `True`, remove all but the specified ports from the zone.

port_fwd [None] List of port forwards to add to the zone.

prune_port_fwd [False] If `True`, remove all but the specified port_fwd from the zone.

services [None] List of services to add to the zone.

prune_services [False] If `True`, remove all but the specified services from the zone. .. note:: Currently defaults to True for compatibility, but will be changed to False in a future release.

interfaces [None] List of interfaces to add to the zone.

prune_interfaces [False] If `True`, remove all but the specified interfaces from the zone.

sources [None] List of sources to add to the zone.

prune_sources [False] If `True`, remove all but the specified sources from the zone.

rich_rules [None] List of rich rules to add to the zone.

prune_rich_rules [False] If `True`, remove all but the specified rich rules from the zone.

`salt.states.firewalld.`**`service`**(*name, ports=None, protocols=None*)

Ensure the service exists and encompasses the specified ports and protocols.

New in version 2016.11.0.

21.19.88 salt.states.gem

Installation of Ruby modules packaged as gems

A state module to manage rubygems. Gems can be set up to be installed or removed. This module will use RVM or rbenv if they are installed. In that case, you can specify what ruby version and gemset to target.

```
addressable:
  gem.installed:
    - user: rvm
    - ruby: jruby@jgemset
```

`salt.states.gem.`**`installed`**(*name, ruby=None, gem_bin=None, user=None, version=None, rdoc=False, ri=False, pre_releases=False, proxy=None, source=None*)

Make sure that a gem is installed.

name The name of the gem to install

ruby: **None** Only for RVM or rbenv installations: the ruby version and gemset to target.

gem_bin: None Custom `gem` command to run instead of the default. Use this to install gems to a non-default ruby install. If you are using rvm or rbenv use the ruby argument instead.

user: None The user under which to run the `gem` command

New in version 0.17.0.

version [None] Specify the version to install for the gem. Doesn't play nice with multiple gems at once

rdoc [False] Generate RDoc documentation for the gem(s).

ri [False] Generate RI documentation for the gem(s).

pre_releases [False] Install pre-release version of gem(s) if available.

proxy [None] Use the specified HTTP proxy server for all outgoing traffic. Format: http://hostname{[}:port]

source [None] Use the specified HTTP gem source server to download gem. Format: http://hostname{[}:port]

`salt.states.gem.`**`removed`**(*name*, *ruby=None*, *user=None*, *gem_bin=None*)

Make sure that a gem is not installed.

name The name of the gem to uninstall

gem_bin [None] Full path to `gem` binary to use.

ruby [None] If RVM or rbenv are installed, the ruby version and gemset to use. Ignored if `gem_bin` is specified.

user: None The user under which to run the `gem` command

New in version 0.17.0.

`salt.states.gem.`**`sources_add`**(*name*, *ruby=None*, *user=None*)

Make sure that a gem source is added.

name The URL of the gem source to be added

ruby: None For RVM or rbenv installations: the ruby version and gemset to target.

user: None The user under which to run the `gem` command

New in version 0.17.0.

`salt.states.gem.`**`sources_remove`**(*name*, *ruby=None*, *user=None*)

Make sure that a gem source is removed.

name The URL of the gem source to be removed

ruby: None For RVM or rbenv installations: the ruby version and gemset to target.

user: None The user under which to run the `gem` command

New in version 0.17.0.

21.19.89 salt.states.git

States to manage git repositories and git configuration

Important: Before using git over ssh, make sure your remote host fingerprint exists in your `~/.ssh/known_hosts` file.

Changed in version 2015.8.8: This state module now requires git 1.6.5 (released 10 October 2009) or newer.

`salt.states.git.`**`cloned`**(*name*, *target=None*, *branch=None*, *user=None*, *password=None*, *identity=None*, *https_user=None*, *https_pass=None*, *output_encoding=None*)

New in version 2018.3.3,2019.2.0.

Ensure that a repository has been cloned to the specified target directory. If not, clone that repository. No fetches will be performed once cloned.

name Address of the remote repository

target Name of the target directory where repository should be cloned

branch Remote branch to check out. If unspecified, the default branch (i.e. the one to the remote HEAD points) will be checked out.

Note: The local branch name will match the remote branch name. If the branch name is changed, then that branch will be checked out locally, but keep in mind that remote repository will not be fetched. If your use case requires that you keep the clone up to date with the remote repository, then consider using *`git.latest`*.

---

user User under which to run git commands. By default, commands are run by the user under which the minion is running.

password Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

identity Path to a private key to use for ssh URLs. Works the same way as in *`git.latest`*, see that state's documentation for more information.

https_user HTTP Basic Auth username for HTTPS (only) clones

https_pass HTTP Basic Auth password for HTTPS (only) clones

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

Note: This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

`salt.states.git.`**`config_set`**`(`*name, value=None, multivar=None, repo=None, user=None, password=None, output_encoding=None, **kwargs*`)`

New in version 2014.7.0.

Changed in version 2015.8.0: Renamed from `git.config` to `git.config_set`. For earlier versions, use `git.config`.

Ensure that a config value is set to the desired value(s)

name Name of the git config value to set

value Set a single value for the config item

multivar Set multiple values for the config item

Note: The order matters here, if the same parameters are set but in a different order, they will be removed and replaced in the order specified.

New in version 2015.8.0.

repo Location of the git repository for which the config value should be set. Required unless `global` is set to `True`.

user User under which to run git commands. By default, the commands are run by the user under which the minion is running.

password

Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

New in version 2016.3.4.

global [False] If `True`, this will set a global git config option

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

Note: This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

New in version 2018.3.1.

Local Config Example:

```
# Single value
mylocalrepo:
  git.config_set:
    - name: user.email
    - value: foo@bar.net
    - repo: /path/to/repo

# Multiple values
mylocalrepo:
  git.config_set:
    - name: mysection.myattribute
    - multivar:
      - foo
      - bar
      - baz
    - repo: /path/to/repo
```

Global Config Example (User ``foo``):

```
mylocalrepo:
  git.config_set:
    - name: user.name
    - value: Foo Bar
    - user: foo
    - global: True
```

`salt.states.git.`**`config_unset`**`(`*name*, *value_regex=None*, *repo=None*, *user=None*, *password=None*, *output_encoding=None*, ***kwargs*`)`

New in version 2015.8.0.

Ensure that the named config key is not present

name The name of the configuration key to unset. This value can be a regex, but the regex must match the entire key name. For example, `foo\.` would not match all keys in the `foo` section, it would be necessary to use `foo\..+` to do so.

value_regex Regex indicating the values to unset for the matching key(s)

Note: This option behaves differently depending on whether or not `all` is set to `True`. If it is, then all values matching the regex will be deleted (this is the only way to delete multiple values from a multivar). If `all` is set to `False`, then this state will fail if the regex matches more than one value in a multivar.

---

all [False] If `True`, unset all matches

repo Location of the git repository for which the config value should be set. Required unless `global` is set to `True`.

user User under which to run git commands. By default, commands are run by the user under which the minion is running.

password

Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

New in version 2016.3.4.

global [False] If `True`, this will set a global git config option

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

Note: This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

New in version 2018.3.1.

Examples:

```
# Value matching 'baz'
mylocalrepo:
  git.config_unset:
    - name: foo.bar
    - value_regex: 'baz'
    - repo: /path/to/repo

# Ensure entire multivar is unset
mylocalrepo:
  git.config_unset:
    - name: foo.bar
    - all: True

# Ensure all variables in 'foo' section are unset, including multivars
mylocalrepo:
  git.config_unset:
    - name: 'foo\..+'
    - all: True

# Ensure that global config value is unset
mylocalrepo:
  git.config_unset:
    - name: foo.bar
    - global: True
```

`salt.states.git.`**`detached`**(*name, rev, target=None, remote='origin', user=None, password=None, force_clone=False, force_checkout=False, fetch_remote=True, hard_reset=False, submodules=False, identity=None, https_user=None, https_pass=None, onlyif=None, unless=None, output_encoding=None, **kwargs*)

New in version 2016.3.0.

Make sure a repository is cloned to the given target directory and is a detached HEAD checkout of the commit ID resolved from `rev`.

name Address of the remote repository.

rev The branch, tag, or commit ID to checkout after clone. If a branch or tag is specified it will be resolved to a commit ID and checked out.

target Name of the target directory where repository is about to be cloned.

remote [origin] Git remote to use. If this state needs to clone the repo, it will clone it using this value as the initial remote name. If the repository already exists, and a remote by this name is not present, one will be added.

user User under which to run git commands. By default, commands are run by the user under which the minion is running.

password

Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

New in version 2016.3.4.

force_clone [False] If the `target` directory exists and is not a git repository, then this state will fail. Set this argument to `True` to remove the contents of the target directory and clone the repo into it.

force_checkout [False] When checking out the revision ID, the state will fail if there are unwritten changes. Set this argument to `True` to discard unwritten changes when checking out.

fetch_remote [True] If `False` a fetch will not be performed and only local refs will be reachable.

hard_reset [False] If `True` a hard reset will be performed before the checkout and any uncommitted modifications to the working directory will be discarded. Untracked files will remain in place.

Note: Changes resulting from a hard reset will not trigger requisites.

---

submodules [False] Update submodules

identity A path on the minion (or a SaltStack fileserver URL, e.g. `salt://path/to/identity_file`) to a private key to use for SSH authentication.

https_user HTTP Basic Auth username for HTTPS (only) clones

https_pass HTTP Basic Auth password for HTTPS (only) clones

onlyif A command to run as a check, run the named command only if the command passed to the `onlyif` option returns true

unless A command to run as a check, only run the named command if the command passed to the `unless` option returns false

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

Note: This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

New in version 2018.3.1.

`salt.states.git.`**`latest`**`(`*name, rev='HEAD', target=None, branch=None, user=None, password=None, update_head=True, force_checkout=False, force_clone=False, force_fetch=False, force_reset=False, submodules=False, bare=False, mirror=False, remote='origin', fetch_tags=True, sync_tags=True, depth=None, identity=None, https_user=None, https_pass=None, onlyif=None, unless=None, refspec_branch='*', refspec_tag='*', output_encoding=None, **kwargs*`)`

Make sure the repository is cloned to the given directory and is up-to-date.

name Address of the remote repository, as passed to `git clone`

Note: From the Git documentation, there are two URL formats supported for SSH authentication. The below two examples are equivalent:

```
# ssh:// URL
ssh://user@server/project.git

# SCP-like syntax
user@server:project.git
```

A common mistake is to use an `ssh://` URL, but with a colon after the domain instead of a slash. This is invalid syntax in Git, and will therefore not work in Salt. When in doubt, confirm that a `git clone` works for the URL before using it in Salt.

It has been reported by some users that SCP-like syntax is incompatible with git repos hosted on Atlassian Stash/BitBucket Server. In these cases, it may be necessary to use `ssh://` URLs for SSH authentication.

---

rev [HEAD] The remote branch, tag, or revision ID to checkout after clone / before update. If specified, then Salt will also ensure that the tracking branch is set to `<remote>/<rev>`, unless `rev` refers to a tag or SHA1, in which case Salt will ensure that the tracking branch is unset.

If `rev` is not specified, it will be assumed to be `HEAD`, and Salt will not manage the tracking branch at all.

Changed in version 2015.8.0: If not specified, `rev` now defaults to the remote repository's HEAD.

target Name of the target directory where repository is about to be cloned

branch Name of the local branch into which to checkout the specified rev. If not specified, then Salt will not care what branch is being used locally and will just use whatever branch is currently there.

New in version 2015.8.0.

Note: If this argument is not specified, this means that Salt will not change the local branch if the repository is reset to another branch/tag/SHA1. For example, assume that the following state was run initially:

```
foo_app:
  git.latest:
    - name: https://mydomain.tld/apps/foo.git
    - target: /var/www/foo
    - user: www
```

This would have cloned the HEAD of that repo (since a `rev` wasn't specified), and because `branch` is not specified, the branch in the local clone at `/var/www/foo` would be whatever the default branch is on the remote repository (usually `master`, but not always). Now, assume that it becomes necessary to switch this checkout to the `dev` branch. This would require `rev` to be set, and probably would also require `force_reset` to be enabled:

```
foo_app:
  git.latest:
    - name: https://mydomain.tld/apps/foo.git
    - target: /var/www/foo
    - user: www
    - rev: dev
    - force_reset: True
```

The result of this state would be to perform a hard-reset to `origin/dev`. Since `branch` was not specified though, while `/var/www/foo` would reflect the contents of the remote repo's `dev` branch, the local branch would still remain whatever it was when it was cloned. To make the local branch match the remote one, set `branch` as well, like so:

```
foo_app:
  git.latest:
    - name: https://mydomain.tld/apps/foo.git
    - target: /var/www/foo
    - user: www
    - rev: dev
    - branch: dev
    - force_reset: True
```

This may seem redundant, but Salt tries to support a wide variety of use cases, and doing it this way allows for the use case where the local branch doesn't need to be strictly managed.

---

user Local system user under which to run git commands. By default, commands are run by the user under which the minion is running.

Note: This is not to be confused with the username for http(s)/SSH authentication.

New in version 0.17.0.

password Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

New in version 2016.3.4.

update_head [True] If set to `False`, then the remote repository will be fetched (if necessary) to ensure that the commit to which `rev` points exists in the local checkout, but no changes will be made to the local HEAD.

New in version 2015.8.3.

force_checkout [False] When checking out the local branch, the state will fail if there are unwritten changes. Set this argument to `True` to discard unwritten changes when checking out.

force_clone [False] If the `target` directory exists and is not a git repository, then this state will fail. Set this argument to `True` to remove the contents of the target directory and clone the repo into it.

force_fetch [False] If a fetch needs to be performed, non-fast-forward fetches will cause this state to fail. Set this argument to `True` to force the fetch even if it is a non-fast-forward update.

New in version 2015.8.0.

force_reset [False] If the update is not a fast-forward, this state will fail. Set this argument to `True` to force a hard-reset to the remote revision in these cases.

Changed in version 2019.2.0: This option can now be set to `remote-changes`, which will instruct Salt not to discard local changes if the repo is up-to-date with the remote repository.

submodules [False] Update submodules on clone or branch change

bare [False] Set to `True` if the repository is to be a bare clone of the remote repository.

mirror Set to `True` if the repository is to be a mirror of the remote repository. This implies that `bare` set to `True`, and thus is incompatible with `rev`.

remote [origin] Git remote to use. If this state needs to clone the repo, it will clone it using this value as the initial remote name. If the repository already exists, and a remote by this name is not present, one will be added.

fetch_tags [True] If `True`, then when a fetch is performed all tags will be fetched, even those which are not reachable by any branch on the remote.

sync_tags [True] If `True`, then Salt will delete tags which exist in the local clone but are not found on the remote repository.

New in version 2018.3.4.

depth Defines depth in history when git a clone is needed in order to ensure latest. E.g. `depth: 1` is useful when deploying from a repository with a long history. Use rev to specify branch or tag. This is not compatible with revision IDs.

Changed in version 2019.2.0: This option now supports tags as well as branches, on Git 1.8.0 and newer.

identity Path to a private key to use for ssh URLs. This can be either a single string, or a list of strings. For example:

```yaml
# Single key
git@github.com:user/repo.git:
  git.latest:
    - user: deployer
    - identity: /home/deployer/.ssh/id_rsa

# Two keys
git@github.com:user/repo.git:
  git.latest:
    - user: deployer
    - identity:
      - /home/deployer/.ssh/id_rsa
      - /home/deployer/.ssh/id_rsa_alternate
```

If multiple keys are specified, they will be tried one-by-one in order for each git command which needs to authenticate.

> **Warning:** Unless Salt is invoked from the minion using `salt-call`, the key(s) must be passphraseless. For greater security with passphraseless private keys, see the sshd(8) manpage for information on securing the keypair from the remote side in the `authorized_keys` file.

Changed in version 2015.8.7: Salt will no longer attempt to use passphrase-protected keys unless invoked from the minion using `salt-call`, to prevent blocking waiting for user input.

Changed in version 2016.3.0: Key can now be specified as a SaltStack fileserver URL (e.g. `salt://path/to/identity_file`).

https_user HTTP Basic Auth username for HTTPS (only) clones

New in version 2015.5.0.

https_pass HTTP Basic Auth password for HTTPS (only) clones

New in version 2015.5.0.

onlyif A command to run as a check, run the named command only if the command passed to the `onlyif` option returns true

unless A command to run as a check, only run the named command if the command passed to the `unless` option returns false

refspec_branch [*] A glob expression defining which branches to retrieve when fetching. See git-fetch(1) for more information on how refspecs work.

New in version 2017.7.0.

refspec_tag [*] A glob expression defining which tags to retrieve when fetching. See git-fetch(1) for more information on how refspecs work.

New in version 2017.7.0.

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

> **Note:** This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

New in version 2018.3.1.

Note: Clashing ID declarations can be avoided when including different branches from the same git repository in the same SLS file by using the `name` argument. The example below checks out the `gh-pages` and `gh-pages-prod` branches from the same repository into separate directories. The example also sets up the `ssh_known_hosts` ssh key required to perform the git checkout.

Also, it has been reported that the SCP-like syntax for

```yaml
gitlab.example.com:
  ssh_known_hosts:
    - present
    - user: root
    - enc: ecdsa
    - fingerprint: 4e:94:b0:54:c1:5b:29:a2:70:0e:e1:a3:51:ee:ee:e3

git-website-staging:
  git.latest:
    - name: git@gitlab.example.com:user/website.git
    - rev: gh-pages
```

(continues on next page)

(continued from previous page)

```
    - target: /usr/share/nginx/staging
    - identity: /root/.ssh/website_id_rsa
    - require:
      - pkg: git
      - ssh_known_hosts: gitlab.example.com

git-website-staging:
  git.latest:
    - name: git@gitlab.example.com:user/website.git
    - rev: gh-pages
    - target: /usr/share/nginx/staging
    - identity: salt://website/id_rsa
    - require:
      - pkg: git
      - ssh_known_hosts: gitlab.example.com

git-website-prod:
  git.latest:
    - name: git@gitlab.example.com:user/website.git
    - rev: gh-pages-prod
    - target: /usr/share/nginx/prod
    - identity: /root/.ssh/website_id_rsa
    - require:
      - pkg: git
      - ssh_known_hosts: gitlab.example.com
```

`salt.states.git.`**`mod_run_check`**`(`*cmd_kwargs, onlyif, unless*`)`

Execute the onlyif and unless logic. Return a result dict if:

- onlyif failed (onlyif != 0)
- unless succeeded (unless == 0)

Otherwise, returns `True`

`salt.states.git.`**`present`**`(`*name, force=False, bare=True, template=None, separate_git_dir=None, shared=None, user=None, password=None, output_encoding=None*`)`

Ensure that a repository exists in the given directory

> **Warning:** If the minion has Git 2.5 or later installed, `name` points to a worktree, and `force` is set to `True`, then the worktree will be deleted. This has been corrected in Salt 2015.8.0.

name Path to the directory

Changed in version 2015.8.0: This path must now be absolute

force [False] If `True`, and if `name` points to an existing directory which does not contain a git repository, then the contents of that directory will be recursively removed and a new repository will be initialized in its place.

bare [True] If `True`, and a repository must be initialized, then the repository will be a bare repository.

> **Note:** This differs from the default behavior of `git.init`, make sure to set this value to `False` if a bare repo is not desired.

template If a new repository is initialized, this argument will specify an alternate template directory.

New in version 2015.8.0.

separate_git_dir If a new repository is initialized, this argument will specify an alternate `$GIT_DIR`

New in version 2015.8.0.

shared Set sharing permissions on git repo. See git-init(1) for more details.

New in version 2015.5.0.

user User under which to run git commands. By default, commands are run by the user under which the minion is running.

New in version 0.17.0.

password

Windows only. Required when specifying `user`. This parameter will be ignored on non-Windows platforms.

New in version 2016.3.4.

output_encoding Use this option to specify which encoding to use to decode the output from any git commands which are run. This should not be needed in most cases.

Note: This should only be needed if the files in the repository were created with filenames using an encoding other than UTF-8 to handle Unicode characters.

New in version 2018.3.1.

21.19.90 salt.states.github module

Github User State Module

New in version 2016.3.0..

This state is used to ensure presence of users in the Organization.

```
ensure user test is present in github:
    github.present:
        - name: 'Example TestUser1'
        - email: example@domain.com
        - username: 'gitexample'
```

`salt.states.github.`**`absent`**`(`*name*, *profile='github'*, ***kwargs*`)`

Ensure a github user is absent

```
ensure user test is absent in github:
    github.absent:
        - name: 'Example TestUser1'
        - email: example@domain.com
        - username: 'gitexample'
```

The following parameters are required:

name Github handle of the user in organization

`salt.states.github.`**`present`**`(`*name*, *profile='github'*, ***kwargs*`)`

Ensure a user is present

```
ensure user test is present in github:
    github.present:
        - name: 'gitexample'
```

The following parameters are required:

name This is the github handle of the user in the organization `salt.states.github.`**`repo_absent`**(*name*, *profile='github'*, ***kwargs*)

Ensure a repo is absent.

Example:

```
ensure repo test is absent in github:
    github.repo_absent:
        - name: 'test'
```

The following parameters are required:

name This is the name of the repository in the organization.

New in version 2016.11.0.

`salt.states.github.`**`repo_present`**(*name*, *description=None*, *homepage=None*, *private=None*, *has_issues=None*, *has_wiki=None*, *has_downloads=None*, *auto_init=False*, *gitignore_template=None*, *license_template=None*, *teams=None*, *profile='github'*, ***kwargs*)

Ensure a repository is present

name This is the name of the repository.

description The description of the repository.

homepage The URL with more information about the repository.

private The visiblity of the repository. Note that private repositories require a paid GitHub account.

has_issues Whether to enable issues for this repository.

has_wiki Whether to enable the wiki for this repository.

has_downloads Whether to enable downloads for this repository.

auto_init Whether to create an initial commit with an empty README.

gitignore_template The desired language or platform for a .gitignore, e.g "Haskell".

license_template The desired LICENSE template to apply, e.g "mit" or "mozilla".

teams The teams for which this repo should belong to, specified as a dict of team name to permission ('pull', 'push' or 'admin').

New in version 2017.7.0.

Example:

```
Ensure repo my-repo is present in github:
    github.repo_present:
        - name: 'my-repo'
        - description: 'My very important repository'
```

New in version 2016.11.0.

`salt.states.github.`**`team_absent`**(*name*, *profile='github'*, ***kwargs*)

Ensure a team is absent.

Example:

```
ensure team test is present in github:
    github.team_absent:
        - name: 'test'
```

The following parameters are required:

name This is the name of the team in the organization.

New in version 2016.11.0.

`salt.states.github.`**`team_present`**(*name*, *description=None*, *repo_names=None*, *privacy='secret'*, *permission='pull'*, *members=None*, *enforce_mfa=False*, *no_mfa_grace_seconds=0*, *profile='github'*, ***kwargs*)

Ensure a team is present

name This is the name of the team in the organization.

description The description of the team.

repo_names The names of repositories to add the team to.

privacy The level of privacy for the team, can be 'secret' or 'closed'. Defaults to secret.

permission The default permission for new repositories added to the team, can be 'pull', 'push' or 'admin'. Defaults to pull.

members The members belonging to the team, specified as a dict of member name to optional configuration. Options include 'enforce_mfa_from' and 'mfa_exempt'.

enforce_mfa Whether to enforce MFA requirements on members of the team. If True then all members without *mfa_exempt: True* configured will be removed from the team. Note that *no_mfa_grace_seconds* may be set to allow members a grace period.

no_mfa_grace_seconds The number of seconds of grace time that a member will have to enable MFA before being removed from the team. The grace period will begin from *enforce_mfa_from* on the member configuration, which defaults to 1970/01/01.

Example:

```
Ensure team test is present in github:
    github.team_present:
        - name: 'test'
        - members:
            user1: {}
            user2: {}

Ensure team test_mfa is present in github:
    github.team_present:
        - name: 'test_mfa'
        - members:
            user1:
                enforce_mfa_from: 2016/06/15
        - enforce_mfa: True
```

New in version 2016.11.0.

21.19.91 salt.states.glance

Managing Images in OpenStack Glance

`salt.states.glance.`**`image_present`**(*name*, *visibility='public'*, *protected=None*, *checksum=None*, *location=None*, *disk_format='raw'*, *wait_for=None*, *timeout=30*)

Checks if given image is present with properties set as specified.

An image should got through the stages 'queued', 'saving' before becoming 'active'. The attribute 'checksum' can only be checked once the image is active. If you don't specify 'wait_for' but 'checksum' the function will wait for the image to become active before comparing checksums. If you don't specify checksum either the function will return when the image reached 'saving'. The default timeout for both is 30 seconds.

Supported properties:

- visibility ('public' or 'private')
- protected (bool)
- checksum (string, md5sum)
- location (URL, to copy from)
- disk_format ('raw' (default), 'vhd', 'vhdx', 'vmdk', 'vdi', 'iso', 'qcow2', 'aki', 'ari' or 'ami')

21.19.92 salt.states.glance_image

Management of OpenStack Glance Images

New in version 2018.3.0.

depends shade

configuration see `salt.modules.glanceng` for setup instructions

Example States

```
create image:
  glance_image.present:
    - name: cirros
    - filename: cirros.raw
    - image_format: raw

delete image:
  glance_image.absent:
    - name: cirros
```

`salt.states.glance_image.`**`absent`**`(`*name, auth=None*`)`
Ensure image does not exist
name Name of the image

`salt.states.glance_image.`**`present`**`(`*name, auth=None, **kwargs*`)`
Ensure image exists and is up-to-date
name Name of the image
enabled Boolean to control if image is enabled
description An arbitrary description of the image

21.19.93 salt.states.glusterfs

Manage GlusterFS pool.

`salt.states.glusterfs.`**`add_volume_bricks`**`(`*name, bricks*`)`
Add brick(s) to an existing volume
name Volume name
bricks List of bricks to add to the volume

```
myvolume:
  glusterfs.add_volume_bricks:
    - bricks:
        - host1:/srv/gluster/drive1
        - host2:/srv/gluster/drive2

Replicated Volume:
  glusterfs.add_volume_bricks:
    - name: volume2
    - bricks:
      - host1:/srv/gluster/drive2
      - host2:/srv/gluster/drive3
```

`salt.states.glusterfs.`**`max_op_version`**`(`*name*`)`
New in version 2019.2.0.

Add brick(s) to an existing volume
name Volume name

```
myvolume:
  glusterfs.max_op_version:
    - name: volume1
    - version: 30707
```

salt.states.glusterfs.**op_version**(*name*, *version*)

New in version 2019.2.0.

Add brick(s) to an existing volume

name Volume name

version Version to which the cluster.op-version should be set

```
myvolume:
  glusterfs.op_version:
    - name: volume1
    - version: 30707
```

salt.states.glusterfs.**peered**(*name*)

Check if node is peered.

name The remote host with which to peer.

```
peer-cluster:
  glusterfs.peered:
    - name: two

peer-clusters:
  glusterfs.peered:
    - names:
      - one
      - two
      - three
      - four
```

salt.states.glusterfs.**started**(*name*)

Check if volume has been started

name name of the volume

```
mycluster:
  glusterfs.started: []
```

salt.states.glusterfs.**volume_present**(*name*, *bricks*, *stripe=False*, *replica=False*, *device_vg=False*, *transport='tcp'*, *start=False*, *force=False*, *arbiter=False*)

Ensure that the volume exists

name name of the volume

bricks list of brick paths

replica replica count for volume

arbiter use every third brick as arbiter (metadata only)

New in version 2019.2.0.

start ensure that the volume is also started

```
myvolume:
  glusterfs.volume_present:
    - bricks:
        - host1:/srv/gluster/drive1
        - host2:/srv/gluster/drive2
```

(continues on next page)

(continued from previous page)

```
Replicated Volume:
  glusterfs.volume_present:
    - name: volume2
    - bricks:
      - host1:/srv/gluster/drive2
      - host2:/srv/gluster/drive3
    - replica: 2
    - start: True

Replicated Volume with arbiter brick:
  glusterfs.volume_present:
    - name: volume3
    - bricks:
      - host1:/srv/gluster/drive2
      - host2:/srv/gluster/drive3
      - host3:/srv/gluster/drive4
    - replica: 3
    - arbiter: True
    - start: True
```

21.19.94 salt.states.gnomedesktop

Configuration of the GNOME desktop

Control the GNOME settings

```
localdesktop_wm_prefs:
    gnomedesktop.wm_preferences:
        - user: username
        - audible_bell: false
        - action_double_click_titlebar: 'toggle-maximize'
        - visual_bell: true
        - num_workspaces: 6
localdesktop_lockdown:
    gnomedesktop.desktop_lockdown:
        - user: username
        - disable_user_switching: true
localdesktop_interface:
    gnomedesktop.desktop_interface:
        - user: username
        - clock_show_date: true
        - clock_format: 12h
```

`salt.states.gnomedesktop.`**`desktop_interface`**`(`*name, user=None, automatic_mnemonics=None, buttons_have_icons=None, can_change_accels=None, clock_format=None, clock_show_date=None, clock_show_seconds=None, cursor_blink=None, cursor_blink_time=None, cursor_blink_timeout=None, cursor_size=None, cursor_theme=None, document_font_name=None, enable_animations=None, font_name=None, gtk_color_palette=None, gtk_color_scheme=None, gtk_im_module=None, gtk_im_preedit_style=None, gtk_im_status_style=None, gtk_key_theme=None, gtk_theme=None, gtk_timeout_initial=None, gtk_timeout_repeat=None, icon_theme=None, menubar_accel=None, menubar_detachable=None, menus_have_icons=None, menus_have_tearoff=None, monospace_font_name=None, show_input_method_menu=None, show_unicode_menu=None, text_scaling_factor=None, toolbar_detachable=None, toolbar_icons_size=None, toolbar_style=None, toolkit_accessibility=None, **kwargs*`)`

desktop_interface: sets values in the org.gnome.desktop.interface schema

`salt.states.gnomedesktop.`**`desktop_lockdown`**`(`*name, user=None, disable_application_handlers=None, disable_command_line=None, disable_lock_screen=None, disable_log_out=None, disable_print_setup=None, disable_printing=None, disable_save_to_disk=None, disable_user_switching=None, user_administration_disabled=None, **kwargs*`)`

desktop_lockdown: sets values in the org.gnome.desktop.lockdown schema `salt.states.gnomedesktop.`**`wm_preferences`**(*name*, *user=None*, *action_double_click_titlebar=None*, *action_middle_click_titlebar=None*, *action_right_click_titlebar=None*, *application_based=None*, *audible_bell=None*, *auto_raise=None*, *auto_raise_delay=None*, *button_layout=None*, *disable_workarounds=None*, *focus_mode=None*, *focus_new_windows=None*, *mouse_button_modifier=None*, *num_workspaces=None*, *raise_on_click=None*, *resize_with_right_button=None*, *theme=None*, *titlebar_font=None*, *titlebar_uses_system_font=None*, *visual_bell=None*, *visual_bell_type=None*, *workspace_names=None*, ***kwargs*)

wm_preferences: sets values in the org.gnome.desktop.wm.preferences schema

21.19.95 salt.states.gpg module

Management of the GPG keychains

New in version 2016.3.0.

`salt.states.gpg.`**`absent`**(*name*, *keys=None*, *user=None*, *gnupghome=None*, ***kwargs*)

Ensure GPG public key is absent in keychain

name The unique name or keyid for the GPG public key.

keys The keyId or keyIds to add to the GPG keychain.

user Remove GPG keys from the specified user's keychain

gnupghome Override GNUPG Home directory

`salt.states.gpg.`**`present`**(*name*, *keys=None*, *user=None*, *keyserver=None*, *gnupghome=None*, *trust=None*, ***kwargs*)

Ensure GPG public key is present in keychain

name The unique name or keyid for the GPG public key.

keys The keyId or keyIds to add to the GPG keychain.

user Add GPG keys to the specified user's keychain

keyserver The keyserver to retrieve the keys from.

gnupghome Override GNUPG Home directory

trust Trust level for the key in the keychain, ignored by default. Valid trust levels: expired, unknown, not_trusted, marginally, fully, ultimately

21.19.96 salt.states.grafana

Manage Grafana Dashboards

This module uses `elasticsearch`, which can be installed via package, or pip.

You can specify elasticsearch hosts directly to the module, or you can use an elasticsearch profile via pillars:

```
mygrafanaprofile:
  hosts:
    - es1.example.com:9200
    - es2.example.com:9200
  index: grafana-dash
```

```
# Basic usage (uses default pillar profile key 'grafana')
Ensure myservice dashboard is managed:
  grafana.dashboard_present:
    - name: myservice
    - dashboard_from_pillar: default
    - rows_from_pillar:
        - systemhealth
        - requests

# Passing hosts in
Ensure myservice dashboard is managed:
  grafana.dashboard_present:
    - name: myservice
    - dashboard_from_pillar: default
    - rows:
        - collapse: false
          editable: true
          height: 150px
          title: System Health
          panels:
            - aliasColors: {}
              id: 200000
              annotate:
                enable: false
              bars: false
              datasource: null
              editable: true
              error: false
              fill: 7
              grid:
                leftMax: 100
                leftMin: null
                rightMax: null
                rightMin: null
                threshold1: 60
                threshold1Color: rgb(216, 27, 27)
                threshold2: null
                threshold2Color: rgba(234, 112, 112, 0.22)
              leftYAxisLabel: ''
              legend:
                avg: false
                current: false
                max: false
                min: false
                show: false
                total: false
                values: false
              lines: true
              linewidth: 1
              nullPointMode: connected
              percentage: false
              pointradius: 5
              points: false
              renderer: flot
              resolution: 100
              scale: 1
              seriesOverrides: []
```

(continues on next page)

(continued from previous page)

```
              span: 4
              stack: false
              steppedLine: false
              targets:
                - target: cloudwatch.aws.ec2.mysrv.cpuutilization.average
              title: CPU (asg average)
              tooltip:
                query_as_alias: true
                shared: false
                value_type: cumulative
              type: graph
              x-axis: true
              y-axis: true
              y_formats:
                - short
                - short
              zerofill: true
    - rows_from_pillar:
      - systemhealth
      - requests
    - profile:
        hosts:
          - es1.example.com:9200
          - es2.example.com:9200
        index: grafana-dash

# Using a profile from pillars
Ensure myservice dashboard is managed:
  grafana.dashboard_present:
    - name: myservice
    - dashboard:
        annotations:
          enable: true
          list: []
        editable: true
        hideAllLegends: false
        hideControls: false
        nav:
          - collapse: false
            enable: true
            notice: false
            now: true
            refresh_intervals:
              - 10s
              - 30s
              - 1m
              - 5m
              - 15m
              - 30m
              - 1h
              - 2h
              - 1d
            status: Stable
            time_options:
              - 5m
              - 15m
```

(continues on next page)

(continued from previous page)

```
              - 1h
              - 2h
              - 3h
              - 4h
              - 6h
              - 12h
              - 1d
              - 2d
              - 4d
              - 7d
              - 16d
              - 30d
            type: timepicker
        originalTitle: dockerregistry
        refresh: 1m
        rows: []
        sharedCrosshair: false
        style: dark
        tags: []
        templating:
          enable: true
          list: []
        time:
          from: now-2h
          to: now
        timezone: browser
    - rows_from_pillars:
      - systemhealth
      - requests
    - profile: mygrafanaprofile
```

The behavior of this module is to create dashboards if they do not exist, to add rows if they do not exist in existing dashboards, and to update rows if they exist in dashboards. The module will not manage rows that are not defined, allowing users to manage their own custom rows.

`salt.states.grafana.`**`dashboard_absent`**(*name*, *hosts=None*, *profile='grafana'*)

Ensure the named grafana dashboard is deleted.

name Name of the grafana dashboard.

profile A pillar key or dict that contains a list of hosts and an elasticsearch index to use.

`salt.states.grafana.`**`dashboard_present`**(*name*, *dashboard=None*, *dashboard_from_pillar=None*, *rows=None*, *rows_from_pillar=None*, *profile='grafana'*)

Ensure the grafana dashboard exists and is managed.

name Name of the grafana dashboard.

dashboard A dict that defines a dashboard that should be managed.

dashboard_from_pillar A pillar key that contains a grafana dashboard dict. Mutually exclusive with dashboard.

rows A list of grafana rows.

rows_from_pillar A list of pillar keys that contain lists of grafana dashboard rows. Rows defined in the pillars will be appended to the rows defined in the state.

profile A pillar key or dict that contains a list of hosts and an elasticsearch index to use.

21.19.97 salt.states.grafana4_dashboard module

Manage Grafana v4.0 Dashboards

New in version 2017.7.0.

configuration This state requires a configuration profile to be configured in the minion config, minion pillar, or master config. The module will use the 'grafana' key by default, if defined.

Example configuration using basic authentication:

```
grafana:
  grafana_url: http://grafana.localhost
  grafana_user: admin
  grafana_password: admin
  grafana_timeout: 3
```

Example configuration using token based authentication:

```
grafana:
  grafana_url: http://grafana.localhost
  grafana_token: token
  grafana_timeout: 3
```

The behavior of this module is to create dashboards if they do not exist, to add rows if they do not exist in existing dashboards, and to update rows if they exist in dashboards. The module will not manage rows that are not defined, allowing users to manage their own custom rows.

```
Ensure minimum dashboard is managed:
  grafana4_dashboard.present:
    - name: insightful-dashboard
    - base_dashboards_from_pillar:
      - default_dashboard
    - base_rows_from_pillar:
      - default_row
    - base_panels_from_pillar:
      - default_panel
    - dashboard:
        rows:
          - title: Usage
            panels:
              - targets:
                  - target: alias(constantLine(50), 'max')
                title: Imaginary
                type: graph
```

`salt.states.grafana4_dashboard.`**`absent`**(*name, orgname=None, profile='grafana'*)

Ensure the named grafana dashboard is absent.

name Name of the grafana dashboard.

orgname Name of the organization in which the dashboard should be present.

profile Configuration profile used to connect to the Grafana instance. Default is 'grafana'.

`salt.states.grafana4_dashboard.`**`present`**(*name, base_dashboards_from_pillar=None, base_panels_from_pillar=None, base_rows_from_pillar=None, dashboard=None, orgname=None, profile='grafana'*)

Ensure the grafana dashboard exists and is managed.

name Name of the grafana dashboard.

base_dashboards_from_pillar A pillar key that contains a list of dashboards to inherit from

base_panels_from_pillar A pillar key that contains a list of panels to inherit from

base_rows_from_pillar A pillar key that contains a list of rows to inherit from

dashboard A dict that defines a dashboard that should be managed.

orgname Name of the organization in which the dashboard should be present.

profile Configuration profile used to connect to the Grafana instance. Default is 'grafana'.

21.19.98 salt.states.grafana4_datasource module

Manage Grafana v4.0 data sources

New in version 2017.7.0.

configuration This state requires a configuration profile to be configured in the minion config, minion pillar, or master config. The module will use the 'grafana' key by default, if defined.

Example configuration using basic authentication:

```
grafana:
  grafana_url: http://grafana.localhost
  grafana_user: admin
  grafana_password: admin
  grafana_timeout: 3
```

Example configuration using token based authentication:

```
grafana:
  grafana_url: http://grafana.localhost
  grafana_token: token
  grafana_timeout: 3
```

The behavior of this module is to create data sources if the do not exists, and to update data sources if the already exists.

```
Ensure influxdb data source is present:
  grafana4_datasource.present:
    - name: influxdb
    - type: influxdb
    - url: http://localhost:8086
    - access: proxy
    - basic_auth: true
    - basic_auth_user: myuser
    - basic_auth_password: mypass
    - is_default: true
```

`salt.states.grafana4_datasource.`**`absent`**`(`*name, orgname=None, profile='grafana'*`)`

Ensure that a data source is present.

name Name of the data source to remove.

orgname Name of the organization from which the data source should be absent.

profile Configuration profile used to connect to the Grafana instance. Default is 'grafana'.

`salt.states.grafana4_datasource.`**`present`**`(`*name, type, url, access=None, user=None, password=None, database=None, basic_auth=None, basic_auth_user=None, basic_auth_password=None, tls_auth=None, json_data=None, is_default=None, with_credentials=None, type_logo_url=None, orgname=None, profile='grafana'*`)`

Ensure that a data source is present.

name Name of the data source.
type Type of the datasource ('graphite', 'influxdb' etc.).
access Use proxy or direct. Default: proxy
url The URL to the data source API.
user Optional - user to authenticate with the data source.
password Optional - password to authenticate with the data source.
database Optional - database to use with the data source.
basic_auth Optional - set to True to use HTTP basic auth to authenticate with the data source.
basic_auth_user Optional - HTTP basic auth username.
basic_auth_password Optional - HTTP basic auth password.
json_data Optional - additional json data to post (eg. "timeInterval").
is_default Optional - set data source as default.
with_credentials Optional - Whether credentials such as cookies or auth headers should be sent with cross-site requests.
type_logo_url Optional - Logo to use for this datasource.
orgname Name of the organization in which the data source should be present.
profile Configuration profile used to connect to the Grafana instance. Default is 'grafana'.

21.19.99 salt.states.grafana4_org module

Manage Grafana v4.0 orgs

New in version 2017.7.0.

configuration This state requires a configuration profile to be configured in the minion config, minion pillar, or master config. The module will use the 'grafana' key by default, if defined.

Example configuration using basic authentication:

```
grafana:
  grafana_url: http://grafana.localhost
  grafana_user: admin
  grafana_password: admin
  grafana_timeout: 3
```

Example configuration using token based authentication:

```
grafana:
  grafana_url: http://grafana.localhost
  grafana_token: token
  grafana_timeout: 3
```

```
Ensure foobar org is present:
  grafana4_org.present:
    - name: foobar
    - theme:  ""
    - home_dashboard_id: 0
    - timezone: "utc"
    - address1: ""
    - address2: ""
    - city: ""
    - zip_code: ""
    - state: ""
    - country: ""
```

`salt.states.grafana4_org.`**`absent`**`(`*name, profile='grafana'*`)`
Ensure that a org is present.

name Name of the org to remove.

profile Configuration profile used to connect to the Grafana instance. Default is 'grafana'.

`salt.states.grafana4_org.`**`present`**(*name, users=None, theme=None, home_dashboard_id=None, timezone=None, address1=None, address2=None, city=None, zip_code=None, address_state=None, country=None, profile='grafana'*)

Ensure that an organization is present.

name Name of the org.

users Optional - Dict of user/role associated with the org. Example:

```
users:
  foo: Viewer
  bar: Editor
```

theme Optional - Selected theme for the org.

home_dashboard_id Optional - Home dashboard for the org.

timezone Optional - Timezone for the org (one of: "browser", "utc", or "").

address1 Optional - address1 of the org.

address2 Optional - address2 of the org.

city Optional - city of the org.

zip_code Optional - zip_code of the org.

address_state Optional - state of the org.

country Optional - country of the org.

profile Configuration profile used to connect to the Grafana instance. Default is 'grafana'.

21.19.100 salt.states.grafana4_user module

Manage Grafana v4.0 users

New in version 2017.7.0.

configuration This state requires a configuration profile to be configured in the minion config, minion pillar, or master config. The module will use the 'grafana' key by default, if defined.

Example configuration using basic authentication:

```
grafana:
  grafana_url: http://grafana.localhost
  grafana_user: admin
  grafana_password: admin
  grafana_timeout: 3
```

Example configuration using token based authentication:

```
grafana:
  grafana_url: http://grafana.localhost
  grafana_token: token
  grafana_timeout: 3
```

```
Ensure foobar user is present:
  grafana4_user.present:
    - name: foobar
    - password: mypass
    - email: "foobar@localhost"
    - fullname: Foo Bar
    - is_admin: true
```

`salt.states.grafana4_user.`**`absent`**`(`*name, profile='grafana'*`)`

Ensure that a user is present.

name Name of the user to remove.

profile Configuration profile used to connect to the Grafana instance. Default is 'grafana'.

`salt.states.grafana4_user.`**`present`**`(`*name, password, email, is_admin=False, fullname=None, theme=None, profile='grafana'*`)`

Ensure that a user is present.

name Name of the user.

password Password of the user.

email Email of the user.

is_admin Optional - Set user as admin user. Default: False

fullname Optional - Full name of the user.

theme Optional - Selected theme of the user.

profile Configuration profile used to connect to the Grafana instance. Default is 'grafana'.

21.19.101 salt.states.grafana_dashboard module

Manage Grafana v2.0 Dashboards

New in version 2016.3.0.

```
grafana:
  grafana_timeout: 3
  grafana_token: qwertyuiop
  grafana_url: 'https://url.com'
```

```
Ensure minimum dashboard is managed:
  grafana_dashboard.present:
    - name: insightful-dashboard
    - base_dashboards_from_pillar:
      - default_dashboard
    - base_rows_from_pillar:
      - default_row
    - base_panels_from_pillar:
      - default_panel
    - dashboard:
        rows:
          - title: Usage
            panels:
              - targets:
                  - target: alias(constantLine(50), 'max')
                title: Imaginary
                type: graph
```

The behavior of this module is to create dashboards if they do not exist, to add rows if they do not exist in existing dashboards, and to update rows if they exist in dashboards. The module will not manage rows that are not defined, allowing users to manage their own custom rows.

`salt.states.grafana_dashboard.`**`absent`**`(`*name, profile='grafana'*`)`

Ensure the named grafana dashboard is absent.

name Name of the grafana dashboard.

profile A pillar key or dict that contains grafana information `salt.states.grafana_dashboard.`**`present`**`(`*name, base_dashboards_from_pillar=None, base_panels_from_pillar=None, base_rows_from_pillar=None, dashboard=None, profile='grafana'*`)`

Ensure the grafana dashboard exists and is managed.

name Name of the grafana dashboard.

base_dashboards_from_pillar A pillar key that contains a list of dashboards to inherit from

base_panels_from_pillar A pillar key that contains a list of panels to inherit from

base_rows_from_pillar A pillar key that contains a list of rows to inherit from

dashboard A dict that defines a dashboard that should be managed.

profile A pillar key or dict that contains grafana information

21.19.102 salt.states.grafana_datasource module

Manage Grafana v2.0 data sources

New in version 2016.3.0.

```
grafana:
  grafana_timeout: 3
  grafana_token: qwertyuiop
  grafana_url: 'https://url.com'
```

```
Ensure influxdb data source is present:
  grafana_datasource.present:
    - name: influxdb
    - type: influxdb
    - url: http://localhost:8086
    - access: proxy
    - basic_auth: true
    - basic_auth_user: myuser
    - basic_auth_password: mypass
    - is_default: true
```

`salt.states.grafana_datasource.`**`absent`**`(`*name, profile='grafana'*`)`

Ensure that a data source is present.

name Name of the data source to remove.

`salt.states.grafana_datasource.`**`present`**`(`*name, type, url, access='proxy', user='', password='', database='', basic_auth=False, basic_auth_user='', basic_auth_password='', is_default=False, json_data=None, profile='grafana'*`)`

Ensure that a data source is present.

name Name of the data source.

type Which type of data source it is ('graphite', 'influxdb' etc.).

url The URL to the data source API.

user Optional - user to authenticate with the data source

password Optional - password to authenticate with the data source

basic_auth Optional - set to True to use HTTP basic auth to authenticate with the data source.

basic_auth_user Optional - HTTP basic auth username.

basic_auth_password Optional - HTTP basic auth password.

is_default Default: False

21.19.103 salt.states.grains

Manage grains on the minion

This state allows for grains to be set.

Grains set or altered with this module are stored in the 'grains' file on the minions, By default, this file is located at: `/etc/salt/grains`

Note: This does **NOT** override any grains set in the minion config file.

`salt.states.grains.`**`absent`**(*name*, *destructive=False*, *delimiter=':'*, *force=False*)

New in version 2014.7.0.

Delete a grain from the grains config file

name The grain name

destructive If destructive is True, delete the entire grain. If destructive is False, set the grain's value to None. Defaults to False.

force If force is True, the existing grain will be overwritten regardless of its existing or provided value type. Defaults to False

New in version v2015.8.2.

delimiter A delimiter different from the default can be provided.

New in version v2015.8.2.

Changed in version v2015.8.2.

This state now support nested grains and complex values. It is also more conservative: if a grain has a value that is a list or a dict, it will not be removed unless the *force* parameter is True.

```
grain_name:
  grains.absent
```

`salt.states.grains.`**`append`**(*name*, *value*, *convert=False*, *delimiter=':'*)

New in version 2014.7.0.

Append a value to a list in the grains config file. The grain that is being appended to (name) must exist before the new value can be added.

name The grain name

value The value to append

convert If convert is True, convert non-list contents into a list. If convert is False and the grain contains non-list contents, an error is given. Defaults to False.

delimiter A delimiter different from the default can be provided.

New in version v2015.8.2.

```
grain_name:
  grains.append:
    - value: to_be_appended
```

`salt.states.grains.`**`exists`**(*name*, *delimiter=':'*)

Ensure that a grain is set

name The grain name

delimiter A delimiter different from the default can be provided.

Check whether a grain exists. Does not attempt to check or set the value.

`salt.states.grains.`**`list_absent`**(*name*, *value*, *delimiter=':'*)

Delete a value from a grain formed as a list.

New in version 2014.1.0.

name The grain name.
value The value to delete from the grain list.
delimiter A delimiter different from the default **:** can be provided.

New in version v2015.8.2.

The grain should be list type

```
roles:
  grains.list_absent:
    - value: db
```

For multiple grains, the syntax looks like:

```
roles:
  grains.list_absent:
    - value:
      - web
      - dev
```

`salt.states.grains.`**`list_present`**(*name*, *value*, *delimiter=':'*)

New in version 2014.1.0.

Ensure the value is present in the list-type grain. Note: If the grain that is provided in `name` is not present on the system, this new grain will be created with the corresponding provided value.
name The grain name.
value The value is present in the list type grain.
delimiter A delimiter different from the default **:** can be provided.

New in version v2015.8.2.

The grain should be list type

```
roles:
  grains.list_present:
    - value: web
```

For multiple grains, the syntax looks like:

```
roles:
  grains.list_present:
    - value:
      - web
      - dev
```

`salt.states.grains.`**`make_hashable`**(*list_grain*, *result=None*)

Ensure that a list grain is hashable.
list_grain The list grain that should be hashable
result This function is recursive, so it must be possible to use a sublist as parameter to the function. Should not be used by a caller outside of the function.
Make it possible to compare two list grains to each other if the list contains complex objects.

`salt.states.grains.`**`present`**(*name*, *value*, *delimiter=':'*, *force=False*)

Ensure that a grain is set

Changed in version v2015.8.2.
name The grain name
value The value to set on the grain
force If force is True, the existing grain will be overwritten regardless of its existing or provided value type. Defaults to False

New in version v2015.8.2.

delimiter A delimiter different from the default can be provided.

New in version v2015.8.2.

It is now capable to set a grain to a complex value (ie. lists and dicts) and supports nested grains as well.

If the grain does not yet exist, a new grain is set to the given value. For a nested grain, the necessary keys are created if they don't exist. If a given key is an existing value, it will be converted, but an existing value different from the given key will fail the state.

If the grain with the given name exists, its value is updated to the new value unless its existing or provided value is complex (list or dict). Use *force: True* to overwrite.

```
cheese:
  grains.present:
    - value: edam

nested_grain_with_complex_value:
  grains.present:
    - name: icinga:Apache SSL
    - value:
      - command: check_https
      - params: -H localhost -p 443 -S

with,a,custom,delimiter:
  grains.present:
    - value: yay
    - delimiter: ','
```

21.19.104 salt.states.group

Management of user groups

The group module is used to create and manage group settings, groups can be either present or absent. User/Group names can be passed to the `adduser`, `deluser`, and `members` parameters. `adduser` and `deluser` can be used together but not with `members`.

In Windows, if no domain is specified in the user or group name (i.e. `DOMAIN\username`) the module will assume a local user or group.

```
cheese:
  group.present:
    - gid: 7648
    - system: True
    - addusers:
      - user1
      - users2
    - delusers:
      - foo

cheese:
  group.present:
    - gid: 7648
    - system: True
    - members:
      - foo
      - bar
```

(continues on next page)

(continued from previous page)

```
    - user1
    - user2
```

`salt.states.group.`**`absent`**(*name*)

Ensure that the named group is absent

Parameters **name** (*str*) – The name of the group to remove

Example:

```
# Removes the local group `db_admin`
db_admin:
  group.absent
```

`salt.states.group.`**`present`**(*name*, *gid=None*, *system=False*, *addusers=None*, *delusers=None*, *members=None*)

Ensure that a group is present

Parameters

- **name** (*str*) – The name of the group to manage
- **gid** (*str*) – The group id to assign to the named group; if left empty, then the next available group id will be assigned. Ignored on Windows
- **system** (*bool*) – Whether or not the named group is a system group. This is essentially the '-r' option of 'groupadd'. Ignored on Windows
- **addusers** (*list*) – List of additional users to be added as a group members. Cannot conflict with names in delusers. Cannot be used in conjunction with members.
- **delusers** (*list*) – Ensure these user are removed from the group membership. Cannot conflict with names in addusers. Cannot be used in conjunction with members.
- **members** (*list*) – Replace existing group members with a list of new members. Cannot be used in conjunction with addusers or delusers.

Example:

```
# Adds DOMAIN\db_admins and Administrators to the local db_admin group
# Removes Users
db_admin:
  group.present:
    - addusers:
      - DOMAIN\db_admins
      - Administrators
    - delusers:
      - Users

# Ensures only DOMAIN\domain_admins and the local Administrator are
# members of the local Administrators group. All other users are
# removed
Administrators:
  group.present:
    - members:
      - DOMAIN\domain_admins
      - Administrator
```

21.19.105 salt.states.heat module

Management of Heat

New in version 2017.7.0.

depends

- heat Python module

configuration See `salt.modules.heat` for setup instructions.

The heat module is used to create, show, list and delete Heat staks. Stack can be set as either absent or deploy.

```
heat.deployed:
  - name:
  - template: #Required
  - environment:
  - params: {}
  - poll: 5
  - rollback: False
  - timeout: 60

heat.absent:
  - name:
  - poll: 5
```

mysql:

heat.deployed:

- template: salt://templates/mysql.heat.yaml
- params: image: Debian 7
- rollback: True

New in version 2017.7.5,2018.3.1: The spelling mistake in parameter *enviroment* was corrected to *environment*. The *enviroment* spelling mistake has been removed in Salt 3000.

`salt.states.heat.`**`absent`**`(`*name*, *poll=5*, *timeout=60*, *profile=None*`)`

Ensure that the named stack is absent

name The name of the stack to remove

poll Poll(in sec.) and report events until stack complete

timeout Stack creation timeout in minutes

profile Profile to use

`salt.states.heat.`**`deployed`**`(`*name*, *template=None*, *environment=None*, *params=None*, *poll=5*, *rollback=False*, *timeout=60*, *update=False*, *profile=None*, ***connection_args*`)`

Deploy stack with the specified properties

name The name of the stack

template File of template

environment File of environment

params Parameter dict used to create the stack

poll Poll (in sec.) and report events until stack complete

rollback Enable rollback on create failure

timeout Stack creation timeout in minutes

profile Profile to use

New in version 2017.7.5,2018.3.1: The spelling mistake in parameter *enviroment* was corrected to *environment*. The *enviroment* spelling mistake has been removed in Salt 3000.

21.19.106 salt.states.hg

Interaction with Mercurial repositories

Before using hg over ssh, make sure the remote host fingerprint already exists in ~/.ssh/known_hosts, and the remote host has this host's public key.

```
https://bitbucket.org/example_user/example_repo:
    hg.latest:
      - rev: tip
      - target: /tmp/example_repo
```

`salt.states.hg.latest`(*name*, *rev=None*, *target=None*, *clean=False*, *user=None*, *identity=None*, *force=False*, *opts=False*, *update_head=True*)

Make sure the repository is cloned to the given directory and is up to date

name Address of the remote repository as passed to "hg clone"

rev The remote branch, tag, or revision hash to clone/pull

target Target destination directory path on minion to clone into

clean Force a clean update with -C (Default: False)

user Name of the user performing repository management operations

New in version 0.17.0.

identity Private SSH key on the minion server for authentication (ssh://)

New in version 2015.5.0.

force Force hg to clone into pre-existing directories (deletes contents)

opts Include additional arguments and options to the hg command line

update_head Should we update the head if new changes are found? Defaults to True

New in version 2017.7.0.

21.19.107 salt.states.host

Management of addresses and names in hosts file

The `/etc/hosts` file can be managed to contain definitions for specific hosts:

```
salt-master:
  host.present:
    - ip: 192.168.0.42
```

Or using the `names` directive, you can put several names for the same IP. (Do not try one name with space-separated values).

```
server1:
  host.present:
    - ip: 192.168.0.42
    - names:
      - server1
      - florida
```

Note: Changing the `names` in `host.present` does not cause an update to remove the old entry.

```
server1:
  host.present:
    - ip:
```

(continues on next page)

(continued from previous page)

```
      - 192.168.0.42
      - 192.168.0.43
      - 192.168.0.44
    - names:
      - server1
```

You can replace all existing names for a particular IP address:

```
127.0.1.1:
  host.only:
    - hostnames:
      - foo.example.com
      - foo
```

Or delete all existing names for an address:

```
203.0.113.25:
    host.only:
      - hostnames: []
```

`salt.states.host.`**`absent`**`(`*name*, *ip*`)`
Ensure that the named host is absent

name The host to remove

ip The ip addr(s) of the host to remove

`salt.states.host.`**`only`**`(`*name*, *hostnames*`)`
Ensure that only the given hostnames are associated with the given IP address.

New in version 2016.3.0.

name The IP address to associate with the given hostnames.

hostnames Either a single hostname or a list of hostnames to associate with the given IP address in the given order. Any other hostname associated with the IP address is removed. If no hostnames are specified, all hostnames associated with the given IP address are removed.

`salt.states.host.`**`present`**`(`*name*, *ip*, *clean=False*`)`
Ensures that the named host is present with the given ip

name The host to assign an ip to

ip The ip addr(s) to apply to the host. Can be a single IP or a list of IP addresses.

clean Remove any entries which don't match those configured in the `ip` option. Default is `False`.

New in version 2018.3.4.

21.19.108 salt.states.http

HTTP monitoring states

Perform an HTTP query and statefully return the result

New in version 2015.5.0.

`salt.states.http.`**`query`**`(`*name*, *match=None*, *match_type='string'*, *status=None*, *status_type='string'*, *wait_for=None*, ***kwargs*`)`
Perform an HTTP query and statefully return the result

Passes through all the parameters described in the *`utils.http.query function`*:

name The name of the query.

match Specifies a pattern to look for in the return text. By default, this will perform a string comparison of looking for the value of match in the return text.

match_type Specifies the type of pattern matching to use on match. Default is `string`, but can also be set to `pcre` to use regular expression matching if a more complex pattern matching is required.

> **Note:** Despite the name of `match_type` for this argument, this setting actually uses Python's `re.search()` function rather than Python's `re.match()` function.

status The status code for a URL for which to be checked. Can be used instead of or in addition to the `match` setting. This can be passed as an individual status code or a list of status codes.

status_type Specifies the type of pattern matching to use for status. Default is `string`, but can also be set to `pcre` to use regular expression matching if a more complex pattern matching is required. Additionally, if a list of strings representing statuses is given, the type `list` can be used.

New in version 3000.

> **Note:** Despite the name of `match_type` for this argument, this setting actually uses Python's `re.search()` function rather than Python's `re.match()` function.

If both `match` and `status` options are set, both settings will be checked. However, note that if only one option is `True` and the other is `False`, then `False` will be returned. If this case is reached, the comments in the return data will contain troubleshooting information.

For more information about the `http.query` state, refer to the *HTTP Tutorial*.

```
query_example:
  http.query:
    - name: 'http://example.com/'
    - status: 200

query_example2:
  http.query:
    - name: 'http://example.com/'
    - status:
        - 200
        - 201
    - status_type: list
```

`salt.states.http.`**`wait_for_successful_query`**`(`*name*, *wait_for=300*, ***kwargs*`)`

Like query but, repeat and wait until match/match_type or status is fulfilled. State returns result from last query state in case of success or if no successful query was made within wait_for timeout.

name The name of the query.

wait_for Total time to wait for requests that succeed.

request_interval Optional interval to delay requests by N seconds to reduce the number of requests sent.

> **Note:** All other arguments are passed to the http.query state.

21.19.109 salt.states.icinga2 module

Icinga2 state

New in version 2017.7.0.

depends

- Icinga2 Python module

configuration See `salt.modules.icinga2` for setup instructions.

The icinga2 module is used to execute commands. Its output may be stored in a file or in a grain.

```
command_id:
  icinga2.generate_ticket:
    - name: domain.tld
    - output:  "/tmp/query_id.txt"
```

`salt.states.icinga2.`**`generate_cert`**`(`*name*`)`

Generate an icinga2 certificate and key on the client.

name The domain name for which this certificate and key will be generated

`salt.states.icinga2.`**`generate_ticket`**`(`*name, output=None, grain=None, key=None, overwrite=True*`)`

Generate an icinga2 ticket on the master.

name The domain name for which this ticket will be generated

output grain: output in a grain other: the file to store results None: output to the result comment (default)

grain: grain to store the output (need output=grain)

key: the specified grain will be treated as a dictionary, the result of this state will be stored under the specified key.

overwrite: The file or grain will be overwritten if it already exists (default)

`salt.states.icinga2.`**`node_setup`**`(`*name, master, ticket*`)`

Setup the icinga2 node.

name The domain name for which this certificate will be saved

master Icinga2 master node for which this certificate will be saved

ticket Authentication ticket generated on icinga2 master

`salt.states.icinga2.`**`request_cert`**`(`*name, master, ticket, port='5665'*`)`

Request CA certificate from master icinga2 node.

name The domain name for which this certificate will be saved

master Icinga2 master node for which this certificate will be saved

ticket Authentication ticket generated on icinga2 master

port Icinga2 port, defaults to 5665

`salt.states.icinga2.`**`save_cert`**`(`*name, master*`)`

Save the certificate on master icinga2 node.

name The domain name for which this certificate will be saved

master Icinga2 master node for which this certificate will be saved

21.19.110 salt.states.ifttt

Trigger an event in IFTTT

This state is useful for trigging events in IFTTT.

New in version 2015.8.0.

```
ifttt-event:
  ifttt.trigger_event:
    - event: TestEvent
    - value1: 'This state was executed successfully.'
    - value2: 'Another value we can send.'
    - value3: 'A third value we can send.'
```

The api key can be specified in the master or minion configuration like below: .. code-block:: yaml

ifttt: secret_key: bzMRb-KKIAaNOwKEEw792J7Eb-B3z7muhdhYblJn4V6

`salt.states.ifttt.`**`trigger_event`**`(`*name*, *event*, *value1=None*, *value2=None*, *value3=None*`)`

Trigger an event in IFTTT

```
ifttt-event:
  ifttt.trigger_event:
    - event: TestEvent
    - value1: 'A value that we want to send.'
    - value2: 'A second value that we want to send.'
    - value3: 'A third value that we want to send.'
```

The following parameters are required:

name The unique name for this event.

event The name of the event to trigger in IFTTT.

The following parameters are optional:

value1 One of the values that we can send to IFTT.

value2 One of the values that we can send to IFTT.

value3 One of the values that we can send to IFTT.

21.19.111 salt.states.incron

Management of incron, the inotify cron

The incron state module allows for user incrontabs to be cleanly managed.

Incron declarations require a number of parameters. The parameters needed to be declared: `path`, `mask`, and `cmd`. The `user` whose incrontab is to be edited also needs to be defined.

When making changes to an existing incron job, the `path` declaration is the unique factor, so if an existing cron that looks like this:

```
Watch for modifications in /home/user:
    incron.present:
        - user: root
        - path: /home/user
        - mask:
            - IN_MODIFY
        - cmd: 'echo "$$ $@"'
```

Is changed to this:

```
Watch for modifications and access in /home/user:
    incron.present:
        - user: root
        - path: /home/user
        - mask:
            - IN_MODIFY
            - IN_ACCESS
        - cmd: 'echo "$$ $@"'
```

Then the existing cron will be updated, but if the cron command is changed, then a new cron job will be added to the user's crontab.

New in version 0.17.0.

`salt.states.incron.`**`absent`**`(`*name*, *path*, *mask*, *cmd*, *user='root'*`)`

Verifies that the specified incron job is absent for the specified user; only the name is matched when removing a incron job.

name Unique comment describing the entry
path The path that should be watched
user The name of the user who's crontab needs to be modified, defaults to the root user
mask The mask of events that should be monitored for
cmd The cmd that should be executed

`salt.states.incron.`**`present`**`(`*name*, *path*, *mask*, *cmd*, *user='root'*`)`

Verifies that the specified incron job is present for the specified user. For more advanced information about what exactly can be set in the cron timing parameters, check your incron system's documentation. Most Unix-like systems' incron documentation can be found via the incrontab man page: `man 5 incrontab`.

name Unique comment describing the entry
path The path that should be watched
user The name of the user who's crontab needs to be modified, defaults to the root user
mask The mask of events that should be monitored for
cmd The cmd that should be executed

21.19.112 salt.states.influxdb08_database module

Management of Influxdb 0.8 databases

(compatible with InfluxDB version 0.5-0.8)

New in version 2014.7.0.

`salt.states.influxdb08_database.`**`absent`**`(`*name*, *user=None*, *password=None*, *host=None*, *port=None*`)`

Ensure that the named database is absent

name The name of the database to remove
user The user to connect as (must be able to remove the database)
password The password of the user
host The host to connect to
port The port to connect to

`salt.states.influxdb08_database.`**`present`**`(`*name*, *user=None*, *password=None*, *host=None*, *port=None*`)`

Ensure that the named database is present

name The name of the database to create
user The user to connect as (must be able to remove the database)
password The password of the user
host The host to connect to
port The port to connect to

21.19.113 salt.states.influxdb08_user module

Management of InfluxDB 0.8 users

(compatible with InfluxDB version 0.5-0.8)

New in version 2014.7.0.

`salt.states.influxdb08_user.`**`absent`**(*name, database=None, user=None, password=None, host=None, port=None*)

Ensure that the named cluster admin or database user is absent.

name The name of the user to remove

database The database to remove the user from

user The user to connect as (must be able to remove the user)

password The password of the user

host The host to connect to

port The port to connect to

`salt.states.influxdb08_user.`**`present`**(*name, passwd, database=None, user=None, password=None, host=None, port=None*)

Ensure that the cluster admin or database user is present.

name The name of the user to manage

passwd The password of the user

database The database to create the user in

user The user to connect as (must be able to create the user)

password The password of the user

host The host to connect to

port The port to connect to

21.19.114 salt.states.influxdb_continuous_query module

Management of Influxdb continuous queries

New in version 2017.7.0.

(compatible with InfluxDB version 0.9+)

`salt.states.influxdb_continuous_query.`**`absent`**(*name, database, **client_args*)

Ensure that given continuous query is absent.

name Name of the continuous query to remove.

database Name of the database that the continuous query was defined on.

`salt.states.influxdb_continuous_query.`**`present`**(*name, database, query, resample_time=None, coverage_period=None, **client_args*)

Ensure that given continuous query is present.

name Name of the continuous query to create.

database Database to create continuous query on.

query The query content

resample_time [None] Duration between continuous query resampling.

coverage_period [None] Duration specifying time period per sample.

21.19.115 salt.states.influxdb_database

Management of Influxdb databases

(compatible with InfluxDB version 0.9+)

`salt.states.influxdb_database.`**`absent`**(*name, **client_args*)

Ensure that given database is absent.

name Name of the database to remove.

`salt.states.influxdb_database.`**`present`**(*name*, ***client_args*)

Ensure that given database is present.

name Name of the database to create.

21.19.116 salt.states.influxdb_retention_policy module

Management of Influxdb retention policies

New in version 2017.7.0.

(compatible with InfluxDB version 0.9+)

`salt.states.influxdb_retention_policy.`**`absent`**(*name*, *database*, ***client_args*)

Ensure that given retention policy is absent.

name Name of the retention policy to remove.

database Name of the database that the retention policy was defined on.

`salt.states.influxdb_retention_policy.`**`convert_duration`**(*duration*)

Convert the a duration string into XXhYYmZZs format

duration Duration to convert

Returns: **duration_string** String representation of duration in XXhYYmZZs format

`salt.states.influxdb_retention_policy.`**`present`**(*name*, *database*, *duration='7d'*, *replication=1*, *default=False*, ***client_args*)

Ensure that given retention policy is present.

name Name of the retention policy to create.

database Database to create retention policy on.

21.19.117 salt.states.influxdb_user

Management of InfluxDB users

(compatible with InfluxDB version 0.9+)

`salt.states.influxdb_user.`**`absent`**(*name*, ***client_args*)

Ensure that given user is absent.

name The name of the user to manage

`salt.states.influxdb_user.`**`present`**(*name*, *passwd*, *admin=False*, *grants=None*, ***client_args*)

Ensure that given user is present.

name Name of the user to manage

passwd Password of the user

admin [False] Whether the user should have cluster administration privileges or not.

grants Optional - Dict of database:privilege items associated with the user. Example:

grants: foo_db: read bar_db: all

Example:

```
example user present in influxdb:
  influxdb_user.present:
    - name: example
    - passwd: somepassword
    - admin: False
    - grants:
        foo_db: read
        bar_db: all
```

21.19.118 salt.states.infoblox_a

Infoblox A record managment.

functions accept api_opts:

> api_verifyssl: verify SSL [default to True or pillar value] api_url: server to connect to [default to pillar value] api_username: [default to pillar value] api_password: [default to pillar value]

`salt.states.infoblox_a.`**`absent`**(*name=None*, *ipv4addr=None*, ***api_opts*)

Ensure infoblox A record is removed.

State example:

```yaml
infoblox_a.absent:
    - name: example-ha-0.domain.com

infoblox_a.absent:
    - name:
    - ipv4addr: 127.0.23.23
```

`salt.states.infoblox_a.`**`present`**(*name=None*, *ipv4addr=None*, *data=None*, *ensure_data=True*, ***api_opts*)

Ensure infoblox A record.

When you wish to update a hostname ensure *name* is set to the hostname of the current record. You can give a new name in the *data.name*.

State example:

```yaml
infoblox_a.present:
    - name: example-ha-0.domain.com
    - data:
        name: example-ha-0.domain.com
        ipv4addr: 123.0.31.2
        view: Internal
```

21.19.119 salt.states.infoblox_cname

Infoblox CNAME managment.

functions accept api_opts:

> api_verifyssl: verify SSL [default to True or pillar value] api_url: server to connect to [default to pillar value] api_username: [default to pillar value] api_password: [default to pillar value]

`salt.states.infoblox_cname.`**`absent`**(*name=None*, *canonical=None*, ***api_opts*)

Ensure the CNAME with the given name or canonical name is removed

`salt.states.infoblox_cname.`**`present`**(*name=None*, *data=None*, *ensure_data=True*, ***api_opts*)

Ensure the CNAME with the given data is present.

name CNAME of record

data raw CNAME api data see: https://INFOBLOX/wapidoc

State example:

```yaml
infoblox_cname.present:
    - name: example-ha-0.domain.com
    - data:
        name: example-ha-0.domain.com
```

(continues on next page)

(continued from previous page)

```
        canonical: example.domain.com
        zone: example.com
        view: Internal
        comment: Example comment

infoblox_cname.present:
    - name: example-ha-0.domain.com
    - data:
        name: example-ha-0.domain.com
        canonical: example.domain.com
        zone: example.com
        view: Internal
        comment: Example comment
    - api_url: https://INFOBLOX/wapi/v1.2.1
    - api_username: username
    - api_password: passwd
```

21.19.120 salt.states.infoblox_host_record

Infoblox host record managment.

functions accept api_opts:

> api_verifyssl: verify SSL [default to True or pillar value] api_url: server to connect to [default to pillar value] api_username: [default to pillar value] api_password: [default to pillar value]

`salt.states.infoblox_host_record.`**`absent`**`(`*name=None, ipv4addr=None, mac=None, **api_opts*`)`

Ensure the host with the given Name ipv4addr or mac is removed.

State example:

```
infoblox_host_record.absent:
    - name: hostname.of.record.to.remove

infoblox_host_record.absent:
    - name:
    - ipv4addr: 192.168.0.1

infoblox_host_record.absent:
    - name:
    - mac: 12:02:12:31:23:43
```

`salt.states.infoblox_host_record.`**`present`**`(`*name=None, data=None, ensure_data=True, **api_opts*`)`

This will ensure that a host with the provided name exists. This will try to ensure that the state of the host matches the given data If the host is not found then one will be created.

When trying to update a hostname ensure *name* is set to the hostname of the current record. You can give a new name in the *data.name*.

Avoid race conditions, use func:nextavailableip:

- func:nextavailableip:network/ZG54dfgsrDFEFfsfsLzA:10.0.0.0/8/default
- func:nextavailableip:10.0.0.0/8
- func:nextavailableip:10.0.0.0/8,externalconfigure_for_dns
- func:nextavailableip:10.0.0.3-10.0.0.10

State Example:

```
# this would update `original_hostname.example.ca` to changed `data`.
infoblox_host_record.present:
    - name: original_hostname.example.ca
    - data: {'namhostname.example.cae': 'hostname.example.ca',
        'aliases': ['hostname.math.example.ca'],
        'extattrs': [{'Business Contact': {'value': 'EXAMPLE@example.ca'}}],
        'ipv4addrs': [{'configure_for_dhcp': True,
            'ipv4addr': 'func:nextavailableip:129.97.139.0/24',
            'mac': '00:50:56:84:6e:ae'}],
        'ipv6addrs': [], }
```

21.19.121 salt.states.infoblox_range

Infoblox host record managment.

functions accept api_opts:

> api_verifyssl: verify SSL [default to True or pillar value] api_url: server to connect to [default to pillar value] api_username: [default to pillar value] api_password: [default to pillar value]

`salt.states.infoblox_range.`**`absent`**(*name=None*, *start_addr=None*, *end_addr=None*, *data=None*, ***api_opts*)

Ensure the range is removed

Supplying the end of the range is optional.

State example:

```
infoblox_range.absent:
    - name: 'vlan10'

infoblox_range.absent:
    - name:
    - start_addr: 127.0.1.20
```

`salt.states.infoblox_range.`**`present`**(*name=None*, *start_addr=None*, *end_addr=None*, *data=None*, ***api_opts*)

Ensure range record is present.
infoblox_range.present: start_addr: '129.97.150.160', end_addr: '129.97.150.170',
Verbose state example:

```
infoblox_range.present:
    data: {
        'always_update_dns': False,
        'authority': False,
        'comment': 'range of IP addresses used for salt.. was used for ghost
↪images deployment',
        'ddns_generate_hostname': True,
        'deny_all_clients': False,
        'deny_bootp': False,
        'disable': False,
        'email_list': [],
        'enable_ddns': False,
        'enable_dhcp_thresholds': False,
        'enable_email_warnings': False,
        'enable_ifmap_publishing': False,
        'enable_snmp_warnings': False,
```

(continues on next page)

(continued from previous page)

```
        'end_addr': '129.97.150.169',
        'exclude': [],
        'extattrs': {},
        'fingerprint_filter_rules': [],
        'high_water_mark': 95,
        'high_water_mark_reset': 85,
        'ignore_dhcp_option_list_request': False,
        'lease_scavenge_time': -1,
        'logic_filter_rules': [],
        'low_water_mark': 0,
        'low_water_mark_reset': 10,
        'mac_filter_rules': [],
        'member': {'_struct': 'dhcpmember',
                'ipv4addr': '129.97.128.9',
                'name': 'cn-dhcp-mc.example.ca'},
        'ms_options': [],
        'nac_filter_rules': [],
        'name': 'ghost-range',
        'network': '129.97.150.0/24',
        'network_view': 'default',
        'option_filter_rules': [],
        'options': [{'name': 'dhcp-lease-time',
                    'num': 51,
                    'use_option': False,
                    'value': '43200',
                    'vendor_class': 'DHCP'}],
        'recycle_leases': True,
        'relay_agent_filter_rules': [],
        'server_association_type': 'MEMBER',
        'start_addr': '129.97.150.160',
        'update_dns_on_lease_renewal': False,
        'use_authority': False,
        'use_bootfile': False,
        'use_bootserver': False,
        'use_ddns_domainname': False,
        'use_ddns_generate_hostname': True,
        'use_deny_bootp': False,
        'use_email_list': False,
        'use_enable_ddns': False,
        'use_enable_dhcp_thresholds': False,
        'use_enable_ifmap_publishing': False,
        'use_ignore_dhcp_option_list_request': False,
        'use_known_clients': False,
        'use_lease_scavenge_time': False,
        'use_nextserver': False,
        'use_options': False,
        'use_recycle_leases': False,
        'use_unknown_clients': False,
        'use_update_dns_on_lease_renewal': False
    }
```

21.19.122 salt.states.ini_manage

Manage ini files

maintainer <akilesh1597@gmail.com>

maturity new

depends re

platform all

`salt.states.ini_manage.`**`options_absent`**(*name*, *sections=None*, *separator='='*)

```
/home/saltminion/api-paste.ini:
  ini.options_absent:
    - separator: '='
    - sections:
        test:
          - testkey
          - secondoption
        test1:
          - testkey1
```

options present in file and not specified in sections dict will be untouched

changes dict will contain the list of changes made

`salt.states.ini_manage.`**`options_present`**(*name*, *sections=None*, *separator='='*, *strict=False*)

```
/home/saltminion/api-paste.ini:
  ini.options_present:
    - separator: '='
    - strict: True
    - sections:
        test:
          testkey: 'testval'
          secondoption: 'secondvalue'
        test1:
          testkey1: 'testval121'
```

options present in file and not specified in sections dict will be untouched, unless *strict: True* flag is used

changes dict will contain the list of changes made

`salt.states.ini_manage.`**`sections_absent`**(*name*, *sections=None*, *separator='='*)

```
/home/saltminion/api-paste.ini:
  ini.sections_absent:
    - separator: '='
    - sections:
        - test
        - test1
```

options present in file and not specified in sections will be deleted changes dict will contain the sections that changed

`salt.states.ini_manage.`**`sections_present`**(*name*, *sections=None*, *separator='='*)

```
/home/saltminion/api-paste.ini:
  ini.sections_present:
    - separator: '='
    - sections:
        - section_one
        - section_two
```

This will only create empty sections. To also create options, use options_present state

options present in file and not specified in sections will be deleted changes dict will contain the sections that changed

21.19.123 salt.states.ipmi

Manage IPMI devices over LAN

The following configuration defaults can be defined in the minion, master config or pillar:

```
ipmi.config:
    api_host: 127.0.0.1
    api_user: admin
    api_pass: apassword
    api_port: 623
    api_kg: None
```

Every call can override the config defaults:

```
ensure myipmi system is set to network boot:
    ipmi.boot_device:
        - name: network
        - api_host: myipmi.hostname.com
        - api_user: root
        - api_pass: apassword
        - api_kg: None

ensure myipmi system is powered on:
    ipmi.power:
        - name: boot
        - api_host: myipmi.hostname.com
        - api_user: root
        - api_pass: apassword
```

`salt.states.ipmi.`**`boot_device`**(*name='default'*, ***kwargs*)

Request power state change

name = `default`

- network – Request network boot
- hd – Boot from hard drive
- safe – Boot from hard drive, requesting 'safe mode'
- optical – boot from CD/DVD/BD drive
- setup – Boot into setup utility
- default – remove any IPMI directed boot device request

kwargs

- api_host=localhost
- api_user=admin
- api_pass=

- api_port=623
- api_kg=None

`salt.states.ipmi.`**`power`**`(`*name='power_on', wait=300, **kwargs*`)`

Request power state change

name

Ensure power state one of:

- power_on – system turn on
- power_off – system turn off (without waiting for OS)
- shutdown – request OS proper shutdown
- reset – reset (without waiting for OS)
- boot – If system is off, then 'on', else 'reset'

wait wait X seconds for the job to complete before forcing. (defaults to 300 seconds)

kwargs

- api_host=localhost
- api_user=admin
- api_pass=
- api_port=623
- api_kg=None

`salt.states.ipmi.`**`user_absent`**`(`*name, channel=14, **kwargs*`)`

Remove user Delete all user (uid) records having the matching name.

name string name of user to delete

channel channel to remove user access from defaults to 14 for auto.

kwargs

- api_host=localhost
- api_user=admin
- api_pass=
- api_port=623
- api_kg=None

`salt.states.ipmi.`**`user_present`**`(`*name, uid, password, channel=14, callback=False, link_auth=True, ipmi_msg=True, privilege_level='administrator', **kwargs*`)`

Ensure IPMI user and user privileges.

name name of user (limit 16 bytes)

uid user id number (1 to 7)

password user password (limit 16 bytes)

channel ipmi channel defaults to 14 for auto

callback User Restricted to Callback

False = User Privilege Limit is determined by the User Privilege Limit parameter privilege_level, for both callback and non-callback connections.

True = User Privilege Limit is determined by the privilege_level parameter for callback connections, but is restricted to Callback level for non-callback connections. Thus, a user can only initiate a Callback when they 'call in' to the BMC, but once the callback connection has been made, the user could potentially establish a session as an Operator.

link_auth User Link authentication True/False user name and password information will be used for link authentication, e.g. PPP CHAP) for the given channel. Link authentication itself is a global setting for the channel and is enabled/disabled via the serial/modem configuration parameters.

ipmi_msg User IPMI Messaging True/False user name and password information will be used for IPMI Messaging. In this case, 'IPMI Messaging' refers to the ability to execute generic IPMI commands that are not associated with a particular payload type. For example, if IPMI Messaging is disabled for a user, but that user is enabled for activating the SOL payload type, then IPMI commands associated with SOL and session management, such as Get SOL Configuration Parameters and Close Session are available, but generic IPMI commands such as Get SEL Time are unavailable.) ipmi_msg

privilege_level

- callback

- user
- operator
- administrator
- proprietary
- no_access

kwargs

- api_host=localhost
- api_user=admin
- api_pass=
- api_port=623
- api_kg=None

21.19.124 salt.states.ipset

Management of ipsets

This is an ipset-specific module designed to manage IPSets for use in IPTables Firewalls.

```
setname:
  ipset.set_present:
    - set_type: bitmap:ip
    - range: 192.168.0.0/16
    - comment: True

setname:
  ipset.set_absent:
    - set_type: bitmap:ip
    - range: 192.168.0.0/16
    - comment: True

setname_entries:
  ipset.present:
    - set_name: setname
    - entry: 192.168.0.3
    - comment: Hello
    - require:
        - ipset: baz

setname_entries:
  ipset.present:
    - set_name: setname
    - entry:
        - 192.168.0.3
        - 192.168.1.3
    - comment: Hello
    - require:
        - ipset: baz

setname_entries:
  ipset.absent:
    - set_name: setname
    - entry:
        - 192.168.0.3
        - 192.168.1.3
    - comment: Hello
```

(continues on next page)

(continued from previous page)

```
    - require:
        - ipset: baz

setname:
  ipset.flush:
```

`salt.states.ipset.`**`absent`**(*name*, *entry=None*, *entries=None*, *family='ipv4'*, ***kwargs*)
New in version 2014.7.0.

Remove a entry or entries from a chain

name A user-defined name to call this entry by in another part of a state or formula. This should not be an actual entry.

family Network family, ipv4 or ipv6.

`salt.states.ipset.`**`flush`**(*name*, *family='ipv4'*, ***kwargs*)
New in version 2014.7.0.

Flush current ipset set

family Networking family, either ipv4 or ipv6

`salt.states.ipset.`**`present`**(*name*, *entry=None*, *family='ipv4'*, ***kwargs*)
New in version 2014.7.0.

Append a entry to a set

name A user-defined name to call this entry by in another part of a state or formula. This should not be an actual entry.

entry A single entry to add to a set or a list of entries to add to a set

family Network family, ipv4 or ipv6.

`salt.states.ipset.`**`set_absent`**(*name*, *family='ipv4'*, ***kwargs*)
New in version 2014.7.0.

Verify the set is absent.

family Networking family, either ipv4 or ipv6

`salt.states.ipset.`**`set_present`**(*name*, *set_type*, *family='ipv4'*, ***kwargs*)
New in version 2014.7.0.

Verify the set exists.

name A user-defined set name.

set_type The type for the set.

family Networking family, either ipv4 or ipv6

21.19.125 salt.states.iptables

Management of iptables

This is an iptables-specific module designed to manage Linux firewalls. It is expected that this state module, and other system-specific firewall states, may at some point be deprecated in favor of a more generic `firewall` state.

```
httpd:
  iptables.append:
    - table: filter
    - chain: INPUT
    - jump: ACCEPT
    - match: state
```

(continues on next page)

(continued from previous page)

```
    - connstate: NEW
    - dport: 80
    - protocol: tcp
    - sport: 1025:65535
    - save: True

httpd:
  iptables.append:
    - table: filter
    - chain: INPUT
    - jump: ACCEPT
    - match:
        - state
        - comment
    - comment: "Allow HTTP"
    - connstate: NEW
    - dport: 80
    - protocol: tcp
    - sport: 1025:65535
    - save: True

httpd:
  iptables.append:
    - table: filter
    - chain: INPUT
    - jump: ACCEPT
    - match:
        - state
        - comment
    - comment: "Allow HTTP"
    - connstate: NEW
    - source: '127.0.0.1'
    - dport: 80
    - protocol: tcp
    - sport: 1025:65535
    - save: True

.. Invert Rule
httpd:
  iptables.append:
    - table: filter
    - chain: INPUT
    - jump: ACCEPT
    - match:
        - state
        - comment
    - comment: "Allow HTTP"
    - connstate: NEW
    - source: '! 127.0.0.1'
    - dport: 80
    - protocol: tcp
    - sport: 1025:65535
    - save: True

httpd:
  iptables.append:
```

(continues on next page)

(continued from previous page)

```
    - table: filter
    - chain: INPUT
    - jump: ACCEPT
    - match:
        - state
        - comment
    - comment: "Allow HTTP"
    - connstate: NEW
    - source: 'not 127.0.0.1'
    - dport: 80
    - protocol: tcp
    - sport: 1025:65535
    - save: True

httpd:
  iptables.append:
    - table: filter
    - family: ipv6
    - chain: INPUT
    - jump: ACCEPT
    - match: state
    - connstate: NEW
    - dport: 80
    - protocol: tcp
    - sport: 1025:65535
    - save: True

httpd:
  iptables.append:
    - table: filter
    - family: ipv4
    - chain: INPUT
    - jump: ACCEPT
    - match: state
    - connstate: NEW
    - dports:
        - 80
        - 443
    - protocol: tcp
    - sport: 1025:65535
    - save: True

httpd:
  iptables.insert:
    - position: 1
    - table: filter
    - chain: INPUT
    - jump: ACCEPT
    - match: state
    - connstate: NEW
    - dport: 80
    - protocol: tcp
    - sport: 1025:65535
    - save: True

httpd:
```

(continues on next page)

(continued from previous page)

```
  iptables.insert:
    - position: 1
    - table: filter
    - family: ipv6
    - chain: INPUT
    - jump: ACCEPT
    - match: state
    - connstate: NEW
    - dport: 80
    - protocol: tcp
    - sport: 1025:65535
    - save: True

httpd:
  iptables.delete:
    - table: filter
    - chain: INPUT
    - jump: ACCEPT
    - match: state
    - connstate: NEW
    - dport: 80
    - protocol: tcp
    - sport: 1025:65535
    - save: True

httpd:
  iptables.delete:
    - position: 1
    - table: filter
    - chain: INPUT
    - jump: ACCEPT
    - match: state
    - connstate: NEW
    - dport: 80
    - protocol: tcp
    - sport: 1025:65535
    - save: True

httpd:
  iptables.delete:
    - table: filter
    - family: ipv6
    - chain: INPUT
    - jump: ACCEPT
    - match: state
    - connstate: NEW
    - dport: 80
    - protocol: tcp
    - sport: 1025:65535
    - save: True

default to accept:
  iptables.set_policy:
    - chain: INPUT
    - policy: ACCEPT
```

Note: Whereas iptables will accept `-p`, `--proto[c[o[l]]]` as synonyms of `--protocol`, if `--proto` appears in an iptables command after the appearance of `-m policy`, it is interpreted as the `--proto` option of the policy extension (see the iptables-extensions(8) man page).

Example rules for IPSec policy:

```
accept_esp_in:
  iptables.append:
    - table: filter
    - chain: INPUT
    - jump: ACCEPT
    - source: 10.20.0.0/24
    - destination: 10.10.0.0/24
    - in-interface: eth0
    - match: policy
    - dir: in
    - pol: ipsec
    - reqid: 1
    - proto: esp
accept_esp_forward_in:
  iptables.append:
    - use:
      - iptables: accept_esp_in
    - chain: FORWARD

accept_esp_out:
  iptables.append:
    - table: filter
    - chain: OUTPUT
    - jump: ACCEPT
    - source: 10.10.0.0/24
    - destination: 10.20.0.0/24
    - out-interface: eth0
    - match: policy
    - dir: out
    - pol: ipsec
    - reqid: 1
    - proto: esp
accept_esp_forward_out:
  iptables.append:
    - use:
      - iptables: accept_esp_out
    - chain: FORWARD
```

Note: Various functions of the `iptables` module use the `--check` option. If the version of `iptables` on the target system does not include this option, an alternate version of this check will be performed using the output of iptables-save. This may have unintended consequences on legacy releases of `iptables`.

`salt.states.iptables.`**`append`**(*name*, *table='filter'*, *family='ipv4'*, ***kwargs*)

New in version 0.17.0.

Add a rule to the end of the specified chain.

name A user-defined name to call this rule by in another part of a state or formula. This should not be an actual rule.

table The table that owns the chain which should be modified

family Network family, ipv4 or ipv6.

All other arguments are passed in with the same name as the long option that would normally be used for iptables, with one exception: `--state` is specified as *connstate* instead of *state* (not to be confused with *ctstate*).

Jump options that doesn't take arguments should be passed in with an empty string.

`salt.states.iptables.`**`chain_absent`**(*name*, *table='filter'*, *family='ipv4'*)

New in version 2014.1.0.

Verify the chain is absent.

table The table to remove the chain from

family Networking family, either ipv4 or ipv6

`salt.states.iptables.`**`chain_present`**(*name*, *table='filter'*, *family='ipv4'*)

New in version 2014.1.0.

Verify the chain is exist.

name A user-defined chain name.

table The table to own the chain.

family Networking family, either ipv4 or ipv6

`salt.states.iptables.`**`delete`**(*name*, *table='filter'*, *family='ipv4'*, ***kwargs*)

New in version 2014.1.0.

Delete a rule to a chain

name A user-defined name to call this rule by in another part of a state or formula. This should not be an actual rule.

table The table that owns the chain that should be modified

family Networking family, either ipv4 or ipv6

All other arguments are passed in with the same name as the long option that would normally be used for iptables, with one exception: `--state` is specified as *connstate* instead of *state* (not to be confused with *ctstate*).

Jump options that doesn't take arguments should be passed in with an empty string.

`salt.states.iptables.`**`flush`**(*name*, *table='filter'*, *family='ipv4'*, ***kwargs*)

New in version 2014.1.0.

Flush current iptables state

table The table that owns the chain that should be modified

family Networking family, either ipv4 or ipv6

`salt.states.iptables.`**`insert`**(*name*, *table='filter'*, *family='ipv4'*, ***kwargs*)

New in version 2014.1.0.

Insert a rule into a chain

name A user-defined name to call this rule by in another part of a state or formula. This should not be an actual rule.

table The table that owns the chain that should be modified

family Networking family, either ipv4 or ipv6

position The numerical representation of where the rule should be inserted into the chain. Note that `-1` is not a supported position value.

All other arguments are passed in with the same name as the long option that would normally be used for iptables, with one exception: `--state` is specified as *connstate* instead of *state* (not to be confused with *ctstate*).

Jump options that doesn't take arguments should be passed in with an empty string.

`salt.states.iptables.`**`mod_aggregate`**`(`*low*, *chunks*, *running*`)`

The mod_aggregate function which looks up all rules in the available low chunks and merges them into a single rules ref in the present low data

`salt.states.iptables.`**`set_policy`**`(`*name*, *table='filter'*, *family='ipv4'*, ***kwargs*`)`

New in version 2014.1.0.

Sets the default policy for iptables firewall tables

table The table that owns the chain that should be modified

family Networking family, either ipv4 or ipv6

policy The requested table policy

21.19.126 salt.states.jboss7

Manage JBoss 7 Application Server via CLI interface

New in version 2015.5.0.

This state uses the jboss-cli.sh script from a JBoss or Wildfly installation and parses its output to determine the execution result.

In order to run each state, a jboss_config dictionary with the following properties must be passed:

```
jboss:
   cli_path: '/opt/jboss/jboss-7.0/bin/jboss-cli.sh'
   controller: 10.11.12.13:9999
   cli_user: 'jbossadm'
   cli_password: 'jbossadm'
```

If the controller doesn't require a password, then the cli_user and cli_password parameters are optional.

Since same dictionary with configuration will be used in all the states, it may be more convenient to move JBoss configuration and other properties to the pillar.

Example of application deployment from local filesystem:

```
application_deployed:
  jboss7.deployed:
    - salt_source:
        target_file: '/tmp/webapp.war'
    - jboss_config: {{ pillar['jboss'] }}
```

For the sake of brevity, examples for each state assume that jboss_config is contained in the pillar.

`salt.states.jboss7.`**`bindings_exist`**`(`*name*, *jboss_config*, *bindings*, *profile=None*`)`

Ensures that given JNDI binding are present on the server. If a binding doesn't exist on the server it will be created. If it already exists its value will be changed.

jboss_config: Dict with connection properties (see state description)

bindings: Dict with bindings to set.

profile: The profile name (domain mode only)

Example:

```
jndi_entries_created:
  jboss7.bindings_exist:
   - bindings:
      'java:global/sampleapp/environment': 'DEV'
      'java:global/sampleapp/configurationFile': '/var/opt/sampleapp/config.
↪properties'
   - jboss_config: {{ pillar['jboss'] }}
```

`salt.states.jboss7.`**`datasource_exists`**`(`*name, jboss_config, datasource_properties, recreate=False, profile=None*`)`

Ensures that a datasource with given properties exist on the jboss instance. If datasource doesn't exist, it is created, otherwise only the properties that are different will be updated.

name Datasource property name

jboss_config Dict with connection properties (see state description)

datasource_properties Dict with datasource properties

recreate [False] If set to True and datasource exists it will be removed and created again. However, if there are deployments that depend on the datasource, it will not me possible to remove it.

profile [None] The profile name for this datasource (domain mode only)

Example:

```
sampleDS:
  jboss7.datasource_exists:
   - recreate: False
   - datasource_properties:
       driver-name: mysql
       connection-url: 'jdbc:mysql://localhost:3306/sampleDatabase'
       jndi-name: 'java:jboss/datasources/sampleDS'
       user-name: sampleuser
       password: secret
       min-pool-size: 3
       use-java-context: True
   - jboss_config: {{ pillar['jboss'] }}
   - profile: full-ha
```

`salt.states.jboss7.`**`deployed`**`(`*name, jboss_config, salt_source=None*`)`

Ensures that the given application is deployed on server.

jboss_config: Dict with connection properties (see state description)

salt_source:

How to find the artifact to be deployed.

- **target_file**: Where to look in the minion's file system for the artifact to be deployed (e.g. '/tmp/application-web-0.39.war'). When source is specified, also specifies where to save the retrieved file.
- **source**: (optional) File on salt master (e.g. salt://application-web-0.39.war). If absent, no files will be retrieved and the artifact in target_file will be used for the deployment.
- **undeploy**: (optional) Regular expression to match against existing deployments. When present, if there is a deployment that matches the regular expression, it will be undeployed before the new artifact is deployed.
- **undeploy_force**: (optional) If True, the artifact will be undeployed although it has not changed.

Examples:

Deployment of a file from minion's local file system:

```
application_deployed:
  jboss7.deployed:
    - salt_source:
        target_file: '/tmp/webapp.war'
    - jboss_config: {{ pillar['jboss'] }}
```

It is assumed that /tmp/webapp.war was made available by some other means. No applications will be undeployed; if an existing deployment that shares that name exists, then it will be replaced with the updated version.

Deployment of a file from the Salt master's file system:

```
application_deployed:
  jboss7.deployed:
   - salt_source:
        source: salt://application-web-0.39.war
        target_file: '/tmp/application-web-0.39.war'
        undeploy: 'application-web-.*'
   - jboss_config: {{ pillar['jboss'] }}
```

Here, application-web-0.39.war file is downloaded from Salt file system to /tmp/application-web-0.39.war file on minion. Existing deployments are checked if any of them matches 'application-web-.*' regular expression, and if so then it is undeployed before deploying the application. This is useful to automate deployment of new application versions.

If the source parameter of salt_source is specified, it can use any protocol that the file states use. This includes not only downloading from the master but also HTTP, HTTPS, FTP, Amazon S3, and OpenStack Swift.

`salt.states.jboss7.`**`reloaded`**`(`*name*, *jboss_config*, *timeout=60*, *interval=5*`)`

Reloads configuration of jboss server.

jboss_config: Dict with connection properties (see state description)

timeout: Time to wait until jboss is back in running state. Default timeout is 60s.

interval: Interval between state checks. Default interval is 5s. Decreasing the interval may slightly decrease waiting time but be aware that every status check is a call to jboss-cli which is a java process. If interval is smaller than process cleanup time it may easily lead to excessive resource consumption.

This step performs the following operations:

- Ensures that server is in running or reload-required state (by reading server-state attribute)
- Reloads configuration
- Waits for server to reload and be in running state

Example:

```
configuration_reloaded:
   jboss7.reloaded:
    - jboss_config: {{ pillar['jboss'] }}
```

21.19.127 salt.states.jenkins module

Management of Jenkins

New in version 2016.3.0.

`salt.states.jenkins.`**`absent`**`(`*name*, ***kwargs*`)`

Ensure the job is absent from the Jenkins configured jobs

name The name of the Jenkins job to remove

`salt.states.jenkins.`**`present`**`(`*name*, *config=None*, ***kwargs*`)`

Ensure the job is present in the Jenkins configured jobs

name The unique name for the Jenkins job

config The Salt URL for the file to use for configuring the job

21.19.128 salt.states.junos module

State modules to interact with Junos devices.

maturity new

dependencies junos-eznc, jxmlease

Note: Those who wish to use junos-eznc (PyEZ) version >= 2.1.0, must use the latest salt code from github until the next release.

Refer to `junos` for information on connecting to junos proxy.

`salt.states.junos.`**`cli`**`(`*name*, ***kwargs*`)`

Executes the CLI commands and reutrns the text output.

```
show version:
  junos:
    - cli
    - format: xml
```

Parameters

- **`Required`** –
  - command: The command that need to be executed on Junos CLI. (default = None)
- **`Optional`** –
  - kwargs: **Keyworded arguments which can be provided like-**
    - format: Format in which to get the CLI output. (text or xml, default = 'text')
    - timeout: Set NETCONF RPC timeout. Can be used for commands which take a while to execute. (default = 30 seconds)
    - dest: The destination file where the CLI output can be stored. (default = None)

`salt.states.junos.`**`commit`**`(`*name*, ***kwargs*`)`

Commits the changes loaded into the candidate configuration.

```
commit the changes:
  junos:
    - commit
    - confirm: 10
```

Parameters **`Optional`** –

- kwargs: **Keyworded arguments which can be provided like-**
  - timeout: Set NETCONF RPC timeout. Can be used for commands which take a while to execute. (default = 30 seconds)
  - comment: Provide a comment to the commit. (default = None)
  - confirm: Provide time in minutes for commit confirmation. If this option is specified, the commit will be rollbacked in the given time unless the commit is confirmed.
  - sync: On dual control plane systems, requests that the candidate configuration on one control plane be copied to the other control plane,checked for correct syntax, and committed on both Routing Engines. (default = False)
  - force_sync: On dual control plane systems, force the candidate configuration on one control plane to be copied to the other control plane.
  - full: When set to True requires all the daemons to check and evaluate the new configuration.
  - detail: When true return commit detail.

`salt.states.junos.`**`commit_check`**`(`*name*`)`

Perform a commit check on the configuration.

```
perform commit check:
  junos.commit_check
```

`salt.states.junos.`**`diff`**(*name*, *d_id*)

Gets the difference between the candidate and the current configuration.

```
get the diff:
  junos:
    - diff
    - id: 10
```

Parameters **`Optional`** –
- id: The rollback id value [0-49]. (default = 0)

`salt.states.junos.`**`file_copy`**(*name*, *dest=None*, ***kwargs*)

Copies the file from the local device to the junos device.

```
/home/m2/info.txt:
  junos:
    - file_copy
    - dest: info_copy.txt
```

Parameters **`Required`** –
- src: The sorce path where the file is kept.
- dest: The destination path where the file will be copied.

`salt.states.junos.`**`install_config`**(*name*, ***kwargs*)

Loads and commits the configuration provided.

```
Install the mentioned config:
  junos:
    - install_config
    - path: salt//configs/interface.set
    - timeout: 100
    - diffs_file: 'var/log/diff'
```

```
Install the mentioned config:
  junos:
    - install_config
    - template_path: salt//configs/interface.set
    - timeout: 100
    - template_vars:
        interface_name: lo0
        description: Creating interface via SaltStack.
```

name Path where the configuration/template file is present. If the file has a `*.conf` extension, the content is treated as text format. If the file has a `*.xml` extension, the content is treated as XML format. If the file has a `*.set` extension, the content is treated as Junos OS `set` commands

template_vars The dictionary of data for the jinja variables present in the jinja template

timeout [30] Set NETCONF RPC timeout. Can be used for commands which take a while to execute.

overwrite [False]

Set to `True` if you want this file is to completely replace the configuration file.

replace [False] Specify whether the configuration file uses "replace:" statements. Only those statements under the 'replace' tag will be changed.

comment Provide a comment to the commit. (default = None)

confirm Provide time in minutes for commit confirmation. If this option is specified, the commit will be rolled back in the given time unless the commit is confirmed.

diffs_file Path to the file where the diff (difference in old configuration and the committed configuration) will be stored.

> **Note:** The file will be stored on the proxy minion. To push the files to the master use *`cp.push`*.

`salt.states.junos.`**`install_os`**`(`*name*, ***kwargs*`)`

Installs the given image on the device. After the installation is complete the device is rebooted, if reboot=True is given as a keyworded argument.

```
salt://images/junos_image.tgz:
  junos:
    - install_os
    - timeout: 100
    - reboot: True
```

Parameters

- **`Required`** –
  - path: Path where the image file is present on the pro xy minion.
- **`Optional`** –
  - **kwargs: keyworded arguments to be given such as timeout, reboot etc**
    - timeout: Set NETCONF RPC timeout. Can be used to RPCs which take a while to execute. (default = 30 seconds)
    - reboot: Whether to reboot after installation (default = False)
    - no_copy: When True the software package will not be SCP'd to the device. (default = False)

`salt.states.junos.`**`load`**`(`*name*, ***kwargs*`)`

Loads the configuration provided onto the junos device.

```
Install the mentioned config:
  junos:
    - load
    - path: salt//configs/interface.set
```

```
Install the mentioned config:
  junos:
    - load
    - template_path: salt//configs/interface.set
    - template_vars:
        interface_name: lo0
        description: Creating interface via SaltStack.
```

name Path where the configuration/template file is present. If the file has a `*.conf` extension, the content is treated as text format. If the file has a `*.xml` extension, the content is treated as XML format. If the file has a `*.set` extension, the content is treated as Junos OS `set` commands.

overwrite [False] Set to `True` if you want this file is to completely replace the configuration file.

replace [False] Specify whether the configuration file uses "replace:" statements. Only those statements under the 'replace' tag will be changed.

format: Determines the format of the contents.

update [False] Compare a complete loaded configuration against the candidate configuration. For each hierarchy level or configuration object that is different in the two configurations, the version in the loaded configuration replaces the version in the candidate configuration. When the configuration is later committed, only system processes that are affected by the changed configuration elements parse the new configuration. This action is supported from PyEZ 2.1 (default = False)

template_vars Variables to be passed into the template processing engine in addition to those present in __pillar__, __opts__, __grains__, etc. You may reference these variables in your template like so: {{ template_vars["var_name"] }}

`salt.states.junos.`**`lock`**`(`*name*`)`

Attempts an exclusive lock on the candidate configuration. This is a non-blocking call.

Note: Any user who wishes to use lock, must necessarily unlock the configuration too. Ensure `unlock` is called in the same orchestration run in which the lock is called.

```
lock the config:
  junos.lock
```

`salt.states.junos.`**`resultdecorator`**`(`*function*`)`

`salt.states.junos.`**`rollback`**`(`*name, id, **kwargs*`)`

Rollbacks the committed changes.

```
rollback the changes:
  junos:
    - rollback
    - id: 5
```

Parameters **`Optional`** –

- id: The rollback id value [0-49]. (default = 0)
- **kwargs: Keyworded arguments which can be provided like-**
  - timeout: Set NETCONF RPC timeout. Can be used for commands which take a while to execute. (default = 30 seconds)
  - comment: Provide a comment to the commit. (default = None)
  - confirm: Provide time in minutes for commit confirmation. If this option is specified, the commit will be rollbacked in the given time unless the commit is confirmed.
  - diffs_file: Path to the file where any diffs will be written. (default = None)

`salt.states.junos.`**`rpc`**`(`*name, dest=None, format='xml', args=None, **kwargs*`)`

Executes the given rpc. The returned data can be stored in a file by specifying the destination path with dest as an argument

```
get-interface-information:
    junos:
      - rpc
      - dest: /home/user/rpc.log
      - interface_name: lo0
```

Parameters

- **`Required`** –
  - cmd: The rpc to be executed. (default = None)
- **`Optional`** –
  - dest: Destination file where the rpc output is stored. (default = None) Note that the file will be stored on the proxy minion. To push the files to the master use the salt's following execution module: `cp.push`
  - format: The format in which the rpc reply must be stored in file specified in the dest (used only when dest is specified) (default = xml)
  - **kwargs: keyworded arguments taken by rpc call like-**
    - timeout: Set NETCONF RPC timeout. Can be used for commands which take a while to execute. (default= 30 seconds)

      - filter: Only to be used with 'get-config' rpc to get specific configuration.
      - terse: Amount of information you want.
      - interface_name: Name of the interface whose information you want.

salt.states.junos.**set_hostname**(*name*, ***kwargs*)

Changes the hostname of the device.

```
device_name:
  junos:
    - set_hostname
    - comment: "Host-name set via saltstack."
```

Parameters

- **Required** –
  - hostname: The name to be set. (default = None)
- **Optional** –
  - kwargs: **Keyworded arguments which can be provided like-**
    - timeout: Set NETCONF RPC timeout. Can be used for commands which take a while to execute. (default = 30 seconds)
    - comment: Provide a comment to the commit. (default = None)
    - confirm: Provide time in minutes for commit confirmation. If this option is specified, the commit will be rollbacked in the given time unless the commit is confirmed.

salt.states.junos.**shutdown**(*name*, ***kwargs*)

Shuts down the device.

```
shut the device:
  junos:
    - shutdown
    - in_min: 10
```

Parameters **Optional** –

- kwargs:
  - reboot: Whether to reboot instead of shutdown. (default=False)
  - at: Specify time for reboot. (To be used only if reboot=yes)
  - in_min: Specify delay in minutes for shutdown

salt.states.junos.**unlock**(*name*)

Unlocks the candidate configuration.

```
unlock the config:
  junos.unlock
```

salt.states.junos.**zeroize**(*name*)

Resets the device to default factory settings.

```
reset my device:
  junos.zeroize
```

name: can be anything

21.19.129 salt.states.kapacitor module

Kapacitor state module.

configuration This module accepts connection configuration details either as parameters or as configuration settings in /etc/salt/minion on the relevant minions:

```
kapacitor.unsafe_ssl: 'false'
kapacitor.protocol: 'http'
kapacitor.host: 'localhost'
kapacitor.port: 9092
```

This data can also be passed into pillar. Options passed into opts will overwrite options passed into pillar.

New in version 2016.11.0.

`salt.states.kapacitor.`**`task_absent`**`(`*name*`)`

Ensure that a task is absent from Kapacitor.

name Name of the task.

`salt.states.kapacitor.`**`task_present`**`(`*name, tick_script, task_type='stream', database=None, retention_policy='default', enable=True, dbrps=None*`)`

Ensure that a task is present and up-to-date in Kapacitor.

name Name of the task.

tick_script Path to the TICK script for the task. Can be a salt:// source.

task_type Task type. Defaults to 'stream'

dbrps A list of databases and retention policies in "dbname"."rpname" format to fetch data from. For backward compatibility, the value of 'database' and 'retention_policy' will be merged as part of dbrps.

New in version 2019.2.0.

database Which database to fetch data from. Defaults to None, which will use the default database in InfluxDB.

retention_policy Which retention policy to fetch data from. Defaults to 'default'.

enable Whether to enable the task or not. Defaults to True.

21.19.130 salt.states.kernelpkg

Manage kernel packages and active kernel version

Example state to install the latest kernel from package repositories:

```
install-latest-kernel:
  kernel.latest_installed: []
```

Example state to boot the system if a new kernel has been installed:

```
boot-latest-kernel:
  kernel.latest_active:
    - at_time: 1
```

Example state chaining the install and reboot operations:

```
install-latest-kernel:
  kernel.latest_installed: []

boot-latest-kernel:
  kernel.latest_active:
    - at_time: 1
    - onchanges:
      - kernel: install-latest-kernel
```

Chaining can also be achieved using wait/listen requisites:

```
install-latest-kernel:
  kernel.latest_installed: []

boot-latest-kernel:
  kernel.latest_wait:
    - at_time: 1
    - listen:
      - kernel: install-latest-kernel
```

`salt.states.kernelpkg.`**`latest_active`**(*name*, *at_time=None*, ***kwargs*)

Initiate a reboot if the running kernel is not the latest one installed.

Note: This state does not install any patches. It only compares the running kernel version number to other kernel versions also installed in the system. If the running version is not the latest one installed, this state will reboot the system.

See `kernelpkg.upgrade` and `latest_installed()` for ways to install new kernel packages.

This module does not attempt to understand or manage boot loader configurations it is possible to have a new kernel installed, but a boot loader configuration that will never activate it. For this reason, it would not be advisable to schedule this state to run automatically.

Because this state function may cause the system to reboot, it may be preferable to move it to the very end of the state run. See `latest_wait()` for a waitable state that can be called with the *listen* requesite.

---

name Arbitrary name for the state. Does not affect behavior.
at_time The wait time in minutes before the system will be rebooted.

`salt.states.kernelpkg.`**`latest_installed`**(*name*, ***kwargs*)

Ensure that the latest version of the kernel available in the repositories is installed.

Note: This state only installs the kernel, but does not activate it. The new kernel should become active at the next reboot. See `kernelpkg.needs_reboot` for details on how to detect this condition, and `latest_active()` to initiale a reboot when needed.

---

name Arbitrary name for the state. Does not affect behavior.

`salt.states.kernelpkg.`**`latest_wait`**(*name*, *at_time=None*, ***kwargs*)

Initiate a reboot if the running kernel is not the latest one installed. This is the waitable version of `latest_active()` and will not take any action unless triggered by a watch or listen requesite.

Note: Because this state function may cause the system to reboot, it may be preferable to move it to the very end of the state run using *listen* or *listen_in* requisites.

```
system-up-to-date:
  pkg.uptodate:
    - refresh: true

boot-latest-kernel:
  kernelpkg.latest_wait:
    - at_time: 1
    - listen:
      - pkg: system-up-to-date
```

---

name Arbitrary name for the state. Does not affect behavior.

at_time The wait time in minutes before the system will be rebooted.

salt.states.kernelpkg.**mod_watch**(*name*, *sfun*, ***kwargs*)
Execute a kernelpkg state based on a watch or listen call

21.19.131 salt.states.keyboard

Management of keyboard layouts

The keyboard layout can be managed for the system:

```
us:
  keyboard.system
```

Or it can be managed for XOrg:

```
us:
  keyboard.xorg
```

salt.states.keyboard.**system**(*name*)
Set the keyboard layout for the system

name The keyboard layout to use

salt.states.keyboard.**xorg**(*name*)
Set the keyboard layout for XOrg

layout The keyboard layout to use

21.19.132 salt.states.keystone

Management of Keystone users

depends

- keystoneclient Python module

configuration See `salt.modules.keystone` for setup instructions.

```
Keystone tenants:
  keystone.tenant_present:
    - names:
      - admin
      - demo
      - service

Keystone roles:
  keystone.role_present:
    - names:
      - admin
      - Member

admin:
  keystone.user_present:
    - password: R00T_4CC3SS
```

(continues on next page)

(continued from previous page)

```
    - email: admin@domain.com
    - roles:
        admin:   # tenants
          - admin  # roles
        service:
          - admin
          - Member
    - require:
      - keystone: Keystone tenants
      - keystone: Keystone roles

nova:
  keystone.user_present:
    - password: '$up3rn0v4'
    - email: nova@domain.com
    - tenant: service
    - roles:
        service:
          - admin
    - require:
      - keystone: Keystone tenants
      - keystone: Keystone roles

demo:
  keystone.user_present:
    - password: 'd3m0n$trati0n'
    - email: demo@domain.com
    - tenant: demo
    - roles:
        demo:
          - Member
    - require:
      - keystone: Keystone tenants
      - keystone: Keystone roles

nova service:
  keystone.service_present:
    - name: nova
    - service_type: compute
    - description: OpenStack Compute Service
```

`salt.states.keystone.`**`endpoint_absent`**`(`*name, region=None, profile=None, interface=None, **connection_args*`)`

Ensure that the endpoint for a service doesn't exist in Keystone catalog

name The name of the service whose endpoints should not exist

region (optional) The region of the endpoint. Defaults to `RegionOne`.

interface The interface type, which describes the visibility of the endpoint. (for V3 API)

`salt.states.keystone.`**`endpoint_present`**`(`*name, publicurl=None, internalurl=None, adminurl=None, region=None, profile=None, url=None, interface=None, **connection_args*`)`

Ensure the specified endpoints exists for service

name The Service name

publicurl The public url of service endpoint (for V2 API)

internalurl The internal url of service endpoint (for V2 API)

adminurl The admin url of the service endpoint (for V2 API)

region The region of the endpoint

url The endpoint URL (for V3 API)
interface The interface type, which describes the visibility of the endpoint. (for V3 API)

`salt.states.keystone.`**`project_absent`**`(`*name*, *profile=None*, ***connection_args*`)`
Ensure that the keystone project is absent. Alias for tenant_absent from V2 API to fulfill V3 API naming convention.

New in version 2016.11.0.
name The name of the project that should not exist

```
delete_nova:
    keystone.project_absent:
        - name: nova
```

`salt.states.keystone.`**`project_present`**`(`*name*, *description=None*, *enabled=True*, *profile=None*, ***connection_args*`)`
Ensures that the keystone project exists Alias for tenant_present from V2 API to fulfill V3 API naming convention.

New in version 2016.11.0.
name The name of the project to manage
description The description to use for this project
enabled Availability state for this project

```
nova:
    keystone.project_present:
        - enabled: True
        - description: 'Nova Compute Service'
```

`salt.states.keystone.`**`role_absent`**`(`*name*, *profile=None*, ***connection_args*`)`
Ensure that the keystone role is absent.
name The name of the role that should not exist

`salt.states.keystone.`**`role_present`**`(`*name*, *profile=None*, ***connection_args*`)`
' Ensures that the keystone role exists
name The name of the role that should be present

`salt.states.keystone.`**`service_absent`**`(`*name*, *profile=None*, ***connection_args*`)`
Ensure that the service doesn't exist in Keystone catalog
name The name of the service that should not exist

`salt.states.keystone.`**`service_present`**`(`*name*, *service_type*, *description=None*, *profile=None*, ***connection_args*`)`
Ensure service present in Keystone catalog
name The name of the service
service_type The type of Openstack Service
description (optional) Description of the service

`salt.states.keystone.`**`tenant_absent`**`(`*name*, *profile=None*, ***connection_args*`)`
Ensure that the keystone tenant is absent.
name The name of the tenant that should not exist

`salt.states.keystone.`**`tenant_present`**`(`*name*, *description=None*, *enabled=True*, *profile=None*, ***connection_args*`)`
Ensures that the keystone tenant exists
name The name of the tenant to manage
description The description to use for this tenant
enabled Availability state for this tenant

`salt.states.keystone.`**`user_absent`**(*name*, *profile=None*, ***connection_args*)

Ensure that the keystone user is absent.

name The name of the user that should not exist

`salt.states.keystone.`**`user_present`**(*name*, *password*, *email*, *tenant=None*, *enabled=True*, *roles=None*, *profile=None*, *password_reset=True*, *project=None*, ***connection_args*)

Ensure that the keystone user is present with the specified properties.

name The name of the user to manage

password The password to use for this user.

> **Note:** If the user already exists and a different password was set for the user than the one specified here, the password for the user will be updated. Please set the `password_reset` option to `False` if this is not the desired behavior.

password_reset Whether or not to reset password after initial set. Defaults to `True`.

email The email address for this user

tenant The tenant (name) for this user

project The project (name) for this user (overrides tenant in api v3)

enabled Availability state for this user

roles The roles the user should have under given tenants. Passed as a dictionary mapping tenant names to a list of roles in this tenant, i.e.:

```
roles:
    admin:   # tenant
      - admin  # role
    service:
      - admin
      - Member
```

21.19.133 salt.states.keystone_domain

Management of OpenStack Keystone Domains

New in version 2018.3.0.

depends shade

configuration see *`salt.modules.keystoneng`* for setup instructions

Example States

```
create domain:
  keystone_domain.present:
    - name: domain1

create domain with optional params:
  keystone_domain.present:
    - name: domain1
    - enabled: False
    - description: 'my domain'

delete domain:
  keystone_domain.absent:
    - name: domain1
```

`salt.states.keystone_domain.`**`absent`**`(`*name, auth=None*`)`

Ensure domain does not exist

name Name of the domain

`salt.states.keystone_domain.`**`present`**`(`*name, auth=None, **kwargs*`)`

Ensure domain exists and is up-to-date

name Name of the domain

enabled Boolean to control if domain is enabled

description An arbitrary description of the domain

21.19.134 salt.states.keystone_endpoint

Management of OpenStack Keystone Endpoints

New in version 2018.3.0.

depends shade

configuration see `salt.modules.keystoneng` for setup instructions

Example States

```
create endpoint:
  keystone_endpoint.present:
    - name: public
    - url: https://example.org:9292
    - region: RegionOne
    - service_name: glance

destroy endpoint:
  keystone_endpoint.absent:
    - name: public
    - url: https://example.org:9292
    - region: RegionOne
    - service_name: glance

create multiple endpoints:
  keystone_endpoint.absent:
    - names:
        - public
        - admin
        - internal
    - url: https://example.org:9292
    - region: RegionOne
    - service_name: glance
```

`salt.states.keystone_endpoint.`**`absent`**`(`*name, service_name, auth=None, **kwargs*`)`

Ensure an endpoint does not exists

name Interface name

url URL of the endpoint

service_name Service name or ID

region The region name to assign the endpoint

`salt.states.keystone_endpoint.`**`present`**`(`*name, service_name, auth=None, **kwargs*`)`

Ensure an endpoint exists and is up-to-date

name Interface name

url URL of the endpoint

service_name Service name or ID
region The region name to assign the endpoint
enabled Boolean to control if endpoint is enabled

21.19.135 salt.states.keystone_group

Management of OpenStack Keystone Groups

New in version 2018.3.0.

depends shade

configuration see `salt.modules.keystoneng` for setup instructions

Example States

```
create group:
  keystone_group.present:
    - name: group1

delete group:
  keystone_group.absent:
    - name: group1

create group with optional params:
  keystone_group.present:
    - name: group1
    - domain: domain1
    - description: 'my group'
```

`salt.states.keystone_group.`**`absent`**`(`*name, auth=None, **kwargs*`)`
Ensure group does not exist
name Name of the group
domain The name or id of the domain

`salt.states.keystone_group.`**`present`**`(`*name, auth=None, **kwargs*`)`
Ensure an group exists and is up-to-date
name Name of the group
domain The name or id of the domain
description An arbitrary description of the group

21.19.136 salt.states.keystone_project

Management of OpenStack Keystone Projects

New in version 2018.3.0.

depends shade

configuration see `salt.modules.keystoneng` for setup instructions

Example States

```
create project:
  keystone_project.present:
    - name: project1
```

(continues on next page)

(continued from previous page)

```
delete project:
  keystone_project.absent:
    - name: project1

create project with optional params:
  keystone_project.present:
    - name: project1
    - domain: domain1
    - enabled: False
    - description: 'my project'
```

`salt.states.keystone_project.`**`absent`**(*name*, *auth=None*, ***kwargs*)

Ensure a project does not exists

name Name of the project

domain The name or id of the domain

`salt.states.keystone_project.`**`present`**(*name*, *auth=None*, ***kwargs*)

Ensure a project exists and is up-to-date

name Name of the project

domain The name or id of the domain

description An arbitrary description of the project

21.19.137 salt.states.keystone_role

Management of OpenStack Keystone Roles

New in version 2018.3.0.

depends shade

configuration see *`salt.modules.keystoneng`* for setup instructions

Example States

```
create role:
  keystone_role.present:
    - name: role1

delete role:
  keystone_role.absent:
    - name: role1

create role with optional params:
  keystone_role.present:
    - name: role1
    - description: 'my group'
```

`salt.states.keystone_role.`**`absent`**(*name*, *auth=None*, ***kwargs*)

Ensure role does not exist

name Name of the role

`salt.states.keystone_role.`**`present`**(*name*, *auth=None*, ***kwargs*)

Ensure an role exists

name Name of the role

description An arbitrary description of the role

21.19.138 salt.states.keystone_role_grant

Management of OpenStack Keystone Role Grants

New in version 2018.3.0.

depends shade

configuration see `salt.modules.keystoneng` for setup instructions

Example States

```
create group:
  keystone_group.present:
    - name: group1

delete group:
  keystone_group.absent:
    - name: group1

create group with optional params:
  keystone_group.present:
    - name: group1
    - domain: domain1
    - description: 'my group'
```

`salt.states.keystone_role_grant.`**`absent`**(*name*, *auth=None*, ***kwargs*)

`salt.states.keystone_role_grant.`**`present`**(*name*, *auth=None*, ***kwargs*)

21.19.139 salt.states.keystone_service

Management of OpenStack Keystone Services

New in version 2018.3.0.

depends shade

configuration see `salt.modules.keystoneng` for setup instructions

Example States

```
create service:
  keystone_service.present:
    - name: glance
    - type: image

delete service:
  keystone_service.absent:
    - name: glance

create service with optional params:
  keystone_service.present:
    - name: glance
    - type: image
    - enabled: False
    - description: 'OpenStack Image'
```

`salt.states.keystone_service.`**`absent`**`(`*name*, *auth=None*`)`

Ensure service does not exist

name Name of the service

`salt.states.keystone_service.`**`present`**`(`*name*, *auth=None*, ***kwargs*`)`

Ensure an service exists and is up-to-date

name Name of the group

type Service type

enabled Boolean to control if service is enabled

description An arbitrary description of the service

21.19.140 salt.states.keystone_user

Management of OpenStack Keystone Users

New in version 2018.3.0.

depends shade

configuration see *`salt.modules.keystoneng`* for setup instructions

Example States

```
create user:
  keystone_user.present:
    - name: user1

delete user:
  keystone_user.absent:
    - name: user1

create user with optional params:
  keystone_user.present:
    - name: user1
    - domain: domain1
    - enabled: False
    - password: password123
    - email: "user1@example.org"
    - description: 'my user'
```

`salt.states.keystone_user.`**`absent`**`(`*name*, *auth=None*, ***kwargs*`)`

Ensure user does not exists

name Name of the user

domain The name or id of the domain

`salt.states.keystone_user.`**`present`**`(`*name*, *auth=None*, ***kwargs*`)`

Ensure domain exists and is up-to-date

name Name of the domain

domain The name or id of the domain

enabled Boolean to control if domain is enabled

description An arbitrary description of the domain

password The user password

email The users email address

21.19.141 salt.states.keystore

State management of a java keystore

`salt.states.keystore.`**`managed`**(*name*, *passphrase*, *entries*, *force_remove=False*)

Create or manage a java keystore.

name The path to the keystore file

passphrase The password to the keystore

entries A list containing an alias, certificate, and optional private_key. The certificate and private_key can be a file or a string

```
- entries:
  - alias: hostname2
    certificate: /path/to/cert.crt
    private_key: /path/to/key.key
  - alias: stringhost
    certificate: |
      -----BEGIN CERTIFICATE-----
      MIICEjCCAXsCAg36MA0GCSqGSIb3DQEBBQUAMIGbMQswCQYDVQQGEwJKUDEOMAwG
      ...
      2VguKv4SWjRFoRkIfIlHX0qVviMhSlNy2ioFLy7JcPZb+v3ftDGywUqcBiVDoea0
      -----END CERTIFICATE-----
```

force_remove If True will cause the state to remove any entries found in the keystore which are not defined in the state. The default is False.

Example

```
define_keystore:
  keystore.managed:
    - name: /path/to/keystore
    - passphrase: changeit
    - force_remove: True
    - entries:
      - alias: hostname1
        certificate: /path/to/cert.crt
      - alias: remotehost
        certificate: /path/to/cert2.crt
        private_key: /path/to/key2.key
      - alias: pillarhost
        certificate: {{ salt.pillar.get('path:to:cert') }}
```

21.19.142 salt.states.kmod

Loading and unloading of kernel modules

The Kernel modules on a system can be managed cleanly with the kmod state module:

```
add_kvm:
  kmod.present:
    - name: kvm_amd
remove_beep:
  kmod.absent:
    - name: pcspkr
```

Multiple modules can be specified for both kmod.present and kmod.absent.

```
add_sound:
  kmod.present:
    - mods:
      - snd_hda_codec_hdmi
      - snd_hda_codec
      - snd_hwdep
      - snd_hda_core
      - snd_pcm
      - snd_timer
      - snd
```

`salt.states.kmod.`**`absent`**`(`*name, persist=False, comment=True, mods=None*`)`

Verify that the named kernel module is not loaded

name The name of the kernel module to verify is not loaded

persist Remove module from `/etc/modules`

comment Comment out module in `/etc/modules` rather than remove it

mods A list of modules to verify are unloaded. If this argument is used, the `name` argument, although still required, is not used, and becomes a placeholder

New in version 2016.3.0.

`salt.states.kmod.`**`present`**`(`*name, persist=False, mods=None*`)`

Ensure that the specified kernel module is loaded

name The name of the kernel module to verify is loaded

persist Also add module to `/etc/modules`

mods A list of modules to verify are loaded. If this argument is used, the `name` argument, although still required, is not used, and becomes a placeholder

New in version 2016.3.0.

21.19.143 salt.states.kubernetes

Manage kubernetes resources as salt states

NOTE: This module requires the proper pillar values set. See salt.modules.kubernetesmod for more information.

Warning: Configuration options will change in 2019.2.0.

The kubernetes module is used to manage different kubernetes resources.

```
my-nginx:
  kubernetes.deployment_present:
    - namespace: default
      metadata:
        app: frontend
      spec:
        replicas: 1
        template:
          metadata:
            labels:
              run: my-nginx
          spec:
            containers:
            - name: my-nginx
```

(continues on next page)

(continued from previous page)

```
              image: nginx
              ports:
              - containerPort: 80

my-mariadb:
  kubernetes.deployment_absent:
    - namespace: default

# kubernetes deployment as specified inside of
# a file containing the definition of the the
# deployment using the official kubernetes format
redis-master-deployment:
  kubernetes.deployment_present:
    - name: redis-master
    - source: salt://k8s/redis-master-deployment.yml
  require:
    - pip: kubernetes-python-module

# kubernetes service as specified inside of
# a file containing the definition of the the
# service using the official kubernetes format
redis-master-service:
  kubernetes.service_present:
    - name: redis-master
    - source: salt://k8s/redis-master-service.yml
  require:
    - kubernetes.deployment_present: redis-master

# kubernetes deployment as specified inside of
# a file containing the definition of the the
# deployment using the official kubernetes format
# plus some jinja directives
 nginx-source-template:
  kubernetes.deployment_present:
    - source: salt://k8s/nginx.yml.jinja
    - template: jinja
  require:
    - pip: kubernetes-python-module


# Kubernetes secret
k8s-secret:
  kubernetes.secret_present:
    - name: top-secret
      data:
        key1: value1
        key2: value2
        key3: value3
```

salt.states.kubernetes.**configmap_absent**(name, namespace=’default’, **kwargs)

Ensures that the named configmap is absent from the given namespace.

name The name of the configmap

namespace The namespace holding the configmap. The ’default’ one is going to be used unless a different one is specified.

salt.states.kubernetes.**configmap_present**(name, namespace=’default’, data=None, source=None, template=None, **kwargs)

Ensures that the named configmap is present inside of the specified namespace with the given data. If the configmap exists it will be replaced.

name The name of the configmap.

namespace The namespace holding the configmap. The 'default' one is going to be used unless a different one is specified.

data The dictionary holding the configmaps.

source A file containing the data of the configmap in plain format.

template Template engine to be used to render the source file.

`salt.states.kubernetes.`**`deployment_absent`**(*name*, *namespace='default'*, ***kwargs*)

Ensures that the named deployment is absent from the given namespace.

name The name of the deployment

namespace The name of the namespace

`salt.states.kubernetes.`**`deployment_present`**(*name*, *namespace='default'*, *metadata=None*, *spec=None*, *source="*, *template="*, ***kwargs*)

Ensures that the named deployment is present inside of the specified namespace with the given metadata and spec. If the deployment exists it will be replaced.

name The name of the deployment.

namespace The namespace holding the deployment. The 'default' one is going to be used unless a different one is specified.

metadata The metadata of the deployment object.

spec The spec of the deployment object.

source A file containing the definition of the deployment (metadata and spec) in the official kubernetes format.

template Template engine to be used to render the source file.

`salt.states.kubernetes.`**`namespace_absent`**(*name*, ***kwargs*)

Ensures that the named namespace is absent.

name The name of the namespace

`salt.states.kubernetes.`**`namespace_present`**(*name*, ***kwargs*)

Ensures that the named namespace is present.

name The name of the namespace.

`salt.states.kubernetes.`**`node_label_absent`**(*name*, *node*, ***kwargs*)

Ensures that the named label is absent from the node.

name The name of the label

node The name of the node

`salt.states.kubernetes.`**`node_label_folder_absent`**(*name*, *node*, ***kwargs*)

Ensures the label folder doesn't exist on the specified node.

name The name of label folder

node The name of the node

`salt.states.kubernetes.`**`node_label_present`**(*name*, *node*, *value*, ***kwargs*)

Ensures that the named label is set on the named node with the given value. If the label exists it will be replaced.

name The name of the label.

value Value of the label.

node Node to change.

`salt.states.kubernetes.`**`pod_absent`**(*name*, *namespace='default'*, ***kwargs*)

Ensures that the named pod is absent from the given namespace.

name The name of the pod

namespace The name of the namespace

`salt.states.kubernetes.`**`pod_present`**(*name*, *namespace='default'*, *metadata=None*, *spec=None*, *source="*, *template="*, ***kwargs*)

Ensures that the named pod is present inside of the specified namespace with the given metadata and spec. If the pod exists it will be replaced.

name The name of the pod.

namespace The namespace holding the pod. The 'default' one is going to be used unless a different one is specified.

metadata The metadata of the pod object.

spec The spec of the pod object.

source A file containing the definition of the pod (metadata and spec) in the official kubernetes format.

template Template engine to be used to render the source file.

`salt.states.kubernetes.`**`secret_absent`**`(`*name*, *namespace='default'*, ***kwargs*`)`

Ensures that the named secret is absent from the given namespace.

name The name of the secret

namespace The name of the namespace

`salt.states.kubernetes.`**`secret_present`**`(`*name*, *namespace='default'*, *data=None*, *source=None*, *template=None*, ***kwargs*`)`

Ensures that the named secret is present inside of the specified namespace with the given data. If the secret exists it will be replaced.

name The name of the secret.

namespace The namespace holding the secret. The 'default' one is going to be used unless a different one is specified.

data The dictionary holding the secrets.

source A file containing the data of the secret in plain format.

template Template engine to be used to render the source file.

`salt.states.kubernetes.`**`service_absent`**`(`*name*, *namespace='default'*, ***kwargs*`)`

Ensures that the named service is absent from the given namespace.

name The name of the service

namespace The name of the namespace

`salt.states.kubernetes.`**`service_present`**`(`*name*, *namespace='default'*, *metadata=None*, *spec=None*, *source=''*, *template=''*, ***kwargs*`)`

Ensures that the named service is present inside of the specified namespace with the given metadata and spec. If the deployment exists it will be replaced.

name The name of the service.

namespace The namespace holding the service. The 'default' one is going to be used unless a different one is specified.

metadata The metadata of the service object.

spec The spec of the service object.

source A file containing the definition of the service (metadata and spec) in the official kubernetes format.

template Template engine to be used to render the source file.

21.19.144 salt.states.layman

Management of Gentoo Overlays using layman

A state module to manage Gentoo package overlays via layman

```
sunrise:
    layman.present
```

`salt.states.layman.`**`absent`**`(`*name*`)`

Verify that the overlay is absent

name The name of the overlay to delete `salt.states.layman.`**`present`**`(`*name*`)`

Verify that the overlay is present

name The name of the overlay to add

21.19.145 salt.states.ldap

Manage entries in an LDAP database

New in version 2016.3.0.

The `states.ldap` state module allows you to manage LDAP entries and their attributes.

`salt.states.ldap.`**`managed`**`(`*name*, *entries*, *connect_spec=None*`)`

Ensure the existence (or not) of LDAP entries and their attributes

Example:

```
ldapi:///:
  ldap.managed:
    - connect_spec:
        bind:
          method: sasl

    - entries:

      # make sure the entry doesn't exist
      - cn=foo,ou=users,dc=example,dc=com:
        - delete_others: True

      # make sure the entry exists with only the specified
      # attribute values
      - cn=admin,dc=example,dc=com:
        - delete_others: True
        - replace:
            cn:
              - admin
            description:
              - LDAP administrator
            objectClass:
              - simpleSecurityObject
              - organizationalRole
            userPassword:
              - {{pillar.ldap_admin_password}}

      # make sure the entry exists, its olcRootDN attribute
      # has only the specified value, the olcRootDN attribute
      # doesn't exist, and all other attributes are ignored
      - 'olcDatabase={1}hdb,cn=config':
        - replace:
            olcRootDN:
              - cn=admin,dc=example,dc=com
            # the admin entry has its own password attribute
            olcRootPW: []

      # note the use of 'default'.  also note how you don't
      # have to use list syntax if there is only one attribute
      # value
```

(continues on next page)

(continued from previous page)

```
      - cn=foo,ou=users,dc=example,dc=com:
        - delete_others: True
        - default:
            userPassword: changeme
            shadowLastChange: 0
            # keep sshPublicKey if present, but don't create
            # the attribute if it is missing
            sshPublicKey: []
        - replace:
            cn: foo
            uid: foo
            uidNumber: 1000
            gidNumber: 1000
            gecos: Foo Bar
            givenName: Foo
            sn: Bar
            homeDirectory: /home/foo
            loginShell: /bin/bash
            objectClass:
              - inetOrgPerson
              - posixAccount
              - top
              - ldapPublicKey
              - shadowAccount
```

Parameters

- **`name`** – The URL of the LDAP server. This is ignored if `connect_spec` is either a connection object or a dict with a `'url'` entry.
- **`entries`** – A description of the desired state of zero or more LDAP entries.

  `entries` is an iterable of dicts. Each of these dict's keys are the distinguished names (DNs) of LDAP entries to manage. Each of these dicts is processed in order. A later dict can reference an LDAP entry that was already mentioned in an earlier dict, which makes it possible for later dicts to enhance or alter the desired state of an LDAP entry.

  The DNs are mapped to a description of the LDAP entry's desired state. These LDAP entry descriptions are themselves iterables of dicts. Each dict in the iterable is processed in order. They contain directives controlling the entry's state. The key names the directive type and the value is state information for the directive. The specific structure of the state information depends on the directive type.

  The structure of `entries` looks like this:

  ```
  [{dn1: [{directive1: directive1_state,
           directive2: directive2_state},
          {directive3: directive3_state}],
    dn2: [{directive4: directive4_state,
           directive5: directive5_state}]},
   {dn3: [{directive6: directive6_state}]}]
  ```

  These are the directives:
  - **`'delete_others'`** Boolean indicating whether to delete attributes not mentioned in this dict or any of the other directive dicts for this DN. Defaults to `False`.

    If you don't want to delete an attribute if present, but you also don't want to add it if it is missing or modify it if it is present, you can use either the `'default'` directive or the `'add'` directive with an empty value list.

    - **`'default'`** A dict mapping an attribute name to an iterable of default values for that attribute. If the attribute already exists, it is left alone. If not, it is created using the given list of values.

      An empty value list is useful when you don't want to create an attribute if it is missing but you do want to preserve it if the `'delete_others'` key is `True`.
    - **`'add'`** Attribute values to add to the entry. This is a dict mapping an attribute name to an iterable of values to add.

      An empty value list is useful when you don't want to create an attribute if it is missing but you do want to preserve it if the `'delete_others'` key is `True`.
    - **`'delete'`** Attribute values to remove from the entry. This is a dict mapping an attribute name to an iterable of values to delete from the attribute. If the iterable is empty, all of the attribute's values are deleted.
    - **`'replace'`** Attributes to replace. This is a dict mapping an attribute name to an iterable of values. Any existing values for the attribute are deleted, then the given values are added. The iterable may be empty.

  In the above directives, the iterables of attribute values may instead be `None`, in which case an empty list is used, or a scalar such as a string or number, in which case a new list containing the scalar is used.

  Note that if all attribute values are removed from an entry, the entire entry is deleted.
- **`connect_spec`** – See the description of the `connect_spec` parameter of the *`ldap3.connect`* function in the *`ldap3`* execution module. If this is a dict and the `'url'` entry is not specified, the `'url'` entry is set to the value of the `name` parameter.

Returns

A dict with the following keys:

- **`'name'`** This is the same object passed to the `name` parameter.
- **`'changes'`** This is a dict describing the changes made (or, in test mode, the changes that would have been attempted). If no changes were made (or no changes would have been attempted), then this dict is empty. Only successful changes are included.

  Each key is a DN of an entry that was changed (or would have been changed). Entries that were not changed (or would not have been changed) are not included. The value is a dict with two keys:
  - **`'old'`** The state of the entry before modification. If the entry did not previously exist, this key maps to `None`. Otherwise, the value is a dict mapping each of the old entry's attributes to a list of its values before any modifications were made. Unchanged attributes are excluded from this dict.
  - **`'new'`** The state of the entry after modification. If the entry was deleted, this key maps to `None`. Otherwise, the value is a dict mapping each of the entry's attributes to a list of its values after the modifications were made. Unchanged attributes are excluded from this dict.

  Example `'changes'` dict where a new entry was created with a single attribute containing two values:

  ```
  {'dn1': {'old': None,
           'new': {'attr1': ['val1', 'val2']}}}
  ```

  Example `'changes'` dict where a new attribute was added to an existing entry:

```
{'dn1': {'old': {},
         'new': {'attr2': ['val3']}}}
```

- **'result'** One of the following values:
  - `True` if no changes were necessary or if all changes were applied successfully.
  - `False` if at least one change was unable to be applied.
  - `None` if changes would be applied but it is in test mode.

21.19.146 salt.states.libcloud_dns module

Manage DNS records and zones using libcloud

codeauthor Anthony Shaw <anthonyshaw@apache.org>

New in version 2016.11.0.

Create and delete DNS records or zones through Libcloud. Libcloud's DNS system supports over 20 DNS providers including Amazon, Google, GoDaddy, Softlayer

This module uses `libcloud`, which can be installed via package, or pip.

configuration This module uses a configuration profile for one or multiple DNS providers

```
libcloud_dns:
  profile1:
    driver: godaddy
    key: 2orgk34kgk34g
  profile2:
    driver: route53
    key: blah
    secret: blah
```

Example:

```
my-zone:
  libcloud_dns.zone_present:
    - name: mywebsite.com
    - profile: profile1
my-website:
  libcloud_dns.record_present:
    - name: www
    - zone: mywebsite.com
    - type: A
    - data: 12.34.32.3
    - profile: profile1
    - require:
      - libcloud_dns: my-zone
```

depends apache-libcloud

`salt.states.libcloud_dns.`**`record_absent`**(*name*, *zone*, *type*, *data*, *profile*)

Ensures a record is absent.

Parameters

- **name** (`str`) – Record name without the domain name (e.g. www). Note: If you want to create a record for a base domain name, you should specify empty string (") for this argument.

- **zone** (`str`) – Zone where the requested record is created, the domain name
- **type** (`str`) – DNS record type (A, AAAA, …).
- **data** (`str`) – Data for the record (depends on the record type).
- **profile** (`str`) – The profile key

`salt.states.libcloud_dns.`**`record_present`**(*name*, *zone*, *type*, *data*, *profile*)

Ensures a record is present.

Parameters

- **name** (`str`) – Record name without the domain name (e.g. www). Note: If you want to create a record for a base domain name, you should specify empty string (") for this argument.
- **zone** (`str`) – Zone where the requested record is created, the domain name
- **type** (`str`) – DNS record type (A, AAAA, …).
- **data** (`str`) – Data for the record (depends on the record type).
- **profile** (`str`) – The profile key

`salt.states.libcloud_dns.`**`state_result`**(*result*, *message*, *name*, *changes=None*)

`salt.states.libcloud_dns.`**`zone_absent`**(*domain*, *profile*)

Ensures a record is absent.

Parameters

- **domain** (`str`) – Zone name, i.e. the domain name
- **profile** (`str`) – The profile key

`salt.states.libcloud_dns.`**`zone_present`**(*domain*, *type*, *profile*)

Ensures a record is present.

Parameters

- **domain** (`str`) – Zone name, i.e. the domain name
- **type** (`str`) – Zone type (master / slave), defaults to master
- **profile** (`str`) – The profile key

21.19.147 salt.states.libcloud_loadbalancer module

Apache Libcloud Load Balancer State

Manage load balancers using libcloud

codeauthor `Anthony Shaw <anthonyshaw@apache.org>`

Apache Libcloud load balancer management for a full list of supported clouds, see http://libcloud.readthedocs.io/en/latest/loadbalancer/supported_providers.html

Clouds include Amazon ELB, ALB, Google, Aliyun, CloudStack, Softlayer

New in version 2018.3.0.

configuration This module uses a configuration profile for one or multiple Cloud providers

```
libcloud_loadbalancer:
    profile_test1:
      driver: gce
      key: GOOG0123456789ABCXYZ
      secret: mysecret
    profile_test2:
      driver: alb
      key: 12345
      secret: mysecret
```

Example:

Using States to deploy a load balancer with extended arguments to specify region

```
lb_test:
    libcloud_loadbalancer.balancer_present:
        - name: example
        - port: 80
        - protocol: http
        - profile: google
        - ex_region: us-east1
```

depends apache-libcloud

`salt.states.libcloud_loadbalancer.`**`balancer_absent`**`(`*name*, *profile*, ***libcloud_kwargs*`)`

Ensures a load balancer is absent.

Parameters

- **`name`** (`str`) – Load Balancer name
- **`profile`** (`str`) – The profile key

`salt.states.libcloud_loadbalancer.`**`balancer_present`**`(`*name*, *port*, *protocol*, *profile*, *algorithm=None*, *members=None*, ***libcloud_kwargs*`)`

Ensures a load balancer is present.

Parameters

- **`name`** (`str`) – Load Balancer name
- **`port`** (`str`) – Port the load balancer should listen on, defaults to 80
- **`protocol`** (`str`) – Loadbalancer protocol, defaults to http.
- **`profile`** (`str`) – The profile key
- **`algorithm`** (`str`) – Load balancing algorithm, defaults to ROUND_ROBIN. See Algorithm type in Libcloud documentation for a full listing.
- **`members`** (`list` of `dict` (ip, port)) – An optional list of members to create on deployment

`salt.states.libcloud_loadbalancer.`**`member_absent`**`(`*ip*, *port*, *balancer_id*, *profile*, ***libcloud_kwargs*`)`

Ensure a load balancer member is absent, based on IP and Port

Parameters

- **`ip`** (`str`) – IP address for the member
- **`port`** (`int`) – Port for the member
- **`balancer_id`** (`str`) – id of a load balancer you want to detach the member from
- **`profile`** (`str`) – The profile key

`salt.states.libcloud_loadbalancer.`**`member_present`**`(`*ip*, *port*, *balancer_id*, *profile*, ***libcloud_kwargs*`)`

Ensure a load balancer member is present

Parameters

- **`ip`** (`str`) – IP address for the new member
- **`port`** (`int`) – Port for the new member
- **`balancer_id`** (`str`) – id of a load balancer you want to attach the member to
- **`profile`** (`str`) – The profile key

`salt.states.libcloud_loadbalancer.`**`state_result`**`(`*result*, *message*, *name*, *changes=None*`)`

21.19.148 salt.states.libcloud_storage module

Apache Libcloud Storage State

Manage cloud storage using libcloud

codeauthor `Anthony Shaw <anthonyshaw@apache.org>`

Apache Libcloud Storage (object/blob) management for a full list of supported clouds, see http://libcloud.readthedocs.io/en/latest/storage/supported_providers.html

Clouds include Amazon S3, Google Storage, Aliyun, Azure Blobs, Ceph, OpenStack swift

New in version 2018.3.0.

configuration This module uses a configuration profile for one or multiple Storage providers

```
libcloud_storage:
    profile_test1:
      driver: google_storage
      key: GOOG0123456789ABCXYZ
      secret: mysecret
    profile_test2:
      driver: s3
      key: 12345
      secret: mysecret
```

Examples

Creating a container and uploading a file

```
web_things:
  libcloud_storage.container_present:
    name: my_container_name
    profile: profile1
  libcloud_storage.object_present:
    name: my_file.jpg
    container: my_container_name
    path: /path/to/local/file.jpg
    profile: profile1
```

Downloading a file

This example will download the file from the remote cloud and keep it locally

```
web_things:
  libcloud_storage.file_present:
    name: my_file.jpg
    container: my_container_name
    path: /path/to/local/file.jpg
    profile: profile1
```

depends apache-libcloud

`salt.states.libcloud_storage.`**`container_absent`**`(`*name, profile*`)`

Ensures a container is absent.

Parameters

- **`name`** (`str`) – Container name

- **profile** (str) – The profile key

`salt.states.libcloud_storage.`**`container_present`**(*name, profile*)

Ensures a container is present.

Parameters

- **name** (str) – Container name
- **profile** (str) – The profile key

`salt.states.libcloud_storage.`**`file_present`**(*container, name, path, profile, overwrite_existing=False*)

Ensures a object is downloaded locally.

Parameters

- **container** (str) – Container name
- **name** (str) – Object name in cloud
- **path** (str) – Local path to file
- **profile** (str) – The profile key
- **overwrite_existing** (bool) – Replace if already exists

`salt.states.libcloud_storage.`**`object_absent`**(*container, name, profile*)

Ensures a object is absent.

Parameters

- **container** (str) – Container name
- **name** (str) – Object name in cloud
- **profile** (str) – The profile key

`salt.states.libcloud_storage.`**`object_present`**(*container, name, path, profile*)

Ensures a object is presnt.

Parameters

- **container** (str) – Container name
- **name** (str) – Object name in cloud
- **path** (str) – Local path to file
- **profile** (str) – The profile key

`salt.states.libcloud_storage.`**`state_result`**(*result, message, name, changes*)

21.19.149 salt.states.linux_acl

Linux File Access Control Lists

The Linux ACL state module requires the *getfacl* and *setfacl* binaries.

Ensure a Linux ACL is present

```
root:
  acl.present:
    - name: /root
    - acl_type: user
    - acl_name: damian
    - perms: rwx
```

Ensure a Linux ACL does not exist

```
root:
  acl.absent:
    - name: /root
    - acl_type: user
    - acl_name: damian
    - perms: rwx
```

Ensure a Linux ACL list is present

```
root:
  acl.list_present:
    - name: /root
    - acl_type: user
    - acl_names:
      - damian
      - homer
    - perms: rwx
```

Ensure a Linux ACL list does not exist

```
root:
  acl.list_absent:
    - name: /root
    - acl_type: user
    - acl_names:
      - damian
      - homer
    - perms: rwx
```

`salt.states.linux_acl.`**`absent`**(*name, acl_type, acl_name=”, perms=”, recurse=False*)

Ensure a Linux ACL does not exist

name The acl path

acl_type The type of the acl is used for, it can be 'user' or 'group'

acl_name The user or group

perms Remove the permissions eg.: rwx

recurse Set the permissions recursive in the path

`salt.states.linux_acl.`**`list_absent`**(*name, acl_type, acl_names=None, recurse=False*)

Ensure a Linux ACL list does not exist

Takes a list of acl names and remove them from the given path

name The acl path

acl_type The type of the acl is used for, it can be 'user' or 'group'

acl_names The list of users or groups

perms Remove the permissions eg.: rwx

recurse Set the permissions recursive in the path

`salt.states.linux_acl.`**`list_present`**(*name, acl_type, acl_names=None, perms=”, recurse=False, force=False*)

Ensure a Linux ACL list is present

Takes a list of acl names and add them to the given path

name The acl path

acl_type The type of the acl is used for it can be 'user' or 'group'

acl_names The list of users or groups

perms Set the permissions eg.: rwx

recurse Set the permissions recursive in the path

force Wipe out old permissions and ensure only the new permissions are set

`salt.states.linux_acl.`**`present`**(*name, acl_type, acl_name=”, perms=”, recurse=False, force=False*)

Ensure a Linux ACL is present

name The acl path

acl_type The type of the acl is used for it can be 'user' or 'group'

acl_name The user or group

perms Set the permissions eg.: rwx

recurse Set the permissions recursive in the path
force Wipe out old permissions and ensure only the new permissions are set

21.19.150 salt.states.locale

Management of languages/locales

Manage the available locales and the system default:

```
us_locale:
  locale.present:
    - name: en_US.UTF-8

default_locale:
  locale.system:
    - name: en_US.UTF-8
    - require:
      - locale: us_locale
```

`salt.states.locale.`**`present`**`(`*name*`)`

Generate a locale if it is not present

New in version 2014.7.0.

name The name of the locale to be present. Some distributions require the charmap to be specified as part of the locale at this point.

`salt.states.locale.`**`system`**`(`*name*`)`

Set the locale for the system
name The name of the locale to use

21.19.151 salt.states.logadm module

Management of logs using Solaris logadm.

maintainer Jorge Schrauwen <sjorge@blackdot.be>

maturity new

depends salt.modulus.logadm

platform Oracle Solaris, Sun Solaris, illumos

New in version nitrogen.

```
.. note::
    TODO
```

`salt.states.logadm.`**`remove`**`(`*name, log_file=None*`)`

Remove a log from the logadm configuration
name [string] entryname
log_file [string] (optional) log file path

Note: If log_file is specified it will be used instead of the entry name.

`salt.states.logadm.`**`rotate`**`(`*name, **kwargs*`)`

Add a log to the logadm configuration

name [string] alias for entryname
kwargs [boolean|string|int] optional additional flags and parameters

21.19.152 salt.states.logrotate module

Module for managing logrotate.

New in version 2017.7.0.

`salt.states.logrotate.set_`(*name*, *key*, *value*, *setting=None*, *conf_file='/etc/logrotate.conf'*)

Set a new value for a specific configuration line.

Parameters

- **key** (*str*) – The command or block to configure.
- **value** (*str*) – The command value or command of the block specified by the key parameter.
- **setting** (*str*) – The command value for the command specified by the value parameter.
- **conf_file** (*str*) – The logrotate configuration file.

Example of usage with only the required arguments:

```
logrotate-rotate:
  logrotate.set:
    - key: rotate
    - value: 2
```

Example of usage specifying all available arguments:

```
logrotate-wtmp-rotate:
  logrotate.set:
    - key: /var/log/wtmp
    - value: rotate
    - setting: 2
    - conf_file: /etc/logrotate.conf
```

21.19.153 salt.states.loop module

Loop state

Allows for looping over execution modules.

New in version 2017.7.0.

In both examples below, the execution module function `boto_elb.get_instance_health` returns a list of dicts. The condition checks the `state`-key of the first dict in the returned list and compares its value to the string *InService*.

```
wait_for_service_to_be_healthy:
  loop.until:
    - name: boto_elb.get_instance_health
    - condition: m_ret[0]['state'] == 'InService'
    - period: 5
    - timeout: 20
    - m_args:
      - {{ elb }}
    - m_kwargs:
        keyid: {{ access_key }}
```

(continues on next page)

(continued from previous page)

```
        key: {{ secret_key }}
        instances: "{{ instance }}"
```

Warning: This state allows arbitrary python code to be executed through the condition parameter which is literally evaluated within the state. Please use caution.

Changed in version 3000.

A version that does not use eval is now available. It uses either the python `operator` to compare the result of the function called in `name`, which can be one of the following: lt, le, eq (default), ne, ge, gt. Alternatively, *compare_operator* can be filled with a function from an execution module in `__salt__` or `__utils__` like the example below. The function `data.subdict_match` checks if the `expected` expression matches the data returned by calling the `name` function (with passed `args` and `kwargs`).

```
Wait for service to be healthy:
  loop.until_no_eval:
    - name: boto_elb.get_instance_health
    - expected: '0:state:InService'
    - compare_operator: data.subdict_match
    - period: 5
    - timeout: 20
    - args:
      - {{ elb }}
    - kwargs:
        keyid: {{ access_key }}
        key: {{ secret_key }}
        instances: "{{ instance }}"
```

`salt.states.loop.`**`until`**(*name*, *m_args=None*, *m_kwargs=None*, *condition=None*, *period=1*, *timeout=60*)

Loop over an execution module until a condition is met.

Parameters

- **name** (*str*) – The name of the execution module
- **m_args** (*list*) – The execution module's positional arguments
- **m_kwargs** (*dict*) – The execution module's keyword arguments
- **condition** (*str*) – The condition which must be met for the loop to break. This should contain `m_ret` which is the return from the execution module.
- **period** (*int or float*) – The number of seconds to wait between executions
- **timeout** (*int or float*) – The timeout in seconds

`salt.states.loop.`**`until_no_eval`**(*name*, *expected*, *compare_operator='eq'*, *timeout=60*, *period=1*, *init_wait=0*, *args=None*, *kwargs=None*)

Generic waiter state that waits for a specific salt function to produce an expected result. The state fails if the function does not exist or raises an exception, or does not produce the expected result within the allotted retries.

Parameters

- **name** (*str*) – Name of the module.function to call
- **expected** – Expected return value. This can be almost anything.
- **compare_operator** (*str*) – Operator to use to compare the result of the module.function call with the expected value. This can be anything present in __salt__ or __utils__. Will be called with 2 args: result, expected.
- **timeout** (*int or float*) – Abort after this amount of seconds (excluding init_wait).
- **period** (*int or float*) – Time (in seconds) to wait between attempts.

- **init_wait** (*int or float*) – Time (in seconds) to wait before trying anything.
- **args** (*list*) – args to pass to the salt module.function.
- **kwargs** (*dict*) – kwargs to pass to the salt module.function.

New in version 3000.

21.19.154 salt.states.lvm

Management of Linux logical volumes

A state module to manage LVMs

```
/dev/sda:
  lvm.pv_present

my_vg:
  lvm.vg_present:
    - devices: /dev/sda

lvroot:
  lvm.lv_present:
    - vgname: my_vg
    - size: 10G
    - stripes: 5
    - stripesize: 8K
```

`salt.states.lvm.`**`lv_absent`**(*name, vgname=None*)

Remove a given existing Logical Volume from a named existing volume group

name The Logical Volume to remove

vgname The name of the Volume Group on which the Logical Volume resides

`salt.states.lvm.`**`lv_present`**(*name, vgname=None, size=None, extents=None, snapshot=None, pv=", thinvolume=False, thinpool=False, force=False, **kwargs*)

Create a new Logical Volume

name The name of the Logical Volume

vgname The name of the Volume Group on which the Logical Volume resides

size The initial size of the Logical Volume

extents The number of logical extents to allocate

snapshot The name of the snapshot

pv The Physical Volume to use

kwargs Any supported options to lvcreate. See `linux_lvm` for more details.

New in version to_complete.

thinvolume Logical Volume is thinly provisioned

thinpool Logical Volume is a thin pool

New in version 2018.3.0.

force Assume yes to all prompts

`salt.states.lvm.`**`pv_absent`**(*name*)

Ensure that a Physical Device is not being used by lvm

name The device name to initialize.

`salt.states.lvm.`**`pv_present`**(*name, **kwargs*)

Set a Physical Device to be used as an LVM Physical Volume

name The device name to initialize.

kwargs Any supported options to pvcreate. See `linux_lvm` for more details.

`salt.states.lvm.`**`vg_absent`**(*name*)

Remove an LVM volume group

name The volume group to remove

`salt.states.lvm.`**`vg_present`**(*name*, *devices=None*, ***kwargs*)

Create an LVM Volume Group

name The Volume Group name to create

devices A list of devices that will be added to the Volume Group

kwargs Any supported options to vgcreate. See `linux_lvm` for more details.

21.19.155 salt.states.lvs_server

Management of LVS (Linux Virtual Server) Real Server

`salt.states.lvs_server.`**`absent`**(*name*, *protocol=None*, *service_address=None*, *server_address=None*)

Ensure the LVS Real Server in specified service is absent.

name The name of the LVS server.

protocol The service protocol(only support `tcp`, `udp` and `fwmark` service).

service_address The LVS service address.

server_address The LVS real server address.

`salt.states.lvs_server.`**`present`**(*name*, *protocol=None*, *service_address=None*, *server_address=None*, *packet_forward_method='dr'*, *weight=1*)

Ensure that the named service is present.

name The LVS server name

protocol The service protocol

service_address The LVS service address

server_address The real server address.

packet_forward_method The LVS packet forwarding method(`dr` for direct routing, `tunnel` for tunneling, `nat` for network access translation).

weight The capacity of a server relative to the others in the pool.

```
lvsrs:
  lvs_server.present:
    - protocol: tcp
    - service_address: 1.1.1.1:80
    - server_address: 192.168.0.11:8080
    - packet_forward_method: dr
    - weight: 10
```

21.19.156 salt.states.lvs_service

Management of LVS (Linux Virtual Server) Service

`salt.states.lvs_service.`**`absent`**(*name*, *protocol=None*, *service_address=None*)

Ensure the LVS service is absent.

name The name of the LVS service

protocol The service protocol

service_address The LVS service address

`salt.states.lvs_service.`**`present`**(*name*, *protocol=None*, *service_address=None*, *scheduler='wlc'*)

Ensure that the named service is present.

name The LVS service name

protocol The service protocol
service_address The LVS service address
scheduler Algorithm for allocating TCP connections and UDP datagrams to real servers.

```
lvstest:
  lvs_service.present:
    - service_address: 1.1.1.1:80
    - protocol: tcp
    - scheduler: rr
```

21.19.157 salt.states.lxc

Manage Linux Containers

`salt.states.lxc.`**`absent`**(*name*, *stop=False*, *path=None*)

Ensure a container is not present, destroying it if present
name Name of the container to destroy
stop stop before destroying default: false

New in version 2015.5.2.

path path to the container parent default: /var/lib/lxc (system default)

New in version 2015.8.0.

```
web01:
  lxc.absent
```

`salt.states.lxc.`**`edited_conf`**(*name*, *lxc_conf=None*, *lxc_conf_unset=None*)

Warning: This state is unsuitable for setting parameters that appear more than once in an LXC config file, or parameters which must appear in a certain order (such as when configuring more than one network interface).

Issue #35523 was opened to track the addition of a suitable replacement or fix.

Edit LXC configuration options

Deprecated since version 2015.5.0.

path path to the container parent default: /var/lib/lxc (system default)

New in version 2015.8.0.

```
setconf:
  lxc.edited_conf:
    - name: ubuntu
    - lxc_conf:
        - network.ipv4.ip: 10.0.3.6
    - lxc_conf_unset:
        - lxc.utsname
```

`salt.states.lxc.`**`frozen`**(*name*, *start=True*, *path=None*)

New in version 2015.5.0.

Ensure that a container is frozen

Note: This state does not enforce the existence of the named container, it just freezes the container if it is running. To ensure that the named container exists, use *`lxc.present`*.

---

name The name of the container

path path to the container parent default: /var/lib/lxc (system default)

New in version 2015.8.0.

start [True] Start container first, if necessary. If `False`, then this state will fail if the container is not running.

```
web01:
  lxc.frozen

web02:
  lxc.frozen:
    - start: False
```

`salt.states.lxc.`**`present`**`(`*name, running=None, clone_from=None, snapshot=False, profile=None, network_profile=None, template=None, options=None, image=None, config=None, fstype=None, size=None, backing=None, vgname=None, lvname=None, thinpool=None, path=None*`)`

Changed in version 2015.8.0: The `lxc.created` state has been renamed to `lxc.present`, and the `lxc.cloned` state has been merged into this state.

Create the named container if it does not exist

name The name of the container to be created

path path to the container parent default: /var/lib/lxc (system default)

New in version 2015.8.0.

running [False]

- If `True`, ensure that the container is running
- If `False`, ensure that the container is stopped
- If `None`, do nothing with regards to the running state of the container

New in version 2015.8.0.

clone_from Create named container as a clone of the specified container

snapshot [False] Use Copy On Write snapshots (LVM). Only supported with `clone_from`.

profile Profile to use in container creation (see the *LXC Tutorial* for more information). Values in a profile will be overridden by the parameters listed below.

network_profile Network Profile to use in container creation (see the *LXC Tutorial* for more information). Values in a profile will be overridden by the parameters listed below.

New in version 2015.5.2.

Container Creation Arguments

template The template to use. For example, `ubuntu` or `fedora`. For a full list of available templates, check out the *`lxc.templates`* function.

Conflicts with the `image` argument.

Note: The `download` template requires the following three parameters to be defined in `options`:

- **dist** - The name of the distribution
- **release** - Release name/version
- **arch** - Architecture of the container

The available images can be listed using the *`lxc.images`* function.

---

options

New in version 2015.5.0.

Template-specific options to pass to the lxc-create command. These correspond to the long options (ones beginning with two dashes) that the template script accepts. For example:

```
web01:
  lxc.present:
    - template: download
    - options:
        dist: centos
        release: 6
        arch: amd64
```

Remember to double-indent the options, due to *how PyYAML works*.

For available template options, refer to the lxc template scripts which are ususally located under `/usr/share/lxc/templates`, or run `lxc-create -t <template> -h`.

image A tar archive to use as the rootfs for the container. Conflicts with the `template` argument.

backing The type of storage to use. Set to `lvm` to use an LVM group. Defaults to filesystem within /var/lib/lxc.

fstype Filesystem type to use on LVM logical volume

size Size of the volume to create. Only applicable if `backing` is set to `lvm`.

vgname [lxc] Name of the LVM volume group in which to create the volume for this container. Only applicable if `backing` is set to `lvm`.

lvname Name of the LVM logical volume in which to create the volume for this container. Only applicable if `backing` is set to `lvm`.

thinpool Name of a pool volume that will be used for thin-provisioning this container. Only applicable if `backing` is set to `lvm`.

`salt.states.lxc.`**`running`**(*name*, *restart=False*, *path=None*)

Changed in version 2015.5.0: The `lxc.started` state has been renamed to `lxc.running`

Ensure that a container is running

Note: This state does not enforce the existence of the named container, it just starts the container if it is not running. To ensure that the named container exists, use *`lxc.present`*.

---

name The name of the container

path path to the container parent default: /var/lib/lxc (system default)

New in version 2015.8.0.

restart [False] Restart container if it is already running

```
web01:
  lxc.running

web02:
  lxc.running:
    - restart: True
```

`salt.states.lxc.`**`set_pass`**(*name*, ***kwargs*)

Deprecated since version 2015.5.0.

This state function has been disabled, as it did not conform to design guidelines. Specifically, due to the fact that *`lxc.set_password`* uses `chpasswd(8)` to set the password, there was no method to make this action idempotent (in other words, the password would be changed every time). This makes this state redundant, since the following state will do the same thing:

```
setpass:
  module.run:
    - name: set_pass
    - m_name: root
    - password: secret
```

`salt.states.lxc.`**`stopped`**(*name*, *kill=False*, *path=None*)

Ensure that a container is stopped

Note: This state does not enforce the existence of the named container, it just stops the container if it running or frozen. To ensure that the named container exists, use `lxc.present`, or use the `lxc.absent` state to ensure that the container does not exist.

---

name The name of the container

path path to the container parent default: /var/lib/lxc (system default)

New in version 2015.8.0.

kill [False] Do not wait for the container to stop, kill all tasks in the container. Older LXC versions will stop containers like this irrespective of this argument.

New in version 2015.5.0.

```
web01:
  lxc.stopped
```

21.19.158 salt.states.mac_assistive module

Allows you to manage assistive access on macOS minions with 10.9+

Install, enable and disable assistive access on macOS minions

```
/usr/bin/osacript:
  assistive.installed:
    - enabled: True
```

`salt.states.mac_assistive.`**`installed`**(*name*, *enabled=True*)

Make sure that we have the given bundle ID or path to command installed in the assistive access panel.

name The bundle ID or path to command

enable Should assistive access be enabled on this application?

21.19.159 salt.states.macdefaults

Writing/reading defaults from a macOS minion

`salt.states.macdefaults.`**`absent`**(*name*, *domain*, *user=None*)

Make sure the defaults value is absent

name The key of the given domain to remove

domain The name of the domain to remove from

user The user to write the defaults to

`salt.states.macdefaults.`**`write`**(*name*, *domain*, *value*, *vtype='string'*, *user=None*)

Write a default to the system

name The key of the given domain to write to

domain The name of the domain to write to

value The value to write to the given key

vtype The type of value to be written, valid types are string, data, int[eger], float, bool[ean], date, array, array-add, dict, dict-add

user The user to write the defaults to

21.19.160 salt.states.mac_keychain module

Installing of certificates to the keychain

Install certificats to the macOS keychain

```
/mnt/test.p12:
  keychain.installed:
    - password: test123
```

`salt.states.mac_keychain.`**`default_keychain`**(*name*, *domain='user'*, *user=None*)

Set the default keychain to use

name The chain in which to use as the default

domain The domain to use valid values are user|system|common|dynamic, the default is user

user The user to run as

`salt.states.mac_keychain.`**`installed`**(*name*, *password*, *keychain='/Library/Keychains/System.keychain'*, ***kwargs*)

Install a p12 certificate file into the macOS keychain

name The certificate to install

password The password for the certificate being installed formatted in the way described for openssl command in the PASS PHRASE ARGUMENTS section

keychain The keychain to install the certificate to, this defaults to /Library/Keychains/System.keychain

allow_any Allow any application to access the imported certificate without warning

keychain_password If your keychain is likely to be locked pass the password and it will be unlocked before running the import

`salt.states.mac_keychain.`**`uninstalled`**(*name*, *password*, *keychain='/Library/Keychains/System.keychain'*, *keychain_password=None*)

Uninstall a p12 certificate file from the macOS keychain

name The certificate to uninstall, this can be a path for a .p12 or the friendly name

password The password for the certificate being installed formatted in the way described for openssl command in the PASS PHRASE ARGUMENTS section

cert_name The friendly name of the certificate, this can be used instead of giving a certificate

keychain The keychain to remove the certificate from, this defaults to /Library/Keychains/System.keychain

keychain_password If your keychain is likely to be locked pass the password and it will be unlocked before running the import

21.19.161 salt.states.macpackage

Installing of mac pkg files

Install any kind of pkg, dmg or app file on macOS:

```
/mnt/test.pkg:
  macpackage.installed:
```

(continues on next page)

(continued from previous page)

```
    - store: True

/mnt/test.dmg:
  macpackage.installed:
    - dmg: True

/mnt/xcode.dmg:
  macpackage.installed:
    - dmg: True
    - app: True
    - target: /Applications/Xcode.app
    - version_check: xcodebuild -version=Xcode 7.1\n.*7B91b
```

`salt.states.macpackage.`**`installed`**(*name, target='LocalSystem', dmg=False, store=False, app=False, mpkg=False, user=None, onlyif=None, unless=None, force=False, allow_untrusted=False, version_check=None*)

Install a Mac OS Package from a pkg or dmg file, if given a dmg file it will first be mounted in a temporary location

name The pkg or dmg file to install

target The location in which to install the package. This can be a path or LocalSystem

dmg Is the given file a dmg file?

store Should the pkg be installed as if it was from the Mac OS Store?

app Is the file a .app? If so then we'll just copy that to /Applications/ or the given target

mpkg Is the file a .mpkg? If so then we'll check all of the .pkg files found are installed

user Name of the user performing the unless or onlyif checks

onlyif A command to run as a check, run the named command only if the command passed to the `onlyif` option returns true

unless A command to run as a check, only run the named command if the command passed to the `unless` option returns false

force Force the package to be installed even if its already been found installed

allow_untrusted Allow the installation of untrusted packages

version_check The command and version that we want to check against, the version number can use regex.

```
version_check: python --version_check=2.7.[0-9]
```

21.19.162 salt.states.mac_xattr module

Allows you to manage extended attributes on files or directories

Install, enable and disable assistive access on macOS minions

```
/path/to/file:
  xattr.exists:
    - attributes:
        - com.file.attr=test
        - com.apple.quarantine=0x00001111
```

`salt.states.mac_xattr.`**`delete`**(*name, attributes*)

Make sure the given attributes are deleted from the file/directory

name The path to the file/directory

attributes The attributes that should be removed from the file/directory, this is accepted as an array.

`salt.states.mac_xattr.`**`exists`**(*name, attributes*)

Make sure the given attributes exist on the file/directory

name The path to the file/directory

attributes The attributes that should exist on the file/directory, this is accepted as an array, with key and value split with an equals sign, if you want to specify a hex value then add 0x to the beginning of the value.

21.19.163 salt.states.makeconf

Management of Gentoo make.conf

A state module to manage Gentoo's `make.conf` file

```
makeopts:
  makeconf.present:
    - value: '-j3'
```

`salt.states.makeconf.`**`absent`**`(`*name*`)`

Verify that the variable is not in the `make.conf`.

name The variable name. This will automatically be converted to upper case since variables in `make.conf` are in upper case

`salt.states.makeconf.`**`present`**`(`*name, value=None, contains=None, excludes=None*`)`

Verify that the variable is in the `make.conf` and has the provided settings. If value is set, contains and excludes will be ignored.

name The variable name. This will automatically be converted to upper case since variables in `make.conf` are in upper case

value Enforce that the value of the variable is set to the provided value

contains Enforce that the value of the variable contains the provided value

excludes Enforce that the value of the variable does not contain the provided value.

21.19.164 salt.states.marathon_app module

Configure Marathon apps via a salt proxy.

```
my_app:
  marathon_app.config:
    - config:
        cmd: "while [ true ] ; do echo 'Hello Marathon' ; sleep 5 ; done"
        cpus: 0.1
        mem: 10
        instances: 3
```

New in version 2015.8.2.

`salt.states.marathon_app.`**`absent`**`(`*name*`)`

Ensure that the marathon app with the given id is not present.

Parameters **`name`** – The app name/id

Returns A standard Salt changes dictionary

`salt.states.marathon_app.`**`config`**`(`*name, config*`)`

Ensure that the marathon app with the given id is present and is configured to match the given config values.

Parameters

- **`name`** – The app name/id
- **`config`** – The configuration to apply (dict)

Returns A standard Salt changes dictionary `salt.states.marathon_app.`**`running`**(*name*, *restart=False*, *force=True*)

Ensure that the marathon app with the given id is present and restart if set.

Parameters

- **`name`** – The app name/id
- **`restart`** – Restart the app
- **`force`** – Override the current deployment

Returns A standard Salt changes dictionary

21.19.165 salt.states.mdadm_raid

Managing software RAID with mdadm

depends mdadm

A state module for creating or destroying software RAID devices.

```
/dev/md0:
  raid.present:
    - level: 5
    - devices:
      - /dev/xvdd
      - /dev/xvde
      - /dev/xvdf
    - chunk: 256
    - run: True
```

`salt.states.mdadm_raid.`**`absent`**(*name*)

Verify that the raid is absent

name The name of raid device to be destroyed

```
/dev/md0:
  raid:
    - absent
```

`salt.states.mdadm_raid.`**`present`**(*name*, *level*, *devices*, ***kwargs*)

Verify that the raid is present

Changed in version 2014.7.0.

name The name of raid device to be created

level The RAID level to use when creating the raid.

devices A list of devices used to build the array.

kwargs Optional arguments to be passed to mdadm.

Example:

```
/dev/md0:
  raid.present:
    - level: 5
    - devices:
      - /dev/xvdd
      - /dev/xvde
      - /dev/xvdf
    - chunk: 256
    - run: True
```

21.19.166 salt.states.memcached

States for Management of Memcached Keys

New in version 2014.1.0.

`salt.states.memcached.`**`absent`**(*name*, *value=None*, *host='127.0.0.1'*, *port=11211*, *time=0*)

Ensure that a memcached key is not present.

name The key

value [None] If specified, only ensure that the key is absent if it matches the specified value.

host The memcached server IP address

port The memcached server port

```
foo:
  memcached.absent

bar:
  memcached.absent:
    - host: 10.0.0.1
```

`salt.states.memcached.`**`managed`**(*name*, *value=None*, *host='127.0.0.1'*, *port=11211*, *time=0*, *min_compress_len=0*)

Manage a memcached key.

name The key to manage

value The value to set for that key

host The memcached server IP address

port The memcached server port

```
foo:
  memcached.managed:
    - value: bar
```

21.19.167 salt.states.modjk

State to control Apache modjk

`salt.states.modjk.`**`worker_activated`**(*name*, *workers=None*, *profile='default'*)

Activate all the workers in the modjk load balancer

Example:

```
loadbalancer:
  modjk.worker_activated:
    - workers:
      - app1
      - app2
```

`salt.states.modjk.`**`worker_disabled`**(*name*, *workers=None*, *profile='default'*)

Disable all the workers in the modjk load balancer

Example:

```
loadbalancer:
  modjk.worker_disabled:
    - workers:
      - app1
      - app2
```

`salt.states.modjk.`**`worker_recover`**(*name*, *workers=None*, *profile='default'*)

Recover all the workers in the modjk load balancer

Example:

```
loadbalancer:
  modjk.worker_recover:
    - workers:
      - app1
      - app2
```

`salt.states.modjk.`**`worker_stopped`**(*name*, *workers=None*, *profile='default'*)

Stop all the workers in the modjk load balancer

Example:

```
loadbalancer:
  modjk.worker_stopped:
    - workers:
      - app1
      - app2
```

21.19.168 salt.states.modjk_worker

Manage modjk workers

Send commands to a **modjk** load balancer via the peer system.

This module can be used with the *prereq* requisite to remove/add the worker from the load balancer before deploying/restarting service.

Mandatory Settings:

- The minion needs to have permission to publish the **modjk.*** functions (see *here* for information on configuring peer publishing permissions)
- The modjk load balancer must be configured as stated in the **modjk** execution module *`documentation`*

`salt.states.modjk_worker.`**`activate`**(*name*, *lbn*, *target*, *profile='default'*, *tgt_type='glob'*)

Changed in version 2017.7.0: The `expr_form` argument has been renamed to `tgt_type`, earlier releases must use `expr_form`.

Activate the named worker from the lbn load balancers at the targeted minions

Example:

```
disable-before-deploy:
  modjk_worker.activate:
    - name: {{ grains['id'] }}
    - lbn: application
    - target: 'roles:balancer'
    - tgt_type: grain
```

`salt.states.modjk_worker.`**`disable`**(*name*, *lbn*, *target*, *profile='default'*, *tgt_type='glob'*)

Changed in version 2017.7.0: The `expr_form` argument has been renamed to `tgt_type`, earlier releases must use `expr_form`.

Disable the named worker from the lbn load balancers at the targeted minions. The worker will get traffic only for current sessions and won't get new ones.

Example:

```
disable-before-deploy:
  modjk_worker.disable:
    - name: {{ grains['id'] }}
    - lbn: application
    - target: 'roles:balancer'
    - tgt_type: grain
```

`salt.states.modjk_worker.`**`stop`**`(`*name, lbn, target, profile='default', tgt_type='glob'*`)`

Changed in version 2017.7.0: The `expr_form` argument has been renamed to `tgt_type`, earlier releases must use `expr_form`.

Stop the named worker from the lbn load balancers at the targeted minions The worker won't get any traffic from the lbn

Example:

```
disable-before-deploy:
  modjk_worker.stop:
    - name: {{ grains['id'] }}
    - lbn: application
    - target: 'roles:balancer'
    - tgt_type: grain
```

21.19.169 salt.states.module

Execution of Salt modules from within states

Note: There are two styles of calling `module.run`. **The legacy style will no longer be available starting in the Sodium release**. To opt-in early to the new style you must add the following to your `/etc/salt/minion` config file:

```
use_superseded:
  - module.run
```

With *module.run* these states allow individual execution module calls to be made via states. Here's a contrived example, to show you how it's done:

```
# New Style
test.random_hash:
  module.run:
    - test.random_hash:
      - size: 42
      - hash_type: sha256

# Legacy Style
test.random_hash:
  module.run:
    - size: 42
    - hash_type: sha256
```

In the new style, the state ID (`test.random_hash`, in this case) is irrelevant when using `module.run`. It could have very well been written:

```
Generate a random hash:
  module.run:
    - test.random_hash:
      - size: 42
      - hash_type: sha256
```

For a simple state like that it's not a big deal, but if the module you're using has certain parameters, things can get cluttered, fast. Using the contrived custom module (stuck in `/srv/salt/_modules/foo.py`, or your configured file_roots):

```
def bar(name, names, fun, state, saltenv):
    return "Name: {name} Names: {names} Fun: {fun} State: {state} Saltenv: {saltenv}".
↪format(**locals())
```

Your legacy state has to look like this:

```
# Legacy style
Unfortunate example:
  module.run:
  - name: foo.bar
  - m_name: Some name
  - m_names:
    - Such names
    - very wow
  - m_state: Arkansas
  - m_fun: Such fun
  - m_saltenv: Salty
```

With the new style it's much cleaner:

```
# New style
Better:
  module.run:
  - foo.bar:
    - name: Some name
    - names:
      - Such names
      - very wow
    - state: Arkansas
    - fun: Such fun
    - saltenv: Salty
```

The new style also allows multiple modules in one state. For instance, you can do this:

```
Do many things:
  module.run:
    - test.random_hash:
      - size: 10
      - hash_type: md5
    # Note the `:` at the end
    - test.true:
    - test.arg:
      - this
      - has
      - args
      - and: kwargs
      - isn't: that neat?
```

(continues on next page)

(continued from previous page)

```
    # Note the `:` at the end, too
    - test.version:
    - test.fib:
      - 4
```

Where in the legacy style you would have had to split your states like this:

```
test.random_hash:
  module.run:
    - size: 10
    - hash_type: md5

test.nop:
  module.run

test.arg:
  module.run:
    - args:
      - this
      - has
      - args
    - kwargs:
        and: kwargs
        isn't: that neat?

test.version:
  module.run
```

Another difference is that in the legacy style, unconsumed arguments to the `module` state were simply passed into the module function being executed:

```
show off module.run with args:
  module.run:
    - name: test.random_hash
    - size: 42
    - hash_type: sha256
```

The new style is much more explicit, with the arguments and keyword arguments being nested under the name of the function:

```
show off module.run with args:
  module.run:
    # Note the lack of `name: `, and trailing `:`
    - test.random_hash:
      - size: 42
      - hash_type: sha256
```

If the function takes `*args`, they can be passed in as well:

```
args and kwargs:
  module.run:
    - test.arg:
      - isn't
      - this
      - fun
      - this: that
      - salt: stack
```

Modern Examples

Here are some other examples using the modern `module.run`:

```
fetch_out_of_band:
  module.run:
    - git.fetch:
      - cwd: /path/to/my/repo
      - user: myuser
      - opts: '--all'
```

A more complex example:

```
eventsviewer:
  module.run:
    - task.create_task:
      - name: events-viewer
      - user_name: System
      - action_type: Execute
      - cmd: 'c:\netops\scripts\events_viewer.bat'
      - trigger_type: 'Daily'
      - start_date: '2017-1-20'
      - start_time: '11:59PM'
```

It is sometimes desirable to trigger a function call after a state is executed, for this the *`module.wait`* state can be used:

```
add example to hosts:
  file.append:
    - name: /etc/hosts
    - text: 203.0.113.13     example.com

# New Style
mine.send:
  module.wait:
    # Again, note the trailing `:`
    - hosts.list_hosts:
    - watch:
      - file: add example to hosts
```

Legacy (Default) Examples

If you're using the legacy `module.run`, due to how the state system works, if a module function accepts an argument called, `name`, then `m_name` must be used to specify that argument, to avoid a collision with the `name` argument.

Here is a list of keywords hidden by the state system, which must be prefixed with `m_`:

- fun
- name
- names
- state
- saltenv

For example:

```
disable_nfs:
  module.run:
    - name: service.disable
    - m_name: nfs
```

Note that some modules read all or some of the arguments from a list of keyword arguments. For example:

```
mine.send:
  module.run:
    - func: network.ip_addrs
    - kwargs:
        interface: eth0
```

```
cloud.create:
  module.run:
    - func: cloud.create
    - provider: test-provider
    - m_names:
      - test-vlad
    - kwargs: {
          ssh_username: 'ubuntu',
          image: 'ami-8d6d9daa',
          securitygroup: 'default',
          size: 'c3.large',
          location: 'ap-northeast-1',
          delvol_on_destroy: 'True'
      }
```

Other modules take the keyword arguments using this style:

```
mac_enable_ssh:
  module.run:
    - name: system.set_remote_login
    - enable: True
```

Another example that creates a recurring task that runs a batch file on a Windows system:

```
eventsviewer:
  module.run:
    - name: task.create_task
    - m_name: 'events-viewer'
    - user_name: System
    - kwargs: {
          action_type: 'Execute',
          cmd: 'c:\netops\scripts\events_viewer.bat',
          trigger_type: 'Daily',
          start_date: '2017-1-20',
          start_time: '11:59PM'
    }
```

`salt.states.module.`**`run`**`(`***kwargs*`)`

Run a single module function or a range of module functions in a batch. Supersedes `module.run` function, which requires `m_` prefix to function-specific parameters.

Parameters

- **`returner`** – Specify a common returner for the whole batch to send the return data
- **`kwargs`** – Pass any arguments needed to execute the function(s)

```
some_id_of_state:
  module.run:
    - network.ip_addrs:
      - interface: eth0
    - cloud.create:
      - names:
        - test-isbm-1
        - test-isbm-2
      - ssh_username: sles
      - image: sles12sp2
      - securitygroup: default
      - size: 'c3.large'
      - location: ap-northeast-1
      - delvol_on_destroy: True
```

Returns

`salt.states.module.`**`wait`**(*name*, ***kwargs*)

Run a single module function only if the watch statement calls it

`name` The module function to execute

`**kwargs` Pass any arguments needed to execute the function

Note: Like the *`cmd.run`* state, this state will return `True` but not actually execute, unless one of the following two things happens:

1. The state has a *watch requisite*, and the state which it is watching changes.
2. Another state has a *watch_in requisite* which references this state, and the state wth the `watch_in` changes.

21.19.170 salt.states.mongodb_database

Management of MongoDB Databases

depends

- pymongo Python module

Only deletion is supported, creation doesn't make sense and can be done using *`mongodb_user.present`*.

`salt.states.mongodb_database.`**`absent`**(*name*, *user=None*, *password=None*, *host=None*, *port=None*, *authdb=None*)

Ensure that the named database is absent. Note that creation doesn't make sense in MongoDB.

name The name of the database to remove

user The user to connect as (must be able to create the user)

password The password of the user

host The host to connect to

port The port to connect to

authdb The database in which to authenticate

21.19.171 salt.states.mongodb_user

Management of MongoDB Users

depends

- pymongo Python module

`salt.states.mongodb_user.`**`absent`**(*name*, *user=None*, *password=None*, *host=None*, *port=None*, *database='admin'*, *authdb=None*)

Ensure that the named user is absent

name The name of the user to remove

user MongoDB user with sufficient privilege to create the user

password Password for the admin user specified by the `user` parameter

host The hostname/IP address of the MongoDB server

port The port on which MongoDB is listening

database The database from which to remove the user specified by the `name` parameter

authdb The database in which to authenticate

`salt.states.mongodb_user.`**`present`**(*name*, *passwd*, *database='admin'*, *user=None*, *password=None*, *host='localhost'*, *port=27017*, *authdb=None*, *roles=None*)

Ensure that the user is present with the specified properties

name The name of the user to manage

passwd The password of the user to manage

user MongoDB user with sufficient privilege to create the user

password Password for the admin user specified with the `user` parameter

host The hostname/IP address of the MongoDB server

port The port on which MongoDB is listening

database The database in which to create the user

Note: If the database doesn't exist, it will be created.

authdb The database in which to authenticate

roles The roles assigned to user specified with the `name` parameter

Example:

```
mongouser-myapp:
  mongodb_user.present:
  - name: myapp
  - passwd: password-of-myapp
  - database: admin
  # Connect as admin:sekrit
  - user: admin
  - password: sekrit
  - roles:
      - readWrite
      - userAdmin
      - dbOwner
```

21.19.172 salt.states.monit

Monit state

Manage monit states

```
monit_enable_service_monitoring:
  monit.monitor:
    - name: service

monit_disable_service_monitoring:
```

(continues on next page)

(continued from previous page)

```
  monit.unmonitor:
    - name: service
```

Note: Use of these states require that the *monit* execution module is available.

`salt.states.monit.`**`monitor`**(*name*)

Get the summary from module monit and try to see if service is being monitored. If not then monitor the service.

`salt.states.monit.`**`unmonitor`**(*name*)

Get the summary from module monit and try to see if service is being monitored. If it is then stop monitoring the service.

21.19.173 salt.states.mount

Mounting of filesystems

Mount any type of mountable filesystem with the mounted function:

```
/mnt/sdb:
  mount.mounted:
    - device: /dev/sdb1
    - fstype: ext4
    - mkmnt: True
    - opts:
      - defaults

/srv/bigdata:
  mount.mounted:
    - device: UUID=066e0200-2867-4ebe-b9e6-f30026ca2314
    - fstype: xfs
    - opts: nobootwait,noatime,nodiratime,nobarrier,logbufs=8
    - dump: 0
    - pass_num: 2
    - persist: True
    - mkmnt: True

/var/lib/bigdata:
  mount.mounted:
    - device: /srv/bigdata
    - fstype: none
    - opts: bind
    - dump: 0
    - pass_num: 0
    - persist: True
    - mkmnt: True
```

`salt.states.mount.`**`fstab_absent`**(*name, fs_file, mount_by=None, config='/etc/fstab'*)

Makes sure that a fstab mount point is absent.

name The name of block device. Can be any valid fs_spec value.

fs_file Mount point (target) for the filesystem.

mount_by Select the final value for fs_spec. Can be [`None`, `device`, `label`, `uuid`, `partlabel`, `partuuid`]. If `None`, the value for fs_spect will be the parameter `name`, in other case will search the correct value based on the device name. For example, for `uuid`, the value for fs_spec will be of type 'UUID=xxx' instead of the device name set in `name`.

config Place where the fstab file lives

`salt.states.mount.`**`fstab_present`**`(`*name, fs_file, fs_vfstype, fs_mntops='defaults', fs_freq=0, fs_passno=0, mount_by=None, config='/etc/fstab', mount=True, match_on='auto', not_change=False*`)`

Makes sure that a fstab mount point is pressent.

name The name of block device. Can be any valid fs_spec value.

fs_file Mount point (target) for the filesystem.

fs_vfstype The type of the filesystem (e.g. ext4, xfs, btrfs, …)

fs_mntops The mount options associated with the filesystem. Default is `defaults`.

fs_freq Field is used by dump to determine which fs need to be dumped. Default is `0`

fs_passno Field is used by fsck to determine the order in which filesystem checks are done at boot time. Default is `0`

mount_by Select the final value for fs_spec. Can be [`None`, `device`, `label`, `uuid`, `partlabel`, `partuuid`]. If `None`, the value for fs_spect will be the parameter `name`, in other case will search the correct value based on the device name. For example, for `uuid`, the value for fs_spec will be of type 'UUID=xxx' instead of the device name set in `name`.

config Place where the fstab file lives. Default is `/etc/fstab`

mount Set if the mount should be mounted immediately. Default is `True`

match_on A name or list of fstab properties on which this state should be applied. Default is `auto`, a special value indicating to guess based on fstype. In general, `auto` matches on name for recognized special devices and device otherwise.

not_change By default, if the entry is found in the fstab file but is different from the expected content (like different options), the entry will be replaced with the correct content. If this parameter is set to `True` and the line is found, the original content will be preserved.

`salt.states.mount.`**`mod_watch`**`(`*name, user=None, **kwargs*`)`

The mounted watcher, called to invoke the watch command.

Note: This state exists to support special handling of the `watch` *requisite*. It should not be called directly.

Parameters for this function should be set by the state being triggered.

name The name of the mount point

`salt.states.mount.`**`mounted`**`(`*name, device, fstype, mkmnt=False, opts='defaults', dump=0, pass_num=0, config='/etc/fstab', persist=True, mount=True, user=None, match_on='auto', device_name_regex=None, extra_mount_invisible_options=None, extra_mount_invisible_keys=None, extra_mount_ignore_fs_keys=None, extra_mount_translate_options=None, hidden_opts=None, **kwargs*`)`

Verify that a device is mounted

name The path to the location where the device is to be mounted

device The device name, typically the device node, such as `/dev/sdb1` or `UUID=066e0200-2867-4ebe-b9e6-f30026ca2314` or `LABEL=DATA`

fstype The filesystem type, this will be `xfs`, `ext2/3/4` in the case of classic filesystems, `fuse` in the case of fuse mounts, and `nfs` in the case of nfs mounts

mkmnt If the mount point is not present then the state will fail, set `mkmnt: True` to create the mount point if it is otherwise not present

opts A list object of options or a comma delimited list

dump The dump value to be passed into the fstab, Default is `0`

pass_num The pass value to be passed into the fstab, Default is `0`

config Set an alternative location for the fstab, Default is `/etc/fstab`

persist Set if the mount should be saved in the fstab, Default is `True`

mount Set if the mount should be mounted immediately, Default is `True`

user The account used to execute the mount; this defaults to the user salt is running as on the minion

match_on A name or list of fstab properties on which this state should be applied. Default is `auto`, a special value indicating to guess based on fstype. In general, `auto` matches on name for recognized special devices and device otherwise.

device_name_regex A list of device exact names or regular expressions which should not force a remount. For example, glusterfs may be mounted with a comma-separated list of servers in fstab, but the /proc/self/mountinfo will show only the first available server.

```
{% set glusterfs_ip_list = ['10.0.0.1', '10.0.0.2', '10.0.0.3'] %}

mount glusterfs volume:
  mount.mounted:
    - name: /mnt/glusterfs_mount_point
    - device: {{ glusterfs_ip_list|join(',') }}:/volume_name
    - fstype: glusterfs
    - opts: _netdev,rw,defaults,direct-io-mode=disable
    - mkmnt: True
    - persist: True
    - dump: 0
    - pass_num: 0
    - device_name_regex:
      - ({{ glusterfs_ip_list|join('|') }}):/volume_name
```

New in version 2016.11.0.

extra_mount_invisible_options A list of extra options that are not visible through the `/proc/self/mountinfo` interface.

If a option is not visible through this interface it will always remount the device. This option extends the builtin `mount_invisible_options` list.

extra_mount_invisible_keys A list of extra key options that are not visible through the `/proc/self/mountinfo` interface.

If a key option is not visible through this interface it will always remount the device. This option extends the builtin `mount_invisible_keys` list.

A good example for a key option is the password option:

```
password=badsecret
```

extra_mount_ignore_fs_keys A dict of filesystem options which should not force a remount. This will update the internal dictionary. The dict should look like this:

```
{
    'ramfs': ['size']
}
```

extra_mount_translate_options A dict of mount options that gets translated when mounted. To prevent a remount add additional options to the default dictionary. This will update the internal dictionary. The dictionary should look like this:

```
{
    'tcp': 'proto=tcp',
    'udp': 'proto=udp'
}
```

hidden_opts A list of mount options that will be ignored when considering a remount as part of the state application

New in version 2015.8.2.

`salt.states.mount.`**`swap`**`(`*name*, *persist=True*, *config='/etc/fstab'*`)`

Activates a swap device

```
/root/swapfile:
  mount.swap
```

Note: `swap` does not currently support LABEL

`salt.states.mount.`**`unmounted`**`(`*name*, *device=None*, *config='/etc/fstab'*, *persist=False*, *user=None*, ***kwargs*`)`

New in version 0.17.0.

Verify that a device is not mounted

name The path to the location where the device is to be unmounted from

device The device to be unmounted. This is optional because the device could be mounted in multiple places.

New in version 2015.5.0.

config Set an alternative location for the fstab, Default is `/etc/fstab`

persist Set if the mount should be purged from the fstab, Default is `False`

user The user to own the mount; this defaults to the user salt is running as on the minion

21.19.174 salt.states.msteams module

Send a message card to Microsoft Teams

This state is useful for sending messages to Teams during state runs.

New in version 2017.7.0.

```
teams-message:
  msteams.post_card:
    - message: 'This state was executed successfully.'
    - hook_url:  https://outlook.office.com/webhook/837
```

The hook_url can be specified in the master or minion configuration like below:

```
msteams:
  hook_url: https://outlook.office.com/webhook/837
```

`salt.states.msteams.`**`post_card`**`(`*name*, *message*, *hook_url=None*, *title=None*, *theme_color=None*`)`

Send a message to a Microsft Teams channel

```
send-msteams-message:
  msteams.post_card:
    - message: 'This state was executed successfully.'
    - hook_url: https://outlook.office.com/webhook/837
```

The following parameters are required:

message The message that is to be sent to the MS Teams channel.

The following parameters are optional:

hook_url The webhook URL given configured in Teams interface, if not specified in the configuration options of master or minion.

title The title for the card posted to the channel

theme_color A hex code for the desired highlight color

21.19.175 salt.states.mysql_database

Management of MySQL databases (schemas)

depends

- MySQLdb Python module

configuration See `salt.modules.mysql` for setup instructions.

The mysql_database module is used to create and manage MySQL databases. Databases can be set as either absent or present.

```
frank:
  mysql_database.present
```

`salt.states.mysql_database.`**`absent`**`(`*name*, ***connection_args*`)`
Ensure that the named database is absent
name The name of the database to remove

`salt.states.mysql_database.`**`present`**`(`*name*, *character_set=None*, *collate=None*, ***connection_args*`)`
Ensure that the named database is present with the specified properties
name The name of the database to manage

21.19.176 salt.states.mysql_grants

Management of MySQL grants (user permissions)

depends

- MySQLdb Python module

configuration See `salt.modules.mysql` for setup instructions.

The mysql_grants module is used to grant and revoke MySQL permissions.

The `name` you pass in purely symbolic and does not have anything to do with the grant itself.

The `database` parameter needs to specify a 'priv_level' in the same specification as defined in the MySQL documentation:

- *
- *.*
- db_name.*
- db_name.tbl_name
- etc...

This state is not able to set password for the permission from the specified host. See `salt.states.mysql_user` for further instructions.

```
frank_exampledb:
   mysql_grants.present:
    - grant: select,insert,update
    - database: exampledb.*
    - user: frank
    - host: localhost

frank_otherdb:
  mysql_grants.present:
    - grant: all privileges
    - database: otherdb.*
    - user: frank

restricted_singletable:
  mysql_grants.present:
    - grant: select
    - database: somedb.sometable
    - user: joe
```

`salt.states.mysql_grants.`**`absent`**`(`*name, grant=None, database=None, user=None, host='localhost', grant_option=False, escape=True, **connection_args*`)`

Ensure that the grant is absent

name The name (key) of the grant to add

grant The grant priv_type (i.e. select,insert,update OR all privileges)

database The database priv_level (i.e. db.tbl OR db.*)

user The user to apply the grant to

host The network/host that the grant should apply to

`salt.states.mysql_grants.`**`present`**`(`*name, grant=None, database=None, user=None, host='localhost', grant_option=False, escape=True, revoke_first=False, ssl_option=False, **connection_args*`)`

Ensure that the grant is present with the specified properties

name The name (key) of the grant to add

grant The grant priv_type (i.e. select,insert,update OR all privileges)

database The database priv_level (i.e. db.tbl OR db.*)

user The user to apply the grant to

host The network/host that the grant should apply to

grant_option Adds the WITH GRANT OPTION to the defined grant. Default is `False`

escape Defines if the database value gets escaped or not. Default is `True`

revoke_first By default, MySQL will not do anything if you issue a command to grant privileges that are more restrictive than what's already in place. This effectively means that you cannot downgrade permissions without first revoking permissions applied to a db.table/user pair first.

To have Salt forcibly revoke perms before applying a new grant, enable the 'revoke_first options.

WARNING: This will *remove* permissions for a database before attempting to apply new permissions. There is no guarantee that new permissions will be applied correctly which can leave your database security in an unknown and potentially dangerous state. Use with caution!

Default is `False`

ssl_option Adds the specified ssl options for the connecting user as requirements for this grant. Value is a list of single-element dicts corresponding to the list of ssl options to use.

Possible key/value pairings for the dicts in the value:

```
- SSL: True
- X509: True
```

(continues on next page)

(continued from previous page)

```
- SUBJECT: <subject>
- ISSUER: <issuer>
- CIPHER: <cipher>
```

The non-boolean ssl options take a string as their values, which should be an appropriate value as specified by the MySQL documentation for these options.

Default is `False` (no ssl options will be used)

21.19.177 salt.states.mysql_query

Execution of MySQL queries

New in version 2014.7.0.

depends

- MySQLdb Python module

configuration See `salt.modules.mysql` for setup instructions.

The mysql_query module is used to execute queries on MySQL databases. Its output may be stored in a file or in a grain.

```
query_id:
  mysql_query.run
    - database: my_database
    - query:    "SELECT * FROM table;"
    - output:   "/tmp/query_id.txt"
```

`salt.states.mysql_query.`**`run`**(*name*, *database*, *query*, *output=None*, *grain=None*, *key=None*, *overwrite=True*, *check_db_exists=True*, ***connection_args*)

Execute an arbitrary query on the specified database

name Used only as an ID

database The name of the database to execute the query on

query The query to execute

output grain: output in a grain other: the file to store results None: output to the result comment (default)

grain: grain to store the output (need output=grain)

key: the specified grain will be treated as a dictionary, the result of this state will be stored under the specified key.

overwrite: The file or grain will be overwritten if it already exists (default)

check_db_exists: The state run will check that the specified database exists (default=True) before running any queries

`salt.states.mysql_query.`**`run_file`**(*name*, *database*, *query_file=None*, *output=None*, *grain=None*, *key=None*, *overwrite=True*, *saltenv=None*, *check_db_exists=True*, ***connection_args*)

Execute an arbitrary query on the specified database

New in version 2017.7.0.

name Used only as an ID

database The name of the database to execute the query_file on

query_file The file of mysql commands to run

output grain: output in a grain other: the file to store results None: output to the result comment (default)

grain: grain to store the output (need output=grain)

key: the specified grain will be treated as a dictionary, the result of this state will be stored under the specified key.

overwrite: The file or grain will be overwritten if it already exists (default)

saltenv: The saltenv to pull the query_file from

check_db_exists: The state run will check that the specified database exists (default=True) before running any queries

21.19.178 salt.states.mysql_user

Management of MySQL users

depends

- MySQLdb Python module

configuration See `salt.modules.mysql` for setup instructions.

```
frank:
  mysql_user.present:
    - host: localhost
    - password: bobcat
```

New in version 0.16.2: Authentication overrides have been added.

The MySQL authentication information specified in the minion config file can be overridden in states using the following arguments: `connection_host`, `connection_port`, `connection_user`, `connection_pass`, `connection_db`, `connection_unix_socket`, `connection_default_file` and `connection_charset`.

```
frank:
  mysql_user.present:
    - host: localhost
    - password: "bob@cat"
    - connection_user: someuser
    - connection_pass: somepass
    - connection_charset: utf8
    - saltenv:
      - LC_ALL: "en_US.utf8"
```

This state is not able to grant permissions for the user. See `salt.states.mysql_grants` for further instructions.

`salt.states.mysql_user.`**`absent`**`(`*name*, *host='localhost'*, ***connection_args*`)`

Ensure that the named user is absent

name The name of the user to remove

`salt.states.mysql_user.`**`present`**`(`*name*, *host='localhost'*, *password=None*, *password_hash=None*, *allow_passwordless=False*, *unix_socket=False*, *password_column=None*, *auth_plugin='mysql_native_password'*, ***connection_args*`)`

Ensure that the named user is present with the specified properties. A passwordless user can be configured by omitting `password` and `password_hash`, and setting `allow_passwordless` to `True`.

name The name of the user to manage

host Host for which this user/password combo applies

password The password to use for this user. Will take precedence over the `password_hash` option if both are specified.

password_hash The password in hashed form. Be sure to quote the password because YAML doesn't like the `*`. A password hash can be obtained from the mysql command-line client like so:

```
mysql> SELECT PASSWORD('mypass');
+-------------------------------------------+
| PASSWORD('mypass')                        |
+-------------------------------------------+
| *6C8989366EAF75BB670AD8EA7A7FC1176A95CEF4 |
+-------------------------------------------+
1 row in set (0.00 sec)
```

allow_passwordless If `True`, then `password` and `password_hash` can be omitted to permit a passwordless login.

New in version 0.16.2.

unix_socket If `True` and allow_passwordless is `True`, the unix_socket auth plugin will be used.

21.19.179 salt.states.netacl

Network ACL

Manage the firewall configuration on the network device managed through NAPALM. The firewall configuration is generated by Capirca.

New in version 2017.7.0.

codeauthor Mircea Ulinic <ping@mirceaulinic.net>

maturity new

depends capirca, napalm

platform unix

Dependencies

Capirca

To install Capirca, execute: `pip install capirca`.

NAPALM

To be able to load configuration on network devices, it requires NAPALM library to be installed: `pip install napalm`. Please check Installation for complete details.

`salt.states.netacl.`**`filter`**(*name, filter_name, filter_options=None, terms=None, prepend=True, pillar_key='acl', pillarenv=None, saltenv=None, merge_pillar=False, only_lower_merge=False, revision_id=None, revision_no=None, revision_date=True, revision_date_format='%Y/%m/%d', test=False, commit=True, debug=False*)

Generate and load the configuration of a policy filter.

filter_name The name of the policy filter.

filter_options Additional filter options. These options are platform-specific. See the complete list of options.

terms Dictionary of terms for this policy filter. If not specified or empty, will try to load the configuration from the pillar, unless `merge_pillar` is set as `False`.

prepend: `True` When `merge_pillar` is set as `True`, the final list of terms generated by merging the terms from `terms` with those defined in the pillar (if any): new terms are prepended at the beginning, while existing ones will preserve the position. To add the new terms at the end of the list, set this argument to `False`.

pillar_key: `acl` The key in the pillar containing the default attributes values. Default: `acl`.

pillarenv Query the master to generate fresh pillar data on the fly, specifically from the requested pillar environment.

saltenv Included only for compatibility with *`pillarenv_from_saltenv`*, and is otherwise ignored.

merge_pillar: `False` Merge `terms` with the corresponding value from the pillar. Default: `False`.

Note: By default this state does not merge, to avoid any unexpected behaviours.

The merge logic depends on the `prepend` argument.

The terms specified through the `terms` argument have higher priority than the pillar.

only_lower_merge: `False` Specify if it should merge only the terms fields. Otherwise it will try to merge also filters fields. Default: `False`. This option requires `merge_pillar`, otherwise it is ignored.

revision_id Add a comment in the filter config having the description for the changes applied.

revision_no The revision count.

revision_date: `True` Boolean flag: display the date when the filter configuration was generated. Default: `True`.

revision_date_format: `%Y/%m/%d` The date format to be used when generating the perforce data. Default: `%Y/%m/%d` (<year>/<month>/<day>).

test: `False` Dry run? If set as `True`, will apply the config, discard and return the changes. Default: `False` and will commit the changes on the device.

commit: `True` Commit? Default: `True`.

debug: `False` Debug mode. Will insert a new key under the output dictionary, as `loaded_config` containing the raw configuration loaded on the device.

CLI Example:

```
salt 'edge01.flw01' state.sls router.acl test=True
```

Output Example:

```
edge01.flw01:
----------
          ID: my-filter
    Function: netacl.filter
      Result: None
     Comment: Testing mode: Configuration discarded.
     Started: 12:24:40.598232
    Duration: 2437.139 ms
     Changes:
              ----------
              diff:
                  ---
                  +++
                  @@ -1228,9 +1228,24 @@
                  !
                  +ipv4 access-list my-filter
                  + 10 remark $Id: my-filter_state $
                  + 20 remark $Revision: 5 $
                  + 30 remark my-other-term
                  + 40 permit tcp any range 5678 5680 any
                  +!
```

(continues on next page)

(continued from previous page)

```
                  +!
              loaded:
                  ! $Id: my-filter_state $
                  ! $Revision: 5 $
                  no ipv6 access-list my-filter
                  ipv6 access-list my-filter
                   remark $Id: my-filter_state $
                   remark $Revision: 5 $
                   remark my-other-term
                   permit tcp any range 5678 5680 any
                  exit

Summary for edge01.flw01
------------
Succeeded: 1 (unchanged=1, changed=1)
Failed:    0
------------
Total states run:     1
Total run time:   2.437 s
```

Pillar example:

```
acl:
  - my-filter:
      options:
        - inet6
      terms:
        - my-term:
            source_port: [1234, 1235]
            protocol:
              - tcp
              - udp
            source_address: 1.2.3.4
            action: reject
        - my-other-term:
            source_port:
              - [5678, 5680]
            protocol: tcp
            action: accept
```

State SLS Example:

```
{%- set filter_name = 'my-filter' -%}
{%- set my_filter_cfg = salt.netacl.get_filter_pillar(filter_name, pillar_key=
↪'firewall') -%}
my_first_filter_state:
  netacl.filter:
    - filter_name: {{ filter_name }}
    - options: {{ my_filter_cfg['options'] | json }}
    - terms: {{ my_filter_cfg['terms'] | json }}
    - revision_date: false
    - revision_no: 5
    - debug: true
```

Or:

```
my_first_filter_state:
  netacl.filter:
    - filter_name: my-filter
    - merge_pillar: true
    - pillar_key: firewall
    - revision_date: false
    - revision_no: 5
    - debug: true
```

In the example above, as `inet6` has been specified in the `filter_options`, the configuration chunk referring to `my-term` has been ignored as it referred to IPv4 only (from `source_address` field).

Note: The first method allows the user to eventually apply complex manipulation and / or retrieve the data from external services before passing the data to the state. The second one is more straightforward, for less complex cases when loading the data directly from the pillar is sufficient.

Note: When passing retrieved pillar data into the state file, it is strongly recommended to use the json serializer explicitly (`` | json``), instead of relying on the default Python serializer.

`salt.states.netacl.`**`managed`**(*name, filters=None, prepend=True, pillar_key='acl', pillarenv=None, saltenv=None, merge_pillar=False, only_lower_merge=False, revision_id=None, revision_no=None, revision_date=True, revision_date_format='%Y/%m/%d', test=False, commit=True, debug=False*)

Manage the whole firewall configuration.

filters Dictionary of filters for this policy. If not specified or empty, will try to load the configuration from the pillar, unless `merge_pillar` is set as `False`.

prepend: **`True`** When `merge_pillar` is set as `True`, the final list of filters generated by merging the filters from `filters` with those defined in the pillar (if any): new filters are prepended at the beginning, while existing ones will preserve the position. To add the new filters at the end of the list, set this argument to `False`.

pillar_key: **`acl`** The key in the pillar containing the default attributes values. Default: `acl`.

pillarenv Query the master to generate fresh pillar data on the fly, specifically from the requested pillar environment.

saltenv Included only for compatibility with *`pillarenv_from_saltenv`*, and is otherwise ignored.

merge_pillar: **`False`** Merge the `filters` will the corresponding values from the pillar. Default: `False`.

> **Note:** By default this state does not merge, to avoid any unexpected behaviours.
>
> The merge logic depends on the `prepend` argument.
>
> The filters specified through the `filters` argument have higher priority than the pillar.

only_lower_merge: **`False`** Specify if it should merge only the filters and terms fields. Otherwise it will try to merge everything at the policy level. Default: `False`. This option requires `merge_pillar`, otherwise it is ignored.

test: **`False`** Dry run? If set as `True`, will apply the config, discard and return the changes. Default: `False` and will commit the changes on the device.

revision_id Add a comment in the policy config having the description for the changes applied.

revision_no The revision count.

revision_date: **`True`** Boolean flag: display the date when the policy configuration was generated. Default: `True`.

revision_date_format: **`%Y/%m/%d`** The date format to be used when generating the perforce data. Default: `%Y/%m/%d` (<year>/<month>/<day>).

commit: **`True`** Commit? Default: `True`.

debug: **`False`** Debug mode. Will insert a new key under the output dictionary, as `loaded_config` containing the raw configuration loaded on the device.

CLI Example:

```
salt 'edge01.bjm01' state.sls router.acl test=True
```

Output Example:

```
edge01.bjm01:
-------------
              ID: netacl_example
        Function: netacl.managed
          Result: None
         Comment: Testing mode: Configuration discarded.
         Started: 12:03:24.807023
        Duration: 5569.453 ms
         Changes:
                  ----------
                  diff:
                      [edit firewall]
                      +    family inet {
                      +        /*
                      +         ** $Id: netacl_example $
                      +         ** $Date: 2017/07/03 $
                      +         ** $Revision: 2 $
                      +         **
                      +         */
                      +        filter my-filter {
                      +            interface-specific;
                      +            term my-term {
                      +                from {
                      +                    source-address {
                      +                        1.2.3.4/32;
                      +                    }
                      +                    protocol [ tcp udp ];
                      +                    source-port [ 1234 1235 ];
                      +                }
                      +                then {
                      +                    reject;
                      +                }
                      +            }
                      +            term my-other-term {
                      +                from {
                      +                    protocol tcp;
                      +                    source-port 5678-5680;
                      +                }
                      +                then accept;
                      +            }
                      +        }
                      +        /*
                      +         ** $Id: netacl_example $
                      +         ** $Date: 2017/07/03 $
                      +         ** $Revision: 2 $
                      +         **
                      +         */
```

(continues on next page)

(continued from previous page)

```
                      +        filter block-icmp {
                      +            interface-specific;
                      +            term first-term {
                      +                from {
                      +                    protocol icmp;
                      +                }
                      +                then {
                      +                    reject;
                      +                }
                      +            }
                      +        }
                      +    }
                  loaded:
                      firewall {
                          family inet {
                              replace:
                              /*
                              ** $Id: netacl_example $
                              ** $Date: 2017/07/03 $
                              ** $Revision: 2 $
                              **
                              */
                              filter my-filter {
                                  interface-specific;
                                  term my-term {
                                      from {
                                          source-address {
                                              1.2.3.4/32;
                                          }
                                          protocol [ tcp udp ];
                                          source-port [ 1234 1235 ];
                                      }
                                      then {
                                          reject;
                                      }
                                  }
                                  term my-other-term {
                                      from {
                                          protocol tcp;
                                          source-port 5678-5680;
                                      }
                                      then accept;
                                  }
                              }
                          }
                      }
                      firewall {
                          family inet {
                              replace:
                              /*
                              ** $Id: netacl_example $
                              ** $Date: 2017/07/03 $
                              ** $Revision: 2 $
                              **
                              */
                              filter block-icmp {
```

(continues on next page)

(continued from previous page)

```
                                  interface-specific;
                                  term first-term {
                                      from {
                                          protocol icmp;
                                      }
                                      then {
                                          reject;
                                      }
                                  }
                              }
                          }
                      }

    Summary for edge01.bjm01
    ------------
    Succeeded: 1 (unchanged=1, changed=1)
    Failed:    0
    ------------
    Total states run:     1
    Total run time:   5.569 s
```

The policy configuration has been loaded from the pillar, having the following structure:

```
firewall:
  - my-filter:
      terms:
        - my-term:
            source_port: [1234, 1235]
            protocol:
              - tcp
              - udp
            source_address: 1.2.3.4
            action: reject
        - my-other-term:
            source_port:
              - [5678, 5680]
            protocol: tcp
            action: accept
  - block-icmp:
      terms:
        - first-term:
            protocol:
              - icmp
            action: reject
```

Example SLS file:

```
{%- set fw_filters = pillar.get('firewall', {}) -%}
netacl_example:
  netacl.managed:
    - filters: {{ fw_filters | json }}
    - revision_no: 2
    - debug: true
```

Or:

```
netacl_example:
  netacl.managed:
    - pillar_key: firewall
    - merge_pillar: true
    - revision_no: 2
    - debug: true
```

Note: The first method allows the user to eventually apply complex manipulation and / or retrieve the data from external services before passing the data to the state. The second one is more straightforward, for less complex cases when loading the data directly from the pillar is sufficient.

Note: When passing retrieved pillar data into the state file, it is strongly recommended to use the json serializer explicitly (`` | json``), instead of relying on the default Python serializer.

`salt.states.netacl.`**`term`**`(`*name, filter_name, term_name, filter_options=None, pillar_key='acl', pillarenv=None, saltenv=None, merge_pillar=False, revision_id=None, revision_no=None, revision_date=True, revision_date_format='%Y/%m/%d', test=False, commit=True, debug=False, source_service=None, destination_service=None, **term_fields*`)`

Manage the configuration of a specific policy term.

filter_name The name of the policy filter.

term_name The name of the term.

filter_options Additional filter options. These options are platform-specific. See the complete list of options.

pillar_key: **`acl`** The key in the pillar containing the default attributes values. Default: `acl`.

pillarenv Query the master to generate fresh pillar data on the fly, specifically from the requested pillar environment.

saltenv Included only for compatibility with *`pillarenv_from_saltenv`*, and is otherwise ignored.

merge_pillar: **`False`** Merge the CLI variables with the pillar. Default: `False`.

The properties specified through the state arguments have higher priority than the pillar.

revision_id Add a comment in the term config having the description for the changes applied.

revision_no The revision count.

revision_date: **`True`** Boolean flag: display the date when the term configuration was generated. Default: `True`.

revision_date_format: **`%Y/%m/%d`** The date format to be used when generating the perforce data. Default: `%Y/%m/%d` (<year>/<month>/<day>).

test: **`False`** Dry run? If set as `True`, will apply the config, discard and return the changes. Default: `False` and will commit the changes on the device.

commit: **`True`** Commit? Default: `True`.

debug: **`False`** Debug mode. Will insert a new key under the output dictionary, as `loaded_config` containing the raw configuration loaded on the device.

source_service A special service to choose from. This is a helper so the user is able to select a source just using the name, instead of specifying a source_port and protocol.

As this module is available on Unix platforms only, it reads the IANA port assignment from /etc/services.

If the user requires additional shortcuts to be referenced, they can add entries under /etc/services, which can be managed using the *`file state`*.

destination_service A special service to choose from. This is a helper so the user is able to select a source just using the name, instead of specifying a destination_port and protocol. Allows the same options as `source_service`.

term_fields Term attributes. To see what fields are supported, please consult the list of supported keywords.

Some platforms have few other optional keywords.

Note: The following fields are accepted:

- action
- address
- address_exclude
- comment
- counter
- expiration
- destination_address
- destination_address_exclude
- destination_port
- destination_prefix
- forwarding_class
- forwarding_class_except
- logging
- log_name
- loss_priority
- option
- policer
- port
- precedence
- principals
- protocol
- protocol_except
- qos
- pan_application
- routing_instance
- source_address
- source_address_exclude
- source_port
- source_prefix
- verbatim
- packet_length
- fragment_offset
- hop_limit
- icmp_type
- ether_type
- traffic_class_count
- traffic_type
- translated
- dscp_set
- dscp_match
- dscp_except
- next_ip
- flexible_match_range
- source_prefix_except
- destination_prefix_except
- vpn
- source_tag
- destination_tag
- source_interface
- destination_interface
- flattened

- flattened_addr
- flattened_saddr
- flattened_daddr
- priority

Note: The following fields can be also a single value and a list of values:

- action
- address
- address_exclude
- comment
- destination_address
- destination_address_exclude
- destination_port
- destination_prefix
- forwarding_class
- forwarding_class_except
- logging
- option
- port
- precedence
- principals
- protocol
- protocol_except
- pan_application
- source_address
- source_address_exclude
- source_port
- source_prefix
- verbatim
- icmp_type
- ether_type
- traffic_type
- dscp_match
- dscp_except
- flexible_match_range
- source_prefix_except
- destination_prefix_except
- source_tag
- destination_tag
- source_service
- destination_service

Example: `destination_address` can be either defined as:

```
destination_address: 172.17.17.1/24
```

or as a list of destination IP addresses:

```
destination_address:
    - 172.17.17.1/24
    - 172.17.19.1/24
```

or a list of services to be matched:

```
source_service:
    - ntp
    - snmp
    - ldap
    - bgpd
```

Note: The port fields `source_port` and `destination_port` can be used as above to select either a single value, either a list of values, but also they can select port ranges. Example:

```
source_port:
    - [1000, 2000]
    - [3000, 4000]
```

With the configuration above, the user is able to select the 1000-2000 and 3000-4000 source port ranges.

CLI Example:

```
salt 'edge01.bjm01' state.sls router.acl
```

Output Example:

```
edge01.bjm01:
----------
          ID: update_icmp_first_term
    Function: netacl.term
      Result: None
     Comment: Testing mode: Configuration discarded.
     Started: 12:49:09.174179
    Duration: 5751.882 ms
     Changes:
              ----------
              diff:
                  [edit firewall]
                  +    family inet {
                  +        /*
                  +         ** $Id: update_icmp_first_term $
                  +         ** $Date: 2017/02/30 $
                  +         **
                  +         */
                  +        filter block-icmp {
                  +            term first-term {
                  +                from {
                  +                    protocol icmp;
                  +                }
                  +                then {
                  +                    reject;
                  +                }
                  +            }
                  +        }
                  +    }

Summary for edge01.bjm01
------------
Succeeded: 1 (unchanged=1, changed=1)
```

(continues on next page)

(continued from previous page)

```
Failed:    0
------------
Total states run:     1
Total run time:   5.752 s
```

Pillar example:

```
firewall:
  - block-icmp:
      terms:
        - first-term:
            protocol:
              - icmp
            action: reject
```

State SLS example:

```
{%- set filter_name = 'block-icmp' -%}
{%- set term_name = 'first-term' -%}
{%- set my_term_cfg = salt.netacl.get_term_pillar(filter_name, term_name) -%}

update_icmp_first_term:
  netacl.term:
    - filter_name: {{ filter_name }}
    - filter_options:
        - not-interface-specific
    - term_name: {{ term_name }}
    - {{ my_term_cfg | json }}
```

Or directly referencing the pillar keys:

```
update_icmp_first_term:
  netacl.term:
    - filter_name: block-icmp
    - filter_options:
        - not-interface-specific
    - term_name: first-term
    - merge_pillar: true
```

Note: The first method allows the user to eventually apply complex manipulation and / or retrieve the data from external services before passing the data to the state. The second one is more straightforward, for less complex cases when loading the data directly from the pillar is sufficient.

Note: When passing retrieved pillar data into the state file, it is strongly recommended to use the json serializer explicitly (`` | json``), instead of relying on the default Python serializer.

21.19.180 salt.states.netconfig

Network Config

Manage the configuration on a network device given a specific static config or template.

codeauthor Mircea Ulinic <ping@mirceaulinic.net> & Jerome Fleury <jf@cloudflare.com>

maturity new

depends napalm

platform unix

Dependencies

- *NAPALM proxy minion*
- *Network-related basic features execution module*

New in version 2017.7.0.

`salt.states.netconfig.`**`commit_cancelled`**`(`*name*`)`

New in version 2019.2.0.

Cancel a commit scheduled to be executed via the `commit_in` and `commit_at` arguments from the *`net.load_template`* or *`net.load_config`* execution functions. The commit ID is displayed when the commit is scheduled via the functions named above.

State SLS Example:

```
'20180726083540640360':
  netconfig.commit_cancelled
```

`salt.states.netconfig.`**`commit_confirmed`**`(`*name*`)`

New in version 2019.2.0.

Confirm a commit scheduled to be reverted via the `revert_in` and `revert_at` arguments from the *`net.load_template`* or *`net.load_config`* execution functions. The commit ID is displayed when the commit confirmed is scheduled via the functions named above.

State SLS Example:

```
'20180726083540640360':
  netconfig.commit_confirmed
```

`salt.states.netconfig.`**`managed`**`(`*name, template_name=None, template_source=None, template_hash=None, template_hash_name=None, saltenv='base', template_engine='jinja', skip_verify=False, context=None, defaults=None, test=False, commit=True, debug=False, replace=False, commit_in=None, commit_at=None, revert_in=None, revert_at=None, **template_vars*`)`

Manages the configuration on network devices.

By default this state will commit the changes on the device. If there are no changes required, it does not commit and the field `already_configured` from the output dictionary will be set as `True` to notify that.

To avoid committing the configuration, set the argument `test` to `True` (or via the CLI argument `test=True`) and will discard (dry run).

To preserve the changes, set `commit` to `False` (either as CLI argument, either as state parameter). However, this is recommended to be used only in exceptional cases when there are applied few consecutive states and/or configuration changes. Otherwise the user might forget that the config DB is locked and the candidate config buffer is not cleared/merged in the running config.

To replace the config, set `replace` to `True`. This option is recommended to be used with caution!

template_name Identifies path to the template source. The template can be either stored on the local machine, either remotely. The recommended location is under the `file_roots` as specified in the master config file. For example, let's suppose the `file_roots` is configured as:

```
file_roots:
  base:
     - /etc/salt/states
```

Placing the template under `/etc/salt/states/templates/example.jinja`, it can be used as `salt://templates/example.jinja`. Alternatively, for local files, the user can specify the absolute path. If remotely, the source can be retrieved via `http`, `https` or `ftp`.

Examples:

- `salt://my_template.jinja`
- `/absolute/path/to/my_template.jinja`
- `http://example.com/template.cheetah`
- `https:/example.com/template.mako`
- `ftp://example.com/template.py`

Changed in version 2019.2.0: This argument can now support a list of templates to be rendered. The resulting configuration text is loaded at once, as a single configuration chunk.

template_source: **None** Inline config template to be rendered and loaded on the device.

template_hash: **None** Hash of the template file. Format: `{hash_type: 'md5', 'hsum': <md5sum>}`

template_hash_name: **None** When `template_hash` refers to a remote file, this specifies the filename to look for in that file.

saltenv: **base** Specifies the template environment. This will influence the relative imports inside the templates.

template_engine: **jinja** The following templates engines are supported:

- *`cheetah`*
- *`genshi`*
- *`jinja`*
- *`mako`*
- *`py`*
- *`wempy`*

skip_verify: **False** If `True`, hash verification of remote file sources (`http://`, `https://`, `ftp://`) will be skipped, and the `source_hash` argument will be ignored.

Changed in version 2017.7.1.

test: **False** Dry run? If set to `True`, will apply the config, discard and return the changes. Default: `False` (will commit the changes on the device).

commit: **True** Commit? Default: `True`.

debug: **False** Debug mode. Will insert a new key under the output dictionary, as `loaded_config` containing the raw result after the template was rendered.

Note: This argument cannot be used directly on the command line. Instead, it can be passed through the `pillar` variable when executing either of the *`state.sls`* or `state.apply` (see below for an example).

---

commit_in: **`None`** Commit the changes in a specific number of minutes / hours. Example of accepted formats: `5` (commit in 5 minutes), `2m` (commit in 2 minutes), `1h` (commit the changes in 1 hour)`, `5h30m` (commit the changes in 5 hours and 30 minutes).

Note: This feature works on any platforms, as it does not rely on the native features of the network operating system.

Note: After the command is executed and the `diff` is not satisfactory, or for any other reasons you have to discard the commit, you are able to do so using the *`net.cancel_commit`* execution function, using the commit ID returned by this function.

Warning: Using this feature, Salt will load the exact configuration you expect, however the diff may change in time (i.e., if an user applies a manual configuration change, or a different process or command changes the configuration in the meanwhile).

New in version 2019.2.0.

commit_at: **`None`** Commit the changes at a specific time. Example of accepted formats: `1am` (will commit the changes at the next 1AM), `13:20` (will commit at 13:20), `1:20am`, etc.

Note: This feature works on any platforms, as it does not rely on the native features of the network operating system.

Note: After the command is executed and the `diff` is not satisfactory, or for any other reasons you have to discard the commit, you are able to do so using the *`net.cancel_commit`* execution function, using the commit ID returned by this function.

Warning: Using this feature, Salt will load the exact configuration you expect, however the diff may change in time (i.e., if an user applies a manual configuration change, or a different process or command changes the configuration in the meanwhile).

New in version 2019.2.0.

revert_in: **`None`** Commit and revert the changes in a specific number of minutes / hours. Example of accepted formats: `5` (revert in 5 minutes), `2m` (revert in 2 minutes), `1h` (revert the changes in 1 hour)`, `5h30m` (revert the changes in 5 hours and 30 minutes).

Note: To confirm the commit, and prevent reverting the changes, you will have to execute the *`net.confirm_commit`* function, using the commit ID returned by this function.

Warning: This works on any platform, regardless if they have or don't have native capabilities to confirming a commit. However, please be *very* cautious when using this feature: on Junos (as it is the only NAPALM core platform supporting this natively) it executes a commit confirmed as you would do from the command line. All the other platforms don't have this capability natively, therefore the revert is done via Salt. That means, your device needs to be reachable at the moment when Salt will attempt to revert your changes. Be cautious when pushing configuration changes that would prevent you reach the device.

Similarly, if an user or a different process apply other configuration changes in the meanwhile (between the moment you commit and till the changes are reverted), these changes would be equally reverted, as Salt cannot be aware of them.

New in version 2019.2.0.

revert_at: `None` Commit and revert the changes at a specific time. Example of accepted formats: `1am` (will commit and revert the changes at the next 1AM), `13:20` (will commit and revert at 13:20), `1:20am`, etc.

Note: To confirm the commit, and prevent reverting the changes, you will have to execute the *`net.confirm_commit`* function, using the commit ID returned by this function.

Warning: This works on any platform, regardless if they have or don't have native capabilities to confirming a commit. However, please be *very* cautious when using this feature: on Junos (as it is the only NAPALM core platform supporting this natively) it executes a commit confirmed as you would do from the command line. All the other platforms don't have this capability natively, therefore the revert is done via Salt. That means, your device needs to be reachable at the moment when Salt will attempt to revert your changes. Be cautious when pushing configuration changes that would prevent you reach the device.

Similarly, if an user or a different process apply other configuration changes in the meanwhile (between the moment you commit and till the changes are reverted), these changes would be equally reverted, as Salt cannot be aware of them.

New in version 2019.2.0.

replace: False Load and replace the configuration. Default: `False` (will apply load merge).

context: None Overrides default context variables passed to the template.

New in version 2019.2.0.

defaults: None Default variables/context passed to the template.

template_vars Dictionary with the arguments/context to be used when the template is rendered. Do not explicitly specify this argument. This represents any other variable that will be sent to the template rendering system. Please see an example below! In both `ntp_peers_example_using_pillar` and `ntp_peers_example`, `peers` is sent as template variable.

Note: It is more recommended to use the `context` argument instead, to avoid any conflicts with other arguments.

SLS Example (e.g.: under salt://router/config.sls) :

```
whole_config_example:
    netconfig.managed:
        - template_name: salt://path/to/complete_config.jinja
        - debug: True
        - replace: True
bgp_config_example:
    netconfig.managed:
        - template_name: /absolute/path/to/bgp_neighbors.mako
        - template_engine: mako
prefix_lists_example:
    netconfig.managed:
        - template_name: prefix_lists.cheetah
        - debug: True
        - template_engine: cheetah
ntp_peers_example:
    netconfig.managed:
        - template_name: http://bit.ly/2gKOj20
```

(continues on next page)

(continued from previous page)

```
        - skip_verify: False
        - debug: True
        - peers:
            - 192.168.0.1
            - 192.168.0.1
ntp_peers_example_using_pillar:
    netconfig.managed:
        - template_name: http://bit.ly/2gKOj20
        - peers: {{ pillar.get('ntp.peers', []) }}
```

Multi template example:

```
hostname_and_ntp:
  netconfig.managed:
    - template_name:
        - https://bit.ly/2OhSgqP
        - https://bit.ly/2M6C4Lx
        - https://bit.ly/2OIWVTs
    - debug: true
    - context:
        hostname: {{ opts.id }}
        servers:
          - 172.17.17.1
          - 172.17.17.2
        peers:
          - 192.168.0.1
          - 192.168.0.2
```

Usage examples:

```
$ sudo salt 'juniper.device' state.sls router.config test=True

$ sudo salt -N all-routers state.sls router.config pillar="{'debug': True}"
```

`router.config` depends on the location of the SLS file (see above). Running this command, will be executed all five steps from above. These examples above are not meant to be used in a production environment, their sole purpose is to provide usage examples.

Output example:

```
$ sudo salt 'juniper.device' state.sls router.config test=True
juniper.device:
----------
          ID: ntp_peers_example_using_pillar
    Function: netconfig.managed
      Result: None
     Comment: Testing mode: Configuration discarded.
     Started: 12:01:40.744535
    Duration: 8755.788 ms
     Changes:
              ----------
              diff:
                  [edit system ntp]
                       peer 192.168.0.1 { ... }
                  +    peer 172.17.17.1;
                  +    peer 172.17.17.3;
```

(continues on next page)

(continued from previous page)

```
Summary for juniper.device
------------
Succeeded: 1 (unchanged=1, changed=1)
Failed:    0
------------
Total states run:     1
Total run time:   8.756 s
```

Raw output example (useful when the output is reused in other states/execution modules):

```
$ sudo salt --out=pprint 'juniper.device' state.sls router.config test=True
↪debug=True
```

```
{
    'juniper.device': {
        'netconfig_|-ntp_peers_example_using_pillar_|-ntp_peers_example_using_
↪pillar_|-managed': {
            '__id__': 'ntp_peers_example_using_pillar',
            '__run_num__': 0,
            'already_configured': False,
            'changes': {
                'diff': '[edit system ntp]   peer 192.168.0.1 { ... }+   peer 172.
↪17.17.1;+   peer 172.17.17.3;'
            },
            'comment': 'Testing mode: Configuration discarded.',
            'duration': 7400.759,
            'loaded_config': 'system {  ntp {  peer 172.17.17.1;  peer 172.17.17.
↪3; } }',
            'name': 'ntp_peers_example_using_pillar',
            'result': None,
            'start_time': '12:09:09.811445'
        }
    }
}
```

`salt.states.netconfig.`**`replace_pattern`**(*name, pattern, repl, count=0, flags=8, bufsize=1, append_if_not_found=False, prepend_if_not_found=False, not_found_content=None, search_only=False, show_changes=True, backslash_literal=False, source='running', path=None, test=False, replace=True, debug=False, commit=True*)

New in version 2019.2.0.

Replace occurrences of a pattern in the configuration source. If `show_changes` is `True`, then a diff of what changed will be returned, otherwise a `True` will be returned when changes are made, and `False` when no changes are made. This is a pure Python implementation that wraps Python's `sub()`.

pattern A regular expression, to be matched using Python's `search()`.

repl The replacement text.

count: `0` Maximum number of pattern occurrences to be replaced. If count is a positive integer `n`, only `n` occurrences will be replaced, otherwise all occurrences will be replaced.

flags (list or int): `8` A list of flags defined in the `re` module documentation from the Python standard library. Each list item should be a string that will correlate to the human-friendly flag name. E.g., `['IGNORECASE', 'MULTILINE']`. Optionally, `flags` may be an int, with a value corresponding to the XOR (`|`) of all the desired flags. Defaults to 8 (which supports 'MULTILINE').

bufsize (int or str): `1` How much of the configuration to buffer into memory at once. The default value `1` processes one line at a time. The special value `file` may be specified which will read the entire file into memory before processing.

append_if_not_found: **`False`** If set to `True`, and pattern is not found, then the content will be appended to the file.

prepend_if_not_found: **`False`** If set to `True` and pattern is not found, then the content will be prepended to the file.

not_found_content Content to use for append/prepend if not found. If None (default), uses `repl`. Useful when `repl` uses references to group in pattern.

search_only: **`False`** If set to true, this no changes will be performed on the file, and this function will simply return `True` if the pattern was matched, and `False` if not.

show_changes: **`True`** If `True`, return a diff of changes made. Otherwise, return `True` if changes were made, and `False` if not.

backslash_literal: **`False`** Interpret backslashes as literal backslashes for the repl and not escape characters. This will help when using append/prepend so that the backslashes are not interpreted for the repl on the second run of the state.

source: **`running`** The configuration source. Choose from: `running`, `candidate`, or `startup`. Default: `running`.

path Save the temporary configuration to a specific path, then read from there.

test: **`False`** Dry run? If set as `True`, will apply the config, discard and return the changes. Default: `False` and will commit the changes on the device.

commit: **`True`** Commit the configuration changes? Default: `True`.

debug: **`False`** Debug mode. Will insert a new key in the output dictionary, as `loaded_config` containing the raw configuration loaded on the device.

replace: **`True`** Load and replace the configuration. Default: `True`.

If an equal sign (=) appears in an argument to a Salt command it is interpreted as a keyword argument in the format `key=val`. That processing can be bypassed in order to pass an equal sign through to the remote shell command by manually specifying the kwarg:

State SLS Example:

```
update_policy_name:
  netconfig.replace_pattern:
    - pattern: OLD-POLICY-NAME
    - repl: new-policy-name
    - debug: true
```

`salt.states.netconfig.`**`saved`**(*name, source='running', user=None, group=None, mode=None, attrs=None, makedirs=False, dir_mode=None, replace=True, backup='', show_changes=True, create=True, tmp_dir='', tmp_ext='', encoding=None, encoding_errors='strict', allow_empty=False, follow_symlinks=True, check_cmd=None, win_owner=None, win_perms=None, win_deny_perms=None, win_inheritance=True, win_perms_reset=False, **kwargs*)

New in version 2019.2.0.

Save the configuration to a file on the local file system.

name Absolute path to file where to save the configuration. To push the files to the Master, use *`cp.push`* Execution function.

source: **`running`** The configuration source. Choose from: `running`, `candidate`, `startup`. Default: `running`.

user The user to own the file, this defaults to the user salt is running as on the minion

group The group ownership set for the file, this defaults to the group salt is running as on the minion. On Windows, this is ignored

mode The permissions to set on this file, e.g. `644`, `0775`, or `4664`. The default mode for new files and directories corresponds to the umask of the salt process. The mode of existing files and directories will only be changed if `mode` is specified.

Note: This option is **not** supported on Windows.

attrs The attributes to have on this file, e.g. `a`, `i`. The attributes can be any or a combination of the following characters: `aAcCdDeijPsStTu`.

Note: This option is **not** supported on Windows.

makedirs: **`False`** If set to `True`, then the parent directories will be created to facilitate the creation of the named file. If `False`, and the parent directory of the destination file doesn't exist, the state will fail.

dir_mode If directories are to be created, passing this option specifies the permissions for those directories. If this is not set, directories will be assigned permissions by adding the execute bit to the mode of the files.

The default mode for new files and directories corresponds umask of salt process. For existing files and directories it's not enforced.

replace: **`True`** If set to `False` and the file already exists, the file will not be modified even if changes would otherwise be made. Permissions and ownership will still be enforced, however.

backup Overrides the default backup mode for this specific file. See *backup_mode documentation* for more details.

show_changes: **`True`** Output a unified diff of the old file and the new file. If `False` return a boolean if any changes were made.

create: **`True`** If set to `False`, then the file will only be managed if the file already exists on the system.

encoding If specified, then the specified encoding will be used. Otherwise, the file will be encoded using the system locale (usually UTF-8). See https://docs.python.org/3/library/codecs.html#standard-encodings for the list of available encodings.

encoding_errors: **`'strict'`** Error encoding scheme. Default is ```'strict'```. See https://docs.python.org/2/library/codecs.html#codec-base-classes for the list of available schemes.

allow_empty: **`True`** If set to `False`, then the state will fail if the contents specified by `contents_pillar` or `contents_grains` are empty.

follow_symlinks: **`True`** If the desired path is a symlink follow it and make changes to the file to which the symlink points.

check_cmd The specified command will be run with an appended argument of a *temporary* file containing the new managed contents. If the command exits with a zero status the new managed contents will be written to the managed destination. If the command exits with a nonzero exit code, the state will fail and no changes will be made to the file.

tmp_dir Directory for temp file created by `check_cmd`. Useful for checkers dependent on config file location (e.g. daemons restricted to their own config directories by an apparmor profile).

tmp_ext Suffix for temp file created by `check_cmd`. Useful for checkers dependent on config file extension (e.g. the init-checkconf upstart config checker).

win_owner: **`None`** The owner of the directory. If this is not passed, user will be used. If user is not passed, the account under which Salt is running will be used.

win_perms: **`None`** A dictionary containing permissions to grant and their propagation. For example: `{'Administrators': {'perms': 'full_control'}}` Can be a single basic perm or a list of advanced perms. `perms` must be specified. `applies_to` does not apply to file objects.

win_deny_perms: **`None`** A dictionary containing permissions to deny and their propagation. For example: `{'Administrators': {'perms': 'full_control'}}` Can be a single basic perm or a list of advanced perms. `perms` must be specified. `applies_to` does not apply to file objects.

win_inheritance: **`True`** True to inherit permissions from the parent directory, False not to inherit permission.

win_perms_reset: **`False`** If `True` the existing DACL will be cleared and replaced with the settings defined in this function. If `False`, new entries will be appended to the existing DACL. Default is `False`.

State SLS Example:

```
/var/backups/{{ opts.id }}/{{ salt.status.time('%s') }}.cfg:
  netconfig.saved:
    - source: running
    - makedirs: true
```

The state SLS above would create a backup config grouping the files by the Minion ID, in chronological files. For example, if the state is executed at on the 3rd of August 2018, at 5:15PM, on the Minion `core1.lon01`, the configuration would saved in the file: `/var/backups/core01.lon01/1533316558.cfg`

21.19.181 salt.states.netntp

Network NTP

Manage the configuration of NTP peers and servers on the network devices through the NAPALM proxy.

codeauthor Mircea Ulinic <ping@mirceaulinic.net> & Jerome Fleury <jf@cloudflare.com>

maturity new

depends napalm

platform unix

Dependencies

- Requires netaddr to be installed: *pip install netaddr* to check if IP Addresses are correctly specified
- Requires dnspython to be installed: *pip install dnspython* to resolve the nameserver entities (in case the user does not configure the peers/servers using their IP addresses)
- *NAPALM proxy minion*
- *NTP operational and configuration management module*

`salt.states.netntp.`**`managed`**`(`*name*, *peers=None*, *servers=None*`)`

Manages the configuration of NTP peers and servers on the device, as specified in the state SLS file. NTP entities not specified in these lists will be removed whilst entities not configured on the device will be set.

SLS Example:

```
netntp_example:
    netntp.managed:
         - peers:
            - 192.168.0.1
            - 172.17.17.1
         - servers:
            - 24.124.0.251
            - 138.236.128.36
```

Output example:

```
{
    'edge01.nrt04': {
        'netntp_|-netntp_example_|-netntp_example_|-managed': {
            'comment': 'NTP servers already configured as needed.',
            'name': 'netntp_example',
            'start_time': '12:45:24.056659',
```

(continues on next page)

(continued from previous page)

```
                'duration': 2938.857,
                'changes': {
                    'peers': {
                        'removed': [
                            '192.168.0.2',
                            '192.168.0.3'
                        ],
                        'added': [
                            '192.168.0.1',
                            '172.17.17.1'
                        ]
                    }
                },
                'result': None
            }
        }
}
```

21.19.182 salt.states.netsnmp

Network SNMP

Manage the SNMP configuration on network devices.

codeauthor Mircea Ulinic <ping@mirceaulinic.net>

maturity new

depends napalm

platform unix

Dependencies

- *napalm snmp management module (salt.modules.napalm_snmp)*

salt.states.netsnmp.**managed**(*name*, *config=None*, *defaults=None*)

Configures the SNMP on the device as specified in the SLS file.

SLS Example:

```
snmp_example:
    netsnmp.managed:
         - config:
            location: Honolulu, HI, US
         - defaults:
            contact: noc@cloudflare.com
```

Output example (for the SLS above, e.g. called snmp.sls under /router/):

```
$ sudo salt edge01.hnl01 state.sls router.snmp test=True
edge01.hnl01:
----------
          ID: snmp_example
    Function: snmp.managed
```

(continues on next page)

(continued from previous page)

```
      Result: None
     Comment: Testing mode: configuration was not changed!
     Started: 13:29:06.872363
    Duration: 920.466 ms
     Changes:
              ----------
              added:
                  ----------
                  chassis_id:
                      None
                  contact:
                      noc@cloudflare.com
                  location:
                      Honolulu, HI, US

Summary for edge01.hnl01
------------
Succeeded: 1 (unchanged=1, changed=1)
Failed:    0
------------
Total states run:     1
Total run time: 920.466 ms
```

21.19.183 salt.states.netusers

Network Users

Manage the users configuration on network devices via the NAPALM proxy.

codeauthor Mircea Ulinic <ping@mirceaulinic.net>

maturity new

depends napalm

platform unix

Dependencies

- *NAPALM proxy minion*
- *Users configuration management module*

New in version 2016.11.0.

`salt.states.netusers.`**`managed`**(*name, users=None, defaults=None*)

Manages the configuration of the users on the device, as specified in the state SLS file. Users not defined in that file will be remove whilst users not configured on the device, will be added.

SLS Example:

```
netusers_example:
    netusers.managed:
         - users:
            admin:
                level: 15
```

(continues on next page)

(continued from previous page)

```
                password: $1$knmhgPPv$g8745biu4rb.Zf.IT.F/U1
                sshkeys: []
            restricted:
                level: 1
                password: $1$j34j5k4b$4d5SVjTiz1l.Zf.IT.F/K7
            martin:
                level: 15
                password: ''
                sshkeys:
                    - ssh-dss AAAAB3NzaC1kc3MAAACBAK9dP3KariMlM/
↪JmFW9rTSm5cXs4nR0+o6fTHP9o+bOLXMBTP8R4vwWHh0w
                       ⊠
↪JPjQmJYafAqZTnlgi0srGjyifFwPtODppDWLCgLe2M4LXnu3OMqknr54w344zPHP3iFwWxHrBrZKtCjO8LhbWCa+
                       ⊠
↪X528+i87t6r5e4ersdfxgchvjbknlio87t6r5drcfhgjhbknio8976tycv7t86ftyiu87Oz1nKsKuNzm2csoUQlJ
                       ⊠
↪trmRfpjsOPNookmOz5wG0YxhwDmKeo6fWK+ATk1OiP+QT39fn4G77j8o+e4WAwxM570s35Of/
↪vV0zoOccj753sXn
                       ⊠
↪pvJenvwpM2H6o3a9ALvehAJKWodAgZT7X8+iu786r5drtycghvjbiu78t+wAAAIBURwSPZVElXe+9a43sF6M4ysT
                        7Xv+6wTsa8q86E3+RYyu8O2ObI2kwNLC3/HTgFniE/YqRG+WJac81/
↪VHWQNP822gns8RVrWKjqBktmQoEm7z5yy0
                        bkjui78675dytcghvjkoi9y7t867ftcuvhbuu9t78gy/v+zvMmv8KvQgHg
            jonathan:
                level: 15
                password: ''
                sshkeys:
                    - ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDcgxE6HZF/
↪xjFtIt0thEDKPjFJxW9BpZtTVstYbDgGR9zPkHG
                        ZJT/j345jk345jk453jk43545j35nl3kln34n5kl4ghv3/JzWt/
↪0Js5KZp/51KRNCs9O4t07qaoqwpLB15GwLfEX
                        Bx9dW26zc4O+hi6754trxcfghvjbo98765drt/
↪LYIEg0KSQPWyJEK1g31gacbxN7Ab006xeHh7rv7HtXF6zH3WId
                       ⊠
↪Uhq9rtdUag6kYnv6qvjG7sbCyHGYu5vZB7GytnNuVNbZuI+RdFvmHSnErV9HCu9xZBq6DBb+sESMS4s7nFcsruMo
                        edb+BAc3aww0naeWpogjSt+We7y2N
```

CLI Example:

salt 'edge01.kix01' state.sls router.users

Output example (raw python - can be reused in other modules):

```
{
    'netusers_|-netusers_example_|-netusers_example_|-managed': {
        'comment': 'Configuration updated!',
        'name': 'netusers_example',
        'start_time': '10:57:08.678811',
        '__id__': 'netusers_example',
        'duration': 1620.982,
        '__run_num__': 0,
        'changes': {
            'updated': {
                'admin': {
                    'level': 15
                },
                'restricted': {
                    'level': 1
```

(continues on next page)

(continued from previous page)

```
                },
                'martin': {
                    'sshkeys': [
                        'ssh-dss AAAAB3NzaC1kc3MAAACBAK9dP3KariMlM/
↪JmFW9rTSm5cXs4nR0+o6fTHP9o+bOLXMBTP8R4vwWHh0w
                       ⊠
↪JPjQmJYafAqZTnlgi0srGjyifFwPtODppDWLCgLe2M4LXnu3OMqknr54w344zPHP3iFwWxHrBrZKtCjO8LhbWCa+
                       ⊠
↪X528+i87t6r5e4ersdfxgchvjbknlio87t6r5drcfhgjhbknio8976tycv7t86ftyiu87Oz1nKsKuNzm2csoUQlJ
                       ⊠
↪trmRfpjsOPNookmOz5wG0YxhwDmKeo6fWK+ATk1OiP+QT39fn4G77j8o+e4WAwxM570s35Of/
↪vV0zoOccj753sXn
                       ⊠
↪pvJenvwpM2H6o3a9ALvehAJKWodAgZT7X8+iu786r5drtycghvjbiu78t+wAAAIBURwSPZVElXe+9a43sF6M4ysT
                        7Xv+6wTsa8q86E3+RYyu8O2ObI2kwNLC3/HTgFniE/YqRG+WJac81/
↪VHWQNP822gns8RVrWKjqBktmQoEm7z5yy0
                        bkjui78675dytcghvjkoi9y7t867ftcuvhbuu9t78gy/v+zvMmv8KvQgHg
↪'
                    ]
                }
            },
            'added': {
                'jonathan': {
                    'password': '',
                    'sshkeys': [
                        'ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDcgxE6HZF/
↪xjFtIt0thEDKPjFJxW9BpZtTVstYbDgGR9zPkHG
                        ZJT/j345jk345jk453jk43545j35nl3kln34n5kl4ghv3/JzWt/
↪0Js5KZp/51KRNCs9O4t07qaoqwpLB15GwLfEX
                        Bx9dW26zc4O+hi6754trxcfghvjbo98765drt/
↪LYIEg0KSQPWyJEK1g31gacbxN7Ab006xeHh7rv7HtXF6zH3WId
                       ⊠
↪Uhq9rtdUag6kYnv6qvjG7sbCyHGYu5vZB7GytnNuVNbZuI+RdFvmHSnErV9HCu9xZBq6DBb+sESMS4s7nFcsruMo
                        edb+BAc3aww0naeWpogjSt+We7y2N'
                    ],
                    'level': 15
                }
            },
            'removed': {
            }
        },
        'result': True
    }
}
```

CLI Output:

```
edge01.kix01:
    ----------
              ID: netusers_example
        Function: netusers.managed
          Result: True
         Comment: Configuration updated!
         Started: 11:03:31.957725
        Duration: 1220.435 ms
         Changes:
```

(continues on next page)

(continued from previous page)

```
                  ----------
                  added:
                      ----------
                      jonathan:
                          ----------
                          level:
                              15
                          password:
                          sshkeys:
                              - ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDcgxE6HZF/
↪xjFtIt0thEDKPjFJxW9BpZtTVstYbDgG
                              R9zPkHGZJT/
↪j345jk345jk453jk43545j35nl3kln34n5kl4ghv3/JzWt/0Js5KZp/51KRNCs9O4t07qao
                              qwpLB15GwLfEXBx9dW26zc4O+hi6754trxcfghvjbo98765drt/
↪LYIEg0KSQPWyJEK1g31gacbxN7Ab006
                             ⊠
↪xeHh7rv7HtXF6zH3WIdUhq9rtdUag6kYnv6qvjG7sbCyHGYu5vZB7GytnNuVNbZuI+RdFvmHSnErV9HCu9
                             ⊠
↪xZBq6DBb+sESMS4s7nFcsruMoedb+BAc3aww0naeWpogjSt+We7y2N
                  removed:
                      ----------
                  updated:
                      ----------
                      martin:
                          ----------
                          sshkeys:
                              - ssh-dss AAAAB3NzaC1kc3MAAACBAK9dP3KariMlM/
↪JmFW9rTSm5cXs4nR0+o6fTHP9o+bOLXMBTP8R4
                             ⊠
↪vwWHh0wJPjQmJYafAqZTnlgi0srGjyifFwPtODppDWLCgLe2M4LXnu3OMqknr54w344zPHP3iFwWxHrBrZ
                             ⊠
↪KtCjO8LhbWCa+X528+i87t6r5e4ersdfxgchvjbknlio87t6r5drcfhgjhbknio8976tycv7t86ftyiu87
                             ⊠
↪Oz1nKsKuNzm2csoUQlJtrmRfpjsOPNookmOz5wG0YxhwDmKeo6fWK+ATk1OiP+QT39fn4G77j8o+e4WAwx
                              M570s35Of/
↪vV0zoOccj753sXnpvJenvwpM2H6o3a9ALvehAJKWodAgZT7X8+iu786r5drtycghvjbiu78t
                             ⊠
↪+wAAAIBURwSPZVElXe+9a43sF6M4ysT7Xv+6wTsa8q86E3+RYyu8O2ObI2kwNLC3/HTgFniE/
↪YqRG+WJac
                              81/
↪VHWQNP822gns8RVrWKjqBktmQoEm7z5yy0bkjui78675dytcghvjkoi9y7t867ftcuvhbuu9t78gy/v
                              +zvMmv8KvQgHg
                      admin:
                          ----------
                          level:
                              15
                      restricted:
                          ----------
                          level:
                              1
    Summary for edge01.kix01
    ------------
    Succeeded: 1 (changed=1)
    Failed:    0
    ------------
    Total states run:     1
    Total run time:   1.220 s
```

21.19.184 salt.states.network

Configuration of network interfaces

The network module is used to create and manage network settings, interfaces can be set as either managed or ignored. By default all interfaces are ignored unless specified.

Note: RedHat-based systems (RHEL, CentOS, Scientific, etc.) have been supported since version 2014.1.0.

Debian-based systems (Debian, Ubuntu, etc.) have been supported since version 2017.7.0. The following options are not supported: ipaddr_start, and ipaddr_end.

Other platforms are not yet supported.

Note: On Debian-based systems, networking configuration can be specified in */etc/network/interfaces* or via included files such as (by default) */etc/network/interfaces.d/**. This can be problematic for configuration management. It is recommended to use either *file.managed or network.managed*.

If using *network.managed*, it can be useful to ensure *interfaces.d/* is empty. This can be done using:

/etc/network/interfaces.d:

  file.directory:

  - clean: True

```
system:
  network.system:
    - enabled: True
    - hostname: server1.example.com
    - gateway: 192.168.0.1
    - gatewaydev: eth0
    - nozeroconf: True
    - nisdomain: example.com
    - require_reboot: True

eth0:
  network.managed:
    - enabled: True
    - type: eth
    - proto: static
    - ipaddr: 10.1.0.7
    - netmask: 255.255.255.0
    - gateway: 10.1.0.1
    - enable_ipv6: true
    - ipv6proto: static
    - ipv6ipaddrs:
      - 2001:db8:dead:beef::3/64
      - 2001:db8:dead:beef::7/64
    - ipv6gateway: 2001:db8:dead:beef::1
    - ipv6netmask: 64
    - dns:
      - 8.8.8.8
      - 8.8.4.4

eth0-range0:
```

(continues on next page)

(continued from previous page)

```
  network.managed:
    - type: eth
    - ipaddr_start: 192.168.1.1
    - ipaddr_end: 192.168.1.10
    - clonenum_start: 10
    - mtu: 9000

bond0-range0:
  network.managed:
    - type: eth
    - ipaddr_start: 192.168.1.1
    - ipaddr_end: 192.168.1.10
    - clonenum_start: 10
    - mtu: 9000

eth1.0-range0:
  network.managed:
    - type: eth
    - ipaddr_start: 192.168.1.1
    - ipaddr_end: 192.168.1.10
    - clonenum_start: 10
    - vlan: True
    - mtu: 9000

bond0.1-range0:
  network.managed:
    - type: eth
    - ipaddr_start: 192.168.1.1
    - ipaddr_end: 192.168.1.10
    - clonenum_start: 10
    - vlan: True
    - mtu: 9000

.. note::
    add support of ranged interfaces (vlan, bond and eth) for redhat system,
    Important:type must be eth.

routes:
  network.routes:
    - name: eth0
    - routes:
      - name: secure_network
        ipaddr: 10.2.0.0
        netmask: 255.255.255.0
        gateway: 10.1.0.3
      - name: HQ_network
        ipaddr: 10.100.0.0
        netmask: 255.255.0.0
        gateway: 10.1.0.10

eth2:
  network.managed:
    - enabled: True
    - type: slave
    - master: bond0
```

(continues on next page)

(continued from previous page)

```
eth3:
  network.managed:
    - enabled: True
    - type: slave
    - master: bond0

eth4:
  network.managed:
    - enabled: True
    - type: eth
    - proto: dhcp
    - bridge: br0

eth5:
  network.managed:
    - enabled: True
    - type: eth
    - proto: dhcp
    - noifupdown: True  # Do not restart the interface
                        # you need to reboot/reconfigure manualy

bond0:
  network.managed:
    - type: bond
    - ipaddr: 10.1.0.1
    - netmask: 255.255.255.0
    - mode: gre
    - proto: static
    - dns:
      - 8.8.8.8
      - 8.8.4.4
    - enabled: False
    - slaves: eth2 eth3
    - require:
      - network: eth2
      - network: eth3
    - miimon: 100
    - arp_interval: 250
    - downdelay: 200
    - lacp_rate: fast
    - max_bonds: 1
    - updelay: 0
    - use_carrier: on
    - hashing-algorithm: layer2
    - mtu: 9000
    - autoneg: on
    - speed: 1000
    - duplex: full
    - rx: on
    - tx: off
    - sg: on
    - tso: off
    - ufo: off
    - gso: off
    - gro: off
    - lro: off
```

(continues on next page)

(continued from previous page)

```
bond0.2:
  network.managed:
    - type: vlan
    - ipaddr: 10.1.0.2
    - use:
      - network: bond0
    - require:
      - network: bond0

bond0.3:
  network.managed:
    - type: vlan
    - ipaddr: 10.1.0.3
    - use:
      - network: bond0
    - require:
      - network: bond0

bond0.10:
  network.managed:
    - type: vlan
    - ipaddr: 10.1.0.4
    - use:
      - network: bond0
    - require:
      - network: bond0

bond0.12:
  network.managed:
    - type: vlan
    - ipaddr: 10.1.0.5
    - use:
      - network: bond0
    - require:
      - network: bond0
br0:
  network.managed:
    - enabled: True
    - type: bridge
    - proto: dhcp
    - bridge: br0
    - delay: 0
    - ports: eth4
    - bypassfirewall: True
    - use:
      - network: eth4
    - require:
      - network: eth4

eth6:
  network.managed:
    - type: eth
    - noifupdown: True

    # IPv4
```

(continues on next page)

(continued from previous page)

```
    - proto: static
    - ipaddr: 192.168.4.9
    - netmask: 255.255.255.0
    - gateway: 192.168.4.1
    - enable_ipv6: True

    # IPv6
    - ipv6proto: static
    - ipv6ipaddr: 2001:db8:dead:c0::3
    - ipv6netmask: 64
    - ipv6gateway: 2001:db8:dead:c0::1
    # override shared; makes those options v4-only
    - ipv6ttl: 15

    # Shared
    - mtu: 1480
    - ttl: 18
    - dns:
      - 8.8.8.8
      - 8.8.4.4

eth7:
    - type: eth
    - proto: static
    - ipaddr: 10.1.0.7
    - netmask: 255.255.255.0
    - gateway: 10.1.0.1
    - enable_ipv6: True
    - ipv6proto: static
    - ipv6ipaddr: 2001:db8:dead:beef::3
    - ipv6netmask: 64
    - ipv6gateway: 2001:db8:dead:beef::1
    - noifupdown: True

eth8:
  network.managed:
    - enabled: True
    - type: eth
    - proto: static
    - enable_ipv6: true
    - ipv6proto: static
    - ipv6ipaddrs:
      - 2001:db8:dead:beef::3/64
      - 2001:db8:dead:beef::7/64
    - ipv6gateway: 2001:db8:dead:beef::1
    - ipv6netmask: 64
    - dns:
      - 8.8.8.8
      - 8.8.4.4

system:
  network.system:
    - enabled: True
    - hostname: server1.example.com
    - gateway: 192.168.0.1
    - gatewaydev: eth0
```

(continues on next page)

(continued from previous page)

```
    - nozeroconf: True
    - nisdomain: example.com
    - require_reboot: True
    - apply_hostname: True

lo:
  network.managed:
    - name: lo
    - type: eth
    - proto: loopback
    - onboot: yes
    - userctl: no
    - ipv6_autoconf: no
    - enable_ipv6: true

.. note::
    Apply changes to hostname immediately.

.. versionadded:: 2015.5.0

system:
  network.system:
    - hostname: server2.example.com
    - apply_hostname: True
    - retain_settings: True

.. note::
    Use `retain_settings` to retain current network settings that are not
    otherwise specified in the state. Particularly useful if only setting
    the hostname. Default behavior is to delete unspecified network
    settings.

.. versionadded:: 2016.11.0
```

Note: When managing bridged interfaces on a Debian or Ubuntu based system, the ports argument is required. Red Hat systems will ignore the argument.

`salt.states.network.`**`managed`**(*name, type, enabled=True, **kwargs*)

Ensure that the named interface is configured properly.

name The name of the interface to manage

type Type of interface and configuration.

enabled Designates the state of this interface.

kwargs The IP parameters for this interface.

`salt.states.network.`**`routes`**(*name, **kwargs*)

Manage network interface static routes.

name Interface name to apply the route to.

kwargs Named routes

`salt.states.network.`**`system`**(*name, **kwargs*)

Ensure that global network settings are configured properly.

name Custom name to represent this configuration change.

kwargs The global parameters for the system.

21.19.185 salt.states.net_napalm_yang

NAPALM YANG state

Manage the configuration of network devices according to the YANG models (OpenConfig/IETF).

New in version 2017.7.0.

Dependencies

- napalm-yang
- pyangbing > 0.5.11

To be able to load configuration on network devices, it requires NAPALM library to be installed: `pip install napalm`. Please check Installation for complete details.

`salt.states.net_napalm_yang.`**`configured`**`(`*name*, *data*, ***kwargs*`)`

Configure the network device, given the input data strucuted according to the YANG models.

> **Note:** The main difference between this function and `managed` is that the later generates and loads the configuration only when there are differences between the existing configuration on the device and the expected configuration. Depending on the platform and hardware capabilities, one could be more optimal than the other. Additionally, the output of the `managed` is different, in such a way that the `pchange` field in the output contains structured data, rather than text.

data YANG structured data.

models A list of models to be used when generating the config.

profiles: **`None`** Use certain profiles to generate the config. If not specified, will use the platform default profile(s).

test: **`False`** Dry run? If set as `True`, will apply the config, discard and return the changes. Default: `False` and will commit the changes on the device.

commit: **`True`** Commit? Default: `True`.

debug: **`False`** Debug mode. Will insert a new key under the output dictionary, as `loaded_config` containing the raw configuration loaded on the device.

replace: **`False`** Should replace the config with the new generate one?

State SLS example:

```
{%- set expected_config =  pillar.get('openconfig_interfaces_cfg') -%}
interfaces_config:
  napalm_yang.configured:
    - data: {{ expected_config | json }}
    - models:
      - models.openconfig_interfaces
    - debug: true
```

Pillar example:

```
openconfig_interfaces_cfg:
  _kwargs:
    filter: true
  interfaces:
    interface:
      Et1:
```

(continues on next page)

(continued from previous page)

```
        config:
          mtu: 9000
      Et2:
        config:
          description: "description example"
```

`salt.states.net_napalm_yang.`**`managed`**`(`*name*, *data*, ***kwargs*`)`

Manage the device configuration given the input data structured according to the YANG models.

data YANG structured data.

models A list of models to be used when generating the config.

profiles: **`None`** Use certain profiles to generate the config. If not specified, will use the platform default profile(s).

compliance_report: **`False`** Return the compliance report in the comment.

New in version 2017.7.3.

test: **`False`** Dry run? If set as `True`, will apply the config, discard and return the changes. Default: `False` and will commit the changes on the device.

commit: **`True`** Commit? Default: `True`.

debug: **`False`** Debug mode. Will insert a new key under the output dictionary, as `loaded_config` containing the raw configuration loaded on the device.

replace: **`False`** Should replace the config with the new generate one?

State SLS example:

```
{%- set expected_config =  pillar.get('openconfig_interfaces_cfg') -%}
interfaces_config:
  napalm_yang.managed:
    - data: {{ expected_config | json }}
    - models:
      - models.openconfig_interfaces
    - debug: true
```

Pillar example:

```
openconfig_interfaces_cfg:
  _kwargs:
    filter: true
  interfaces:
    interface:
      Et1:
        config:
          mtu: 9000
      Et2:
        config:
          description: "description example"
```

21.19.186 salt.states.neutron_network

Management of OpenStack Neutron Networks

New in version 2018.3.0.

depends shade

configuration see *`salt.modules.neutronng`* for setup instructions

Example States

```
create network:
  neutron_network.present:
    - name: network1

delete network:
  neutron_network.absent:
    - name: network1

create network with optional params:
  neutron_network.present:
    - name: network1
    - vlan: 200
    - shared: False
    - external: False
    - project: project1
```

`salt.states.neutron_network.`**`absent`**(*name*, *auth=None*, ***kwargs*)

Ensure a network does not exists

name Name of the network

`salt.states.neutron_network.`**`present`**(*name*, *auth=None*, ***kwargs*)

Ensure a network exists and is up-to-date

name Name of the network

provider A dict of network provider options.

shared Set the network as shared.

external Whether this network is externally accessible.

admin_state_up Set the network administrative state to up.

vlan Vlan ID. Alias for provider

- physical_network: provider
- network_type: vlan
- segmentation_id: (vlan id)

21.19.187 salt.states.neutron_secgroup

Management of OpenStack Neutron Security Groups

New in version 2018.3.0.

depends shade

configuration see `salt.modules.neutronng` for setup instructions

Example States

```
create security group;
  neutron_secgroup.present:
    - name: security_group1
    - description: "Very Secure Security Group"

delete security group:
  neutron_secgroup.absent:
    - name_or_id: security_group1
    - project_name: Project1

create security group with optional params:
```

(continues on next page)

(continued from previous page)

```
  neutron_secgroup.present:
    - name: security_group1
    - description: "Very Secure Security Group"
    - project_id: 1dcac318a83b4610b7a7f7ba01465548

create security group with optional params:
  neutron_secgroup.present:
    - name: security_group1
    - description: "Very Secure Security Group"
    - project_name: Project1
```

`salt.states.neutron_secgroup.`**`absent`**(*name*, *auth=None*, ***kwargs*)

Ensure a security group does not exist

name Name of the security group

`salt.states.neutron_secgroup.`**`present`**(*name*, *auth=None*, ***kwargs*)

Ensure a security group exists.

You can supply either project_name or project_id.

Creating a default security group will not show up as a change; it gets created through the lookup process.

name Name of the security group

description Description of the security group

project_name Name of Project

project_id ID of Project

21.19.188 salt.states.neutron_secgroup_rule

Management of OpenStack Neutron Security Group Rules

New in version 2018.3.0.

depends shade

configuration see *salt.modules.neutronng* for setup instructions

Example States

```
create security group rule:
  neutron_secgroup_rule.present:
    - name: security_group1
    - project_name: Project1
    - protocol: icmp

delete security group:
  neutron_secgroup_rule.absent:
    - name_or_id: security_group1

create security group with optional params:
  neutron_secgroup_rule.present:
    - name: security_group1
    - description: "Very Secure Security Group"
    - project_id: 1dcac318a83b4610b7a7f7ba01465548
```

`salt.states.neutron_secgroup_rule.`**`absent`**(*name*, *auth=None*, ***kwargs*)

Ensure a security group rule does not exist

name name or id of the security group rule to delete

rule_id uuid of the rule to delete
project_id id of project to delete rule from

`salt.states.neutron_secgroup_rule.`**`present`**`(`*name, auth=None, **kwargs*`)`
Ensure a security group rule exists
defaults: port_range_min=None, port_range_max=None, protocol=None, remote_ip_prefix=None, remote_group_id=None, direction='ingress', ethertype='IPv4', project_id=None
name Name of the security group to associate with this rule
project_name Name of the project associated with the security group
protocol The protocol that is matched by the security group rule. Valid values are None, tcp, udp, and icmp.

21.19.189 salt.states.neutron_subnet

Management of OpenStack Neutron Subnets

New in version 2018.3.0.

depends shade

configuration see `salt.modules.neutronng` for setup instructions

Example States

```
create subnet:
  neutron_subnet.present:
    - name: subnet1
    - network_name_or_id: network1
    - cidr: 192.168.199.0/24


delete subnet:
  neutron_subnet.absent:
    - name: subnet2

create subnet with optional params:
  neutron_subnet.present:
    - name: subnet1
    - network_name_or_id: network1
    - enable_dhcp: True
    - cidr: 192.168.199.0/24
    - allocation_pools:
      - start: 192.168.199.5
        end: 192.168.199.250
    - host_routes:
      - destination: 192.168..0.0/24
        nexthop: 192.168.0.1
    - gateway_ip: 192.168.199.1
    - dns_nameservers:
      - 8.8.8.8
      - 8.8.8.7

create ipv6 subnet:
  neutron_subnet.present:
    - name: v6subnet1
    - network_name_or_id: network1
    - ip_version: 6
```

`salt.states.neutron_subnet.`**`absent`**`(`*name*, *auth=None*`)`
Ensure a subnet does not exists
name Name of the subnet

`salt.states.neutron_subnet.`**`present`**`(`*name*, *auth=None*, ***kwargs*`)`
Ensure a subnet exists and is up-to-date
name Name of the subnet
network_name_or_id The unique name or ID of the attached network. If a non-unique name is supplied, an exception is raised.
allocation_pools A list of dictionaries of the start and end addresses for the allocation pools
gateway_ip The gateway IP address.
dns_nameservers A list of DNS name servers for the subnet.
host_routes A list of host route dictionaries for the subnet.
ipv6_ra_mode IPv6 Router Advertisement mode. Valid values are: 'dhcpv6-stateful', 'dhcpv6-stateless', or 'slaac'.
ipv6_address_mode IPv6 address mode. Valid values are: 'dhcpv6-stateful', 'dhcpv6-stateless', or 'slaac'.

21.19.190 salt.states.nfs_export

Management of NFS exports

New in version 2018.3.0.

To ensure an NFS export exists:

```
add_simple_export:
  nfs_export.present:
    - name:     '/srv/nfs'
    - hosts:    '10.0.2.0/24'
    - options:
      - 'rw'
```

This creates the following in /etc/exports:

```
/srv/nfs 10.0.2.0/24(rw)
```

For more complex exports with multiple groups of hosts, use 'clients':

```
add_complex_export:
  nfs_export.present:
    - name: '/srv/nfs'
    - clients:
      # First export, same as simple one above
      - hosts: '10.0.2.0/24'
        options:
          - 'rw'
      # Second export
      - hosts: '*.example.com'
        options:
          - 'ro'
          - 'subtree_check'
```

This creates the following in /etc/exports:

```
/srv/nfs 10.0.2.0/24(rw) 192.168.0.0/24,172.19.0.0/16(ro,subtree_check)
```

Any export of the given path will be modified to match the one specified.

To ensure an NFS export is absent:

```
delete_export:
  nfs_export.absent:
    - name: '/srv/nfs'
```

`salt.states.nfs_export.`**`absent`**`(`*name, exports='/etc/exports'*`)`

Ensure that the named path is not exported

name The export path to remove

`salt.states.nfs_export.`**`present`**`(`*name, clients=None, hosts=None, options=None, exports='/etc/exports'*`)`

Ensure that the named export is present with the given options

name The export path to configure

clients A list of hosts and the options applied to them. This option may not be used in combination with the 'hosts' or 'options' shortcuts.

```
- clients:
  # First export
  - hosts: '10.0.2.0/24'
    options:
      - 'rw'
  # Second export
  - hosts: '*.example.com'
    options:
      - 'ro'
      - 'subtree_check'
```

hosts A string matching a number of hosts, for example:

```
hosts: '10.0.2.123'

hosts: '10.0.2.0/24'

hosts: 'minion1.example.com'

hosts: '*.example.com'

hosts: '*'
```

options A list of NFS options, for example:

```
options:
  - 'rw'
  - 'subtree_check'
```

21.19.191 salt.states.nftables

Management of nftables

This is an nftables-specific module designed to manage Linux firewalls. It is expected that this state module, and other system-specific firewall states, may at some point be deprecated in favor of a more generic *firewall* state.

```
httpd:
  nftables.append:
    - table: filter
    - chain: input
    - jump: accept
    - match: state
    - connstate: new
    - dport: 80
    - proto: tcp
    - sport: 1025:65535
    - save: True

httpd:
  nftables.append:
    - table: filter
    - family: ipv6
    - chain: INPUT
    - jump: ACCEPT
    - match: state
    - connstate: NEW
    - dport: 80
    - proto: tcp
    - sport: 1025:65535
    - save: True

httpd:
  nftables.insert:
    - position: 1
    - table: filter
    - chain: INPUT
    - jump: ACCEPT
    - match: state
    - connstate: NEW
    - dport: 80
    - proto: tcp
    - sport: 1025:65535
    - save: True

httpd:
  nftables.insert:
    - position: 1
    - table: filter
    - family: ipv6
    - chain: INPUT
    - jump: ACCEPT
    - match: state
    - connstate: NEW
    - dport: 80
    - proto: tcp
    - sport: 1025:65535
    - save: True

httpd:
  nftables.delete:
    - table: filter
    - chain: INPUT
    - jump: ACCEPT
```

(continues on next page)

(continued from previous page)

```
    - match: state
    - connstate: NEW
    - dport: 80
    - proto: tcp
    - sport: 1025:65535
    - save: True

httpd:
  nftables.delete:
    - position: 1
    - table: filter
    - chain: INPUT
    - jump: ACCEPT
    - match: state
    - connstate: NEW
    - dport: 80
    - proto: tcp
    - sport: 1025:65535
    - save: True

httpd:
  nftables.delete:
    - table: filter
    - family: ipv6
    - chain: INPUT
    - jump: ACCEPT
    - match: state
    - connstate: NEW
    - dport: 80
    - proto: tcp
    - sport: 1025:65535
    - save: True

output:
  nftables.chain_present:
    - family: ip
    - table: filter

output:
  nftables.chain_absent:
    - family: ip
    - table: filter
```

`salt.states.nftables.`**`append`**`(`*name*, *family='ipv4'*, ***kwargs*`)`

New in version 0.17.0.

Append a rule to a chain

name A user-defined name to call this rule by in another part of a state or formula. This should not be an actual rule.

family Network family, ipv4 or ipv6.

All other arguments are passed in with the same name as the long option that would normally be used for nftables, with one exception: *–state* is specified as *connstate* instead of *state* (not to be confused with *ctstate*).

`salt.states.nftables.`**`chain_absent`**`(`*name*, *table='filter'*, *family='ipv4'*`)`

New in version 2014.7.0.

Verify the chain is absent.

family Networking family, either ipv4 or ipv6 `salt.states.nftables.`**`chain_present`**(*name*, *table='filter'*, *table_type=None*, *hook=None*, *priority=None*, *family='ipv4'*)

New in version 2014.7.0.

Verify the chain is exist.

name A user-defined chain name.

table The table to own the chain.

family Networking family, either ipv4 or ipv6

`salt.states.nftables.`**`delete`**(*name*, *family='ipv4'*, ***kwargs*)

New in version 2014.7.0.

Delete a rule to a chain

name A user-defined name to call this rule by in another part of a state or formula. This should not be an actual rule.

family Networking family, either ipv4 or ipv6

All other arguments are passed in with the same name as the long option that would normally be used for nftables, with one exception: *–state* is specified as *connstate* instead of *state* (not to be confused with *ctstate*).

`salt.states.nftables.`**`flush`**(*name*, *family='ipv4'*, ***kwargs*)

New in version 2014.7.0.

Flush current nftables state

family Networking family, either ipv4 or ipv6

`salt.states.nftables.`**`insert`**(*name*, *family='ipv4'*, ***kwargs*)

New in version 2014.7.0.

Insert a rule into a chain

name A user-defined name to call this rule by in another part of a state or formula. This should not be an actual rule.

family Networking family, either ipv4 or ipv6

All other arguments are passed in with the same name as the long option that would normally be used for nftables, with one exception: *–state* is specified as *connstate* instead of *state* (not to be confused with *ctstate*).

21.19.192 salt.states.npm

Installation of NPM Packages

These states manage the installed packages for node.js using the Node Package Manager (npm). Note that npm must be installed for these states to be available, so npm states should include a requisite to a pkg.installed state for the package which provides npm (simply `npm` in most cases). Example:

```
npm:
  pkg.installed

yaml:
  npm.installed:
    - require:
      - pkg: npm
```

`salt.states.npm.`**`bootstrap`**(*name*, *user=None*, *silent=True*)

Bootstraps a node.js application.

Will execute 'npm install –json' on the specified directory.

user The user to run NPM with

New in version 0.17.0.

`salt.states.npm.`**`cache_cleaned`**`(`*name=None, user=None, force=False*`)`

Ensure that the given package is not cached.

If no package is specified, this ensures the entire cache is cleared.

name The name of the package to remove from the cache, or None for all packages

user The user to run NPM with

force Force cleaning of cache. Required for npm@5 and greater

New in version 2016.11.6.

`salt.states.npm.`**`installed`**`(`*name, pkgs=None, dir=None, user=None, force_reinstall=False, registry=None, env=None*`)`

Verify that the given package is installed and is at the correct version (if specified).

```
coffee-script:
  npm.installed:
    - user: someuser

coffee-script@1.0.1:
  npm.installed: []
```

name The package to install

Changed in version 2014.7.2: This parameter is no longer lowercased by salt so that case-sensitive NPM package names will work.

pkgs A list of packages to install with a single npm invocation; specifying this argument will ignore the `name` argument

New in version 2014.7.0.

dir The target directory in which to install the package, or None for global installation

user The user to run NPM with

New in version 0.17.0.

registry The NPM registry from which to install the package

New in version 2014.7.0.

env A list of environment variables to be set prior to execution. The format is the same as the *`cmd.run`*. state function.

New in version 2014.7.0.

force_reinstall Install the package even if it is already installed

`salt.states.npm.`**`removed`**`(`*name, dir=None, user=None*`)`

Verify that the given package is not installed.

dir The target directory in which to install the package, or None for global installation

user The user to run NPM with

New in version 0.17.0.

21.19.193 salt.states.ntp

Management of NTP servers

New in version 2014.1.0.

This state is used to manage NTP servers. Currently only Windows is supported.

```
win_ntp:
  ntp.managed:
    - servers:
      - pool.ntp.org
      - us.pool.ntp.org
```

`salt.states.ntp.`**`managed`**`(`*name, servers=None*`)`

Manage NTP servers

servers A list of NTP servers

21.19.194 salt.states.nxos module

State module for Cisco NX OS Switches Proxy minions

For documentation on setting up the nxos proxy minion look in the documentation for `salt.proxy.nxos`.

`salt.states.nxos.`**`config_absent`**`(`*name*`)`

Ensure a specific configuration line does not exist in the running config

name config line to remove

Examples:

```
add snmp group:
  nxos.config_absent:
    - names:
      - snmp-server community randoSNMPstringHERE group network-operator
      - snmp-server community AnotherRandomSNMPSTring group network-admin
```

Note: For certain cases extra lines could be removed based on dependencies. In this example, included after the example for config_present, the ACLs would be removed because they depend on the existence of the group.

`salt.states.nxos.`**`config_present`**`(`*name*`)`

Ensure a specific configuration line exists in the running config

name config line to set

Examples:

```
add snmp group:
  nxos.config_present:
    - names:
      - snmp-server community randoSNMPstringHERE group network-operator
      - snmp-server community AnotherRandomSNMPSTring group network-admin

add snmp acl:
  nxos.config_present:
    - names:
      - snmp-server community randoSNMPstringHERE use-acl snmp-acl-ro
      - snmp-server community AnotherRandomSNMPSTring use-acl snmp-acl-rw
```

`salt.states.nxos.`**`replace`**`(`*name, repl, full_match=False*`)`

Replace all instances of a string or full line in the running config

name String to replace

repl The replacement text

full_match Whether *name* will match the full line or only a subset of the line. Defaults to False. When False, .* is added around *name* for matching in the *show run* config.

Examples:

```
replace snmp string:
  nxos.replace:
    - name: randoSNMPstringHERE
    - repl: NEWrandoSNMPstringHERE

replace full snmp string:
  nxos.replace:
    - name: ^snmp-server community randoSNMPstringHERE group network-operator$
    - repl: snmp-server community NEWrandoSNMPstringHERE group network-operator
    - full_match: True
```

Note: The first example will replace the SNMP string on both the group and the ACL, so you will not lose the ACL setting. Because the second is an exact match of the line, when the group is removed, the ACL is removed, but not readded, because it was not matched.

`salt.states.nxos.`**`user_absent`**`(`*name*`)`

Ensure a user is not present

name username to remove if it exists

Examples:

```
delete:
  nxos.user_absent:
    - name: daniel
```

`salt.states.nxos.`**`user_present`**`(`*name, password=None, roles=None, encrypted=False, crypt_salt=None, algorithm='sha256'*`)`

Ensure a user is present with the specified groups

name Name of user

password Encrypted or Plain Text password for user

roles List of roles the user should be assigned. Any roles not in this list will be removed

encrypted Whether the password is encrypted already or not. Defaults to False

crypt_salt Salt to use when encrypting the password. Default is None (salt is randomly generated for unhashed passwords)

algorithm Algorithm to use for hashing password. Defaults to sha256. Accepts md5, blowfish, sha256, sha512

Examples:

```
create:
  nxos.user_present:
    - name: daniel
    - roles:
      - vdc-admin

set_password:
  nxos.user_present:
    - name: daniel
    - password: admin
    - roles:
      - network-admin

update:
  nxos.user_present:
    - name: daniel
    - password: AiN9jaoP
```

(continues on next page)

(continued from previous page)

```
    - roles:
      - network-admin
      - vdc-admin
```

21.19.195 salt.states.openstack_config

Manage OpenStack configuration file settings.

maintainer Jeffrey C. Ollie <jeff@ocjtech.us>

maturity new

depends

platform linux

`salt.states.openstack_config.`**`absent`**(*name*, *filename*, *section*, *parameter=None*)

Ensure a value is not set in an OpenStack configuration file.
filename The full path to the configuration file
section The section in which the parameter will be set
parameter (optional) The parameter to change. If the parameter is not supplied, the name will be used as the parameter.

`salt.states.openstack_config.`**`present`**(*name*, *filename*, *section*, *value*, *parameter=None*)

Ensure a value is set in an OpenStack configuration file.
filename The full path to the configuration file
section The section in which the parameter will be set
parameter (optional) The parameter to change. If the parameter is not supplied, the name will be used as the parameter.
value The value to set

21.19.196 salt.states.openvswitch_bridge module

Management of Open vSwitch bridges.

`salt.states.openvswitch_bridge.`**`absent`**(*name*)

Ensures that the named bridge does not exist, eventually deletes it.

Parameters **`name`** – The name of the bridge.

`salt.states.openvswitch_bridge.`**`present`**(*name*)

Ensures that the named bridge exists, eventually creates it.

Parameters **`name`** – The name of the bridge.

21.19.197 salt.states.openvswitch_port module

Management of Open vSwitch ports.

`salt.states.openvswitch_port.`**`absent`**(*name*, *bridge=None*)

Ensures that the named port exists on bridge, eventually deletes it. If bridge is not set, port is removed from whatever bridge contains it.

Parameters

- **`name`** – The name of the port.
- **`bridge`** – The name of the bridge.

`salt.states.openvswitch_port.`**`present`**`(`*name, bridge, tunnel_type=None, id=None, remote=None, dst_port=None, internal=False*`)`

Ensures that the named port exists on bridge, eventually creates it.

Parameters

- **`name`** – The name of the port.
- **`bridge`** – The name of the bridge.
- **`tunnel_type`** – Optional type of interface to create, currently supports: vlan, vxlan and gre.
- **`id`** – Optional tunnel's key.
- **`remote`** – Remote endpoint's IP address.
- **`dst_port`** – Port to use when creating tunnelport in the switch.
- **`internal`** – Create an internal port if one does not exist

21.19.198 salt.states.opsgenie

Create/Close an alert in OpsGenie

New in version 2018.3.0.

This state is useful for creating or closing alerts in OpsGenie during state runs.

```
used_space:
  disk.status:
    - name: /
    - maximum: 79%
    - minimum: 20%

opsgenie_create_action_sender:
  opsgenie.create_alert:
    - api_key: XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX
    - reason: 'Disk capacity is out of designated range.'
    - name: disk.status
    - onfail:
      - disk: used_space

opsgenie_close_action_sender:
  opsgenie.close_alert:
    - api_key: XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX
    - name: disk.status
    - require:
      - disk: used_space
```

`salt.states.opsgenie.`**`close_alert`**`(`*name=None, api_key=None, reason='Conditions are met.', action_type='Close'*`)`

Close an alert in OpsGenie. It's a wrapper function for create_alert. Example usage with Salt's requisites and other global state arguments could be found above.

Required Parameters:

name It will be used as alert's alias. If you want to use the close functionality you must provide name field for both states like in above case.

Optional Parameters:

api_key It's the API Key you've copied while adding integration in OpsGenie.

reason It will be used as alert's default message in OpsGenie.

action_type OpsGenie supports the default values Create/Close for action_type. You can customize this field with OpsGenie's custom actions for other purposes like adding notes or acknowledging alerts.

`salt.states.opsgenie.`**`create_alert`**`(`*name=None, api_key=None, reason=None, action_type='Create'*`)`

Create an alert in OpsGenie. Example usage with Salt's requisites and other global state arguments could be found above.

Required Parameters:

api_key It's the API Key you've copied while adding integration in OpsGenie.

reason It will be used as alert's default message in OpsGenie.

Optional Parameters:

name It will be used as alert's alias. If you want to use the close functionality you must provide name field for both states like in above case.

action_type OpsGenie supports the default values Create/Close for action_type. You can customize this field with OpsGenie's custom actions for other purposes like adding notes or acknowledging alerts.

21.19.199 salt.states.pagerduty

Create an Event in PagerDuty

New in version 2014.1.0.

This state is useful for creating events on the PagerDuty service during state runs.

```
server-warning-message:
  pagerduty.create_event:
    - name: 'This is a server warning message'
    - details: 'This is a much more detailed message'
    - service_key: 9abcd123456789efabcde362783cdbaf
    - profile: my-pagerduty-account
```

`salt.states.pagerduty.`**`create_event`**`(`*name, details, service_key, profile*`)`

Create an event on the PagerDuty service

```
server-warning-message:
  pagerduty.create_event:
    - name: 'This is a server warning message'
    - details: 'This is a much more detailed message'
    - service_key: 9abcd123456789efabcde362783cdbaf
    - profile: my-pagerduty-account
```

The following parameters are required:

name This is a short description of the event.

details This can be a more detailed description of the event.

service_key This key can be found by using pagerduty.list_services.

profile This refers to the configuration profile to use to connect to the PagerDuty service.

21.19.200 salt.states.pagerduty_escalation_policy

Manage PagerDuty escalation policies.

Schedules and users can be referenced by pagerduty ID, or by name, or by email address.

For example:

```
ensure test escalation policy:
    pagerduty_escalation_policy.present:
```

(continues on next page)

(continued from previous page)

```
        - name: bruce test escalation policy
        - escalation_rules:
            - targets:
                - type: schedule
                  id: 'bruce test schedule level1'
                - type: user
                  id: 'Bruce Sherrod'
              escalation_delay_in_minutes: 15
            - targets:
                - type: schedule
                  id: 'bruce test schedule level2'
              escalation_delay_in_minutes: 15
            - targets:
                - type: user
                  id: 'Bruce TestUser1'
                - type: user
                  id: 'Bruce TestUser2'
                - type: user
                  id: 'Bruce TestUser3'
                - type: user
                  id:  'bruce+test4@lyft.com'
              escalation_delay_in_minutes: 15
```

`salt.states.pagerduty_escalation_policy.`**`absent`**`(`*profile='pagerduty', subdomain=None, api_key=None, **kwargs*`)`

Ensure that a PagerDuty escalation policy does not exist. Accepts all the arguments that pagerduty_escalation_policy.present accepts; but ignores all arguments except the name.

Name can be the escalation policy id or the escalation policy name.

`salt.states.pagerduty_escalation_policy.`**`present`**`(`*profile='pagerduty', subdomain=None, api_key=None, **kwargs*`)`

Ensure that a pagerduty escalation policy exists. Will create or update as needed.

This method accepts as args everything defined in https://developer.pagerduty.com/documentation/rest/escalation_policies/create. In addition, user and schedule id's will be translated from name (or email address) into PagerDuty unique ids. For example:

```
```

pagerduty_escalation_policy.present:

- name: bruce test escalation policy
- **escalation_rules**:
    - **targets**:
        - type: schedule id: 'bruce test schedule level1'
        - type: user id: 'Bruce Sherrod'

In this example, 'Bruce Sherrod' will be looked up and replaced with the PagerDuty id (usually a 7 digit all-caps string, e.g. PX6GQL7)

21.19.201 salt.states.pagerduty_schedule

Manage PagerDuty schedules.

Example:

```
ensure test schedule:
    pagerduty_schedule.present:
        - name: 'bruce test schedule level1'
        - schedule:
            name: 'bruce test schedule level1'
            time_zone: 'Pacific Time (US & Canada)'
            schedule_layers:
                - name: 'Schedule Layer 1'
                  start: '2015-01-01T00:00:00'
                  users:
                    - user:
                        'id': 'Bruce TestUser1'
                      member_order: 1
                    - user:
                        'id': 'Bruce TestUser2'
                      member_order: 2
                    - user:
                        'id': 'bruce+test3@lyft.com'
                      member_order: 3
                    - user:
                        'id': 'bruce+test4@lyft.com'
                      member_order: 4
                  rotation_virtual_start: '2015-01-01T00:00:00'
                  priority: 1
                  rotation_turn_length_seconds: 604800
```

`salt.states.pagerduty_schedule.`**`absent`**(*profile='pagerduty', subdomain=None, api_key=None, **kwargs*)

Ensure that a pagerduty schedule does not exist. Name can be pagerduty schedule id or pagerduty schedule name.

`salt.states.pagerduty_schedule.`**`present`**(*profile='pagerduty', subdomain=None, api_key=None, **kwargs*)

Ensure that a pagerduty schedule exists. This method accepts as args everything defined in https://developer.pagerduty.com/documentation/rest/schedules/create. This means that most arguments are in a dict called "schedule."

User id's can be pagerduty id, or name, or email address.

21.19.202 salt.states.pagerduty_service

Manage PagerDuty services

Escalation policies can be referenced by pagerduty ID or by namea.

For example:

```
ensure test service
    pagerduty_service.present:
        - name: 'my service'
        - escalation_policy_id: 'my escalation policy'
        - type: nagios
```

`salt.states.pagerduty_service.`**`absent`**(*profile='pagerduty', subdomain=None, api_key=None, **kwargs*)

Ensure a pagerduty service does not exist. Name can be the service name or pagerduty service id.

`salt.states.pagerduty_service.`**`present`**(*profile='pagerduty', subdomain=None, api_key=None, **kwargs*)

Ensure pagerduty service exists. This method accepts as arguments everything defined in https://developer.pagerduty.com/documentation/rest/services/create

Note that many arguments are mutually exclusive, depending on the "type" argument.

Examples:

```
# create a PagerDuty email service at test-email@DOMAIN.pagerduty.com
ensure generic email service exists:
    pagerduty_service.present:
        - name: my email service
        - service:
          description: "email service controlled by salt"
          escalation_policy_id: "my escalation policy"
          type: "generic_email"
          service_key: "test-email"
```

```
# create a pagerduty service using cloudwatch integration
ensure my cloudwatch service exists:
    pagerduty_service.present:
        - name: my cloudwatch service
        - service:
          escalation_policy_id: "my escalation policy"
          type: aws_cloudwatch
          description: "my cloudwatch service controlled by salt"
```

21.19.203 salt.states.pagerduty_user

Manage PagerDuty users.

Example

ensure bruce test user 1:

pagerduty.user_present:

- name: 'Bruce TestUser1'
- email: bruce+test1@lyft.com
- requester_id: P1GV5NT

`salt.states.pagerduty_user.`**`absent`**(*profile='pagerduty', subdomain=None, api_key=None, **kwargs*)

Ensure pagerduty user does not exist. Name can be pagerduty id, email address, or user name.

`salt.states.pagerduty_user.`**`present`**(*profile='pagerduty', subdomain=None, api_key=None, **kwargs*)

Ensure pagerduty user exists. Arguments match those supported by https://developer.pagerduty.com/documentation/rest/users/create.

21.19.204 salt.states.panos

A state module to manage Palo Alto network devices.

codeauthor `Spencer Ervin <spencer_ervin@hotmail.com>`

maturity new

depends none

platform unix

About

This state module was designed to handle connections to a Palo Alto based firewall. This module relies on the Palo Alto proxy module to interface with the devices.

This state module is designed to give extreme flexibility in the control over XPATH values on the PANOS device. It exposes the core XML API commands and allows state modules to chain complex XPATH commands.

Below is an example of how to construct a security rule and move to the top of the policy. This will take a config lock to prevent execution during the operation, then remove the lock. After the XPATH has been deployed, it will commit to the device.

```
panos/takelock:
    panos.add_config_lock
panos/service_tcp_22:
    panos.set_config:
        - xpath: /config/devices/entry[@name='localhost.localdomain']/vsys/
↪entry[@name='vsys1']/service
        - value: <entry name='tcp-22'><protocol><tcp><port>22</port></tcp></protocol>
↪</entry>
        - commit: False
panos/create_rule1:
    panos.set_config:
        - xpath: /config/devices/entry[@name='localhost.localdomain']/vsys/
↪entry[@name='vsys1']/rulebase/security/rules
        - value: '
          <entry name="rule1">
            <from><member>trust</member></from>
            <to><member>untrust</member></to>
            <source><member>10.0.0.1</member></source>
            <destination><member>10.0.1.1</member></destination>
            <service><member>tcp-22</member></service>
            <application><member>any</member></application>
            <action>allow</action>
            <disabled>no</disabled>
          </entry>'
        - commit: False
panos/moveruletop:
    panos.move_config:
        - xpath: /config/devices/entry[@name='localhost.localdomain']/vsys/
↪entry[@name='vsys1']/rulebase/security/rules/entry[@name='rule1']
        - where: top
        - commit: False
panos/removelock:
    panos.remove_config_lock
panos/commit:
    panos.commit
```

Version Specific Configurations

Palo Alto devices running different versions will have different supported features and different command structures. In order to account for this, the proxy module can be leveraged to check if the panos device is at a specific revision level.

The proxy['panos.is_required_version'] method will check if a panos device is currently running a version equal or greater than the passed version. For example, proxy['panos.is_required_version']('7.0.0') would match both 7.1.0 and 8.0.0.

```
{% if proxy['panos.is_required_version']('8.0.0') %}
panos/deviceconfig/system/motd-and-banner:
  panos.set_config:
    - xpath: /config/devices/entry[@name='localhost.localdomain']/deviceconfig/system/
↪motd-and-banner
    - value: |
      <banner-header>BANNER TEXT</banner-header>
      <banner-header-color>color2</banner-header-color>
      <banner-header-text-color>color18</banner-header-text-color>
      <banner-header-footer-match>yes</banner-header-footer-match>
    - commit: False
{% endif %}
```

See also:

Palo Alto Proxy Module

`salt.states.panos.`**`add_config_lock`**(*name*)

Prevent other users from changing configuration until the lock is released.

name: The name of the module function to execute.

SLS Example:

```
panos/takelock:
    panos.add_config_lock
```

`salt.states.panos.`**`address_exists`**(*name, addressname=None, vsys=1, ipnetmask=None, iprange=None, fqdn=None, description=None, commit=False*)

Ensures that an address object exists in the configured state. If it does not exist or is not configured with the specified attributes, it will be adjusted to match the specified values.

This module will only process a single address type (ip-netmask, ip-range, or fqdn). It will process the specified value if the following order: ip-netmask, ip-range, fqdn. For proper execution, only specify a single address type.

name: The name of the module function to execute.

addressname(str): The name of the address object. The name is case-sensitive and can have up to 31 characters, which an be letters, numbers, spaces, hyphens, and underscores. The name must be unique on a firewall and, on Panorama, unique within its device group and any ancestor or descendant device groups.

vsys(str): The string representation of the VSYS ID. Defaults to VSYS 1.

ipnetmask(str): The IPv4 or IPv6 address or IP address range using the format ip_address/mask or ip_address where the mask is the number of significant binary digits used for the network portion of the address. Ideally, for IPv6, you specify only the network portion, not the host portion.

iprange(str): A range of addresses using the format ip_address–ip_address where both addresses can be IPv4 or both can be IPv6.

fqdn(str): A fully qualified domain name format. The FQDN initially resolves at commit time. Entries are subsequently refreshed when the firewall performs a check every 30 minutes; all changes in the IP address for the entries are picked up at the refresh cycle.

description(str): A description for the policy (up to 255 characters).

commit(bool): If true the firewall will commit the changes, if false do not commit changes.

SLS Example:

```
panos/address/h-10.10.10.10:
    panos.address_exists:
      - addressname: h-10.10.10.10
      - vsys: 1
      - ipnetmask: 10.10.10.10
      - commit: False

panos/address/10.0.0.1-10.0.0.50:
    panos.address_exists:
      - addressname: r-10.0.0.1-10.0.0.50
      - vsys: 1
      - iprange: 10.0.0.1-10.0.0.50
      - commit: False

panos/address/foo.bar.com:
    panos.address_exists:
      - addressname: foo.bar.com
      - vsys: 1
      - fqdn: foo.bar.com
      - description: My fqdn object
      - commit: False
```

`salt.states.panos.`**`address_group_exists`**(*name*, *groupname=None*, *vsys=1*, *members=None*, *description=None*, *commit=False*)

Ensures that an address group object exists in the configured state. If it does not exist or is not configured with the specified attributes, it will be adjusted to match the specified values.

This module will enforce group membership. If a group exists and contains members this state does not include, those members will be removed and replaced with the specified members in the state.

name: The name of the module function to execute.

groupname(str): The name of the address group object. The name is case-sensitive and can have up to 31 characters, which an be letters, numbers, spaces, hyphens, and underscores. The name must be unique on a firewall and, on Panorama, unique within its device group and any ancestor or descendant device groups.

vsys(str): The string representation of the VSYS ID. Defaults to VSYS 1.

members(str, list): The members of the address group. These must be valid address objects or address groups on the system that already exist prior to the execution of this state.

description(str): A description for the policy (up to 255 characters).

commit(bool): If true the firewall will commit the changes, if false do not commit changes.

SLS Example:

```
panos/address-group/my-group:
    panos.address_group_exists:
      - groupname: my-group
      - vsys: 1
```

(continues on next page)

(continued from previous page)

```
    - members:
      - my-address-object
      - my-other-address-group
    - description: A group that needs to exist
    - commit: False
```

salt.states.panos.**clone_config**(*name*, *xpath=None*, *newname=None*, *commit=False*)

Clone a specific XPATH and set it to a new name.

name: The name of the module function to execute.

xpath(str): The XPATH of the configuration API tree to clone.

newname(str): The new name of the XPATH clone.

commit(bool): If true the firewall will commit the changes, if false do not commit changes.

SLS Example:

```
panos/clonerule:
    panos.clone_config:
      - xpath: /config/devices/entry/vsys/entry[@name='vsys1']/rulebase/security/
↪rules&from=/config/devices/
      entry/vsys/entry[@name='vsys1']/rulebase/security/rules/entry[@name='rule1']
      - value: rule2
      - commit: True
```

salt.states.panos.**commit_config**(*name*)

Commits the candidate configuration to the running configuration.

name: The name of the module function to execute.

SLS Example:

```
panos/commit:
    panos.commit_config
```

salt.states.panos.**delete_config**(*name*, *xpath=None*, *commit=False*)

Deletes a Palo Alto XPATH to a specific value.

Use the xpath parameter to specify the location of the object to be deleted.

name: The name of the module function to execute.

xpath(str): The XPATH of the configuration API tree to control.

commit(bool): If true the firewall will commit the changes, if false do not commit changes.

SLS Example:

```
panos/deletegroup:
    panos.delete_config:
      - xpath: /config/devices/entry/vsys/entry[@name='vsys1']/address-group/
↪entry[@name='test']
      - commit: True
```

salt.states.panos.**download_software**(*name*, *version=None*, *synch=False*, *check=False*)

Ensures that a software version is downloaded.

name: The name of the module function to execute.

version(str): The software version to check. If this version is not already downloaded, it will attempt to download the file from Palo Alto.

synch(bool): If true, after downloading the file it will be synched to its peer.

check(bool): If true, the PANOS device will first attempt to pull the most recent software inventory list from Palo Alto.

SLS Example:

```
panos/version8.0.0:
    panos.download_software:
      - version: 8.0.0
      - synch: False
      - check: True
```

`salt.states.panos.`**`edit_config`**(*name*, *xpath=None*, *value=None*, *commit=False*)

Edits a Palo Alto XPATH to a specific value. This will always overwrite the existing value, even if it is not changed.

You can replace an existing object hierarchy at a specified location in the configuration with a new value. Use the xpath parameter to specify the location of the object, including the node to be replaced.

This is the recommended state to enforce configurations on a xpath.

name: The name of the module function to execute.

xpath(str): The XPATH of the configuration API tree to control.

value(str): The XML value to edit. This must be a child to the XPATH.

commit(bool): If true the firewall will commit the changes, if false do not commit changes.

SLS Example:

```
panos/addressgroup:
    panos.edit_config:
      - xpath: /config/devices/entry/vsys/entry[@name='vsys1']/address-group/
↪entry[@name='test']
      - value: <static><entry name='test'><member>abc</member><member>xyz</member>
↪</entry></static>
      - commit: True
```

`salt.states.panos.`**`move_config`**(*name*, *xpath=None*, *where=None*, *dst=None*, *commit=False*)

Moves a XPATH value to a new location.

Use the xpath parameter to specify the location of the object to be moved, the where parameter to specify type of move, and dst parameter to specify the destination path.

name: The name of the module function to execute.

xpath(str): The XPATH of the configuration API tree to move.

where(str): The type of move to execute. Valid options are after, before, top, bottom. The after and before options will require the dst option to specify the destination of the action. The top action will move the XPATH to the top of its structure. The botoom action will move the XPATH to the bottom of its structure.

dst(str): Optional. Specifies the destination to utilize for a move action. This is ignored for the top or bottom action.

commit(bool): If true the firewall will commit the changes, if false do not commit changes. If the operation is not successful, it will not commit.

SLS Example:

```
panos/moveruletop:
    panos.move_config:
      - xpath: /config/devices/entry/vsys/entry[@name='vsys1']/rulebase/security/
↪rules/entry[@name='rule1']
      - where: top
      - commit: True

panos/moveruleafter:
    panos.move_config:
      - xpath: /config/devices/entry/vsys/entry[@name='vsys1']/rulebase/security/
↪rules/entry[@name='rule1']
      - where: after
      - dst: rule2
      - commit: True
```

`salt.states.panos.`**`remove_config_lock`**(*name*)

Release config lock previously held.

name: The name of the module function to execute.

SLS Example:

```
panos/takelock:
    panos.remove_config_lock
```

`salt.states.panos.`**`rename_config`**(*name*, *xpath=None*, *newname=None*, *commit=False*)

Rename a Palo Alto XPATH to a specific value. This will always rename the value even if a change is not needed.

name: The name of the module function to execute.

xpath(str): The XPATH of the configuration API tree to control.

newname(str): The new name of the XPATH value.

commit(bool): If true the firewall will commit the changes, if false do not commit changes.

SLS Example:

```
panos/renamegroup:
    panos.rename_config:
      - xpath: /config/devices/entry/vsys/entry[@name='vsys1']/address/
↪entry[@name='old_address']
      - value: new_address
      - commit: True
```

`salt.states.panos.`**`security_rule_exists`**(*name*, *rulename=None*, *vsys='1'*, *action=None*, *disabled=None*, *sourcezone=None*, *destinationzone=None*, *source=None*, *destination=None*, *application=None*, *service=None*, *description=None*, *logsetting=None*, *logstart=None*, *logend=None*, *negatesource=None*, *negatedestination=None*, *profilegroup=None*, *datafilter=None*, *fileblock=None*, *spyware=None*, *urlfilter=None*, *virus=None*, *vulnerability=None*, *wildfire=None*, *move=None*, *movetarget=None*, *commit=False*)

Ensures that a security rule exists on the device. Also, ensure that all configurations are set appropriately.

This method will create the rule if it does not exist. If the rule does exist, it will ensure that the configurations are set appropriately.

If the rule does not exist and is created, any value that is not provided will be provided as the default. The action, to, from, source, destination, application, and service fields are mandatory and must be provided.

This will enforce the exact match of the rule. For example, if the rule is currently configured with the log-end option, but this option is not specified in the state method, it will be removed and reset to the system default.

It is strongly recommended to specify all options to ensure proper operation.

When defining the profile group settings, the device can only support either a profile group or individual settings. If both are specified, the profile group will be preferred and the individual settings are ignored. If neither are specified, the value will be set to system default of none.

name: The name of the module function to execute.

rulename(str): The name of the security rule. The name is case-sensitive and can have up to 31 characters, which can be letters, numbers, spaces, hyphens, and underscores. The name must be unique on a firewall and, on Panorama, unique within its device group and any ancestor or descendant device groups.

vsys(str): The string representation of the VSYS ID. Defaults to VSYS 1.

action(str): The action that the security rule will enforce. Valid options are: allow, deny, drop, reset-client, reset-server, reset-both.

disabled(bool): Controls if the rule is disabled. Set 'True' to disable and 'False' to enable.

sourcezone(str, list): The source zone(s). The value 'any' will match all zones.

destinationzone(str, list): The destination zone(s). The value 'any' will match all zones.

source(str, list): The source address(es). The value 'any' will match all addresses.

destination(str, list): The destination address(es). The value 'any' will match all addresses.

application(str, list): The application(s) matched. The value 'any' will match all applications.

service(str, list): The service(s) matched. The value 'any' will match all services. The value 'application-default' will match based upon the application defined ports.

description(str): A description for the policy (up to 255 characters).

logsetting(str): The name of a valid log forwarding profile.

logstart(bool): Generates a traffic log entry for the start of a session (disabled by default).

logend(bool): Generates a traffic log entry for the end of a session (enabled by default).

negatesource(bool): Match all but the specified source addresses.

negatedestination(bool): Match all but the specified destination addresses.

profilegroup(str): A valid profile group name.

datafilter(str): A valid data filter profile name. Ignored with the profilegroup option set.

fileblock(str): A valid file blocking profile name. Ignored with the profilegroup option set.

spyware(str): A valid spyware profile name. Ignored with the profilegroup option set.

urlfilter(str): A valid URL filtering profile name. Ignored with the profilegroup option set.

virus(str): A valid virus profile name. Ignored with the profilegroup option set.

vulnerability(str): A valid vulnerability profile name. Ignored with the profilegroup option set.

wildfire(str): A valid vulnerability profile name. Ignored with the profilegroup option set.

move(str): An optional argument that ensure the rule is moved to a specific location. Valid options are 'top', 'bottom', 'before', or 'after'. The 'before' and 'after' options require the use of the 'movetarget' argument to define the location of the move request.

movetarget(str): An optional argument that defines the target of the move operation if the move argument is set to 'before' or 'after'.

commit(bool): If true the firewall will commit the changes, if false do not commit changes.

SLS Example:

```
panos/rulebase/security/rule01:
    panos.security_rule_exists:
      - rulename: rule01
      - vsys: 1
      - action: allow
      - disabled: False
      - sourcezone: untrust
      - destinationzone: trust
      - source:
        - 10.10.10.0/24
        - 1.1.1.1
      - destination:
        - 2.2.2.2-2.2.2.4
      - application:
        - any
      - service:
        - tcp-25
      - description: My test security rule
      - logsetting: logprofile
      - logstart: False
      - logend: True
      - negatesource: False
      - negatedestination: False
      - profilegroup: myprofilegroup
      - move: top
      - commit: False

panos/rulebase/security/rule01:
    panos.security_rule_exists:
      - rulename: rule01
      - vsys: 1
      - action: allow
      - disabled: False
      - sourcezone: untrust
      - destinationzone: trust
      - source:
        - 10.10.10.0/24
        - 1.1.1.1
      - destination:
        - 2.2.2.2-2.2.2.4
      - application:
        - any
      - service:
        - tcp-25
      - description: My test security rule
      - logsetting: logprofile
      - logstart: False
      - logend: False
```

(continues on next page)

(continued from previous page)

```
        - datafilter: foobar
        - fileblock: foobar
        - spyware: foobar
        - urlfilter: foobar
        - virus: foobar
        - vulnerability: foobar
        - wildfire: foobar
        - move: after
        - movetarget: rule02
        - commit: False
```

`salt.states.panos.`**`service_exists`**(*name*, *servicename=None*, *vsys=1*, *protocol=None*, *port=None*, *description=None*, *commit=False*)

Ensures that a service object exists in the configured state. If it does not exist or is not configured with the specified attributes, it will be adjusted to match the specified values.

name: The name of the module function to execute.

servicename(str): The name of the security object. The name is case-sensitive and can have up to 31 characters, which an be letters, numbers, spaces, hyphens, and underscores. The name must be unique on a firewall and, on Panorama, unique within its device group and any ancestor or descendant device groups.

vsys(str): The string representation of the VSYS ID. Defaults to VSYS 1.

protocol(str): The protocol that is used by the service object. The only valid options are tcp and udp.

port(str): The port number that is used by the service object. This can be specified as a single integer or a valid range of ports.

description(str): A description for the policy (up to 255 characters).

commit(bool): If true the firewall will commit the changes, if false do not commit changes.

SLS Example:

```
panos/service/tcp-80:
    panos.service_exists:
      - servicename: tcp-80
      - vsys: 1
      - protocol: tcp
      - port: 80
      - description: Hypertext Transfer Protocol
      - commit: False

panos/service/udp-500-550:
    panos.service_exists:
      - servicename: udp-500-550
      - vsys: 3
      - protocol: udp
      - port: 500-550
      - commit: False
```

`salt.states.panos.`**`service_group_exists`**(*name*, *groupname=None*, *vsys=1*, *members=None*, *description=None*, *commit=False*)

Ensures that a service group object exists in the configured state. If it does not exist or is not configured with the specified attributes, it will be adjusted to match the specified values.

This module will enforce group membership. If a group exists and contains members this state does not include, those members will be removed and replaced with the specified members in the state.

name: The name of the module function to execute.

groupname(str): The name of the service group object. The name is case-sensitive and can have up to 31 characters, which an be letters, numbers, spaces, hyphens, and underscores. The name must be unique on a firewall and, on Panorama, unique within its device group and any ancestor or descendant device groups.

vsys(str): The string representation of the VSYS ID. Defaults to VSYS 1.

members(str, list): The members of the service group. These must be valid service objects or service groups on the system that already exist prior to the execution of this state.

description(str): A description for the policy (up to 255 characters).

commit(bool): If true the firewall will commit the changes, if false do not commit changes.

SLS Example:

```
panos/service-group/my-group:
    panos.service_group_exists:
      - groupname: my-group
      - vsys: 1
      - members:
        - tcp-80
        - custom-port-group
      - description: A group that needs to exist
      - commit: False
```

`salt.states.panos.`**`set_config`**(*name*, *xpath=None*, *value=None*, *commit=False*)

Sets a Palo Alto XPATH to a specific value. This will always overwrite the existing value, even if it is not changed.

You can add or create a new object at a specified location in the configuration hierarchy. Use the xpath parameter to specify the location of the object in the configuration

name: The name of the module function to execute.

xpath(str): The XPATH of the configuration API tree to control.

value(str): The XML value to set. This must be a child to the XPATH.

commit(bool): If true the firewall will commit the changes, if false do not commit changes.

SLS Example:

```
panos/hostname:
    panos.set_config:
      - xpath: /config/devices/entry[@name='localhost.localdomain']/deviceconfig/
↪system
      - value: <hostname>foobar</hostname>
      - commit: True
```

21.19.205 salt.states.pcs module

Management of Pacemaker/Corosync clusters with PCS

A state module to manage Pacemaker/Corosync clusters with the Pacemaker/Corosync configuration system (PCS)

New in version 2016.110.

depends pcs

Walkthrough of a complete PCS cluster setup: http://clusterlabs.org/doc/en-US/Pacemaker/1.1/html/Clusters_from_Scratch/

Requirements: PCS is installed, pcs service is started and the password for the hacluster user is set and known.

Remark on the cibname variable used in the examples: The use of the cibname variable is optional. Use it only if you want to deploy your changes into a cibfile first and then push it. This makes only sense if you want to deploy multiple changes (which require each other) at once to the cluster.

At first the cibfile must be created:

```
mysql_pcs__cib_present_cib_for_galera:
    pcs.cib_present:
        - cibname: cib_for_galera
        - scope: None
        - extra_args: None
```

Then the cibfile can be modified by creating resources (creating only 1 resource for demonstration, see also 7.):

```
mysql_pcs__resource_present_galera:
    pcs.resource_present:
        - resource_id: galera
        - resource_type: "ocf:heartbeat:galera"
        - resource_options:
            - 'wsrep_cluster_address=gcomm://node1.example.org,node2.example.org,
↪node3.example.org'
            - '--master'
        - cibname: cib_for_galera
```

After modifying the cibfile, it can be pushed to the live CIB in the cluster:

```
mysql_pcs__cib_pushed_cib_for_galera:
    pcs.cib_pushed:
        - cibname: cib_for_galera
        - scope: None
        - extra_args: None
```

Create a cluster from scratch:

1. Authorize nodes to each other:

```
pcs_auth__auth:
    pcs.auth:
        - nodes:
            - node1.example.com
            - node2.example.com
        - pcsuser: hacluster
        - pcspasswd: hoonetorg
        - extra_args: []
```

2. Do the initial cluster setup:

```
pcs_setup__setup:
    pcs.cluster_setup:
        - nodes:
            - node1.example.com
            - node2.example.com
        - pcsclustername: pcscluster
        - extra_args:
```

(continues on next page)

(continued from previous page)

```
            - '--start'
            - '--enable'
```

3. Optional: Set cluster properties:

```
pcs_properties__prop_has_value_no-quorum-policy:
    pcs.prop_has_value:
        - prop: no-quorum-policy
        - value: ignore
        - cibname: cib_for_cluster_settings
```

4. Optional: Set resource defaults:

```
pcs_properties__resource_defaults_to_resource-stickiness:
    pcs.resource_defaults_to:
        - default: resource-stickiness
        - value: 100
        - cibname: cib_for_cluster_settings
```

5. Optional: Set resource op defaults:

```
pcs_properties__resource_op_defaults_to_monitor-interval:
    pcs.resource_op_defaults_to:
        - op_default: monitor-interval
        - value: 60s
        - cibname: cib_for_cluster_settings
```

6. Configure Fencing (!is often not optional on production ready cluster!):

```
pcs_stonith__created_eps_fence:
    pcs.stonith_present:
        - stonith_id: eps_fence
        - stonith_device_type: fence_eps
        - stonith_device_options:
            - 'pcmk_host_map=node1.example.org:01;node2.example.org:02'
            - 'ipaddr=myepsdevice.example.org'
            - 'power_wait=5'
            - 'verbose=1'
            - 'debug=/var/log/pcsd/eps_fence.log'
            - 'login=hidden'
            - 'passwd=hoonetorg'
        - cibname: cib_for_stonith
```

7. Add resources to your cluster:

```
mysql_pcs__resource_present_galera:
    pcs.resource_present:
        - resource_id: galera
        - resource_type: "ocf:heartbeat:galera"
        - resource_options:
            - 'wsrep_cluster_address=gcomm://node1.example.org,node2.example.org,
↪node3.example.org'
             - '--master'
         - cibname: cib_for_galera
```

8. Optional: Add constraints (locations, colocations, orders):

```
haproxy_pcs__constraint_present_colocation-vip_galera-haproxy-clone-INFINITY:
    pcs.constraint_present:
        - constraint_id: colocation-vip_galera-haproxy-clone-INFINITY
        - constraint_type: colocation
        - constraint_options:
            - 'add'
            - 'vip_galera'
            - 'with'
            - 'haproxy-clone'
        - cibname: cib_for_haproxy
```

New in version 2016.3.0.

`salt.states.pcs.`**`auth`**(*name*, *nodes*, *pcsuser='hacluster'*, *pcspasswd='hacluster'*, *extra_args=None*)

Ensure all nodes are authorized to the cluster

name Irrelevant, not used (recommended: pcs_auth__auth)

nodes a list of nodes which should be authorized to the cluster

pcsuser user for communication with pcs (default: hacluster)

pcspasswd password for pcsuser (default: hacluster)

extra_args list of extra args for the 'pcs cluster auth' command

Example:

```
pcs_auth__auth:
    pcs.auth:
        - nodes:
            - node1.example.com
            - node2.example.com
        - pcsuser: hacluster
        - pcspasswd: hoonetorg
        - extra_args: []
```

`salt.states.pcs.`**`cib_present`**(*name*, *cibname*, *scope=None*, *extra_args=None*)

Ensure that a CIB-file with the content of the current live CIB is created

Should be run on one cluster node only (there may be races)

name Irrelevant, not used (recommended: {{formulaname}}__cib_present_{{cibname}})

cibname name/path of the file containing the CIB

scope specific section of the CIB (default:

extra_args additional options for creating the CIB-file

Example:

```
mysql_pcs__cib_present_cib_for_galera:
    pcs.cib_present:
        - cibname: cib_for_galera
        - scope: None
        - extra_args: None
```

`salt.states.pcs.`**`cib_pushed`**(*name*, *cibname*, *scope=None*, *extra_args=None*)

Ensure that a CIB-file is pushed if it is changed since the creation of it with pcs.cib_present

Should be run on one cluster node only (there may be races)

name Irrelevant, not used (recommended: {{formulaname}}__cib_pushed_{{cibname}})

cibname name/path of the file containing the CIB

scope specific section of the CIB

extra_args additional options for creating the CIB-file

Example:

```
mysql_pcs__cib_pushed_cib_for_galera:
    pcs.cib_pushed:
        - cibname: cib_for_galera
        - scope: None
        - extra_args: None
```

`salt.states.pcs.`**`cluster_node_present`**(*name*, *node*, *extra_args=None*)

Add a node to the Pacemaker cluster via PCS Should be run on one cluster node only (there may be races) Can only be run on a already setup/added node

name Irrelevant, not used (recommended: pcs_setup__node_add_{{node}})
node node that should be added
extra_args list of extra args for the 'pcs cluster node add' command
Example:

```
pcs_setup__node_add_node1.example.com:
    pcs.cluster_node_present:
        - node: node1.example.com
        - extra_args:
            - '--start'
            - '--enable'
```

`salt.states.pcs.`**`cluster_setup`**(*name*, *nodes*, *pcsclustername='pcscluster'*, *extra_args=None*)

Setup Pacemaker cluster on nodes. Should be run on one cluster node only (there may be races)

name Irrelevant, not used (recommended: pcs_setup__setup)
nodes a list of nodes which should be set up
pcsclustername Name of the Pacemaker cluster
extra_args list of extra args for the 'pcs cluster setup' command
Example:

```
pcs_setup__setup:
    pcs.cluster_setup:
        - nodes:
            - node1.example.com
            - node2.example.com
        - pcsclustername: pcscluster
        - extra_args:
            - '--start'
            - '--enable'
```

`salt.states.pcs.`**`constraint_present`**(*name*, *constraint_id*, *constraint_type*, *constraint_options=None*, *cibname=None*)

Ensure that a constraint is created

Should be run on one cluster node only (there may be races) Can only be run on a node with a functional pacemaker/corosync

name Irrelevant, not used (recommended: {{formulaname}}__constraint_present_{{constraint_id}})
constraint_id name for the constraint (try first to create manually to find out the autocreated name)
constraint_type constraint type (location, colocation, order)
constraint_options options for creating the constraint
cibname use a cached CIB-file named like cibname instead of the live CIB
Example:

```
haproxy_pcs__constraint_present_colocation-vip_galera-haproxy-clone-INFINITY:
    pcs.constraint_present:
        - constraint_id: colocation-vip_galera-haproxy-clone-INFINITY
        - constraint_type: colocation
```

(continues on next page)

(continued from previous page)

```
        - constraint_options:
            - 'add'
            - 'vip_galera'
            - 'with'
            - 'haproxy-clone'
        - cibname: cib_for_haproxy
```

`salt.states.pcs.`**`prop_has_value`**(*name*, *prop*, *value*, *extra_args=None*, *cibname=None*)

Ensure that a property in the cluster is set to a given value

Should be run on one cluster node only (there may be races)

name Irrelevant, not used (recommended: pcs_properties__prop_has_value_{{prop}})

prop name of the property

value value of the property

extra_args additional options for the pcs property command

cibname use a cached CIB-file named like cibname instead of the live CIB

Example:

```
pcs_properties__prop_has_value_no-quorum-policy:
    pcs.prop_has_value:
        - prop: no-quorum-policy
        - value: ignore
        - cibname: cib_for_cluster_settings
```

`salt.states.pcs.`**`resource_defaults_to`**(*name*, *default*, *value*, *extra_args=None*, *cibname=None*)

Ensure a resource default in the cluster is set to a given value

Should be run on one cluster node only (there may be races) Can only be run on a node with a functional pacemaker/corosync

name Irrelevant, not used (recommended: pcs_properties__resource_defaults_to_{{default}})

default name of the default resource property

value value of the default resource property

extra_args additional options for the pcs command

cibname use a cached CIB-file named like cibname instead of the live CIB

Example:

```
pcs_properties__resource_defaults_to_resource-stickiness:
    pcs.resource_defaults_to:
        - default: resource-stickiness
        - value: 100
        - cibname: cib_for_cluster_settings
```

`salt.states.pcs.`**`resource_op_defaults_to`**(*name*, *op_default*, *value*, *extra_args=None*, *cibname=None*)

Ensure a resource operation default in the cluster is set to a given value

Should be run on one cluster node only (there may be races) Can only be run on a node with a functional pacemaker/corosync

name Irrelevant, not used (recommended: pcs_properties__resource_op_defaults_to_{{op_default}})

op_default name of the operation default resource property

value value of the operation default resource property

extra_args additional options for the pcs command

cibname use a cached CIB-file named like cibname instead of the live CIB

Example:

```
pcs_properties__resource_op_defaults_to_monitor-interval:
    pcs.resource_op_defaults_to:
        - op_default: monitor-interval
        - value: 60s
        - cibname: cib_for_cluster_settings
```

salt.states.pcs.**resource_present**(*name*, *resource_id*, *resource_type*, *resource_options=None*, *cibname=None*)

Ensure that a resource is created

Should be run on one cluster node only (there may be races) Can only be run on a node with a functional pacemaker/corosync

name Irrelevant, not used (recommended: {{formulaname}}__resource_present_{{resource_id}})
resource_id name for the resource
resource_type resource type (f.e. ocf:heartbeat:IPaddr2 or VirtualIP)
resource_options additional options for creating the resource
cibname use a cached CIB-file named like cibname instead of the live CIB
Example:

```
mysql_pcs__resource_present_galera:
    pcs.resource_present:
        - resource_id: galera
        - resource_type: "ocf:heartbeat:galera"
        - resource_options:
            - 'wsrep_cluster_address=gcomm://node1.example.org,node2.example.org,
↪node3.example.org'
            - '--master'
        - cibname: cib_for_galera
```

salt.states.pcs.**stonith_present**(*name*, *stonith_id*, *stonith_device_type*, *stonith_device_options=None*, *cibname=None*)

Ensure that a fencing resource is created

Should be run on one cluster node only (there may be races) Can only be run on a node with a functional pacemaker/corosync

name Irrelevant, not used (recommended: pcs_stonith__created_{{stonith_id}})
stonith_id name for the stonith resource
stonith_device_type name of the stonith agent fence_eps, fence_xvm f.e.
stonith_device_options additional options for creating the stonith resource
cibname use a cached CIB-file named like cibname instead of the live CIB
Example:

```
pcs_stonith__created_eps_fence:
    pcs.stonith_present:
        - stonith_id: eps_fence
        - stonith_device_type: fence_eps
        - stonith_device_options:
            - 'pcmk_host_map=node1.example.org:01;node2.example.org:02'
            - 'ipaddr=myepsdevice.example.org'
            - 'power_wait=5'
            - 'verbose=1'
            - 'debug=/var/log/pcsd/eps_fence.log'
            - 'login=hidden'
            - 'passwd=hoonetorg'
        - cibname: cib_for_stonith
```

21.19.206 salt.states.pecl

Installation of PHP Extensions Using pecl

These states manage the installed pecl extensions. Note that php-pear must be installed for these states to be available, so pecl states should include a requisite to a pkg.installed state for the package which provides pecl (`php-pear` in most cases). Example:

```
php-pear:
  pkg.installed

mongo:
  pecl.installed:
    - require:
      - pkg: php-pear
```

`salt.states.pecl.`**`installed`**`(`*name, version=None, defaults=False, force=False, preferred_state='stable'*`)`

New in version 0.17.0.

Make sure that a pecl extension is installed.

name The pecl extension name to install

version The pecl extension version to install. This option may be ignored to install the latest stable version.

defaults Use default answers for extensions such as pecl_http which ask questions before installation. Without this option, the pecl.installed state will hang indefinitely when trying to install these extensions.

force Whether to force the installed version or not

preferred_state The pecl extension state to install

`salt.states.pecl.`**`removed`**`(`*name*`)`

Make sure that a pecl extension is not installed.

name The pecl extension name to uninstall

21.19.207 salt.states.pdbedit

Manage accounts in Samba's passdb using pdbedit

maintainer Jorge Schrauwen <sjorge@blackdot.be>

maturity new

depends pdbedit

platform posix

New in version 2017.7.0.

```
wash:
  pdbedit.absent

kaylee:
  pdbedit.managed:
    - password: A70C708517B5DD0EDB67714FE25336EB
    - password_hashed: True
    - drive: 'X:'
    - homedir: '\\serenity\mechanic\profile'
```

`salt.states.pdbedit.`**`absent`**`(`*name*`)`

Ensure user account is absent

name [string] username

`salt.states.pdbedit.`**`managed`**`(`*name*, ***kwargs*`)`

Manage user account

login [string] login name
password [string] password
password_hashed [boolean] set if password is a nt hash instead of plain text
domain [string] users domain
profile [string] profile path
script [string] logon script
drive [string] home drive
homedir [string] home directory
fullname [string] full name
account_desc [string] account description
machine_sid [string] specify the machines new primary group SID or rid
user_sid [string] specify the users new primary group SID or rid
account_control [string] specify user account control properties

> **Note:** Only the following can be set: - N: No password required - D: Account disabled - H: Home directory required - L: Automatic Locking - X: Password does not expire

reset_login_hours [boolean] reset the users allowed logon hours
reset_bad_password_count [boolean] reset the stored bad login counter

`salt.states.pdbedit.`**`present`**`(`*name*, ***kwargs*`)`

Alias for pdbedit.managed

21.19.208 salt.states.pip_state

Installation of Python Packages Using pip

These states manage system installed python packages. Note that pip must be installed for these states to be available, so pip states should include a requisite to a pkg.installed state for the package which provides pip (`python-pip` in most cases). Example:

```yaml
python-pip:
  pkg.installed

virtualenvwrapper:
  pip.installed:
    - require:
      - pkg: python-pip
```

`salt.states.pip_state.installed`(*name, pkgs=None, pip_bin=None, requirements=None, bin_env=None, use_wheel=False, no_use_wheel=False, log=None, proxy=None, timeout=None, repo=None, editable=None, find_links=None, index_url=None, extra_index_url=None, no_index=False, mirrors=None, build=None, target=None, download=None, download_cache=None, source=None, upgrade=False, force_reinstall=False, ignore_installed=False, exists_action=None, no_deps=False, no_install=False, no_download=False, install_options=None, global_options=None, user=None, cwd=None, pre_releases=False, cert=None, allow_all_external=False, allow_external=None, allow_unverified=None, process_dependency_links=False, env_vars=None, use_vt=False, trusted_host=None, no_cache_dir=False, cache_dir=None, no_binary=None, extra_args=None, **kwargs*)

Make sure the package is installed

name The name of the python package to install. You can also specify version numbers here using the standard operators `==, >=, <=`. If `requirements` is given, this parameter will be ignored.

Example:

```
django:
  pip.installed:
    - name: django >= 1.6, <= 1.7
    - require:
      - pkg: python-pip
```

This will install the latest Django version greater than 1.6 but less than 1.7.

requirements Path to a pip requirements file. If the path begins with salt:// the file will be transferred from the master file server.

user The user under which to run pip

use_wheel [False] Prefer wheel archives (requires pip>=1.4)

no_use_wheel [False] Force to not use wheel archives (requires pip>=1.4)

no_binary Force to not use binary packages (requires pip >= 7.0.0) Accepts either :all: to disable all binary packages, :none: to empty the set, or a list of one or more packages

Example:

```
django:
  pip.installed:
    - no_binary: ':all:'

flask:
  pip.installed:
    - no_binary:
      - itsdangerous
      - click
```

log Log file where a complete (maximum verbosity) record will be kept

proxy Specify a proxy in the form user:passwd@proxy.server:port. Note that the user:password@ is optional and required only if you are behind an authenticated proxy. If you provide user@proxy.server:port then you will be prompted for a password.

timeout Set the socket timeout (default 15 seconds)

editable install something editable (i.e. git+https://github.com/worldcompany/djangoembed.git#egg=djangoembed)

find_links URL to look for packages at

index_url Base URL of Python Package Index

extra_index_url Extra URLs of package indexes to use in addition to `index_url`

no_index Ignore package index
mirrors Specific mirror URL(s) to query (automatically adds –use-mirrors)
build Unpack packages into `build` dir
target Install packages into `target` dir
download Download packages into `download` instead of installing them
download_cache Cache downloaded packages in `download_cache` dir
source Check out `editable` packages into `source` dir
upgrade Upgrade all packages to the newest available version
force_reinstall When upgrading, reinstall all packages even if they are already up-to-date.
ignore_installed Ignore the installed packages (reinstalling instead)
exists_action Default action when a path already exists: (s)witch, (i)gnore, (w)ipe, (b)ackup
no_deps Ignore package dependencies
no_install Download and unpack all packages, but don't actually install them
no_cache_dir: Disable the cache.
cwd Current working directory to run pip from
pre_releases Include pre-releases in the available versions
cert Provide a path to an alternate CA bundle
allow_all_external Allow the installation of all externally hosted files
allow_external Allow the installation of externally hosted files (comma separated list)
allow_unverified Allow the installation of insecure and unverifiable files (comma separated list)
process_dependency_links Enable the processing of dependency links
bin_env [None] Absolute path to a virtual environment directory or absolute path to a pip executable. The example below assumes a virtual environment has been created at `/foo/.virtualenvs/bar`.
env_vars Add or modify environment variables. Useful for tweaking build steps, such as specifying INCLUDE or LIBRARY paths in Makefiles, build scripts or compiler calls. This must be in the form of a dictionary or a mapping.

Example:

```
django:
  pip.installed:
    - name: django_app
    - env_vars:
        CUSTOM_PATH: /opt/django_app
        VERBOSE: True
```

use_vt Use VT terminal emulation (see output while installing)
trusted_host Mark this host as trusted, even though it does not have valid or any HTTPS.

Example:

```
django:
  pip.installed:
    - name: django >= 1.6, <= 1.7
    - bin_env: /foo/.virtualenvs/bar
    - require:
      - pkg: python-pip
```

Or

Example:

```
django:
  pip.installed:
    - name: django >= 1.6, <= 1.7
    - bin_env: /foo/.virtualenvs/bar/bin/pip
    - require:
      - pkg: python-pip
```

Attention

The following arguments are deprecated, do not use.

---

pip_bin [None] Deprecated, use `bin_env`

Changed in version 0.17.0: `use_wheel` option added.

install_options
: Extra arguments to be supplied to the setup.py install command. If you are using an option with a directory path, be sure to use absolute path.

Example:

```
django:
  pip.installed:
    - name: django
    - install_options:
      - --prefix=/blah
    - require:
      - pkg: python-pip
```

global_options Extra global options to be supplied to the setup.py call before the install command.

New in version 2014.1.3.

Attention

As of Salt 0.17.0 the pip state **needs** an importable pip module. This usually means having the system's pip package installed or running Salt from an active virtualenv.

The reason for this requirement is because `pip` already does a pretty good job parsing its own requirements. It makes no sense for Salt to do `pip` requirements parsing and validation before passing them to the `pip` library. It's functionality duplication and it's more error prone.

Attention

Please set `reload_modules: True` to have the salt minion import this module after installation.

Example:

```
pyopenssl:
    pip.installed:
        - name: pyOpenSSL
        - reload_modules: True
        - exists_action: i
```

extra_args pip keyword and positional arguments not yet implemented in salt

```
pandas:
  pip.installed:
    - name: pandas
    - extra_args:
      - --latest-pip-kwarg: param
      - --latest-pip-arg
```

> **Warning:** If unsupported options are passed here that are not supported in a minion's version of pip, a *No such option error* will be thrown.

`salt.states.pip_state.`**`pip_has_exceptions_mod`**`(`*ver*`)`
True when the pip version has the *pip.exceptions* module

`salt.states.pip_state.`**`pip_has_internal_exceptions_mod`**`(`*ver*`)`
True when the pip version has the *pip._internal.exceptions* module

`salt.states.pip_state.`**`purge_pip`**`()`
Purge pip and it's sub-modules

`salt.states.pip_state.`**`removed`**`(`*name, requirements=None, bin_env=None, log=None, proxy=None, timeout=None, user=None, cwd=None, use_vt=False*`)`
Make sure that a package is not installed.

name The name of the package to uninstall
user The user under which to run pip
bin_env [None] the pip executable or virtualenenv to use
use_vt Use VT terminal emulation (see output while installing)

`salt.states.pip_state.`**`uptodate`**`(`*name, bin_env=None, user=None, cwd=None, use_vt=False*`)`
New in version 2015.5.0.

Verify that the system is completely up to date.
name The name has no functional value and is only used as a tracking reference
user The user under which to run pip
bin_env the pip executable or virtualenenv to use
use_vt Use VT terminal emulation (see output while installing)

21.19.209 salt.states.pkg

Installation of packages using OS package managers such as yum or apt-get

Note: On minions running systemd>=205, as of version 2015.8.12, 2016.3.3, and 2016.11.0, systemd-run(1) is now used to isolate commands which modify installed packages from the `salt-minion` daemon's control group. This is done to keep systemd from killing the package manager commands spawned by Salt, when Salt updates itself (see `KillMode` in the systemd.kill(5) manpage for more information). If desired, usage of systemd-run(1) can be suppressed by setting a *`config option`* called `systemd.use_scope`, with a value of `False` (no quotes).

Salt can manage software packages via the pkg state module, packages can be set up to be installed, latest, removed and purged. Package management declarations are typically rather simple:

```
vim:
  pkg.installed
```

A more involved example involves pulling from a custom repository.

```
base:
  pkgrepo.managed:
    - name: ppa:wolfnet/logstash
    - dist: precise
    - file: /etc/apt/sources.list.d/logstash.list
    - keyid: 28B04E4A
```

(continues on next page)

(continued from previous page)

```
    - keyserver: keyserver.ubuntu.com

logstash:
  pkg.installed:
    - fromrepo: ppa:wolfnet/logstash
```

Multiple packages can also be installed with the use of the pkgs state module

```
dotdeb.repo:
  pkgrepo.managed:
    - name: deb http://packages.dotdeb.org wheezy-php55 all
    - dist: wheezy-php55
    - file: /etc/apt/sources.list.d/dotbeb.list
    - keyid: 89DF5277
    - keyserver: keys.gnupg.net
    - refresh_db: true

php.packages:
  pkg.installed:
    - fromrepo: wheezy-php55
    - pkgs:
      - php5-fpm
      - php5-cli
      - php5-curl
```

Warning: Package names are currently case-sensitive. If the minion is using a package manager which is not case-sensitive (such as *pkgng*), then this state will fail if the proper case is not used. This will be addressed in a future release of Salt.

`salt.states.pkg.`**`downloaded`**(*name*, *version=None*, *pkgs=None*, *fromrepo=None*, *ignore_epoch=None*, ***kwargs*)

New in version 2017.7.0.

Ensure that the package is downloaded, and that it is the correct version (if specified).

Currently supported for the following pkg providers: *yumpkg*, `zypper` and *zypper*

Parameters

- **name** (*str*) – The name of the package to be downloaded. This parameter is ignored if either "pkgs" is used. Additionally, please note that this option can only be used to download packages from a software repository.
- **version** (*str*) – Download a specific version of a package.

Important: As of version 2015.8.7, for distros which use yum/dnf, packages which have a version with a nonzero epoch (that is, versions which start with a number followed by a colon must have the epoch included when specifying the version number. For example:

```
vim-enhanced:
  pkg.downloaded:
    - version: 2:7.4.160-1.el7
```

An **ignore_epoch** argument has been added to which causes the epoch to be disregarded when the state checks to see if the desired version was installed.

You can install a specific version when using the `pkgs` argument by including the version after the package:

```
common_packages:
  pkg.downloaded:
    - pkgs:
      - unzip
      - dos2unix
      - salt-minion: 2015.8.5-1.el6
```

- **resolve_capabilities** (*bool*) – Turn on resolving capabilities. This allow one to name "provides" or alias names for packages.

  New in version 2018.3.0.

CLI Example:

```
zsh:
  pkg.downloaded:
    - version: 5.0.5-4.63
    - fromrepo: "myrepository"
```

`salt.states.pkg.`**`group_installed`**(*name*, *skip=None*, *include=None*, ***kwargs*)

New in version 2015.8.0.

Changed in version 2016.11.0: Added support in `pacman`

Ensure that an entire package group is installed. This state is currently only supported for the *yum* and `pacman` package managers.

skip Packages that would normally be installed by the package group ("default" packages), which should not be installed.

```
Load Balancer:
  pkg.group_installed:
    - skip:
      - piranha
```

include Packages which are included in a group, which would not normally be installed by a `yum groupinstall` ("optional" packages). Note that this will not enforce group membership; if you include packages which are not members of the specified groups, they will still be installed.

```
Load Balancer:
  pkg.group_installed:
    - include:
      - haproxy
```

Changed in version 2016.3.0: This option can no longer be passed as a comma-separated list, it must now be passed as a list (as shown in the above example).

Note: Because this is essentially a wrapper around *pkg.install*, any argument which can be passed to pkg.install may also be included here, and it will be passed on to the call to *pkg.install*.

`salt.states.pkg.`**`installed`**(*name*, *version=None*, *refresh=None*, *fromrepo=None*, *skip_verify=False*, *skip_suggestions=False*, *pkgs=None*, *sources=None*, *allow_updates=False*, *pkg_verify=False*, *normalize=True*, *ignore_epoch=False*, *reinstall=False*, *update_holds=False*, ***kwargs*)

Ensure that the package is installed, and that it is the correct version (if specified).

Parameters

- **name** (*str*) – The name of the package to be installed. This parameter is ignored if either "pkgs" or "sources" is used. Additionally, please note that this option can only be used to install packages from a software repository. To install a package file manually, use the "sources" option detailed below.
- **version** (*str*) – Install a specific version of a package. This option is ignored if "sources" is used. Currently, this option is supported for the following pkg providers: *apt*, `ebuild`, `pacman`, *pkgin*, *win_pkg*, *yumpkg*, and `zypper`. The version number includes the release designation where applicable, to allow Salt to target a specific release of a given version. When in doubt, using the `pkg.latest_version` function for an uninstalled package will tell you the version available.

```
# salt myminion pkg.latest_version vim-enhanced
myminion:
    2:7.4.160-1.el7
```

Important: As of version 2015.8.7, for distros which use yum/dnf, packages which have a version with a nonzero epoch (that is, versions which start with a number followed by a colon like in the `pkg.latest_version` output above) must have the epoch included when specifying the version number. For example:

```
vim-enhanced:
  pkg.installed:
    - version: 2:7.4.160-1.el7
```

In version 2015.8.9, an **ignore_epoch** argument has been added to *pkg.installed*, *pkg.removed*, and *pkg.purged* states, which causes the epoch to be disregarded when the state checks to see if the desired version was installed.

Also, while this function is not yet implemented for all pkg frontends, *pkg.list_repo_pkgs* will show all versions available in the various repositories for a given package, irrespective of whether or not it is installed.

```
# salt myminion pkg.list_repo_pkgs bash
myminion:
----------
    bash:
        - 4.2.46-21.el7_3
        - 4.2.46-20.el7_2
```

This function was first added for *pkg.list_repo_pkgs* in 2014.1.0, and was expanded to *Debian/Ubuntu* and `Arch Linux`-based distros in the 2017.7.0 release.

The version strings returned by either of these functions can be used as version specifiers in pkg states.

You can install a specific version when using the `pkgs` argument by including the version after the package:

```
common_packages:
  pkg.installed:
    - pkgs:
      - unzip
      - dos2unix
      - salt-minion: 2015.8.5-1.el6
```

If the version given is the string `latest`, the latest available package version will be installed à la `pkg.latest`.

WILDCARD VERSIONS

As of the 2017.7.0 release, this state now supports wildcards in package versions for SUSE SLES/Leap/Tumbleweed, Debian/Ubuntu, RHEL/CentOS, Arch Linux, and their derivatives. Using wildcards can be useful for packages where the release name is built into the version in some way, such as for RHEL/CentOS which typically has version numbers like `1.2.34-5.el7`. An example of the usage for this would be:

```
mypkg:
  pkg.installed:
    - version: '1.2.34*'
```

Keep in mind that using wildcard versions will result in a slower state run since Salt must gather the available versions of the specified packages and figure out which of them match the specified wildcard expression.

- **`refresh`** (*`bool`*) – This parameter controls whether or not the package repo database is updated prior to installing the requested package(s).

  If `True`, the package database will be refreshed (`apt-get update` or equivalent, depending on platform) before installing.

  If `False`, the package database will *not* be refreshed before installing.

  If unset, then Salt treats package database refreshes differently depending on whether or not a `pkg` state has been executed already during the current Salt run. Once a refresh has been performed in a `pkg` state, for the remainder of that Salt run no other refreshes will be performed for `pkg` states which do not explicitly set `refresh` to `True`. This prevents needless additional refreshes from slowing down the Salt run.
- **`cache_valid_time`** (*`str`*) – New in version 2016.11.0.

  This parameter sets the value in seconds after which the cache is marked as invalid, and a cache update is necessary. This overwrites the `refresh` parameter's default behavior.

  Example:

```
httpd:
  pkg.installed:
    - fromrepo: mycustomrepo
    - skip_verify: True
    - skip_suggestions: True
    - version: 2.0.6~ubuntu3
    - refresh: True
    - cache_valid_time: 300
    - allow_updates: True
    - hold: False
```

  In this case, a refresh will not take place for 5 minutes since the last `apt-get update` was executed on the system.

  Note: This parameter is available only on Debian based distributions and has no effect on the rest.

- **`fromrepo`** (*`str`*) – Specify a repository from which to install

---

Note: Distros which use APT (Debian, Ubuntu, etc.) do not have a concept of repositories, in the same way as YUM-based distros do. When a source is added, it is assigned to a given release. Consider the following source configuration:

```
deb http://ppa.launchpad.net/saltstack/salt/ubuntu precise main
```

The packages provided by this source would be made available via the `precise` release, therefore `fromrepo` would need to be set to `precise` for Salt to install the package from this source.

Having multiple sources in the same release may result in the default install candidate being newer than what is desired. If this is the case, the desired version must be specified using the `version` parameter.

If the `pkgs` parameter is being used to install multiple packages in the same state, then instead of using `version`, use the method of version specification described in the **Multiple Package Installation Options** section below.

Running the shell command `apt-cache policy pkgname` on a minion can help elucidate the APT configuration and aid in properly configuring states:

```
root@saltmaster:~# salt ubuntu01 cmd.run 'apt-cache policy
↪ffmpeg'
ubuntu01:
    ffmpeg:
    Installed: (none)
    Candidate: 7:0.10.11-1~precise1
    Version table:
        7:0.10.11-1~precise1 0
            500 http://ppa.launchpad.net/jon-severinsson/
↪ffmpeg/ubuntu/ precise/main amd64 Packages
        4:0.8.10-0ubuntu0.12.04.1 0
            500 http://us.archive.ubuntu.com/ubuntu/ precise-
↪updates/main amd64 Packages
            500 http://security.ubuntu.com/ubuntu/ precise-
↪security/main amd64 Packages
        4:0.8.1-0ubuntu1 0
            500 http://us.archive.ubuntu.com/ubuntu/ precise/
↪main amd64 Packages
```

The release is located directly after the source's URL. The actual release name is the part before the slash, so to install version **4:0.8.10-0ubuntu0.12.04.1** either `precise-updates` or `precise-security` could be used for the `fromrepo` value.

- **`skip_verify`** (*bool*) – Skip the GPG verification check for the package to be installed
- **`skip_suggestions`** (*bool*) – Force strict package naming. Disables lookup of package alternatives.

  New in version 2014.1.1.
- **`resolve_capabilities`** (*bool*) – Turn on resolving capabilities. This allow one to name "provides" or alias names for packages.

  New in version 2018.3.0.
- **`allow_updates`** (*bool*) – Allow the package to be updated outside Salt's control (e.g. auto updates on Windows). This means a package on the Minion can have

a newer version than the latest available in the repository without enforcing a re-installation of the package.

New in version 2014.7.0.

Example:

```
httpd:
  pkg.installed:
    - fromrepo: mycustomrepo
    - skip_verify: True
    - skip_suggestions: True
    - version: 2.0.6~ubuntu3
    - refresh: True
    - allow_updates: True
    - hold: False
```

- **pkg_verify** (*bool*) – New in version 2014.7.0.

  For requested packages that are already installed and would not be targeted for upgrade or downgrade, use pkg.verify to determine if any of the files installed by the package have been altered. If files have been altered, the reinstall option of pkg.install is used to force a reinstall. Types to ignore can be passed to pkg.verify. Additionally, `verify_options` can be used to modify further the behavior of pkg.verify. See examples below. Currently, this option is supported for the following pkg providers: *yumpkg*.

  Examples:

```
httpd:
  pkg.installed:
    - version: 2.2.15-30.el6.centos
    - pkg_verify: True
```

```
mypkgs:
  pkg.installed:
    - pkgs:
      - foo
      - bar: 1.2.3-4
      - baz
    - pkg_verify:
      - ignore_types:
        - config
        - doc
```

```
mypkgs:
  pkg.installed:
    - pkgs:
      - foo
      - bar: 1.2.3-4
      - baz
    - pkg_verify:
      - ignore_types:
        - config
        - doc
      - verify_options:
        - nodeps
        - nofiledigest
```

- **ignore_types** (*list*) – List of types to ignore when verifying the package

New in version 2014.7.0.

- **verify_options** (*list*) – List of additional options to pass when verifying the package. These options will be added to the `rpm -V` command, prepended with `--` (for example, when `nodeps` is passed in this option, `rpm -V` will be run with `--nodeps`).

  New in version 2016.11.0.

- **normalize** (*bool*) – Normalize the package name by removing the architecture, if the architecture of the package is different from the architecture of the operating system. The ability to disable this behavior is useful for poorly-created packages which include the architecture as an actual part of the name, such as kernel modules which match a specific kernel version.

  New in version 2014.7.0.

  Example:

```
gpfs.gplbin-2.6.32-279.31.1.el6.x86_64:
  pkg.installed:
    - normalize: False
```

- **ignore_epoch** (*bool*) – When a package version contains an non-zero epoch (e.g. `1:3.14.159-2.el7`, and a specific version of a package is desired, set this option to `True` to ignore the epoch when comparing versions. This allows for the following SLS to be used:

```
# Actual vim-enhanced version: 2:7.4.160-1.el7
vim-enhanced:
  pkg.installed:
    - version: 7.4.160-1.el7
    - ignore_epoch: True
```

  Without this option set to `True` in the above example, the package would be installed, but the state would report as failed because the actual installed version would be `2:7.4.160-1.el7`. Alternatively, this option can be left as `False` and the full version string (with epoch) can be specified in the SLS file:

```
vim-enhanced:
  pkg.installed:
    - version: 2:7.4.160-1.el7
```

  New in version 2015.8.9.

MULTIPLE PACKAGE INSTALLATION OPTIONS: **(not supported in pkgng)**

Parameters

- **pkgs** (*list*) – A list of packages to install from a software repository. All packages listed under `pkgs` will be installed via a single command.

```
mypkgs:
  pkg.installed:
    - pkgs:
      - foo
      - bar
```

(continues on next page)

(continued from previous page)

```
      - baz
    - hold: True
```

NOTE: For *apt*, `ebuild`, `pacman`, *winrepo*, *yumpkg*, and `zypper`, version numbers can be specified in the `pkgs` argument. For example:

```
mypkgs:
  pkg.installed:
    - pkgs:
      - foo
      - bar: 1.2.3-4
      - baz
```

Additionally, `ebuild`, `pacman`, `zypper`, *yum/dnf*, and *apt* support the <, <=, >=, and > operators for more control over what versions will be installed. For example:

```
mypkgs:
  pkg.installed:
    - pkgs:
      - foo
      - bar: '>=1.2.3-4'
      - baz
```

NOTE: When using comparison operators, the expression must be enclosed in quotes to avoid a YAML render error.

With `ebuild` is also possible to specify a use flag list and/or if the given packages should be in package.accept_keywords file and/or the overlay from which you want the package to be installed. For example:

```
mypkgs:
  pkg.installed:
    - pkgs:
      - foo: '~'
      - bar: '~>=1.2:slot::overlay[use,-otheruse]'
      - baz
```

- **sources** (*list*) – A list of packages to install, along with the source URI or local path from which to install each package. In the example below, `foo`, `bar`, `baz`, etc. refer to the name of the package, as it would appear in the output of the `pkg.version` or `pkg.list_pkgs` salt CLI commands.

```
mypkgs:
  pkg.installed:
    - sources:
      - foo: salt://rpms/foo.rpm
      - bar: http://somesite.org/bar.rpm
      - baz: ftp://someothersite.org/baz.rpm
      - qux: /minion/path/to/qux.rpm
```

PLATFORM-SPECIFIC ARGUMENTS

These are specific to each OS. If it does not apply to the execution module for your OS, it is ignored.

Parameters

- **hold** (*bool*) – Force the package to be held at the current installed version. Currently works with YUM/DNF & APT based systems.

New in version 2014.7.0.

- **`update_holds`** (*bool*) – If `True`, and this function would update the package version, any packages which are being held will be temporarily unheld so that they can be updated. Otherwise, if this function attempts to update a held package, the held package(s) will be skipped and the state will fail. By default, this parameter is set to `False`.

  Currently works with YUM/DNF & APT based systems.

  New in version 2016.11.0.

- **`names`** (*list*) – A list of packages to install from a software repository. Each package will be installed individually by the package manager.

  Warning: Unlike `pkgs`, the `names` parameter cannot specify a version. In addition, it makes a separate call to the package management frontend to install each package, whereas `pkgs` makes just a single call. It is therefore recommended to use `pkgs` instead of `names` to install multiple packages, both for the additional features and the performance improvement that it brings.

- **`install_recommends`** (*bool*) – Whether to install the packages marked as recommended. Default is `True`. Currently only works with APT-based systems.

  New in version 2015.5.0.

```
httpd:
  pkg.installed:
    - install_recommends: False
```

- **`only_upgrade`** (*bool*) – Only upgrade the packages, if they are already installed. Default is `False`. Currently only works with APT-based systems.

  New in version 2015.5.0.

```
httpd:
  pkg.installed:
    - only_upgrade: True
```

Note: If this parameter is set to True and the package is not already installed, the state will fail.

- **`report_reboot_exit_codes`** (*bool*) – If the installer exits with a recognized exit code indicating that a reboot is required, the module function

  win_system.set_reboot_required_witnessed

  will be called, preserving the knowledge of this event for the remainder of the current boot session. For the time being, `3010` is the only recognized exit code, but this is subject to future refinement. The value of this param defaults to `True`. This parameter has no effect on non-Windows systems.

  New in version 2016.11.0.

```
ms vcpp installed:
  pkg.installed:
    - name: ms-vcpp
    - version: 10.0.40219
    - report_reboot_exit_codes: False
```

Returns A dictionary containing the state of the software installation

Rtype dict

Note: The `pkg.installed` state supports the usage of `reload_modules`. This functionality allows you to force Salt to reload all modules. In many cases, Salt is clever enough to transparently reload the modules. For example, if you install a package, Salt reloads modules because some other module or state might require the package which was installed. However, there are some edge cases where this may not be the case, which is what `reload_modules` is meant to resolve.

You should only use `reload_modules` if your `pkg.installed` does some sort of installation where if you do not reload the modules future items in your state which rely on the software being installed will fail. Please see the *Reloading Modules* documentation for more information.

See also:

unless and onlyif

You can use the *unless* or *onlyif* syntax to skip a full package run. This can be helpful in large environments with multiple states that include requisites for packages to be installed.

```
# Using file.file_exists for a single-factor check
install_nginx:
  pkg.installed:
    - name: nginx
    - unless:
      - fun: file.file_exists
        args:
          - /etc/nginx/nginx.conf
```

```
# Using file.search for a two-factor check
install_nginx:
  pkg.installed:
    - name: nginx
    - unless:
      - fun: file.search
        args:
          - /etc/nginx/nginx.conf
          - 'user www-data;'
```

The above examples use two different methods to reasonably ensure that a package has already been installed. First, with checking for a file that would be created with the package. Second, by checking for specific text within a file that would be created or managed by salt. With these requisists satisfied, unless will return `True` and the `pkg.installed` state will be skipped.

```
# Example of state run without unless used
salt 'saltdev' state.apply nginx
saltdev:
----------
          ID: install_nginx
          Function: pkg.installed
          Name: nginx
          Result: True
          Comment: All specified packages are already installed
          Started: 20:11:56.388331
          Duration: 4290.0 ms
          Changes:

# Example of state run using unless requisite
```

(continues on next page)

(continued from previous page)

```
salt 'saltdev' state.apply nginx
saltdev:
----------
          ID: install_nginx
    Function: pkg.installed
        Name: nginx
      Result: True
     Comment: unless condition is true
     Started: 20:10:50.659215
    Duration: 1530.0 ms
     Changes:
```

The result is a reduction of almost 3 seconds. In larger environments, small reductions in waiting time can add up.

Unless Requisite

`salt.states.pkg.`**`latest`**(*name*, *refresh=None*, *fromrepo=None*, *skip_verify=False*, *pkgs=None*, *watch_flags=True*, ***kwargs*)

Ensure that the named package is installed and the latest available package. If the package can be updated, this state function will update the package. Generally it is better for the *`installed`* function to be used, as *`latest`* will update the package whenever a new package is available.

name The name of the package to maintain at the latest available version. This parameter is ignored if "pkgs" is used.

fromrepo Specify a repository from which to install

skip_verify Skip the GPG verification check for the package to be installed

refresh This parameter controls whether or not the package repo database is updated prior to checking for the latest available version of the requested packages.

If `True`, the package database will be refreshed (`apt-get update` or equivalent, depending on platform) before checking for the latest available version of the requested packages.

If `False`, the package database will *not* be refreshed before checking.

If unset, then Salt treats package database refreshes differently depending on whether or not a `pkg` state has been executed already during the current Salt run. Once a refresh has been performed in a `pkg` state, for the remainder of that Salt run no other refreshes will be performed for `pkg` states which do not explicitly set `refresh` to `True`. This prevents needless additional refreshes from slowing down the Salt run.

param str cache_valid_time New in version 2016.11.0.

This parameter sets the value in seconds after which the cache is marked as invalid, and a cache update is necessary. This overwrites the `refresh` parameter's default behavior.

Example:

```
httpd:
  pkg.latest:
    - refresh: True
    - cache_valid_time: 300
```

In this case, a refresh will not take place for 5 minutes since the last `apt-get update` was executed on the system.

Note: This parameter is available only on Debian based distributions and has no effect on the rest.

param bool resolve_capabilities Turn on resolving capabilities. This allow one to name "provides" or alias names for packages.

New in version 2018.3.0.

Multiple Package Installation Options:

(Not yet supported for: FreeBSD, OpenBSD, MacOS, and Solaris pkgutil)
pkgs A list of packages to maintain at the latest available version.

```
mypkgs:
  pkg.latest:
    - pkgs:
      - foo
      - bar
      - baz
```

install_recommends Whether to install the packages marked as recommended. Default is `True`. Currently only works with APT-based systems.

New in version 2015.5.0.

```
httpd:
  pkg.latest:
    - install_recommends: False
```

only_upgrade Only upgrade the packages, if they are already installed. Default is `False`. Currently only works with APT-based systems.

New in version 2015.5.0.

```
httpd:
  pkg.latest:
    - only_upgrade: True
```

Note: If this parameter is set to True and the package is not already installed, the state will fail.

report_reboot_exit_codes If the installer exits with a recognized exit code indicating that a reboot is required, the module function

win_system.set_reboot_required_witnessed

will be called, preserving the knowledge of this event for the remainder of the current boot session. For the time being, `3010` is the only recognized exit code, but this is subject to future refinement. The value of this param defaults to `True`. This parameter has no effect on non-Windows systems.

New in version 2016.11.0.

```
ms vcpp installed:
  pkg.latest:
    - name: ms-vcpp
    - report_reboot_exit_codes: False
```

`salt.states.pkg.`**`mod_aggregate`**`(`*low, chunks, running*`)`

The mod_aggregate function which looks up all packages in the available low chunks and merges them into a single pkgs ref in the present low data `salt.states.pkg.`**`mod_watch`**`(`*name*, ***kwargs*`)`

Install/reinstall a package based on a watch requisite

Note: This state exists to support special handling of the `watch` *requisite*. It should not be called directly.

Parameters for this function should be set by the state being triggered.

`salt.states.pkg.`**`patch_downloaded`**`(`*name*, *advisory_ids=None*, ***kwargs*`)`

New in version 2017.7.0.

Ensure that packages related to certain advisory ids are downloaded.

Currently supported for the following pkg providers: *`yumpkg`* and `zypper`

CLI Example:

```
preparing-to-fix-issues:
  pkg.patch_downloaded:
    - advisory_ids:
      - SUSE-SLE-SERVER-12-SP2-2017-185
      - SUSE-SLE-SERVER-12-SP2-2017-150
      - SUSE-SLE-SERVER-12-SP2-2017-120
```

`salt.states.pkg.`**`patch_installed`**`(`*name*, *advisory_ids=None*, *downloadonly=None*, ***kwargs*`)`

New in version 2017.7.0.

Ensure that packages related to certain advisory ids are installed.

Currently supported for the following pkg providers: *`yumpkg`* and `zypper`

CLI Example:

```
issue-foo-fixed:
  pkg.patch_installed:
    - advisory_ids:
      - SUSE-SLE-SERVER-12-SP2-2017-185
      - SUSE-SLE-SERVER-12-SP2-2017-150
      - SUSE-SLE-SERVER-12-SP2-2017-120
```

`salt.states.pkg.`**`purged`**`(`*name*, *version=None*, *pkgs=None*, *normalize=True*, *ignore_epoch=False*, ***kwargs*`)`

Verify that a package is not installed, calling `pkg.purge` if necessary to purge the package. All configuration files are also removed.

name The name of the package to be purged.

version The version of the package that should be removed. Don't do anything if the package is installed with an unmatching version.

Important: As of version 2015.8.7, for distros which use yum/dnf, packages which have a version with a nonzero epoch (that is, versions which start with a number followed by a colon like in the example above) must have the epoch included when specifying the version number. For example:

```
vim-enhanced:
  pkg.purged:
    - version: 2:7.4.160-1.el7
```

In version 2015.8.9, an **ignore_epoch** argument has been added to *`pkg.installed`*, *`pkg.removed`*, and *`pkg.purged`* states, which causes the epoch to be disregarded when the state checks to see if the desired version was installed. If **ignore_epoch** was not set to `True`, and instead of `2:7.4.160-1.el7` a version of `7.4.160-1.el7` were used, this state would report success since the actual installed version includes the epoch, and the specified version would not match.

---

normalize [True] Normalize the package name by removing the architecture, if the architecture of the package is different from the architecture of the operating system. The ability to disable this behavior is useful for poorly-created packages which include the architecture as an actual part of the name, such as kernel modules which match a specific kernel version.

New in version 2015.8.0.

ignore_epoch [False] When a package version contains an non-zero epoch (e.g. `1:3.14.159-2.el7`, and a specific version of a package is desired, set this option to `True` to ignore the epoch when comparing versions. This allows for the following SLS to be used:

```
# Actual vim-enhanced version: 2:7.4.160-1.el7
vim-enhanced:
  pkg.purged:
    - version: 7.4.160-1.el7
    - ignore_epoch: True
```

Without this option set to `True` in the above example, the state would falsely report success since the actual installed version is `2:7.4.160-1.el7`. Alternatively, this option can be left as `False` and the full version string (with epoch) can be specified in the SLS file:

```
vim-enhanced:
  pkg.purged:
    - version: 2:7.4.160-1.el7
```

New in version 2015.8.9.

Multiple Package Options:

pkgs A list of packages to purge. Must be passed as a python list. The `name` parameter will be ignored if this option is passed. It accepts version numbers as well.

New in version 0.16.0.

`salt.states.pkg.`**`removed`**(*name*, *version=None*, *pkgs=None*, *normalize=True*, *ignore_epoch=False*, ***kwargs*)

Verify that a package is not installed, calling `pkg.remove` if necessary to remove the package.

name The name of the package to be removed.

version The version of the package that should be removed. Don't do anything if the package is installed with an unmatching version.

Important: As of version 2015.8.7, for distros which use yum/dnf, packages which have a version with a nonzero epoch (that is, versions which start with a number followed by a colon like in the example above) must have the epoch included when specifying the version number. For example:

```
vim-enhanced:
  pkg.removed:
    - version: 2:7.4.160-1.el7
```

In version 2015.8.9, an **ignore_epoch** argument has been added to *`pkg.installed`*, *`pkg.removed`*, and *`pkg.purged`* states, which causes the epoch to be disregarded when the state checks to see if the desired version was installed. If **ignore_epoch** was not set to `True`, and instead of `2:7.4.160-1.el7` a version of `7.4.160-1.el7` were used, this state would report success since the actual installed version includes the epoch, and the specified version would not match.

---

normalize [True] Normalize the package name by removing the architecture, if the architecture of the package is different from the architecture of the operating system. The ability to disable this behavior is useful for poorly-created packages which include the architecture as an actual part of the name, such as kernel modules which match a specific kernel version.

New in version 2015.8.0.

ignore_epoch [False] When a package version contains an non-zero epoch (e.g. `1:3.14.159-2.el7`, and a specific version of a package is desired, set this option to `True` to ignore the epoch when comparing versions. This allows for the following SLS to be used:

```
# Actual vim-enhanced version: 2:7.4.160-1.el7
vim-enhanced:
  pkg.removed:
    - version: 7.4.160-1.el7
    - ignore_epoch: True
```

Without this option set to `True` in the above example, the state would falsely report success since the actual installed version is `2:7.4.160-1.el7`. Alternatively, this option can be left as `False` and the full version string (with epoch) can be specified in the SLS file:

```
vim-enhanced:
  pkg.removed:
    - version: 2:7.4.160-1.el7
```

New in version 2015.8.9.

Multiple Package Options:

pkgs A list of packages to remove. Must be passed as a python list. The `name` parameter will be ignored if this option is passed. It accepts version numbers as well.

New in version 0.16.0.

`salt.states.pkg.`**`uptodate`**(*name*, *refresh=False*, *pkgs=None*, ***kwargs*)

New in version 2014.7.0.

Changed in version 2018.3.0: Added support for the `pkgin` provider.

Verify that the system is completely up to date.

name The name has no functional value and is only used as a tracking reference

refresh refresh the package database before checking for new upgrades

pkgs list of packages to upgrade

Parameters

- **`cache_valid_time`** (*`str`*) – This parameter sets the value in seconds after which cache marked as invalid, and cache update is necessary. This overwrite `refresh` parameter default behavior.

  In this case cache_valid_time is set, refresh will not take place for amount in seconds since last `apt-get update` executed on the system.

  Note: This parameter available only on Debian based distributions, and have no effect on the rest.

- **`resolve_capabilities`** (*`bool`*) – Turn on resolving capabilities. This allow one to name "provides" or alias names for packages.

  New in version 2018.3.0.

kwargs Any keyword arguments to pass through to `pkg.upgrade`.

New in version 2015.5.0.

21.19.210 salt.states.pkgbuild

The pkgbuild state is the front of Salt package building backend. It automatically builds DEB and RPM packages from specified sources

New in version 2015.8.0.

```
salt_2015.5.2:
  pkgbuild.built:
    - runas: thatch
    - results:
      - salt-2015.5.2-2.el7.centos.noarch.rpm
      - salt-api-2015.5.2-2.el7.centos.noarch.rpm
      - salt-cloud-2015.5.2-2.el7.centos.noarch.rpm
      - salt-master-2015.5.2-2.el7.centos.noarch.rpm
      - salt-minion-2015.5.2-2.el7.centos.noarch.rpm
      - salt-ssh-2015.5.2-2.el7.centos.noarch.rpm
      - salt-syndic-2015.5.2-2.el7.centos.noarch.rpm
    - dest_dir: /tmp/pkg
    - spec: salt://pkg/salt/spec/salt.spec
    - template: jinja
    - deps:
      - salt://pkg/salt/sources/required_dependency.rpm
    - tgt: epel-7-x86_64
    - sources:
      - salt://pkg/salt/sources/logrotate.salt
      - salt://pkg/salt/sources/README.fedora
      - salt://pkg/salt/sources/salt-2015.5.2.tar.gz
      - salt://pkg/salt/sources/salt-2015.5.2-tests.patch
      - salt://pkg/salt/sources/salt-api
      - salt://pkg/salt/sources/salt-api.service
      - salt://pkg/salt/sources/salt-master
      - salt://pkg/salt/sources/salt-master.service
      - salt://pkg/salt/sources/salt-minion
      - salt://pkg/salt/sources/salt-minion.service
      - salt://pkg/salt/sources/saltpkg.sls
      - salt://pkg/salt/sources/salt-syndic
      - salt://pkg/salt/sources/salt-syndic.service
      - salt://pkg/salt/sources/SaltTesting-2015.5.8.tar.gz
/tmp/pkg:
  pkgbuild.repo
```

`salt.states.pkgbuild.`**`built`**(*name, runas, dest_dir, spec, sources, tgt, template=None, deps=None, env=None, results=None, force=False, saltenv='base', log_dir='/var/log/salt/pkgbuild'*)

Ensure that the named package is built and exists in the named directory

name The name to track the build, the name value is otherwise unused

runas The user to run the build process as

dest_dir The directory on the minion to place the built package(s)

spec The location of the spec file (used for rpms)

sources The list of package sources

tgt The target platform to run the build on

template Run the spec file through a templating engine

> Changed in version 2015.8.2: This argument is now optional, allowing for no templating engine to be used if none is desired.

deps Packages required to ensure that the named package is built can be hosted on either the salt master server or on an HTTP or FTP server. Both HTTPS and HTTP are supported as well as downloading directly from Amazon S3 compatible URLs with both pre-configured and automatic IAM credentials

env A dictionary of environment variables to be set prior to execution. Example:

```
- env:
    DEB_BUILD_OPTIONS: 'nocheck'
```

> **Warning:** The above illustrates a common PyYAML pitfall, that **yes**, **no**, **on**, **off**, **true**, and **false** are all loaded as boolean `True` and `False` values, and must be enclosed in quotes to be used as strings. More info on this (and other) PyYAML idiosyncrasies can be found *here*.

results The names of the expected rpms that will be built

force [False] If `True`, packages will be built even if they already exist in the `dest_dir`. This is useful when building a package for continuous or nightly package builds.

New in version 2015.8.2.

saltenv The saltenv to use for files downloaded from the salt filesever

log_dir [/var/log/salt/rpmbuild] Root directory for log files created from the build. Logs will be organized by package name, version, OS release, and CPU architecture under this directory.

New in version 2015.8.2.

`salt.states.pkgbuild.`**`repo`**(*name, keyid=None, env=None, use_passphrase=False, gnupghome='/etc/salt/gpgkeys', runas='builder', timeout=15.0*)

Make a package repository and optionally sign it and packages present

The name is directory to turn into a repo. This state is best used with onchanges linked to your package building states.

name The directory to find packages that will be in the repository

keyid Changed in version 2016.3.0.

Optional Key ID to use in signing packages and repository. Utilizes Public and Private keys associated with keyid which have been loaded into the minion's Pillar data.

For example, contents from a Pillar data file with named Public and Private keys as follows:

```
gpg_pkg_priv_key: |
  -----BEGIN PGP PRIVATE KEY BLOCK-----
  Version: GnuPG v1

  lQO+BFciIfQBCADAPCtzx7I5Rl32escCMZsPzaEKWe7bIX1em4KCKkBoX47IG54b
  w82PCE8Y1jF/9Uk2m3RKVWp3YcLlc7Ap3gj6VO4ysvVz28UbnhPxsIkOlf2cq8qc
  .
  .
  Ebe+8JCQTwqSXPRTzXmy/b5WXDeM79CkLWvuGpXFor76D+ECMRPv/rawukEcNptn
  R5OmgHqvydEnO4pWbn8JzQO9YX/Us0SMHBVzLC8eIi5ZIopzalvX
  =JvW8
  -----END PGP PRIVATE KEY BLOCK-----

gpg_pkg_priv_keyname: gpg_pkg_key.pem

gpg_pkg_pub_key: |
  -----BEGIN PGP PUBLIC KEY BLOCK-----
  Version: GnuPG v1

  mQENBFciIfQBCADAPCtzx7I5Rl32escCMZsPzaEKWe7bIX1em4KCKkBoX47IG54b
  w82PCE8Y1jF/9Uk2m3RKVWp3YcLlc7Ap3gj6VO4ysvVz28UbnhPxsIkOlf2cq8qc
  .
```

(continues on next page)

(continued from previous page)

```
    .
    bYP7t5iwJmQzRMyFInYRt77wkJBPCpJc9FPNebL9vlZcN4zv0KQta+4alcWivvoP
    4QIxE+/+trC6QRw2m2dHk6aAeq/J0Sc7ilZufwnNA71hf9SzRIwcFXMsLx4iLlki
    inNqW9c=
    =s1CX
    -----END PGP PUBLIC KEY BLOCK-----

gpg_pkg_pub_keyname: gpg_pkg_key.pub
```

env Changed in version 2016.3.0.

A dictionary of environment variables to be utilized in creating the repository. Example:

```
- env:
    OPTIONS: 'ask-passphrase'
```

Warning: The above illustrates a common `PyYAML` pitfall, that **yes**, **no**, **on**, **off**, **true**, and **false** are all loaded as boolean `True` and `False` values, and must be enclosed in quotes to be used as strings. More info on this (and other) `PyYAML` idiosyncrasies can be found *here*.

Use of `OPTIONS` on some platforms, for example: `ask-passphrase`, will require `gpg-agent` or similar to cache passphrases.

Note: This parameter is not used for making `yum` repositories.

use_passphrase [False] New in version 2016.3.0.

Use a passphrase with the signing key presented in `keyid`. Passphrase is received from Pillar data which could be passed on the command line with `pillar` parameter. For example:

```
pillar='{ "gpg_passphrase" : "my_passphrase" }'
```

gnupghome [/etc/salt/gpgkeys] New in version 2016.3.0.

Location where GPG related files are stored, used with 'keyid'

runas [builder] New in version 2016.3.0.

User to create the repository as, and optionally sign packages.

Note: Ensure the user has correct permissions to any files and directories which are to be utilized.

timeout [15.0] New in version 2016.3.4.

Timeout in seconds to wait for the prompt for inputting the passphrase.

21.19.211 salt.states.pkgng

Manage package remote repo using FreeBSD pkgng

Salt can manage the URL pkgng pulls packages from. ATM the state and module are small so use cases are typically rather simple:

```
pkgng_clients:
  pkgng.update_packaging_site:
    - name: "http://192.168.0.2"
```

salt.states.pkgng.**update_packaging_site**(*name*)

21.19.212 salt.states.pkgrepo

Management of APT/DNF/YUM/Zypper package repos

States for managing software package repositories on Linux distros. Supported package managers are APT, DNF, YUM and Zypper. Here is some example SLS:

```
base:
  pkgrepo.managed:
    - humanname: CentOS-$releasever - Base
    - mirrorlist: http://mirrorlist.centos.org/?release=$releasever&arch=$basearch&
↪repo=os
    - comments:
        - 'http://mirror.centos.org/centos/$releasever/os/$basearch/'
    - gpgcheck: 1
    - gpgkey: file:///etc/pki/rpm-gpg/RPM-GPG-KEY-CentOS-6
```

```
base:
  pkgrepo.managed:
    - name: deb http://ppa.launchpad.net/wolfnet/logstash/ubuntu precise main
    - dist: precise
    - file: /etc/apt/sources.list.d/logstash.list
    - keyid: 28B04E4A
    - keyserver: keyserver.ubuntu.com
    - require_in:
      - pkg: logstash

  pkg.latest:
    - name: logstash
    - refresh: True
```

```
base:
  pkgrepo.managed:
    - name: deb http://www.deb-multimedia.org stable main
    - file: /etc/apt/sources.list.d/deb-multimedia.list
    - key_url: salt://deb-multimedia/files/marillat.pub
```

```
base:
  pkgrepo.managed:
    - name: deb http://dl.google.com/linux/chrome/deb/ stable main
    - dist: stable
    - file: /etc/apt/sources.list.d/chrome-browser.list
    - require_in:
      - pkg: google-chrome-stable
    - gpgcheck: 1
    - key_url: https://dl-ssl.google.com/linux/linux_signing_key.pub
```

```
base:
  pkgrepo.managed:
    - ppa: wolfnet/logstash
  pkg.latest:
    - name: logstash
    - refresh: True
```

Note: On Ubuntu systems, the `python-software-properties` package should be installed for better support of PPA repositories. To check if this package is installed, run `dpkg -l python-software-properties`.

Also, some Ubuntu releases have a bug in their `python-software-properties` package, a missing dependency on pycurl, so `python-pycurl` will need to be manually installed if it is not present once `python-software-properties` is installed.

On Ubuntu & Debian systems, the `python-apt` package is required to be installed. To check if this package is installed, run `dpkg -l python-apt`. `python-apt` will need to be manually installed if it is not present.

`salt.states.pkgrepo.`**`absent`**`(`*name*, ***kwargs*`)`

This function deletes the specified repo on the system, if it exists. It is essentially a wrapper around pkg.del_repo.

name The name of the package repo, as it would be referred to when running the regular package manager commands.

UBUNTU-SPECIFIC OPTIONS

ppa On Ubuntu, you can take advantage of Personal Package Archives on Launchpad simply by specifying the user and archive name.

```
logstash-ppa:
  pkgrepo.absent:
    - ppa: wolfnet/logstash
```

ppa_auth For Ubuntu PPAs there can be private PPAs that require authentication to access. For these PPAs the username/password can be specified. This is required for matching if the name format uses the `ppa:` specifier and is private (requires username/password to access, which is encoded in the URI).

```
logstash-ppa:
  pkgrepo.absent:
    - ppa: wolfnet/logstash
    - ppa_auth: username:password
```

keyid If passed, then the GPG key corresponding to the passed KeyID will also be removed.

keyid_ppa [False] If set to `True`, the GPG key's ID will be looked up from ppa.launchpad.net and removed, and the `keyid` argument will be ignored.

Note: This option will be disregarded unless the `ppa` argument is present.

`salt.states.pkgrepo.`**`managed`**`(`*name*, *ppa=None*, ***kwargs*`)`

This state manages software package repositories. Currently, *`yum`*, *`apt`*, and `zypper` repositories are supported.

YUM/DNF/ZYPPER-BASED SYSTEMS

Note: One of `baseurl` or `mirrorlist` below is required. Additionally, note that this state is not presently capable of managing more than one repo in a single repo file, so each instance of this state will manage a single repo file containing the configuration for a single repo.

---

name This value will be used in two ways: Firstly, it will be the repo ID, as seen in the entry in square brackets (e.g. `[foo]`) for a given repo. Secondly, it will be the name of the file as stored in /etc/yum.repos.d (e.g. `/etc/yum.repos.d/foo.conf`).

enabled [True] Whether or not the repo is enabled. Can be specified as True/False or 1/0.

disabled [False] Included to reduce confusion due to APT's use of the `disabled` argument. If this is passed for a YUM/DNF/Zypper-based distro, then the reverse will be passed as `enabled`. For example passing `disabled=True` will assume `enabled=False`.

humanname This is used as the "name" value in the repo file in `/etc/yum.repos.d/` (or `/etc/zypp/repos.d` for SUSE distros).

baseurl The URL to a yum repository

mirrorlist A URL which points to a file containing a collection of baseurls

comments Sometimes you want to supply additional information, but not as enabled configuration. Anything supplied for this list will be saved in the repo configuration with a comment marker (#) in front.

gpgautoimport Only valid for Zypper package manager. If set to True, automatically trust and import public GPG key for the repository. The key should be specified with `gpgkey` parameter. See details below.

Additional configuration values seen in YUM/DNF/Zypper repo files, such as `gpgkey` or `gpgcheck`, will be used directly as key-value pairs. For example:

```
foo:
  pkgrepo.managed:
    - humanname: Personal repo for foo
    - baseurl: https://mydomain.tld/repo/foo/$releasever/$basearch
    - gpgkey: file:///etc/pki/rpm-gpg/foo-signing-key
    - gpgcheck: 1
```

APT-BASED SYSTEMS

ppa On Ubuntu, you can take advantage of Personal Package Archives on Launchpad simply by specifying the user and archive name. The keyid will be queried from launchpad and everything else is set automatically. You can override any of the below settings by simply setting them as you would normally. For example:

```
logstash-ppa:
  pkgrepo.managed:
    - ppa: wolfnet/logstash
```

ppa_auth For Ubuntu PPAs there can be private PPAs that require authentication to access. For these PPAs the username/password can be passed as an HTTP Basic style username/password combination.

```
logstash-ppa:
  pkgrepo.managed:
    - ppa: wolfnet/logstash
    - ppa_auth: username:password
```

name On apt-based systems this must be the complete entry as it would be seen in the sources.list file. This can have a limited subset of components (i.e. 'main') which can be added/modified with the `comps` option.

```
precise-repo:
  pkgrepo.managed:
    - name: deb http://us.archive.ubuntu.com/ubuntu precise main
```

Note: The above example is intended as a more readable way of configuring the SLS, it is equivalent to the following:

```
'deb http://us.archive.ubuntu.com/ubuntu precise main':
  pkgrepo.managed
```

---

disabled [False] Toggles whether or not the repo is used for resolving dependencies and/or installing packages.

enabled [True] Included to reduce confusion due to YUM/DNF/Zypper's use of the `enabled` argument. If this is passed for an APT-based distro, then the reverse will be passed as `disabled`. For example, passing `enabled=False` will assume `disabled=True`.

architectures On apt-based systems, architectures can restrict the available architectures that the repository provides (e.g. only amd64). architectures should be a comma-separated list.

comps On apt-based systems, comps dictate the types of packages to be installed from the repository (e.g. main, nonfree, ...). For purposes of this, comps should be a comma-separated list.

file The filename for the .list that the repository is configured in. It is important to include the full-path AND make sure it is in a directory that APT will look in when handling packages

dist This dictates the release of the distro the packages should be built for. (e.g. unstable). This option is rarely needed.

keyid The KeyID or a list of KeyIDs of the GPG key to install. This option also requires the `keyserver` option to be set.

keyserver This is the name of the keyserver to retrieve gpg keys from. The `keyid` option must also be set for this option to work.

key_url URL to retrieve a GPG key from. Allows the usage of `http://`, `https://` as well as `salt://`.

Note: Use either `keyid`/`keyserver` or `key_url`, but not both.

key_text

The string representation of the GPG key to install.

New in version 2018.3.0.

Note: Use either `keyid`/`keyserver`, `key_url`, or `key_text` but not more than one method.

consolidate [False] If set to `True`, this will consolidate all sources definitions to the sources.list file, cleanup the now unused files, consolidate components (e.g. main) for the same URI, type, and architecture to a single line, and finally remove comments from the sources.list file. The consolidate will run every time the state is processed. The option only needs to be set on one repo managed by salt to take effect.

clean_file [False] If set to `True`, empty the file before config repo

Note: Use with care. This can be dangerous if multiple sources are configured in the same file.

New in version 2015.8.0.

refresh [True] If set to `False` this will skip refreshing the apt package database on debian based systems.

refresh_db [True] This argument has been deprecated. Please use `refresh` instead. The `refresh_db` argument will continue to work to ensure backwards compatibility, but we recommend using the preferred `refresh` argument instead.

require_in Set this to a list of pkg.installed or pkg.latest to trigger the running of apt-get update prior to attempting to install these packages. Setting a require in the pkg state will not work for this.

21.19.213 salt.states.portage_config

Management of Portage package configuration on Gentoo

A state module to manage Portage configuration on Gentoo

```
salt:
    portage_config.flags:
        - use:
            - openssl
```

`salt.states.portage_config.`**`flags`**(*name, use=None, accept_keywords=None, env=None, license=None, properties=None, unmask=False, mask=False*)

Enforce the given flags on the given package or `DEPEND` atom.

> **Warning:** In most cases, the affected package(s) need to be rebuilt in order to apply the changes.

name The name of the package or its DEPEND atom
use A list of `USE` flags
accept_keywords A list of keywords to accept. `~ARCH` means current host arch, and will be translated into a line without keywords
env A list of environment files
license A list of accepted licenses
properties A list of additional properties
unmask A boolean to unmask the package
mask A boolean to mask the package

21.19.214 salt.states.ports

Manage software from FreeBSD ports

New in version 2014.1.0.

Note: It may be helpful to use a higher timeout when running a *`ports.installed`* state, since compiling the port may exceed Salt's timeout.

```
salt -t 1200 '*' state.highstate
```

`salt.states.ports.`**`installed`**(*name, options=None*)

Verify that the desired port is installed, and that it was compiled with the desired options.

options Make sure that the desired non-default options are set

> **Warning:** Any build options not passed here assume the default values for the port, and are not just differences from the existing cached options from a previous `make config`.

Example usage:

```
security/nmap:
  ports.installed:
    - options:
      - IPV6: off
```

21.19.215 salt.states.postgres_cluster module

Management of PostgreSQL clusters

The postgres_cluster state module is used to manage PostgreSQL clusters. Clusters can be set as either absent or present

```
create cluster 9.3 main:
  postgres_cluster.present:
      - name: 'main'
      - version: '9.3'
```

`salt.states.postgres_cluster.`**`absent`**`(`*version*, *name*`)`

Ensure that the named cluster is absent

version Version of the postgresql server of the cluster to remove

name The name of the cluster to remove

New in version 2015.XX.

`salt.states.postgres_cluster.`**`present`**`(`*version*, *name*, *port=None*, *encoding=None*, *locale=None*, *datadir=None*`)`

Ensure that the named cluster is present with the specified properties. For more information about all of these options see man pg_createcluster(1)

version Version of the postgresql cluster

name The name of the cluster

port Cluster port

encoding The character encoding scheme to be used in this database

locale Locale with which to create cluster

datadir Where the cluster is stored

New in version 2015.XX.

21.19.216 salt.states.postgres_database

Management of PostgreSQL databases

The postgres_database module is used to create and manage Postgres databases. Databases can be set as either absent or present

```
frank:
  postgres_database.present
```

`salt.states.postgres_database.`**`absent`**`(`*name*, *user=None*, *maintenance_db=None*, *db_password=None*, *db_host=None*, *db_port=None*, *db_user=None*`)`

Ensure that the named database is absent

name The name of the database to remove

db_user database username if different from config or default

db_password user password if any password for a specified user

db_host Database host if different from config or default

db_port Database port if different from config or default

user System user all operations should be performed on behalf of

New in version 0.17.0.

`salt.states.postgres_database.`**`present`**(*name, tablespace=None, encoding=None, lc_collate=None, lc_ctype=None, owner=None, owner_recurse=False, template=None, user=None, maintenance_db=None, db_password=None, db_host=None, db_port=None, db_user=None*)

Ensure that the named database is present with the specified properties. For more information about all of these options see man createdb(1)

name The name of the database to manage
tablespace Default tablespace for the database
encoding The character encoding scheme to be used in this database
lc_collate The LC_COLLATE setting to be used in this database
lc_ctype The LC_CTYPE setting to be used in this database
owner The username of the database owner
owner_recurse Recurse owner change to all relations in the database
template The template database from which to build this database
user System user all operations should be performed on behalf of
db_user database username if different from config or default
db_password user password if any password for a specified user
db_host Database host if different from config or default
db_port Database port if different from config or default

New in version 0.17.0.

21.19.217 salt.states.postgres_extension

Management of PostgreSQL extensions

A module used to install and manage PostgreSQL extensions.

```
adminpack:
  postgres_extension.present
```

New in version 2014.7.0.

`salt.states.postgres_extension.`**`absent`**(*name, if_exists=None, restrict=None, cascade=None, user=None, maintenance_db=None, db_user=None, db_password=None, db_host=None, db_port=None*)

Ensure that the named extension is absent.

name Extension name of the extension to remove
if_exists Add if exist slug
restrict Add restrict slug
cascade Drop on cascade
user System user all operations should be performed on behalf of
maintenance_db Database to act on
db_user Database username if different from config or default
db_password User password if any password for a specified user
db_host Database host if different from config or default
db_port Database port if different from config or default

`salt.states.postgres_extension.`**`present`**(*name, if_not_exists=None, schema=None, ext_version=None, from_version=None, user=None, maintenance_db=None, db_user=None, db_password=None, db_host=None, db_port=None*)

Ensure that the named extension is present.

Note: Before you can use the state to load an extension into a database, the extension's supporting files must be already installed.

For more information about all of these options see `CREATE EXTENSION` SQL command reference in the PostgreSQL documentation.

name The name of the extension to be installed
if_not_exists Add an `IF NOT EXISTS` parameter to the DDL statement
schema Schema to install the extension into
ext_version Version to install
from_version Old extension version if already installed
user System user all operations should be performed on behalf of
maintenance_db Database to act on
db_user Database username if different from config or default
db_password User password if any password for a specified user
db_host Database host if different from config or default
db_port Database port if different from config or default

21.19.218 salt.states.postgres_group

Management of PostgreSQL groups (roles)

The postgres_group module is used to create and manage Postgres groups.

```
frank:
  postgres_group.present
```

`salt.states.postgres_group.`**`absent`**(*name, user=None, maintenance_db=None, db_password=None, db_host=None, db_port=None, db_user=None*)

Ensure that the named group is absent
name The groupname of the group to remove
user System user all operations should be performed on behalf of

New in version 0.17.0.

db_user database username if different from config or defaul
db_password user password if any password for a specified user
db_host Database host if different from config or default
db_port Database port if different from config or default

`salt.states.postgres_group.`**`present`**(*name, createdb=None, createroles=None, encrypted=None, superuser=None, inherit=None, login=None, replication=None, password=None, refresh_password=None, groups=None, user=None, maintenance_db=None, db_password=None, db_host=None, db_port=None, db_user=None*)

Ensure that the named group is present with the specified privileges Please note that the user/group notion in postgresql is just abstract, we have roles, where users can be seen as roles with the `LOGIN` privilege and groups the others.
name The name of the group to manage
createdb Is the group allowed to create databases?
createroles Is the group allowed to create other roles/users
encrypted Should the password be encrypted in the system catalog?
login Should the group have login perm

inherit Should the group inherit permissions
superuser Should the new group be a "superuser"
replication Should the new group be allowed to initiate streaming replication
password The group's password It can be either a plain string or a md5 postgresql hashed password:

```
'md5{MD5OF({password}{role}}'
```

If encrypted is `None` or `True`, the password will be automatically encrypted to the previous format if it is not already done.

refresh_password Password refresh flag

Boolean attribute to specify whether to password comparison check should be performed.

If refresh_password is `True`, the password will be automatically updated without extra password change check.

This behaviour makes it possible to execute in environments without superuser access available, e.g. Amazon RDS for PostgreSQL

groups A string of comma separated groups the group should be in
user System user all operations should be performed on behalf of

New in version 0.17.0.

db_user database username if different from config or default
db_password user password if any password for a specified user
db_host Database host if different from config or default
db_port Database port if different from config or default

21.19.219 salt.states.postgres_initdb

Initialization of PostgreSQL data directory

The postgres_initdb module is used to initialize the postgresql data directory.

New in version 2016.3.0.

```
pgsql-data-dir:
  postgres_initdb.present:
    - name: /var/lib/pgsql/data
    - auth: password
    - user: postgres
    - password: strong_password
    - encoding: UTF8
    - locale: C
    - runas: postgres
```

`salt.states.postgres_initdb.`**`present`**(*name, user=None, password=None, auth='password', encoding='UTF8', locale=None, runas=None, waldir=None, checksums=False*)

Initialize the PostgreSQL data directory

name The name of the directory to initialize
user The database superuser name
password The password to set for the postgres user
auth The default authentication method for local connections
encoding The default encoding for new databases
locale The default locale for new databases
waldir The transaction log (WAL) directory (default is to keep WAL inside the data directory)

New in version 2019.2.0.

checksums If True, the cluster will be created with data page checksums.

> **Note:** Data page checksums are supported since PostgreSQL 9.3.

New in version 2019.2.0.

runas The system user the operation should be performed on behalf of

21.19.220 salt.states.postgres_language

Management of PostgreSQL languages

The postgres_language module is used to create and manage Postgres languages. Languages can be set as either absent or present

New in version 2016.3.0.

```
plpgsql:
  postgres_language.present:
    - maintenance_db: testdb
```

```
plpgsql:
  postgres_language.absent:
    - maintenance_db: testdb
```

`salt.states.postgres_language.`**`absent`**`(`*name*, *maintenance_db*, *user=None*, *db_password=None*, *db_host=None*, *db_port=None*, *db_user=None*`)`

Ensure that a named language is absent in the specified database.

name The name of the language to remove

maintenance_db The name of the database in which the language is to be installed

user System user all operations should be performed on behalf of

db_user database username if different from config or default

db_password user password if any password for a specified user

db_host Database host if different from config or default

db_port Database port if different from config or default

`salt.states.postgres_language.`**`present`**`(`*name*, *maintenance_db*, *user=None*, *db_password=None*, *db_host=None*, *db_port=None*, *db_user=None*`)`

Ensure that a named language is present in the specified database.

name The name of the language to install

maintenance_db The name of the database in which the language is to be installed

user System user all operations should be performed on behalf of

db_user database username if different from config or default

db_password user password if any password for a specified user

db_host Database host if different from config or default

db_port Database port if different from config or default

21.19.221 salt.states.postgres_privileges

Management of PostgreSQL Privileges

The postgres_privileges module is used to manage Postgres privileges. Privileges can be set as either absent or present.

Privileges can be set on the following database object types:

- database
- schema
- tablespace
- table
- sequence
- language
- group

Setting the grant option is supported as well.

New in version 2016.3.0.

```
baruwa:
  postgres_privileges.present:
    - object_name: awl
    - object_type: table
    - privileges:
      - SELECT
      - INSERT
      - DELETE
    - grant_option: False
    - prepend: public
    - maintenance_db: testdb
```

```
andrew:
  postgres_privileges.present:
    - object_name: admins
    - object_type: group
    - grant_option: False
    - maintenance_db: testdb
```

```
baruwa:
  postgres_privileges.absent:
    - object_name: awl
    - object_type: table
    - privileges:
      - SELECT
      - INSERT
      - DELETE
    - prepend: public
    - maintenance_db: testdb
```

```
andrew:
  postgres_privileges.absent:
    - object_name: admins
    - object_type: group
    - maintenance_db: testdb
```

`salt.states.postgres_privileges.`**`absent`**`(`*name, object_name, object_type, privileges=None, prepend='public', maintenance_db=None, user=None, db_password=None, db_host=None, db_port=None, db_user=None*`)`

Revoke the requested privilege(s) on the specificed object(s)

name Name of the role whose privileges should be revoked

object_name Name of the object on which the revoke is to be performed

object_type The object type, which can be one of the following:

- table
- sequence
- schema
- tablespace
- language
- database
- group
- function

View permissions should specify *object_type: table*.

privileges Comma separated list of privileges to revoke, from the list below:

- INSERT
- CREATE
- TRUNCATE
- CONNECT
- TRIGGER
- SELECT
- USAGE
- TEMPORARY
- UPDATE
- EXECUTE
- REFERENCES
- DELETE
- ALL

note privileges should not be set when revoking group membership

prepend Table and Sequence object types live under a schema so this should be provided if the object is not under the default *public* schema

maintenance_db The name of the database in which the language is to be installed

user System user all operations should be performed on behalf of

db_user database username if different from config or default

db_password user password if any password for a specified user

db_host Database host if different from config or default

db_port Database port if different from config or default

`salt.states.postgres_privileges.`**`present`**`(`*name, object_name, object_type, privileges=None, grant_option=None, prepend='public', maintenance_db=None, user=None, db_password=None, db_host=None, db_port=None, db_user=None*`)`

Grant the requested privilege(s) on the specified object to a role

name Name of the role to which privileges should be granted

object_name Name of the object on which the grant is to be performed. 'ALL' may be used for objects of type 'table' or 'sequence'.

object_type The object type, which can be one of the following:

- table
- sequence
- schema
- tablespace

- language
- database
- group
- function

View permissions should specify *object_type: table*.

privileges List of privileges to grant, from the list below:

- INSERT
- CREATE
- TRUNCATE
- CONNECT
- TRIGGER
- SELECT
- USAGE
- TEMPORARY
- UPDATE
- EXECUTE
- REFERENCES
- DELETE
- ALL

note privileges should not be set when granting group membership

grant_option If grant_option is set to True, the recipient of the privilege can in turn grant it to others

prepend Table and Sequence object types live under a schema so this should be provided if the object is not under the default *public* schema

maintenance_db The name of the database in which the language is to be installed

user System user all operations should be performed on behalf of

db_user database username if different from config or default

db_password user password if any password for a specified user

db_host Database host if different from config or default

db_port Database port if different from config or default

21.19.222 salt.states.postgres_schema

Management of PostgreSQL schemas

The postgres_schemas module is used to create and manage Postgres schemas.

```
public:
  postgres_schema.present 'dbname' 'name'
```

`salt.states.postgres_schema.`**`absent`**(*dbname*, *name*, *user=None*, *db_user=None*, *db_password=None*, *db_host=None*, *db_port=None*)

Ensure that the named schema is absent.

dbname The database's name will work on

name The name of the schema to remove

user system user all operations should be performed on behalf of

db_user database username if different from config or default

db_password user password if any password for a specified user

db_host Database host if different from config or default

db_port Database port if different from config or default

`salt.states.postgres_schema.`**`present`**(*dbname*, *name*, *owner=None*, *user=None*, *db_user=None*, *db_password=None*, *db_host=None*, *db_port=None*)

Ensure that the named schema is present in the database.

dbname The database's name will work on

name The name of the schema to manage
user system user all operations should be performed on behalf of
db_user database username if different from config or default
db_password user password if any password for a specified user
db_host Database host if different from config or default
db_port Database port if different from config or default

21.19.223 salt.states.postgres_tablespace

Management of PostgreSQL tablespace

A module used to create and manage PostgreSQL tablespaces.

```
ssd-tablespace:
  postgres_tablespace.present:
    - name: indexes
    - directory: /mnt/ssd-data
```

New in version 2015.8.0.

`salt.states.postgres_tablespace.`**`absent`**`(`*name, user=None, maintenance_db=None, db_user=None, db_password=None, db_host=None, db_port=None*`)`

Ensure that the named tablespace is absent.
name The name of the tablespace to remove
user System user all operations should be performed on behalf of
maintenance_db Database to act on
db_user Database username if different from config or defaul
db_password User password if any password for a specified user
db_host Database host if different from config or default
db_port Database port if different from config or default

`salt.states.postgres_tablespace.`**`present`**`(`*name, directory, options=None, owner=None, user=None, maintenance_db=None, db_password=None, db_host=None, db_port=None, db_user=None*`)`

Ensure that the named tablespace is present with the specified properties. For more information about all of these options run `man 7 create_tablespace`.
name The name of the tablespace to create/manage.
directory The directory where the tablespace will be located, must already exist
options A dictionary of options to specify for the tablespace. Currently, the only tablespace options supported are `seq_page_cost` and `random_page_cost`. Default values are shown in the example below:

```
my_space:
  postgres_tablespace.present:
    - directory: /srv/my_tablespace
    - options:
        seq_page_cost: 1.0
        random_page_cost: 4.0
```

owner The database user that will be the owner of the tablespace. Defaults to the user executing the command (i.e. the *user* option)
user System user all operations should be performed on behalf of
maintenance_db Database to act on
db_user Database username if different from config or default
db_password User password if any password for a specified user

db_host Database host if different from config or default
db_port Database port if different from config or default

21.19.224 salt.states.postgres_user

Management of PostgreSQL users (roles)

The postgres_users module is used to create and manage Postgres users.

```
frank:
  postgres_user.present
```

`salt.states.postgres_user.`**`absent`**`(`*name*, *user=None*, *maintenance_db=None*, *db_password=None*, *db_host=None*, *db_port=None*, *db_user=None*`)`

Ensure that the named user is absent
name The username of the user to remove
user System user all operations should be performed on behalf of

New in version 0.17.0.
db_user database username if different from config or default
db_password user password if any password for a specified user
db_host Database host if different from config or default
db_port Database port if different from config or default

`salt.states.postgres_user.`**`present`**`(`*name*, *createdb=None*, *createroles=None*, *encrypted=None*, *superuser=None*, *replication=None*, *inherit=None*, *login=None*, *password=None*, *default_password=None*, *refresh_password=None*, *valid_until=None*, *groups=None*, *user=None*, *maintenance_db=None*, *db_password=None*, *db_host=None*, *db_port=None*, *db_user=None*`)`

Ensure that the named user is present with the specified privileges Please note that the user/group notion in postgresql is just abstract, we have roles, where users can be seens as roles with the LOGIN privilege and groups the others.
name The name of the system user to manage.
createdb Is the user allowed to create databases?
createroles Is the user allowed to create other users?
encrypted Should the password be encrypted in the system catalog?
login Should the group have login perm
inherit Should the group inherit permissions
superuser Should the new user be a "superuser"
replication Should the new user be allowed to initiate streaming replication
password The system user's password. It can be either a plain string or a md5 postgresql hashed password:

```
'md5{MD5OF({password}{role}}'
```

If encrypted is None or True, the password will be automatically encrypted to the previous format if it is not already done.
default_password The password used only when creating the user, unless password is set.

New in version 2016.3.0.
refresh_password Password refresh flag

Boolean attribute to specify whether to password comparison check should be performed.

If refresh_password is `True`, the password will be automatically updated without extra password change check.

This behaviour makes it possible to execute in environments without superuser access available, e.g. Amazon RDS for PostgreSQL

valid_until A date and time after which the role's password is no longer valid.

groups A string of comma separated groups the user should be in

user System user all operations should be performed on behalf of

New in version 0.17.0.

db_user Postgres database username, if different from config or default.

db_password Postgres user's password, if any password, for a specified db_user.

db_host Postgres database host, if different from config or default.

db_port Postgres database port, if different from config or default.

21.19.225 salt.states.powerpath

Powerpath configuration support

Allows configuration of EMC Powerpath. Currently only addition/deletion of licenses is supported.

```
key:
  powerpath.license_present: []
```

`salt.states.powerpath.`**`license_absent`**`(`*name*`)`

Ensures that the specified PowerPath license key is absent on the host.

name The license key to ensure is absent

`salt.states.powerpath.`**`license_present`**`(`*name*`)`

Ensures that the specified PowerPath license key is present on the host.

name The license key to ensure is present

21.19.226 salt.states.probes

Network Probes

Configure RPM (JunOS)/SLA (Cisco) probes on the device via NAPALM proxy.

codeauthor Mircea Ulinic <ping@mirceaulinic.net> & Jerome Fleury <jf@cloudflare.com>

maturity new

depends napalm

platform unix

Dependencies

- *`napalm probes management module`*

`salt.states.probes.`**`managed`**`(`*name*, *probes*, *defaults=None*`)`

Ensure the networks device is configured as specified in the state SLS file. Probes not specified will be removed, while probes not confiured as expected will trigger config updates.

Parameters

- **`probes`** – Defines the probes as expected to be configured on the device. In order to ease the configuration and avoid repeating the same parameters for each probe, the next parameter (defaults) can be used, providing common characteristics.
- **`defaults`** – Specifies common parameters for the probes.

SLS Example:

```
rpmprobes:
    probes.managed:
        - probes:
            probe_name1:
                probe1_test1:
                    source: 192.168.0.2
                    target: 192.168.0.1
                probe1_test2:
                    target: 172.17.17.1
                probe1_test3:
                    target: 8.8.8.8
                    probe_type: http-ping
            probe_name2:
                probe2_test1:
                    test_interval: 100
        - defaults:
            target: 10.10.10.10
            probe_count: 15
            test_interval: 3
            probe_type: icmp-ping
```

In the probes configuration, the only mandatory attribute is *target* (specified either in probes configuration, either in the defaults dictionary). All the other parameters will use the operating system defaults, if not provided:

- `source` - Specifies the source IP Address to be used during the tests. If not specified will use the IP Address of the logical interface loopback0.
- `target` - Destination IP Address.
- `probe_count` - Total number of probes per test (1..15). System defaults: 1 on both JunOS & Cisco.
- `probe_interval` - Delay between tests (0..86400 seconds). System defaults: 3 on JunOS, 5 on Cisco.
- `probe_type` - Probe request type. Available options:
  - icmp-ping
  - tcp-ping
  - udp-ping

Using the example configuration above, after running the state, on the device will be configured 4 probes, with the following properties:

```
probe_name1:
    probe1_test1:
        source: 192.168.0.2
        target: 192.168.0.1
        probe_count: 15
        test_interval: 3
        probe_type: icmp-ping
    probe1_test2:
        target: 172.17.17.1
        probe_count: 15
        test_interval: 3
        probe_type: icmp-ping
    probe1_test3:
        target: 8.8.8.8
        probe_count: 15
        test_interval: 3
        probe_type: http-ping
probe_name2:
    probe2_test1:
```

(continues on next page)

(continued from previous page)

```
        target: 10.10.10.10
        probe_count: 15
        test_interval: 3
        probe_type: icmp-ping
```

21.19.227 salt.states.process

Process Management

Ensure a process matching a given pattern is absent.

```
httpd-absent:
  process.absent:
    - name: apache2
```

`salt.states.process.`**`absent`**`(`*name*, *user=None*, *signal=None*`)`

Ensures that the named command is not running.

name The pattern to match.

user The user to which the process belongs

signal Signal to send to the process(es).

21.19.228 salt.states.proxy module

Allows you to manage proxy settings on minions

Setup proxy settings on minions

```
192.168.1.4:
  proxy.managed:
    - port: 3128
    - bypass_domains:
        - localhost
        - 127.0.0.1
```

`salt.states.proxy.`**`managed`**`(`*name*, *port*, *services=None*, *user=None*, *password=None*, *bypass_domains=None*, *network_service='Ethernet'*`)`

Manages proxy settings for this mininon

name The proxy server to use

port The port used by the proxy server

services A list of the services that should use the given proxy settings, valid services include http, https and ftp. If no service is given all of the valid services will be used.

user The username to use for the proxy server if required

password The password to use for the proxy server if required

bypass_domains An array of the domains that should bypass the proxy

network_service The network service to apply the changes to, this only necessary on macOS

21.19.229 salt.states.pushover

Send a message to PushOver

This state is useful for sending messages to PushOver during state runs.

New in version 2015.5.0.

```
pushover-message:
  pushover.post_message:
    - user: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
    - token: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
    - title: Salt Returner
    - device: phone
    - priority: -1
    - expire: 3600
    - retry: 5
    - message: 'This state was executed successfully.'
```

The api key can be specified in the master or minion configuration like below: .. code-block:: yaml

pushover: token: peWcBiMOS9HrZG15peWcBiMOS9HrZG15

`salt.states.pushover.`**`post_message`**(*name*, *user=None*, *device=None*, *message=None*, *title=None*, *priority=None*, *expire=None*, *retry=None*, *sound=None*, *api_version=1*, *token=None*)

Send a message to a PushOver channel.

```
pushover-message:
  pushover.post_message:
    - user: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
    - token: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
    - title: Salt Returner
    - device: phone
    - priority: -1
    - expire: 3600
    - retry: 5
```

The following parameters are required:

name The unique name for this event.

user The user or group of users to send the message to. Must be ID of user, not name or email address.

message The message that is to be sent to the PushOver channel.

The following parameters are optional:

title The title to use for the message.

device The device for the user to send the message to.

priority The priority for the message.

expire The message should expire after specified amount of seconds.

retry The message should be resent this many times.

token The token for PushOver to use for authentication, if not specified in the configuration options of master or minion.

21.19.230 salt.states.pyenv

Managing python installations with pyenv

This module is used to install and manage python installations with pyenv. Different versions of python can be installed, and uninstalled. pyenv will be installed automatically the first time it is needed and can be updated later. This module will *not* automatically install packages which pyenv will need to compile the versions of python.

If pyenv is run as the root user then it will be installed to /usr/local/pyenv, otherwise it will be installed to the users ~/.pyenv directory. To make pyenv available in the shell you may need to add the pyenv/shims and pyenv/bin directories to the users PATH. If you are installing as root and want other users to be able to access pyenv then you will need to add pyenv_ROOT to their environment.

This is how a state configuration could look like:

```
pyenv-deps:
  pkg.installed:
    - pkgs:
      - make
      - build-essential
      - libssl-dev
      - zlib1g-dev
      - libbz2-dev
      - libreadline-dev
      - libsqlite3-dev
      - wget
      - curl
      - llvm
python-2.6:
  pyenv.absent:
    - require:
      - pkg: pyenv-deps

python-2.7.6:
  pyenv.installed:
    - default: True
    - require:
      - pkg: pyenv-deps
```

Note: Git needs to be installed and available via PATH if pyenv is to be installed automatically by the module.

`salt.states.pyenv.`**`absent`**(*name*, *user=None*)

Verify that the specified python is not installed with pyenv. pyenv is installed if necessary.

name The version of python to uninstall

user: **None** The user to run pyenv as.

> New in version 0.17.0.

New in version 0.16.0.

`salt.states.pyenv.`**`install_pyenv`**(*name*, *user=None*)

Install pyenv if not installed. Allows you to require pyenv be installed prior to installing the plugins. Useful if you want to install pyenv plugins via the git or file modules and need them installed before installing any rubies.

Use the pyenv.root configuration option to set the path for pyenv if you want a system wide install that is not in a user home dir.

user: **None** The user to run pyenv as.

`salt.states.pyenv.`**`installed`**(*name*, *default=False*, *user=None*)

Verify that the specified python is installed with pyenv. pyenv is installed if necessary.

name The version of python to install

default [False] Whether to make this python the default.

user: **None** The user to run pyenv as.

> New in version 0.17.0.

New in version 0.16.0.

21.19.231 salt.states.pyrax_queues

Manage Rackspace Queues

New in version 2015.5.0.

Create and destroy Rackspace queues. Be aware that this interacts with Rackspace's services, and so may incur charges.

This module uses `pyrax`, which can be installed via package, or pip. This module is greatly inspired by boto_* modules from SaltStack code source.

```
myqueue:
    pyrax_queues.present:
        - provider: my-pyrax

myqueue:
    pyrax_queues.absent:
        - provider: my-pyrax
```

`salt.states.pyrax_queues.`**`absent`**`(`*name*, *provider*`)`
Ensure the named Rackspace queue is deleted.
name Name of the Rackspace queue.
provider Salt Cloud provider

`salt.states.pyrax_queues.`**`present`**`(`*name*, *provider*`)`
Ensure the RackSpace queue exists.
name Name of the Rackspace queue.
provider Salt Cloud Provider

21.19.232 salt.states.quota

Management of POSIX Quotas

The quota can be managed for the system:

```
/:
  quota.mode:
    mode: off
    quotatype: user
```

`salt.states.quota.`**`mode`**`(`*name*, *mode*, *quotatype*`)`
Set the quota for the system
name The filesystem to set the quota mode on
mode Whether the quota system is on or off
quotatype Must be `user` or `group`

21.19.233 salt.states.rabbitmq_cluster

Manage RabbitMQ Clusters

Example:

```
rabbit@rabbit.example.com:
  rabbitmq_cluster.join:
    - user: rabbit
    - host: rabbit.example.com
```

`salt.states.rabbitmq_cluster.`**`join`**(*name, host, user='rabbit', ram_node=None, runas='root'*)

This function is an alias of `joined`.

Ensure the current node joined to a cluster with node user@host

name Irrelevant, not used (recommended: user@host)

user The user of node to join to (default: rabbit)

host The host of node to join to

ram_node Join node as a RAM node

runas The user to run the rabbitmq command as

`salt.states.rabbitmq_cluster.`**`joined`**(*name, host, user='rabbit', ram_node=None, runas='root'*)

Ensure the current node joined to a cluster with node user@host

name Irrelevant, not used (recommended: user@host)

user The user of node to join to (default: rabbit)

host The host of node to join to

ram_node Join node as a RAM node

runas The user to run the rabbitmq command as

21.19.234 salt.states.rabbitmq_plugin

Manage RabbitMQ Plugins

New in version 2014.1.0.

Example:

```
some_plugin:
  rabbitmq_plugin.enabled: []
```

`salt.states.rabbitmq_plugin.`**`disabled`**(*name, runas=None*)

Ensure the RabbitMQ plugin is disabled.

name The name of the plugin

runas The user to run the rabbitmq-plugin command as

`salt.states.rabbitmq_plugin.`**`enabled`**(*name, runas=None*)

Ensure the RabbitMQ plugin is enabled.

name The name of the plugin

runas The user to run the rabbitmq-plugin command as

21.19.235 salt.states.rabbitmq_policy

Manage RabbitMQ Policies

maintainer Benn Eichhorn <benn@getlocalmeasure.com>

maturity new

platform all

Example:

```
rabbit_policy:
  rabbitmq_policy.present:
    - name: HA
    - pattern: '.*'
    - definition: '{"ha-mode": "all"}'
```

`salt.states.rabbitmq_policy.`**`absent`**`(`*name*, *vhost='/'*, *runas=None*`)`

Ensure the named policy is absent

Reference: http://www.rabbitmq.com/ha.html
name The name of the policy to remove
runas Name of the user to run the command as

`salt.states.rabbitmq_policy.`**`present`**`(`*name*, *pattern*, *definition*, *priority=0*, *vhost='/'*, *runas=None*, *apply_to=None*`)`

Ensure the RabbitMQ policy exists.

Reference: http://www.rabbitmq.com/ha.html
name Policy name
pattern A regex of queues to apply the policy to
definition A json dict describing the policy
priority Priority (defaults to 0)
vhost Virtual host to apply to (defaults to '/')
runas Name of the user to run the command as
apply_to Apply policy to 'queues', 'exchanges' or 'all' (default to 'all')

21.19.236 salt.states.rabbitmq_upstream

Manage RabbitMQ Upstreams

Example:

```
rabbit_upstream:
  rabbitmq_upstream.present:
  - name: upstream_1
  - uri: amqp://my_user:my_password@rabbitmq_host
  - trust_user_id: True
  - ack_mode: on-confirm
  - max_hops: 1
```

New in version 3000.

`salt.states.rabbitmq_upstream.`**`absent`**`(`*name*, *runas=None*`)`

Ensure the named upstream is absent.

Parameters

- **name** (*str*) – The name of the upstream to remove
- **runas** (*str*) – User to run the command

New in version 3000.

`salt.states.rabbitmq_upstream.`**`present`**`(`*name*, *uri*, *prefetch_count=None*, *reconnect_delay=None*, *ack_mode=None*, *trust_user_id=None*, *exchange=None*, *max_hops=None*, *expires=None*, *message_ttl=None*, *ha_policy=None*, *queue=None*, *runas=None*`)`

Ensure the RabbitMQ upstream exists.

Parameters

- **name** (*str*) – The name of the upstream connection
- **uri** (*str*) – The URI to connect to. If upstream is a cluster and can have several URIs, you can enter them here separated by spaces. Examples: - amqp://user:password@server_name - amqp://user:password@server_name/vhost When connecting with SSL, several URI-parameters need also be specified: - cacertfile = /path/to/cacert.pem - certfile = /path/to/cert.pem - keyfile = /part/to/key.pem -

verity = verify_peer - fail_if_no_peer_cert = true | false - auth_mechanism = external Example: - amqp://user:password@server_name?cacertfile=/path/to/cacert.pem& certfile=/path/to/cert.pem&keyfile=/path/to/key.pem&verify=verify_peer - amqp://server-name?cacertfile=/path/to/cacert.pem&certfile=/path/to/cert.pem& keyfile=/path/to/key.pem&verify=verify_peer&fail_if_no_peer_cert=true&auth_mechanism=external

- **prefetch_count** (*int*) – Maximum number of unacknowledged messages that may be in flight over a federation link at one time. Default: 1000
- **reconnect_delay** (*int*) – Time in seconds to wait after a network link goes down before attempting reconnection. Default: 5
- **ack_mode** (*str*) – The following values are allowed: on-confirm: Messages are acknowledged to the upstream broker after they have been confirmed downstream. Handles network errors and broker failures without losing messages. The slowest option, and the default. on-publish: Messages are acknowledged to the upstream broker after they have been published downstream. Handles network errors without losing messages, but may lose messages in the event of broker failures. no-ack: Message acknowledgements are not used. The fastest option, but you may lose messages in the event of network or broker failures.
- **trust_user_id** (*bool*) – Set `True` to preserve the "user-id" field across a federation link, even if the user-id does not match that used to republish the message. Set to `False` to clear the "user-id" field when messages are federated. Only set this to `True` if you trust the upstream broker not to forge user-ids.
- **exchange** (*str*) – The name of the upstream exchange. Default is to use the same name as the federated exchange.
- **max_hops** (*int*) – Maximum number of federation links that messages can traverse before being dropped. Defaults to 1 if not set.
- **expires** (*int*) – Time in milliseconds that the upstream should remember about this node for. After this time all upstream state will be removed. Set to `None` (Default) to mean "forever".
- **message_ttl** (*int*) – Time in milliseconds that undelivered messages should be held upstream when there is a network outage or backlog. Set to `None` (default) to mean "forever".
- **ha_policy** (*str*) – Determines the "x-ha-policy"-argument for the upstream queue for a federated exchange. Default is "none" meaning the queue is not HA.
- **queue** (*str*) – The name of the upstream queue. Default is to use the same name as the federated queue.

New in version 3000.

21.19.237 salt.states.rabbitmq_user

Manage RabbitMQ Users

Example:

```
rabbit_user:
  rabbitmq_user.present:
    - password: password
    - force: True
    - tags:
      - monitoring
      - user
    - perms:
      - '/':
        - '.*'
```

(continues on next page)

(continued from previous page)

```
        - '.*'
        - '.*'
    - runas: rabbitmq
```

`salt.states.rabbitmq_user.`**`absent`**`(`*name*, *runas=None*`)`

Ensure the named user is absent

name The name of the user to remove

runas User to run the command

`salt.states.rabbitmq_user.`**`present`**`(`*name*, *password=None*, *force=False*, *tags=None*, *perms=()*, *runas=None*`)`

Ensure the RabbitMQ user exists.

name User name

password User's password, if one needs to be set

force If user exists, forcibly change the password

tags Optional list of tags for the user

perms A list of dicts with vhost keys and 3-tuple values

runas Name of the user to run the command

21.19.238 salt.states.rabbitmq_vhost

Manage RabbitMQ Virtual Hosts

Example:

```
virtual_host:
  rabbitmq_vhost.present:
    - user: rabbit_user
    - conf: .*
    - write: .*
    - read: .*
```

`salt.states.rabbitmq_vhost.`**`absent`**`(`*name*`)`

Ensure the RabbitMQ Virtual Host is absent

name Name of the Virtual Host to remove

runas User to run the command

Deprecated since version 2015.8.0.

`salt.states.rabbitmq_vhost.`**`present`**`(`*name*`)`

Ensure the RabbitMQ VHost exists.

name VHost name

user Initial user permission to set on the VHost, if present

Deprecated since version 2015.8.0.

owner Initial owner permission to set on the VHost, if present

Deprecated since version 2015.8.0.

conf Initial conf string to apply to the VHost and user. Defaults to .*

Deprecated since version 2015.8.0.

write Initial write permissions to apply to the VHost and user. Defaults to .*

Deprecated since version 2015.8.0.

read Initial read permissions to apply to the VHost and user. Defaults to .*

Deprecated since version 2015.8.0.

runas Name of the user to run the command

Deprecated since version 2015.8.0.

21.19.239 salt.states.rbac_solaris

Management of Solaris RBAC

maintainer Jorge Schrauwen <sjorge@blackdot.be>

maturity new

depends rbac_solaris,solaris_user

platform solaris,illumos

New in version 2016.11.0.

```
sjorge:
  rbac.managed:
    - roles:
        - netcfg
    - profiles:
        - System Power
    - authorizations:
        - solaris.audit.*
```

`salt.states.rbac_solaris.managed`(*name*, *roles=None*, *profiles=None*, *authorizations=None*)

Manage RBAC properties for user

name [string] username

roles [list] list of roles for user

profiles [list] list of profiles for user

authorizations [list] list of authorizations for user

> **Warning:** All existing roles, profiles and authorizations will be replaced! An empty list will remove everything.
>
> Set the property to *None* to not manage it.

21.19.240 salt.states.rbenv

Managing Ruby installations with rbenv

This module is used to install and manage ruby installations with rbenv and the ruby-build plugin. Different versions of ruby can be installed, and uninstalled. Rbenv will be installed automatically the first time it is needed and can be updated later. This module will *not* automatically install packages which rbenv will need to compile the versions of ruby. If your version of ruby fails to install, refer to the ruby-build documentation to verify you are not missing any dependencies: https://github.com/rbenv/ruby-build/wiki

If rbenv is run as the root user then it will be installed to /usr/local/rbenv, otherwise it will be installed to the users ~/.rbenv directory. To make rbenv available in the shell you may need to add the rbenv/shims and rbenv/bin directories to the users PATH. If you are installing as root and want other users to be able to access rbenv then you will need to add RBENV_ROOT to their environment.

The following state configuration demonstrates how to install Ruby 1.9.x and 2.x using rbenv on Ubuntu/Debian:

```
rbenv-deps:
  pkg.installed:
    - names:
      - bash
      - git
      - openssl
      - libssl-dev
      - make
      - curl
      - autoconf
      - bison
      - build-essential
      - libffi-dev
      - libyaml-dev
      - libreadline6-dev
      - zlib1g-dev
      - libncurses5-dev

ruby-1.9.3-p429:
  rbenv.absent:
    - require:
      - pkg: rbenv-deps

ruby-2.0.0-p598:
  rbenv.installed:
    - default: True
    - require:
      - pkg: rbenv-deps
```

`salt.states.rbenv.`**`absent`**`(`*name*, *user=None*`)`

Verify that the specified ruby is not installed with rbenv. Rbenv is installed if necessary.

name The version of ruby to uninstall

user: **None** The user to run rbenv as.

New in version 0.17.0.

New in version 0.16.0.

`salt.states.rbenv.`**`install_rbenv`**`(`*name*, *user=None*`)`

Install rbenv if not installed. Allows you to require rbenv be installed prior to installing the plugins. Useful if you want to install rbenv plugins via the git or file modules and need them installed before installing any rubies.

Use the rbenv.root configuration option to set the path for rbenv if you want a system wide install that is not in a user home dir.

user: **None** The user to run rbenv as.

`salt.states.rbenv.`**`installed`**`(`*name*, *default=False*, *user=None*`)`

Verify that the specified ruby is installed with rbenv. Rbenv is installed if necessary.

name The version of ruby to install

default [False] Whether to make this ruby the default.

user: **None** The user to run rbenv as.

New in version 0.17.0.

New in version 0.16.0.

21.19.241 salt.states.rdp

Manage RDP Service on Windows servers

`salt.states.rdp.`**`disabled`**`(`*name*`)`
Disable the RDP service

`salt.states.rdp.`**`enabled`**`(`*name*`)`
Enable the RDP service and make sure access to the RDP port is allowed in the firewall configuration

21.19.242 salt.states.redismod

Management of Redis server

New in version 2014.7.0.

depends

- redis Python module

configuration See `salt.modules.redis` for setup instructions.

```
key_in_redis:
  redis.string:
    - value: string data
```

The redis server information specified in the minion config file can be overridden in states using the following arguments: `host`, `post`, `db`, `password`.

```
key_in_redis:
  redis.string:
    - value: string data
    - host: localhost
    - port: 6379
    - db: 0
    - password: somuchkittycat
```

`salt.states.redismod.`**`absent`**`(`*name, keys=None, **connection_args*`)`
Ensure key absent from redis
name Key to ensure absent from redis
keys list of keys to ensure absent, name will be ignored if this is used

`salt.states.redismod.`**`slaveof`**`(`*name, sentinel_host=None, sentinel_port=None, sentinel_password=None, **connection_args*`)`
Set this redis instance as a slave.
name Master to make this a slave of
sentinel_host Ip of the sentinel to check for the master
sentinel_port Port of the sentinel to check for the master

`salt.states.redismod.`**`string`**`(`*name, value, expire=None, expireat=None, **connection_args*`)`
Ensure that the key exists in redis with the value specified
name Redis key to manage
value Data to persist in key
expire Sets time to live for key in seconds
expireat Sets expiration time for key via UNIX timestamp, overrides *expire*

21.19.243 salt.states.reg

Manage the Windows registry

Many python developers think of registry keys as if they were python keys in a dictionary which is not the case. The windows registry is broken down into the following components:

Hives

This is the top level of the registry. They all begin with HKEY.

- HKEY_CLASSES_ROOT (HKCR)
- HKEY_CURRENT_USER(HKCU)
- HKEY_LOCAL MACHINE (HKLM)
- HKEY_USER (HKU)
- HKEY_CURRENT_CONFIG

Keys

Hives contain keys. These are basically the folders beneath the hives. They can contain any number of subkeys.

When passing the hivekey values they must be quoted correctly depending on the backslashes being used (\ vs \\). The way backslashes are handled in the state file is different from the way they are handled when working on the CLI. The following are valid methods of passing the hivekey:

Using single backslashes: HKLMSOFTWAREPython ’HKLMSOFTWAREPython’

Using double backslashes: ”HKLM\SOFTWARE\Python”

Values or Entries

Values or Entries are the name/data pairs beneath the keys and subkeys. All keys have a default name/data pair. The name is `(Default)` with a displayed value of `(value not set)`. The actual value is Null.

Example

The following example is taken from the windows startup portion of the registry:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run]
"RTHDVCPL"="\"C:\\Program Files\\Realtek\\Audio\\HDA\\RtkNGUI64.exe\" -s"
"NvBackend"="\"C:\\Program Files (x86)\\NVIDIA Corporation\\Update Core\\NvBackend.
↪exe\""
"BTMTrayAgent"="rundll32.exe \"C:\\Program Files (x86)\\Intel\\Bluetooth\\btmshellex.
↪dll\",TrayApp"
```

In this example these are the values for each:

Hive: `HKEY_LOCAL_MACHINE`

Key and subkeys: `SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run`

Value:

- There are 3 value names: `RTHDVCPL`, `NvBackend`, and `BTMTrayAgent`
- Each value name has a corresponding value

`salt.states.reg.`**`absent`**(*name*, *vname=None*, *use_32bit_registry=False*)

Ensure a registry value is removed. To remove a key use key_absent.

Parameters

- **name** (*str*) – A string value representing the full path of the key to include the HIVE, Key, and all Subkeys. For example:

  `HKEY_LOCAL_MACHINE\\SOFTWARE\\Salt`

  Valid hive values include:
  - HKEY_CURRENT_USER or HKCU
  - HKEY_LOCAL_MACHINE or HKLM
  - HKEY_USERS or HKU
- **vname** (*str*) – The name of the value you'd like to create beneath the Key. If this parameter is not passed it will assume you want to set the `(Default)` value
- **use_32bit_registry** (*bool*) – Use the 32bit portion of the registry. Applies only to 64bit windows. 32bit Windows will ignore this parameter. Default is False.

Returns A dictionary showing the results of the registry operation.

Return type dict

CLI Example:

```
'HKEY_CURRENT_USER\\SOFTWARE\\Salt':
  reg.absent
    - vname: version
```

In the above example the value named `version` will be removed from the SOFTWARE\Salt key in the HKEY_CURRENT_USER hive. If `vname` was not passed, the `(Default)` value would be deleted.

`salt.states.reg.`**`key_absent`**(*name*, *use_32bit_registry=False*)

New in version 2015.5.4.

Ensure a registry key is removed. This will remove the key, subkeys, and all value entries.

Parameters

- **name** (*str*) – A string representing the full path to the key to be removed to include the hive and the keypath. The hive can be any of the following:
  - HKEY_LOCAL_MACHINE or HKLM
  - HKEY_CURRENT_USER or HKCU
  - HKEY_USER or HKU
- **use_32bit_registry** (*bool*) – Use the 32bit portion of the registry. Applies only to 64bit windows. 32bit Windows will ignore this parameter. Default is False.

Returns A dictionary showing the results of the registry operation.

Return type dict

CLI Example:

The following example will delete the `SOFTWARE\DeleteMe` key in the `HKEY_LOCAL_MACHINE` hive including all its subkeys and value pairs.

```
remove_key_demo:
  reg.key_absent:
    - name: HKEY_CURRENT_USER\SOFTWARE\DeleteMe
```

In the above example the path is interpreted as follows:

- `HKEY_CURRENT_USER` is the hive
- `SOFTWARE\DeleteMe` is the key

`salt.states.reg.`**`present`**(*name*, *vname=None*, *vdata=None*, *vtype='REG_SZ'*, *use_32bit_registry=False*, *win_owner=None*, *win_perms=None*, *win_deny_perms=None*, *win_inheritance=True*, *win_perms_reset=False*)

Ensure a registry key or value is present.

Parameters

- **`name`** (*`str`*) – A string value representing the full path of the key to include the HIVE, Key, and all Subkeys. For example:

  `HKEY_LOCAL_MACHINE\\SOFTWARE\\Salt`

  Valid hive values include:
  - HKEY_CURRENT_USER or HKCU
  - HKEY_LOCAL_MACHINE or HKLM
  - HKEY_USERS or HKU
- **`vname`** (*`str`*) – The name of the value you'd like to create beneath the Key. If this parameter is not passed it will assume you want to set the `(Default)` value
- **`vdata`** (*`str, int, list, bytes`*) – The value you'd like to set. If a value name (`vname`) is passed, this will be the data for that value name. If not, this will be the `(Default)` value for the key.

  The type of data this parameter expects is determined by the value type specified in `vtype`. The correspondence is as follows:
  - REG_BINARY: Binary data (str in Py2, bytes in Py3)
  - REG_DWORD: int
  - REG_EXPAND_SZ: str
  - REG_MULTI_SZ: list of str
  - REG_QWORD: int
  - REG_SZ: str

  Note: When setting REG_BINARY, string data will be converted to binary automatically. To pass binary data, use the built-in yaml tag `!!binary` to denote the actual binary characters. For example, the following lines will both set the same data in the registry:
  `- vdata: Salty Test`
  `- vdata: !!binary U2FsdHkgVGVzdA==\n`
  For more information about the `!!binary` tag see here

  Note: The type for the `(Default)` value is always REG_SZ and cannot be changed. This parameter is optional. If not passed, the Key will be created with no associated item/value pairs.

- **`vtype`** (*`str`*) – The value type for the data you wish to store in the registry. Valid values are:
  - REG_BINARY
  - REG_DWORD
  - REG_EXPAND_SZ
  - REG_MULTI_SZ
  - REG_QWORD
  - REG_SZ (Default)
- **`use_32bit_registry`** (*`bool`*) – Use the 32bit portion of the registry. Applies only to 64bit windows. 32bit Windows will ignore this parameter. Default is False.
- **`win_owner`** (*`str`*) – The owner of the registry key. If this is not passed, the account under which Salt is running will be used.

Note: Owner is set for the key that contains the value/data pair. You cannot set ownership on value/data pairs themselves.

New in version 2019.2.0.

- **`win_perms`** (*`dict`*) – A dictionary containing permissions to grant and their propagation. If not passed the 'Grant' permissions will not be modified.

Note: Permissions are set for the key that contains the value/data pair. You cannot set permissions on value/data pairs themselves.

For each user specify the account name, with a sub dict for the permissions to grant and the 'Applies to' setting. For example: `{'Administrators': {'perms': 'full_control', 'applies_to': 'this_key_subkeys'}}`. `perms` must be specified.

Registry permissions are specified using the `perms` key. You can specify a single basic permission or a list of advanced perms. The following are valid perms:

Basic (passed as a string):
  - full_control
  - read
  - write

Advanced (passed as a list):
  - delete
  - query_value
  - set_value
  - create_subkey
  - enum_subkeys
  - notify
  - create_link
  - read_control
  - write_dac
  - write_owner

The 'Applies to' setting is optional. It is specified using the `applies_to` key. If not specified `this_key_subkeys` is used. Valid options are:

Applies to settings:
  - this_key_only
  - this_key_subkeys
  - subkeys_only

New in version 2019.2.0.

- **`win_deny_perms`** (*`dict`*) – A dictionary containing permissions to deny and their propagation. If not passed the *Deny* permissions will not be modified.

Note: Permissions are set for the key that contains the value/data pair. You cannot set permissions on value/data pairs themselves.

Valid options are the same as those specified in `win_perms`

Note:
'Deny' permissions always take precedence over 'grant' permissions.

New in version 2019.2.0.

- **win_inheritance** (*bool*) – `True` to inherit permissions from the parent key. `False` to disable inheritance. Default is `True`.

  Note: Inheritance is set for the key that contains the value/data pair. You cannot set inheritance on value/data pairs themselves.

  New in version 2019.2.0.

- **win_perms_reset** (*bool*) – If `True` the existing DACL will be cleared and replaced with the settings defined in this function. If `False`, new entries will be appended to the existing DACL. Default is `False`

  Note: Perms are reset for the key that contains the value/data pair. You cannot set permissions on value/data pairs themselves.

  New in version 2019.2.0.

Returns A dictionary showing the results of the registry operation.

Return type dict

Example:

The following example will set the `(Default)` value for the `SOFTWARE\\Salt` key in the `HKEY_CURRENT_USER` hive to `2016.3.1`:

```
HKEY_CURRENT_USER\\SOFTWARE\\Salt:
  reg.present:
    - vdata: 2016.3.1
```

Example:

The following example will set the value for the `version` entry under the `SOFTWARE\\Salt` key in the `HKEY_CURRENT_USER` hive to `2016.3.1`. The value will be reflected in `Wow6432Node`:

```
HKEY_CURRENT_USER\\SOFTWARE\\Salt:
  reg.present:
    - vname: version
    - vdata: 2016.3.1
```

In the above example the path is interpreted as follows:

- `HKEY_CURRENT_USER` is the hive
- `SOFTWARE\\Salt` is the key
- `vname` is the value name ('version') that will be created under the key
- `vdata` is the data that will be assigned to 'version'

Example:

Binary data can be set in two ways. The following two examples will set a binary value of `Salty Test`

```
no_conversion:
  reg.present:
    - name: HKLM\SOFTWARE\SaltTesting
    - vname: test_reg_binary_state
    - vdata: Salty Test
    - vtype: REG_BINARY

conversion:
```

(continues on next page)

(continued from previous page)

```
  reg.present:
    - name: HKLM\SOFTWARE\SaltTesting
    - vname: test_reg_binary_state_with_tag
    - vdata: !!binary U2FsdHkgVGVzdA==\n
    - vtype: REG_BINARY
```

Example:

To set a `REG_MULTI_SZ` value:

```
reg_multi_sz:
  reg.present:
    - name: HKLM\SOFTWARE\Salt
    - vname: reg_multi_sz
    - vdata:
      - list item 1
      - list item 2
```

Example:

To ensure a key is present and has permissions:

```
set_key_permissions:
  reg.present:
    - name: HKLM\SOFTWARE\Salt
    - vname: version
    - vdata: 2016.3.1
    - win_owner: Administrators
    - win_perms:
        jsnuffy:
          perms: full_control
        sjones:
          perms:
            - read_control
            - enum_subkeys
            - query_value
          applies_to:
            - this_key_only
    - win_deny_perms:
        bsimpson:
          perms: full_control
          applies_to: this_key_subkeys
    - win_inheritance: True
    - win_perms_reset: True
```

21.19.244 salt.states.rsync

State to synchronize files and directories with rsync.

New in version 2016.3.0.

```
/opt/user-backups:
  rsync.synchronized:
    - source: /home
    - force: True
```

`salt.states.rsync.`**`synchronized`**(*name, source, delete=False, force=False, update=False, passwordfile=None, exclude=None, excludefrom=None, prepare=False, dryrun=False, additional_opts=None*)

Guarantees that the source directory is always copied to the target.

name Name of the target directory.

source Source directory.

prepare Create destination directory if it does not exists.

delete Delete extraneous files from the destination dirs (True or False)

force Force deletion of dirs even if not empty

update Skip files that are newer on the receiver (True or False)

passwordfile Read daemon-access password from the file (path)

exclude Exclude files, that matches pattern.

excludefrom Read exclude patterns from the file (path)

dryrun Perform a trial run with no changes made. Is the same as doing test=True

New in version 2016.3.1.

additional_opts Pass additional options to rsync, should be included as a list.

New in version 2018.3.0.

21.19.245 salt.states.rvm

Managing Ruby installations and gemsets with Ruby Version Manager (RVM)

This module is used to install and manage ruby installations and gemsets with RVM, the Ruby Version Manager. Different versions of ruby can be installed and gemsets created. RVM itself will be installed automatically if it's not present. This module will not automatically install packages that RVM depends on or ones that are needed to build ruby. If you want to run RVM as an unprivileged user (recommended) you will have to create this user yourself. This is how a state configuration could look like:

```yaml
rvm:
  group.present: []
  user.present:
    - gid: rvm
    - home: /home/rvm
    - require:
      - group: rvm

rvm-deps:
  pkg.installed:
    - pkgs:
      - bash
      - coreutils
      - gzip
      - bzip2
      - gawk
      - sed
      - curl
      - git-core
      - subversion

mri-deps:
  pkg.installed:
    - pkgs:
      - build-essential
      - openssl
```

(continues on next page)

(continued from previous page)

```
      - libreadline6
      - libreadline6-dev
      - curl
      - git-core
      - zlib1g
      - zlib1g-dev
      - libssl-dev
      - libyaml-dev
      - libsqlite3-0
      - libsqlite3-dev
      - sqlite3
      - libxml2-dev
      - libxslt1-dev
      - autoconf
      - libc6-dev
      - libncurses5-dev
      - automake
      - libtool
      - bison
      - subversion
      - ruby

jruby-deps:
  pkg.installed:
    - pkgs:
      - curl
      - g++
      - openjdk-6-jre-headless

ruby-1.9.2:
  rvm.installed:
    - default: True
    - user: rvm
    - require:
      - pkg: rvm-deps
      - pkg: mri-deps
      - user: rvm

jruby:
  rvm.installed:
    - user: rvm
    - require:
      - pkg: rvm-deps
      - pkg: jruby-deps
      - user: rvm

jgemset:
  rvm.gemset_present:
    - ruby: jruby
    - user: rvm
    - require:
      - rvm: jruby

mygemset:
  rvm.gemset_present:
    - ruby: ruby-1.9.2
```

(continues on next page)

(continued from previous page)

```
    - user: rvm
    - require:
      - rvm: ruby-1.9.2
```

`salt.states.rvm.`**`gemset_present`**(*name*, *ruby='default'*, *user=None*)
Verify that the gemset is present.

name The name of the gemset.

ruby: **default** The ruby version this gemset belongs to.

user: **None** The user to run rvm as.

New in version 0.17.0.

`salt.states.rvm.`**`installed`**(*name*, *default=False*, *user=None*, *opts=None*, *env=None*)
Verify that the specified ruby is installed with RVM. RVM is installed when necessary.

name The version of ruby to install

default [False] Whether to make this ruby the default.

user: **None** The user to run rvm as.

env: **None** A list of environment variables to set (ie, RUBY_CONFIGURE_OPTS)

opts: **None** A list of option flags to pass to RVM (ie -C, –patch)

New in version 0.17.0.

21.19.246 salt.states.salt_proxy module

Salt proxy state

New in version 2015.8.2.

State to deploy and run salt-proxy processes on a minion.

Set up pillar data for your proxies per the documentation.

Run the state as below

..code-block:: yaml

salt-proxy-configure:

salt_proxy.configure_proxy:

- proxyname: p8000
- start: True

This state will configure the salt proxy settings within /etc/salt/proxy (if /etc/salt/proxy doesn't exists) and start the salt-proxy process (default true), if it isn't already running.

`salt.states.salt_proxy.`**`configure_proxy`**(*name*, *proxyname='p8000'*, *start=True*)
Create the salt proxy file and start the proxy process if required

Parameters

- **`name`** – The name of this state
- **`proxyname`** – Name to be used for this proxy (should match entries in pillar)
- **`start`** – Boolean indicating if the process should be started

Example:

..code-block:: yaml

salt-proxy-configure:

salt_proxy.configure_proxy:

- proxyname: p8000
- start: True

21.19.247 salt.states.saltmod

Control the Salt command interface

This state is intended for use from the Salt Master. It provides access to sending commands down to minions as well as access to executing master-side modules. These state functions wrap Salt's *Python API*.

> Support for masterless minions was added to the `salt.state` function, so they can run orchestration sls files. This is particularly useful when the rendering of a state is dependent on the execution of another state. Orchestration will render and execute each orchestration block independently, while honoring requisites to ensure the states are applied in the correct order.

See also:

More Orchestrate documentation

- *Full Orchestrate Tutorial*
- *`The Orchestrate runner`*

`salt.states.saltmod.`**`function`**(*name, tgt, ssh=False, tgt_type='glob', ret='', ret_config=None, ret_kwargs=None, expect_minions=False, fail_minions=None, fail_function=None, arg=None, kwarg=None, timeout=None, batch=None, subset=None, failhard=None, **kwargs*)

Execute a single module function on a remote minion via salt or salt-ssh

name The name of the function to run, aka cmd.run or pkg.install

tgt The target specification, aka '*' for all minions

tgt_type The target type, defaults to `glob`

arg The list of arguments to pass into the function

kwarg The dict (not a list) of keyword arguments to pass into the function

ret Optionally set a single or a list of returners to use

ret_config Use an alternative returner configuration

ret_kwargs Override individual returner configuration items

expect_minions An optional boolean for failing if some minions do not respond

fail_minions An optional list of targeted minions where failure is an option

fail_function An optional string that points to a salt module that returns True or False based on the returned data dict for individual minions

ssh Set to *True* to use the ssh client instead of the standard salt client

batch Execute the command *in batches*. E.g.: `10%`.

subset Number of minions from the targeted set to randomly use

New in version 2017.7.0.

failhard pass failhard down to the executing state

New in version 2019.2.2.

`salt.states.saltmod.`**`parallel_runners`**(*name, runners, **kwargs*)

Executes multiple runner modules on the master in parallel.

New in version 2017.x.0: (Nitrogen)

A separate thread is spawned for each runner. This state is intended to be used with the orchestrate runner in place of the `saltmod.runner` state when different tasks should be run in parallel. In general, Salt states are not safe when used concurrently, so ensure that they are used in a safe way (e.g. by only targeting separate minions in parallel tasks).

name: name identifying this state. The name is provided as part of the output, but not used for anything else.

runners: list of runners that should be run in parallel. Each element of the list has to be a dictionary. This dictionary's name entry stores the name of the runner function that shall be invoked. The optional kwarg entry stores a dictionary of named arguments that are passed to the runner function.

```
parallel-state:
   salt.parallel_runners:
     - runners:
         my_runner_1:
           - name: state.orchestrate
           - kwarg:
               mods: orchestrate_state_1
         my_runner_2:
           - name: state.orchestrate
           - kwarg:
               mods: orchestrate_state_2
```

`salt.states.saltmod.`**`runner`**`(`*name*, ***kwargs*`)`

Execute a runner module on the master

New in version 2014.7.0.

name The name of the function to run

kwargs Any keyword arguments to pass to the runner function

```
run-manage-up:
 salt.runner:
   - name: manage.up
```

`salt.states.saltmod.`**`state`**`(`*name, tgt, ssh=False, tgt_type='glob', ret='', ret_config=None, ret_kwargs=None, highstate=None, sls=None, top=None, saltenv=None, test=None, pillar=None, pillarenv=None, expect_minions=True, fail_minions=None, allow_fail=0, concurrent=False, timeout=None, batch=None, queue=False, subset=None, orchestration_jid=None, failhard=None, **kwargs*`)`

Invoke a state run on a given target

name An arbitrary name used to track the state execution

tgt The target specification for the state run.

Masterless support: When running on a masterless minion, the `tgt` is ignored and will always be the local minion.

tgt_type The target type to resolve, defaults to `glob`

ret Optionally set a single or a list of returners to use

ret_config Use an alternative returner configuration

ret_kwargs Override individual returner configuration items

highstate Defaults to None, if set to True the target systems will ignore any sls references specified in the sls option and call state.highstate on the targeted minions

top Should be the name of a top file. If set state.top is called with this top file instead of state.sls.

sls A group of sls files to execute. This can be defined as a single string containing a single sls file, or a list of sls files

test Pass `test=true` or `test=false` through to the state function. This can be used to overide a test mode set in the minion's config file. If left as the default of None and the 'test' mode is supplied on the command line, that value is passed instead.

pillar Pass the `pillar` kwarg through to the state function

pillarenv The pillar environment to grab pillars from

New in version 2017.7.0.

saltenv The default salt environment to pull sls files from

ssh Set to *True* to use the ssh client instead of the standard salt client

roster In the event of using salt-ssh, a roster system can be set

expect_minions An optional boolean for failing if some minions do not respond

fail_minions An optional list of targeted minions where failure is an option

allow_fail Pass in the number of minions to allow for failure before setting the result of the execution to False

concurrent Allow multiple state runs to occur at once.

> WARNING: This flag is potentially dangerous. It is designed for use when multiple state runs can safely be run at the same Do not use this flag for performance optimization.

queue Pass `queue=true` through to the state function

batch Execute the command *in batches*. E.g.: `10%`.

> New in version 2016.3.0.

subset Number of minions from the targeted set to randomly use

> New in version 2017.7.0.

failhard pass failhard down to the executing state

> New in version 2019.2.2.

Examples:

Run a list of sls files via `state.sls` on target minions:

```
webservers:
  salt.state:
    - tgt: 'web*'
    - sls:
      - apache
      - django
      - core
    - saltenv: prod
```

Run a full `state.highstate` on target mininons.

```
databases:
  salt.state:
    - tgt: role:database
    - tgt_type: grain
    - highstate: True
```

`salt.states.saltmod.`**`wait_for_event`**(*name, id_list, event_id='id', timeout=300, node='master'*)

Watch Salt's event bus and block until a condition is met

New in version 2014.7.0.

name An event tag to watch for; supports Reactor-style globbing.

id_list A list of event identifiers to watch for – usually the minion ID. Each time an event tag is matched the event data is inspected for `event_id`, if found it is removed from `id_list`. When `id_list` is empty this function returns success.

event_id [id] The name of a key in the event data. Default is `id` for the minion ID, another common value is `name` for use with orchestrating salt-cloud events.

timeout [300] The maximum time in seconds to wait before failing.

The following example blocks until all the listed minions complete a restart and reconnect to the Salt master:

```
reboot_all_minions:
  salt.function:
    - name: system.reboot
    - tgt: '*'

wait_for_reboots:
  salt.wait_for_event:
    - name: salt/minion/*/start
    - id_list:
      - jerry
```

(continues on next page)

(continued from previous page)

```
      - stuart
      - dave
      - phil
      - kevin
      - mike
    - require:
      - salt: reboot_all_minions
```

`salt.states.saltmod.`**`wheel`**`(`*name*, ***kwargs*`)`

Execute a wheel module on the master

New in version 2014.7.0.

name The name of the function to run

kwargs Any keyword arguments to pass to the wheel function

```
accept_minion_key:
  salt.wheel:
    - name: key.accept
    - match: frank
```

21.19.248 salt.states.saltutil

Saltutil State

This state wraps the saltutil execution modules to make them easier to run from a states. Rather than needing to to use `module.run` this state allows for improved change detection.

`salt.states.saltutil.`**`sync_all`**`(`*name*, ***kwargs*`)`

Performs the same task as saltutil.sync_all module See *`saltutil module for full list of options`*

```
sync_everything:
  saltutil.sync_all:
    - refresh: True
```

`salt.states.saltutil.`**`sync_beacons`**`(`*name*, ***kwargs*`)`

Performs the same task as saltutil.sync_beacons module See *`saltutil module for full list of options`*

```
sync_everything:
  saltutil.sync_beacons:
    - refresh: True
```

`salt.states.saltutil.`**`sync_clouds`**`(`*name*, ***kwargs*`)`

Performs the same task as saltutil.sync_clouds module See *`saltutil module for full list of options`*

```
sync_everything:
  saltutil.sync_clouds:
    - refresh: True
```

salt.states.saltutil.**sync_engines**(*name*, ***kwargs*)

Performs the same task as saltutil.sync_engines module See *saltutil module for full list of options*

```
sync_everything:
  saltutil.sync_engines:
    - refresh: True
```

salt.states.saltutil.**sync_executors**(*name*, ***kwargs*)

Performs the same task as saltutil.sync_executors module See *saltutil module for full list of options*

```
sync_everything:
  saltutil.sync_executors:
    - refresh: True
```

salt.states.saltutil.**sync_grains**(*name*, ***kwargs*)

Performs the same task as saltutil.sync_grains module See *saltutil module for full list of options*

```
sync_everything:
  saltutil.sync_grains:
    - refresh: True
```

salt.states.saltutil.**sync_log_handlers**(*name*, ***kwargs*)

Performs the same task as saltutil.sync_log_handlers module See *saltutil module for full list of options*

```
sync_everything:
  saltutil.sync_log_handlers:
    - refresh: True
```

salt.states.saltutil.**sync_matchers**(*name*, ***kwargs*)

Performs the same task as saltutil.sync_matchers module See *saltutil module for full list of options*

```
sync_everything:
  saltutil.sync_matchers:
    - refresh: True
```

salt.states.saltutil.**sync_modules**(*name*, ***kwargs*)

Performs the same task as saltutil.sync_modules module See *saltutil module for full list of options*

```
sync_everything:
  saltutil.sync_modules:
    - refresh: True
```

salt.states.saltutil.**sync_output**(*name*, ***kwargs*)

Performs the same task as saltutil.sync_output module See *saltutil module for full list of options*

```
sync_everything:
  saltutil.sync_output:
    - refresh: True
```

salt.states.saltutil.**sync_outputters**(*name*, ***kwargs*)

Performs the same task as saltutil.sync_outputters module See *saltutil module for full list of options*

```
sync_everything:
  saltutil.sync_outputters:
    - refresh: True
```

salt.states.saltutil.**sync_pillar**(*name*, ***kwargs*)

Performs the same task as saltutil.sync_pillar module See *saltutil module for full list of options*

```
sync_everything:
  saltutil.sync_pillar:
    - refresh: True
```

salt.states.saltutil.**sync_proxymodules**(*name*, ***kwargs*)

Performs the same task as saltutil.sync_proxymodules module See *saltutil module for full list of options*

```
sync_everything:
  saltutil.sync_proxymodules:
    - refresh: True
```

salt.states.saltutil.**sync_renderers**(*name*, ***kwargs*)

Performs the same task as saltutil.sync_renderers module See *saltutil module for full list of options*

```
sync_everything:
  saltutil.sync_renderers:
    - refresh: True
```

salt.states.saltutil.**sync_returners**(*name*, ***kwargs*)

Performs the same task as saltutil.sync_returners module See *saltutil module for full list of options*

```
sync_everything:
  saltutil.sync_returners:
    - refresh: True
```

salt.states.saltutil.**sync_sdb**(*name*, ***kwargs*)

Performs the same task as saltutil.sync_sdb module See *saltutil module for full list of options*

```
sync_everything:
  saltutil.sync_sdb:
    - refresh: True
```

salt.states.saltutil.**sync_serializers**(*name*, ***kwargs*)

Performs the same task as saltutil.sync_serializers module See *saltutil module for full list of options*

```
sync_everything:
  saltutil.sync_serializers:
    - refresh: True
```

`salt.states.saltutil.`**`sync_states`**(*name*, ***kwargs*)

Performs the same task as saltutil.sync_states module See *saltutil module for full list of options*

```
sync_everything:
  saltutil.sync_states:
    - refresh: True
```

`salt.states.saltutil.`**`sync_thorium`**(*name*, ***kwargs*)

Performs the same task as saltutil.sync_thorium module See *saltutil module for full list of options*

```
sync_everything:
  saltutil.sync_thorium:
    - refresh: True
```

`salt.states.saltutil.`**`sync_utils`**(*name*, ***kwargs*)

Performs the same task as saltutil.sync_utils module See *saltutil module for full list of options*

```
sync_everything:
  saltutil.sync_utils:
    - refresh: True
```

21.19.249 salt.states.schedule

Management of the Salt scheduler

```
job3:
  schedule.present:
    - function: test.ping
    - seconds: 3600
    - splay: 10

This will schedule the command: test.ping every 3600 seconds
(every hour) splaying the time between 0 and 10 seconds

job2:
  schedule.present:
    - function: test.ping
    - seconds: 15
    - splay:
        start: 10
        end: 20

This will schedule the command: test.ping every 15 seconds
splaying the time between 10 and 20 seconds

job1:
  schedule.present:
    - function: state.sls
    - job_args:
      - httpd
    - job_kwargs:
        test: True
```

(continues on next page)

(continued from previous page)

```
    - when:
        - Monday 5:00pm
        - Tuesday 3:00pm
        - Wednesday 5:00pm
        - Thursday 3:00pm
        - Friday 5:00pm

This will schedule the command: state.sls httpd test=True at 5pm on Monday,
Wednesday and Friday, and 3pm on Tuesday and Thursday.  Requires that
python-dateutil is installed on the minion.

job1:
  schedule.present:
    - function: state.sls
    - job_args:
      - httpd
    - job_kwargs:
        test: True
    - cron: '*/5 * * * *'

Scheduled jobs can also be specified using the format used by cron.  This will
schedule the command: state.sls httpd test=True to run every 5 minutes.  Requires
that python-croniter is installed on the minion.

job1:
  schedule.present:
    - function: state.sls
    - job_args:
      - httpd
    - job_kwargs:
        test: True
    - when:
        - Monday 5:00pm
        - Tuesday 3:00pm
        - Wednesday 5:00pm
        - Thursday 3:00pm
        - Friday 5:00pm
    - returner: xmpp
    - return_config: xmpp_state_run
    - return_kwargs:
        recipient: user@domain.com

This will schedule the command: state.sls httpd test=True at 5pm on Monday,
Wednesday and Friday, and 3pm on Tuesday and Thursday.  Using the xmpp returner
to return the results of the scheduled job, with the alternative configuration
options found in the xmpp_state_run section.

job1:
  schedule.present:
    - function: state.sls
    - job_args:
      - httpd
    - job_kwargs:
        test: True
    - hours: 1
    - skip_during_range:
```

(continues on next page)

(continued from previous page)

```
        - start: 2pm
        - end: 3pm
    - run_after_skip_range: True

This will schedule the command: state.sls httpd test=True at 5pm on Monday,
Wednesday and Friday, and 3pm on Tuesday and Thursday.  Requires that
python-dateutil is installed on the minion.
```

`salt.states.schedule.`**`absent`**`(`*name*, ***kwargs*`)`
Ensure a job is absent from the schedule

name The unique name that is given to the scheduled job.
persist Whether the job should persist between minion restarts, defaults to True.

`salt.states.schedule.`**`disabled`**`(`*name*, ***kwargs*`)`
Ensure a job is disabled in the schedule

name The unique name that is given to the scheduled job.
persist Whether the job should persist between minion restarts, defaults to True.

`salt.states.schedule.`**`enabled`**`(`*name*, ***kwargs*`)`
Ensure a job is enabled in the schedule

name The unique name that is given to the scheduled job.
persist Whether the job should persist between minion restarts, defaults to True.

`salt.states.schedule.`**`present`**`(`*name*, ***kwargs*`)`
Ensure a job is present in the schedule

name The unique name that is given to the scheduled job.
seconds The scheduled job will be executed after the specified number of seconds have passed.
minutes The scheduled job will be executed after the specified number of minutes have passed.
hours The scheduled job will be executed after the specified number of hours have passed.
days The scheduled job will be executed after the specified number of days have passed.
when This will schedule the job at the specified time(s). The when parameter must be a single value or a dictionary with the date string(s) using the dateutil format. Requires python-dateutil.
cron This will schedule the job at the specified time(s) using the crontab format. Requires python-croniter.
run_on_start Whether the job will run when Salt minion starts, or the job will be skipped **once** and run at the next scheduled run. Value should be a boolean.
function The function that should be executed by the scheduled job.
job_args The arguments that will be used by the scheduled job.
job_kwargs The keyword arguments that will be used by the scheduled job.
maxrunning Ensure that there are no more than N copies of a particular job running.
jid_include Include the job into the job cache.
splay The amount of time in seconds to splay a scheduled job. Can be specified as a single value in seconds or as a dictionary range with 'start' and 'end' values.
range This will schedule the command within the range specified. The range parameter must be a dictionary with the date strings using the dateutil format. Requires python-dateutil.
once This will schedule a job to run once on the specified date.
once_fmt The default date format is ISO 8601 but can be overridden by also specifying the `once_fmt` option.
enabled Whether the job should be enabled or disabled. Value should be a boolean.
return_job Whether to return information to the Salt master upon job completion.
metadata Using the metadata parameter special values can be associated with a scheduled job. These values are not used in the execution of the job, but can be used to search for specific jobs later if combined with the return_job parameter. The metadata parameter must be specified as a dictionary, othewise it will be ignored.
returner The returner to use to return the results of the scheduled job.
return_config The alternative configuration to use for returner configuration options.
return_kwargs Any individual returner configuration items to override. Should be passed as a dictionary.

persist Whether the job should persist between minion restarts, defaults to True.

skip_during_range This will ensure that the scheduled command does not run within the range specified. The range parameter must be a dictionary with the date strings using the dateutil format. Requires python-dateutil.

run_after_skip_range Whether the job should run immediately after the skip_during_range time period ends.

21.19.250 salt.states.selinux

Management of SELinux rules

If SELinux is available for the running system, the mode can be managed and booleans can be set.

```
enforcing:
    selinux.mode

samba_create_home_dirs:
    selinux.boolean:
      - value: True
      - persist: True

nginx:
    selinux.module:
      - enabled: False
```

Note: Use of these states require that the `selinux` execution module is available.

`salt.states.selinux.`**`boolean`**`(`*name, value, persist=False*`)`

Set up an SELinux boolean

name The name of the boolean to set

value The value to set on the boolean

persist Defaults to False, set persist to true to make the boolean apply on a reboot

`salt.states.selinux.`**`fcontext_policy_absent`**`(`*name, filetype='a', sel_type=None, sel_user=None, sel_level=None*`)`

New in version 2017.7.0.

Makes sure an SELinux file context policy for a given filespec (name), filetype and SELinux context type is absent.

name filespec of the file or directory. Regex syntax is allowed.

filetype The SELinux filetype specification. Use one of [a, f, d, c, b, s, l, p]. See also *man semanage-fcontext*. Defaults to 'a' (all files).

sel_type The SELinux context type. There are many.

sel_user The SELinux user.

sel_level The SELinux MLS range.

`salt.states.selinux.`**`fcontext_policy_applied`**`(`*name, recursive=False*`)`

New in version 2017.7.0.

Checks and makes sure the SELinux policies for a given filespec are applied.

`salt.states.selinux.`**`fcontext_policy_present`**`(`*name, sel_type, filetype='a', sel_user=None, sel_level=None*`)`

New in version 2017.7.0.

Makes sure a SELinux policy for a given filespec (name), filetype and SELinux context type is present.

name filespec of the file or directory. Regex syntax is allowed.

sel_type SELinux context type. There are many.

filetype The SELinux filetype specification. Use one of [a, f, d, c, b, s, l, p]. See also *man semanage-fcontext*. Defaults to 'a' (all files).

sel_user The SELinux user.

sel_level The SELinux MLS range.

`salt.states.selinux.`**`mode`**`(`*name*`)`

Verifies the mode SELinux is running in, can be set to enforcing, permissive, or disabled

Note: A change to or from disabled mode requires a system reboot. You will need to perform this yourself.

name The mode to run SELinux in, permissive, enforcing, or disabled.

`salt.states.selinux.`**`module`**`(`*name*, *module_state='Enabled'*, *version='any'*, ***opts*`)`

Enable/Disable and optionally force a specific version for an SELinux module

name The name of the module to control

module_state Should the module be enabled or disabled?

version Defaults to no preference, set to a specified value if required. Currently can only alert if the version is incorrect.

install Setting to True installs module

source Points to module source file, used only when install is True

remove Setting to True removes module

New in version 2016.3.0.

`salt.states.selinux.`**`module_install`**`(`*name*`)`

Installs custom SELinux module from given file

name Path to file with module to install

New in version 2016.11.6.

`salt.states.selinux.`**`module_remove`**`(`*name*`)`

Removes SELinux module

name The name of the module to remove

New in version 2016.11.6.

`salt.states.selinux.`**`port_policy_absent`**`(`*name*, *sel_type=None*, *protocol=None*, *port=None*`)`

New in version 2019.2.0.

Makes sure an SELinux port policy for a given port, protocol and SELinux context type is absent.

name The protocol and port spec. Can be formatted as `(tcp|udp)/(port|port-range)`.

sel_type The SELinux Type. Optional; can be used in determining if policy is present, ignored by `semanage port --delete`.

protocol The protocol for the port, `tcp` or `udp`. Required if name is not formatted.

port The port or port range. Required if name is not formatted.

`salt.states.selinux.`**`port_policy_present`**`(`*name*, *sel_type*, *protocol=None*, *port=None*, *sel_range=None*`)`

New in version 2019.2.0.

Makes sure an SELinux port policy for a given port, protocol and SELinux context type is present.

name The protocol and port spec. Can be formatted as `(tcp|udp)/(port|port-range)`.

sel_type The SELinux Type.

protocol The protocol for the port, `tcp` or `udp`. Required if name is not formatted.

port The port or port range. Required if name is not formatted.

sel_range The SELinux MLS/MCS Security Range.

21.19.251 salt.states.serverdensity_device

Monitor Server with Server Density

New in version 2014.7.0.

Server Density Is a hosted monitoring service.

Warning: This state module is beta. It might be changed later to include more or less automation.

Note: This state module requires a pillar for authentication with Server Density To install a v1 agent:

```
serverdensity:
  api_token: "b97da80a41c4f61bff05975ee51eb1aa"
  account_url: "https://your-account.serverdensity.io"
```

To install a v2 agent:

```
serverdensity:
  api_token: "b97da80a41c4f61bff05975ee51eb1aa"
  account_name: "your-account"
```

Note: Although Server Density allows duplicate device names in its database, this module will raise an exception if you try monitoring devices with the same name.

Example:

```
'server_name':
  serverdensity_device.monitored
```

`salt.states.serverdensity_device.`**`monitored`**(*name, group=None, salt_name=True, salt_params=True, agent_version=1, **params*)

Device is monitored with Server Density.

name Device name in Server Density.

salt_name If `True` (default), takes the name from the `id` grain. If `False`, the provided name is used.

group Group name under with device will appear in Server Density dashboard. Default - *None*.

agent_version The agent version you want to use. Valid values are 1 or 2. Default - 1.

salt_params If `True` (default), needed config parameters will be sourced from grains and from *`status.all_status`*.

params Add parameters that you want to appear in the Server Density dashboard. Will overwrite the *salt_params* parameters. For more info, see the API docs.

Usage example:

```
'server_name':
  serverdensity_device.monitored
```

```
'server_name':
  serverdensity_device.monitored:
    - group: web-servers
```

```
'my_special_server':
  serverdensity_device.monitored:
    - salt_name: False
    - group: web-servers
    - cpuCores: 2
    - os: '{"code": "linux", "name": "Linux"}'
```

21.19.252 salt.states.service

Starting or restarting of services and daemons

Services are defined as system daemons and are typically launched using system init or rc scripts. This service state uses whichever service module is loaded on the minion with the virtualname of `service`. Services can be defined as either running or dead.

If you need to know if your init system is supported, see the list of supported `service modules` for your desired init system (systemd, sysvinit, launchctl, etc.).

Note that Salt's service execution module, and therefore this service state, uses OS grains to ascertain which service module should be loaded and used to execute service functions. As existing distributions change init systems or new distributions are created, OS detection can sometimes be incomplete. If your service states are running into trouble with init system detection, please see the *Overriding Virtual Module Providers* section of Salt's module documentation to work around possible errors.

Note: The current status of a service is determined by the return code of the init/rc script status command. A status return code of 0 it is considered running. Any other return code is considered dead.

```
httpd:
  service.running: []
```

The service can also be set to start at runtime via the enable option:

```
openvpn:
  service.running:
    - enable: True
```

By default if a service is triggered to refresh due to a watch statement the service is restarted. If the desired behavior is to reload the service, then set the reload value to True:

```
redis:
  service.running:
    - enable: True
    - reload: True
    - watch:
      - pkg: redis
```

Note: More details regarding `watch` can be found in the *Requisites* documentation.

`salt.states.service.`**`dead`**(*name, enable=None, sig=None, init_delay=None, **kwargs*)

Ensure that the named service is dead by stopping the service if it is running

name The name of the init or rc script used to manage the service

enable Set the service to be enabled at boot time, `True` sets the service to be enabled, `False` sets the named service to be disabled. The default is `None`, which does not enable or disable anything.

sig The string to search for when looking for the service process with ps

init_delay Add a sleep command (in seconds) before the check to make sure service is killed.

New in version 2017.7.0.

no_block [False] **For systemd minions only**. Stops the service using `--no-block`.

New in version 2017.7.0.

`salt.states.service.`**`disabled`**`(`*name*, ***kwargs*`)`

Ensure that the service is disabled on boot, only use this state if you don't want to manage the running process, remember that if you want to disable a service to use the enable: False option for the running or dead function.

name The name of the init or rc script used to manage the service

`salt.states.service.`**`enabled`**`(`*name*, ***kwargs*`)`

Ensure that the service is enabled on boot, only use this state if you don't want to manage the running process, remember that if you want to enable a running service to use the enable: True option for the running or dead function.

name The name of the init or rc script used to manage the service

`salt.states.service.`**`masked`**`(`*name*, *runtime=False*`)`

New in version 2017.7.0.

Note: This state is only available on minions which use systemd.

Ensures that the named service is masked (i.e. prevented from being started).

name Name of the service to mask

runtime [False] By default, this state will manage an indefinite mask for the named service. Set this argument to `True` to runtime mask the service.

Note: It is possible for a service to have both indefinite and runtime masks set for it. Therefore, this state will manage a runtime or indefinite mask independently of each other. This means that if the service is already indefinitely masked, running this state with `runtime` set to `True` will _not_ remove the indefinite mask before setting a runtime mask. In these cases, if it is desirable to ensure that the service is runtime masked and not indefinitely masked, pair this state with a *`service.unmasked`* state, like so:

```
mask_runtime_foo:
  service.masked:
    - name: foo
    - runtime: True

unmask_indefinite_foo:
  service.unmasked:
    - name: foo
    - runtime: False
```

`salt.states.service.`**`mod_watch`**`(`*name*, *sfun=None*, *sig=None*, *reload=False*, *full_restart=False*, *init_delay=None*, *force=False*, ***kwargs*`)`

The service watcher, called to invoke the watch command. When called, it will restart or reload the named service.

Note: This state exists to support special handling of the `watch` *requisite*. It should not be called directly.

Parameters for this function should be set by the watching service. (i.e. `service.running`)

name The name of the init or rc script used to manage the service

sfun The original function which triggered the mod_watch call (*service.running*, for example).

sig The string to search for when looking for the service process with ps

reload When set, reload the service instead of restarting it. (i.e. `service nginx reload`)

full_restart Perform a full stop/start of a service by passing `--full-restart`. This option is ignored if `reload` is set and is supported by only a few `service modules`.

force Use service.force_reload instead of reload (needs reload to be set to True)

init_delay Add a sleep command (in seconds) before the service is restarted/reloaded

`salt.states.service.`**`running`**`(`*name*, *enable=None*, *sig=None*, *init_delay=None*, ***kwargs*`)`

Ensure that the service is running

name The name of the init or rc script used to manage the service

enable Set the service to be enabled at boot time, `True` sets the service to be enabled, `False` sets the named service to be disabled. The default is `None`, which does not enable or disable anything.

sig The string to search for when looking for the service process with ps

init_delay Some services may not be truly available for a short period after their startup script indicates to the system that they are. Provide an 'init_delay' to specify that this state should wait an additional given number of seconds after a service has started before returning. Useful for requisite states wherein a dependent state might assume a service has started but is not yet fully initialized.

no_block [False] **For systemd minions only**. Starts the service using `--no-block`.

New in version 2017.7.0.

unmask [False] **For systemd minions only**. Set to `True` to remove an indefinite mask before attempting to start the service.

New in version 2017.7.0: In previous releases, Salt would simply unmask a service before making any changes. This behavior is no longer the default.

unmask_runtime [False] **For systemd minions only**. Set to `True` to remove a runtime mask before attempting to start the service.

New in version 2017.7.0: In previous releases, Salt would simply unmask a service before making any changes. This behavior is no longer the default.

wait [3] **For systemd minions only**. Passed through when using *`service.status`* to determine whether the service is running or not.

New in version 2019.2.3.

Note:

`watch` can be used with service.running to restart a service when another state changes (example: a file.managed state that creates the service's config file). More details regarding `watch` can be found in the *Requisites* documentation.

`salt.states.service.`**`unmasked`**`(`*name*, *runtime=False*`)`

New in version 2017.7.0.

Note: This state is only available on minions which use systemd.

Ensures that the named service is unmasked

name Name of the service to unmask

runtime [False] By default, this state will manage an indefinite mask for the named service. Set this argument to `True` to ensure that the service is runtime masked.

Note: It is possible for a service to have both indefinite and runtime masks set for it. Therefore, this state will manage a runtime or indefinite mask independently of each other. This means that if the service is indefinitely masked, running this state with `runtime` set to `True` will _not_ remove the indefinite mask.

21.19.253 salt.states.slack

Send a message to Slack

This state is useful for sending messages to Slack during state runs.

New in version 2015.5.0.

```
slack-message:
  slack.post_message:
    - channel: '#general'
    - from_name: SuperAdmin
    - message: 'This state was executed successfully.'
    - api_key: peWcBiMOS9HrZG15peWcBiMOS9HrZG15
```

The api key can be specified in the master or minion configuration like below:

```
slack:
  api_key: peWcBiMOS9HrZG15peWcBiMOS9HrZG15
```

`salt.states.slack.`**`post_message`**`(`*name*, ***kwargs*`)`

Send a message to a Slack channel.

```
slack-message:
  slack.post_message:
    - channel: '#general'
    - from_name: SuperAdmin
    - message: 'This state was executed successfully.'
    - api_key: peWcBiMOS9HrZG15peWcBiMOS9HrZG15
```

The following parameters are required:

api_key parameters:

name The unique name for this event.

channel The channel to send the message to. Can either be the ID or the name.

from_name The name of that is to be shown in the "from" field.

message The message that is to be sent to the Slack channel.

The following parameters are optional:

api_key The api key for Slack to use for authentication, if not specified in the configuration options of master or minion.

icon URL to an image to use as the icon for this message

webhook parameters:

name The unique name for this event.

message The message that is to be sent to the Slack channel.

color The color of border of left side

short An optional flag indicating whether the value is short enough to be displayed side-by-side with other values.

identifier The identifier of WebHook.

channel The channel to use instead of the WebHook default.

username Username to use instead of WebHook default.

icon_emoji Icon to use instead of WebHook default.

21.19.254 salt.states.smartos

Management of SmartOS Standalone Compute Nodes

maintainer Jorge Schrauwen <sjorge@blackdot.be>

maturity new

depends vmadm, imgadm

platform smartos

New in version 2016.3.0.

```
vmtest.example.org:
  smartos.vm_present:
    - config:
        reprovision: true
    - vmconfig:
        image_uuid: c02a2044-c1bd-11e4-bd8c-dfc1db8b0182
        brand: joyent
        alias: vmtest
        quota: 5
        max_physical_memory: 512
        tags:
          label: 'test vm'
          owner: 'sjorge'
        nics:
          "82:1b:8e:49:e9:12":
            nic_tag: trunk
            mtu: 1500
            ips:
              - 172.16.1.123/16
              - 192.168.2.123/24
            vlan_id: 10
          "82:1b:8e:49:e9:13":
            nic_tag: trunk
            mtu: 1500
            ips:
              - dhcp
            vlan_id: 30
        filesystems:
          "/bigdata":
            source: "/bulk/data"
            type: lofs
            options:
              - ro
              - nodevices

kvmtest.example.org:
  smartos.vm_present:
    - vmconfig:
        brand: kvm
        alias: kvmtest
        cpu_type: host
        ram: 512
        vnc_port: 9
        tags:
          label: 'test kvm'
          owner: 'sjorge'
```

(continues on next page)

(continued from previous page)

```
        disks:
          disk0
            size: 2048
            model: virtio
            compression: lz4
            boot: true
        nics:
          "82:1b:8e:49:e9:15":
            nic_tag: trunk
            mtu: 1500
            ips:
              - dhcp
            vlan_id: 30

docker.example.org:
  smartos.vm_present:
    - config:
        auto_import: true
        reprovision: true
    - vmconfig:
        image_uuid: emby/embyserver:latest
        brand: lx
        alias: mydockervm
        quota: 5
        max_physical_memory: 1024
        tags:
          label: 'my emby docker'
          owner: 'sjorge'
        resolvers:
          - 172.16.1.1
        nics:
          "82:1b:8e:49:e9:18":
            nic_tag: trunk
            mtu: 1500
            ips:
              - 172.16.1.118/24
            vlan_id: 10
        filesystems:
          "/config:
            source: "/vmdata/emby_config"
            type: lofs
            options:
              - nodevices

cleanup_images:
  smartos.image_vacuum
```

Note: Keep in mind that when removing properties from vmconfig they will not get removed from the vm's current configuration, except for nics, disk, tags, … they get removed via add_*, set_*, update_*, and remove_*. Properties must be manually reset to their default value. The same behavior as when using 'vmadm update'.

salt.states.smartos.**config_absent**(*name*)

Ensure configuration property is absent in /usbkey/config

name [string] name of property

`salt.states.smartos.`**`config_present`**`(`*name*, *value*`)`

Ensure configuration property is set to value in /usbkey/config

name [string] name of property

value [string] value of property

`salt.states.smartos.`**`image_absent`**`(`*name*`)`

Ensure image is absent on the computenode

name [string] uuid of image

Note: computenode.image_absent will only remove the image if it is not used by a vm.

`salt.states.smartos.`**`image_present`**`(`*name*`)`

Ensure image is present on the computenode

name [string] uuid of image

`salt.states.smartos.`**`image_vacuum`**`(`*name*`)`

Delete images not in use or installed via image_present

Warning: Only image_present states that are included via the top file will be detected.

`salt.states.smartos.`**`source_absent`**`(`*name*`)`

Ensure an image source is absent on the computenode

name [string] source url

`salt.states.smartos.`**`source_present`**`(`*name*, *source_type='imgapi'*`)`

Ensure an image source is present on the computenode

name [string] source url

source_type [string] source type (imgapi or docker)

`salt.states.smartos.`**`vm_absent`**`(`*name*, *archive=False*`)`

Ensure vm is absent on the computenode

name [string] hostname of vm

archive [boolean] toggle archiving of vm on removal

Note: State ID is used as hostname. Hostnames must be unique.

`salt.states.smartos.`**`vm_present`**`(`*name*, *vmconfig*, *config=None*`)`

Ensure vm is present on the computenode

name [string] hostname of vm

vmconfig [dict] options to set for the vm

config [dict] fine grain control over vm_present

Note:

The following configuration properties can be toggled in the config parameter.

- kvm_reboot (true) - reboots of kvm zones if needed for a config update
- auto_import (false) - automatic importing of missing images
- auto_lx_vars (true) - copy kernel_version and docker:* variables from image
- reprovision (false) - reprovision on image_uuid changes
- enforce_tags (true) - false = add tags only, true = add, update, and remove tags
- enforce_routes (true) - false = add tags only, true = add, update, and remove routes
- enforce_internal_metadata (true) - false = add metadata only, true = add, update, and remove metadata

  - enforce_customer_metadata (true) - false = add metadata only, true = add, update, and remove metadata

---

Note: State ID is used as hostname. Hostnames must be unique.

Note: If hostname is provided in vmconfig this will take president over the State ID. This allows multiple states to be applied to the same vm.

Note:

The following instances should have a unique ID.

  - nic : mac
  - filesystem: target
  - disk : path or diskN for zvols

e.g. disk0 will be the first disk added, disk1 the 2nd,...

Changed in version 2019.2.0: Added support for docker image uuids, added auto_lx_vars configuration, documented some missing configuration options.

`salt.states.smartos.`**`vm_running`**`(`*name*`)`

Ensure vm is in the running state on the computenode

name [string] hostname of vm

Note: State ID is used as hostname. Hostnames must be unique.

`salt.states.smartos.`**`vm_stopped`**`(`*name*`)`

Ensure vm is in the stopped state on the computenode

name [string] hostname of vm

Note: State ID is used as hostname. Hostnames must be unique.

21.19.255 salt.states.smtp

Sending Messages via SMTP

New in version 2014.7.0.

This state is useful for firing messages during state runs, using the SMTP protocol

```
server-warning-message:
  smtp.send_msg:
    - name: 'This is a server warning message'
    - profile: my-smtp-account
    - recipient: admins@example.com
```

`salt.states.smtp.`**`send_msg`**`(`*name, recipient, subject, sender=None, profile=None, use_ssl='True', attachments=None*`)`

Send a message via SMTP

```
server-warning-message:
  smtp.send_msg:
    - name: 'This is a server warning message'
    - profile: my-smtp-account
    - subject: 'Message from Salt'
    - recipient: admin@example.com
    - sender: admin@example.com
    - use_ssl: True
    - attachments:
        - /var/log/syslog
        - /var/log/messages
```

name The message to send via SMTP

21.19.256 salt.states.snapper module

Managing implicit state and baselines using snapshots

New in version 2016.11.0.

Salt can manage state against explicitly defined state, for example if your minion state is defined by:

```
/etc/config_file:
  file.managed:
    - source: salt://configs/myconfig
```

If someone modifies this file, the next application of the highstate will allow the admin to correct this deviation and the file will be corrected.

Now, what happens if somebody creates a file `/etc/new_config_file` and deletes `/etc/important_config_file`? Unless you have a explicit rule, this change will go unnoticed.

The snapper state module allows you to manage state implicitly, in addition to explicit rules, in order to define a baseline and iterate with explicit rules as they show that they work in production.

The workflow is: once you have a working and audited system, you would create your baseline snapshot (eg. with `salt tgt snapper.create_snapshot`) and define in your state this baseline using the identifier of the snapshot (in this case: 20):

```
my_baseline:
  snapper.baseline_snapshot:
    - number: 20
    - include_diff: False
    - ignore:
      - /var/log
      - /var/cache
```

Baseline snapshots can be also referenced by tag. Most recent baseline snapshot is used in case of multiple snapshots with the same tag:

my_baseline_external_storage:

snapper.baseline_snapshot:

- tag: my_custom_baseline_tag
- config: external
- ignore: - /mnt/tmp_files/

If you have this state, and you haven't done changes to the system since the snapshot, and you add a user, the state will show you the changes (including full diffs) to `/etc/passwd`, `/etc/shadow`, etc if you call it with `test=True` and will undo all changes if you call it without.

This allows you to add more explicit state knowing that you are starting from a very well defined state, and that you can audit any change that is not part of your explicit configuration.

So after you made this your state, you decided to introduce a change in your configuration:

```
my_baseline:
  snapper.baseline_snapshot:
    - number: 20
    - ignore:
      - /var/log
      - /var/cache

hosts_entry:
  file.blockreplace:
    - name: /etc/hosts
    - content: 'First line of content'
    - append_if_not_found: True
```

The change in `/etc/hosts` will be done after any other change that deviates from the specified snapshot are reverted. This could be for example, modifications to the `/etc/passwd` file or changes in the `/etc/hosts` that could render your the `hosts_entry` rule void or dangerous.

Once you take a new snapshot and you update the baseline snapshot number to include the change in `/etc/hosts` the `hosts_entry` rule will basically do nothing. You are free to leave it there for documentation, to ensure that the change is made in case the snapshot is wrong, but if you remove anything that comes after the `snapper.baseline_snapshot` as it will have no effect; by the moment the state is evaluated, the baseline state was already applied and include this change.

Warning: Make sure you specify the baseline state before other rules, otherwise the baseline state will revert all changes if they are not present in the snapshot.

Warning: Do not specify more than one baseline rule as only the last one will affect the result.

codeauthor Duncan Mac-Vicar P. <dmacvicar@suse.de>

codeauthor Pablo Suárez Hernández <psuarezhernandez@suse.de>

maturity new

platform Linux

`salt.states.snapper.`**`baseline_snapshot`**(*name*, *number=None*, *tag=None*, *include_diff=True*, *config='root'*, *ignore=None*)

Enforces that no file is modified comparing against a previously defined snapshot identified by number.

number Number of selected baseline snapshot.

tag Tag of the selected baseline snapshot. Most recent baseline baseline snapshot is used in case of multiple snapshots with the same tag. (*tag* and *number* cannot be used at the same time)

include_diff Include a diff in the response (Default: True)

config Snapper config name (Default: root)

ignore List of files to ignore. (Default: None)

21.19.257 salt.states.solrcloud module

States for solrcloud alias and collection configuration

New in version 2017.7.0.

`salt.states.solrcloud.`**`alias`**(*name*, *collections*, ***kwargs*)

Create alias and enforce collection list.

Use the solrcloud module to get alias members and set them.

You can pass additional arguments that will be forwarded to http.query
name The collection name
collections list of collections to include in the alias

`salt.states.solrcloud.`**`collection`**(*name*, *options=None*, ***kwargs*)

Create collection and enforce options.

Use the solrcloud module to get collection parameters.

You can pass additional arguments that will be forwarded to http.query
name The collection name
options [{}] options to ensure

21.19.258 salt.states.splunk

Splunk User State Module

New in version 2016.3.0..

This state is used to ensure presence of users in splunk.

```
ensure example test user 1:
    splunk.present:
        - name: 'Example TestUser1'
        - email: example@domain.com
```

`salt.states.splunk.`**`absent`**(*email*, *profile='splunk'*, ***kwargs*)

Ensure a splunk user is absent

```
ensure example test user 1:
    splunk.absent:
        - email: 'example@domain.com'
        - name: 'exampleuser'
```

The following parameters are required:
email This is the email of the user in splunk
name This is the splunk username used to identify the user.

`salt.states.splunk.`**`present`**(*email*, *profile='splunk'*, ***kwargs*)

Ensure a user is present

```
ensure example test user 1:
    splunk.user_present:
        - realname: 'Example TestUser1'
        - name: 'exampleuser'
        - email: 'example@domain.com'
        - roles: ['user']
```

The following parameters are required:

email This is the email of the user in splunk

21.19.259 salt.states.splunk_search

Splunk Search State Module

New in version 2015.5.0.

This state is used to ensure presence of splunk searches.

```
server-warning-message:
  splunk_search.present:
    - name: This is the splunk search name
    - search: index=main sourcetype=
```

`salt.states.splunk_search.`**`absent`**(*name*, *profile='splunk'*)

Ensure a search is absent

```
API Error Search:
  splunk_search.absent
```

The following parameters are required:

name This is the name of the search in splunk

`salt.states.splunk_search.`**`present`**(*name*, *profile='splunk'*, ***kwargs*)

Ensure a search is present

```
API Error Search:
  splunk_search.present:
    search: index=main sourcetype=blah
    template: alert_5min
```

The following parameters are required:

name This is the name of the search in splunk

21.19.260 salt.states.sqlite3

Management of SQLite3 databases

New in version 2016.3.0.

depends

- SQLite3 Python Module

configuration See `salt.modules.sqlite3` for setup instructions

The sqlite3 module is used to create and manage sqlite3 databases and execute queries

Here is an example of creating a table using sql statements:

```
users:
  sqlite3.table_present:
    - db: /var/www/data/app.sqlite
    - schema: CREATE TABLE `users` (`username` TEXT COLLATE NOCASE UNIQUE NOT NULL,
↪`password` BLOB NOT NULL, `salt` BLOB NOT NULL, `last_login` INT)
```

Here is an example of creating a table using yaml/jinja instead of sql:

```
users:
  sqlite3.table_present:
    - db: /var/www/app.sqlite
    - schema:
      - email TEXT COLLATE NOCASE UNIQUE NOT NULL
      - firstname TEXT NOT NULL
      - lastname TEXT NOT NULL
      - company TEXT NOT NULL
      - password BLOB NOT NULL
      - salt BLOB NOT NULL
```

Here is an example of making sure a table is absent:

```
badservers:
  sqlite3.table_absent:
    - db: /var/www/data/users.sqlite
```

Sometimes you would to have specific data in tables to be used by other services Here is an example of making sure rows with specific data exist:

```
user_john_doe_xyz:
  sqlite3.row_present:
    - db: /var/www/app.sqlite
    - table: users
    - where_sql: email='john.doe@companyxyz.com'
    - data:
        email: john.doe@companyxyz.com
        lastname: doe
        firstname: john
        company: companyxyz.com
        password: abcdef012934125
        salt: abcdef012934125
    - require:
      - sqlite3: users
```

Here is an example of removing a row from a table:

```
user_john_doe_abc:
  sqlite3.row_absent:
    - db: /var/www/app.sqlite
    - table: users
    - where_sql: email="john.doe@companyabc.com"
    - require:
      - sqlite3: users
```

Note that there is no explicit state to perform random queries, however, this can be approximated with sqlite3's module functions and module.run:

```
zone-delete:
  module.run:
    - name: sqlite3.modify
    - db: {{ db }}
    - sql: "DELETE FROM records WHERE id > {{ count[0] }} AND domain_id = {{ domain_
↪id }}"
    - watch:
      - sqlite3: zone-insert-12
```

`salt.states.sqlite3.`**`row_absent`**(*name, db, table, where_sql, where_args=None*)

Makes sure the specified row is absent in db. If multiple rows match where_sql, then the state will fail.

name Only used as the unique ID

db The database file name

table The table name to check

where_sql The sql to select the row to check

where_args The list parameters to substitute in where_sql

`salt.states.sqlite3.`**`row_present`**`(`*name, db, table, data, where_sql, where_args=None, update=False*`)`

Checks to make sure the given row exists. If row exists and update is True then row will be updated with data. Otherwise it will leave existing row unmodified and check it against data. If the existing data doesn't match data_check the state will fail. If the row doesn't exist then it will insert data into the table. If more than one row matches, then the state will fail.

name Only used as the unique ID

db The database file name

table The table name to check the data

data The dictionary of key/value pairs to check against if row exists, insert into the table if it doesn't

where_sql The sql to select the row to check

where_args The list parameters to substitute in where_sql

update True will replace the existing row with data When False and the row exists and data does not equal the row data then the state will fail

`salt.states.sqlite3.`**`table_absent`**`(`*name, db*`)`

Make sure the specified table does not exist

name The name of the table

db The name of the database file

`salt.states.sqlite3.`**`table_present`**`(`*name, db, schema, force=False*`)`

Make sure the specified table exists with the specified schema

name The name of the table

db The name of the database file

schema The dictionary containing the schema information

force If the name of the table exists and force is set to False, the state will fail. If force is set to True, the existing table will be replaced with the new table

21.19.261 salt.states.ssh_auth

Control of entries in SSH authorized_key files

The information stored in a user's SSH authorized key file can be easily controlled via the ssh_auth state. Defaults can be set by the enc, options, and comment keys. These defaults can be overridden by including them in the name.

Since the YAML specification limits the length of simple keys to 1024 characters, and since SSH keys are often longer than that, you may have to use a YAML 'explicit key', as demonstrated in the second example below.

```
AAAAB3NzaC1kc3MAAACBAL0sQ9fJ5bYTEyY==:
  ssh_auth.present:
    - user: root
    - enc: ssh-dss

? AAAAB3NzaC1kc3MAAACBAL0sQ9fJ5bYTEyY==...
:
  ssh_auth.present:
    - user: root
    - enc: ssh-dss
```

(continues on next page)

(continued from previous page)

```
thatch:
  ssh_auth.present:
    - user: root
    - source: salt://ssh_keys/thatch.id_rsa.pub
    - config: '%h/.ssh/authorized_keys'

sshkeys:
  ssh_auth.present:
    - user: root
    - enc: ssh-rsa
    - options:
      - option1="value1"
      - option2="value2 flag2"
    - comment: myuser
    - names:
      - AAAAB3NzaC1kc3MAAACBAL0sQ9fJ5bYTEyY==
      - ssh-dss AAAAB3NzaCL0sQ9fJ5bYTEyY== user@domain
      - option3="value3" ssh-dss AAAAB3NzaC1kcQ9J5bYTEyY== other@testdomain
      - AAAAB3NzaC1kcQ9fJFF435bYTEyY== newcomment

sshkeys:
  ssh_auth.manage:
    - user: root
    - enc: ssh-rsa
    - options:
      - option1="value1"
      - option2="value2 flag2"
    - comment: myuser
    - ssh_keys:
      - AAAAB3NzaC1kc3MAAACBAL0sQ9fJ5bYTEyY==
      - ssh-dss AAAAB3NzaCL0sQ9fJ5bYTEyY== user@domain
      - option3="value3" ssh-dss AAAAB3NzaC1kcQ9J5bYTEyY== other@testdomain
      - AAAAB3NzaC1kcQ9fJFF435bYTEyY== newcomment
```

`salt.states.ssh_auth.`**`absent`**(*name, user, enc='ssh-rsa', comment='', source='', options=None, config='.ssh/authorized_keys', fingerprint_hash_type=None*)

Verifies that the specified SSH key is absent

name The SSH key to manage

user The user who owns the SSH authorized keys file to modify

enc Defines what type of key is being used; can be ed25519, ecdsa, ssh-rsa or ssh-dss

comment The comment to be placed with the SSH public key

options The options passed to the key, pass a list object

source The source file for the key(s). Can contain any number of public keys, in standard "authorized_keys" format. If this is set, comment, enc and options will be ignored.

New in version 2015.8.0.

config The location of the authorized keys file relative to the user's home directory, defaults to ".ssh/authorized_keys". Token expansion %u and %h for username and home path supported.

fingerprint_hash_type The public key fingerprint hash type that the public key fingerprint was originally hashed with. This defaults to `sha256` if not specified.

New in version 2016.11.7.

`salt.states.ssh_auth.`**`manage`**(*name, ssh_keys, user, enc='ssh-rsa', comment='', source='', options=None, config='.ssh/authorized_keys', fingerprint_hash_type=None, **kwargs*)

New in version 3000.

Ensures that only the specified ssh_keys are present for the specified user

ssh_keys The SSH key to manage

user The user who owns the SSH authorized keys file to modify

enc Defines what type of key is being used; can be ed25519, ecdsa, ssh-rsa or ssh-dss

comment The comment to be placed with the SSH public key

source The source file for the key(s). Can contain any number of public keys, in standard "authorized_keys" format. If this is set, comment and enc will be ignored.

Note: The source file must contain keys in the format `<enc> <key> <comment>`. If you have generated a keypair using PuTTYgen, then you will need to do the following to retrieve an OpenSSH-compatible public key.

1. In PuTTYgen, click `Load`, and select the *private* key file (not the public key), and click `Open`.
2. Copy the public key from the box labeled `Public key for pasting into OpenSSH authorized_keys file`.
3. Paste it into a new file.

---

options The options passed to the keys, pass a list object

config The location of the authorized keys file relative to the user's home directory, defaults to ".ssh/authorized_keys". Token expansion %u and %h for username and home path supported.

fingerprint_hash_type The public key fingerprint hash type that the public key fingerprint was originally hashed with. This defaults to `sha256` if not specified.

`salt.states.ssh_auth.`**`present`**`(`*name*, *user*, *enc='ssh-rsa'*, *comment="*, *source="*, *options=None*, *config='.ssh/authorized_keys'*, *fingerprint_hash_type=None*, ***kwargs*`)`

Verifies that the specified SSH key is present for the specified user

name The SSH key to manage

user The user who owns the SSH authorized keys file to modify

enc Defines what type of key is being used; can be ed25519, ecdsa, ssh-rsa or ssh-dss

comment The comment to be placed with the SSH public key

source The source file for the key(s). Can contain any number of public keys, in standard "authorized_keys" format. If this is set, comment and enc will be ignored.

Note: The source file must contain keys in the format `<enc> <key> <comment>`. If you have generated a keypair using PuTTYgen, then you will need to do the following to retrieve an OpenSSH-compatible public key.

1. In PuTTYgen, click `Load`, and select the *private* key file (not the public key), and click `Open`.
2. Copy the public key from the box labeled `Public key for pasting into OpenSSH authorized_keys file`.
3. Paste it into a new file.

---

options The options passed to the key, pass a list object

config The location of the authorized keys file relative to the user's home directory, defaults to ".ssh/authorized_keys". Token expansion %u and %h for username and home path supported.

fingerprint_hash_type The public key fingerprint hash type that the public key fingerprint was originally hashed with. This defaults to `sha256` if not specified.

21.19.262 salt.states.ssh_known_hosts

Control of SSH known_hosts entries

Manage the information stored in the known_hosts files.

```
github.com:
  ssh_known_hosts:
    - present
    - user: root
    - fingerprint: 16:27:ac:a5:76:28:2d:36:63:1b:56:4d:eb:df:a6:48
    - fingerprint_hash_type: md5

example.com:
  ssh_known_hosts:
    - absent
    - user: root
```

`salt.states.ssh_known_hosts.`**`absent`**`(`*name*, *user=None*, *config=None*`)`

Verifies that the specified host is not known by the given user

name The host name Note that only single host names are supported. If foo.example.com and bar.example.com are the same machine and you need to exclude both, you will need one Salt state for each.

user The user who owns the ssh authorized keys file to modify

config The location of the authorized keys file relative to the user's home directory, defaults to ".ssh/known_hosts". If no user is specified, defaults to "/etc/ssh/ssh_known_hosts". If present, must be an absolute path when a user is not specified.

`salt.states.ssh_known_hosts.`**`present`**`(`*name*, *user=None*, *fingerprint=None*, *key=None*, *port=None*, *enc=None*, *config=None*, *hash_known_hosts=True*, *timeout=5*, *fingerprint_hash_type=None*`)`

Verifies that the specified host is known by the specified user

On many systems, specifically those running with openssh 4 or older, the `enc` option must be set, only openssh 5 and above can detect the key type.

name The name of the remote host (e.g. "github.com") Note that only a single hostname is supported, if foo.example.com and bar.example.com have the same host you will need two separate Salt States to represent them.

user The user who owns the ssh authorized keys file to modify

fingerprint The fingerprint of the key which must be present in the known_hosts file (optional if key specified)

key The public key which must be present in the known_hosts file (optional if fingerprint specified)

port optional parameter, port which will be used to when requesting the public key from the remote host, defaults to port 22.

enc Defines what type of key is being used, can be ed25519, ecdsa ssh-rsa or ssh-dss

config The location of the authorized keys file relative to the user's home directory, defaults to ".ssh/known_hosts". If no user is specified, defaults to "/etc/ssh/ssh_known_hosts". If present, must be an absolute path when a user is not specified.

hash_known_hosts [True] Hash all hostnames and addresses in the known hosts file.

timeout [int] Set the timeout for connection attempts. If `timeout` seconds have elapsed since a connection was initiated to a host or since the last time anything was read from that host, then the connection is closed and the host in question considered unavailable. Default is 5 seconds.

New in version 2016.3.0.

fingerprint_hash_type The public key fingerprint hash type that the public key fingerprint was originally hashed with. This defaults to `sha256` if not specified.

New in version 2016.11.4.

Changed in version 2017.7.0:: default changed from `md5` to `sha256`

21.19.263 salt.states.stateconf

Stateconf System

The stateconf system is intended for use only with the stateconf renderer. This State module presents the set function. This function does not execute any functionality, but is used to interact with the stateconf renderer.

`salt.states.stateconf.`**`context`**`(`*name*, ***kwargs*`)`
No-op state to support state config via the stateconf renderer.

`salt.states.stateconf.`**`set`**`(`*name*, ***kwargs*`)`
No-op state to support state config via the stateconf renderer.

21.19.264 salt.states.status

Minion status monitoring

Maps to the *status* execution module.

`salt.states.status.`**`loadavg`**`(`*name*, *maximum=None*, *minimum=None*`)`
Return the current load average for the specified minion. Available values for name are *1-min*, *5-min* and *15-min*. *minimum* and *maximum* values should be passed in as strings.

`salt.states.status.`**`process`**`(`*name*`)`
Return whether the specified signature is found in the process tree. This differs slightly from the services states, in that it may refer to a process that is not managed via the init system.

21.19.265 salt.states.statuspage

StatusPage

Manage the StatusPage configuration.

In the minion configuration file, the following block is required:

```
statuspage:
  api_key: <API_KEY>
  page_id: <PAGE_ID>
```

New in version 2017.7.0.

`salt.states.statuspage.`**`create`**`(`*name*, *endpoint='incidents'*, *api_url=None*, *page_id=None*, *api_key=None*, *api_version=None*, ***kwargs*`)`
Insert a new entry under a specific endpoint.

endpoint: incidents Insert under this specific endpoint.

page_id Page ID. Can also be specified in the config file.

api_key API key. Can also be specified in the config file.

api_version: 1 API version. Can also be specified in the config file.

api_url Custom API URL in case the user has a StatusPage service running in a custom environment.

kwargs Other params.

SLS Example:

```
create-my-component:
    statuspage.create:
        - endpoint: components
        - name: my component
        - group_id: 993vgplshj12
```

`salt.states.statuspage.`**`delete`**(*name*, *endpoint='incidents'*, *id=None*, *api_url=None*, *page_id=None*, *api_key=None*, *api_version=None*)

Remove an entry from an endpoint.

endpoint: **incidents** Request a specific endpoint.

page_id Page ID. Can also be specified in the config file.

api_key API key. Can also be specified in the config file.

api_version: **1** API version. Can also be specified in the config file.

api_url Custom API URL in case the user has a StatusPage service running in a custom environment.

SLS Example:

```
delete-my-component:
    statuspage.delete:
        - endpoint: components
        - id: ftgks51sfs2d
```

`salt.states.statuspage.`**`managed`**(*name*, *config*, *api_url=None*, *page_id=None*, *api_key=None*, *api_version=None*, *pace=1*, *allow_empty=False*)

Manage the StatusPage configuration.

config Dictionary with the expected configuration of the StatusPage. The main level keys of this dictionary represent the endpoint name. If a certain endpoint does not exist in this structure, it will be ignored / not configured.

page_id Page ID. Can also be specified in the config file.

api_key API key. Can also be specified in the config file.

api_version: **1** API version. Can also be specified in the config file.

api_url Custom API URL in case the user has a StatusPage service running in a custom environment.

pace: **1** Max requests per second allowed by the API.

allow_empty: **False** Allow empty config.

SLS example:

```
my-statuspage-config:
    statuspage.managed:
        - config:
            components:
                - name: component1
                  group_id: uy4g37rf
                - name: component2
                  group_id: 3n4uyu4gf
            incidents:
                - name: incident1
                  status: resolved
                  impact: major
                  backfilled: false
                - name: incident2
                  status: investigating
                  impact: minor
```

`salt.states.statuspage.`**`update`**(*name*, *endpoint='incidents'*, *id=None*, *api_url=None*, *page_id=None*, *api_key=None*, *api_version=None*, ***kwargs*)

Update attribute(s) of a specific endpoint.

id The unique ID of the enpoint entry.

endpoint: **incidents** Endpoint name.

page_id Page ID. Can also be specified in the config file.

api_key API key. Can also be specified in the config file.

api_version: **1** API version. Can also be specified in the config file.

api_url Custom API URL in case the user has a StatusPage service running in a custom environment.

SLS Example:

```
update-my-incident:
    statuspage.update:
        - id: dz959yz2nd4l
        - status: resolved
```

21.19.266 salt.states.supervisord

Interaction with the Supervisor daemon

```
wsgi_server:
  supervisord.running:
    - require:
      - pkg: supervisor
    - watch:
      - file: /etc/nginx/sites-enabled/wsgi_server.conf
```

`salt.states.supervisord.`**`dead`**(*name*, *user=None*, *conf_file=None*, *bin_env=None*, ***kwargs*)

Ensure the named service is dead (not running).

name Service name as defined in the supervisor configuration file

user Name of the user to run the supervisorctl command

New in version 0.17.0.

conf_file path to supervisorctl config file

bin_env path to supervisorctl bin or path to virtualenv with supervisor installed

`salt.states.supervisord.`**`mod_watch`**(*name*, *restart=True*, *update=False*, *user=None*, *conf_file=None*, *bin_env=None*, ***kwargs*)

The supervisord watcher, called to invoke the watch command. Always restart on watch

Note: This state exists to support special handling of the `watch` *requisite*. It should not be called directly.

Parameters for this function should be set by the state being triggered.

`salt.states.supervisord.`**`running`**(*name*, *restart=False*, *update=False*, *user=None*, *conf_file=None*, *bin_env=None*, ***kwargs*)

Ensure the named service is running.

name Service name as defined in the supervisor configuration file

restart Whether to force a restart

update Whether to update the supervisor configuration.

user Name of the user to run the supervisorctl command

New in version 0.17.0.

conf_file path to supervisorctl config file

bin_env path to supervisorctl bin or path to virtualenv with supervisor installed

21.19.267 salt.states.svn

Manage SVN repositories

Manage repository checkouts via the svn vcs system. Note that subversion must be installed for these states to be available, so svn states should include a requisite to a pkg.installed state for the package which provides subversion (`subversion` in most cases). Example:

```
subversion:
  pkg.installed

http://unladen-swallow.googlecode.com/svn/trunk/:
  svn.latest:
    - target: /tmp/swallow
```

`salt.states.svn.`**`dirty`**`(`*name, target, user=None, username=None, password=None, ignore_unversioned=False*`)`

Determine if the working directory has been changed.

`salt.states.svn.`**`export`**`(`*name, target=None, rev=None, user=None, username=None, password=None, force=False, overwrite=False, externals=True, trust=False, trust_failures=None*`)`

Export a file or directory from an SVN repository

name Address and path to the file or directory to be exported.

target Name of the target directory where the checkout will put the working directory

rev [None] The name revision number to checkout. Enable "force" if the directory already exists.

user [None] Name of the user performing repository management operations

username [None] The user to access the name repository with. The svn default is the current user

password Connect to the Subversion server with this password

New in version 0.17.0.

force [False] Continue if conflicts are encountered

overwrite [False] Overwrite existing target

externals [True] Change to False to not checkout or update externals

trust [False] Automatically trust the remote server. SVN's –trust-server-cert

trust_failures [None] Comma-separated list of certificate trust failures, that shall be ignored. This can be used if trust=True is not sufficient. The specified string is passed to SVN's –trust-server-cert-failures option as-is.

New in version 2019.2.0.

`salt.states.svn.`**`latest`**`(`*name, target=None, rev=None, user=None, username=None, password=None, force=False, externals=True, trust=False, trust_failures=None*`)`

Checkout or update the working directory to the latest revision from the remote repository.

name Address of the name repository as passed to "svn checkout"

target Name of the target directory where the checkout will put the working directory

rev [None] The name revision number to checkout. Enable "force" if the directory already exists.

user [None] Name of the user performing repository management operations

username [None] The user to access the name repository with. The svn default is the current user

password Connect to the Subversion server with this password

New in version 0.17.0.

force [False] Continue if conflicts are encountered

externals [True] Change to False to not checkout or update externals

trust [False] Automatically trust the remote server. SVN's –trust-server-cert

trust_failures [None] Comma-separated list of certificate trust failures, that shall be ignored. This can be used if trust=True is not sufficient. The specified string is passed to SVN's –trust-server-cert-failures option as-is.

New in version 2019.2.0.

21.19.268 salt.states.sysctl

Configuration of the kernel using sysctl

Control the kernel sysctl system.

```
vm.swappiness:
  sysctl.present:
    - value: 20
```

`salt.states.sysctl.`**`present`**`(`*name*, *value*, *config=None*`)`

Ensure that the named sysctl value is set in memory and persisted to the named configuration file. The default sysctl configuration file is /etc/sysctl.conf

name The name of the sysctl value to edit

value The sysctl value to apply

config The location of the sysctl configuration file. If not specified, the proper location will be detected based on platform.

21.19.269 salt.states.syslog_ng

State module for syslog_ng

maintainer Tibor Benke <btibi@sch.bme.hu>

maturity new

depends cmd, ps, syslog_ng

platform all

Users can generate syslog-ng configuration files from YAML format or use plain ones and reload, start, or stop their syslog-ng by using this module.

Details

The service module is not available on all system, so this module includes `syslog_ng.reloaded`, `syslog_ng.stopped`, and `syslog_ng.started` functions. If the service module is available on the computers, users should use that.

Users can generate syslog-ng configuration with `syslog_ng.config` function. For more information see *syslog-ng state usage*.

Syslog-ng configuration file format

The syntax of a configuration snippet in syslog-ng.conf:

object_type object_id {<options>};

These constructions are also called statements. There are options inside of them:

option(parameter1, parameter2); option2(parameter1, parameter2);

You can find more information about syslog-ng's configuration syntax in the Syslog-ng Admin guide: http://www.balabit.com/sites/default/files/documents/syslog-ng-ose-3.5-guides/en/syslog-ng-ose-v3.5-guide-admin/html-single/index.html#syslog-ng.conf.5

`salt.states.syslog_ng.`**`config`**(*name*, *config*, *write=True*)

Builds syslog-ng configuration.

name : the id of the Salt document config : the parsed YAML code write : if True, it writes the config into the configuration file, otherwise just returns it

`salt.states.syslog_ng.`**`reloaded`**(*name*)

Reloads syslog-ng.

`salt.states.syslog_ng.`**`started`**(*name=None*, *user=None*, *group=None*, *chroot=None*, *caps=None*, *no_caps=False*, *pidfile=None*, *enable_core=False*, *fd_limit=None*, *verbose=False*, *debug=False*, *trace=False*, *yydebug=False*, *persist_file=None*, *control=None*, *worker_threads=None*, **args*, ***kwargs*)

Ensures, that syslog-ng is started via the given parameters.

Users shouldn't use this function, if the service module is available on their system.

`salt.states.syslog_ng.`**`stopped`**(*name=None*)

Kills syslog-ng.

21.19.270 salt.states.sysrc

State to work with sysrc

`salt.states.sysrc.`**`absent`**(*name*, ***kwargs*)

Ensure a sysrc variable is absent.
name The variable name to set
file (optional) The rc file to add the variable to.
jail (option) the name or JID of the jail to set the value in.

`salt.states.sysrc.`**`managed`**(*name*, *value*, ***kwargs*)

Ensure a sysrc variable is set to a specific value.
name The variable name to set
value Value to set the variable to
file (optional) The rc file to add the variable to.
jail (option) the name or JID of the jail to set the value in.
Example:

```
syslogd:
  sysrc.managed:
    - name: syslogd_flags
    - value: -ss
```

21.19.271 salt.states.telemetry_alert

Manage Telemetry alert configurations

New in version 2016.3.0.

Create, Update and destroy Mongo Telemetry alert configurations.

This module uses requests, which can be installed via package, or pip.

This module accepts explicit credential (telemetry api key) or can also read api key credentials from a pillar. Example:

```
ensure telemetry alert X is defined on deployment Y:
    telemetry_alert.present:
        - deployment_id: "rs-XXXXXX"
        - metric_name: "testMetric"
        - alert_config:
            max: 1
            filter:  SERVER_ROLE_MONGOD_PRIMARY
            escalate_to: "example@pagerduty.com"
        - name: "**MANAGED BY ORCA DO NOT EDIT BY HAND** manages alarm on testMetric"
```

`salt.states.telemetry_alert.`**`absent`**`(`*name, deployment_id, metric_name, api_key=None, profile='telemetry'*`)`

Ensure the telemetry alert config is deleted

name An optional description of the alarms (not currently supported by telemetry API)

deployment_id Specifies the ID of the root deployment resource (replica set cluster or sharded cluster) to which this alert definition is attached

metric_name Specifies the unique ID of the metric to whose values these thresholds will be applied

api_key Telemetry api key for the user

profile A dict with telemetry config data. If present, will be used instead of api_key.

`salt.states.telemetry_alert.`**`present`**`(`*name, deployment_id, metric_name, alert_config, api_key=None, profile='telemetry'*`)`

Ensure the telemetry alert exists.

name An optional description of the alarm (not currently supported by telemetry API)

deployment_id Specifies the ID of the root deployment resource (replica set cluster or sharded cluster) to which this alert definition is attached

metric_name Specifies the unique ID of the metric to whose values these thresholds will be applied

alert_config: Is a list of dictionaries where each dict contains the following fields:

- **filter** By default the alert will apply to the deployment and all its constituent resources. If the alert only applies to a subset of those resources, a filter may be specified to narrow this scope.
- **min** the smallest "ok" value the metric may take on; if missing or null, no minimum is enforced.
- **max** the largest "ok" value the metric may take on; if missing or null, no maximum is enforced.
- **notify_all** Used to indicate if you want to alert both onCallEngineer and apiNotifications

api_key Telemetry api key for the user

profile A dict of telemetry config information. If present, will be used instead of api_key.

21.19.272 salt.states.test

Test States

Provide test case states that enable easy testing of things to do with state calls, e.g. running, calling, logging, output filtering etc.

```
always-passes-with-any-kwarg:
  test.nop:
    - name: foo
    - something: else
    - foo: bar

always-passes:
  test.succeed_without_changes:
    - name: foo

always-fails:
```

(continues on next page)

(continued from previous page)

```
  test.fail_without_changes:
    - name: foo

always-changes-and-succeeds:
  test.succeed_with_changes:
    - name: foo

always-changes-and-fails:
  test.fail_with_changes:
    - name: foo

my-custom-combo:
  test.configurable_test_state:
    - name: foo
    - changes: True
    - result: False
    - comment: bar.baz
    - warnings: A warning

is-pillar-foo-present-and-bar-is-int:
  test.check_pillar:
    - present:
        - foo
    - integer:
        - bar
```

`salt.states.test.`**`check_pillar`**(*name, present=None, boolean=None, integer=None, string=None, listing=None, dictionary=None, verbose=False*)

Checks the presence and, optionally, the type of given keys in Pillar. Supported kwargs for types are: - boolean (bool) - integer (int) - string (str) - listing (list) - dictionary (dict)

Checking for None type pillars is not implemented yet.

```
is-pillar-foo-present-and-bar-is-int:
  test.check_pillar:
    - present:
        - foo
    - integer:
        - bar
```

`salt.states.test.`**`configurable_test_state`**(*name, changes=True, result=True, comment=", warnings=None*)

A configurable test state which determines its output based on the inputs.

New in version 2014.7.0.
name: A unique string.
changes: Do we return anything in the changes field? Accepts True, False, and 'Random' Default is True
result: Do we return successfully or not? Accepts True, False, and 'Random' Default is True If test is True and changes is True, this will be None. If test is True and and changes is False, this will be True.
comment: String to fill the comment field with. Default is "
New in version 3000.

Warning: A string (or a list of strings) to fill the warnings field with. Default is None

`salt.states.test.`**`fail_with_changes`**(*name, **kwargs*)

Returns failure and changes is not empty.

New in version 2014.7.0.
name: A unique string.

`salt.states.test.`**`fail_without_changes`**`(`*name*, ***kwargs*`)`
Returns failure.

New in version 2014.7.0.
name: A unique string.

`salt.states.test.`**`mod_watch`**`(`*name*, *sfun=None*, ***kwargs*`)`
Call this function via a watch statement

New in version 2014.7.0.

Any parameters in the state return dictionary can be customized by adding the keywords `result`, `comment`, and `changes`.

```
this_state_will_return_changes:
  test.succeed_with_changes

this_state_will_NOT_return_changes:
  test.succeed_without_changes

this_state_is_watching_another_state:
  test.succeed_without_changes:
    - comment: 'This is a custom comment'
    - watch:
      - test: this_state_will_return_changes
      - test: this_state_will_NOT_return_changes

this_state_is_also_watching_another_state:
  test.succeed_without_changes:
    - watch:
      - test: this_state_will_NOT_return_changes
```

`salt.states.test.`**`nop`**`(`*name*, ***kwargs*`)`
A no-op state that does nothing. Useful in conjunction with the *use* requisite, or in templates which could otherwise be empty due to jinja rendering

New in version 2015.8.1.

`salt.states.test.`**`show_notification`**`(`*name*, *text=None*, ***kwargs*`)`
Simple notification using text argument.

New in version 2015.8.0.
name A unique string.
text Text to return in the comment.

`salt.states.test.`**`succeed_with_changes`**`(`*name*, ***kwargs*`)`
Returns successful and changes is not empty

New in version 2014.7.0.
name: A unique string.

`salt.states.test.`**`succeed_without_changes`**`(`*name*, ***kwargs*`)`
Returns successful.

New in version 2014.7.0.
name A unique string.

21.19.273 salt.states.testinframod module

21.19.274 salt.states.timezone

Management of timezones

The timezone can be managed for the system:

```
America/Denver:
  timezone.system
```

The system and the hardware clock are not necessarily set to the same time. By default, the hardware clock is set to localtime, meaning it is set to the same time as the system clock. If *utc* is set to True, then the hardware clock will be set to UTC, and the system clock will be an offset of that.

```
America/Denver:
  timezone.system:
    - utc: True
```

The Ubuntu community documentation contains an explanation of this setting, as it applies to systems that dual-boot with Windows. This is explained in greater detail here.

`salt.states.timezone.`**`system`**`(`*name*, *utc=True*`)`

Set the timezone for the system.

name The name of the timezone to use (e.g.: America/Denver)

utc Whether or not to set the hardware clock to UTC (default is True)

21.19.275 salt.states.tls

Enforce state for SSL/TLS

`salt.states.tls.`**`valid_certificate`**`(`*name*, *weeks=0*, *days=0*, *hours=0*, *minutes=0*, *seconds=0*`)`

Verify that a TLS certificate is valid now and (optionally) will be valid for the time specified through weeks, days, hours, minutes, and seconds.

21.19.276 salt.states.tomcat

Manage Apache Tomcat web applications

Note: This state requires the Tomcat Manager webapp to be installed and running.

The following grains/pillars must be set for communication with Tomcat Manager to work:

```
tomcat-manager:
    user: 'tomcat-manager'
    passwd: 'Passw0rd'
```

Configuring Tomcat Manager

To manage webapps via the Tomcat Manager, you'll need to configure a valid user in the file `conf/tomcat-users.xml`.

Listing 1: conf/tomcat-users.xml

```
<?xml version='1.0' encoding='utf-8'?>
<tomcat-users>
    <role rolename="manager-script"/>
    <user username="tomcat-manager" password="Passw0rd" roles="manager-script"/>
</tomcat-users>
```

Notes

- Using multiple versions (aka. parallel deployments) on the same context path is not supported.
- More information about the Tomcat Manager: http://tomcat.apache.org/tomcat-7.0-doc/manager-howto.html
- If you use only this module for deployments you might want to restrict access to the manager so it's only accessible via localhost. For more info: http://tomcat.apache.org/tomcat-7.0-doc/manager-howto.html#Configuring_Manager_Application_Access
- **Last tested on**:

  Tomcat Version: Apache Tomcat/7.0.54

  JVM Vendor: Oracle Corporation

  JVM Version: 1.8.0_101-b13

  OS Architecture: amd64

  OS Name: Linux

  OS Version: 3.10.0-327.22.2.el7.x86_64

`salt.states.tomcat.`**`mod_watch`**(*name*, *url='http://localhost:8080/manager'*, *timeout=180*)

The tomcat watcher, called to invoke the watch command. When called, it will reload the webapp in question

---

Note: This state exists to support special handling of the `watch` *requisite*. It should not be called directly.

Parameters for this function should be set by the state being triggered.

`salt.states.tomcat.`**`undeployed`**(*name*, *url='http://localhost:8080/manager'*, *timeout=180*)

Enforce that the WAR will be undeployed from the server

name The context path to undeploy.

url [http://localhost:8080/manager] The URL of the server with the Tomcat Manager webapp.

timeout [180] Timeout for HTTP request to the Tomcat Manager.

Example:

```
jenkins:
  tomcat.undeployed:
    - name: /ran
    - require:
      - service: application-service
```

`salt.states.tomcat.`**`wait`**(*name*, *url='http://localhost:8080/manager'*, *timeout=180*)

Wait for the Tomcat Manager to load.

Notice that if tomcat is not running we won't wait for it start and the state will fail. This state can be required in the tomcat.war_deployed state to make sure tomcat is running and that the manager is running as well and ready for deployment.

url [http://localhost:8080/manager] The URL of the server with the Tomcat Manager webapp.
timeout [180] Timeout for HTTP request to the Tomcat Manager.
Example:

```
tomcat-service:
  service.running:
    - name: tomcat
    - enable: True

wait-for-tomcatmanager:
  tomcat.wait:
    - timeout: 300
    - require:
      - service: tomcat-service

jenkins:
  tomcat.war_deployed:
    - name: /ran
    - war: salt://jenkins-1.2.4.war
    - require:
      - tomcat: wait-for-tomcatmanager
```

`salt.states.tomcat.`**`war_deployed`**`(`*name, war, force=False, url='http://localhost:8080/manager', timeout=180, temp_war_location=None, version=True*`)`

Enforce that the WAR will be deployed and started in the context path, while making use of WAR versions in the filename.

Note: For more info about Tomcats file paths and context naming, please see http://tomcat.apache.org/tomcat-7.0-doc/config/context.html#Naming

---

name The context path to deploy (incl. forward slash) the WAR to.
war Absolute path to WAR file (should be accessible by the user running Tomcat) or a path supported by the `salt.modules.cp.get_url` function.
force [False] Force deployment even if the version strings are the same. Disabled by default.
url [http://localhost:8080/manager] The URL of the Tomcat Web Application Manager.
timeout [180] Timeout for HTTP requests to the Tomcat Manager.
temp_war_location [None] Use another location to temporarily copy the WAR file to. By default the system's temp directory is used.
version [''] Specify the WAR version. If this argument is provided, it overrides the version encoded in the WAR file name, if one is present.

New in version 2015.8.6.

Use `False` or blank value to prevent guessing the version and keeping it blank.

New in version 2016.11.0.

Example:

```
jenkins:
  tomcat.war_deployed:
    - name: /salt-powered-jenkins
    - war: salt://jenkins-1.2.4.war
    - require:
      - service: application-service
```

Note: Be aware that in the above example the WAR `jenkins-1.2.4.war` will be deployed to the context path `salt-powered-jenkins##1.2.4`. To avoid this either specify a version yourself, or set version to `False`.

21.19.277 salt.states.trafficserver

Control Apache Traffic Server

New in version 2015.8.0.

`salt.states.trafficserver.`**`bounce_cluster`**`(`*name*`)`

Bounce all Traffic Server nodes in the cluster. Bouncing Traffic Server shuts down and immediately restarts Traffic Server, node-by-node.

```
bounce_ats_cluster:
  trafficserver.bounce_cluster
```

`salt.states.trafficserver.`**`bounce_local`**`(`*name, drain=False*`)`

Bounce Traffic Server on the local node. Bouncing Traffic Server shuts down and immediately restarts the Traffic Server node.

This option modifies the behavior of traffic_line -b and traffic_line -L such that traffic_server is not shut down until the number of active client connections drops to the number given by the proxy.config.restart.active_client_threshold configuration variable.

```
bounce_ats_local:
  trafficserver.bounce_local

bounce_ats_local:
  trafficserver.bounce_local
    - drain: True
```

`salt.states.trafficserver.`**`clear_cluster`**`(`*name*`)`

Clears accumulated statistics on all nodes in the cluster.

```
clear_ats_cluster:
  trafficserver.clear_cluster
```

`salt.states.trafficserver.`**`clear_node`**`(`*name*`)`

Clears accumulated statistics on the local node.

```
clear_ats_node:
  trafficserver.clear_node
```

`salt.states.trafficserver.`**`config`**`(`*name, value*`)`

Set Traffic Server configuration variable values.

```
proxy.config.proxy_name:
  trafficserver.config:
    - value: cdn.site.domain.tld

OR

traffic_server_setting:
```

(continues on next page)

(continued from previous page)

```
  trafficserver.config:
    - name: proxy.config.proxy_name
    - value: cdn.site.domain.tld
```

`salt.states.trafficserver.offline(name, path)`

Mark a cache storage device as offline. The storage is identified by a path which must match exactly a path specified in storage.config. This removes the storage from the cache and redirects requests that would have used this storage to other storage. This has exactly the same effect as a disk failure for that storage. This does not persist across restarts of the traffic_server process.

```
offline_ats_path:
  trafficserver.offline:
    - path: /path/to/cache
```

`salt.states.trafficserver.refresh(name)`

Initiate a Traffic Server configuration file reread. Use this command to update the running configuration after any configuration file modification.

The timestamp of the last reconfiguration event (in seconds since epoch) is published in the proxy.node.config.reconfigure_time metric.

```
refresh_ats:
  trafficserver.refresh
```

`salt.states.trafficserver.restart_cluster(name)`

Restart the traffic_manager process and the traffic_server process on all the nodes in a cluster.

```
restart_ats_cluster:
  trafficserver.restart_cluster
```

`salt.states.trafficserver.restart_local(name, drain=False)`

Restart the traffic_manager and traffic_server processes on the local node.

This option modifies the behavior of traffic_line -b and traffic_line -L such that traffic_server is not shut down until the number of active client connections drops to the number given by the proxy.config.restart.active_client_threshold configuration variable.

```
restart_ats_local:
  trafficserver.restart_local

restart_ats_local_drain:
  trafficserver.restart_local
    - drain: True
```

`salt.states.trafficserver.shutdown(name)`

Shut down Traffic Server on the local node.

```
shutdown_ats:
  trafficserver.shutdown
```

`salt.states.trafficserver.startup(name)`

Start Traffic Server on the local node.

```
startup_ats:
  trafficserver.startup
```

`salt.states.trafficserver.`**`zero_cluster`**`(`*name*`)`

Reset performance statistics to zero across the cluster.

```
zero_ats_cluster:
  trafficserver.zero_cluster
```

`salt.states.trafficserver.`**`zero_node`**`(`*name*`)`

Reset performance statistics to zero on the local node.

```
zero_ats_node:
  trafficserver.zero_node
```

21.19.278 salt.states.tuned

Interface to Red Hat tuned-adm module

maintainer Syed Ali <alicsyed@gmail.com>

maturity new

depends cmd.run

platform Linux

`salt.states.tuned.`**`off`**`(`*name=None*`)`

Turns 'tuned' off. Example tuned.sls file for turning tuned off:
tuned: tuned.off: []
To see a valid list of states call execution module: *`tuned.list`*

`salt.states.tuned.`**`profile`**`(`*name*`)`

This state module allows you to modify system tuned parameters

Example tuned.sls file to set profile to virtual-guest
tuned:
    tuned:
        - profile
        - name: virtual-guest

name tuned profile name to set the system to
To see a valid list of states call execution module: *`tuned.list`*

21.19.279 salt.states.uptime

Monitor Web Server with Uptime

Uptime is an open source remote monitoring application using Node.js, MongoDB, and Twitter Bootstrap.

Warning: This state module is beta. It might be changed later to include more or less automation.

Note: This state module requires a pillar to specify the location of your uptime install

```
uptime:
  application_url: "http://uptime-url.example.org"
```

Example:

```
url:
  uptime.monitored
url/sitemap.xml:
  uptime.monitored:
     - polling: 600 # every hour
```

`salt.states.uptime.`**`monitored`**(*name*, ***params*)

Makes sure an URL is monitored by uptime. Checks if URL is already monitored, and if not, adds it.

21.19.280 salt.states.user

Management of user accounts

The user module is used to create and manage user settings, users can be set as either absent or present

```
fred:
  user.present:
    - fullname: Fred Jones
    - shell: /bin/zsh
    - home: /home/fred
    - uid: 4000
    - gid: 4000
    - groups:
      - wheel
      - storage
      - games

testuser:
  user.absent
```

`salt.states.user.`**`absent`**(*name*, *purge=False*, *force=False*)

Ensure that the named user is absent

name The name of the user to remove

purge Set purge to True to delete all of the user's files as well as the user, Default is `False`.

force If the user is logged in, the absent state will fail. Set the force option to True to remove the user even if they are logged in. Not supported in FreeBSD and Solaris, Default is `False`.

`salt.states.user.`**`present`**(*name*, *uid=None*, *gid=None*, *gid_from_name=False*, *groups=None*, *optional_groups=None*, *remove_groups=True*, *home=None*, *createhome=True*, *password=None*, *hash_password=False*, *enforce_password=True*, *empty_password=False*, *shell=None*, *unique=True*, *system=False*, *fullname=None*, *roomnumber=None*, *workphone=None*, *homephone=None*, *other=None*, *loginclass=None*, *date=None*, *mindays=None*, *maxdays=None*, *inactdays=None*, *warndays=None*, *expire=None*, *win_homedrive=None*, *win_profile=None*, *win_logonscript=None*, *win_description=None*, *nologinit=False*, *allow_uid_change=False*, *allow_gid_change=False*)

Ensure that the named user is present with the specified properties

name The name of the user to manage

uid The user id to assign. If not specified, and the user does not exist, then the next available uid will be assigned.

gid The id of the default group to assign to the user. Either a group name or gid can be used. If not specified, and the user does not exist, then the next available gid will be assigned.

gid_from_name [False] If `True`, the default group id will be set to the id of the group with the same name as the user. If the group does not exist the state will fail.

allow_uid_change [False] Set to `True` to allow the state to update the uid.

New in version 2018.3.1.

allow_gid_change [False] Set to `True` to allow the state to update the gid.

New in version 2018.3.1.

groups A list of groups to assign the user to, pass a list object. If a group specified here does not exist on the minion, the state will fail. If set to the empty list, the user will be removed from all groups except the default group. If unset, salt will assume current groups are still wanted (see issue #28706).

optional_groups A list of groups to assign the user to, pass a list object. If a group specified here does not exist on the minion, the state will silently ignore it.

NOTE: If the same group is specified in both "groups" and "optional_groups", then it will be assumed to be required and not optional.

remove_groups Remove groups that the user is a member of that weren't specified in the state, Default is `True`.

home The custom login directory of user. Uses default value of underlying system if not set. Notice that this directory does not have to exist. This also the location of the home directory to create if createhome is set to True.

createhome [True] If set to `False`, the home directory will not be created if it doesn't already exist.

> **Warning:** Not supported on Windows or Mac OS.
>
> Additionally, parent directories will *not* be created. The parent directory for `home` must already exist.

nologinit [False] If set to `True`, it will not add the user to lastlog and faillog databases.

> **Note:** Not supported on Windows or Mac OS.

password A password hash to set for the user. This field is only supported on Linux, FreeBSD, NetBSD, OpenBSD, and Solaris. If the `empty_password` argument is set to `True` then `password` is ignored. For Windows this is the plain text password. For Linux, the hash can be generated with `openssl passwd -1`.

Changed in version 0.16.0: BSD support added.

hash_password Set to True to hash the clear text password. Default is `False`.

enforce_password Set to False to keep the password from being changed if it has already been set and the password hash differs from what is specified in the "password" field. This option will be ignored if "password" is not specified, Default is `True`.

empty_password Set to True to enable password-less login for user, Default is `False`.

shell The login shell, defaults to the system default shell

unique Require a unique UID, Default is `True`.

system Choose UID in the range of FIRST_SYSTEM_UID and LAST_SYSTEM_UID, Default is `False`.

loginclass The login class, defaults to empty (BSD only)

User comment field (GECOS) support (currently Linux, BSD, and MacOS only):

The below values should be specified as strings to avoid ambiguities when the values are loaded. (Especially the phone and room number fields which are likely to contain numeric data)

fullname The user's full name

roomnumber The user's room number (not supported in MacOS)

workphone The user's work phone number (not supported in MacOS)

homephone The user's home phone number (not supported in MacOS)

other The user's other attribute (not supported in MacOS) If GECOS field contains more than 4 commas, this field will have the rest of 'em

Changed in version 2014.7.0: Shadow attribute support added.

Shadow attributes support (currently Linux only):

The below values should be specified as integers.

date Date of last change of password, represented in days since epoch (January 1, 1970).

mindays The minimum number of days between password changes.

maxdays The maximum number of days between password changes.

inactdays The number of days after a password expires before an account is locked.

warndays Number of days prior to maxdays to warn users.

expire Date that account expires, represented in days since epoch (January 1, 1970).

The below parameters apply to windows only:

win_homedrive (Windows Only) The drive letter to use for the home directory. If not specified the home directory will be a unc path. Otherwise the home directory will be mapped to the specified drive. Must be a letter followed by a colon. Because of the colon, the value must be surrounded by single quotes. ie: - win_homedrive: 'U:

Changed in version 2015.8.0.

win_profile (Windows Only) The custom profile directory of the user. Uses default value of underlying system if not set.

Changed in version 2015.8.0.

win_logonscript (Windows Only) The full path to the logon script to run when the user logs in.

Changed in version 2015.8.0.

win_description (Windows Only) A brief description of the purpose of the users account.

Changed in version 2015.8.0.

21.19.281 salt.states.vagrant

Manage Vagrant VMs

Manange execution of Vagrant virtual machines on Salt minions.

Vagrant is a tool for building and managing virtual machine environments. It can use various providers, such as VirtualBox, Docker, or VMware, to run its VMs. Vagrant provides some of the functionality of a light-weight hypervisor. The combination of Salt modules, Vagrant running on the host, and a virtual machine provider, gives hypervisor-like functionality for developers who use Vagrant to quickly define their virtual environments.

New in version 2018.3.0.

The configuration of each virtual machine is defined in a file named `Vagrantfile` which must exist on the VM host machine. The essential parameters which must be defined to start a Vagrant VM are the directory where the `Vagrantfile` is located (argument `cwd:`), and the username which will own the `Vagrant box` created for the VM (argument `vagrant_runas:`).

A single `Vagrantfile` may define one or more virtual machines. Use the `machine` argument to chose among them. The default (blank) value will select the `primary` (or only) machine in the Vagrantfile.

[NOTE:] Each virtual machine host must have the following:

- a working salt-minion
- a Salt sdb database configured for `vagrant_sdb_data`.
- Vagrant installed and the `vagrant` command working
- a suitable VM provider

```
# EXAMPLE:
# file /etc/salt/minion.d/vagrant_sdb.conf on the host computer
#  -- this sdb database is required by the Vagrant module --
vagrant_sdb_data:  # The sdb database must have this name.
  driver: sqlite3  # Let's use SQLite to store the data ...
  database: /var/cache/salt/vagrant.sqlite  # ... in this file ...
  table: sdb  # ... using this table name.
  create_table: True  # if not present
```

`salt.states.vagrant.`**`destroyed`**`(`*name*`)`

Stops a VM (or VMs) and removes all references to it (them). (Runs `vagrant destroy`.)

Subsequent re-use of the same machine will requere another operation of `vagrant.running` or a call to the `vagrant.init` execution module.

Parameters **`name`** – May be a Salt_id node or a POSIX-style wildcard string.

```
node_name:
  vagrant.destroyed
```

`salt.states.vagrant.`**`initialized`**`(`*name*, ***kwargs*`)`

Defines a new VM with specified arguments, but does not start it.

Parameters **`name`** – the Salt_id node name you wish your VM to have.

Each machine must be initialized individually using this function or the "vagrant.running" function, or the vagrant.init execution module call.

This command will not change the state of a running or paused machine.

Possible keyword arguments:

- cwd: The directory (path) containing the Vagrantfile
- machine: (") the name of the machine (in the Vagrantfile) if not default
- vagrant_runas: ('root') the username who owns the vagrantbox file
- vagrant_provider: the provider to run the VM (usually 'virtualbox')
- vm: ({}) a dictionary containing these or other keyword arguments

```
node_name1:
  vagrant.initialized
    - cwd: /projects/my_project
    - vagrant_runas: my_username
    - machine: machine1

node_name2:
  vagrant.initialized
    - cwd: /projects/my_project
    - vagrant_runas: my_username
    - machine: machine2

start_nodes:
  vagrant.start:
    - name: node_name?
```

`salt.states.vagrant.`**`paused`**`(`*name*`)`

Stores the state of a VM (or VMs) for fast restart. (Runs `vagrant suspend`.)

Parameters **`name`** – May be a Salt_id node or a POSIX-style wildcard string.

```
node_name:
  vagrant.paused
```

`salt.states.vagrant.`**`powered_off`**`(`*name*`)`

Stops a VM (or VMs) by power off. (Runs `vagrant halt`.)

This method is provided for compatibility with other VM-control state modules. For Vagrant, the action is identical with `stopped`.

Parameters **`name`** – May be a Salt_id node or a POSIX-style wildcard string.

```
node_name:
  vagrant.unpowered
```

`salt.states.vagrant.`**`rebooted`**`(`*name*`)`

Reboots a running, paused, or stopped VM (or VMs). (Runs `vagrant reload`.)

The will re-run the provisioning

Parameters **`name`** – May be a Salt_id node or a POSIX-style wildcard string.

```
node_name:
  vagrant.reloaded
```

`salt.states.vagrant.`**`running`**`(`*name, **kwargs*`)`

Defines and starts a new VM with specified arguments, or restart a VM (or group of VMs). (Runs `vagrant up`.)

Parameters **`name`** – the Salt_id node name you wish your VM to have.

If `name` contains a "?" or "*" then it will re-start a group of VMs which have been paused or stopped.

Each machine must be initially started individually using this function or the vagrant.init execution module call.

[NOTE:] Keyword arguments are silently ignored when re-starting an existing VM.

Possible keyword arguments:

- cwd: The directory (path) containing the Vagrantfile
- machine: (") the name of the machine (in the Vagrantfile) if not default
- vagrant_runas: ('root') the username who owns the vagrantbox file
- vagrant_provider: the provider to run the VM (usually 'virtualbox')
- vm: ({}) a dictionary containing these or other keyword arguments

```
node_name:
  vagrant.running
```

```
node_name:
  vagrant.running:
    - cwd: /projects/my_project
    - vagrant_runas: my_username
    - machine: machine1
```

`salt.states.vagrant.`**`stopped`**`(`*name*`)`

Stops a VM (or VMs) by shutting it (them) down nicely. (Runs `vagrant halt`)

Parameters **`name`** – May be a Salt_id node, or a POSIX-style wildcard string.

```
node_name:
  vagrant.stopped
```

21.19.282 salt.states.vault module

States for managing Hashicorp Vault. Currently handles policies. Configuration instructions are documented in the execution module docs.

maintainer SaltStack

maturity new

platform all

New in version 2017.7.0.

`salt.states.vault.`**`policy_present`**`(`*name*, *rules*`)`

Ensure a Vault policy with the given name and rules is present.

name The name of the policy

rules Rules formatted as in-line HCL

```
demo-policy:
  vault.policy_present:
    - name: foo/bar
    - rules: |
      path "secret/top-secret/*" {
        policy = "deny"
      }
      path "secret/not-very-secret/*" {
        policy = "write"
      }
```

21.19.283 salt.states.vbox_guest

VirtualBox Guest Additions installer state

`salt.states.vbox_guest.`**`additions_installed`**`(`*name*, *reboot=False*, *upgrade_os=False*`)`

Ensure that the VirtualBox Guest Additions are installed. Uses the CD, connected by VirtualBox.

name The name has no functional value and is only used as a tracking reference.

reboot [False] Restart OS to complete installation.

upgrade_os [False] Upgrade OS (to ensure the latests version of kernel and developer tools installed).

`salt.states.vbox_guest.`**`additions_removed`**`(`*name*, *force=False*`)`

Ensure that the VirtualBox Guest Additions are removed. Uses the CD, connected by VirtualBox.

To connect VirtualBox Guest Additions via VirtualBox graphical interface press 'Host+D' ('Host' is usually 'Right Ctrl').

name The name has no functional value and is only used as a tracking reference.

force Force VirtualBox Guest Additions removing.

`salt.states.vbox_guest.`**`grant_access_to_shared_folders_to`**`(`*name*, *users=None*`)`

Grant access to auto-mounted shared folders to the users.

User is specified by it's name. To grant access for several users use argument *users*.

name Name of the user to grant access to auto-mounted shared folders to.

users List of names of users to grant access to auto-mounted shared folders to. If specified, *name* will not be taken into account.

21.19.284 salt.states.victorops

Create an Event in VictorOps

New in version 2015.8.0.

This state is useful for creating events on the VictorOps service during state runs.

```
webserver-warning-message:
  victorops.create_event:
    - message_type: 'CRITICAL'
    - entity_id: 'webserver/diskspace'
    - state_message: 'Webserver diskspace is low.'
```

`salt.states.victorops.`**`create_event`**`(`*name*, *message_type*, *routing_key='everyone'*, ***kwargs*`)`

Create an event on the VictorOps service

```
webserver-warning-message:
  victorops.create_event:
    - message_type: 'CRITICAL'
    - entity_id: 'webserver/diskspace'
    - state_message: 'Webserver diskspace is low.'

database-server-warning-message:
  victorops.create_event:
    - message_type: 'WARNING'
    - entity_id: 'db_server/load'
    - state_message: 'Database Server load is high.'
    - entity_is_host: True
    - entity_display_name: 'dbdserver.example.com'
```

The following parameters are required:

name This is a short description of the event.

message_type One of the following values: INFO, WARNING, ACKNOWLEDGEMENT, CRITICAL, RECOVERY.

The following parameters are optional:

routing_key The key for where messages should be routed. By default, sent to 'everyone' route.

entity_id The name of alerting entity. If not provided, a random name will be assigned.

timestamp Timestamp of the alert in seconds since epoch. Defaults to the time the alert is received at VictorOps.

timestamp_fmt The date format for the timestamp parameter. Defaults to "%Y-%m-%dT%H:%M:%S'.

state_start_time The time this entity entered its current state (seconds since epoch). Defaults to the time alert is received.

state_start_time_fmt The date format for the timestamp parameter. Defaults to '%Y-%m-%dT%H:%M:%S'.

state_message Any additional status information from the alert item.

entity_is_host Used within VictorOps to select the appropriate display format for the incident.

entity_display_name Used within VictorOps to display a human-readable name for the entity.

ack_message A user entered comment for the acknowledgment.

ack_author The user that acknowledged the incident.

21.19.285 salt.states.virt module

Manage virt

For the key certificate this state uses the external pillar in the master to call for the generation and signing of certificates for systems running libvirt:

```
libvirt_keys:
  virt.keys
```

`salt.states.virt.`**`keys`**`(`*name*, *basepath='/etc/pki'*, ***kwargs*`)`

Manage libvirt keys.

name The name variable used to track the execution

basepath Defaults to `/etc/pki`, this is the root location used for libvirt keys on the hypervisor

The following parameters are optional:

country The country that the certificate should use. Defaults to US.

New in version 2018.3.0.

state The state that the certificate should use. Defaults to Utah.

New in version 2018.3.0.

locality The locality that the certificate should use. Defaults to Salt Lake City.

New in version 2018.3.0.

organization The organization that the certificate should use. Defaults to Salted.

New in version 2018.3.0.

expiration_days The number of days that the certificate should be valid for. Defaults to 365 days (1 year)

New in version 2018.3.0.

`salt.states.virt.`**`network_running`**`(`*name*, *bridge*, *forward*, *vport=None*, *tag=None*, *ipv4_config=None*, *ipv6_config=None*, *autostart=True*, *connection=None*, *username=None*, *password=None*`)`

Defines and starts a new network with specified arguments.

Parameters

- **`bridge`** – Bridge name
- **`forward`** – Forward mode(bridge, router, nat)
- **`vport`** – Virtualport type (Default: `'None'`)
- **`tag`** – Vlan tag (Default: `'None'`)
- **`ipv4_config`** – IPv4 network configuration. See the :py:func‘virt.network_define <salt.modules.virt.network_define>‘ function corresponding parameter documentation for more details on this dictionary. (Default: None).

  New in version 3000.
- **`ipv6_config`** – IPv6 network configuration. See the :py:func‘virt.network_define <salt.modules.virt.network_define>‘ function corresponding parameter documentation for more details on this dictionary. (Default: None).

  New in version 3000.
- **`autostart`** – Network autostart (default `'True'`)
- **`connection`** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.
- **`username`** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **`password`** – password to connect with, overriding defaults

  New in version 2019.2.0.

```
domain_name:
  virt.network_define
```

```
network_name:
  virt.network_define:
    - bridge: main
    - forward: bridge
    - vport: openvswitch
    - tag: 180
    - autostart: True
```

```
network_name:
  virt.network_define:
    - bridge: natted
    - forward: nat
    - ipv4_config:
        cidr: 192.168.42.0/24
        dhcp_ranges:
          - start: 192.168.42.10
            end: 192.168.42.25
          - start: 192.168.42.100
            end: 192.168.42.150
    - autostart: True
```

`salt.states.virt.`**`pool_deleted`**(*name, purge=False, connection=None, username=None, password=None*)

Deletes a virtual storage pool.

Parameters

- **`name`** – the name of the pool to delete.
- **`purge`** – if `True`, the volumes contained in the pool will be deleted as well as the pool itself. Note that these will be lost for ever. If `False` the pool will simply be undefined. (Default: `False`)
- **`connection`** – libvirt connection URI, overriding defaults
- **`username`** – username to connect with, overriding defaults
- **`password`** – password to connect with, overriding defaults

In order to be purged a storage pool needs to be running to get the list of volumes to delete.

Some libvirt storage drivers may not implement deleting, those actions are implemented on a best effort idea. In any case check the result's comment property to see if any of the action was unsupported.

New in version 3000.

`salt.states.virt.`**`pool_running`**(*name, ptype=None, target=None, permissions=None, source=None, transient=False, autostart=True, connection=None, username=None, password=None*)

Defines and starts a new pool with specified arguments.

New in version 2019.2.0.

Parameters

- **`ptype`** – libvirt pool type
- **`target`** – full path to the target device or folder. (Default: `None`)
- **`permissions`** – target permissions. See *Permissions definition* for more details on this structure.
- **`source`** – dictionary containing keys matching the `source_*` parameters in function *`salt.modules.virt.pool_define()`*.
- **`transient`** – when set to `True`, the pool will be automatically undefined after being stopped. (Default: `False`)
- **`autostart`** – Whether to start the pool when booting the host. (Default: `True`)
- **`start`** – When `True`, define and start the pool, otherwise the pool will be left stopped.
- **`connection`** – libvirt connection URI, overriding defaults
- **`username`** – username to connect with, overriding defaults
- **`password`** – password to connect with, overriding defaults

```
pool_name:
  virt.pool_define
```

```
pool_name:
  virt.pool_define:
    - ptype: netfs
    - target: /mnt/cifs
    - permissions:
        - mode: 0770
        - owner: 1000
        - group: 100
    - source:
        dir: samba_share
        hosts:
          - one.example.com
          - two.example.com
        format: cifs
    - autostart: True
```

`salt.states.virt.`**`powered_off`**`(`*name, connection=None, username=None, password=None*`)`

Stops a VM by power off.

New in version 2016.3.0.

Parameters

- **`connection`** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.
- **`username`** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **`password`** – password to connect with, overriding defaults

  New in version 2019.2.0.

```
domain_name:
  virt.stopped
```

`salt.states.virt.`**`rebooted`**`(`*name, connection=None, username=None, password=None*`)`

Reboots VMs

New in version 2016.3.0.

Parameters

- **`name`** –
- **`connection`** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.
- **`username`** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **`password`** – password to connect with, overriding defaults

  New in version 2019.2.0.

`salt.states.virt.`**`reverted`**`(`*name, snapshot=None, cleanup=False*`)`

Deprecated since version 2016.3.0.

Reverts to the particular snapshot.

New in version 2016.3.0.

```
domain_name:
  virt.reverted:
    - cleanup: True
```

(continues on next page)

(continued from previous page)

```
domain_name_1:
  virt.reverted:
    - snapshot: snapshot_name
    - cleanup: False
```

`salt.states.virt.`**`running`**`(`*name, cpu=None, mem=None, image=None, vm_type=None, disk_profile=None, disks=None, nic_profile=None, interfaces=None, graphics=None, seed=True, install=True, pub_key=None, priv_key=None, update=False, connection=None, username=None, password=None, os_type=None, arch=None, boot=None*`)`

Starts an existing guest, or defines and starts a new VM with specified arguments.

New in version 2016.3.0.

Parameters

- **`name`** – name of the virtual machine to run
- **`cpu`** – number of CPUs for the virtual machine to create
- **`mem`** – amount of memory in MiB for the new virtual machine
- **`image`** – disk image to use for the first disk of the new VM

  Deprecated since version 2019.2.0.
- **`vm_type`** – force virtual machine type for the new VM. The default value is taken from the host capabilities. This could be useful for example to use `'qemu'` type instead of the `'kvm'` one.

  New in version 2019.2.0.
- **`disk_profile`** – Name of the disk profile to use for the new virtual machine

  New in version 2019.2.0.
- **`disks`** – List of disk to create for the new virtual machine. See *Disks Definitions* for more details on the items on this list.

  New in version 2019.2.0.
- **`nic_profile`** – Name of the network interfaces profile to use for the new virtual machine

  New in version 2019.2.0.
- **`interfaces`** – List of network interfaces to create for the new virtual machine. See *Network Interfaces Definitions* for more details on the items on this list.

  New in version 2019.2.0.
- **`graphics`** – Graphics device to create for the new virtual machine. See *Graphics Definition* for more details on this dictionary

  New in version 2019.2.0.
- **`saltenv`** – Fileserver environment (Default: `'base'`). See *`cp module for more details`*

  New in version 2019.2.0.
- **`seed`** –

  `True` to seed the disk image. Only used when the `image` parameter is provided.

  (Default: `True`)

  New in version 2019.2.0.
- **`install`** – install salt minion if absent (Default: `True`)

  New in version 2019.2.0.
- **`pub_key`** – public key to seed with (Default: `None`)

  New in version 2019.2.0.

- **`priv_key`** – public key to seed with (Default: `None`)

  New in version 2019.2.0.
- **`seed_cmd`** – Salt command to execute to seed the image. (Default: `'seed.apply'`)

  New in version 2019.2.0.
- **`update`** – set to `True` to update a defined module. (Default: `False`)

  New in version 2019.2.0.
- **`connection`** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.
- **`username`** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **`password`** – password to connect with, overriding defaults

  New in version 2019.2.0.
- **`os_type`** – type of virtualization as found in the `//os/type` element of the libvirt definition. The default value is taken from the host capabilities, with a preference for `hvm`. Only used when creating a new virtual machine.

  New in version 3000.
- **`arch`** – architecture of the virtual machine. The default value is taken from the host capabilities, but `x86_64` is prefed over `i686`. Only used when creating a new virtual machine.

  New in version 3000.
- **`boot`** – Specifies kernel for the virtual machine, as well as boot parameters for the virtual machine. This is an optionl parameter, and all of the keys are optional within the dictionary. If a remote path is provided to kernel or initrd, salt will handle the downloading of the specified remote fild, and will modify the XML accordingly.

  ```
  {
      'kernel': '/root/f8-i386-vmlinuz',
      'initrd': '/root/f8-i386-initrd',
      'cmdline': 'console=ttyS0 ks=http://example.com/f8-i386/os/
  ↪'
  }
  ```

  New in version 3000.

Example States

Make sure an already-defined virtual machine called `domain_name` is running:

```
domain_name:
  virt.running
```

Do the same, but define the virtual machine if needed:

```
domain_name:
  virt.running:
    - cpu: 2
    - mem: 2048
    - disk_profile: prod
    - disks:
```

(continues on next page)

(continued from previous page)

```
      - name: system
        size: 8192
        overlay_image: True
        pool: default
        image: /path/to/image.qcow2
      - name: data
        size: 16834
    - nic_profile: prod
    - interfaces:
      - name: eth0
        mac: 01:23:45:67:89:AB
      - name: eth1
        type: network
        source: admin
    - graphics:
        type: spice
        listen:
            type: address
            address: 192.168.0.125
```

`salt.states.virt.`**`saved`**`(`*name*, *suffix=None*`)`

Deprecated since version 2016.3.0: Use `snapshot()` instead.

Takes a snapshot of a particular VM or by a UNIX-style wildcard.

New in version 2016.3.0.

```
domain_name:
  virt.saved:
    - suffix: periodic

domain*:
  virt.saved:
    - suffix: periodic
```

`salt.states.virt.`**`snapshot`**`(`*name*, *suffix=None*, *connection=None*, *username=None*, *password=None*`)`

Takes a snapshot of a particular VM or by a UNIX-style wildcard.

New in version 2016.3.0.

Parameters

- **connection** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.
- **username** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **password** – password to connect with, overriding defaults

  New in version 2019.2.0.

```
domain_name:
  virt.snapshot:
    - suffix: periodic

domain*:
  virt.snapshot:
    - suffix: periodic
```

`salt.states.virt.`**`stopped`**`(`*name*, *connection=None*, *username=None*, *password=None*`)`

Stops a VM by shutting it down nicely.

New in version 2016.3.0.

Parameters

- **`connection`** – libvirt connection URI, overriding defaults

  New in version 2019.2.0.
- **`username`** – username to connect with, overriding defaults

  New in version 2019.2.0.
- **`password`** – password to connect with, overriding defaults

  New in version 2019.2.0.

```
domain_name:
  virt.stopped
```

`salt.states.virt.`**`unpowered`**`(`*name*`)`

Deprecated since version 2016.3.0: Use `powered_off()` instead.

Stops a VM by power off.

New in version 2016.3.0.

```
domain_name:
  virt.stopped
```

21.19.286 salt.states.virtualenv

Setup of Python virtualenv sandboxes.

New in version 0.17.0.

`salt.states.virtualenv_mod.`**`managed`**`(`*name*, *venv_bin=None*, *requirements=None*, *system_site_packages=False*, *distribute=False*, *use_wheel=False*, *clear=False*, *python=None*, *extra_search_dir=None*, *never_download=None*, *prompt=None*, *user=None*, *cwd=None*, *index_url=None*, *extra_index_url=None*, *pre_releases=False*, *no_deps=False*, *pip_download=None*, *pip_download_cache=None*, *pip_exists_action=None*, *pip_ignore_installed=False*, *proxy=None*, *use_vt=False*, *env_vars=None*, *no_use_wheel=False*, *pip_upgrade=False*, *pip_pkgs=None*, *pip_no_cache_dir=False*, *pip_cache_dir=None*, *process_dependency_links=False*, *no_binary=None*, ***kwargs*`)`

Create a virtualenv and optionally manage it with pip

name Path to the virtualenv.

venv_bin: virtualenv The name (and optionally path) of the virtualenv command. This can also be set globally in the minion config file as `virtualenv.venv_bin`.

requirements: None Path to a pip requirements file. If the path begins with `salt://` the file will be transferred from the master file server.

use_wheel: False Prefer wheel archives (requires pip >= 1.4).

python [None] Python executable used to build the virtualenv

user: None The user under which to run virtualenv and pip.

cwd: None Path to the working directory where *pip install* is executed.

no_deps: False Pass *–no-deps* to *pip install.*

pip_exists_action: None Default action of pip when a path already exists: (s)witch, (i)gnore, (w)ipe, (b)ackup.

proxy: None Proxy address which is passed to *pip install*.

env_vars: None Set environment variables that some builds will depend on. For example, a Python C-module may have a Makefile that needs INCLUDE_PATH set to pick up a header file while compiling.

no_use_wheel: False Force to not use wheel archives (requires pip>=1.4)

no_binary Force to not use binary packages (requires pip >= 7.0.0) Accepts either :all: to disable all binary packages, :none: to empty the set, or a list of one or more packages

pip_upgrade: False Pass *–upgrade* to *pip install*.

pip_pkgs: None As an alternative to *requirements*, pass a list of pip packages that should be installed.

process_dependency_links: False Run pip install with the –process_dependency_links flag.

New in version 2017.7.0.

Also accepts any kwargs that the virtualenv module will. However, some kwargs, such as the `pip` option, require `- distribute: True`.

```
/var/www/myvirtualenv.com:
  virtualenv.managed:
    - system_site_packages: False
    - requirements: salt://REQUIREMENTS.txt
    - env_vars:
        PATH_VAR: '/usr/local/bin/'
```

21.19.287 salt.states.webutil

Support for htpasswd module. Requires the apache2-utils package for Debian-based distros.

New in version 2014.7.0.

```
username:
  webutil.user_exists:
    - password: secr3t
    - htpasswd_file: /etc/nginx/htpasswd
    - options: d
    - force: true
```

`salt.states.webutil.`**`user_absent`**(*name*, *htpasswd_file=None*, *runas=None*)

Make sure the user is not in the specified htpasswd file

name User name

htpasswd_file Path to the htpasswd file

runas The system user to run htpasswd command with

`salt.states.webutil.`**`user_exists`**(*name*, *password=None*, *htpasswd_file=None*, *options=''*, *force=False*, *runas=None*, *update=False*)

Make sure the user is inside the specified htpasswd file

name User name

password User password

htpasswd_file Path to the htpasswd file

options See `salt.modules.htpasswd.useradd`

force Touch the file even if user already created

runas The system user to run htpasswd command with

update Update an existing user's password if it's different from what's in the htpasswd file (unlike force, which updates regardless)

21.19.288 salt.states.win_certutil module

Installing of certificates to the Windows Certificate Manager

Install certificates to the Windows Certificate Manager

```
salt://certs/cert.cer:
  certutil.add_store:
    - store: TrustedPublisher
```

`salt.states.win_certutil.`**`add_store`**`(`*name*, *store*, *saltenv='base'*`)`

Store a certificate to the given store

name The certificate to store, this can use local paths or salt:// paths

store The store to add the certificate to

saltenv The salt environment to use, this is ignored if a local path is specified

`salt.states.win_certutil.`**`del_store`**`(`*name*, *store*, *saltenv='base'*`)`

Remove a certificate in the given store

name The certificate to remove, this can use local paths or salt:// paths

store The store to remove the certificate from

saltenv The salt environment to use, this is ignored if a local path is specified

21.19.289 salt.states.win_dacl

Windows Object Access Control Lists

Ensure an ACL is present

parameters: name - the path of the object objectType - Registry/File/Directory user - user account or SID for the ace permission - permission for the ace (see module win_acl for available permissions for each objectType) acetype - Allow/Deny propagation - how the ACL should apply to child objects (see module win_acl for available propagation types)

```
addAcl:
  win_dacl.present:
    - name: HKEY_LOCAL_MACHINE\SOFTWARE\mykey
    - objectType: Registry
    - user: FakeUser
    - permission: FullControl
    - acetype: ALLOW
    - propagation: KEY&SUBKEYS
```

Ensure an ACL does not exist

parameters: name - the path of the object objectType - Registry/File/Directory user - user account or SID for the ace permission - permission for the ace (see module win_acl for available permissions for each objectType) acetype - Allow/Deny propagation - how the ACL should apply to child objects (see module win_acl for available propagation types)

```
removeAcl:
  win_dacl.absent:
    - name: HKEY_LOCAL_MACHINE\SOFTWARE\mykey
    - objectType: Registry
    - user: FakeUser
    - permission: FullControl
    - acetype: ALLOW
    - propagation: KEY&SUBKEYS
```

Ensure an object is inheriting permissions

parameters: name - the path of the object objectType - Registry/File/Directory clear_existing_acl - True/False - when inheritance is enabled, should the existing ACL be kept or cleared out

```
eInherit:
  win_dacl.enableinheritance:
    - name: HKEY_LOCAL_MACHINE\SOFTWARE\mykey
    - objectType: Registry
    - clear_existing_acl: True
```

Ensure an object is not inheriting permissions

parameters: name - the path of the object objectType - Registry/File/Directory copy_inherited_acl - True/False - if inheritance is enabled, should the inherited permissions be copied to the ACL when inheritance is disabled

```
dInherit:
  win_dacl.disableinheritance:
    - name: HKEY_LOCAL_MACHINE\SOFTWARE\mykey
    - objectType: Registry
    - copy_inherited_acl: False
```

`salt.states.win_dacl.`**`absent`**(*name*, *objectType*, *user*, *permission*, *acetype*, *propagation*)
Ensure an ACL does not exist

`salt.states.win_dacl.`**`disinherit`**(*name*, *objectType*, *copy_inherited_acl=True*)
Ensure an object is not inheriting ACLs from its parent

`salt.states.win_dacl.`**`inherit`**(*name*, *objectType*, *clear_existing_acl=False*)
Ensure an object is inheriting ACLs from its parent

`salt.states.win_dacl.`**`present`**(*name*, *objectType*, *user*, *permission*, *acetype*, *propagation*)
Ensure an ACE is present

21.19.290 salt.states.win_dism module

Installing of Windows features using DISM

Install windows features/capabilties with DISM

```
Language.Basic~~~en-US~0.0.1.0:
  dism.capability_installed

NetFx3:
  dism.feature_installed
```

`salt.states.win_dism.`**`capability_installed`**(*name*, *source=None*, *limit_access=False*, *image=None*, *restart=False*)
Install a DISM capability

Parameters

- **name** (*str*) – The capability to install
- **source** (*str*) – The optional source of the capability
- **limit_access** (*bool*) – Prevent DISM from contacting Windows Update for online images
- **image** (*Optional[str]*) – The path to the root directory of an offline Windows image. If *None* is passed, the running operating system is targeted. Default is None.
- **restart** (*Optional[bool]*) – Reboot the machine if required by the install

Example

Run `dism.available_capabilities` to get a list of available capabilities. This will help you get the proper name to use.

```
install_dotnet35:
  dism.capability_installed:
    - name: NetFX3~~~~
```

`salt.states.win_dism.`**`capability_removed`**(*name*, *image=None*, *restart=False*)

Uninstall a DISM capability

Parameters

- **name** (*str*) – The capability to uninstall
- **image** (*Optional[str]*) – The path to the root directory of an offline Windows image. If *None* is passed, the running operating system is targeted. Default is None.
- **restart** (*Optional[bool]*) – Reboot the machine if required by the install

Example

Run `dism.installed_capabilities` to get a list of installed capabilities. This will help you get the proper name to use.

```
remove_dotnet35:
  dism.capability_removed:
    - name: NetFX3~~~~
```

`salt.states.win_dism.`**`feature_installed`**(*name*, *package=None*, *source=None*, *limit_access=False*, *enable_parent=False*, *image=None*, *restart=False*)

Install a DISM feature

Parameters

- **name** (*str*) – The feature in which to install
- **package** (*Optional[str]*) – The parent package for the feature. You do not have to specify the package if it is the Windows Foundation Package. Otherwise, use package to specify the parent package of the feature
- **source** (*str*) – The optional source of the feature
- **limit_access** (*bool*) – Prevent DISM from contacting Windows Update for online images
- **enable_parent** (*Optional[bool]*) – True will enable all parent features of the specified feature
- **image** (*Optional[str]*) – The path to the root directory of an offline Windows image. If *None* is passed, the running operating system is targeted. Default is None.
- **restart** (*Optional[bool]*) – Reboot the machine if required by the install

Example

Run `dism.available_features` to get a list of available features. This will help you get the proper name to use.

```
install_telnet_client:
  dism.feature_installed:
    - name: TelnetClient
```

`salt.states.win_dism.`**`feature_removed`**`(`*name, remove_payload=False, image=None, restart=False*`)`

Disables a feature.

Parameters

- **`name`** (*`str`*) – The feature to disable
- **`remove_payload`** (*`Optional[bool]`*) – Remove the feature's payload. Must supply source when enabling in the future.
- **`image`** (*`Optional[str]`*) – The path to the root directory of an offline Windows image. If *None* is passed, the running operating system is targeted. Default is None.
- **`restart`** (*`Optional[bool]`*) – Reboot the machine if required by the install

Example

Run `dism.installed_features` to get a list of installed features. This will help you get the proper name to use.

```
remove_telnet_client:
  dism.feature_removed:
    - name: TelnetClient
    - remove_payload: True
```

`salt.states.win_dism.`**`package_installed`**`(`*name, ignore_check=False, prevent_pending=False, image=None, restart=False*`)`

Install a package.

Parameters

- **`name`** (*`str`*) – The package to install. Can be a .cab file, a .msu file, or a folder
- **`ignore_check`** (*`Optional[bool]`*) – Skip installation of the package if the applicability checks fail
- **`prevent_pending`** (*`Optional[bool]`*) – Skip the installation of the package if there are pending online actions
- **`image`** (*`Optional[str]`*) – The path to the root directory of an offline Windows image. If *None* is passed, the running operating system is targeted. Default is None.
- **`restart`** (*`Optional[bool]`*) – Reboot the machine if required by the install

Example

```
install_KB123123123:
  dism.package_installed:
    - name: C:\Packages\KB123123123.cab
```

`salt.states.win_dism.`**`package_removed`**`(`*name, image=None, restart=False*`)`

Uninstall a package

Parameters

- **`name`** (*`str`*) – The full path to the package. Can be either a .cab file or a folder. Should point to the original source of the package, not to where the file is installed. This can also be the name of a package as listed in `dism.installed_packages`
- **`image`** (*`Optional[str]`*) – The path to the root directory of an offline Windows image. If *None* is passed, the running operating system is targeted. Default is None.
- **`restart`** (*`Optional[bool]`*) – Reboot the machine if required by the install

Example

```
# Example using source
remove_KB1231231:
  dism.package_installed:
    - name: C:\Packages\KB1231231.cab

# Example using name from ``dism.installed_packages``
remove_KB1231231:
  dism.package_installed:
    - name: Package_for_KB1231231~31bf3856ad364e35~amd64~~10.0.1.3
```

21.19.291 salt.states.win_dns_client

Module for configuring DNS Client on Windows systems

`salt.states.win_dns_client.`**`dns_dhcp`**(*name*, *interface='Local Area Connection'*)
Configure the DNS server list from DHCP Server

`salt.states.win_dns_client.`**`dns_exists`**(*name*, *servers=None*, *interface='Local Area Connection'*, *replace=False*)
Configure the DNS server list in the specified interface

Example:

```
config_dns_servers:
  win_dns_client.dns_exists:
    - replace: True #remove any servers not in the "servers" list, default is␣
↪False
    - servers:
      - 8.8.8.8
      - 8.8.8.9
```

`salt.states.win_dns_client.`**`primary_suffix`**(*name*, *suffix=None*, *updates=False*)
New in version 2014.7.0.

Configure the global primary DNS suffix of a DHCP client.

suffix [None] The suffix which is advertised for this client when acquiring a DHCP lease When none is set, the explicitly configured DNS suffix will be removed.

updates [False] Allow syncing the DNS suffix with the AD domain when the client's AD domain membership changes

```
primary_dns_suffix:
    win_dns_client.primary_suffix:
        - suffix: sub.domain.tld
        - updates: True
```

21.19.292 salt.states.win_firewall

State for configuring Windows Firewall

`salt.states.win_firewall.`**`add_rule`**(*name*, *localport*, *protocol='tcp'*, *action='allow'*, *dir='in'*, *remoteip='any'*)
Add a new inbound or outbound rule to the firewall policy

Parameters

- **name** (*str*) – The name of the rule. Must be unique and cannot be "all". Required.

- **localport** (*int*) – The port the rule applies to. Must be a number between 0 and 65535. Can be a range. Can specify multiple ports separated by commas. Required.
- **protocol** (*Optional[str]*) – The protocol. Can be any of the following:
  - A number between 0 and 255
  - icmpv4
  - icmpv6
  - tcp
  - udp
  - any
- **action** (*Optional[str]*) – The action the rule performs. Can be any of the following:
  - allow
  - block
  - bypass
- **dir** (*Optional[str]*) – The direction. Can be `in` or `out`.
- **remoteip** (*Optional [str]*) – The remote IP. Can be any of the following:
  - any
  - localsubnet
  - dns
  - dhcp
  - wins
  - defaultgateway
  - Any valid IPv4 address (192.168.0.12)
  - Any valid IPv6 address (2002:9b3b:1a31:4:208:74ff:fe39:6c43)
  - Any valid subnet (192.168.1.0/24)
  - Any valid range of IP addresses (192.168.0.1-192.168.0.12)
  - A list of valid IP addresses

  Can be combinations of the above separated by commas.

  New in version 2016.11.6.

Example:

```
open_smb_port:
  win_firewall.add_rule:
    - name: SMB (445)
    - localport: 445
    - protocol: tcp
    - action: allow
```

`salt.states.win_firewall.`**`disabled`**(*name='allprofiles'*)

Disable all the firewall profiles (Windows only)

Parameters **profile** (*Optional[str]*) – The name of the profile to disable. Default is `allprofiles`. Valid options are:

- allprofiles
- domainprofile
- privateprofile
- publicprofile

Example:

```
# To disable the domain profile
disable_domain:
  win_firewall.disabled:
    - name: domainprofile

# To disable all profiles
```

(continues on next page)

(continued from previous page)

```
disable_all:
  win_firewall.disabled:
    - name: allprofiles
```

`salt.states.win_firewall.`**`enabled`**(*name='allprofiles'*)

Enable all the firewall profiles (Windows only)

Parameters **`profile`** (`Optional[str]`) – The name of the profile to enable. Default is `allprofiles`. Valid options are:

- allprofiles
- domainprofile
- privateprofile
- publicprofile

Example:

```
# To enable the domain profile
enable_domain:
  win_firewall.enabled:
    - name: domainprofile

# To enable all profiles
enable_all:
  win_firewall.enabled:
    - name: allprofiles
```

21.19.293 salt.states.win_iis module

Microsoft IIS site management

This module provides the ability to add/remove websites and application pools from Microsoft IIS.

New in version 2016.3.0.

`salt.states.win_iis.`**`container_setting`**(*name*, *container*, *settings=None*)

Set the value of the setting for an IIS container.

Parameters

- **`name`** (`str`) – The name of the IIS container.
- **`container`** (`str`) – The type of IIS container. The container types are: AppPools, Sites, SslBindings
- **`settings`** (`str`) – A dictionary of the setting names and their values. Example of usage for the `AppPools` container:

```
site0-apppool-setting:
    win_iis.container_setting:
        - name: site0
        - container: AppPools
        - settings:
            managedPipelineMode: Integrated
            processModel.maxProcesses: 1
            processModel.userName: TestUser
            processModel.password: TestPassword
            processModel.identityType: SpecificUser
```

Example of usage for the `Sites` container:

```
site0-site-setting:
    win_iis.container_setting:
        - name: site0
        - container: Sites
        - settings:
            logFile.logFormat: W3C
            logFile.period: Daily
            limits.maxUrlSegments: 32
```

`salt.states.win_iis.`**`create_app`**(*name, site, sourcepath, apppool=None*)

Create an IIS application.

Parameters

- **name** (*str*) – The IIS application.
- **site** (*str*) – The IIS site name.
- **sourcepath** (*str*) – The physical path.
- **apppool** (*str*) – The name of the IIS application pool.

Example of usage with only the required arguments:

```
site0-v1-app:
    win_iis.create_app:
        - name: v1
        - site: site0
        - sourcepath: C:\inetpub\site0\v1
```

Example of usage specifying all available arguments:

```
site0-v1-app:
    win_iis.create_app:
        - name: v1
        - site: site0
        - sourcepath: C:\inetpub\site0\v1
        - apppool: site0
```

`salt.states.win_iis.`**`create_apppool`**(*name*)

Create an IIS application pool.

Parameters **name** (*str*) – The name of the IIS application pool.

Usage:

```
site0-apppool:
    win_iis.create_apppool:
        - name: site0
```

`salt.states.win_iis.`**`create_binding`**(*name, site, hostheader='', ipaddress='*', port=80, protocol='http', sslflags=0*)

Create an IIS binding.

Parameters

- **site** (*str*) – The IIS site name.
- **hostheader** (*str*) – The host header of the binding.
- **ipaddress** (*str*) – The IP address of the binding.
- **port** (*str*) – The TCP port of the binding.
- **protocol** (*str*) – The application protocol of the binding.
- **sslflags** (*str*) – The flags representing certificate type and storage of the binding.

Example of usage with only the required arguments:

```
site0-https-binding:
    win_iis.create_binding:
```

(continues on next page)

(continued from previous page)

```
        - site: site0
```

Example of usage specifying all available arguments:

```
site0-https-binding:
    win_iis.create_binding:
        - site: site0
        - hostheader: site0.local
        - ipaddress: '*'
        - port: 443
        - protocol: https
        - sslflags: 0
```

`salt.states.win_iis.`**`create_cert_binding`**(*name*, *site*, *hostheader=''*, *ipaddress='*'*, *port=443*, *sslflags=0*)

Assign a certificate to an IIS binding.

Parameters

- **name** (*str*) – The thumbprint of the certificate.
- **site** (*str*) – The IIS site name.
- **hostheader** (*str*) – The host header of the binding.
- **ipaddress** (*str*) – The IP address of the binding.
- **port** (*str*) – The TCP port of the binding.
- **sslflags** (*str*) – Flags representing certificate type and certificate storage of the binding.

Example of usage with only the required arguments:

```
site0-cert-binding:
    win_iis.create_cert_binding:
        - name: 9988776655443322111000AAABBBCCCDDDEEEFFF
        - site: site0
```

Example of usage specifying all available arguments:

```
site0-cert-binding:
    win_iis.create_cert_binding:
        - name: 9988776655443322111000AAABBBCCCDDDEEEFFF
        - site: site0
        - hostheader: site0.local
        - ipaddress: 192.168.1.199
        - port: 443
        - sslflags: 1
```

New in version 2016.11.0.

`salt.states.win_iis.`**`create_vdir`**(*name*, *site*, *sourcepath*, *app='/'*)

Create an IIS virtual directory.

Parameters

- **name** (*str*) – The virtual directory name.
- **site** (*str*) – The IIS site name.
- **sourcepath** (*str*) – The physical path.
- **app** (*str*) – The IIS application.

Example of usage with only the required arguments:

```
site0-foo-vdir:
    win_iis.create_vdir:
```

(continues on next page)

(continued from previous page)

```
        - name: foo
        - site: site0
        - sourcepath: C:\inetpub\vdirs\foo
```

Example of usage specifying all available arguments:

```
site0-foo-vdir:
    win_iis.create_vdir:
        - name: foo
        - site: site0
        - sourcepath: C:\inetpub\vdirs\foo
        - app: v1
```

`salt.states.win_iis.`**`deployed`**(*name*, *sourcepath*, *apppool=''*, *hostheader=''*, *ipaddress='*'*, *port=80*, *protocol='http'*)

Ensure the website has been deployed.

Parameters

- **name** (*str*) – The IIS site name.
- **sourcepath** (*str*) – The physical path of the IIS site.
- **apppool** (*str*) – The name of the IIS application pool.
- **hostheader** (*str*) – The host header of the binding.
- **ipaddress** (*str*) – The IP address of the binding.
- **port** (*str*) – The TCP port of the binding.
- **protocol** (*str*) – The application protocol of the binding.

Example of usage with only the required arguments. This will default to using the default application pool assigned by IIS:

```
site0-deployed:
    win_iis.deployed:
        - name: site0
        - sourcepath: C:\inetpub\site0
```

Example of usage specifying all available arguments:

```
site0-deployed:
    win_iis.deployed:
        - name: site0
        - sourcepath: C:\inetpub\site0
        - apppool: site0
        - hostheader: site0.local
        - ipaddress: '*'
        - port: 443
        - protocol: https
```

`salt.states.win_iis.`**`remove_app`**(*name*, *site*)

Remove an IIS application.

Parameters

- **name** (*str*) – The application name.
- **site** (*str*) – The IIS site name.

Usage:

```
site0-v1-app-remove:
    win_iis.remove_app:
        - name: v1
        - site: site0
```

`salt.states.win_iis.`**`remove_apppool`**(*name*)

Remove an IIS application pool.

Parameters **`name`** (*`str`*) – The name of the IIS application pool.

Usage:

```
defaultapppool-remove:
    win_iis.remove_apppool:
        - name: DefaultAppPool
```

`salt.states.win_iis.`**`remove_binding`**(*name, site, hostheader='', ipaddress='*', port=80*)

Remove an IIS binding.

Parameters

- **`site`** (*`str`*) – The IIS site name.
- **`hostheader`** (*`str`*) – The host header of the binding.
- **`ipaddress`** (*`str`*) – The IP address of the binding.
- **`port`** (*`str`*) – The TCP port of the binding.

Example of usage with only the required arguments:

```
site0-https-binding-remove:
    win_iis.remove_binding:
        - site: site0
```

Example of usage specifying all available arguments:

```
site0-https-binding-remove:
    win_iis.remove_binding:
        - site: site0
        - hostheader: site0.local
        - ipaddress: '*'
        - port: 443
```

`salt.states.win_iis.`**`remove_cert_binding`**(*name, site, hostheader='', ipaddress='*', port=443*)

Remove a certificate from an IIS binding.

Parameters

- **`name`** (*`str`*) – The thumbprint of the certificate.
- **`site`** (*`str`*) – The IIS site name.
- **`hostheader`** (*`str`*) – The host header of the binding.
- **`ipaddress`** (*`str`*) – The IP address of the binding.
- **`port`** (*`str`*) – The TCP port of the binding.

Example of usage with only the required arguments:

```
site0-cert-binding-remove:
    win_iis.remove_cert_binding:
        - name: 9988776655443322111000AAABBBCCCDDDEEEFFF
        - site: site0
```

Example of usage specifying all available arguments:

```
site0-cert-binding-remove:
    win_iis.remove_cert_binding:
        - name: 9988776655443322111000AAABBBCCCDDDEEEFFF
        - site: site0
        - hostheader: site0.local
        - ipaddress: 192.168.1.199
        - port: 443
```

New in version 2016.11.0.

`salt.states.win_iis.`**`remove_site`**`(`*name*`)`

Delete a website from IIS.

Parameters **name** (*str*) – The IIS site name.

Usage:

```
defaultwebsite-remove:
    win_iis.remove_site:
        - name: Default Web Site
```

`salt.states.win_iis.`**`remove_vdir`**`(`*name, site, app='/'*`)`

Remove an IIS virtual directory.

Parameters

- **name** (*str*) – The virtual directory name.
- **site** (*str*) – The IIS site name.
- **app** (*str*) – The IIS application.

Example of usage with only the required arguments:

```
site0-foo-vdir-remove:
    win_iis.remove_vdir:
        - name: foo
        - site: site0
```

Example of usage specifying all available arguments:

```
site0-foo-vdir-remove:
    win_iis.remove_vdir:
        - name: foo
        - site: site0
        - app: v1
```

`salt.states.win_iis.`**`set_app`**`(`*name, site, settings=None*`)`

New in version 2017.7.0.

Set the value of the setting for an IIS web application.

Note: This function only configures existing app. Params are case sensitive.

Parameters

- **name** (*str*) – The IIS application.
- **site** (*str*) – The IIS site name.
- **settings** (*str*) – A dictionary of the setting names and their values.

Available settings:

- `physicalPath` - The physical path of the webapp
- `applicationPool` - The application pool for the webapp
- `userName` "connectAs" user
- `password` "connectAs" password for user

Return type bool

Example of usage:

```
site0-webapp-setting:
    win_iis.set_app:
        - name: app0
        - site: Default Web Site
```

(continues on next page)

(continued from previous page)

```
        - settings:
            userName: domain\\user
            password: pass
            physicalPath: c:\inetpub\wwwroot
            applicationPool: appPool0
```

`salt.states.win_iis.`**`webconfiguration_settings`**`(`*name*, *settings=None*`)`

Set the value of webconfiguration settings.

Parameters

- **name** (*str*) – The name of the IIS PSPath containing the settings. Possible PSPaths are : MACHINE, MACHINE/WEBROOT, IIS:, IIS:Sitessitename, …
- **settings** (*dict*) – Dictionaries of dictionaries. You can match a specific item in a collection with this syntax inside a key: 'Collection[{name: site0}].logFile.directory'

Example of usage for the `MACHINE/WEBROOT` PSPath:

```
MACHINE-WEBROOT-level-security:
  win_iis.webconfiguration_settings:
    - name: 'MACHINE/WEBROOT'
    - settings:
        system.web/authentication/forms:
          requireSSL: True
          protection: "All"
          credentials.passwordFormat: "SHA1"
        system.web/httpCookies:
          httpOnlyCookies: True
```

Example of usage for the `IIS:\Sites\site0` PSPath:

```
site0-IIS-Sites-level-security:
  win_iis.webconfiguration_settings:
    - name: 'IIS:\Sites\site0'
    - settings:
        system.webServer/httpErrors:
          errorMode: "DetailedLocalOnly"
        system.webServer/security/requestFiltering:
          allowDoubleEscaping: False
          verbs.Collection:
            - verb: TRACE
              allowed: False
          fileExtensions.allowUnlisted: False
```

Example of usage for the `IIS:\` PSPath with a collection matching:

```
site0-IIS-level-security:
  win_iis.webconfiguration_settings:
    - name: 'IIS:\'
    - settings:
        system.applicationHost/sites:
          'Collection[{name: site0}].logFile.directory': 'C:\logs\iis\site0'
```

21.19.294 salt.states.win_lgpo module

Manage Windows Local Group Policy

New in version 2016.11.0.

This state allows configuring local Windows Group Policy

The state can be used to ensure the setting of a single policy or multiple policies in one pass.

Single policies must specify the policy name, the setting, and the policy class (Machine/User/Both)

Example single policy configuration

```
Ensure Account Lockout Duration:
  lgpo.set:
    - name: Account lockout duration
    - setting: 90
    - policy_class: Machine
```

```
Account lockout duration:
  lgpo.set:
    - setting: 120
    - policy_class: Machine
```

Multiple policy configuration

```
Company Local Group Policy:
    lgpo.set:
        - computer_policy:
            Deny log on locally:
              - Guest
            Account lockout duration: 120
            Account lockout threshold: 10
            Reset account lockout counter after: 120
            Enforce password history: 24
            Maximum password age: 60
            Minimum password age: 1
            Minimum password length: 14
            Password must meet complexity requirements: Enabled
            Store passwords using reversible encryption: Disabled
            Configure Automatic Updates:
                Configure automatic updating: 4 - Auto download and schedule the␣
↪intsall
                Scheduled install day: 7 - Every Saturday
                Scheduled install time: 17:00
            Specify intranet Microsoft update service location:
                Set the intranet update service for detecting updates: http://mywsus
                Set the intranet statistics server: http://mywsus
        - user_policy:
            Do not process the legacy run list: Enabled
```

```
server_policy:
  lgpo.set:
    - computer_policy:
        Maximum password age: 60
        Minimum password age: 1
        Minimum password length: 14
        Account lockout duration: 120
        Account lockout threshold: 10
        Reset account lockout counter after: 120
        Manage auditing and security log:
          - "BUILTIN\Administrators"
        Replace a process level token:
```

(continues on next page)

(continued from previous page)

```
         - "NT AUTHORITY\NETWORK SERVICE"
         - "NT AUTHORITY\LOCAL SERVICE"
       "Accounts: Guest account status": Disabled
       "Accounts: Rename guest account": Not_4_U
       "Audit: Audit the use of Backup and Restore privilege": Enabled
       "Interactive logon: Do not display last user name": Enabled
       "Network\DNS Client\Dynamic update": Disabled
       "System\Logon\Do not display the Getting Started welcome screen at logon":
↪Enabled
       "Windows Components\Remote Desktop Services\Remote Desktop Session
↪Host\Connections\Select RDP transport protocols":
           "Select Transport Type": "Use both UDP and TCP"
       "Windows Components\Windows Update\Allow Automatic Updates immediate
↪installation": Enabled
       "Windows Components\Windows Update\Allow non-administrators to receive update
↪notifications": Disabled
       "Windows Components\Windows Update\Always automatically restart at the
↪scheduled time":
           "The restart timer will give users this much time to save their work
↪(minutes)": 15
       "Windows Components\Windows Update\Automatic Updates detection frequency":
           "Check for updates at the following interval (hours)": 1
       "Windows Components\Windows Update\Configure Automatic Updates":
           "Configure automatic updating": 4 - Auto download and schedule the install
           "Install during automatic maintenance": False
           "Scheduled install day": 7 - Every Saturday
           "Scheduled install time": "17:00"
       "Windows Components\Windows Update\Delay Restart for scheduled installations":
           "Wait the following period before proceeding with a scheduled restart
↪(minutes)": 1
       "Windows Components\Windows Update\No auto-restart with logged on users for
↪scheduled automatic updates installations": Disabled
       "Windows Components\Windows Update\Re-prompt for restart with scheduled
↪installations":
           "Wait the following period before prompting again with a scheduled
↪restart (minutes)": 30
       "Windows Components\Windows Update\Reschedule Automatic Updates scheduled
↪installations": Disabled
       "Windows Components\Windows Update\Specify intranet Microsoft update service
↪location":
           "Set the intranet update service for detecting updates": http://mywsus
           "Set the intranet statistics server": http://mywsus
   - cumulative_rights_assignments: True
```

`salt.states.win_lgpo.`**`set_`**(*name*, *setting=None*, *policy_class=None*, *computer_policy=None*, *user_policy=None*, *cumulative_rights_assignments=True*, *adml_language='en-US'*)

Ensure the specified policy is set.

> **Warning:** The `setting` argument cannot be used in conjunction with the `computer_policy` or `user_policy` arguments

Parameters

- **`name`** (*`str`*) – The name of a single policy to configure
- **`setting`** (*`str, dict, list`*) – The configuration setting for the single named

policy. If this argument is used the `computer_policy` / `user_policy` arguments will be ignored

- **policy_class** (*str*) – The policy class of the single named policy to configure. This can `machine`, `user`, or `both`
- **computer_policy** (*dict*) – A dictionary of containing the policy name and key/value pairs of a set of computer policies to configure. If this argument is used, the `name` / `policy_class` arguments will be ignored
- **user_policy** (*dict*) – A dictionary of containing the policy name and key/value pairs of a set of user policies to configure. If this argument is used, the `name` / `policy_class` arguments will be ignored
- **cumulative_rights_assignments** (*bool*) – If user rights assignments are being configured, determines if any user right assignment policies specified will be cumulative or explicit
- **adml_language** (*str*) – The adml language to use for AMDX policy data/display conversions. Default is `en-US`

21.19.295 salt.states.win_license module

Installation and activation of windows licenses

Install and activate windows licenses

```
XXXXX-XXXXX-XXXXX-XXXXX-XXXXX:
  license.activate
```

`salt.states.win_license.`**`activate`**(*name*)

Install and activate the given product key

name The 5x5 product key given to you by Microsoft

21.19.296 salt.states.win_network

Configuration of network interfaces on Windows hosts

New in version 2014.1.0.

This module provides the `network` state(s) on Windows hosts. DNS servers, IP addresses and default gateways can currently be managed.

Below is an example of the configuration for an interface that uses DHCP for both DNS servers and IP addresses:

```
Local Area Connection #2:
  network.managed:
    - dns_proto: dhcp
    - ip_proto: dhcp
```

Note: Both the `dns_proto` and `ip_proto` arguments are required.

Static DNS and IP addresses can be configured like so:

```
Local Area Connection #2:
  network.managed:
    - dns_proto: static
```

(continues on next page)

(continued from previous page)

```
    - dns_servers:
      - 8.8.8.8
      - 8.8.4.4
    - ip_proto: static
    - ip_addrs:
      - 10.2.3.4/24
```

Note: IP addresses are specified using the format `<ip-address>/<subnet-length>`. Salt provides a convenience function called *`ip.get_subnet_length`* to calculate the subnet length from a netmask.

Optionally, if you are setting a static IP address, you can also specify the default gateway using the `gateway` parameter:

```
Local Area Connection #2:
  network.managed:
    - dns_proto: static
    - dns_servers:
      - 8.8.8.8
      - 8.8.4.4
    - ip_proto: static
    - ip_addrs:
      - 10.2.3.4/24
    - gateway: 10.2.3.1
```

`salt.states.win_network.`**`managed`**(*name*, *dns_proto=None*, *dns_servers=None*, *ip_proto=None*, *ip_addrs=None*, *gateway=None*, *enabled=True*, ***kwargs*)

Ensure that the named interface is configured properly.

Parameters

- **name** (*str*) – The name of the interface to manage
- **dns_proto** (*str*) – None Set to `static` and use the `dns_servers` parameter to provide a list of DNS nameservers. set to `dhcp` to use DHCP to get the DNS servers.
- **dns_servers** (*list*) – None A list of static DNS servers. To clear the list of DNS servers pass an empty list (`[]`). `None` will make no changes.
- **ip_proto** (*str*) – None Set to `static` and use the `ip_addrs` and (optionally) `gateway` parameters to provide a list of static IP addresses and the default gateway. Set to `dhcp` to use DHCP.
- **ip_addrs** (*list*) – None A list of static IP addresses with netmask flag, ie: 192.168.0.11/24
- **gateway** (*str*) – None The gateway to set for the interface
- **enabled** (*bool*) – True Set to `False` to ensure that this interface is disabled.

Returns A dictionary of old and new settings

Return type dict

Example:

```
Ethernet1:
  network.managed:
    - dns_proto: static
    - dns_servers:
      - 8.8.8.8
      - 8.8.4.4
    - ip_proto: static
    - ip_addrs:
      - 192.168.0.100/24
```

Clear DNS entries example:

```
Ethernet1:
  network.managed:
    - dns_proto: static
    - dns_servers: []
    - ip_proto: dhcp
```

21.19.297 salt.states.win_path

Manage the Windows System PATH

`salt.states.win_path.`**`absent`**`(`*name*`)`

Remove the directory from the SYSTEM path

Example:

```
'C:\sysinternals':
  win_path.absent
```

`salt.states.win_path.`**`exists`**`(`*name, index=None*`)`

Add the directory to the system PATH at index location

index Position where the directory should be placed in the PATH. This is 0-indexed, so 0 means to prepend at the very start of the PATH.

> **Note:** If the index is not specified, and the directory needs to be added to the PATH, then the directory will be appended to the PATH, and this state will not enforce its location within the PATH.

Examples:

```
'C:\python27':
  win_path.exists

'C:\sysinternals':
  win_path.exists:
    - index: 0

'C:\mystuff':
  win_path.exists:
    - index: -1
```

21.19.298 salt.states.win_pki module

Microsoft certificate management via the Pki PowerShell module.

platform Windows

New in version 2016.11.0.

`salt.states.win_pki.`**`import_cert`**`(`*name, cert_format='cer', context='LocalMachine', store='My', exportable=True, password='', saltenv='base'*`)`

Import the certificate file into the given certificate store.

Parameters

- **name** (*str*) – The path of the certificate file to import.

- **cert_format** (*str*) – The certificate format. Specify 'cer' for X.509, or 'pfx' for PKCS #12.
- **context** (*str*) – The name of the certificate store location context.
- **store** (*str*) – The name of the certificate store.
- **exportable** (*bool*) – Mark the certificate as exportable. Only applicable to pfx format.
- **password** (*str*) – The password of the certificate. Only applicable to pfx format.
- **saltenv** (*str*) – The environment the file resides in.

Example of usage with only the required arguments:

```
site0-cert-imported:
    win_pki.import_cert:
        - name: salt://win/webserver/certs/site0.cer
```

Example of usage specifying all available arguments:

```
site0-cert-imported:
    win_pki.import_cert:
        - name: salt://win/webserver/certs/site0.pfx
        - cert_format: pfx
        - context: LocalMachine
        - store: My
        - exportable: True
        - password: TestPassword
        - saltenv: base
```

`salt.states.win_pki.`**`remove_cert`**(*name*, *thumbprint*, *context='LocalMachine'*, *store='My'*)

Remove the certificate from the given certificate store.

Parameters

- **thumbprint** (*str*) – The thumbprint value of the target certificate.
- **context** (*str*) – The name of the certificate store location context.
- **store** (*str*) – The name of the certificate store.

Example of usage with only the required arguments:

```
site0-cert-removed:
    win_pki.remove_cert:
        - thumbprint: 9988776655443322111000AAABBBCCCDDDEEEFFF
```

Example of usage specifying all available arguments:

```
site0-cert-removed:
    win_pki.remove_cert:
        - thumbprint: 9988776655443322111000AAABBBCCCDDDEEEFFF
        - context: LocalMachine
        - store: My
```

21.19.299 salt.states.win_powercfg

This module allows you to control the power settings of a windows minion via powercfg.

New in version 2015.8.0.

```
# Set timeout to 30 minutes on battery power
monitor:
    powercfg.set_timeout:
```

(continues on next page)

(continued from previous page)

```
      - value: 30
      - power: dc
```

`salt.states.win_powercfg.`**`set_timeout`**(*name*, *value*, *power='ac'*, *scheme=None*)

Set the sleep timeouts of specific items such as disk, monitor, etc.

Parameters

- **name** (*str*) – The setting to change, can be one of the following:
  - `monitor`
  - `disk`
  - `standby`
  - `hibernate`
- **value** (*int*) – The amount of time in minutes before the item will timeout
- **power** (*str*) – Set the value for AC or DC power. Default is `ac`. Valid options are:
  - `ac` (AC Power)
  - `dc` (Battery)
- **scheme** (*str*) – The scheme to use, leave as `None` to use the current. Default is `None`. This can be the GUID or the Alias for the Scheme. Known Aliases are:
  - `SCHEME_BALANCED` - Balanced
  - `SCHEME_MAX` - Power saver
  - `SCHEME_MIN` - High performance

CLI Example:

```
# Set monitor timeout to 30 minutes on Battery
monitor:
  powercfg.set_timeout:
    - value: 30
    - power: dc

# Set disk timeout to 10 minutes on AC Power
disk:
  powercfg.set_timeout:
    - value: 10
    - power: ac
```

21.19.300 salt.states.win_servermanager

Manage Windows features via the ServerManager powershell module. Can install and remove roles/features.

maintainer Shane Lee <slee@saltstack.com>

platform Windows Server 2008R2 or greater

depends win_servermanager.install

depends win_servermanager.remove

`salt.states.win_servermanager.`**`installed`**(*name*, *features=None*, *recurse=False*, *restart=False*, *source=None*, *exclude=None*, ***kwargs*)

Install the windows feature. To install a single feature, use the `name` parameter. To install multiple features, use the `features` parameter.

Note: Some features require reboot after un/installation. If so, until the server is restarted other features can not be installed!

Parameters

- **name** (*str*) – Short name of the feature (the right column in win_servermanager.list_available). This can be a single feature or a string of features in a comma delimited list (no spaces)

  Note: A list is not allowed in the name parameter of any state. Use the `features` parameter if you want to pass the features as a list

  ---

- **features** (*Optional[list]*) – A list of features to install. If this is passed it will be used instead of the `name` parameter.

  New in version 2018.3.0.
- **recurse** (*Optional[bool]*) – Install all sub-features as well. If the feature is installed but one of its sub-features are not installed set this will install additional sub-features. This argument was previously renamed from `force`. To ensure backwards compatibility `force` will continue to work but please update your states to use the preferred `recurse` arg.
- **source** (*Optional[str]*) – Path to the source files if missing from the target system. None means that the system will use windows update services to find the required files. Default is None
- **restart** (*Optional[bool]*) – Restarts the computer when installation is complete, if required by the role/feature installed. Default is False
- **exclude** (*Optional[str]*) – The name of the feature to exclude when installing the named feature. This can be a single feature, a string of features in a comma-delimited list (no spaces), or a list of features.

  > **Warning:** As there is no exclude option for the `Add-WindowsFeature` or `Install-WindowsFeature` PowerShell commands the features named in `exclude` will be installed with other sub-features and will then be removed. **If the feature named in ``exclude`` is not a sub-feature of one of the installed items it will still be removed.**

Example

Do not use the role or feature names mentioned in the PKGMGR documentation. To get a list of available roles and features run the following command:

```
salt <minion_name> win_servermanager.list_available
```

Use the name in the right column of the results.

```
# Installs the IIS Web Server Role (Web-Server)
IIS-WebServerRole:
  win_servermanager.installed:
    - recurse: True
    - name: Web-Server

# Install multiple features, exclude the Web-Service
install_multiple_features:
  win_servermanager.installed:
    - recurse: True
    - features:
      - RemoteAccess
```

(continues on next page)

(continued from previous page)

```
    - XPS-Viewer
    - SNMP-Service
  - exclude:
    - Web-Server
```

`salt.states.win_servermanager.`**`removed`**(*name*, *features=None*, *remove_payload=False*, *restart=False*)

Remove the windows feature To remove a single feature, use the `name` parameter. To remove multiple features, use the `features` parameter.

Parameters

- **`name`** (*`str`*) – Short name of the feature (the right column in win_servermanager.list_available). This can be a single feature or a string of features in a comma-delimited list (no spaces)

  Note: A list is not allowed in the name parameter of any state. Use the `features` parameter if you want to pass the features as a list

  ---

- **`features`** (*`Optional[list]`*) – A list of features to remove. If this is passed it will be used instead of the `name` parameter.

  New in version 2018.3.0.
- **`remove_payload`** (*`Optional[bool]`*) – True will cause the feature to be removed from the side-by-side store. To install the feature in the future you will need to specify the `source`
- **`restart`** (*`Optional[bool]`*) – Restarts the computer when uninstall is complete if required by the role/feature uninstall. Default is False

---

Note: Some features require a reboot after uninstall. If so the feature will not be completely uninstalled until the server is restarted.

Example

Do not use the role or feature names mentioned in the PKGMGR documentation. To get a list of available roles and features run the following command:

```
salt <minion_name> win_servermanager.list_available
```

Use the name in the right column of the results.

```
# Uninstall the IIS Web Server Rol (Web-Server)
IIS-WebserverRole:
  win_servermanager.removed:
    - name: Web-Server

# Uninstall multiple features, reboot if required
uninstall_multiple_features:
  win_servermanager.removed:
    - features:
      - RemoteAccess
      - XPX-Viewer
      - SNMP-Service
    - restart: True
```

21.19.301 salt.states.win_smtp_server module

Module for managing IIS SMTP server configuration on Windows servers.

`salt.states.win_smtp_server.`**`active_log_format`**(*name, log_format, server='SmtpSvc/1'*)

Manage the active log format for the SMTP server.

Parameters

- **`log_format`** (*`str`*) – The log format name.
- **`server`** (*`str`*) – The SMTP server name.

Example of usage:

```
smtp-log-format:
    win_smtp_server.active_log_format:
        - log_format: Microsoft IIS Log File Format
```

`salt.states.win_smtp_server.`**`connection_ip_list`**(*name, addresses=None, grant_by_default=False, server='SmtpSvc/1'*)

Manage IP list for SMTP connections.

Parameters

- **`addresses`** (*`str`*) – A dictionary of IP + subnet pairs.
- **`grant_by_default`** (*`bool`*) – Whether the addresses should be a blacklist or whitelist.
- **`server`** (*`str`*) – The SMTP server name.

Example of usage for creating a whitelist:

```
smtp-connection-whitelist:
    win_smtp_server.connection_ip_list:
        - addresses:
            127.0.0.1: 255.255.255.255
            172.16.1.98: 255.255.255.255
            172.16.1.99: 255.255.255.255
        - grant_by_default: False
```

Example of usage for creating a blacklist:

```
smtp-connection-blacklist:
    win_smtp_server.connection_ip_list:
        - addresses:
            172.16.1.100: 255.255.255.255
            172.16.1.101: 255.255.255.255
        - grant_by_default: True
```

Example of usage for allowing any source to connect:

```
smtp-connection-blacklist:
    win_smtp_server.connection_ip_list:
        - addresses: {}
        - grant_by_default: True
```

`salt.states.win_smtp_server.`**`relay_ip_list`**(*name, addresses=None, server='SmtpSvc/1'*)

Manage IP list for SMTP relay connections.

Due to the unusual way that Windows stores the relay IPs, it is advisable to retrieve the existing list you wish to set from a pre-configured server.

For example, setting '127.0.0.1' as an allowed relay IP through the GUI would generate an actual relay IP list similar to the following:

```
['24.0.0.128', '32.0.0.128', '60.0.0.128', '68.0.0.128', '1.0.0.0', '76.0.0.0',
  '0.0.0.0', '0.0.0.0', '1.0.0.0', '1.0.0.0', '2.0.0.0', '2.0.0.0', '4.0.0.0',
  '0.0.0.0', '76.0.0.128', '0.0.0.0', '0.0.0.0', '0.0.0.0', '0.0.0.0',
  '255.255.255.255', '127.0.0.1']
```

Note: Setting the list to None corresponds to the restrictive 'Only the list below' GUI parameter with an empty access list configured, and setting an empty list/tuple corresponds to the more permissive 'All except the list below' GUI parameter.

Parameters

- **addresses** (*str*) – A list of the relay IPs. The order of the list is important.
- **server** (*str*) – The SMTP server name.

Example of usage:

```
smtp-relay-list:
  win_smtp_server.relay_ip_list:
    - addresses:
        - 24.0.0.128
        - 32.0.0.128
        - 60.0.0.128
        - 1.0.0.0
        - 76.0.0.0
        - 0.0.0.0
        - 0.0.0.0
        - 1.0.0.0
        - 1.0.0.0
        - 2.0.0.0
        - 2.0.0.0
        - 4.0.0.0
        - 0.0.0.0
        - 76.0.0.128
        - 0.0.0.0
        - 0.0.0.0
        - 0.0.0.0
        - 0.0.0.0
        - 255.255.255.255
        - 127.0.0.1
```

Example of usage for disabling relaying:

```
smtp-relay-list:
    win_smtp_server.relay_ip_list:
        - addresses: None
```

Example of usage for allowing relaying from any source:

```
smtp-relay-list:
    win_smtp_server.relay_ip_list:
        - addresses: []
```

`salt.states.win_smtp_server.`**`server_setting`**(*name, settings=None, server='SmtpSvc/1'*)

Ensure the value is set for the specified setting.

Note: The setting names are case-sensitive.

Parameters
- **settings** (*str*) – A dictionary of the setting names and their values.
- **server** (*str*) – The SMTP server name.

Example of usage:

```
smtp-settings:
    win_smtp_server.server_setting:
        - settings:
            LogType: 1
            LogFilePeriod: 1
            MaxMessageSize: 16777216
            MaxRecipients: 10000
            MaxSessionSize: 16777216
```

21.19.302 salt.states.win_snmp module

Module for managing SNMP service settings on Windows servers.

`salt.states.win_snmp.`**`agent_settings`**(*name*, *contact*, *location*, *services=None*)

Manage the SNMP sysContact, sysLocation, and sysServices settings.

Parameters
- **contact** (*str*) – The SNMP contact.
- **location** (*str*) – The SNMP location.
- **services** (*str*) – A list of selected services.

Example of usage:

```
snmp-agent-settings:
    win_snmp.agent_settings:
        - contact: Test Contact
        - location: Test Location
        - services:
            - Physical
            - Internet
```

`salt.states.win_snmp.`**`auth_traps_enabled`**(*name*, *status=True*)

Manage the sending of authentication traps.

Parameters **status** (*bool*) – The enabled status.

Example of usage:

```
snmp-auth-traps:
    win_snmp.auth_traps_enabled:
        - status: True
```

`salt.states.win_snmp.`**`community_names`**(*name*, *communities=None*)

Manage the SNMP accepted community names and their permissions.

Parameters **communities** (*str*) – A dictionary of SNMP communities and permissions.

Example of usage:

```
snmp-community-names:
    win_snmp.community_names:
        - communities:
```

(continues on next page)

(continued from previous page)

```
            TestCommunity: Read Only
            OtherCommunity: Read Write
```

21.19.303 salt.states.win_system

Management of Windows system information

New in version 2014.1.0.

This state is used to manage system information such as the computer name and description.

```
ERIK-WORKSTATION:
  system.computer_name: []

This is Erik's computer, don't touch!:
  system.computer_desc: []
```

`salt.states.win_system.`**`computer_desc`**`(`*name*`)`

Manage the computer's description field

name The desired computer description

`salt.states.win_system.`**`computer_name`**`(`*name*`)`

Manage the computer's name

name The desired computer name

`salt.states.win_system.`**`hostname`**`(`*name*`)`

New in version 2016.3.0.

Manage the hostname of the computer

name The hostname to set

`salt.states.win_system.`**`join_domain`**`(`*name, username=None, password=None, account_ou=None, account_exists=False, restart=False*`)`

Checks if a computer is joined to the Domain. If the computer is not in the Domain, it will be joined.

Parameters

- **name** (*str*) – The name of the Domain.
- **username** (*str*) – Username of an account which is authorized to join computers to the specified domain. Need to be either fully qualified like user@domain.tld or simply user.
- **password** (*str*) – Password of the account to add the computer to the Domain.
- **account_ou** (*str*) – The DN of the OU below which the account for this computer should be created when joining the domain, e.g. ou=computers,ou=departm_432,dc=my-company,dc=com.
- **account_exists** (*bool*) – Needs to be set to `True` to allow re-using an existing computer account.
- **restart** (*bool*) – Needs to be set to `True` to restart the computer after a successful join.

Example:

```
join_to_domain:
  system.join_domain:
    - name: mydomain.local.com
    - username: myaccount@mydomain.local.com
    - password: mysecretpassword
    - restart: True
```

`salt.states.win_system.`**`reboot`**(*name*, *message=None*, *timeout=5*, *force_close=True*, *in_seconds=False*, *only_on_pending_reboot=True*)

Reboot the computer

Parameters

- **`message`** (*`str`*) – An optional message to display to users. It will also be used as a comment in the event log entry.

  The default value is None.
- **`timeout`** (*`int`*) – The number of minutes or seconds before a reboot will occur. Whether this number represents minutes or seconds depends on the value of `in_seconds`.

  The default value is 5.
- **`in_seconds`** (*`bool`*) – If this is True, the value of `timeout` will be treated as a number of seconds. If this is False, the value of `timeout` will be treated as a number of minutes.

  The default value is False.
- **`force_close`** (*`bool`*) – If this is True, running applications will be forced to close without warning. If this is False, running applications will not get the opportunity to prompt users about unsaved data.

  The default value is True.
- **`only_on_pending_reboot`** (*`bool`*) – If this is True, the reboot will only occur if the system reports a pending reboot. If this is False, the reboot will always occur.

  The default value is True.

`salt.states.win_system.`**`shutdown`**(*name*, *message=None*, *timeout=5*, *force_close=True*, *reboot=False*, *in_seconds=False*, *only_on_pending_reboot=False*)

Shutdown the computer

Parameters

- **`message`** (*`str`*) – An optional message to display to users. It will also be used as a comment in the event log entry.

  The default value is None.
- **`timeout`** (*`int`*) – The number of minutes or seconds before a shutdown will occur. Whether this number represents minutes or seconds depends on the value of `in_seconds`.

  The default value is 5.
- **`in_seconds`** (*`bool`*) – If this is True, the value of `timeout` will be treated as a number of seconds. If this is False, the value of `timeout` will be treated as a number of minutes.

  The default value is False.
- **`force_close`** (*`bool`*) – If this is True, running applications will be forced to close without warning. If this is False, running applications will not get the opportunity to prompt users about unsaved data.

  The default value is True.
- **`reboot`** (*`bool`*) – If this is True, the computer will restart immediately after shutting down. If False the system flushes all caches to disk and safely powers down the system.

  The default value is False.
- **`only_on_pending_reboot`** (*`bool`*) – If this is True, the shutdown will only occur if the system reports a pending reboot. If this is False, the shutdown will always occur.

The default value is False.

21.19.304 salt.states.win_wua module

Installation of Windows Updates using the Windows Update Agent

New in version 2017.7.0.

Salt can manage Windows updates via the "wua" state module. Updates can be installed and removed. Update management declarations are as follows:

For installation:

```
# Install a single update using the KB
KB3194343:
  wua.installed

# Install a single update using the name parameter
install_update:
  wua.installed:
    - name: KB3194343

# Install multiple updates using the updates parameter and a combination of
# KB number and GUID
install_updates:
  wua.installed:
   - updates:
     - KB3194343
     - bb1dbb26-3fb6-45fd-bb05-e3c8e379195c
```

For removal:

```
# Remove a single update using the KB
KB3194343:
  wua.removed

# Remove a single update using the name parameter
remove_update:
  wua.removed:
    - name: KB3194343

# Remove multiple updates using the updates parameter and a combination of
# KB number and GUID
remove_updates:
  wua.removed:
   - updates:
     - KB3194343
     - bb1dbb26-3fb6-45fd-bb05-e3c8e379195c
```

`salt.states.win_wua.`**`installed`**(*name*, *updates=None*)

Ensure Microsoft Updates are installed. Updates will be downloaded if needed.

Parameters

- **name** (*str*) – The identifier of a single update to install.
- **updates** (*list*) – A list of identifiers for updates to be installed. Overrides `name`. Default is None.

Note: Identifiers can be the GUID, the KB number, or any part of the Title of the Microsoft update. GUIDs and KBs are the preferred method to ensure you're installing the correct update.

Warning: Using a partial KB number or a partial Title could result in more than one update being installed.

Returns A dictionary containing the results of the update

Return type dict

CLI Example:

```
# using a GUID
install_update:
  wua.installed:
    - name: 28cf1b09-2b1a-458c-9bd1-971d1b26b211

# using a KB
install_update:
  wua.installed:
    - name: KB3194343

# using the full Title
install_update:
  wua.installed:
    - name: Security Update for Adobe Flash Player for Windows 10 Version 1607
↪(for x64-based Systems) (KB3194343)

# Install multiple updates
install_updates:
  wua.installed:
    - updates:
      - KB3194343
      - 28cf1b09-2b1a-458c-9bd1-971d1b26b211
```

`salt.states.win_wua.`**`removed`**(*name*, *updates=None*)

Ensure Microsoft Updates are uninstalled.

Parameters

- **name** (*str*) – The identifier of a single update to uninstall.
- **updates** (*list*) – A list of identifiers for updates to be removed. Overrides `name`. Default is None.

Note: Identifiers can be the GUID, the KB number, or any part of the Title of the Microsoft update. GUIDs and KBs are the preferred method to ensure you're uninstalling the correct update.

Warning: Using a partial KB number or a partial Title could result in more than one update being removed.

Returns A dictionary containing the results of the removal

Return type dict

CLI Example:

```
# using a GUID
uninstall_update:
  wua.removed:
    - name: 28cf1b09-2b1a-458c-9bd1-971d1b26b211
```

(continues on next page)

(continued from previous page)

```
# using a KB
uninstall_update:
  wua.removed:
    - name: KB3194343

# using the full Title
uninstall_update:
  wua.removed:
    - name: Security Update for Adobe Flash Player for Windows 10 Version 1607␣
↪(for x64-based Systems) (KB3194343)

# Install multiple updates
uninstall_updates:
  wua.removed:
    - updates:
      - KB3194343
      - 28cf1b09-2b1a-458c-9bd1-971d1b26b211
```

`salt.states.win_wua.`**`uptodate`**(*name, software=True, drivers=False, skip_hidden=False, skip_mandatory=False, skip_reboot=True, categories=None, severities=None*)

Ensure Microsoft Updates that match the passed criteria are installed. Updates will be downloaded if needed.

This state allows you to update a system without specifying a specific update to apply. All matching updates will be installed.

Parameters

- **name** (*str*) – The name has no functional value and is only used as a tracking reference
- **software** (*bool*) – Include software updates in the results (default is True)
- **drivers** (*bool*) – Include driver updates in the results (default is False)
- **skip_hidden** (*bool*) – Skip updates that have been hidden. Default is False.
- **skip_mandatory** (*bool*) – Skip mandatory updates. Default is False.
- **skip_reboot** (*bool*) – Skip updates that require a reboot. Default is True.
- **categories** (*list*) – Specify the categories to list. Must be passed as a list. All categories returned by default.

  Categories include the following:
  - Critical Updates
  - Definition Updates
  - Drivers (make sure you set drivers=True)
  - Feature Packs
  - Security Updates
  - Update Rollups
  - Updates
  - Update Rollups
  - Windows 7
  - Windows 8.1
  - Windows 8.1 drivers
  - Windows 8.1 and later drivers
  - Windows Defender
- **severities** (*list*) – Specify the severities to include. Must be passed as a list. All severities returned by default.

  Severities include the following:
  - Critical

    - Important

Returns A dictionary containing the results of the update

Return type dict

CLI Example:

```
# Update the system using the state defaults
update_system:
  wua.uptodate

# Update the drivers
update_drivers:
  wua.uptodate:
    - software: False
    - drivers: True
    - skip_reboot: False

# Apply all critical updates
update_critical:
  wua.uptodate:
    - severities:
      - Critical
```

21.19.305 salt.states.win_wusa module

Microsoft Updates (KB) Management

This module provides the ability to enforce KB installations from files (.msu), without WSUS or Windows Update

New in version 2018.3.4.

`salt.states.win_wusa.`**`installed`**(*name*, *source*)

Ensure an update is installed on the minion

Parameters

- **name** (*str*) – Name of the Windows KB ("KB123456")
- **source** (*str*) – Source of .msu file corresponding to the KB

Example:

```
KB123456:
  wusa.installed:
    - source: salt://kb123456.msu
```

`salt.states.win_wusa.`**`uninstalled`**(*name*)

Ensure an update is uninstalled from the minion

Parameters **name** (*str*) – Name of the Windows KB ("KB123456")

Example:

```
KB123456:
  wusa.uninstalled
```

21.19.306 salt.states.winrepo

Manage Windows Package Repository

`salt.states.winrepo.`**`genrepo`**(*name*, *force=False*, *allow_empty=False*)

Refresh the winrepo.p file of the repository (salt-run winrepo.genrepo)

If `force` is `True` no checks will be made and the repository will be generated if `allow_empty` is `True` then the state will not return an error if there are 0 packages,

---

Note: This state only loads on minions that have the `roles: salt-master` grain set.

Example:

```
winrepo:
  winrepo.genrepo
```

21.19.307 salt.states.x509

Manage X509 Certificates

New in version 2015.8.0.

depends M2Crypto

This module can enable managing a complete PKI infrastructure including creating private keys, CAs, certificates and CRLs. It includes the ability to generate a private key on a server, and have the corresponding public key sent to a remote CA to create a CA signed certificate. This can be done in a secure manner, where private keys are always generated locally and never moved across the network.

Here is a simple example scenario. In this example `ca` is the ca server, and `www` is a web server that needs a certificate signed by `ca`.

For remote signing, peers must be permitted to remotely call the *`sign_remote_certificate`* function.

/etc/salt/master.d/peer.conf

```
peer:
  .*:
    - x509.sign_remote_certificate
```

/srv/salt/top.sls

```
base:
  '*':
    - cert
  'ca':
    - ca
  'www':
    - www
```

This state creates the CA key, certificate and signing policy. It also publishes the certificate to the mine where it can be easily retrieved by other minions.

/srv/salt/ca.sls

```
salt-minion:
  service.running:
    - enable: True
    - listen:
      - file: /etc/salt/minion.d/signing_policies.conf

/etc/salt/minion.d/signing_policies.conf:
  file.managed:
```

(continues on next page)

(continued from previous page)

```
    - source: salt://signing_policies.conf

/etc/pki:
  file.directory

/etc/pki/issued_certs:
  file.directory

/etc/pki/ca.crt:
  x509.certificate_managed:
    - signing_private_key: /etc/pki/ca.key
    - CN: ca.example.com
    - C: US
    - ST: Utah
    - L: Salt Lake City
    - basicConstraints: "critical CA:true"
    - keyUsage: "critical cRLSign, keyCertSign"
    - subjectKeyIdentifier: hash
    - authorityKeyIdentifier: keyid,issuer:always
    - days_valid: 3650
    - days_remaining: 0
    - backup: True
    - managed_private_key:
        name: /etc/pki/ca.key
        bits: 4096
        backup: True
    - require:
      - file: /etc/pki

mine.send:
  module.run:
    - func: x509.get_pem_entries
    - kwargs:
        glob_path: /etc/pki/ca.crt
    - onchanges:
      - x509: /etc/pki/ca.crt
```

The signing policy defines properties that override any property requested or included in a CRL. It also can define a restricted list of minions which are allowed to remotely invoke this signing policy.

/srv/salt/signing_policies.conf

```
x509_signing_policies:
  www:
    - minions: 'www'
    - signing_private_key: /etc/pki/ca.key
    - signing_cert: /etc/pki/ca.crt
    - C: US
    - ST: Utah
    - L: Salt Lake City
    - basicConstraints: "critical CA:false"
    - keyUsage: "critical keyEncipherment"
    - subjectKeyIdentifier: hash
    - authorityKeyIdentifier: keyid,issuer:always
    - days_valid: 90
    - copypath: /etc/pki/issued_certs/
```

This state will instruct all minions to trust certificates signed by our new CA. Using Jinja to strip newlines from the text avoids dealing with newlines in the rendered YAML, and the `sign_remote_certificate` state will handle properly formatting the text before writing the output.

/srv/salt/cert.sls

```
/usr/local/share/ca-certificates:
  file.directory

/usr/local/share/ca-certificates/intca.crt:
  x509.pem_managed:
    - text: {{ salt['mine.get']('ca', 'x509.get_pem_entries')['ca']['/etc/pki/ca.crt
↪']|replace('\n', '') }}
```

This state creates a private key then requests a certificate signed by ca according to the www policy.

/srv/salt/www.sls

```
/etc/pki/www.crt:
  x509.certificate_managed:
    - ca_server: ca
    - signing_policy: www
    - public_key: /etc/pki/www.key
    - CN: www.example.com
    - days_remaining: 30
    - backup: True
    - managed_private_key:
        name: /etc/pki/www.key
        bits: 4096
        backup: True
```

`salt.states.x509.`**`certificate_managed`**(*name*, *days_remaining=90*, *managed_private_key=None*, *append_certs=None*, ***kwargs*)

Manage a Certificate

name Path to the certificate

days_remaining [90] The minimum number of days remaining when the certificate should be recreated. A value of 0 disables automatic renewal.

managed_private_key Manages the private key corresponding to the certificate. All of the arguments supported by *`x509.private_key_managed`* are supported. If *name* is not specified or is the same as the name of the certificate, the private key and certificate will be written together in the same file.

append_certs: A list of certificates to be appended to the managed file.

kwargs: Any arguments supported by *`x509.create_certificate`* or *`file.managed`* are supported.

Examples:

```
/etc/pki/ca.crt:
  x509.certificate_managed:
    - signing_private_key: /etc/pki/ca.key
    - CN: ca.example.com
    - C: US
    - ST: Utah
    - L: Salt Lake City
    - basicConstraints: "critical CA:true"
    - keyUsage: "critical cRLSign, keyCertSign"
    - subjectKeyIdentifier: hash
    - authorityKeyIdentifier: keyid,issuer:always
    - days_valid: 3650
    - days_remaining: 0
    - backup: True
```

```
/etc/ssl/www.crt:
  x509.certificate_managed:
    - ca_server: pki
    - signing_policy: www
    - public_key: /etc/ssl/www.key
    - CN: www.example.com
    - days_valid: 90
    - days_remaining: 30
    - backup: True
```

`salt.states.x509.`**`crl_managed`**(*name, signing_private_key, signing_private_key_passphrase=None, signing_cert=None, revoked=None, days_valid=100, digest='', days_remaining=30, include_expired=False, **kwargs*)

Manage a Certificate Revocation List

name Path to the certificate

signing_private_key The private key that will be used to sign the CRL. This is usually your CA's private key.

signing_private_key_passphrase Passphrase to decrypt the private key.

signing_cert The certificate of the authority that will be used to sign the CRL. This is usually your CA's certificate.

revoked A list of certificates to revoke. Must include either a serial number or a the certificate itself. Can optionally include the revocation date and notAfter date from the certificate. See example below for details.

days_valid [100] The number of days the certificate should be valid for.

digest The digest to use for signing the CRL. This has no effect on versions of pyOpenSSL less than 0.14.

days_remaining [30] The CRL should be automatically recreated if there are less than `days_remaining` days until the CRL expires. Set to 0 to disable automatic renewal.

include_expired [False] If `True`, include expired certificates in the CRL.

kwargs Any arguments supported by *`file.managed`* are supported.

Example:

```
/etc/pki/ca.crl:
  x509.crl_managed:
    - signing_private_key: /etc/pki/myca.key
    - signing_cert: /etc/pki/myca.crt
    - revoked:
      - compromized_Web_key:
        - certificate: /etc/pki/certs/badweb.crt
        - revocation_date: 2015-03-01 00:00:00
        - reason: keyCompromise
      - terminated_vpn_user:
        - serial_number: D6:D2:DC:D8:4D:5C:C0:F4
        - not_after: 2016-01-01 00:00:00
        - revocation_date: 2015-02-25 00:00:00
        - reason: cessationOfOperation
```

`salt.states.x509.`**`csr_managed`**(*name, **kwargs*)

Manage a Certificate Signing Request

name: Path to the CSR

properties: The properties to be added to the certificate request, including items like subject, extensions and public key. See above for valid properties.

kwargs: Any arguments supported by *`file.managed`* are supported.

Example:

```
/etc/pki/mycert.csr:
  x509.csr_managed:
```

(continues on next page)

(continued from previous page)

```
      - private_key: /etc/pki/mycert.key
      - CN: www.example.com
      - C: US
      - ST: Utah
      - L: Salt Lake City
      - keyUsage: 'critical dataEncipherment'
```

`salt.states.x509.`**`pem_managed`**`(`*name*, *text*, *backup=False*, ***kwargs*`)`

Manage the contents of a PEM file directly with the content in text, ensuring correct formatting.

name: The path to the file to manage

text: The PEM formatted text to write.

kwargs: Any arguments supported by *`file.managed`* are supported.

`salt.states.x509.`**`private_key_managed`**`(`*name*, *bits=2048*, *passphrase=None*, *cipher='aes_128_cbc'*, *new=False*, *overwrite=False*, *verbose=True*, ***kwargs*`)`

Manage a private key's existence.

name: Path to the private key

bits: Key length in bits. Default 2048.

passphrase: Passphrase for encrypting the private key.

cipher: Cipher for encrypting the private key.

new: Always create a new key. Defaults to `False`. Combining new with `prereq`, or when used as part of a *managed_private_key* can allow key rotation whenever a new certificate is generated.

overwrite: Overwrite an existing private key if the provided passphrase cannot decrypt it.

verbose: Provide visual feedback on stdout, dots while key is generated. Default is True.

New in version 2016.11.0.

kwargs: Any kwargs supported by file.managed are supported.

Example:

The JINJA templating in this example ensures a private key is generated if the file doesn't exist and that a new private key is generated whenever the certificate that uses it is to be renewed.

```
/etc/pki/www.key:
  x509.private_key_managed:
    - bits: 4096
    - new: True
    {% if salt['file.file_exists']('/etc/pki/www.key') -%}
    - prereq:
      - x509: /etc/pki/www.crt
    {%- endif %}
```

21.19.308 salt.states.xml

XML Manager

State managment of XML files

`salt.states.xml.`**`value_present`**`(`*name*, *xpath*, *value*, ***kwargs*`)`

New in version 3000.

Manages a given XML file

name [string] The location of the XML file to manage, as an absolute path.

xpath [string] xpath location to manage

value [string] value to ensure present

```
ensure_value_true:
  xml.value_present:
    - name: /tmp/test.xml
    - xpath: .//playwright[@id='1']
    - value: William Shakespeare
```

21.19.309 salt.states.xmpp

Sending Messages over XMPP

New in version 2014.1.0.

This state is useful for firing messages during state runs, using the XMPP protocol

```
server-warning-message:
  xmpp.send_msg:
    - name: 'This is a server warning message'
    - profile: my-xmpp-account
    - recipient: admins@xmpp.example.com/salt
```

`salt.states.xmpp.`**`send_msg`**`(`*name, recipient, profile*`)`

Send a message to an XMPP user

```
server-warning-message:
  xmpp.send_msg:
    - name: 'This is a server warning message'
    - profile: my-xmpp-account
    - recipient: admins@xmpp.example.com/salt
```

name The message to send to the XMPP user

`salt.states.xmpp.`**`send_msg_multi`**`(`*name, profile, recipients=None, rooms=None*`)`

Send a message to an list of recipients or rooms

```
server-warning-message:
  xmpp.send_msg:
    - name: 'This is a server warning message'
    - profile: my-xmpp-account
    - recipients:
      - admins@xmpp.example.com/salt
    - rooms:
      - qa@conference.xmpp.example.com
```

name The message to send to the XMPP user

21.19.310 salt.states.zabbix_action

New in version 2017.7.

Management of Zabbix Action object over Zabbix API.

codeauthor Jakub Sliva <jakub.sliva@ultimum.io>

`salt.states.zabbix_action.`**`absent`**`(`*name, **kwargs*`)`

Makes the Zabbix Action to be absent (either does not exist or delete it).

Parameters

- **name** – Zabbix Action name
- **_connection_user** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **_connection_password** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **_connection_url** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

```
zabbix-action-absent:
    zabbix_action.absent:
        - name: Action name
```

salt.states.zabbix_action.**present**(*name*, *params*, ***kwargs*)

Creates Zabbix Action object or if differs update it according defined parameters

Parameters

- **name** – Zabbix Action name
- **params** – Definition of the Zabbix Action
- **_connection_user** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **_connection_password** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **_connection_url** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

If there is a need to get a value from current zabbix online (e.g. id of a hostgroup you want to put a discovered system into), put a dictionary with two keys "query_object" and "query_name" instead of the value. In this example we want to get object id of hostgroup named "Virtual machines" and "Databases".

```
zabbix-action-present:
    zabbix_action.present:
        - name: VMs
        - params:
            eventsource: 2
            status: 0
            filter:
                evaltype: 2
                conditions:
                    - conditiontype: 24
                      operator: 2
                      value: 'virtual'
                    - conditiontype: 24
                      operator: 2
                      value: 'kvm'
            operations:
                - operationtype: 2
                - operationtype: 4
                  opgroup:
                      - groupid:
                          query_object: hostgroup
                          query_name: Virtual machines
                      - groupid:
                          query_object: hostgroup
                          query_name: Databases
```

21.19.311 salt.states.zabbix_host module

Management of Zabbix hosts.

codeauthor Jiri Kotlin <jiri.kotlin@ultimum.io>

`salt.states.zabbix_host.`**`absent`**(*name*, ***kwargs*)

Ensures that the host does not exists, eventually deletes host.

New in version 2016.3.0.

Param name: technical name of the host

Parameters

- **`_connection_user`** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **`_connection_password`** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **`_connection_url`** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

```
TestHostWithInterfaces:
    zabbix_host.absent
```

`salt.states.zabbix_host.`**`assign_templates`**(*host*, *templates*, ***kwargs*)

Ensures that templates are assigned to the host.

New in version 2017.7.0.

Parameters

- **`host`** – technical name of the host
- **`_connection_user`** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **`_connection_password`** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **`_connection_url`** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

```
add_zabbix_templates_to_host:
    zabbix_host.assign_templates:
        - host: TestHost
        - templates:
            - "Template OS Linux"
            - "Template App MySQL"
```

`salt.states.zabbix_host.`**`present`**(*host*, *groups*, *interfaces*, ***kwargs*)

Ensures that the host exists, eventually creates new host. NOTE: please use argument visible_name instead of name to not mess with name from salt sls. This function accepts all standard host properties: keyword argument names differ depending on your zabbix version, see: https://www.zabbix.com/documentation/2.4/manual/api/reference/host/object#host

New in version 2016.3.0.

Parameters

- **`host`** – technical name of the host
- **`groups`** – groupids of host groups to add the host to
- **`interfaces`** – interfaces to be created for the host
- **`proxy_host`** – Optional proxy name or proxyid to monitor host
- **`inventory`** – Optional list of inventory names and values
- **`_connection_user`** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)

- **_connection_password** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **_connection_url** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)
- **visible_name** – Optional - string with visible name of the host, use 'visible_name' instead of 'name' parameter to not mess with value supplied from Salt sls file.

```
create_test_host:
    zabbix_host.present:
        - host: TestHostWithInterfaces
        - proxy_host: 12345
        - groups:
            - 5
            - 6
            - 7
        - interfaces:
            - test1.example.com:
                - ip: '192.168.1.8'
                - type: 'Agent'
                - port: 92
            - testing2_create:
                - ip: '192.168.1.9'
                - dns: 'test2.example.com'
                - type: 'agent'
                - main: false
            - testovaci1_ipmi:
                - ip: '192.168.100.111'
                - type: 'ipmi'
        - inventory:
            - alias: some alias
            - asset_tag: jlm3937
```

21.19.312 salt.states.zabbix_hostgroup module

Management of Zabbix host groups.

codeauthor Jiri Kotlin <jiri.kotlin@ultimum.io>

`salt.states.zabbix_hostgroup.`**`absent`**`(`*name*, ***kwargs*`)`

Ensures that the host group does not exist, eventually delete host group.

New in version 2016.3.0.

Parameters

- **name** – name of the host group
- **_connection_user** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **_connection_password** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **_connection_url** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

```
delete_testing_host_group:
    zabbix_hostgroup.absent:
        - name: 'My hostgroup name'
```

`salt.states.zabbix_hostgroup.`**`present`**`(`*name*, ***kwargs*`)`

Ensures that the host group exists, eventually creates new host group.

New in version 2016.3.0.

Parameters

- **name** – name of the host group
- **_connection_user** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **_connection_password** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **_connection_url** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

```
create_testing_host_group:
    zabbix_hostgroup.present:
        - name: 'My hostgroup name'
```

21.19.313 salt.states.zabbix_mediatype module

Management of Zabbix mediatypes.

codeauthor Raymond Kuiper <qix@the-wired.net>

`salt.states.zabbix_mediatype.`**`absent`**`(`*name*, ***kwargs*`)`

Ensures that the mediatype does not exist, eventually deletes the mediatype.

Parameters

- **name** – name of the mediatype
- **_connection_user** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **_connection_password** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **_connection_url** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

```
delete_mediatype:
    zabbix_mediatype.absent:
        - name: 'Email'
```

`salt.states.zabbix_mediatype.`**`present`**`(`*name*, *mediatype*, ***kwargs*`)`

Creates new mediatype. NOTE: This function accepts all standard mediatype properties: keyword argument names differ depending on your zabbix version, see: https://www.zabbix.com/documentation/3.0/manual/api/reference/host/object#host_inventory

Parameters

- **name** – name of the mediatype
- **_connection_user** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **_connection_password** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **_connection_url** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

```
make_new_mediatype:
    zabbix_mediatype.present:
        - name: 'Email'
        - mediatype: 0
        - smtp_server: smtp.example.com
        - smtp_hello: zabbix.example.com
        - smtp_email: zabbix@example.com
```

21.19.314 salt.states.zabbix_template

New in version 2017.7.

Management of Zabbix Template object over Zabbix API.

codeauthor Jakub Sliva <jakub.sliva@ultimum.io>

`salt.states.zabbix_template.`**`absent`**`(`*name*, ***kwargs*`)`

Makes the Zabbix Template to be absent (either does not exist or delete it).

Parameters

- **`name`** – Zabbix Template name
- **`_connection_user`** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **`_connection_password`** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **`_connection_url`** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

```
zabbix-template-absent:
    zabbix_template.absent:
        - name: Ceph OSD
```

`salt.states.zabbix_template.`**`is_present`**`(`*name*, ***kwargs*`)`

Check if Zabbix Template already exists.

Parameters

- **`name`** – Zabbix Template name
- **`_connection_user`** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **`_connection_password`** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **`_connection_url`** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

```
does_zabbix-template-exist:
    zabbix_template.is_present:
        - name: Template OS Linux
```

`salt.states.zabbix_template.`**`present`**`(`*name*, *params*, *static_host_list=True*, ***kwargs*`)`

Creates Zabbix Template object or if differs update it according defined parameters. See Zabbix API documentation.

Zabbix API version: >3.0

Parameters

- **`name`** – Zabbix Template name
- **`params`** – Additional parameters according to Zabbix API documentation
- **`static_host_list`** – If hosts assigned to the template are controlled only by this state or can be also assigned externally
- **`_connection_user`** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **`_connection_password`** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **`_connection_url`** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

Note: If there is a need to get a value from current zabbix online (e.g. ids of host groups you want the template to be associated with), put a dictionary with two keys "query_object" and "query_name" instead of the value.

In this example we want to create template named "Testing Template", assign it to hostgroup Templates, link it to two ceph nodes and create a macro.

Note: IMPORTANT NOTE: Objects (except for template name) are identified by name (or by other key in some exceptional cases) so changing name of object means deleting old one and creating new one with new ID !!!

Note:

NOT SUPPORTED FEATURES:

- linked templates
- trigger dependencies
- groups and group prototypes for host prototypes

SLS Example:

```
zabbix-template-present:
    zabbix_template.present:
        - name: Testing Template
        # Do not touch existing assigned hosts
        # True will detach all other hosts than defined here
        - static_host_list: False
        - params:
          description: Template for Ceph nodes
          groups:
              # groups must already exist
              # template must be at least in one hostgroup
              - groupid:
                  query_object: hostgroup
                  query_name: Templates
          macros:
              - macro: "{$CEPH_CLUSTER_NAME}"
                value: ceph
          hosts:
              # hosts must already exist
              - hostid:
                  query_object: host
                  query_name: ceph-osd-01
              - hostid:
                  query_object: host
                  query_name: ceph-osd-02
          # templates:
          # Linked templates - not supported by state module but can be linked
↪manually (will not be touched)

          applications:
              - name: Ceph OSD
          items:
              - name: Ceph OSD avg fill item
                key_: ceph.osd_avg_fill
                type: 2
                value_type: 0
                delay: 60
                units: '%'
```

(continues on next page)

(continued from previous page)

```
                  description: 'Average fill of OSD'
                  applications:
                      - applicationid:
                          query_object: application
                          query_name: Ceph OSD
            triggers:
                - description: "Ceph OSD filled more that 90%"
                  expression: "{{'{'}}Testing Template:ceph.osd_avg_fill.last(){{
↪'}'}}>90"
                  priority: 4
            discoveries:
                - name: Mounted filesystem discovery
                  key_: vfs.fs.discovery
                  type: 0
                  delay: 60
                  itemprototypes:
                      - name: Free disk space on {{'{#'}}FSNAME}
                        key_: vfs.fs.size[{{'{#'}}FSNAME},free]
                        type: 0
                        value_type: 3
                        delay: 60
                        applications:
                            - applicationid:
                                query_object: application
                                query_name: Ceph OSD
                  triggerprototypes:
                      - description: "Free disk space is less than 20% on volume {
↪{'{#'}}FSNAME{{'}'}}"
                        expression: "{{'{'}}Testing Template:vfs.fs.size[{{'{#'}}
↪FSNAME},free].last(){{'}'}}<20"
            graphs:
                - name: Ceph OSD avg fill graph
                  width: 900
                  height: 200
                  graphtype: 0
                  gitems:
                      - color: F63100
                        itemid:
                          query_object: item
                          query_name: Ceph OSD avg fill item
            screens:
                - name: Ceph
                  hsize: 1
                  vsize: 1
                  screenitems:
                      - x: 0
                        y: 0
                        resourcetype: 0
                        resourceid:
                            query_object: graph
                            query_name: Ceph OSD avg fill graph
```

21.19.315 salt.states.zabbix_user module

Management of Zabbix users.

codeauthor Jiri Kotlin <jiri.kotlin@ultimum.io>

`salt.states.zabbix_user.`**`absent`**`(`*name*, ***kwargs*`)`

Ensures that the user does not exist, eventually delete user.

New in version 2016.3.0.

Parameters

- **`name`** – user alias
- **`_connection_user`** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **`_connection_password`** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **`_connection_url`** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

```
George:
    zabbix_user.absent
```

`salt.states.zabbix_user.`**`admin_password_present`**`(`*name*, *password=None*, ***kwargs*`)`

Initial change of Zabbix Admin password to password taken from one of the sources (only the most prioritized one):

1. 'password' parameter
2. '_connection_password' parameter
3. pillar 'zabbix.password' setting

1) Tries to log in as Admin with password found in state password parameter or _connection_password or pillar or default zabbix password in this precise order, if any of them is present.
2) If one of above passwords matches, it tries to change the password to the most prioritized one.
3) If not able to connect with any password then it fails.

Parameters

- **`name`** – Just a name of state
- **`password`** – Optional - desired password for Admin to be set
- **`_connection_user`** – Optional - Ignored in this state (always assumed 'Admin')
- **`_connection_password`** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **`_connection_url`** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

```
# password taken from pillar or _connection_password
zabbix-admin-password:
    zabbix_user.admin_password_present

# directly set password
zabbix-admin-password:
    zabbix_user.admin_password_present:
        - password: SECRET_PASS
```

`salt.states.zabbix_user.`**`present`**`(`*alias*, *passwd*, *usrgrps*, *medias=None*, *password_reset=False*, ***kwargs*`)`

Ensures that the user exists, eventually creates new user. NOTE: use argument firstname instead of name to not mess values with name from salt sls.

New in version 2016.3.0.

Parameters

- **`alias`** – user alias
- **`passwd`** – user's password
- **`usrgrps`** – user groups to add the user to

- **medias** – Optional - user's medias to create
- **password_reset** – whether or not to reset password at update
- **_connection_user** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **_connection_password** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **_connection_url** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)
- **firstname** – string with firstname of the user, use 'firstname' instead of 'name' parameter to not mess with value supplied from Salt sls file.

```
make_user:
    zabbix_user.present:
        - alias: George
        - passwd: donottellanyonE@456x
        - password_reset: True
        - usrgrps:
            - 13
            - 7
        - medias:
            - me@example.com:
                - mediatype: mail
                - period: '1-7,00:00-24:00'
                - severity: NIWAHD
            - make_jabber:
                - active: true
                - mediatype: jabber
                - period: '1-5,08:00-19:00'
                - sendto: jabbera@example.com
            - text_me_morning_disabled:
                - active: false
                - mediatype: sms
                - period: '1-5,09:30-10:00'
                - severity: D
                - sendto: '+42032132588568'
```

21.19.316 salt.states.zabbix_usergroup module

Management of Zabbix user groups.

codeauthor Jiri Kotlin <jiri.kotlin@ultimum.io>

salt.states.zabbix_usergroup.**absent**(*name*, ***kwargs*)

Ensures that the user group does not exist, eventually delete user group.

New in version 2016.3.0.

Parameters

- **name** – name of the user group
- **_connection_user** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **_connection_password** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **_connection_url** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

```
delete_thai_monks_usrgrp:
    zabbix_usergroup.absent:
        - name: 'Thai monks'
```

`salt.states.zabbix_usergroup.`**`present`**`(`*name*, ***kwargs*`)`

Creates new user group. NOTE: This function accepts all standard user group properties: keyword argument names differ depending on your zabbix version, see: https://www.zabbix.com/documentation/2.0/manual/appendix/api/usergroup/definitions#user_group

New in version 2016.3.0.

Parameters

- **`name`** – name of the user group
- **`_connection_user`** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **`_connection_password`** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **`_connection_url`** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

```
make_new_thai_monks_usergroup:
    zabbix_usergroup.present:
        - name: 'Thai monks'
        - gui_access: 1
        - debug_mode: 0
        - users_status: 0
```

21.19.317 salt.states.zabbix_valuemap

New in version 2017.7.

Management of Zabbix Valuemap object over Zabbix API.

codeauthor Jakub Sliva <jakub.sliva@ultimum.io>

`salt.states.zabbix_valuemap.`**`absent`**`(`*name*, ***kwargs*`)`

Makes the Zabbix Value map to be absent (either does not exist or delete it).

Parameters

- **`name`** – Zabbix Value map name
- **`_connection_user`** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)
- **`_connection_password`** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
- **`_connection_url`** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

```
zabbix-valuemap-absent:
    zabbix_valuemap.absent:
        - name: Value map name
```

`salt.states.zabbix_valuemap.`**`present`**`(`*name*, *params*, ***kwargs*`)`

Creates Zabbix Value map object or if differs update it according defined parameters

Parameters

- **`name`** – Zabbix Value map name
- **`params`** – Definition of the Zabbix Value map
- **`_connection_user`** – Optional - zabbix user (can also be set in opts or pillar, see module's docstring)

  - **`_connection_password`** – Optional - zabbix password (can also be set in opts or pillar, see module's docstring)
  - **`_connection_url`** – Optional - url of zabbix frontend (can also be set in opts, pillar, see module's docstring)

```
zabbix-valuemap-present:
    zabbix_valuemap.present:
        - name: Number mapping
        - params:
          mappings:
              - value: 1
                newvalue: one
              - value: 2
                newvalue: two
```

21.19.318 salt.states.zcbuildout

Management of zc.buildout

This module is inspired from minitage's buildout maker (https://github.com/minitage/minitage/blob/master/src/minitage/core/makers/buildout.py)

New in version 2016.3.0.

Note: This state module is beta; the API is subject to change and no promise as to performance or functionality is yet present

---

Available Functions

- built

```
installed1
  buildout.installed:
    - name: /path/to/buildout

installed2
  buildout.installed:
    - name: /path/to/buildout
    - parts:
      - a
      - b
    - python: /path/to/pythonpath/bin/python
    - unless: /bin/test_something_installed
    - onlyif: /bin/test_else_installed
```

`salt.states.zcbuildout.`**`installed`**(*name, config='buildout.cfg', quiet=False, parts=None, user=None, env=(), buildout_ver=None, test_release=False, distribute=None, new_st=None, offline=False, newest=False, python='/builds/saltstack/open/docs/builddocs/venv/bin/python', debug=False, verbose=False, unless=None, onlyif=None, use_vt=False, loglevel='debug', **kwargs*)

Install buildout in a specific directory

It is a thin wrapper to modules.buildout.buildout

name directory to execute in

quiet

do not output console & logs

config buildout config to use (default: buildout.cfg)

parts specific buildout parts to run

user user used to run buildout as

New in version 2014.1.4.

env environment variables to set when running

buildout_ver force a specific buildout version (1 | 2)

test_release buildout accept test release

new_st Forcing use of setuptools >= 0.7

distribute use distribute over setuptools if possible

offline does buildout run offline

python python to use

debug run buildout with -D debug flag

onlyif Only execute cmd if statement on the host return 0

unless Do not execute cmd if statement on the host return 0

newest run buildout in newest mode

verbose run buildout in verbose mode (-vvvvv)

use_vt Use the new salt VT to stream output [experimental]

loglevel loglevel for buildout commands

21.19.319 salt.states.zenoss

State to manage monitoring in Zenoss.

New in version 2016.3.0.

This state module depends on the 'zenoss' Salt execution module.

Allows for setting a state of minions in Zenoss using the Zenoss API. Currently Zenoss 4.x and 5.x are supported.

```
enable_monitoring:
  zenoss.monitored:
    - name: web01.example.com
    - device_class: /Servers/Linux
    - collector: localhost
    - prod_state: 1000
```

`salt.states.zenoss.`**`monitored`**(*name*, *device_class=None*, *collector='localhost'*, *prod_state=None*)

Ensure a device is monitored. The 'name' given will be used for Zenoss device name and should be resolvable.

```
enable_monitoring:
  zenoss.monitored:
    - name: web01.example.com
    - device_class: /Servers/Linux
    - collector: localhost
    - prod_state: 1000
```

21.19.320 salt.states.zk_concurrency

Control concurrency of steps within state execution using zookeeper

depends kazoo

configuration See `salt.modules.zookeeper` for setup instructions.

This module allows you to "wrap" a state's execution with concurrency control. This is useful to protect against all hosts executing highstate simultaneously if your services don't all HUP restart. The common way of protecting against this is to run in batch mode, but that doesn't protect from another person running the same batch command (and thereby having 2x the number of nodes deploying at once).

This module will bock while acquiring a slot, meaning that however the command gets called it will coordinate with zookeeper to ensure that no more than max_concurrency steps are executing with a single path.

```
acquire_lock:
  zk_concurrency.lock:
    - name: /trafficserver
    - zk_hosts: 'zookeeper:2181'
    - max_concurrency: 4
    - prereq:
        - service: trafficserver

trafficserver:
  service.running:
    - watch:
      - file: /etc/trafficserver/records.config

/etc/trafficserver/records.config:
  file.managed:
    - source: salt://records.config

release_lock:
  zk_concurrency.unlock:
    - name: /trafficserver
    - require:
        - service: trafficserver
```

This example would allow the file state to change, but would limit the concurrency of the trafficserver service restart to 4.

`salt.states.zk_concurrency.`**`lock`**(*name, zk_hosts=None, identifier=None, max_concurrency=1, timeout=None, ephemeral_lease=False, profile=None, scheme=None, username=None, password=None, default_acl=None*)

Block state execution until you are able to get the lock (or hit the timeout)

`salt.states.zk_concurrency.`**`min_party`**(*name, zk_hosts, min_nodes, blocking=False, profile=None, scheme=None, username=None, password=None, default_acl=None*)

Ensure that there are *min_nodes* in the party at *name*, optionally blocking if not available.

`salt.states.zk_concurrency.`**`unlock`**(*name, zk_hosts=None, identifier=None, max_concurrency=1, ephemeral_lease=False, profile=None, scheme=None, username=None, password=None, default_acl=None*)

Remove lease from semaphore.

21.19.321 salt.states.zfs

States for managing zfs datasets

maintainer Jorge Schrauwen <sjorge@blackdot.be>

maturity new

depends salt.utils.zfs, salt.modules.zfs

platform smartos, illumos, solaris, freebsd, linux

New in version 2016.3.0.

Changed in version 2018.3.1: Big refactor to remove duplicate code, better type converions and improved consistancy in output.

```
test/shares/yuki:
  zfs.filesystem_present:
    - create_parent: true
    - properties:
        quota: 16G

test/iscsi/haruhi:
  zfs.volume_present:
    - create_parent: true
    - volume_size: 16M
    - sparse: true
    - properties:
        readonly: on

test/shares/yuki@frozen:
  zfs.snapshot_present

moka_origin:
  zfs.hold_present:
    - snapshot: test/shares/yuki@frozen

test/shares/moka:
  zfs.filesystem_present:
    - cloned_from: test/shares/yuki@frozen

test/shares/moka@tsukune:
  zfs.snapshot_absent
```

`salt.states.zfs.`**`bookmark_absent`**(*name*, *force=False*, *recursive=False*)

ensure bookmark is absent on the system

name [string] name of snapshot

force [boolean] try harder to destroy the dataset (zfs destroy -f)

recursive [boolean] also destroy all the child datasets (zfs destroy -r)

`salt.states.zfs.`**`bookmark_present`**(*name*, *snapshot*)

ensure bookmark exists

name [string] name of bookmark

snapshot [string] name of snapshot

`salt.states.zfs.`**`filesystem_absent`**(*name*, *force=False*, *recursive=False*)

ensure filesystem is absent on the system

name [string] name of filesystem

force [boolean] try harder to destroy the dataset (zfs destroy -f)

recursive [boolean] also destroy all the child datasets (zfs destroy -r)

> **Warning:** If a volume with `name` exists, this state will succeed without destroying the volume specified by `name`. This module is dataset type sensitive.

`salt.states.zfs.`**`filesystem_present`**(*name*, *create_parent=False*, *properties=None*, *cloned_from=None*)

ensure filesystem exists and has properties set

name [string] name of filesystem

create_parent [boolean] creates all the non-existing parent datasets. any property specified on the command line using the -o option is ignored.

cloned_from [string] name of snapshot to clone

properties [dict] additional zfs properties (-o)

> **Note:** `cloned_from` is only use if the filesystem does not exist yet, when `cloned_from` is set after the filesystem exists it will be ignored.

> **Note:** Properties do not get cloned, if you specify the properties in the state file they will be applied on a subsequent run.

`salt.states.zfs.`**`hold_absent`**(*name*, *snapshot*, *recursive=False*)

ensure hold is absent on the system

name [string] name of hold

snapshot [string] name of snapshot

recursive [boolean] recursively releases a hold with the given tag on the snapshots of all descendent file systems.

`salt.states.zfs.`**`hold_present`**(*name*, *snapshot*, *recursive=False*)

ensure hold is present on the system

name [string] name of holdt

snapshot [string] name of snapshot

recursive [boolean] recursively add hold with the given tag on the snapshots of all descendent file systems.

`salt.states.zfs.`**`promoted`**(*name*)

ensure a dataset is not a clone

name [string] name of fileset or volume

> **Warning:** only one dataset can be the origin, if you promote a clone the original will now point to the promoted dataset

`salt.states.zfs.`**`scheduled_snapshot`**(*name*, *prefix*, *recursive=True*, *schedule=None*)

maintain a set of snapshots based on a schedule

name [string] name of filesystem or volume

prefix [string] prefix for the snapshots e.g. 'test' will result in snapshots being named 'test-yyyymmdd_hhmm'

recursive [boolean] create snapshots for all children also

schedule [dict] dict holding the schedule, the following keys are available (minute, hour, day, month, and year) by default all are set to 0 the value indicated the number of snapshots of that type to keep around.

> **Warning:** snapshots will only be created and pruned every time the state runs. a schedule must be setup to automatically run the state. this means that if you run the state daily the hourly snapshot will only be made once per day!

Changed in version 2018.3.0: switched to localtime from gmtime so times now take into account timezones.

`salt.states.zfs.`**`snapshot_absent`**`(`*name*, *force=False*, *recursive=False*`)`

ensure snapshot is absent on the system

name [string] name of snapshot

force [boolean] try harder to destroy the dataset (zfs destroy -f)

recursive [boolean] also destroy all the child datasets (zfs destroy -r)

`salt.states.zfs.`**`snapshot_present`**`(`*name*, *recursive=False*, *properties=None*`)`

ensure snapshot exists and has properties set

name [string] name of snapshot

recursive [boolean] recursively create snapshots of all descendent datasets

properties [dict] additional zfs properties (-o)

`salt.states.zfs.`**`volume_absent`**`(`*name*, *force=False*, *recursive=False*`)`

ensure volume is absent on the system

name [string] name of volume

force [boolean] try harder to destroy the dataset (zfs destroy -f)

recursive [boolean] also destroy all the child datasets (zfs destroy -r)

Warning: If a filesystem with `name` exists, this state will succeed without destroying the filesystem specified by `name`. This module is dataset type sensitive.

`salt.states.zfs.`**`volume_present`**`(`*name*, *volume_size*, *sparse=False*, *create_parent=False*, *properties=None*, *cloned_from=None*`)`

ensure volume exists and has properties set

name [string] name of volume

volume_size [string] size of volume

sparse [boolean] create sparse volume

create_parent [boolean] creates all the non-existing parent datasets. any property specified on the command line using the -o option is ignored.

cloned_from [string] name of snapshot to clone

properties [dict] additional zfs properties (-o)

Note: `cloned_from` is only use if the volume does not exist yet, when `cloned_from` is set after the volume exists it will be ignored.

Note: Properties do not get cloned, if you specify the properties in the state file they will be applied on a subsequent run.

`volume_size` is considered a property, so the volume's size will be corrected when the properties get updated if it differs from the original volume.

The sparse parameter is ignored when using `cloned_from`.

21.19.322 salt.states.zone

Management of Solaris Zones

maintainer Jorge Schrauwen <sjorge@blackdot.be>

maturity new

depends salt.modules.zoneadm, salt.modules.zonecfg

platform solaris

New in version 2017.7.0.

Below are some examples of how to use this state. Lets start with creating a zone and installing it.

```
omipkg1_configuration:
  zone.present:
    - name: omipkg1
    - brand: ipkg
    - zonepath: /zones/omipkg1
    - properties:
      - autoboot: true
      - ip-type: exclusive
      - cpu-shares: 50
    - resources:
      - attr:
        - name: owner
        - value: Jorge Schrauwen
        - type: string
      - attr:
        - name: description
        - value: OmniOS ipkg zone for testing
        - type: string
      - capped-memory:
        - physical: 64M
omipkg1_installation:
  zone.installed:
    - name: omipkg1
    - require:
        - zone: omipkg1_configuration
omipkg1_running:
  zone.booted:
    - name: omipkg1
    - require:
        - zone: omipkg1_installation
```

A zone without network access is not very useful. We could update the zone.present state in the example above to add a network interface or we could use a separate state for this.

```
omipkg1_network:
  zone.resource_present:
    - name: omipkg1
    - resource_type: net
    - resource_selector_property: mac-addr
    - resource_selector_value: "02:08:20:a2:a3:10"
    - physical: znic1
    - require:
        - zone: omipkg1_configuration
```

Since this is a single tenant system having the owner attribute is pointless. Let's remove that attribute.

Note: The following state run the omipkg1_configuration state will add it again! If the entire configuration is managed it would be better to add resource_prune and optionally the resource_selector_property properties to the resource.

```
omipkg1_strip_owner:
  zone.resource_present:
    - name: omipkg1
    - resource_type: attr
    - resource_selector_property: name
    - resource_selector_value: owner
    - require:
        - zone: omipkg1_configuration
```

Let's bump the zone's CPU shares a bit.

Note: The following state run the omipkg1_configuration state will set it to 50 again. Update the entire zone configuration is managed you should update it there instead.

```
omipkg1_more_cpu:
  zone.property_present:
    - name: omipkg1
    - property: cpu-shares
    - value: 100
```

Or we can remove the limit altogether!

Note: The following state run the omipkg1_configuration state will set it to 50 again. Update the entire zone configuration is managed you should set the property to None (nothing after the :) instead.

```
omipkg1_no_cpu:
  zone.property_absent:
    - name: omipkg1
    - property: cpu-shares
```

`salt.states.zone.`**`absent`**`(`*name, uninstall=False*`)`

Ensure a zone is absent

name [string] name of the zone

uninstall [boolean] when true, uninstall instead of detaching the zone first.

`salt.states.zone.`**`attached`**`(`*name, force=False*`)`

Ensure zone is attached

name [string] name of the zone

force [boolean] force attach the zone

`salt.states.zone.`**`booted`**`(`*name, single=False*`)`

Ensure zone is booted

name [string] name of the zone

single [boolean] boot in single usermode

`salt.states.zone.`**`detached`**`(`*name*`)`

Ensure zone is detached

name [string] name of the zone

`salt.states.zone.`**`export`**`(`*name, path, replace=False*`)`

Export a zones configuration

name [string] name of the zone

path [string] path of file to export too.

replace [boolean] replace the file if it exists `salt.states.zone.`**`halted`**`(`*name*, *graceful=True*`)`

Ensure zone is halted

name [string] name of the zone

graceful [boolean] use shutdown instead of halt if true

`salt.states.zone.`**`import_`**`(`*name*, *path*, *mode='import'*, *nodataset=False*, *brand_opts=None*`)`

Import a zones configuration

name [string] name of the zone

path [string] path of the configuration file to import

mode [string] either import, install, or attach

nodataset [boolean] do not create a ZFS file system

brand_opts [boolean] brand specific options to pass

Note: The mode argument can be set to `import`, `install`, or `attach`. `import`: will only import the configuration `install`: will import and then try to install the zone `attach`: will import and then try to attach of the zone

```
omipkg1:
  zone.import:
    - path: /foo/bar/baz
```

`salt.states.zone.`**`installed`**`(`*name*, *nodataset=False*, *brand_opts=None*`)`

Ensure zone is installed

name [string] name of the zone

nodataset [boolean] do not create a ZFS file system

brand_opts [boolean] brand specific options to pass

`salt.states.zone.`**`present`**`(`*name*, *brand*, *zonepath*, *properties=None*, *resources=None*`)`

Ensure a zone with certain properties and resources

name [string] name of the zone

brand [string] brand of the zone

zonepath [string] path of the zone

properties [list of key-value pairs] dict of properties

resources [list of key-value pairs] dict of resources

Note: If the zone does not exist it will not be installed. You can use the ``zone.installed`` state for this.

Note:

Default resource selectors:

- fs: dir
- net: mac-addr
- device: match
- rctl: name
- attr: name
- dataset: name
- admin: user

Warning: Properties and resource will not be removed when they are absent from the state!

For properties, simple set them to ``None``.

For resources, add the `` `resource_prune` `` property and set it to `` `True` ``. Also specify the `` `resource_selector_property` `` if the default is not the one you want.

`salt.states.zone.`**`property_absent`**`(`*name*, *property*`)`

Ensure property is absent

name [string] name of the zone

property [string] name of property

Note: This does a zoneacfg clear call. So the property may be reset to a default value! Does has the side effect of always having to be called.

`salt.states.zone.`**`property_present`**`(`*name*, *property*, *value*`)`

Ensure property has a certain value

name [string] name of the zone

property [string] name of property

value [string] value of property

`salt.states.zone.`**`resource_absent`**`(`*name*, *resource_type*, *resource_selector_property*, *resource_selector_value*`)`

Ensure resource is absent

name [string] name of the zone

resource_type [string] type of resource

resource_selector_property [string] unique resource identifier

resource_selector_value [string] value for resource selection

Warning: Both resource_selector_property and resource_selector_value must be provided, some properties like `` `name` `` are already reserved by salt in there states.

Note: You can set both resource_selector_property and resource_selector_value to None for resources that do not require them.

`salt.states.zone.`**`resource_present`**`(`*name*, *resource_type*, *resource_selector_property*, *resource_selector_value*, ***kwargs*`)`

Ensure resource exists with provided properties

name [string] name of the zone

resource_type [string] type of resource

resource_selector_property [string] unique resource identifier

resource_selector_value [string] value for resource selection

kwargs [string|int|...] resource properties

Warning: Both resource_selector_property and resource_selector_value must be provided, some properties like `name` are already reserved by salt in states.

Note: You can set both resource_selector_property and resource_selector_value to None for resources that do not require them.

`salt.states.zone.`**`uninstalled`**`(`*name*`)`

Ensure zone is uninstalled

name [string] name of the zone

21.19.323 salt.states.zookeeper

depends kazoo

configuration See `salt.modules.zookeeper` for setup instructions.

ACLS

For more information about acls, please checkout the kazoo documentation.

http://kazoo.readthedocs.io/en/latest/api/security.html#kazoo.security.make_digest_acl

The following options can be included in the acl dictionary:

param username Username to use for the ACL.

param password A plain-text password to hash.

param write Write permission.

type write bool

param create Create permission.

type create bool

param delete Delete permission.

type delete bool

param admin Admin permission.

type admin bool

param all All permissions.

type all bool

`salt.states.zookeeper.`**`absent`**`(`*name, version=-1, recursive=False, profile=None, hosts=None, scheme=None, username=None, password=None, default_acl=None*`)`

Make sure znode is absent
name path to znode
version Specify the version which should be deleted Default: -1 (always match)
recursive Boolean to indicate if children should be recursively deleted Default: False
profile Configured Zookeeper profile to authenticate with (Default: None)
hosts Lists of Zookeeper Hosts (Default: '127.0.0.1:2181)
scheme Scheme to authenticate with (Default: 'digest')
username Username to authenticate (Default: None)
password Password to authenticate (Default: None)
default_acl Default acls to assign if a node is created in this connection (Default: None)

```
delete znode:
  zookeeper.absent:
    - name: /test
    - recursive: True
```

`salt.states.zookeeper.`**`acls`**`(`*name, acls, version=-1, profile=None, hosts=None, scheme=None, username=None, password=None, default_acl=None*`)`

Update acls on a znode

name path to znode
acls list of acl dictionaries to set on znode
version Specify the version which should be deleted Default: -1 (always match)
profile Configured Zookeeper profile to authenticate with (Default: None)
hosts Lists of Zookeeper Hosts (Default: '127.0.0.1:2181)
scheme Scheme to authenticate with (Default: 'digest')
username Username to authenticate (Default: None)
password Password to authenticate (Default: None)
default_acl Default acls to assign if a node is created in this connection (Default: None)

```
update acls:
  zookeeper.acls:
    - name: /test/name
    - acls:
      - username: daniel
        password: test
        all: True
      - username: gtmanfred
        password: test
        all: True
```

`salt.states.zookeeper.`**`present`**`(`*name, value, acls=None, ephemeral=False, sequence=False, makepath=False, version=-1, profile=None, hosts=None, scheme=None, username=None, password=None, default_acl=None*`)`

Make sure znode is present in the correct state with the correct acls
name path to znode
value value znode should be set to
acls list of acl dictionaries to set on znode (make sure the ones salt is connected with are included) Default: None
ephemeral Boolean to indicate if ephemeral znode should be created Default: False
sequence Boolean to indicate if znode path is suffixed with a unique index Default: False
makepath Boolean to indicate if the parent paths should be created Default: False
version For updating, specify the version which should be updated Default: -1 (always match)
profile Configured Zookeeper profile to authenticate with (Default: None)
hosts Lists of Zookeeper Hosts (Default: '127.0.0.1:2181)
scheme Scheme to authenticate with (Default: 'digest')
username Username to authenticate (Default: None)
password Password to authenticate (Default: None)
default_acl Default acls to assign if a node is created in this connection (Default: None)

```
add znode:
  zookeeper.present:
    - name: /test/name
    - value: gtmanfred
    - makepath: True

update znode:
  zookeeper.present:
    - name: /test/name
    - value: daniel
    - acls:
      - username: daniel
        password: test
        read: true
      - username: gtmanfred
```

(continues on next page)

(continued from previous page)

```
        password: test
        read: true
        write: true
        create: true
        delete: true
        admin: true
    - makepath: True
```

21.19.324 salt.states.zpool

States for managing zpools

maintainer Jorge Schrauwen <sjorge@blackdot.be>

maturity new

depends salt.utils.zfs, salt.modules.zpool

platform smartos, illumos, solaris, freebsd, linux

New in version 2016.3.0.

Changed in version 2018.3.1: Big refactor to remove duplicate code, better type conversions and improved consistency in output.

```
oldpool:
  zpool.absent:
    - export: true

newpool:
  zpool.present:
    - config:
        import: false
        force: true
    - properties:
        comment: salty storage pool
    - layout:
        - mirror:
          - /dev/disk0
          - /dev/disk1
        - mirror:
          - /dev/disk2
          - /dev/disk3

partitionpool:
  zpool.present:
    - config:
        import: false
        force: true
    - properties:
        comment: disk partition salty storage pool
        ashift: '12'
        feature@lz4_compress: enabled
    - filesystem_properties:
        compression: lz4
        atime: on
        relatime: on
```

(continues on next page)

(continued from previous page)

```
    - layout:
        - /dev/disk/by-uuid/3e43ce94-77af-4f52-a91b-6cdbb0b0f41b

simplepool:
  zpool.present:
    - config:
        import: false
        force: true
    - properties:
        comment: another salty storage pool
    - layout:
        - /dev/disk0
        - /dev/disk1
```

> **Warning:** The layout will never be updated, it will only be used at time of creation. It's a whole lot of work to figure out if a devices needs to be detached, removed, etc. This is best done by the sysadmin on a case per case basis.
>
> Filesystem properties are also not updated, this should be managed by the zfs state module.

`salt.states.zpool.`**`absent`**`(`*name*, *export=False*, *force=False*`)`

ensure storage pool is absent on the system

name [string] name of storage pool

export [boolean] export instead of destroy the zpool if present

force [boolean] force destroy or export

`salt.states.zpool.`**`present`**`(`*name*, *properties=None*, *filesystem_properties=None*, *layout=None*, *config=None*`)`

ensure storage pool is present on the system

name [string] name of storage pool

properties [dict] optional set of properties to set for the storage pool

filesystem_properties [dict] optional set of filesystem properties to set for the storage pool (creation only)

layout: dict disk layout to use if the pool does not exist (creation only)

config [dict] fine grain control over this state

Note:

The following configuration properties can be toggled in the config parameter.

- import (true) - try to import the pool before creating it if absent
- import_dirs (None) - specify additional locations to scan for devices on import (comma-seperated)
- device_dir (None, SunOS=/dev/dsk, Linux=/dev) - specify device directory to prepend for none absolute device paths
- force (false) - try to force the import or creation

Note: It is no longer needed to give a unique name to each top-level vdev, the old layout format is still supported but no longer recommended.

```
- mirror:
  - /tmp/vdisk3
  - /tmp/vdisk2
- mirror:
  - /tmp/vdisk0
  - /tmp/vdisk1
```

The above yaml will always result in the following zpool create:

```
zpool create mypool mirror /tmp/vdisk3 /tmp/vdisk2 mirror /tmp/vdisk0 /tmp/vdisk1
```

Warning: The legacy format is also still supported but not recommended, because ID's inside the layout dict must be unique they need to have a suffix.

```
mirror-0:
  /tmp/vdisk3
  /tmp/vdisk2
mirror-1:
  /tmp/vdisk0
  /tmp/vdisk1
```

Warning: Pay attention to the order of your dict!

```
- mirror:
  - /tmp/vdisk0
  - /tmp/vdisk1
- /tmp/vdisk2
```

The above will result in the following zpool create:

```
zpool create mypool mirror /tmp/vdisk0 /tmp/vdisk1 /tmp/vdisk2
```

Creating a 3-way mirror! While you probably expect it to be mirror root vdev with 2 devices + a root vdev of 1 device!

21.20 thorium modules

calc	Used to manage the thorium register.
check	The check Thorium state is used to create gateways to commands, the checks make it easy to make states that watch registers for changes and then just succeed or fail based on the state of the register, this creates the pattern of having a command execution get gated by a check state via a requisite.
file	Writes matches to disk to verify activity, helpful when testing
key	The key Thorium State is used to apply changes to the accepted/rejected/pending keys
local	Run remote execution commands via the local client
reg	Used to manage the thorium register.
runner	React by calling asynchronous runners
status	This thorium state is used to track the status beacon events and keep track of the active status of minions
timer	Allow for flow based timers.
wheel	React by calling asynchronous runners

21.20.1 salt.thorium.calc module

Used to manage the thorium register. The thorium register is where compound values are stored and computed, such as averages etc.

New in version 2016.11.0.

depends statistics PyPi module

`salt.thorium.calc.`**`add`**`(`*name, num, minimum=0, maximum=0, ref=None*`)`

Adds together the `num` most recent values. Requires a list.

USAGE:

```
foo:
  calc.add:
    - name: myregentry
    - num: 5
```

`salt.thorium.calc.`**`calc`**`(`*name, num, oper, minimum=0, maximum=0, ref=None*`)`

Perform a calculation on the `num` most recent values. Requires a list. Valid values for `oper` are:

- add: Add last `num` values together
- mul: Multiple last `num` values together
- mean: Calculate mean of last `num` values
- median: Calculate median of last `num` values
- median_low: Calculate low median of last `num` values
- median_high: Calculate high median of last `num` values
- median_grouped: Calculate grouped median of last `num` values
- mode: Calculate mode of last `num` values

USAGE:

```
foo:
  calc.calc:
    - name: myregentry
    - num: 5
    - oper: mean
```

`salt.thorium.calc.`**`mean`**`(`*name, num, minimum=0, maximum=0, ref=None*`)`

Calculates the mean of the `num` most recent values. Requires a list.

USAGE:

```
foo:
  calc.mean:
    - name: myregentry
    - num: 5
```

`salt.thorium.calc.`**`median`**`(`*name, num, minimum=0, maximum=0, ref=None*`)`

Calculates the mean of the `num` most recent values. Requires a list.

USAGE:

```
foo:
  calc.median:
    - name: myregentry
    - num: 5
```

`salt.thorium.calc.`**`median_grouped`**`(`*name, num, minimum=0, maximum=0, ref=None*`)`

Calculates the grouped mean of the `num` most recent values. Requires a list.

USAGE:

```
foo:
  calc.median_grouped:
    - name: myregentry
    - num: 5
```

`salt.thorium.calc.`**`median_high`**`(`*name, num, minimum=0, maximum=0, ref=None*`)`

Calculates the high mean of the `num` most recent values. Requires a list.

USAGE:

```
foo:
  calc.median_high:
    - name: myregentry
    - num: 5
```

`salt.thorium.calc.`**`median_low`**`(`*name, num, minimum=0, maximum=0, ref=None*`)`

Calculates the low mean of the `num` most recent values. Requires a list.

USAGE:

```
foo:
  calc.median_low:
    - name: myregentry
    - num: 5
```

`salt.thorium.calc.`**`mode`**`(`*name, num, minimum=0, maximum=0, ref=None*`)`

Calculates the mode of the `num` most recent values. Requires a list.

USAGE:

```
foo:
  calc.mode:
    - name: myregentry
    - num: 5
```

`salt.thorium.calc.`**`mul`**`(`*name, num, minimum=0, maximum=0, ref=None*`)`

Multiplies together the `num` most recent values. Requires a list.

USAGE:

```
foo:
  calc.mul:
    - name: myregentry
    - num: 5
```

21.20.2 salt.thorium.check module

The check Thorium state is used to create gateways to commands, the checks make it easy to make states that watch registers for changes and then just succeed or fail based on the state of the register, this creates the pattern of having a command execution get gated by a check state via a requisite.

`salt.thorium.check.`**`contains`**`(`*name, value, count_lt=None, count_lte=None, count_eq=None, count_gte=None, count_gt=None, count_ne=None*`)`

Only succeed if the value in the given register location contains the given value

USAGE:

```
foo:
  check.contains:
    - value: itni

run_remote_ex:
  local.cmd:
    - tgt: '*'
    - func: test.ping
    - require:
      - check: foo
```

`salt.thorium.check.`**`eq`**`(`*name*, *value*`)`

Only succeed if the value in the given register location is equal to the given value

USAGE:

```
foo:
  check.eq:
    - value: 42

run_remote_ex:
  local.cmd:
    - tgt: '*'
    - func: test.ping
    - require:
      - check: foo
```

`salt.thorium.check.`**`event`**`(`*name*`)`

Chekcs for a specific event match and returns result True if the match happens

USAGE:

```
salt/foo/*/bar:
  check.event

run_remote_ex:
  local.cmd:
    - tgt: '*'
    - func: test.ping
    - require:
      - check: salt/foo/*/bar
```

`salt.thorium.check.`**`gt`**`(`*name*, *value*`)`

Only succeed if the value in the given register location is greater than the given value

USAGE:

```
foo:
  check.gt:
    - value: 42

run_remote_ex:
  local.cmd:
    - tgt: '*'
    - func: test.ping
    - require:
      - check: foo
```

`salt.thorium.check.`**`gte`**`(`*name*, *value*`)`

Only succeed if the value in the given register location is greater or equal than the given value

USAGE:

```
foo:
  check.gte:
    - value: 42

run_remote_ex:
  local.cmd:
    - tgt: '*'
    - func: test.ping
    - require:
      - check: foo
```

`salt.thorium.check.`**`len_eq`**`(`*name*, *value*`)`

Only succeed if the length of the given register location is equal to the given value.

USAGE:

```
foo:
  check.len_eq:
    - value: 42

run_remote_ex:
  local.cmd:
    - tgt: '*'
    - func: test.ping
    - require:
      - check: foo
```

`salt.thorium.check.`**`len_gt`**`(`*name*, *value*`)`

Only succeed if length of the given register location is greater than the given value.

USAGE:

```
foo:
  check.len_gt:
    - value: 42

run_remote_ex:
  local.cmd:
    - tgt: '*'
    - func: test.ping
    - require:
      - check: foo
```

`salt.thorium.check.`**`len_gte`**`(`*name*, *value*`)`

Only succeed if the length of the given register location is greater or equal than the given value

USAGE:

```
foo:
  check.len_gte:
    - value: 42

run_remote_ex:
  local.cmd:
```

(continues on next page)

(continued from previous page)

```
    - tgt: '*'
    - func: test.ping
    - require:
      - check: foo
```

salt.thorium.check.**len_lt**(*name*, *value*)

Only succeed if the length of the given register location is less than the given value.

USAGE:

```
foo:
  check.len_lt:
    - value: 42

run_remote_ex:
  local.cmd:
    - tgt: '*'
    - func: test.ping
    - require:
      - check: foo
```

salt.thorium.check.**len_lte**(*name*, *value*)

Only succeed if the length of the given register location is less than or equal the given value

USAGE:

```
foo:
  check.len_lte:
    - value: 42

run_remote_ex:
  local.cmd:
    - tgt: '*'
    - func: test.ping
    - require:
      - check: foo
```

salt.thorium.check.**len_ne**(*name*, *value*)

Only succeed if the length of the given register location is not equal to the given value.

USAGE:

```
foo:
  check.len_ne:
    - value: 42

run_remote_ex:
  local.cmd:
    - tgt: '*'
    - func: test.ping
    - require:
      - check: foo
```

salt.thorium.check.**lt**(*name*, *value*)

Only succeed if the value in the given register location is less than the given value

USAGE:

```
foo:
  check.lt:
    - value: 42

run_remote_ex:
  local.cmd:
    - tgt: '*'
    - func: test.ping
    - require:
      - check: foo
```

salt.thorium.check.**lte**(*name*, *value*)

Only succeed if the value in the given register location is less than or equal the given value

USAGE:

```
foo:
  check.lte:
    - value: 42

run_remote_ex:
  local.cmd:
    - tgt: '*'
    - func: test.ping
    - require:
      - check: foo
```

salt.thorium.check.**ne**(*name*, *value*)

Only succeed if the value in the given register location is not equal to the given value

USAGE:

```
foo:
  check.ne:
    - value: 42

run_remote_ex:
  local.cmd:
    - tgt: '*'
    - func: test.ping
    - require:
      - check: foo
```

21.20.3 salt.thorium.file module

Writes matches to disk to verify activity, helpful when testing

Normally this is used by giving the name of the file (without a path) that the data will be saved to. If for instance you use `foo` as the name:

```
foo:
  file.save
```

Then the file will be saved to:

```
<salt cachedir>/thorium/saves/foo
```

You may also provide an absolute path for the file to be saved to:

```
/tmp/foo.save:
    file.save
```

Files will be saved in JSON format. However, JSON does not support `set()``s. If you are saving a register entry that contains a ``set()`, then it will fail to save to JSON format. However, you may pass data through a filter which makes it JSON compliant:

```
foo:
  file.save:
    filter: True
```

Be warned that if you do this, then the file will be saved, but not in a format that can be re-imported into Python.

`salt.thorium.file.`**`save`**`(`*name, filter=False*`)`

Save the register to <salt cachedir>/thorium/saves/<name>, or to an absolute path.

If an absolute path is specified, then the directory will be created non-recursively if it doesn't exist.

USAGE:

```
foo:
  file.save

/tmp/foo:
  file.save
```

21.20.4 salt.thorium.key module

The key Thorium State is used to apply changes to the accepted/rejected/pending keys

New in version 2016.11.0.

`salt.thorium.key.`**`timeout`**`(`*name, delete=0, reject=0*`)`

If any minion's status is older than the timeout value then apply the given action to the timed out key. This example will remove keys to minions that have not checked in for 300 seconds (5 minutes)

USAGE:

```
statreg:
  status.reg

clean_keys:
  key.timeout:
    - require:
      - status: statreg
    - delete: 300
```

21.20.5 salt.thorium.local module

Run remote execution commands via the local client

`salt.thorium.local.`**`cmd`**`(`*name, tgt, func, arg=(), tgt_type='glob', ret='', kwarg=None, **kwargs*`)`

Execute a remote execution command

USAGE:

```
run_remote_ex:
  local.cmd:
    - tgt: '*'
    - func: test.ping

run_remote_ex:
  local.cmd:
    - tgt: '*'
    - func: test.sleep
    - arg:
      - 30

run_remote_ex:
  local.cmd:
    - tgt: '*'
    - func: test.sleep
    - kwarg:
      length: 30
```

21.20.6 salt.thorium.reg module

Used to manage the thorium register. The thorium register is where compound values are stored and computed, such as averages etc.

`salt.thorium.reg.`**`clear`**`(`*name*`)`

Clear the namespace from the register

USAGE:

```
clearns:
  reg.clear:
    - name: myregister
```

`salt.thorium.reg.`**`delete`**`(`*name*`)`

Delete the namespace from the register

USAGE:

```
deletens:
  reg.delete:
    - name: myregister
```

`salt.thorium.reg.`**`list_`**`(`*name, add, match, stamp=False, prune=0*`)`

Add the specified values to the named list

If `stamp` is True, then the timestamp from the event will also be added if `prune` is set to an integer higher than `0`, then only the last `prune` values will be kept in the list.

USAGE:

```
foo:
  reg.list:
    - add: bar
    - match: my/custom/event
    - stamp: True
```

`salt.thorium.reg.`**`mean`**`(`*name*, *add*, *match*`)`

Accept a numeric value from the matched events and store a running average of the values in the given register. If the specified value is not numeric it will be skipped

USAGE:

```
foo:
  reg.mean:
    - add: data_field
    - match: my/custom/event
```

`salt.thorium.reg.`**`set_`**`(`*name*, *add*, *match*`)`

Add a value to the named set

USAGE:

```
foo:
  reg.set:
    - add: bar
    - match: my/custom/event
```

21.20.7 salt.thorium.runner module

React by calling asynchronous runners

`salt.thorium.runner.`**`cmd`**`(`*name*, *func=None*, *arg=()*, ***kwargs*`)`

Execute a runner asynchronous:

USAGE:

```
run_cloud:
  runner.cmd:
    - func: cloud.create
    - arg:
        - my-ec2-config
        - myinstance

run_cloud:
  runner.cmd:
    - func: cloud.create
    - kwargs:
        provider: my-ec2-config
        instances: myinstance
```

21.20.8 salt.thorium.status module

This thorium state is used to track the status beacon events and keep track of the active status of minions

New in version 2016.11.0.

`salt.thorium.status.`**`reg`**`(`*name*`)`

Activate this register to turn on a minion status tracking register, this register keeps the current status beacon data and the time that each beacon was last checked in.

21.20.9 salt.thorium.timer module

Allow for flow based timers. These timers allow for a sleep to exist across multiple runs of the flow

`salt.thorium.timer.`**`hold`**`(`*name, seconds*`)`

Wait for a given period of time, then fire a result of True, requiring this state allows for an action to be blocked for evaluation based on time

USAGE:

```
hold_on_a_moment:
  timer.hold:
    - seconds: 30
```

21.20.10 salt.thorium.wheel module

React by calling asynchronous runners

`salt.thorium.wheel.`**`cmd`**`(`*name, fun=None, arg=(), **kwargs*`)`

Execute a runner asynchronous:

USAGE:

```
run_cloud:
  wheel.cmd:
    - fun: key.delete
    - match: minion_id
```

21.21 auth modules

`localfs`	Stores eauth tokens in the filesystem of the master.
`rediscluster`	Provide token storage in Redis cluster.

21.21.1 salt.tokens.localfs

Stores eauth tokens in the filesystem of the master. Location is configured by the master config option 'token_dir'

`salt.tokens.localfs.`**`get_token`**`(`*opts, tok*`)`

Fetch the token data from the store.

Parameters
- **`opts`** – Salt master config options
- **`tok`** – Token value to get

Returns Token data if successful. Empty dict if failed.

`salt.tokens.localfs.`**`list_tokens`**`(`*opts*`)`

List all tokens in the store.

Parameters **`opts`** – Salt master config options

Returns List of dicts (tokens)

`salt.tokens.localfs.`**`mk_token`**`(`*opts, tdata*`)`

Mint a new token using the config option hash_type and store tdata with 'token' attribute set to the token. This module uses the hash of random 512 bytes as a token.

Parameters

- **`opts`** – Salt master config options
- **`tdata`** – Token data to be stored with 'token' attribute of this dict set to the token.

Returns tdata with token if successful. Empty dict if failed.

`salt.tokens.localfs.`**`rm_token`**`(`*opts*, *tok*`)`

Remove token from the store.

Parameters

- **`opts`** – Salt master config options
- **`tok`** – Token to remove

Returns Empty dict if successful. None if failed.

21.21.2 salt.tokens.rediscluster

Provide token storage in Redis cluster.

To get started simply start a redis cluster and assign all hashslots to the connected nodes. Add the redis hostname and port to master configs as eauth_redis_host and eauth_redis_port. Default values for these configs are as follow:

```
eauth_redis_host: localhost
eauth_redis_port: 6379
```

depends

- redis-py-cluster Python package

`salt.tokens.rediscluster.`**`get_token`**`(`*opts*, *tok*`)`

Fetch the token data from the store.

Parameters

- **`opts`** – Salt master config options
- **`tok`** – Token value to get

Returns Token data if successful. Empty dict if failed.

`salt.tokens.rediscluster.`**`list_tokens`**`(`*opts*`)`

List all tokens in the store.

Parameters **`opts`** – Salt master config options

Returns List of dicts (token_data)

`salt.tokens.rediscluster.`**`mk_token`**`(`*opts*, *tdata*`)`

Mint a new token using the config option hash_type and store tdata with 'token' attribute set to the token. This module uses the hash of random 512 bytes as a token.

Parameters

- **`opts`** – Salt master config options
- **`tdata`** – Token data to be stored with 'token' attribute of this dict set to the token.

Returns tdata with token if successful. Empty dict if failed.

`salt.tokens.rediscluster.`**`rm_token`**`(`*opts*, *tok*`)`

Remove token from the store.

Parameters

- **`opts`** – Salt master config options
- **`tok`** – Token to remove

Returns Empty dict if successful. None if failed.

21.22 master tops modules

cobbler	Cobbler Tops
ext_nodes	External Nodes Classifier
mongo	Read tops data from a mongodb collection
reclass_adapter	Read tops data from a reclass database
saltclass	Saltclass Configuration
varstack_top	Use Varstack to provide tops data

21.22.1 salt.tops.cobbler

Cobbler Tops

Cobbler Tops is a master tops subsystem used to look up mapping information from Cobbler via its API. The same cobbler.* parameters are used for both the Cobbler tops and Cobbler pillar modules.

```
master_tops:
  cobbler: {}
cobbler.url: https://example.com/cobbler_api #default is http://localhost/cobbler_api
cobbler.user: username # default is no username
cobbler.password: password # default is no password
```

Module Documentation

`salt.tops.cobbler.`**`top`**`(`***kwargs*`)`
Look up top data in Cobbler for a minion.

21.22.2 salt.tops.ext_nodes

External Nodes Classifier

The External Nodes Classifier is a master tops subsystem that retrieves mapping information from major configuration management systems. One of the most common external nodes classifiers system is provided by Cobbler and is called `cobbler-ext-nodes`.

The cobbler-ext-nodes command can be used with this configuration:

```
master_tops:
  ext_nodes: cobbler-ext-nodes
```

It is noteworthy that the Salt system does not directly ingest the data sent from the `cobbler-ext-nodes` command, but converts the data into information that is used by a Salt top file.

Any command can replace the call to 'cobbler-ext-nodes' above, but currently the data must be formatted in the same way that the standard 'cobbler-ext-nodes' does.

See (admittedly degenerate and probably not complete) example:

```
classes:
  - basepackages
  - database
```

The above essentially is the same as a top.sls containing the following:

```
base:
  '*':
    - basepackages
    - database

base:
  '*':
    - basepackages
    - database
```

`salt.tops.ext_nodes.`**`top`**`(`***kwargs*`)`

Run the command configured

21.22.3 salt.tops.mongo

Read tops data from a mongodb collection

This module will load tops data from a mongo collection. It uses the node's id for lookups.

Salt Master Mongo Configuration

The module shares the same base mongo connection variables as *`salt.returners.mongo_return`*. These variables go in your master config file.

- `mongo.db` - The mongo database to connect to. Defaults to `'salt'`.
- `mongo.host` - The mongo host to connect to. Supports replica sets by specifying all hosts in the set, comma-delimited. Defaults to `'salt'`.
- `mongo.port` - The port that the mongo database is running on. Defaults to `27017`.
- `mongo.user` - The username for connecting to mongo. Only required if you are using mongo authentication. Defaults to `''`.
- `mongo.password` - The password for connecting to mongo. Only required if you are using mongo authentication. Defaults to `''`.

Configuring the Mongo Tops Subsystem

```
master_tops:
  mongo:
    collection: tops
    id_field: _id
    re_replace: ""
    re_pattern: \.example\.com
    states_field: states
    environment_field: environment
```

Module Documentation

`salt.tops.mongo.`**`top`**`(`***kwargs*`)`

Connect to a mongo database and read per-node tops data.

Parameters

- **`collection`** (**) – The mongodb collection to read data from. Defaults to `'tops'`.

- **id_field** (*) – The field in the collection that represents an individual minion id. Defaults to `'_id'`.
- **re_pattern** (*) – If your naming convention in the collection is shorter than the minion id, you can use this to trim the name. *re_pattern* will be used to match the name, and *re_replace* will be used to replace it. Backrefs are supported as they are in the Python standard library. If `None`, no mangling of the name will be performed - the collection will be searched with the entire minion id. Defaults to `None`.
- **re_replace** (*) – Use as the replacement value in node ids matched with *re_pattern*. Defaults to ". Feel free to use backreferences here.
- **states_field** (*) – The name of the field providing a list of states.
- **environment_field** (*) – The name of the field providing the environment. Defaults to `environment`.

21.22.4 salt.tops.reclass_adapter

Read tops data from a reclass database

This *master_tops* plugin provides access to the **reclass** database, such that state information (top data) are retrieved from **reclass**.

You can find more information about **reclass** at http://reclass.pantsfullofunix.net.

To use the plugin, add it to the `master_tops` list in the Salt master config and tell **reclass** by way of a few options how and where to find the inventory:

```
master_tops:
  reclass:
    storage_type: yaml_fs
    inventory_base_uri: /srv/salt
```

This would cause **reclass** to read the inventory from YAML files in `/srv/salt/nodes` and `/srv/salt/classes`.

If you are also using **reclass** as `ext_pillar` plugin, and you want to avoid having to specify the same information for both, use YAML anchors (take note of the differing data types for `ext_pillar` and `master_tops`):

```
reclass: &reclass
  storage_type: yaml_fs
  inventory_base_uri: /srv/salt
  reclass_source_path: ~/code/reclass

ext_pillar:
  - reclass: *reclass

master_tops:
  reclass: *reclass
```

If you want to run reclass from source, rather than installing it, you can either let the master know via the `PYTHONPATH` environment variable, or by setting the configuration option, like in the example above.

`salt.tops.reclass_adapter.`**`top`**`(`*`**kwargs`*`)`

Query **reclass** for the top data (states of the minions).

21.22.5 salt.tops.saltclass

Saltclass Configuration

```
master_tops:
  saltclass:
    path: /srv/saltclass
```

Description

This module clones the behaviour of reclass (http://reclass.pantsfullofunix.net/), without the need of an external app, and add several features to improve flexibility. Saltclass lets you define your nodes from simple `yaml` files (`.yml`) through hierarchical class inheritance with the possibility to override pillars down the tree.

Features

- Define your nodes through hierarchical class inheritance
- **Reuse your reclass datas with minimal modifications**
  - applications => states
  - parameters => pillars
- Use Jinja templating in your yaml definitions
- **Access to the following Salt objects in Jinja**
  - `__opts__`
  - `__salt__`
  - `__grains__`
  - `__pillars__`
  - `minion_id`
- Chose how to merge or override your lists using ^ character (see examples)
- Expand variables ${} with possibility to escape them if needed \${} (see examples)
- Ignores missing node/class and will simply return empty without breaking the pillar module completely - will be logged

An example subset of datas is available here: http://git.mauras.ch/salt/saltclass/src/master/examples

Terms usable in yaml files	Description
classes	A list of classes that will be processed in order
states	A list of states that will be returned by master_tops function
pillars	A yaml dictionnary that will be returned by the ext_pillar function
environment	Node saltenv that will be used by master_tops

A class consists of:

- zero or more parent classes
- zero or more states
- any number of pillars

A child class can override pillars from a parent class. A node definition is a class in itself with an added `environment` parameter for `saltenv` definition.

Class names

Class names mimic salt way of defining states and pillar files. This means that `default.users` class name will correspond to one of these:

- `<saltclass_path>/classes/default/users.yml`
- `<saltclass_path>/classes/default/users/init.yml`

Saltclass file hierachy

A saltclass tree would look like this:

```
<saltclass_path>
├── classes
│   ├── app
│   │   ├── borgbackup.yml
│   │   └── ssh
│   │       └── server.yml
│   ├── default
│   │   ├── init.yml
│   │   ├── motd.yml
│   │   └── users.yml
│   ├── roles
│   │   ├── app.yml
│   │   └── nginx
│   │       ├── init.yml
│   │       └── server.yml
│   └── subsidiaries
│       ├── gnv.yml
│       ├── qls.yml
│       └── zrh.yml
└── nodes
    ├── geneva
    │   └── gnv.node1.yml
    ├── lausanne
    │   ├── qls.node1.yml
    │   └── qls.node2.yml
    ├── node127.yml
    └── zurich
        ├── zrh.node1.yml
        ├── zrh.node2.yml
        └── zrh.node3.yml
```

Saltclass Examples

`<saltclass_path>/nodes/lausanne/qls.node1.yml`

```
environment: base

classes:
{% for class in ['default'] %}
```

(continues on next page)

(continued from previous page)

```
  - {{ class }}
{% endfor %}
  - subsidiaries.{{ __grains__['id'].split('.')[0] }}
```

`<saltclass_path>/classes/default/init.yml`

```
classes:
  - default.users
  - default.motd

states:
  - openssh

pillars:
  default:
    network:
      dns:
        srv1: 192.168.0.1
        srv2: 192.168.0.2
        domain: example.com
    ntp:
      srv1: 192.168.10.10
      srv2: 192.168.10.20
```

`<saltclass_path>/classes/subsidiaries/gnv.yml`

```
pillars:
  default:
    network:
      sub: Geneva
      dns:
        srv1: 10.20.0.1
        srv2: 10.20.0.2
        srv3: 192.168.1.1
        domain: gnv.example.com
    users:
      adm1:
        uid: 1210
        gid: 1210
        gecos: 'Super user admin1'
        homedir: /srv/app/adm1
      adm3:
        uid: 1203
        gid: 1203
        gecos: 'Super user adm
```

Variable expansions

Escaped variables are rendered as is: `${test}`

Missing variables are rendered as is: `${net:dns:srv2}`

```
pillars:
  app:
  config:
```

(continues on next page)

(continued from previous page)

```
    dns:
      srv1: ${default:network:dns:srv1}
      srv2: ${net:dns:srv2}
    uri: https://application.domain/call?\${test}
    prod_parameters:
      - p1
      - p2
      - p3
  pkg:
    - app-core
    - app-backend
```

List override

Not using ^ as the first entry will simply merge the lists

```
pillars:
  app:
    pkg:
      - ^
      - app-frontend
```

Note: Known limitation

Currently you can't have both a variable and an escaped variable in the same string as the escaped one will not be correctly rendered - '${xx}' will stay as is instead of being rendered as '${xx}'

`salt.tops.saltclass.`**`top`**`(`***kwargs*`)`
Compile tops

21.22.6 salt.tops.varstack_top

Use Varstack to provide tops data

This *master_tops* plugin provides access to the **varstack** hierarchical yaml files, so you can user **varstack** as a full *`external node classifier`* and store state information (top data) in it.

Configuring Varstack

To use varstack as a master top external node classifier, install varstack as documented. Then, add to your master's configuration:

```
master_tops:
  varstack: /path/to/the/config/file/varstack.yaml
```

Varstack will then use /path/to/the/config/file/varstack.yaml (usually /etc/varstack.yaml) to determine which configuration data to return as adapter information. From there you can take a look at the README of varstack to learn how this file is evaluated. The ENC part will just return the 'states' dictionary for the node.

Ie, if my.fqdn.yaml file contains:

```
---
states:
  - sudo
  - openssh
  - apache
  - salt.minion
```

these will be returned as {'base': ['sudo', 'openssh', 'apache', 'salt.minion']} and managed by salt as if given from a top.sls file.

`salt.tops.varstack_top.`**`top`**`(`*`**kwargs`*`)`
Query **varstack** for the top data (states of the minions).

21.23 wheel modules

`config`	Manage the master configuration file
`error`	Error generator to enable integration testing of salt wheel error handling
`file_roots`	Read in files from the file_root and save files to the file root
`key`	Wheel system wrapper for the Salt key system to be used in interactions with the Salt Master programmatically.
`minions`	Wheel system wrapper for connected minions
`pillar_roots`	The *pillar_roots* wheel module is used to manage files under the pillar roots directories on the master server.

21.23.1 salt.wheel.config

Manage the master configuration file

`salt.wheel.config.`**`apply`**`(`*`key, value`*`)`
Set a single key

Note: This will strip comments from your config file

`salt.wheel.config.`**`update_config`**`(`*`file_name, yaml_contents`*`)`
Update master config with `yaml_contents`.

Writes `yaml_contents` to a file named `file_name.conf` under the folder specified by `default_include`. This folder is named `master.d` by default. Please look at `include-configuration` for more information.

Example low data:

```
data = {
    'username': 'salt',
    'password': 'salt',
    'fun': 'config.update_config',
    'file_name': 'gui',
    'yaml_contents': {'id': 1},
    'client': 'wheel',
```

(continues on next page)

(continued from previous page)

```
    'eauth': 'pam',
}
```

`salt.wheel.config.`**`values`**`()`

Return the raw values of the config file

21.23.2 salt.wheel.error

Error generator to enable integration testing of salt wheel error handling

`salt.wheel.error.`**`error`**`(`*name=None, message=''*`)`

If name is None Then return empty dict

Otherwise raise an exception with __name__ from name, message from message

CLI Example:

```
salt-wheel error
salt-wheel error.error name="Exception" message="This is an error."
```

21.23.3 salt.wheel.file_roots

Read in files from the file_root and save files to the file root

`salt.wheel.file_roots.`**`find`**`(`*path, saltenv='base'*`)`

Return a dict of the files located with the given path and environment

`salt.wheel.file_roots.`**`list_env`**`(`*saltenv='base'*`)`

Return all of the file paths found in an environment

`salt.wheel.file_roots.`**`list_roots`**`()`

Return all of the files names in all available environments

`salt.wheel.file_roots.`**`read`**`(`*path, saltenv='base'*`)`

Read the contents of a text file, if the file is binary then

`salt.wheel.file_roots.`**`write`**`(`*data, path, saltenv='base', index=0*`)`

Write the named file, by default the first file found is written, but the index of the file can be specified to write to a lower priority file root

21.23.4 salt.wheel.key

Wheel system wrapper for the Salt key system to be used in interactions with the Salt Master programmatically.

The key module for the wheel system is meant to provide an internal interface for other Salt systems to interact with the Salt Master. The following usage examples assume that a WheelClient is available:

```
import salt.config
import salt.wheel
opts = salt.config.master_config('/etc/salt/master')
wheel = salt.wheel.WheelClient(opts)
```

Note that importing and using the `WheelClient` must be performed on the same machine as the Salt Master and as the same user that runs the Salt Master, unless *`external_auth`* is configured and the user is authorized to execute wheel functions.

The function documentation starts with the `wheel` reference from the code sample above and use the `WheelClient` functions to show how they can be called from a Python interpreter.

The wheel key functions can also be called via a `salt` command at the CLI using the *`saltutil execution module`*.

`salt.wheel.key.`**`accept`**(*match*, *include_rejected=False*, *include_denied=False*)

Accept keys based on a glob match. Returns a dictionary.

match The glob match of keys to accept.

include_rejected To include rejected keys in the match along with pending keys, set this to `True`. Defaults to `False`.

include_denied To include denied keys in the match along with pending keys, set this to `True`. Defaults to `False`.

```
>>> wheel.cmd('key.accept', ['minion1'])
{'minions': ['minion1']}
```

`salt.wheel.key.`**`accept_dict`**(*match*, *include_rejected=False*, *include_denied=False*)

Accept keys based on a dict of keys. Returns a dictionary.

match The dictionary of keys to accept.

include_rejected To include rejected keys in the match along with pending keys, set this to `True`. Defaults to `False`.

New in version 2016.3.4.

include_denied To include denied keys in the match along with pending keys, set this to `True`. Defaults to `False`.

New in version 2016.3.4.

Example to move a list of keys from the `minions_pre` (pending) directory to the `minions` (accepted) directory:

```
>>> wheel.cmd('key.accept_dict',
{
    'minions_pre': [
        'jerry',
        'stuart',
        'bob',
    ],
})
{'minions': ['jerry', 'stuart', 'bob']}
```

`salt.wheel.key.`**`delete`**(*match*)

Delete keys based on a glob match. Returns a dictionary.

match The glob match of keys to delete.

```
>>> wheel.cmd_async({'fun': 'key.delete', 'match': 'minion1'})
{'jid': '20160826201244808521', 'tag': 'salt/wheel/20160826201244808521'}
```

`salt.wheel.key.`**`delete_dict`**(*match*)

Delete keys based on a dict of keys. Returns a dictionary.

match The dictionary of keys to delete.

```
>>> wheel.cmd_async({'fun': 'key.delete_dict',
'match': {
    'minions': [
        'jerry',
        'stuart',
```

(continues on next page)

(continued from previous page)

```
        'bob',
    ],
})
{'jid': '20160826201244808521', 'tag': 'salt/wheel/20160826201244808521'}
```

`salt.wheel.key.`**`finger`**(*match*, *hash_type=None*)

Return the matching key fingerprints. Returns a dictionary.

match The key for with to retrieve the fingerprint.

hash_type The hash algorithm used to calculate the fingerprint

```
>>> wheel.cmd('key.finger', ['minion1'])
{'minions': {'minion1': '5d:f6:79:43:5e:d4:42:3f:57:b8:45:a8:7e:a4:6e:ca'}}
```

`salt.wheel.key.`**`finger_master`**(*hash_type=None*)

Return the fingerprint of the master's public key

hash_type The hash algorithm used to calculate the fingerprint

```
>>> wheel.cmd('key.finger_master')
{'local': {'master.pub': '5d:f6:79:43:5e:d4:42:3f:57:b8:45:a8:7e:a4:6e:ca'}}
```

`salt.wheel.key.`**`gen`**(*id_=None*, *keysize=2048*)

Generate a key pair. No keys are stored on the master. A key pair is returned as a dict containing pub and priv keys. Returns a dictionary containing the the `pub` and `priv` keys with their generated values.

id_ Set a name to generate a key pair for use with salt. If not specified, a random name will be specified.

keysize The size of the key pair to generate. The size must be `2048`, which is the default, or greater. If set to a value less than `2048`, the key size will be rounded up to `2048`.

```
>>> wheel.cmd('key.gen')
{'pub': '-----BEGIN PUBLIC KEY-----\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBC
...
BBPfamX9gGPQTpN9e8HwcZjXQnmg8OrcUl10WHw09SDWLOlnW+ueTWugEQpPt\niQIDAQAB\n
-----END PUBLIC KEY-----',
'priv': '-----BEGIN RSA PRIVATE KEY-----\nMIIEpAIBAAKCAQEA42Kf+w9XeZWgguzv
...
QH3/W74X1+WTBlx4R2KGLYBiH+bCCFEQ/Zvcu4Xp4bIOPtRKozEQ==\n
-----END RSA PRIVATE KEY-----'}
```

`salt.wheel.key.`**`gen_accept`**(*id_*, *keysize=2048*, *force=False*)

Generate a key pair then accept the public key. This function returns the key pair in a dict, only the public key is preserved on the master. Returns a dictionary.

id_ The name of the minion for which to generate a key pair.

keysize The size of the key pair to generate. The size must be `2048`, which is the default, or greater. If set to a value less than `2048`, the key size will be rounded up to `2048`.

force If a public key has already been accepted for the given minion on the master, then the gen_accept function will return an empty dictionary and not create a new key. This is the default behavior. If `force` is set to `True`, then the minion's previously accepted key will be overwritten.

```
>>> wheel.cmd('key.gen_accept', ['foo'])
{'pub': '-----BEGIN PUBLIC KEY-----\nMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBC
...
BBPfamX9gGPQTpN9e8HwcZjXQnmg8OrcUl10WHw09SDWLOlnW+ueTWugEQpPt\niQIDAQAB\n
-----END PUBLIC KEY-----',
'priv': '-----BEGIN RSA PRIVATE KEY-----\nMIIEpAIBAAKCAQEA42Kf+w9XeZWgguzv
...
```

(continues on next page)

(continued from previous page)

```
QH3/W74X1+WTBlx4R2KGLYBiH+bCCFEQ/Zvcu4Xp4bIOPtRKozEQ==\n
-----END RSA PRIVATE KEY-----'}
```

We can now see that the `foo` minion's key has been accepted by the master:

```
>>> wheel.cmd('key.list', ['accepted'])
{'minions': ['foo', 'minion1', 'minion2', 'minion3']}
```

`salt.wheel.key.`**`gen_keys`**(*keydir=None*, *keyname=None*, *keysize=None*, *user=None*)

Generate minion RSA public keypair

`salt.wheel.key.`**`gen_signature`**(*priv*, *pub*, *signature_path*, *auto_create=False*, *keysize=None*)

Generate master public-key-signature

`salt.wheel.key.`**`key_str`**(*match*)

Return information about the key. Returns a dictionary.

match The key to return information about.

```
>>> wheel.cmd('key.key_str', ['minion1'])
{'minions': {'minion1': '-----BEGIN PUBLIC KEY-----\nMIIBIjANBgkqhkiG9w0B
...
TWugEQpPt\niQIDAQAB\n-----END PUBLIC KEY-----'}}
```

`salt.wheel.key.`**`list_`**(*match*)

List all the keys under a named status. Returns a dictionary.

match The type of keys to list. The `pre`, `un`, and `unaccepted` options will list unaccepted/unsigned keys. `acc` or `accepted` will list accepted/signed keys. `rej` or `rejected` will list rejected keys. Finally, `all` will list all keys.

```
>>> wheel.cmd('key.list', ['accepted'])
{'minions': ['minion1', 'minion2', 'minion3']}
```

`salt.wheel.key.`**`list_all`**()

List all the keys. Returns a dictionary containing lists of the minions in each salt-key category, including `minions`, `minions_rejected`, `minions_denied`, etc. Returns a dictionary.

```
>>> wheel.cmd('key.list_all')
{'local': ['master.pem', 'master.pub'], 'minions_rejected': [],
'minions_denied': [], 'minions_pre': [],
'minions': ['minion1', 'minion2', 'minion3']}
```

`salt.wheel.key.`**`name_match`**(*match*)

List all the keys based on a glob match

`salt.wheel.key.`**`reject`**(*match*, *include_accepted=False*, *include_denied=False*)

Reject keys based on a glob match. Returns a dictionary.

match The glob match of keys to reject.

include_accepted To include accepted keys in the match along with pending keys, set this to `True`. Defaults to `False`.

include_denied To include denied keys in the match along with pending keys, set this to `True`. Defaults to `False`.

```
>>> wheel.cmd_async({'fun': 'key.reject', 'match': 'minion1'})
{'jid': '20160826201244808521', 'tag': 'salt/wheel/20160826201244808521'}
```

`salt.wheel.key.`**`reject_dict`**(*match*, *include_accepted=False*, *include_denied=False*)

Reject keys based on a dict of keys. Returns a dictionary.

match The dictionary of keys to reject.

include_accepted To include accepted keys in the match along with pending keys, set this to `True`. Defaults to `False`.

New in version 2016.3.4.

include_denied To include denied keys in the match along with pending keys, set this to `True`. Defaults to `False`.

New in version 2016.3.4.

```
>>> wheel.cmd_async({'fun': 'key.reject_dict',
'match': {
    'minions': [
        'jerry',
        'stuart',
        'bob',
    ],
})
{'jid': '20160826201244808521', 'tag': 'salt/wheel/20160826201244808521'}
```

21.23.5 salt.wheel.minions

Wheel system wrapper for connected minions

`salt.wheel.minions.`**`connected`**`()`
List all connected minions on a salt-master

21.23.6 salt.wheel.pillar_roots

The *pillar_roots* wheel module is used to manage files under the pillar roots directories on the master server.

`salt.wheel.pillar_roots.`**`find`**`(`*path, saltenv='base'*`)`
Return a dict of the files located with the given path and environment

`salt.wheel.pillar_roots.`**`list_env`**`(`*saltenv='base'*`)`
Return all of the file paths found in an environment

`salt.wheel.pillar_roots.`**`list_roots`**`()`
Return all of the files names in all available environments

`salt.wheel.pillar_roots.`**`read`**`(`*path, saltenv='base'*`)`
Read the contents of a text file, if the file is binary then

`salt.wheel.pillar_roots.`**`write`**`(`*data, path, saltenv='base', index=0*`)`
Write the named file, by default the first file found is written, but the index of the file can be specified to write to a lower priority file root

CHAPTER

TWENTYTWO

APIS

22.1 Python client API

Salt provides several entry points for interfacing with Python applications. These entry points are often referred to as `*Client()` APIs. Each client accesses different parts of Salt, either from the master or from a minion. Each client is detailed below.

See also:

There are many ways to access Salt programmatically.

Salt can be used from CLI scripts as well as via a REST interface.

See Salt's *outputter system* to retrieve structured data from Salt as JSON, or as shell-friendly text, or many other formats.

See the `state.event` runner to utilize Salt's event bus from shell scripts.

Salt's netapi module provides access to Salt externally via a REST interface. Review the netapi module documentation for more information.

22.1.1 Salt's `opts` dictionary

Some clients require access to Salt's `opts` dictionary. (The dictionary representation of the *master* or *minion* config files.)

A common pattern for fetching the `opts` dictionary is to defer to environment variables if they exist or otherwise fetch the config from the default location.

`salt.config.`**`client_config`**(*path*, *env_var='SALT_CLIENT_CONFIG'*, *defaults=None*)

Load Master configuration data

Usage:

```
import salt.config
master_opts = salt.config.client_config('/etc/salt/master')
```

Returns a dictionary of the Salt Master configuration file with necessary options needed to communicate with a locally-running Salt Master daemon. This function searches for client specific configurations and adds them to the data from the master configuration.

This is useful for master-side operations like `LocalClient`.

`salt.config.`**`minion_config`**(*path, env_var='SALT_MINION_CONFIG', defaults=None, cache_minion_id=False, ignore_config_errors=True, minion_id=None, role='minion'*)

Reads in the minion configuration file and sets up special options

This is useful for Minion-side operations, such as the *Caller* class, and manually running the loader interface.

```
import salt.config
minion_opts = salt.config.minion_config('/etc/salt/minion')
```

22.1.2 Salt's Loader Interface

Modules in the Salt ecosystem are loaded into memory using a custom loader system. This allows modules to have conditional requirements (OS, OS version, installed libraries, etc) and allows Salt to inject special variables (`__salt__`, `__opts__`, etc).

Most modules can be manually loaded. This is often useful in third-party Python apps or when writing tests. However some modules require and expect a full, running Salt system underneath. Notably modules that facilitate master-to-minion communication such as the *mine*, *publish*, and `peer` execution modules. The error `KeyError: 'master_uri'` is a likely indicator for this situation. In those instances use the *Caller* class to execute those modules instead.

Each module type has a corresponding loader function.

`salt.loader.`**`minion_mods`**(*opts, context=None, utils=None, whitelist=None, initial_load=False, loaded_base_name=None, notify=False, static_modules=None, proxy=None*)

Load execution modules

Returns a dictionary of execution modules appropriate for the current system by evaluating the __virtual__() function in each module.

Parameters

- **`opts`** (*dict*) – The Salt options dictionary
- **`context`** (*dict*) – A Salt context that should be made present inside generated modules in __context__
- **`utils`** (*dict*) – Utility functions which should be made available to Salt modules in __utils__. See *utils_dirs* in salt.config for additional information about configuration.
- **`whitelist`** (*list*) – A list of modules which should be whitelisted.
- **`initial_load`** (*bool*) – Deprecated flag! Unused.
- **`loaded_base_name`** (*str*) – A string marker for the loaded base name.
- **`notify`** (*bool*) – Flag indicating that an event should be fired upon completion of module loading.

```
import salt.config
import salt.loader

__opts__ = salt.config.minion_config('/etc/salt/minion')
__grains__ = salt.loader.grains(__opts__)
__opts__['grains'] = __grains__
__utils__ = salt.loader.utils(__opts__)
__salt__ = salt.loader.minion_mods(__opts__, utils=__utils__)
__salt__['test.ping']()
```

`salt.loader.`**`raw_mod`**(*opts, name, functions, mod='modules'*)

Returns a single module loaded raw and bypassing the __virtual__ function

```
import salt.config
import salt.loader

__opts__ = salt.config.minion_config('/etc/salt/minion')
testmod = salt.loader.raw_mod(__opts__, 'test', None)
testmod['test.ping']()
```

`salt.loader.`**`states`**(*opts, functions, utils, serializers, whitelist=None, proxy=None, context=None*)

Returns the state modules

Parameters

- **opts** (*dict*) – The Salt options dictionary
- **functions** (*dict*) – A dictionary of minion modules, with module names as keys and funcs as values.

```
import salt.config
import salt.loader

__opts__ = salt.config.minion_config('/etc/salt/minion')
statemods = salt.loader.states(__opts__, None, None)
```

`salt.loader.`**`grains`**(*opts, force_refresh=False, proxy=None*)

Return the functions for the dynamic grains and the values for the static grains.

Since grains are computed early in the startup process, grains functions do not have __salt__ or __proxy__ available. At proxy-minion startup, this function is called with the proxymodule LazyLoader object so grains functions can communicate with their controlled device.

```
import salt.config
import salt.loader

__opts__ = salt.config.minion_config('/etc/salt/minion')
__grains__ = salt.loader.grains(__opts__)
print __grains__['id']
```

`salt.loader.`**`grain_funcs`**(*opts, proxy=None*)

Returns the grain functions

```
import salt.config
import salt.loader

__opts__ = salt.config.minion_config('/etc/salt/minion')
grainfuncs = salt.loader.grain_funcs(__opts__)
```

22.1.3 Salt's Client Interfaces

LocalClient

class `salt.client.`**`LocalClient`**(*c_path='/etc/salt/master', mopts=None, skip_perm_errors=False, io_loop=None, keep_loop=False, auto_reconnect=False*)

The interface used by the **salt** CLI tool on the Salt Master

`LocalClient` is used to send a command to Salt minions to execute *execution modules* and return the results to the Salt Master.

Importing and using `LocalClient` must be done on the same machine as the Salt Master and it must be done using the same user that the Salt Master is running as. (Unless *`external_auth`* is configured and authentication credentials are included in the execution).

Note: The LocalClient uses a Tornado IOLoop, this can create issues when using the LocalClient inside an existing IOLoop. If creating the LocalClient in partnership with another IOLoop either create the IOLoop before creating the LocalClient, or when creating the IOLoop use ioloop.current() which will return the ioloop created by LocalClient.

```
import salt.client

local = salt.client.LocalClient()
local.cmd('*', 'test.fib', [10])
```

cmd(*tgt, fun, arg=(), timeout=None, tgt_type='glob', ret='', jid='', full_return=False, kwarg=None, **kwargs*)

Synchronously execute a command on targeted minions

The cmd method will execute and wait for the timeout period for all minions to reply, then it will return all minion data at once.

```
>>> import salt.client
>>> local = salt.client.LocalClient()
>>> local.cmd('*', 'cmd.run', ['whoami'])
{'jerry': 'root'}
```

With extra keyword arguments for the command function to be run:

```
local.cmd('*', 'test.arg', ['arg1', 'arg2'], kwarg={'foo': 'bar'})
```

Compound commands can be used for multiple executions in a single publish. Function names and function arguments are provided in separate lists but the index values must correlate and an empty list must be used if no arguments are required.

```
>>> local.cmd('*', [
        'grains.items',
        'sys.doc',
        'cmd.run',
    ],
    [
        [],
        [],
        ['uptime'],
    ])
```

Parameters

- **tgt** (*string or list*) – Which minions to target for the execution. Default is shell glob. Modified by the `tgt_type` option.
- **fun** (*string or list of strings*) – The module and function to call on the specified minions of the form `module.function`. For example `test.ping` or `grains.items`.

  Compound commands Multiple functions may be called in a single publish by passing a list of commands. This can dramatically lower overhead and speed up the application communicating with Salt.

  This requires that the `arg` param is a list of lists. The `fun` list and the `arg` list must correlate by index meaning a function that does not take arguments must still have a corresponding empty list at the expected index.

- **arg** (*list or list-of-lists*) – A list of arguments to pass to the remote function. If the function takes no arguments `arg` may be omitted except when executing a compound command.
- **timeout** – Seconds to wait after the last minion returns but before all minions return.
- **tgt_type** – The type of `tgt`. Allowed values:
  - `glob` - Bash glob completion - Default
  - `pcre` - Perl style regular expression
  - `list` - Python list of hosts
  - `grain` - Match based on a grain comparison
  - `grain_pcre` - Grain comparison with a regex
  - `pillar` - Pillar data comparison
  - `pillar_pcre` - Pillar data comparison with a regex
  - `nodegroup` - Match on nodegroup
  - `range` - Use a Range server for matching
  - `compound` - Pass a compound match string
  - `ipcidr` - Match based on Subnet (CIDR notation) or IPv4 address.

  Changed in version 2017.7.0: Renamed from `expr_form` to `tgt_type`
- **ret** – The returner to use. The value passed can be single returner, or a comma delimited list of returners to call in order on the minions
- **kwarg** – A dictionary with keyword arguments for the function.
- **full_return** – Output the job return only (default) or the full return including exit code and other job metadata.
- **kwargs** – Optional keyword arguments. Authentication credentials may be passed when using *external_auth*.

  For example: `local.cmd('*', 'test.ping', username='saltdev', password='saltdev', eauth='pam')`. Or: `local.cmd('*', 'test.ping', token='5871821ea51754fdcea8153c1c745433')`

Returns A dictionary with the result of the execution, keyed by minion ID. A compound command will return a sub-dictionary keyed by function name.

cmd_async(*tgt, fun, arg=(), tgt_type='glob', ret='', jid='', kwarg=None, **kwargs*)

Asynchronously send a command to connected minions

The function signature is the same as *cmd()* with the following exceptions.

Returns A job ID or 0 on failure.

```
>>> local.cmd_async('*', 'test.sleep', [300])
'20131219215921857715'
```

cmd_batch(*tgt, fun, arg=(), tgt_type='glob', ret='', kwarg=None, batch='10%', **kwargs*)

Iteratively execute a command on subsets of minions at a time

The function signature is the same as *cmd()* with the following exceptions.

Parameters **batch** – The batch identifier of systems to execute on

Returns A generator of minion returns

```
>>> returns = local.cmd_batch('*', 'state.highstate', batch='10%')
>>> for ret in returns:
...     print(ret)
{'jerry': {...}}
{'dave': {...}}
{'stewart': {...}}
```

cmd_iter(*tgt, fun, arg=(), timeout=None, tgt_type='glob', ret='', kwarg=None, **kwargs*)
Yields the individual minion returns as they come in

The function signature is the same as `cmd()` with the following exceptions.

Normally `cmd_iter()` does not yield results for minions that are not connected. If you want it to return results for disconnected minions set *expect_minions=True* in *kwargs*.

Returns A generator yielding the individual minion returns

```
>>> ret = local.cmd_iter('*', 'test.ping')
>>> for i in ret:
...     print(i)
{'jerry': {'ret': True}}
{'dave': {'ret': True}}
{'stewart': {'ret': True}}
```

cmd_iter_no_block(*tgt, fun, arg=(), timeout=None, tgt_type='glob', ret='', kwarg=None, show_jid=False, verbose=False, **kwargs*)
Yields the individual minion returns as they come in, or None when no returns are available.
The function signature is the same as `cmd()` with the following exceptions.

Returns A generator yielding the individual minion returns, or None when no returns are available. This allows for actions to be injected in between minion returns.

```
>>> ret = local.cmd_iter_no_block('*', 'test.ping')
>>> for i in ret:
...     print(i)
None
{'jerry': {'ret': True}}
{'dave': {'ret': True}}
None
{'stewart': {'ret': True}}
```

cmd_subset(*tgt, fun, arg=(), tgt_type='glob', ret='', kwarg=None, sub=3, cli=False, progress=False, full_return=False, **kwargs*)
Execute a command on a random subset of the targeted systems

The function signature is the same as `cmd()` with the following exceptions.

Parameters

- **sub** – The number of systems to execute on
- **cli** – When this is set to True, a generator is returned, otherwise a dictionary of the minion returns is returned

```
>>> SLC.cmd_subset('*', 'test.ping', sub=1)
{'jerry': True}
```

get_cli_returns(*jid, minions, timeout=None, tgt='*', tgt_type='glob', verbose=False, show_jid=False, **kwargs*)
Starts a watcher looking at the return data for a specified JID

Returns all of the information for the JID

get_event_iter_returns(*jid, minions, timeout=None*)
Gather the return data from the event system, break hard when timeout is reached.

run_job(*tgt, fun, arg=(), tgt_type='glob', ret='', timeout=None, jid='', kwarg=None, listen=False, **kwargs*)
Asynchronously send a command to connected minions

Prep the job directory and publish a command to any targeted minions.

Returns A dictionary of (validated) `pub_data` or an empty dictionary on failure. The `pub_data` contains the job ID and a list of all minions that are expected to return data.

```
>>> local.run_job('*', 'test.sleep', [300])
{'jid': '20131219215650131543', 'minions': ['jerry']}
```

Salt Caller

class `salt.client.`**`Caller`**(*c_path='/etc/salt/minion', mopts=None*)

`Caller` is the same interface used by the **`salt-call`** command-line tool on the Salt Minion.

Changed in version 2015.8.0: Added the `cmd` method for consistency with the other Salt clients. The existing `function` and `sminion.functions` interfaces still exist but have been removed from the docs.

Importing and using `Caller` must be done on the same machine as a Salt Minion and it must be done using the same user that the Salt Minion is running as.

Usage:

```
import salt.client
caller = salt.client.Caller()
caller.cmd('test.ping')
```

Note, a running master or minion daemon is not required to use this class. Running `salt-call --local` simply sets *`file_client`* to `'local'`. The same can be achieved at the Python level by including that setting in a minion config file.

New in version 2014.7.0: Pass the minion config as the `mopts` dictionary.

```
import salt.client
import salt.config
__opts__ = salt.config.minion_config('/etc/salt/minion')
__opts__['file_client'] = 'local'
caller = salt.client.Caller(mopts=__opts__)
```

cmd(*fun, *args, **kwargs*)

Call an execution module with the given arguments and keyword arguments

Changed in version 2015.8.0: Added the `cmd` method for consistency with the other Salt clients. The existing `function` and `sminion.functions` interfaces still exist but have been removed from the docs.

```
caller.cmd('test.arg', 'Foo', 'Bar', baz='Baz')

caller.cmd('event.send', 'myco/myevent/something',
    data={'foo': 'Foo'}, with_env=['GIT_COMMIT'], with_grains=True)
```

Salt Proxy Caller

class `salt.client.`**`ProxyCaller`**(*c_path='/etc/salt/proxy', mopts=None*)

`ProxyCaller` is the same interface used by the **`salt-call`** with the args `--proxyid <proxyid>` command-line tool on the Salt Proxy Minion.

Importing and using `ProxyCaller` must be done on the same machine as a Salt Minion and it must be done using the same user that the Salt Minion is running as.

Usage:

```
import salt.client
caller = salt.client.Caller()
caller.cmd('test.ping')
```

Note, a running master or minion daemon is not required to use this class. Running `salt-call --local` simply sets *`file_client`* to `'local'`. The same can be achieved at the Python level by including that setting in a minion config file.

```
import salt.client
import salt.config
__opts__ = salt.config.proxy_config('/etc/salt/proxy', minion_id='quirky_edison')
__opts__['file_client'] = 'local'
caller = salt.client.ProxyCaller(mopts=__opts__)
```

Note: To use this for calling proxies, the `is_proxy functions` requires that `--proxyid` be an argument on the commandline for the script this is used in, or that the string `proxy` is in the name of the script.

`cmd`(*fun*, **args*, ***kwargs*)
Call an execution module with the given arguments and keyword arguments

```
caller.cmd('test.arg', 'Foo', 'Bar', baz='Baz')

caller.cmd('event.send', 'myco/myevent/something',
    data={'foo': 'Foo'}, with_env=['GIT_COMMIT'], with_grains=True)
```

RunnerClient

class `salt.runner.`**`RunnerClient`**(*opts*, *context=None*)
The interface used by the **`salt-run`** CLI tool on the Salt Master

It executes *runner modules* which run on the Salt Master.

Importing and using `RunnerClient` must be done on the same machine as the Salt Master and it must be done using the same user that the Salt Master is running as.

Salt's *`external_auth`* can be used to authenticate calls. The eauth user must be authorized to execute runner modules: (`@runner`). Only the `master_call()` below supports eauth.

`asynchronous`(*fun*, *low*, *user='UNKNOWN'*, *pub=None*)
Execute the function in a multiprocess and return the event tag to use to watch for the return

`cmd`(*fun*, *arg=None*, *pub_data=None*, *kwarg=None*, *print_event=True*, *full_return=False*)
Execute a function

`cmd_async`(*low*)
Execute a runner function asynchronously; eauth is respected

This function requires that *`external_auth`* is configured and the user is authorized to execute runner functions: (`@runner`).

```
runner.eauth_async({
    'fun': 'jobs.list_jobs',
    'username': 'saltdev',
    'password': 'saltdev',
```

(continues on next page)

(continued from previous page)

```
    'eauth': 'pam',
})
```

cmd_sync(*low*, *timeout=None*, *full_return=False*)

Execute a runner function synchronously; eauth is respected

This function requires that *external_auth* is configured and the user is authorized to execute runner functions: (`@runner`).

```
runner.eauth_sync({
    'fun': 'jobs.list_jobs',
    'username': 'saltdev',
    'password': 'saltdev',
    'eauth': 'pam',
})
```

WheelClient

class `salt.wheel.`**WheelClient**(*opts=None*)

An interface to Salt's wheel modules

Wheel modules interact with various parts of the Salt Master.

Importing and using `WheelClient` must be done on the same machine as the Salt Master and it must be done using the same user that the Salt Master is running as. Unless *external_auth* is configured and the user is authorized to execute wheel functions: (`@wheel`).

Usage:

```
import salt.config
import salt.wheel
opts = salt.config.master_config('/etc/salt/master')
wheel = salt.wheel.WheelClient(opts)
```

asynchronous(*fun*, *low*, *user='UNKNOWN'*, *pub=None*)

Execute the function in a multiprocess and return the event tag to use to watch for the return

cmd(*fun*, *arg=None*, *pub_data=None*, *kwarg=None*, *print_event=True*, *full_return=False*)

Execute a function

```
>>> wheel.cmd('key.finger', ['jerry'])
{'minions': {'jerry': '5d:f6:79:43:5e:d4:42:3f:57:b8:45:a8:7e:a4:6e:ca'}}
```

cmd_async(*low*)

Execute a function asynchronously; eauth is respected

This function requires that *external_auth* is configured and the user is authorized

```
>>> wheel.cmd_async({
    'fun': 'key.finger',
    'match': 'jerry',
    'eauth': 'auto',
    'username': 'saltdev',
    'password': 'saltdev',
})
{'jid': '20131219224744416681', 'tag': 'salt/wheel/20131219224744416681'}
```

cmd_sync(*low, timeout=None, full_return=False*)
Execute a wheel function synchronously; eauth is respected

This function requires that *external_auth* is configured and the user is authorized to execute runner functions: (`@wheel`).

```
>>> wheel.cmd_sync({
'fun': 'key.finger',
'match': 'jerry',
'eauth': 'auto',
'username': 'saltdev',
'password': 'saltdev',
})
{'minions': {'jerry': '5d:f6:79:43:5e:d4:42:3f:57:b8:45:a8:7e:a4:6e:ca'}}
```

CloudClient

class salt.cloud.**CloudClient**(*path=None, opts=None, config_dir=None, pillars=None*)
The client class to wrap cloud interactions

action(*fun=None, cloudmap=None, names=None, provider=None, instance=None, kwargs=None*)
Execute a single action via the cloud plugin backend

Examples:

```
client.action(fun='show_instance', names=['myinstance'])
client.action(fun='show_image', provider='my-ec2-config',
    kwargs={'image': 'ami-10314d79'}
)
```

create(*provider, names, **kwargs*)
Create the named VMs, without using a profile

Example:

```
client.create(provider='my-ec2-config', names=['myinstance'],
    image='ami-1624987f', size='t1.micro', ssh_username='ec2-user',
    securitygroup='default', delvol_on_destroy=True)
```

destroy(*names*)
Destroy the named VMs

extra_action(*names, provider, action, **kwargs*)
Perform actions with block storage devices

Example:

```
client.extra_action(names=['myblock'], action='volume_create',
    provider='my-nova', kwargs={'voltype': 'SSD', 'size': 1000}
)
client.extra_action(names=['salt-net'], action='network_create',
    provider='my-nova', kwargs={'cidr': '192.168.100.0/24'}
)
```

full_query(*query_type='list_nodes_full'*)
Query all instance information

list_images(*provider=None*)
List all available images in configured cloud systems

list_locations(*provider=None*)
List all available locations in configured cloud systems

list_sizes(*provider=None*)
List all available sizes in configured cloud systems

low(*fun, low*)
Pass the cloud function and low data structure to run

map_run(*path=None, **kwargs*)
To execute a map

min_query(*query_type='list_nodes_min'*)
Query select instance information

profile(*profile, names, vm_overrides=None, **kwargs*)
Pass in a profile to create, names is a list of vm names to allocate
vm_overrides is a special dict that will be per node options overrides

Example:

```
>>> client= salt.cloud.CloudClient(path='/etc/salt/cloud')
>>> client.profile('do_512_git', names=['minion01',])
{'minion01': {'backups_active': 'False',
        'created_at': '2014-09-04T18:10:15Z',
        'droplet': {'event_id': 31000502,
                     'id': 2530006,
                     'image_id': 5140006,
                     'name': 'minion01',
                     'size_id': 66},
        'id': '2530006',
        'image_id': '5140006',
        'ip_address': '107.XXX.XXX.XXX',
        'locked': 'True',
        'name': 'minion01',
        'private_ip_address': None,
        'region_id': '4',
        'size_id': '66',
        'status': 'new'}}
```

query(*query_type='list_nodes'*)
Query basic instance information

select_query(*query_type='list_nodes_select'*)
Query select instance information

SSHClient

class salt.client.ssh.client.**SSHClient**(*c_path='/etc/salt/master', mopts=None, disable_custom_roster=False*)
Create a client object for executing routines via the salt-ssh backend

New in version 2015.5.0.

cmd(*tgt, fun, arg=(), timeout=None, tgt_type='glob', kwarg=None, **kwargs*)
Execute a single command via the salt-ssh subsystem and return all routines at once

New in version 2015.5.0.

cmd_iter(*tgt, fun, arg=(), timeout=None, tgt_type='glob', ret='', kwarg=None, **kwargs*)
Execute a single command via the salt-ssh subsystem and return a generator

New in version 2015.5.0.

22.2 `netapi` modules

22.2.1 Introduction to netapi modules

netapi modules provide API-centric access to Salt. Usually externally-facing services such as REST or WebSockets, XMPP, XMLRPC, etc.

In general netapi modules bind to a port and start a service. They are purposefully open-ended. A single module can be configured to run as well as multiple modules simultaneously.

netapi modules are enabled by adding configuration to your Salt Master config file and then starting the **`salt-api`** daemon. Check the docs for each module to see external requirements and configuration settings.

Communication with Salt and Salt satellite projects is done using Salt's own *Python API*. A list of available client interfaces is below.

---

salt-api

Prior to Salt's 2014.7.0 release, netapi modules lived in the separate sister projected `salt-api`. That project has been merged into the main Salt project.

See also:

The full list of netapi modules

22.2.2 Client interfaces

Salt's client interfaces expose executing functions by crafting a dictionary of values that are mapped to function arguments. This allows calling functions simply by creating a data structure. (And this is exactly how much of Salt's own internals work!)

class salt.netapi.**NetapiClient**(*opts*)

Provide a uniform method of accessing the various client interfaces in Salt in the form of low-data data structures. For example:

```
>>> client = NetapiClient(__opts__)
>>> lowstate = {'client': 'local', 'tgt': '*', 'fun': 'test.ping', 'arg': ''}
>>> client.run(lowstate)
```

local(**args*, ***kwargs*)

Run *execution modules* synchronously

See *`salt.client.LocalClient.cmd()`* for all available parameters.

Sends a command from the master to the targeted minions. This is the same interface that Salt's own CLI uses. Note the `arg` and `kwarg` parameters are sent down to the minion(s) and the given function, `fun`, is called with those parameters.

Returns Returns the result from the execution module

local_async(**args*, ***kwargs*)

Run *execution modules* asynchronously

Wraps *`salt.client.LocalClient.run_job()`*.

Returns job ID

local_subset(**args*, ***kwargs*)

Run *execution modules* against subsets of minions

New in version 2016.3.0.

Wraps `salt.client.LocalClient.cmd_subset()`

runner(*fun*, *timeout=None*, *full_return=False*, ***kwargs*)

Run *runner modules <all-salt.runners>* synchronously

Wraps `salt.runner.RunnerClient.cmd_sync()`.

Note that runner functions must be called using keyword arguments. Positional arguments are not supported.

Returns Returns the result from the runner module

runner_async(*fun*, ***kwargs*)

Run *runner modules <all-salt.runners>* asynchronously

Wraps `salt.runner.RunnerClient.cmd_async()`.

Note that runner functions must be called using keyword arguments. Positional arguments are not supported.

Returns event data and a job ID for the executed function.

ssh(**args*, ***kwargs*)

Run salt-ssh commands synchronously

Wraps `salt.client.ssh.client.SSHClient.cmd_sync()`.

Returns Returns the result from the salt-ssh command

wheel(*fun*, ***kwargs*)

Run *wheel modules* synchronously

Wraps `salt.wheel.WheelClient.master_call()`.

Note that wheel functions must be called using keyword arguments. Positional arguments are not supported.

Returns Returns the result from the wheel module

wheel_async(*fun*, ***kwargs*)

Run *wheel modules* asynchronously

Wraps `salt.wheel.WheelClient.master_call()`.

Note that wheel functions must be called using keyword arguments. Positional arguments are not supported.

Returns Returns the result from the wheel module

Writing netapi modules

`netapi` modules, put simply, bind a port and start a service. They are purposefully open-ended and can be used to present a variety of external interfaces to Salt, and even present multiple interfaces at once.

See also:

The full list of netapi modules

Configuration

All `netapi` configuration is done in the *Salt master config* and takes a form similar to the following:

```
rest_cherrypy:
  port: 8000
  debug: True
  ssl_crt: /etc/pki/tls/certs/localhost.crt
  ssl_key: /etc/pki/tls/certs/localhost.key
```

The `__virtual__` function

Like all module types in Salt, `netapi` modules go through Salt's loader interface to determine if they should be loaded into memory and then executed.

The `__virtual__` function in the module makes this determination and should return `False` or a string that will serve as the name of the module. If the module raises an `ImportError` or any other errors, it will not be loaded.

The `start` function

The `start()` function will be called for each `netapi` module that is loaded. This function should contain the server loop that actually starts the service. This is started in a multiprocess.

Multiple instances

New in version 2016.11.0.

`rest_cherrypy` and `rest_tornado` support running multiple instances by copying and renaming entire directory of those. To start the copied multiple `netapi` modules, add configuration blocks for the copied `netapi` modules in the Salt Master config. The name of each added configuration block must match with the name of each directory of the copied `netapi` module.

Inline documentation

As with the rest of Salt, it is a best-practice to include liberal inline documentation in the form of a module docstring and docstrings on any classes, methods, and functions in your `netapi` module.

Loader "magic" methods

The loader makes the `__opts__` data structure available to any function in a `netapi` module.

CHAPTER

TWENTYTHREE

ARCHITECTURE

If you are used to configuration management tools that require you to plan down to the last detail before you install anything, you are probably wondering why this section doesn't appear before the installation instructions. With Salt, you can switch to a high availability architecture at any time, and add additional components to scale your deployment as you go.

Since a single Salt master can manage thousands of systems, we usually recommend that you start by deploying a single Salt master, and then modifying your deployment as needed for redundancy, geographical distribution, and scale.

23.1 High Availability Features in Salt

Salt supports several features for high availability and fault tolerance. Brief documentation for these features is listed alongside their configuration parameters in *Configuration file examples*.

23.1.1 Multimaster

Salt minions can connect to multiple masters at one time by configuring the *master* configuration parameter as a YAML list of all the available masters. By default, all masters are "hot", meaning that any master can direct commands to the Salt infrastructure.

In a multimaster configuration, each master must have the same cryptographic keys, and minion keys must be accepted on all masters separately. The contents of file_roots and pillar_roots need to be kept in sync with processes external to Salt as well

A tutorial on setting up multimaster with "hot" masters is here:

Multimaster Tutorial

23.1.2 Multimaster with Failover

Changing the `master_type` parameter from `str` to `failover` will cause minions to connect to the first responding master in the list of masters. Every *`master_alive_interval`* seconds the minions will check to make sure the current master is still responding. If the master does not respond, the minion will attempt to connect to the next master in the list. If the minion runs out of masters, the list will be recycled in case dead masters have been restored. Note that *`master_alive_interval`* must be present in the minion configuration, or else the recurring job to check master status will not get scheduled.

Failover can be combined with PKI-style encrypted keys, but PKI is NOT REQUIRED to use failover.

Multimaster with PKI and Failover is discussed in *this tutorial* `master_type: failover` can be combined with `master_shuffle: True` to spread minion connections across all masters (one master per minion, not each minion connecting to all masters). Adding Salt Syndics into the mix makes it possible to create a load-balanced Salt infrastructure. If a master fails, minions will notice and select another master from the available list.

23.1.3 Syndic

Salt's Syndic feature is a way to create differing infrastructure topologies. It is not strictly an HA feature, but can be treated as such.

With the syndic, a Salt infrastructure can be partitioned in such a way that certain masters control certain segments of the infrastructure, and "Master of Masters" nodes can control multiple segments underneath them.

Syndics are covered in depth in *Salt Syndic*.

23.1.4 Syndic with Multimaster

New in version 2015.5.0.

Syndic with Multimaster lets you connect a syndic to multiple masters to provide an additional layer of redundancy in a syndic configuration.

Syndics are covered in depth in *Salt Syndic*.

23.2 Salt Syndic

The most basic or typical Salt topology consists of a single Master node controlling a group of Minion nodes. An intermediate node type, called Syndic, when used offers greater structural flexibility and scalability in the construction of Salt topologies than topologies constructed only out of Master and Minion node types.

A Syndic node can be thought of as a special passthrough Minion node. A Syndic node consists of a `salt-syndic` daemon and a `salt-master` daemon running on the same system. The `salt-master` daemon running on the Syndic node controls a group of lower level Minion nodes and the `salt-syndic` daemon connects higher level Master node, sometimes called a Master of Masters.

The `salt-syndic` daemon relays publications and events between the Master node and the local `salt-master` daemon. This gives the Master node control over the Minion nodes attached to the `salt-master` daemon running on the Syndic node.

23.2.1 Configuring the Syndic

To setup a Salt Syndic you need to tell the Syndic node and its Master node about each other. If your Master node is located at `10.10.0.1`, then your configurations would be:

On the Syndic node:

```
# /etc/salt/master
syndic_master: 10.10.0.1  # may be either an IP address or a hostname
```

```
# /etc/salt/minion

# id is shared by the salt-syndic daemon and a possible salt-minion daemon
```

(continues on next page)

(continued from previous page)

```
# on the Syndic node
id: my_syndic
```

On the Master node:

```
# /etc/salt/master
order_masters: True
```

The *syndic_master* option tells the Syndic node where to find the Master node in the same way that the *master* option tells a Minion node where to find a Master node.

The *id* option is used by the `salt-syndic` daemon to identify with the Master node and if unset will default to the hostname or IP address of the Syndic just as with a Minion.

The *order_masters* option configures the Master node to send extra information with its publications that is needed by Syndic nodes connected directly to it.

Note: Each Syndic must provide its own `file_roots` directory. Files will not be automatically transferred from the Master node.

23.2.2 Configuring the Syndic with Multimaster

New in version 2015.5.0.

Syndic with Multimaster lets you connect a syndic to multiple masters to provide an additional layer of redundancy in a syndic configuration.

Higher level masters should first be configured in a multimaster configuration. See *Multimaster Tutorial*.

On the syndic, the *syndic_master* option is populated with a list of the higher level masters.

Since each syndic is connected to each master, jobs sent from any master are forwarded to minions that are connected to each syndic. If the `master_id` value is set in the master config on the higher level masters, job results are returned to the master that originated the request in a best effort fashion. Events/jobs without a `master_id` are returned to any available master.

23.2.3 Running the Syndic

The `salt-syndic` daemon is a separate process that needs to be started in addition to the `salt-master` daemon running on the Syndic node. Starting the `salt-syndic` daemon is the same as starting the other Salt daemons.

The Master node in many ways sees the Syndic as an ordinary Minion node. In particular, the Master will need to accept the Syndic's Minion key as it would for any other Minion.

On the Syndic node:

```
# salt-syndic
or
# service salt-syndic start
```

On the Master node:

```
# salt-key -a my_syndic
```

The Master node will now be able to control the Minion nodes connected to the Syndic. Only the Syndic key will be listed in the Master node's key registry but this also means that key activity between the Syndic's Minions and the Syndic does not encumber the Master node. In this way, the Syndic's key on the Master node can be thought of as a placeholder for the keys of all the Minion and Syndic nodes beneath it, giving the Master node a clear, high level structural view on the Salt cluster.

On the Master node:

```
# salt-key -L
Accepted Keys:
my_syndic
Denied Keys:
Unaccepted Keys:
Rejected Keys:

# salt '*' test.version
minion_1:
    2018.3.4
minion_2:
    2018.3.4
minion_4:
    2018.3.4
minion_3:
    2018.3.4
```

23.2.4 Topology

A Master node (a node which is itself not a Syndic to another higher level Master node) must run a `salt-master` daemon and optionally a `salt-minion` daemon.

A Syndic node must run `salt-syndic` and `salt-master` daemons and optionally a `salt-minion` daemon.

A Minion node must run a `salt-minion` daemon.

When a `salt-master` daemon issues a command, it will be received by the Syndic and Minion nodes directly connected to it. A Minion node will process the command in the way it ordinarily would. On a Syndic node, the `salt-syndic` daemon will relay the command to the `salt-master` daemon running on the Syndic node, which then propagates the command to the Minions and Syndics connected to it.

When events and job return data are generated by `salt-minion` daemons, they are aggregated by the `salt-master` daemon they are connected to, which `salt-master` daemon then relays the data back through its `salt-syndic` daemon until the data reaches the Master or Syndic node that issued the command.

23.2.5 Syndic wait

`syndic_wait` is a master configuration file setting that specifies the number of seconds the Salt client should wait for additional syndics to check in with their lists of expected minions before giving up. This value defaults to `5` seconds.

The `syndic_wait` setting is necessary because the higher-level master does not have a way of knowing which minions are below the syndics. The higher-level master has its own list of expected minions and the masters below them have their own lists as well, so the Salt client does not how long to wait for all returns. The `syndic_wait` option allows time for all minions to return to the Salt client.

Note: To reduce the amount of time the CLI waits for Minions to respond, install a Minion on the Syndic or tune the value of the `syndic_wait` configuration.

While it is possible to run a Syndic without a Minion installed on the same system, it is recommended, for a faster CLI response time, to do so. Without a Minion installed on the Syndic node, the timeout value of `syndic_wait` increases significantly - about three-fold. With a Minion installed on the Syndic, the CLI timeout resides at the value defined in `syndic_wait`.

Note: If you have a very large infrastructure or many layers of Syndics, you may find that the CLI doesn't wait long enough for the Syndics to return their events. If you think this is the case, you can set the *`syndic_wait`* value in the Master configs on the Master or Syndic nodes from which commands are executed. The default value is `5`, and should work for the majority of deployments.

In order for a Master or Syndic node to return information from Minions that are below their Syndics, the CLI requires a short wait time in order to allow the Syndics to gather responses from their Minions. This value is defined in the *`syndic_wait`* config option and has a default of five seconds.

23.2.6 Syndic config options

These are the options that can be used to configure a Syndic node. Note that other than `id`, Syndic config options are placed in the Master config on the Syndic node.

- *`id`*: Syndic id (shared by the `salt-syndic` daemon with a potential `salt-minion` daemon on the same system)
- *`syndic_master`*: Master node IP address or hostname
- *`syndic_master_port`*: Master node ret_port
- *`syndic_log_file`*: path to the logfile (absolute or not)
- *`syndic_pidfile`*: path to the pidfile (absolute or not)
- *`syndic_wait`*: time in seconds to wait on returns from this syndic

23.2.7 Minion Data Cache

Beginning with Salt 2016.11.0, the *Pluggable Minion Data Cache* was introduced. The minion data cache contains the Salt Mine data, minion grains, and minion pillar information cached on the Salt Master. By default, Salt uses the `localfs` cache module, but other external data stores can be used instead.

Using a pluggable minion cache modules allows for the data stored on a Salt Master about Salt Minions to be replicated on other Salt Masters the Minion is connected to. Please see the *Minion Data Cache* documentation for more information and configuration examples.

CHAPTER

TWENTYFOUR

MINION DATA CACHE

New in version 2016.11.0.

The Minion data cache contains the Salt Mine data, minion grains and minion pillar information cached on the Salt Master. By default, Salt uses the `localfs` cache module to save the data in a `msgpack` file on the Salt Master.

24.1 Pluggable Data Cache

While the default Minion data cache is the `localfs` cache, other external data stores can also be used to store this data such as the `consul` module. To configure a Salt Master to use a different data store, the `cache` setting needs to be established:

```
cache: consul
```

The pluggable data cache streamlines using various Salt topologies such as a *Multi-Master* or *Salt Syndics* configuration by allowing the data stored on the Salt Master about a Salt Minion to be available to other Salt Syndics or Salt Masters that a Salt Minion is connected to.

Additional minion data cache modules can be easily created by modeling the custom data store after one of the existing cache modules.

See *cache modules* for a current list.

24.2 Configuring the Minion Data Cache

The default `localfs` Minion data cache module doesn't require any configuration. External data cache modules with external data stores such as Consul require a configuration setting in the master config.

Here's an example config for Consul:

```
consul.host: 127.0.0.1
consul.port: 8500
consul.token: None
consul.scheme: http
consul.consistency: default
consul.dc: dc1
consul.verify: True

cache: consul
```

CHAPTER

TWENTYFIVE

SLOTS

New in version 2018.3.0.

Changed in version 3000.

Note: This functionality is under development and could be changed in the future releases

Many times it is useful to store the results of a command during the course of an execution. Salt Slots are designed to allow you to store this information and use it later during the *highstate* or other job execution.

Slots extend the state syntax and allows you to do things right before the state function is executed. So you can make a decision in the last moment right before a state is executed.

25.1 Execution functions

Note: Using execution modules return data as a state values is a first step of Slots development. Other functionality is under development.

Slots allow you to use the return from a remote-execution function as an argument value in states.

Slot syntax looks close to the simple python function call.

```
__slot__:salt:<module>.<function>(<args>, ..., <kwargs...>, ...)
```

For the 3000 release, this syntax has been updated to support parsing functions which return dictionaries and for appending text to the slot result.

```
__slot__:salt:<module>.<function>(<args>..., <kwargs...>, ...).dictionary ~ append
```

There are some specifics in the syntax coming from the execution functions nature and a desire to simplify the user experience. First one is that you don't need to quote the strings passed to the slots functions. The second one is that all arguments handled as strings.

Here is a simple example:

```
copy-some-file:
  file.copy:
    - name: __slot__:salt:test.echo(text=/tmp/some_file)
    - source: __slot__:salt:test.echo(/etc/hosts)
```

This will execute the `test.echo` execution functions right before calling the state. The functions in the example will return */tmp/some_file* and */etc/hosts* strings that will be used as a target and source arguments in the state function *file.copy*.

Here is an example of result parsing and appending:

```
file-in-user-home:
  file.copy:
    - name: __slot__:salt:user.info(someuser).home ~ /subdirectory
    - source: salt://somefile
```

CHAPTER

TWENTYSIX

WINDOWS

This section contains details on the Windows Package Manager, and specific information you need to use Salt on Windows.

26.1 Windows Software Repository

Note: In 2015.8.0 and later, the Windows Software Repository cache is compiled on the Salt Minion, which enables pillar, grains and other things to be available during compilation time. To support this new functionality, a next-generation (ng) package repository was created. See the *Changes in Version 2015.8.0* for details.

The SaltStack Windows Software Repository provides a package manager and software repository similar to what is provided by yum and apt on Linux. This repository enables the installation of software using the installers on remote Windows systems.

In many senses, the operation is similar to that of the other package managers salt is aware of:

- the `pkg.installed` and similar states work on Windows.
- the `pkg.install` and similar module functions work on Windows.

High level differences to yum and apt are:

- The repository metadata (SLS files) is hosted through either salt or git.
- Packages can be downloaded from within the salt repository, a git repository or from HTTP(S) or FTP URLs.
- No dependencies are managed. Dependencies between packages needs to be managed manually.

Requirements:

- GitPython 0.3 or later, or pygit2 0.20.3 with libgit 0.20.0 or later installed on your Salt master. The Windows package definitions are downloaded and updated using Git.

26.1.1 Configuration

Populate the Repository

The SLS files used to install Windows packages are not distributed by default with Salt. Run the following command to initialize the repository on your Salt master:

```
salt-run winrepo.update_git_repos
```

Sync Repo to Windows Minions

Run `pkg.refresh_db` on each of your Windows minions to synchronize the package repository.

```
salt -G 'os:windows' pkg.refresh_db
```

Note: Use `pkg.refresh_db` from 2016.11 when developing new Windows package definitions to check for errors in the definitions against one or more Windows minions.

26.1.2 Install Windows Software

After completing the configuration steps, you are ready to manage software on your Windows minions.

Show Installed Packages

```
salt -G 'os:windows' pkg.list_pkgs
```

Install a Package

You can query the available version of a package using the Salt pkg module.

```
salt winminion pkg.list_available firefox

winminion:
    - 15.0.1
    - 16.0.2
    - 17.0.1
```

As you can see, there are three versions of Firefox available for installation. You can refer a software package by its `name` or its `full_name` surround by single quotes.

```
salt winminion pkg.install 'firefox'
```

The above line will install the latest version of Firefox.

```
salt winminion pkg.install 'firefox' version=16.0.2
```

The above line will install version 16.0.2 of Firefox.

If a different version of the package is already installed it will be replaced with the version in the winrepo (only if the package itself supports live updating).

You can also specify the full name:

```
salt winminion pkg.install 'Mozilla Firefox 17.0.1 (x86 en-US)'
```

26.1.3 Uninstall Windows Software

Uninstall software using the pkg module:

```
salt winminion pkg.remove firefox
salt winminion pkg.purge firefox
```

Note: `pkg.purge` just executes `pkg.remove` on Windows. At some point in the future `pkg.purge` may direct the installer to remove all configs and settings for software packages that support that option.

26.1.4 Repository Location

Salt maintains a repository of SLS files to install a large number of Windows packages:

- 2015.8.0 and later minions: https://github.com/saltstack/salt-winrepo-ng
- Earlier releases: https://github.com/saltstack/salt-winrepo

By default, these repositories are mirrored to `/srv/salt/win/repo-ng` and `/srv/salt/win/repo`.

This location can be changed in the master config file by setting the *`winrepo_dir_ng`* and *`winrepo_dir`* options.

26.1.5 Maintaining Windows Repo Definitions in Git Repositories

Windows software package definitions can be hosted in one or more Git repositories. The default repositories are hosted on GitHub by SaltStack. These include software definition files for various open source software projects. These software definition files are `.sls` files. There are two default repositories: `salt-winrepo` and `salt-winrepo-ng`. `salt-winrepo` contains software definition files for older minions (older than 2015.8.0). `salt-winrepo-ng` is for newer minions (2015.8.0 and newer).

Each software definition file contains all the information salt needs to install that software on a minion including the HTTP or FTP locations of the installer files, required command-line switches for silent install, etc. Anyone is welcome to send a pull request to this repo to add new package definitions. The repos can be browsed here: salt-winrepo salt-winrepo-ng

Note: The newer software definition files are run through the salt's parser which allows for the use of jinja.

Configure which git repositories the master can search for package definitions by modifying or extending the *`winrepo_remotes`* and *`winrepo_remotes_ng`* options.

Important: `winrepo_remotes` was called `win_gitrepos` in Salt versions earlier than 2015.8.0

Package definitions are pulled down from the online git repository by running the *`winrepo.update_git_repos`* runner. This command is run on the master:

```
salt-run winrepo.update_git_repos
```

This will pull down the software definition files for older minions (`salt-winrepo`) and new minions (`salt-winrepo-ng`). They are stored in the `file_roots` under `win/repo/salt-winrepo` and `win/repo-ng/salt-winrepo-ng` respectively.

Important: If you have customized software definition files that aren't maintained in a repository, those should be stored under `win/repo` for older minions and `win/repo-ng` for newer minions. The reason for this is that the contents of `win/repo/salt-winrepo` and `win/repo-ng/salt-winrepo-ng` are wiped out every time you run a `winrepo.update_git_repos`.

Additionally, when you run `winrepo.genrepo` and `pkg.refresh_db` the entire contents under `win/repo` and `win/repo-ng`, to include all subdirectories, are used to create the meta database file.

The next step (if you have older minions) is to create the meta database file for the repo (`winrepo.p`). This is done by running the *`winrepo.genrepo`* runner. This is also run on the master:

```
salt-run winrepo.genrepo
```

Note: If you have only 2015.8.0 and newer minions, you no longer need to run `salt-run winrepo.genrepo` on the master.

Finally, you need to refresh the minion database by running the *`pkg.refresh_db`* command. This is run on the master as well:

```
salt '*' pkg.refresh_db
```

On older minions (older than 2015.8.0) this will copy the winrepo.p file down to the minion. On newer minions (2015.8.0 and newer) this will copy all the software definition files (.sls) down to the minion and then create the meta database file (`winrepo.p`) locally. The reason this is done locally is because the jinja needs to be parsed using the minion's grains.

Important: Every time you modify the software definition files on the master, either by running `salt-run winrepo.update_git_repos`, modifying existing files, or by creating your own, you need to refresh the database on your minions. For older minions, that means running `salt-run winrepo.genrepo` and then `salt '*' pkg.refresh_db`. For newer minions (2015.8.0 and newer) it is just `salt '*' pkg.refresh_db`.

Note: If the `winrepo.genrepo` or the `pkg.refresh_db` fails, it is likely a problem with the jinja in one of the software definition files. This will cause the operations to stop. You'll need to fix the syntax in order for the meta database file to be created successfully.

To disable one of the repos, set it to an empty list `[]` in the master config. For example, to disable *`winrepo_remotes`* set the following in the master config file:

```
winrepo_remotes: []
```

26.1.6 Creating a Package Definition SLS File

The package definition file is a YAML file that contains all the information needed to install a piece of software using salt. It defines information about the package to include version, full name, flags required for the installer and uninstaller, whether or not to use the Windows task scheduler to install the package, where to find the installation package, etc.

Take a look at this example for Firefox:

```
firefox:
  '17.0.1':
    installer: 'salt://win/repo/firefox/English/Firefox Setup 17.0.1.exe'
    full_name: Mozilla Firefox 17.0.1 (x86 en-US)
    locale: en_US
    reboot: False
    install_flags: '-ms'
    uninstaller: '%ProgramFiles(x86)%/Mozilla Firefox/uninstall/helper.exe'
    uninstall_flags: '/S'
  '16.0.2':
    installer: 'salt://win/repo/firefox/English/Firefox Setup 16.0.2.exe'
    full_name: Mozilla Firefox 16.0.2 (x86 en-US)
    locale: en_US
    reboot: False
    install_flags: '-ms'
    uninstaller: '%ProgramFiles(x86)%/Mozilla Firefox/uninstall/helper.exe'
    uninstall_flags: '/S'
  '15.0.1':
    installer: 'salt://win/repo/firefox/English/Firefox Setup 15.0.1.exe'
    full_name: Mozilla Firefox 15.0.1 (x86 en-US)
    locale: en_US
    reboot: False
    install_flags: '-ms'
    uninstaller: '%ProgramFiles(x86)%/Mozilla Firefox/uninstall/helper.exe'
    uninstall_flags: '/S'
```

Each software definition file begins with a package name for the software. As in the example above `firefox`. The next line is indented two spaces and contains the version to be defined. As in the example above, a software definition file can define multiple versions for the same piece of software. The lines following the version are indented two more spaces and contain all the information needed to install that package.

Warning: The package name and the `full_name` must be unique to all other packages in the software repository.

The version line is the version for the package to be installed. It is used when you need to install a specific version of a piece of software.

Warning: The version must be enclosed in quotes, otherwise the YAML parser will remove trailing zeros.

Note: There are unique situations where previous versions are unavailable. Take Google Chrome for example. There is only one URL provided for a standalone installation of Google Chrome.

(https://dl.google.com/edgedl/chrome/install/GoogleChromeStandaloneEnterprise.msi)

When a new version is released, the URL just points to the new version. To handle situations such as these, set the version to *latest*. Salt will install the version of Chrome at the URL and report that version. Here's an example:

```
chrome:
  latest:
    full_name: 'Google Chrome'
    installer: 'https://dl.google.com/edgedl/chrome/install/
↪GoogleChromeStandaloneEnterprise.msi'
```

(continues on next page)

(continued from previous page)

```
    install_flags: '/qn /norestart'
    uninstaller: 'https://dl.google.com/edgedl/chrome/install/
↪GoogleChromeStandaloneEnterprise.msi'
    uninstall_flags: '/qn /norestart'
    msiexec: True
    locale: en_US
    reboot: False
```

Available parameters are as follows:

param str full_name The Full Name for the software as shown in "Programs and Features" in the control panel. You can also get this information by installing the package manually and then running `pkg.list_pkgs`. Here's an example of the output from `pkg.list_pkgs`:

```
salt 'test-2008' pkg.list_pkgs
test-2008
    ----------
    7-Zip 9.20 (x64 edition):
        9.20.00.0
    Microsoft .NET Framework 4 Client Profile:
        4.0.30319,4.0.30319
    Microsoft .NET Framework 4 Extended:
        4.0.30319,4.0.30319
    Microsoft Visual C++ 2008 Redistributable - x64 9.0.21022:
        9.0.21022
    Mozilla Firefox 17.0.1 (x86 en-US):
        17.0.1
    Mozilla Maintenance Service:
        17.0.1
    NSClient++ (x64):
        0.3.8.76
    Notepad++:
        6.4.2
    Salt Minion 0.16.0:
        0.16.0
```

Notice the Full Name for Firefox: `Mozilla Firefox 17.0.0 (x86 en-US)`. That's exactly what's in the `full_name` parameter in the software definition file.

If any of the software installed on the machine matches one of the software definition files in the repository, the full_name will be automatically renamed to the package name. The example below shows the `pkg.list_pkgs` for a machine that already has Mozilla Firefox 17.0.1 installed.

```
test-2008:
    ----------
    7zip:
        9.20.00.0
    Microsoft .NET Framework 4 Client Profile:
        4.0.30319,4.0.30319
    Microsoft .NET Framework 4 Extended:
        4.0.30319,4.0.30319
    Microsoft Visual C++ 2008 Redistributable - x64 9.0.21022:
        9.0.21022
    Mozilla Maintenance Service:
        17.0.1
    Notepad++:
        6.4.2
```

(continues on next page)

(continued from previous page)

```
    Salt Minion 0.16.0:
        0.16.0
    firefox:
        17.0.1
    nsclient:
        0.3.9.328
```

Important: The version number and `full_name` need to match the output from `pkg.list_pkgs` so that the status can be verified when running a highstate.

Note: It is still possible to successfully install packages using `pkg.install`, even if the `full_name` or the version number don't match. However, this can make troubleshooting issues difficult, so be careful.

Tip: To force salt to display the full name when there's already an existing package definition file on the system, you can pass a bogus `saltenv` parameter to the command like so: `pkg.list_pkgs saltenv=NotARealEnv`

---

param str installer The path to the `.exe` or `.msi` to use to install the package. This can be a path or a URL. If it is a URL or a salt path (`salt://`), the package will be cached locally and then executed. If it is a path to a file on disk or a file share, it will be executed directly.

Note: If storing software in the same location as the winrepo it is best practice to place each installer in its own directory rather than the root of winrepo. Then you can place your package definition file in the same directory. It is best practice to name the file `init.sls`. This will be picked up by `pkg.refresh_db` and processed properly.

param str install_flags Any flags that need to be passed to the installer to make it perform a silent install. These can often be found by adding `/?` or `/h` when running the installer from the command-line. A great resource for finding these silent install flags can be found on the WPKG project's wiki:

Warning: Salt will not return if the installer is waiting for user input so it is imperative that the software package being installed has the ability to install silently.

param str uninstaller The path to the program used to uninstall this software. This can be the path to the same *exe* or *msi* used to install the software. It can also be a GUID. You can find this value in the registry under the following keys:

- Software\Microsoft\Windows\CurrentVersion\Uninstall
- Software\Wow6432None\Microsoft\Windows\CurrentVersion\Uninstall

param str uninstall_flags Any flags that need to be passed to the uninstaller to make it perform a silent uninstall. These can often be found by adding `/?` or `/h` when running the uninstaller from the command-line. A great resource for finding these silent install flags can be found on the WPKG project's wiki:

> **Warning:** Salt will not return if the uninstaller is waiting for user input so it is imperative that the software package being uninstalled has the ability to uninstall silently.

Here are some examples of installer and uninstaller settings:

```
7zip:
  '9.20.00.0':
    installer: salt://win/repo/7zip/7z920-x64.msi
    full_name: 7-Zip 9.20 (x64 edition)
    reboot: False
    install_flags: '/qn /norestart'
    msiexec: True
    uninstaller: '{23170F69-40C1-2702-0920-000001000000}'
    uninstall_flags: '/qn /norestart'
```

Alternatively the `uninstaller` can also simply repeat the URL of an msi file:

```
7zip:
  '9.20.00.0':
    installer: salt://win/repo/7zip/7z920-x64.msi
    full_name: 7-Zip 9.20 (x64 edition)
    reboot: False
    install_flags: '/qn /norestart'
    msiexec: True
    uninstaller: salt://win/repo/7zip/7z920-x64.msi
    uninstall_flags: '/qn /norestart'
```

param msiexec This tells salt to use `msiexec /i` to install the package and `msiexec /x` to uninstall. This is for `.msi` installations. Possible options are: True, False or the path to `msiexec.exe` on your system

```
7zip:
  '9.20.00.0':
    installer: salt://win/repo/7zip/7z920-x64.msi
    full_name: 7-Zip 9.20 (x64 edition)
    reboot: False
    install_flags: '/qn /norestart'
    msiexec: 'C:\Windows\System32\msiexec.exe'
    uninstaller: salt://win/repo/7zip/7z920-x64.msi
    uninstall_flags: '/qn /norestart'
```

param bool allusers This parameter is specific to `.msi` installations. It tells `msiexec` to install the software for all users. The default is `True`.

param bool cache_dir If `True` and the installer URL begins with `salt://`, the entire directory where the installer resides will be recursively cached. This is useful for installers that depend on other files in the same directory for installation.

> **Warning:** Be aware that all files and directories in the same location as the installer file will be copied down to the minion. If you place your installer file in the root of winrepo (`/srv/salt/win/repo-ng`) and `cache_dir: True` the entire contents of winrepo will be cached to the minion. Therefore, it is best practice to place your installer files in a subdirectory if they are to be stored in winrepo.

param str cache_file When the installer URL begins with `salt://`, this indicates a single file to copy down for use with the installer. It is copied to the same location as the installer. Use this over `cache_dir` if there are many files in the directory and you only need a specific file and don't want to cache additional files that may reside in the installer directory.

Here's an example for a software package that has dependent files:

```
sqlexpress:
  '12.0.2000.8':
    installer: 'salt://win/repo/sqlexpress/setup.exe'
    full_name: Microsoft SQL Server 2014 Setup (English)
    reboot: False
    install_flags: '/ACTION=install /IACCEPTSQLSERVERLICENSETERMS /Q'
    cache_dir: True
```

param bool use_scheduler If `True`, Windows will use the task scheduler to run the installation. This is useful for running the Salt installation itself as the installation process kills any currently running instances of Salt.

param str source_hash This tells Salt to compare a hash sum of the installer to the provided hash sum before execution. The value can be formatted as `<hash_algorithm>=<hash_sum>`, or it can be a URI to a file containing the hash sum.

For a list of supported algorithms, see the hashlib documentation.

Here's an example of source_hash usage:

```
messageanalyzer:
  '4.0.7551.0':
    full_name: 'Microsoft Message Analyzer'
    installer: 'salt://win/repo/messageanalyzer/MessageAnalyzer64.msi'
    install_flags: '/quiet /norestart'
    uninstaller: '{1CC02C23-8FCD-487E-860C-311EC0A0C933}'
    uninstall_flags: '/quiet /norestart'
    msiexec: True
    source_hash: 'sha1=62875ff451f13b10a8ff988f2943e76a4735d3d4'
```

param bool reboot Not implemented

param str locale Not implemented

Examples can be found at https://github.com/saltstack/salt-winrepo-ng

26.1.7 Managing Windows Software on a Standalone Windows Minion

The Windows Package Repository functions similar in a standalone environment, with a few differences in the configuration.

To replace the winrepo runner that is used on the Salt master, an `execution module` exists to provide the same functionality to standalone minions. The functions are named the same as the ones in the runner, and are used in the same way; the only difference is that `salt-call` is used instead of `salt-run`:

```
salt-call winrepo.update_git_repos
salt-call winrepo.genrepo
salt-call pkg.refresh_db
```

After executing the previous commands the repository on the standalone system is ready to use.

Custom Location for Repository SLS Files

If *`file_roots`* has not been modified in the minion configuration, then no additional configuration needs to be added to the minion configuration. The `winrepo.genrepo` function from the `winrepo` execution module will by default look for the filename specified by *`winrepo_cachefile`* within `C:\salt\srv\salt\win\repo`.

If the *`file_roots`* parameter has been modified, then *`winrepo_dir`* must be modified to fall within that path, at the proper relative path. For example, if the `base` environment in *`file_roots`* points to `D:\foo`, and *`winrepo_source_dir`* is `salt://win/repo`, then *`winrepo_dir`* must be set to `D:\foo\win\repo` to ensure that `winrepo.genrepo` puts the cachefile into right location.

26.1.8 Configuration options for Minions 2015.8.0 and later

On newer minions (2015.8.0 and later), the *`winrepo_source_dir`* config parameter (default: `salt://win/repo-ng`) controls where *`pkg.refresh_db`* looks for the software definition files that will be downloaded to the minion and used to generate the local database file (`winrepo.p`).

Software package definitions are automatically refreshed if stale after *`winrepo_cache_expire_max`*. Running a highstate normal forces the refresh of the package definition and generation of the meta database, unless the meta database is younger than *`winrepo_cache_expire_max`*. Refreshing the package definitions can take some time, these options were introduced to allow more control of when it occurs.

It's important use *`pkg.refresh_db`* to check for errors and ensure the latest package definition is on any minion your testing new definitions on.

26.1.9 Configuration options for Minions before 2015.8.0

On older minions (before 2015.8.0), the *`winrepo_source_dir`* config parameter (default: `salt://win/repo`) controls where *`pkg.refresh_db`* looks for the cachefile (default: `winrepo.p`). This means that the default location for the winrepo cachefile would be `salt://win/repo/winrepo.p`. Both :conf_minion: winrepo_source_dir' and *`winrepo_cachefile`* can be adjusted to match the actual location of this file on the Salt fileserver.

If connected to a master, the minion will by default look for the winrepo cachefile (the file generated by the *`winrepo.genrepo runner`*) at `salt://win/repo/winrepo.p`. If the cachefile is in a different path on the salt fileserver, then *`win_repo_cachefile`* will need to be updated to reflect the proper location.

26.1.10 Changes in Version 2015.8.0

Git repository management for the Windows Software Repository has changed in version 2015.8.0, and several master/minion config parameters have been renamed to make their naming more consistent with each other.

For a list of the winrepo config options, see *here* for master config options, and *here* for configuration options for masterless Windows minions.

On the master, the *`winrepo.update_git_repos`* runner has been updated to use either pygit2 or GitPython to checkout the git repositories containing repo data. If pygit2 or GitPython is installed, existing winrepo git checkouts should be removed after upgrading to 2015.8.0, to allow them to be checked out again by running *`winrepo.update_git_repos`*.

If neither GitPython nor pygit2 are installed, then Salt will fall back to the pre-existing behavior for *`winrepo.update_git_repos`*, and a warning will be logged in the master log.

Note: Standalone Windows minions do not support the new GitPython/pygit2 functionality, and will instead use the `git.latest` state to keep repositories up-to-date. More information on how to use the Windows Software Repo on a standalone minion can be found *here*.

Config Parameters Renamed

Many of the legacy winrepo configuration parameters have changed in version 2015.8.0 to make the naming more consistent. The old parameter names will still work, but a warning will be logged indicating that the old name is deprecated.

Below are the parameters which have changed for version 2015.8.0:

Master Config

Old Name	New Name
win_repo	`winrepo_dir`
win_repo_mastercachefile	`winrepo_cachefile`
win_gitrepos	`winrepo_remotes`

Note: `winrepo_cachefile` is no longer used by 2015.8.0 and later minions, and the `winrepo_dir` setting is replaced by `winrepo_dir_ng` for 2015.8.0 and later minions.

See *here* for detailed information on all master config options for the Windows Repo.

Minion Config

Old Name	New Name
win_repo	`winrepo_dir`
win_repo_cachefile	`winrepo_cachefile`
win_gitrepos	`winrepo_remotes`

See *here* for detailed information on all minion config options for the Windows Repo.

pygit2/GitPython Support for Maintaining Git Repos

The `winrepo.update_git_repos` runner (and the corresponding `remote execution function` for standalone minions) now makes use of the same underlying code used by the *Git Fileserver Backend* and `Git External Pillar` to maintain and update its local clones of git repositories. If a compatible version of either pygit2 (0.20.3 and later) or GitPython (0.3.0 or later) is installed, then Salt will use it instead of the old method (which invokes the `git.latest` state).

Note: If compatible versions of both pygit2 and GitPython are installed, then Salt will prefer pygit2, to override this behavior use the `winrepo_provider` configuration parameter:

```
winrepo_provider: gitpython
```

The `winrepo execution module` (discussed above in the *Managing Windows Software on a Standalone Windows Minion* section) does not yet officially support the new pygit2/GitPython functionality, but if either pygit2 or GitPython is installed into Salt's bundled Python then it *should* work. However, it should be considered experimental at this time.

To minimize potential issues, it is a good idea to remove any winrepo git repositories that were checked out by the old (pre-2015.8.0) winrepo code when upgrading the master to 2015.8.0 or later, and run *`winrepo.update_git_repos`* to clone them anew after the master is started.

Additional added features include the ability to access authenticated git repositories (**NOTE**: pygit2 only), and to set per-remote config settings. An example of this would be the following:

```
winrepo_remotes:
  - https://github.com/saltstack/salt-winrepo.git
  - git@github.com:myuser/myrepo.git:
    - pubkey: /path/to/key.pub
    - privkey: /path/to/key
    - passphrase: myaw3s0m3pa$$phr4$3
  - https://github.com/myuser/privaterepo.git:
    - user: mygithubuser
    - password: CorrectHorseBatteryStaple
```

Note: Per-remote configuration settings work in the same fashion as they do in gitfs, with global parameters being overridden by their per-remote counterparts (for instance, setting *`winrepo_passphrase`* would set a global passphrase for winrepo that would apply to all SSH-based remotes, unless overridden by a `passphrase` per-remote parameter).

See *here* for more a more in-depth explanation of how per-remote configuration works in gitfs, the same principles apply to winrepo.

There are a couple other changes in how Salt manages git repos using pygit2/GitPython. First of all, a `clean` argument has been added to the *`winrepo.update_git_repos`* runner, which (if set to `True`) will tell the runner to dispose of directories under the *`winrepo_dir`* which are not explicitly configured. This prevents the need to manually remove these directories when a repo is removed from the config file. To clean these old directories, just pass `clean=True`, like so:

```
salt-run winrepo.update_git_repos clean=True
```

However, if a mix of git and non-git Windows Repo definition files are being used, then this should *not* be used, as it will remove the directories containing non-git definitions.

The other major change is that collisions between repo names are now detected, and the *`winrepo.update_git_repos`* runner will not proceed if any are detected. Consider the following configuration:

```
winrepo_remotes:
  - https://foo.com/bar/baz.git
  - https://mydomain.tld/baz.git
  - https://github.com/foobar/baz
```

The *`winrepo.update_git_repos`* runner will refuse to update repos here, as all three of these repos would be checked out to the same directory. To work around this, a per-remote parameter called `name` can be used to resolve these conflicts:

```
winrepo_remotes:
  - https://foo.com/bar/baz.git
  - https://mydomain.tld/baz.git:
    - name: baz_junior
  - https://github.com/foobar/baz:
    - name: baz_the_third
```

26.1.11 Troubleshooting

Incorrect name/version

If the package seems to install properly, but salt reports a failure then it is likely you have a version or `full_name` mismatch.

Check the exact `full_name` and version used by the package. Use `pkg.list_pkgs` to check that the names and version exactly match what is installed.

Changes to sls files not being picked up

Ensure you have (re)generated the repository cache file (for older minions) and then updated the repository cache on the relevant minions:

```
salt-run winrepo.genrepo
salt winminion pkg.refresh_db
```

Packages management under Windows 2003

On Windows server 2003, you need to install optional Windows component ”wmi Windows installer provider” to have full list of installed packages. If you don’t have this, salt-minion can’t report some installed software.

How Success and Failure are Reported

The install state/module function of the Windows package manager works roughly as follows:

1. Execute `pkg.list_pkgs` and store the result
2. Check if any action needs to be taken. (i.e. compare required package and version against `pkg.list_pkgs` results)
3. If so, run the installer command.
4. Execute `pkg.list_pkgs` and compare to the result stored from before installation.
5. Success/Failure/Changes will be reported based on the differences between the original and final `pkg.list_pkgs` results.

If there are any problems in using the package manager it is likely due to the data in your sls files not matching the difference between the pre and post `pkg.list_pkgs` results.

26.2 Windows-specific Behaviour

Salt is capable of managing Windows systems, however due to various differences between the operating systems, there are some things you need to keep in mind.

This document will contain any quirks that apply across Salt or generally across multiple module functions. Any Windows-specific behavior for particular module functions will be documented in the module function documentation. Therefore this document should be read in conjunction with the module function documentation.

26.2.1 Group parameter for files

Salt was originally written for managing Unix-based systems, and therefore the file module functions were designed around that security model. Rather than trying to shoehorn that model on to Windows, Salt ignores these parameters and makes non-applicable module functions unavailable instead.

One of the commonly ignored parameters is the `group` parameter for managing files. Under Windows, while files do have a 'primary group' property, this is rarely used. It generally has no bearing on permissions unless intentionally configured and is most commonly used to provide Unix compatibility (e.g. Services For Unix, NFS services).

Because of this, any file module functions that typically require a group, do not under Windows. Attempts to directly use file module functions that operate on the group (e.g. `file.chgrp`) will return a pseudo-value and cause a log message to appear. No group parameters will be acted on.

If you do want to access and change the 'primary group' property and understand the implications, use the `file.get_pgid` or `file.get_pgroup` functions or the `pgroup` parameter on the `file.chown` module function.

26.2.2 Dealing with case-insensitive but case-preserving names

Windows is case-insensitive, but however preserves the case of names and it is this preserved form that is returned from system functions. This causes some issues with Salt because it assumes case-sensitive names. These issues generally occur in the state functions and can cause bizarre looking errors.

To avoid such issues, always pretend Windows is case-sensitive and use the right case for names, e.g. specify `user=Administrator` instead of `user=administrator`.

Follow issue #11801 for any changes to this behavior.

26.2.3 Dealing with various username forms

Salt does not understand the various forms that Windows usernames can come in, e.g. username, mydomain\username, username@mydomain.tld can all refer to the same user. In fact, Salt generally only considers the raw username value, i.e. the username without the domain or host information.

Using these alternative forms will likely confuse Salt and cause odd errors to happen. Use only the raw username value in the correct case to avoid problems.

Follow issue #11801 for any changes to this behavior.

26.2.4 Specifying the None group

Each Windows system has built-in _None_ group. This is the default 'primary group' for files for users not on a domain environment.

Unfortunately, the word _None_ has special meaning in Python - it is a special value indicating 'nothing', similar to `null` or `nil` in other languages.

To specify the None group, it must be specified in quotes, e.g. `./salt '*' file.chpgrp C:\path\to\file "'None'"`.

26.2.5 Symbolic link loops

Under Windows, if any symbolic link loops are detected or if there are too many levels of symlinks (defaults to 64), an error is always raised.

For some functions, this behavior is different to the behavior on Unix platforms. In general, avoid symlink loops on either platform.

CHAPTER

TWENTYSEVEN

DEVELOPING SALT

27.1 Overview

In its most typical use, Salt is a software application in which clients, called ”minions” can be commanded and controlled from a central command server called a ”master”.

Commands are normally issued to the minions (via the master) by calling a client script simply called, ’salt’.

Salt features a pluggable transport system to issue commands from a master to minions. The default transport is ZeroMQ.

27.2 Salt Client

27.2.1 Overview

The salt client is run on the same machine as the Salt Master and communicates with the salt-master to issue commands and to receive the results and display them to the user.

The primary abstraction for the salt client is called ’LocalClient’.

When LocalClient wants to publish a command to minions, it connects to the master by issuing a request to the master’s ReqServer (TCP: 4506)

The LocalClient system listens to responses for its requests by listening to the master event bus publisher (master_event_pub.ipc).

27.3 Salt Master

27.3.1 Overview

The salt-master daemon runs on the designated Salt master and performs functions such as authenticating minions, sending, and receiving requests from connected minions and sending and receiving requests and replies to the ’salt’ CLI.

27.3.2 Moving Pieces

When a Salt master starts up, a number of processes are started, all of which are called ’salt-master’ in a process-list but have various role categories.

Among those categories are:

- Publisher
- EventPublisher
- MWorker

27.3.3 Publisher

The Publisher process is responsible for sending commands over the designated transport to connected minions. The Publisher is bound to the following:

- TCP: port 4505
- IPC: publish_pull.ipc

Each salt minion establishes a connection to the master Publisher.

27.3.4 EventPublisher

The EventPublisher publishes master events out to any event listeners. It is bound to the following:

- IPC: master_event_pull.ipc
- IPC: master_event_pub.ipc

27.3.5 MWorker

Worker processes manage the back-end operations for the Salt Master.

The number of workers is equivalent to the number of 'worker_threads' specified in the master configuration and is always at least one.

Workers are bound to the following:

- IPC: workers.ipc

27.3.6 ReqServer

The Salt request server takes requests and distributes them to available MWorker processes for processing. It also receives replies back from minions.

The ReqServer is bound to the following:

  - TCP: 4506
  - IPC: workers.ipc

Each salt minion establishes a connection to the master ReqServer.

27.3.7 Job Flow

The Salt master works by always publishing commands to all connected minions and the minions decide if the command is meant for them by checking themselves against the command target.

The typical lifecycle of a salt job from the perspective of the master might be as follows:

1) A command is issued on the CLI. For example, 'salt my_minion test.version'.
2) The 'salt' command uses LocalClient to generate a request to the salt master by connecting to the ReqServer on TCP:4506 and issuing the job.
3) The salt-master ReqServer sees the request and passes it to an available MWorker over workers.ipc.
4) A worker picks up the request and handles it. First, it checks to ensure that the requested user has permissions to issue the command. Then, it sends the publish command to all connected minions. For the curious, this happens in ClearFuncs.publish().
5) The worker announces on the master event bus that it is about to publish a job to connected minions. This happens by placing the event on the master event bus (master_event_pull.ipc) where the EventPublisher picks it up and distributes it to all connected event listeners on master_event_pub.ipc.
6) The message to the minions is encrypted and sent to the Publisher via IPC on publish_pull.ipc.
7) Connected minions have a TCP session established with the Publisher on TCP port 4505 where they await commands. When the Publisher receives the job over publish_pull, it sends the jobs across the wire to the minions for processing.
8) After the minions receive the request, they decrypt it and perform any requested work, if they determine that they are targeted to do so.
9) When the minion is ready to respond, it publishes the result of its job back to the master by sending the encrypted result back to the master on TCP 4506 where it is again picked up by the ReqServer and forwarded to an available MWorker for processing. (Again, this happens by passing this message across workers.ipc to an available worker.)
10) When the MWorker receives the job it decrypts it and fires an event onto the master event bus (master_event_pull.ipc). (Again for the curious, this happens in AESFuncs._return().
11) The EventPublisher sees this event and re-publishes it on the bus to all connected listeners of the master event bus (on master_event_pub.ipc). This is where the LocalClient has been waiting, listening to the event bus for minion replies. It gathers the job and stores the result.
12) When all targeted minions have replied or the timeout has been exceeded, the salt client displays the results of the job to the user on the CLI.

27.4 Salt Minion

27.4.1 Overview

The salt-minion is a single process that sits on machines to be managed by Salt. It can either operate as a stand-alone daemon which accepts commands locally via 'salt-call' or it can connect back to a master and receive commands remotely.

When starting up, salt minions connect *back* to a master defined in the minion config file. They connect to two ports on the master:

- **TCP: 4505** This is the connection to the master Publisher. It is on this port that the minion receives jobs from the master.
- **TCP: 4506** This is the connection to the master ReqServer. It is on this port that the minion sends job results back to the master.

27.4.2 Event System

Similar to the master, a salt-minion has its own event system that operates over IPC by default. The minion event system operates on a push/pull system with IPC files at minion_event_<unique_id>_pub.ipc and minion_event_<unique_id>_pull.ipc.

The astute reader might ask why have an event bus at all with a single-process daemon. The answer is that the salt-minion may fork other processes as required to do the work without blocking the main salt-minion process and this necessitates a mechanism by which those processes can communicate with each other. Secondarily, this provides a bus by which any user with sufficient permissions can read or write to the bus as a common interface with the salt minion.

27.4.3 Minion Job Flow

When a salt minion starts up, it attempts to connect to the Publisher and the ReqServer on the salt master. It then attempts to authenticate and once the minion has successfully authenticated, it simply listens for jobs.

Jobs normally come either come from the 'salt-call' script run by a local user on the salt minion or they can come directly from a master.

The job flow on a minion, coming from the master via a 'salt' command is as follows:

1) A master publishes a job that is received by a minion as outlined by the master's job flow above. 2) The minion is polling its receive socket that's connected to the master Publisher (TCP 4505 on master). When it detects an incoming message, it picks it up from the socket and decrypts it. 3) A new minion process or thread is created and provided with the contents of the decrypted message. The _thread_return() method is provided with the contents of the received message. 4) The new minion thread is created. The _thread_return() function starts up and actually calls out to the requested function contained in the job. 5) The requested function runs and returns a result. [Still in thread.] 6) The result of the function that's run is encrypted and returned to the master's ReqServer (TCP 4506 on master). [Still in thread.] 7) Thread exits. Because the main thread was only blocked for the time that it took to initialize the worker thread, many other requests could have been received and processed during this time.

27.5 A Note on ClearFuncs vs. AESFuncs

A common source of confusion is determining when messages are passed in the clear and when they are passed using encryption. There are two rules governing this behaviour:

1) ClearFuncs is used for intra-master communication and during the initial authentication handshake between a minion and master during the key exchange. 2) AESFuncs is used everywhere else.

27.6 Contributing

There is a great need for contributions to Salt and patches are welcome! The goal here is to make contributions clear, make sure there is a trail for where the code has come from, and most importantly, to give credit where credit is due!

There are a number of ways to contribute to Salt development, including (but not limited to):

- filing well-written bug reports
- enhancing the documentation
- providing workarounds, patches, and other code without tests
- engaging in constructive discussion

- helping out in #salt on Freenode, the Community Slack, the salt-users mailing list, a SaltStack meetup, or Server Fault.
- telling others about problems you solved with Salt

If this or other Salt documentation is unclear, please review *Writing Salt Documentation*. PRs are welcome!

27.6.1 Quickstart

If you just want to get started before reading the rest of this guide, you can get the process started by running the following:

```
python3 -m pip install --user pre-commit
git clone --origin upstream https://github.com/saltstack/salt.git
cd salt
pre-commit install
```

While those commands are running, finish reading the rest of this guide.

27.6.2 Pre-commit

To reduce friction during the development process, SaltStack uses pre-commit. This tool adds pre-commit hooks to git to automate several processes that used to be manual. Rather than having to remember to run several different tools before you commit, you only have to run `git commit`, and you will be notified about style and lint issues before you ever open a PR.

27.6.3 Salt Coding Style

After the 3000 release, SaltStack is joining the ranks of projects in adopting the Black code formatter in order to ease the adoption of a unified code formatting style.

Where Black is silent, SaltStack has its own coding style guide that informs contributors on various style points. Please review the *Salt Coding Style* documentation for information about Salt's particular coding patterns.

Within the *Salt Coding Style* documentation, there is a section about running Salt's `.testing.pylintrc` file. SaltStack recommends running the `.testing.pylintrc` file on any files you are changing with your code contribution before submitting a pull request to Salt's repository.

If you've installed `pre-commit`, this will automatically happen before each commit. Otherwise, see the *Linting* documentation for more information.

27.6.4 Copyright Headers

Copyright headers are not needed for files in the Salt project. Files that have existing copyright headers should be considered legacy and not an example to follow.

27.6.5 Sending a GitHub pull request

Sending pull requests on GitHub is the preferred method for receiving contributions. The workflow advice below mirrors GitHub's own guide and is well worth reading.

1. Fork saltstack/salt on GitHub.
2. Make a local clone of your fork. (Skip this step if you followed the Quickstart)

```
git clone git@github.com:my-account/salt.git
cd salt
```

3. Add saltstack/salt as a git remote.

```
git remote add upstream https://github.com/saltstack/salt.git
```

If you followed the Quickstart, you'll add your own remote instead

```
git remote add my-account git@github.com:my-account/salt.git
```

4. Create a new branch in your clone.

Note: A branch should have one purpose. For example, "Fix bug X," or "Add feature Y". Multiple unrelated fixes and/or features should be isolated into separate branches.

```
git fetch upstream
git checkout -b fix-broken-thing upstream/master
```

5. Edit and commit changes to your branch.

```
vim path/to/file1 path/to/file2 tests/test_file1.py tests/test_file2.py
git diff
git add path/to/file1 path/to/file2
git commit
```

Write a short, descriptive commit title and a longer commit message if necessary. Use an imperative style for the title.

GOOD

```
Fix broken things in file1 and file2

Fixes #31337

We needed to make this change because the underlying dependency
changed. Now this uses the up-to-date API.

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
# On branch fix-broken-thing
# Changes to be committed:
#       modified:   path/to/file1
#       modified:   path/to/file2
```

BAD

```
Fixes broken things

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
# On branch fix-broken-thing
# Changes to be committed:
#       modified:   path/to/file1
#       modified:   path/to/file2
```

Taking a few moments to explain *why* you made a change will save time and effort in the future when others come to investigate a change. A clear explanation of why something changed can help future developers avoid introducing bugs, or breaking an edge case.

Note: If your change fixes a bug or implements a feature already filed in the issue tracker, be sure to reference the issue number in the commit message body.

If you get stuck, there are many introductory Git resources on http://help.github.com.

6. Push your locally-committed changes to your GitHub fork.

```
git push -u origin fix-broken-thing
```

or

```
git push -u origin add-cool-feature
```

Note: You may want to rebase before pushing to work out any potential conflicts:

```
git fetch upstream
git rebase upstream/master fix-broken-thing
git push -u origin fix-broken-thing
```

If you do rebase, and the push is rejected with a `(non-fast-forward)` comment, then run `git status`. You will likely see a message about the branches diverging:

```
On branch fix-broken-thing
Your branch and 'origin/fix-broken-thing' have diverged,
and have 1 and 2 different commits each, respectively.
  (use "git pull" to merge the remote branch into yours)
nothing to commit, working tree clean
```

Do **NOT** perform a `git pull` or `git merge` here. Instead, add `--force-with-lease` to the end of the `git push` command to get the changes pushed to your fork. Pulling or merging, while they will resolve the non-fast-forward issue, will likely add extra commits to the pull request which were not part of your changes.

---

7. Find the branch on your GitHub salt fork.

   https://github.com/my-account/salt/branches/fix-broken-thing

8. Open a new pull request.

   Click on `Pull Request` on the right near the top of the page,

   https://github.com/my-account/salt/pull/new/fix-broken-thing

   1. Choose `master` as the base Salt branch.
   2. Review that the proposed changes are what you expect.
   3. Write a descriptive comment. If you added good information to your git commit message, they will already be present here. Include links to related issues (e.g. 'Fixes #31337.') in the comment field.
   4. Click `Create pull request`.

9. Salt project members will review your pull request and automated tests will run on it.

   If you recognize any test failures as being related to your proposed changes or if a reviewer asks for modifications:

1. Make the new changes in your local clone on the same local branch.
2. Push the branch to GitHub again using the same commands as before.
3. New and updated commits will be added to the pull request automatically.
4. Feel free to add a comment to the discussion.

Note: Jenkins

Pull request against saltstack/salt are automatically tested on a variety of operating systems and configurations. On average these tests take a couple of hours. Depending on your GitHub notification settings you may also receive an email message about the test results.

Test progress and results can be found at http://jenkins.saltstack.com/.

27.6.6 Salt's Branch Topology

Salt will only have one active branch - `master`. This will include bug fixes, features and CVE "Common Vulnerabilities and Exposures".

The release will be cut from the master when the time comes for a new release, which should be every 3 to 4 months.

To be able to merge code:

1. The code must have a well-written test. Note that you are only expected to write tests for what you did, not the whole modules or function.
2. All tests must pass.

The SaltStack employee that reviews your pull request might request changes or deny the pull request for various reasons.

Salt uses a typical branch strategy - `master` is the next expected release. Code should only make it to `master` once it's production ready. This means that typical changes (fixes, features) should have accompanying tests.

Closing GitHub issues from commits

SaltStack encourages using the magic keywords to close a GitHub issue. These should appear in the commit message text directly.

27.6.7 Release Naming Convention

A new convention will start when Salt releases Salt 3000. Every new release name will increment by one 'Salt last_release_number + 1'.

This naming convention is very different from past releases, which was 'YYYY.MM.PATCH'.

27.6.8 Handling CVE

If a CVE is discovered, Salt will create a new release that **only** contains the tests and patch for the CVE. This method should improve the upgrade process by reducing the chances of breaking something.

27.6.9 Backporting Pull Requests

On rare occasions, a serious bug will be found in the middle of a release cycle. These bugs will require a point release. Contributors should still submit fixes directly to `master`, but they should also call attention to the fact that it addresses a critical issue and will need to be back-ported.

27.6.10 Keeping Salt Forks in Sync

Salt advances quickly. It is therefore critical to pull upstream changes from upstream into your fork on a regular basis. Nothing is worse than putting hard work into a pull request only to see bunches of merge conflicts because it has diverged too far from upstream.

See also:

GitHub Fork a Repo Guide

The following assumes `origin` is the name of your fork and `upstream` is the name of the main saltstack/salt repository.

1. View existing remotes.

   ```
   git remote -v
   ```

2. Add the `upstream` remote.

   ```
   # For ssh github
   git remote add upstream git@github.com:saltstack/salt.git

   # For https github
   git remote add upstream https://github.com/saltstack/salt.git
   ```

3. Pull upstream changes into your clone.

   ```
   git fetch upstream
   ```

4. Update your copy of the `master` branch.

   ```
   git checkout master
   git merge --ff-only upstream/master
   ```

   If Git complains that a fast-forward merge is not possible, you have local commits.

   - Run `git pull --rebase origin master` to rebase your changes on top of the upstream changes.
   - Or, run `git branch <branch-name>` to create a new branch with your commits. You will then need to reset your `master` branch before updating it with the changes from upstream.

   If Git complains that local files will be overwritten, you have changes to files in your working directory. Run `git status` to see the files in question.

5. Update your fork.

   ```
   git push origin master
   ```

6. Repeat the previous two steps for any other branches you work with, such as the current release branch.

27.6.11 Posting patches to the mailing list

Patches will also be accepted by email. Format patches using git format-patch and send them to the salt-users mailing list. The contributor will then get credit for the patch, and the Salt community will have an archive of the patch and a place for discussion.

27.6.12 Issue and Pull Request Labeling System

SaltStack uses several labeling schemes to help facilitate code contributions and bug resolution. See the *Labels and Milestones* documentation for more information.

27.6.13 Mentionbot

SaltStack runs a mention-bot which notifies contributors who might be able to help review incoming pull-requests based on their past contribution to files which are being changed.

If you do not wish to receive these notifications, please add your GitHub handle to the blacklist line in the `.mention-bot` file located in the root of the Salt repository.

27.6.14 Bootstrap Script Changes

Salt's Bootstrap Script, known as bootstrap-salt.sh in the Salt repo, has it's own repository, contributing guidelines, and release cadence.

All changes to the Bootstrap Script should be made to salt-bootstrap repo. Any pull requests made to the bootstrap-salt.sh file in the Salt repository will be automatically overwritten upon the next stable release of the Bootstrap Script.

For more information on the release process or how to contribute to the Bootstrap Script, see the Bootstrap Script's Contributing Guidelines.

27.7 Deprecating Code

Salt should remain backwards compatible, though sometimes, this backwards compatibility needs to be broken because a specific feature and/or solution is no longer necessary or required. At first one might think, let me change this code, it seems that it's not used anywhere else so it should be safe to remove. Then, once there's a new release, users complain about functionality which was removed and they where using it, etc. This should, at all costs, be avoided, and, in these cases, *that* specific code should be deprecated.

In order to give users enough time to migrate from the old code behavior to the new behavior, the deprecation time frame should be carefully determined based on the significance and complexity of the changes required by the user.

Salt feature releases are based on the Periodic Table. Any new features going into the develop branch will be named after the next element in the Periodic Table. For example, Beryllium was the feature release name of the develop branch before the 2015.8 branch was tagged. At that point in time, any new features going into the develop branch after 2015.8 was branched were part of the Boron feature release.

A deprecation warning should be in place for at least two major releases before the deprecated code and its accompanying deprecation warning are removed. More time should be given for more complex changes. For example, if the current release under development is `Sodium`, the deprecated code and associated warnings should remain in place and warn for at least `Aluminum`.

To help in this deprecation task, salt provides `salt.utils.versions.warn_until`. The idea behind this helper function is to show the deprecation warning to the user until salt reaches the provided version. Once that provided version is equaled `salt.utils.versions.warn_until` will raise a `RuntimeError` making salt stop its execution. This stoppage is unpleasant and will remind the developer that the deprecation limit has been reached and that the code can then be safely removed.

Consider the following example:

```python
def some_function(bar=False, foo=None):
    if foo is not None:
        salt.utils.versions.warn_until(
            'Aluminum',
            'The \'foo\' argument has been deprecated and its '
            'functionality removed, as such, its usage is no longer '
            'required.'
        )
```

Development begins on the `Aluminum` release when the `Magnesium` branch is forked from the develop branch. Once this occurs, all uses of the `warn_until` function targeting `Aluminum`, along with the code they are warning about should be removed from the code.

27.8 Installing Salt for development

Clone the repository using:

```
git clone https://github.com/saltstack/salt
```

Note: tags

Just cloning the repository is enough to work with Salt and make contributions. However, fetching additional tags from git is required to have Salt report the correct version for itself. To do this, first add the git repository as an upstream source:

```
git remote add upstream https://github.com/saltstack/salt
```

Fetching tags is done with the git ’fetch’ utility:

```
git fetch --tags upstream
```

Create a new virtualenv:

```
virtualenv /path/to/your/virtualenv
```

Avoid making your *virtualenv path too long*.

On Arch Linux, where Python 3 is the default installation of Python, use the `virtualenv2` command instead of `virtualenv`.

On Gentoo you must use `--system-site-packages` to enable pkg and portage_config functionality

---

Note: Using system Python modules in the virtualenv

To use already-installed python modules in virtualenv (instead of having pip download and compile new ones), run `virtualenv --system-site-packages` Using this method eliminates the requirement to install the salt dependencies again, although it does assume that the listed modules are all installed in the system PYTHONPATH at the time of virtualenv creation.

Note: Python development package

Be sure to install python devel package in order to install required Python modules. In Debian/Ubuntu run `sudo apt-get install -y python-dev`. In RedHat based system install `python-devel`

Activate the virtualenv:

```
source /path/to/your/virtualenv/bin/activate
```

Install Salt (and dependencies) into the virtualenv:

```
pip install pyzmq PyYAML pycrypto msgpack-python jinja2 psutil futures tornado
pip install -e ./salt   # the path to the salt git clone from above
```

Note: Installing psutil

Python header files are required to build this module, otherwise the pip install will fail. If your distribution separates binaries and headers into separate packages, make sure that you have the headers installed. In most Linux distributions which split the headers into their own package, this can be done by installing the `python-dev` or `python-devel` package. For other platforms, the package will likely be similarly named.

Note: Installing dependencies on macOS.

You can install needed dependencies on macOS using homebrew or macports. See *macOS Installation*

Warning: Installing on RedHat-based Distros

If installing from pip (or from source using `setup.py install`), be advised that the `yum-utils` package is needed for Salt to manage packages on RedHat-based systems.

27.8.1 Running a self-contained development version

During development it is easiest to be able to run the Salt master and minion that are installed in the virtualenv you created above, and also to have all the configuration, log, and cache files contained in the virtualenv as well.

The `/path/to/your/virtualenv` referenced multiple times below is also available in the variable `$VIRTUAL_ENV` once the virtual environment is activated.

Copy the master and minion config files into your virtualenv:

```
mkdir -p /path/to/your/virtualenv/etc/salt/pki/{master,minion}
cp ./salt/conf/master ./salt/conf/minion /path/to/your/virtualenv/etc/salt/
```

Edit the master config file:

1. Uncomment and change the `user: root` value to your own user.
2. Uncomment and change the `root_dir: /` value to point to `/path/to/your/virtualenv`.

3. Uncomment and change the `pki_dir: /etc/salt/pki/master` value to point to `/path/to/your/virtualenv/etc/salt/pki/master`
4. If you are running version 0.11.1 or older, uncomment, and change the `pidfile: /var/run/salt-master.pid` value to point to `/path/to/your/virtualenv/salt-master.pid`.
5. If you are also running a non-development version of Salt you will have to change the `publish_port` and `ret_port` values as well.

Edit the minion config file:

1. Repeat the edits you made in the master config for the `user` and `root_dir` values as well as any port changes.
2. Uncomment and change the `pki_dir: /etc/salt/pki/minion` value to point to `/path/to/your/virtualenv/etc/salt/pki/minion`
3. If you are running version 0.11.1 or older, uncomment, and change the `pidfile: /var/run/salt-minion.pid` value to point to `/path/to/your/virtualenv/salt-minion.pid`.
4. Uncomment and change the `master: salt` value to point at `localhost`.
5. Uncomment and change the `id:` value to something descriptive like "saltdev". This isn't strictly necessary but it will serve as a reminder of which Salt installation you are working with.
6. If you changed the `ret_port` value in the master config because you are also running a non-development version of Salt, then you will have to change the `master_port` value in the minion config to match.

Note: Using *salt-call* with a *Standalone Minion*

If you plan to run *salt-call* with this self-contained development environment in a masterless setup, you should invoke *salt-call* with `-c /path/to/your/virtualenv/etc/salt` so that salt can find the minion config file. Without the `-c` option, Salt finds its config files in */etc/salt*.

Start the master and minion, accept the minion's key, and verify your local Salt installation is working:

```
cd /path/to/your/virtualenv
salt-master -c ./etc/salt -d
salt-minion -c ./etc/salt -d
salt-key -c ./etc/salt -L
salt-key -c ./etc/salt -A
salt -c ./etc/salt '*' test.version
```

Running the master and minion in debug mode can be helpful when developing. To do this, add `-l debug` to the calls to `salt-master` and `salt-minion`. If you would like to log to the console instead of to the log file, remove the `-d`.

Note: Too long socket path?

Once the minion starts, you may see an error like the following:

```
zmq.core.error.ZMQError: ipc path "/path/to/your/virtualenv/
var/run/salt/minion/minion_event_7824dcbcfd7a8f6755939af70b96249f_pub.ipc"
is longer than 107 characters (sizeof(sockaddr_un.sun_path)).
```

This means that the path to the socket the minion is using is too long. This is a system limitation, so the only workaround is to reduce the length of this path. This can be done in a couple different ways:

1. Create your virtualenv in a path that is short enough.

2. Edit the *`sock_dir`* minion config variable and reduce its length. Remember that this path is relative to the value you set in *`root_dir`*.

`NOTE:` The socket path is limited to 107 characters on Solaris and Linux, and 103 characters on BSD-based systems.

Note: File descriptor limits

Ensure that the system open file limit is raised to at least 2047:

```
# check your current limit
ulimit -n

# raise the limit. persists only until reboot
# use 'limit descriptors 2047' for c-shell
ulimit -n 2047
```

To set file descriptors on macOS, refer to the *macOS Installation* instructions.

Changing Default Paths

Instead of updating your configuration files to point to the new root directory and having to pass the new configuration directory path to all of Salt's CLI tools, you can explicitly tweak the default system paths that Salt expects:

```
GENERATE_SALT_SYSPATHS=1 pip install --global-option='--salt-root-dir=/path/to/your/
↪virtualenv/' \
    -e ./salt   # the path to the salt git clone from above
```

You can now call all of Salt's CLI tools without explicitly passing the configuration directory.

Additional Options

If you want to distribute your virtualenv, you probably don't want to include Salt's clone `.git/` directory, and, without it, Salt won't report the accurate version. You can tell `setup.py` to generate the hardcoded version information which is distributable:

```
GENERATE_SALT_SYSPATHS=1 WRITE_SALT_VERSION=1 pip install --global-option='--salt-
↪root-dir=/path/to/your/virtualenv/' \
    -e ./salt   # the path to the salt git clone from above
```

Instead of passing those two environmental variables, you can just pass a single one which will trigger the other two:

```
MIMIC_SALT_INSTALL=1 pip install --global-option='--salt-root-dir=/path/to/your/
↪virtualenv/' \
    -e ./salt   # the path to the salt git clone from above
```

This last one will grant you an editable salt installation with hardcoded system paths and version information.

27.8.2 Installing Salt from the Python Package Index

If you are installing using `easy_install`, you will need to define a **USE_SETUPTOOLS** environment variable, otherwise dependencies will not be installed:

```
USE_SETUPTOOLS=1 easy_install salt
```

27.8.3 Editing and previewing the documentation

You need `sphinx-build` command to build the docs. In Debian/Ubuntu this is provided in the `python-sphinx` package. Sphinx can also be installed to a virtualenv using pip:

```
pip install Sphinx==1.3.1
```

Change to salt documentation directory, then:

```
cd doc; make html
```

- This will build the HTML docs. Run `make` without any arguments to see the available make targets, which include **html**, **man**, and **text**.
- The docs then are built within the **docs/_build/** folder. To update the docs after making changes, run `make` again.
- The docs use reStructuredText for markup. See a live demo at http://rst.ninjs.org/.
- The help information on each module or state is culled from the python code that runs for that piece. Find them in `salt/modules/` or `salt/states/`.
- To build the docs on Arch Linux, the **python2-sphinx** package is required. Additionally, it is necessary to tell **make** where to find the proper **sphinx-build** binary, like so:

```
make SPHINXBUILD=sphinx-build2 html
```

- To build the docs on RHEL/CentOS 6, the **python-sphinx10** package must be installed from EPEL, and the following make command must be used:

```
make SPHINXBUILD=sphinx-build html
```

Once you've updated the documentation, you can run the following command to launch a simple Python HTTP server to see your changes:

```
cd _build/html; python -m SimpleHTTPServer
```

27.8.4 Running unit and integration tests

Run the test suite with following command:

```
./setup.py test
```

See *here* for more information regarding the test suite.

27.8.5 Issue and Pull Request Labeling System

SaltStack uses several labeling schemes to help facilitate code contributions and bug resolution. See the *Labels and Milestones* documentation for more information.

27.9 GitHub Labels and Milestones

SaltStack uses several label categories, as well as milestones, to triage incoming issues and pull requests in the GitHub issue tracker. Labels are used to sort issues by type, priority, severity, status, functional area, functional group, and targeted release and pull requests by status, functional area, functional group, type of change, and test status. Milestones are used to indicate whether an issue is fully triaged or is scheduled to be fixed by SaltStack in an upcoming sprint.

27.9.1 Milestones

All issues are assigned to a milestone, whereas pull requests are almost never assigned to a milestone as the mean lifetime of pull requests is short enough that there is no need to track them temporally.

SaltStack uses milestones to indicate which issues are blocked on submitter or upstream actions, are approved, or are scheduled to be fixed or implemented in an upcoming sprint. If an issue is not attached to a sprint milestone, you are welcome to work on it at your own desire and convenience. If it is attached to a sprint milestone and you have already begun working on it or have a solution in mind or have other ideas related to the issue, you are encouraged to coordinate with the assignee via the GitHub issue tracker to create the best possible solution or implementation.

- `Approved` - The issue has been validated and has all necessary information.
- `Blocked` - The issue is waiting on actions by parties outside of SaltStack, such as receiving more information from the submitter or resolution of an upstream issue. This milestone is usually applied in conjunction with the labels `Info Needed`, `Question`, `Expected Behavior`, `Won't Fix For Now`, or `Upstream Bug`.

27.9.2 Labels

Labels are used to sort and describe issues and pull requests. Some labels are usually reserved for one or the other, though most labels may be applied to both.

New issues will receive at least one label and a milestone, and new pull requests will receive at least one label. Except for the *functional area* and *functional group* label categories, issues will generally receive only up to one label per category.

Type

Issues are categorized into one of several types. Type labels are almost never used for pull requests. GitHub treats pull requests like issues in many ways, so a pull request could be considered an issue with an implicit `Pull Request` type label applied.

- `Feature` - The issue is a request for new functionality including changes, enhancements, refactors, etc.
- `Bug` - The issue documents broken, incorrect, or confusing behavior. This label is always accompanied by a *severity label*.
- `Duplicate` - The issue is a duplicate of another feature request or bug report.
- `Upstream Bug` - The issue is a result of an upstream issue.
- `Question` - The issue is more of a question than a request for new features or a report of broken features, but can sometimes lead to further discussion or changes of confusing or incongruous behavior or documentation.
- `Expected Behavior` - The issue is a bug report of intended functionality.

Priority

An issue's priority is relative to its *functional area*. If a bug report, for example, about `gitfs` indicates that all users of `gitfs` will encounter this bug, then a `P1` label will be applied, even though users who are not using `gitfs` will not encounter the bug. If a feature is requested by many users, it may be given a high priority.

- `P1` - The issue will be seen by all users.
- `P2` - The issue will be seen by most users.
- `P3` - The issue will be seen by about half of users.
- `P4` - The issue will not be seen by most users. Usually the issue is a very specific use case or corner case.

Severity

Severity labels are almost always only applied to issues labeled `Bug`.

- `Blocker` - The issue is blocking an impending release.
- `Critical` - The issue causes data loss, crashes or hangs salt processes, makes the system unresponsive, etc.
- `High Severity` - The issue reports incorrect functionality, bad functionality, a confusing user experience, etc.
- `Medium Severity` - The issue reports cosmetic items, formatting, spelling, colors, etc.

Functional Area

Many major components of Salt have corresponding GitHub labels. These labels are applied to all issues and pull requests as is reasonably appropriate. They are useful in organizing issues and pull requests according to the source code relevant to issues or the source code changed by pull requests.

- `Execution Module`
- `File Servers`
- `Grains`
- `Multi-Master`
- `Packaging` Related to packaging of Salt, not Salt's support for package management.
- `Pillar`
- `RAET`
- `Returners`
- `Runners`
- `SPM`
- `Salt-API`
- `Salt-Cloud`
- `Salt-SSH`
- `Salt-Syndic`
- `State Module`
- `Tests`

- `Transport`
- `Windows`
- `ZMQ`

Functional Group

These labels sort issues and pull requests according to the internal SaltStack engineering teams.

- `Core` - The issue or pull request relates to code that is central or existential to Salt itself.
- `Platform` - The issue or pull request relates to support and integration with various platforms like traditional operating systems as well as containers, platform-based utilities like filesystems, command schedulers, etc., and system-based applications like webservers, databases, etc.
- `RIoT` - The issue or pull request relates to support and integration with various abstract systems like cloud providers, hypervisors, API-based services, etc.
- `Console` - The issue or pull request relates to the SaltStack enterprise console.
- `Documentation` - The issue or pull request relates to documentation.

Status

Status labels are used to define and track the state of issues and pull requests. Not all potential statuses correspond to a label, but some statuses are common enough that labels have been created for them. If an issue has not been moved beyond the `Blocked` milestone, it is very likely that it will only have a status label.

- `Bugfix - back-port` The pull request needs to be back-ported to an older release branch. This is done by *recreating the pull request* against that branch. Once the back-port is completed, this label is replaced with a `Bugfix - [Done] back-ported` label. Normally, new features should go into the develop and bug fixes into the oldest supported release branch, see *here*.
- `Bugfix - [Done] back-ported` - The pull request has been back-ported to an older branch.
- `Cannot Reproduce` - The issue is a bug and has been reviewed by a SaltStack engineer, but it cannot be replicated with the provided information and context. Those involved with the bug will need to work through additional ideas until the bug can be isolated and verified.
- `Confirmed` - The issue is a bug and has been confirmed by a SaltStack engineer, who often documents a minimal working example that reproduces the bug.
- `Fixed Pending Verification` - The issue is a bug and has been fixed by one or more pull requests, which should link to the issue. Closure of the issue is contingent upon confirmation of resolution from the submitter. If the submitter reports a negative confirmation, this label is removed. If no response is given after a few weeks, then the issue will be assumed fixed and closed.
- `Info Needed` - The issue needs more information before it can be verified and resolved. For a feature request this may include a description of the use cases. Almost all bug reports need to include at least the versions of salt and its dependencies, the system type and version, commands used, debug logs, error messages, and relevant configs.
- `Pending Changes` - The pull request needs additional changes before it can be merged.
- `Pending Discussion` - The issue or pull request needs more discussion before it can be closed or merged. The status of the issue or pull request is not clear or apparent enough for definite action to be taken, or additional input from SaltStack, the submitter, or another party has been requested.

If the issue is not a pull request, once the discussion has arrived at a cogent conclusion, this label will be removed and the issue will be accepted. If it is a pull request, the results of the discussion may require additional changes and thus, a `Pending Changes` label.

- `Won't Fix for Now` - The issue is legitimate, but it is not something the SaltStack team is currently able or willing to fix or implement. Issues having this label may be revisited in the future.

Type of Change

Every pull request should receive a change label. These labels measure the quantity of change as well as the significance of the change. The amount of change and the importance of the code area changed are considered, but often the depth of secondary code review required and the potential repercussions of the change may also advise the label choice.

Core code areas include: state compiler, crypto engine, master and minion and syndic daemons, transport, pillar rendering, loader, transport layer, event system, salt.utils, client, cli, logging, netapi, runner engine, templating engine, top file compilation, file client, file server, mine, salt-ssh, test runner, etc.

Non-core code usually constitutes the specific set of plugins for each of the several plugin layers of Salt: execution modules, states, runners, returners, clouds, etc.

- `Minor Change`
  - Less than 64 lines changed, or
  - Less than 8 core lines changed
- `Medium Change`
  - Less than 256 lines changed, or
  - Less than 64 core lines changed
- `Master Change`
  - More than 256 lines changed, or
  - More than 64 core lines changed
- `Expert Change`
  - Needs specialized, in-depth review

Test Status

These labels relate to the status of the automated tests that run on pull requests. If the tests on a pull request fail and are not overridden by one of these labels, the pull request submitter needs to update the code and/or tests so that the tests pass and the pull request can be merged.

- `Lint` - The pull request has passed all tests except for the code lint checker.
- `Tests Passed` - The pull request has passed all tests even though some test results are negative. Sometimes the automated testing infrastructure will encounter internal errors unrelated to the code change in the pull request that cause test runs to fail. These errors can be caused by cloud host and network issues and also Jenkins issues like erroneously accumulating workspace artifacts, resource exhaustion, and bugs that arise from long running Jenkins processes.

Other

These labels indicate miscellaneous issue types or statuses that are common or important enough to be tracked and sorted with labels.

- `Awesome` - The pull request implements an especially well crafted solution, or a very difficult but necessary change.
- `Help Wanted` - The issue appears to have a simple solution. Issues having this label should be a good starting place for new contributors to Salt.
- `Needs Testcase` - The issue or pull request relates to a feature that needs test coverage. The pull request containing the tests should reference the issue or pull request having this label, whereupon the label should be removed.
- `Regression` - The issue is a bug that breaks functionality known to work in previous releases.
- `Story` - The issue is used by a SaltStack engineer to track progress on multiple related issues in a single place.
- `Stretch` - The issue is an optional goal for the current sprint but may not be delivered.
- `ZD` - The issue is related to a Zendesk customer support ticket.
- `<Release>` - The issue is scheduled to be implemented by `<Release>`. See here for a discussion of Salt's release codenames.

27.10 Logging Internals

TODO

27.11 Package Providers

This page contains guidelines for writing package providers.

27.11.1 Package Functions

One of the most important features of Salt is package management. There is no shortage of package managers, so in the interest of providing a consistent experience in *pkg* states, there are certain functions that should be present in a package provider. Note that these are subject to change as new features are added or existing features are enhanced.

list_pkgs

This function should declare an empty dict, and then add packages to it by calling *`pkg_resource.add_pkg`*, like so:

```
__salt__['pkg_resource.add_pkg'](ret, name, version)
```

The last thing that should be done before returning is to execute *`pkg_resource.sort_pkglist`*. This function does not presently do anything to the return dict, but will be used in future versions of Salt.

```
__salt__['pkg_resource.sort_pkglist'](ret)
```

`list_pkgs` returns a dictionary of installed packages, with the keys being the package names and the values being the version installed. Example return data:

```
{'foo': '1.2.3-4',
 'bar': '5.6.7-8'}
```

latest_version

Accepts an arbitrary number of arguments. Each argument is a package name. The return value for a package will be an empty string if the package is not found or if the package is up-to-date. The only case in which a non-empty string is returned is if the package is available for new installation (i.e. not already installed) or if there is an upgrade available.

If only one argument was passed, this function return a string, otherwise a dict of name/version pairs is returned.

This function must also accept `**kwargs`, in order to receive the `fromrepo` and `repo` keyword arguments from pkg states. Where supported, these arguments should be used to find the install/upgrade candidate in the specified repository. The `fromrepo` kwarg takes precedence over `repo`, so if both of those kwargs are present, the repository specified in `fromrepo` should be used. However, if `repo` is used instead of `fromrepo`, it should still work, to preserve backwards compatibility with older versions of Salt.

version

Like `latest_version`, accepts an arbitrary number of arguments and returns a string if a single package name was passed, or a dict of name/value pairs if more than one was passed. The only difference is that the return values are the currently-installed versions of whatever packages are passed. If the package is not installed, an empty string is returned for that package.

upgrade_available

Deprecated and destined to be removed. For now, should just do the following:

```
return __salt__['pkg.latest_version'](name) != ''
```

install

The following arguments are required and should default to `None`:

1. name (for single-package pkg states)
2. pkgs (for multiple-package pkg states)
3. sources (for binary package file installation)

The first thing that this function should do is call *`pkg_resource.parse_targets`* (see below). This function will convert the SLS input into a more easily parsed data structure. *`pkg_resource.parse_targets`* may need to be modified to support your new package provider, as it does things like parsing package metadata which cannot be done for every package management system.

```
pkg_params, pkg_type = __salt__['pkg_resource.parse_targets'](name,
                                                               pkgs,
                                                               sources)
```

Two values will be returned to the **install** function. The first of them will be a dictionary. The keys of this dictionary will be package names, though the values will differ depending on what kind of installation is being done:

- If **name** was provided (and **pkgs** was not), then there will be a single key in the dictionary, and its value will be `None`. Once the data has been returned, if the **version** keyword argument was provided, then it should replace the `None` value in the dictionary.
- If **pkgs** was provided, then **name** is ignored, and the dictionary will contain one entry for each package in the **pkgs** list. The values in the dictionary will be `None` if a version was not specified for the package, and the desired version if specified. See the **Multiple Package Installation Options** section of the *`pkg.installed`* state for more info.
- If **sources** was provided, then **name** is ignored, and the dictionary values will be the path/URI for the package.

The second return value will be a string with two possible values: `repository` or `file`. The **install** function can use this value (if necessary) to build the proper command to install the targeted package(s).

Both before and after the installing the target(s), you should run **list_pkgs** to obtain a list of the installed packages. You should then return the output of `salt.utils.data.compare_dicts()`:

```
return salt.utils.data.compare_dicts(old, new)
```

remove

Removes the passed package and return a list of the packages removed.

27.11.2 Package Repo Functions

There are some functions provided by `pkg` which are specific to package repositories, and not to packages themselves. When writing modules for new package managers, these functions should be made available as stated below, in order to provide compatibility with the `pkgrepo` state.

All repo functions should accept a basedir option, which defines which directory repository configuration should be found in. The default for this is dictated by the repo manager that is being used, and rarely needs to be changed.

```
basedir = '/etc/yum.repos.d'
__salt__['pkg.list_repos'](basedir)
```

list_repos

Lists the repositories that are currently configured on this system.

```
__salt__['pkg.list_repos']()
```

Returns a dictionary, in the following format:

```
{'reponame': 'config_key_1': 'config value 1',
             'config_key_2': 'config value 2',
             'config_key_3': ['list item 1 (when appropriate)',
                              'list item 2 (when appropriate)]}
```

get_repo

Displays all local configuration for a specific repository.

```
__salt__['pkg.get_repo'](repo='myrepo')
```

The information is formatted in much the same way as list_repos, but is specific to only one repo.

```
{'config_key_1': 'config value 1',
 'config_key_2': 'config value 2',
 'config_key_3': ['list item 1 (when appropriate)',
                  'list item 2 (when appropriate)]}
```

del_repo

Removes the local configuration for a specific repository. Requires a *repo* argument, which must match the locally configured name. This function returns a string, which informs the user as to whether or not the operation was a success.

```
__salt__['pkg.del_repo'](repo='myrepo')
```

mod_repo

Modify the local configuration for one or more option for a configured repo. This is also the way to create new repository configuration on the local system; if a repo is specified which does not yet exist, it will be created.

The options specified for this function are specific to the system; please refer to the documentation for your specific repo manager for specifics.

```
__salt__['pkg.mod_repo'](repo='myrepo', url='http://myurl.com/repo')
```

27.11.3 Low-Package Functions

In general, the standard package functions as describes above will meet your needs. These functions use the system's native repo manager (for instance, yum or the apt tools). In most cases, the repo manager is actually separate from the package manager. For instance, yum is usually a front-end for rpm, and apt is usually a front-end for dpkg. When possible, the package functions that use those package managers directly should do so through the low package functions.

It is normal and sane for `pkg` to make calls to `lowpkgs`, but `lowpkg` must never make calls to `pkg`. This is affects functions which are required by both `pkg` and `lowpkg`, but the technique in `pkg` is more performant than what is available to `lowpkg`. When this is the case, the `lowpkg` function that requires that technique must still use the `lowpkg` version.

list_pkgs

Returns a dict of packages installed, including the package name and version. Can accept a list of packages; if none are specified, then all installed packages will be listed.

```
installed = __salt__['lowpkg.list_pkgs']('foo', 'bar')
```

Example output:

```
{'foo': '1.2.3-4',
 'bar': '5.6.7-8'}
```

verify

Many (but not all) package management systems provide a way to verify that the files installed by the package manager have or have not changed. This function accepts a list of packages; if none are specified, all packages will be included.

```
installed = __salt__['lowpkg.verify']('httpd')
```

Example output:

```
{'/etc/httpd/conf/httpd.conf': {'mismatch': ['size', 'md5sum', 'mtime'],
                                'type': 'config'}}
```

file_list

Lists all of the files installed by all packages specified. If not packages are specified, then all files for all known packages are returned.

```
installed = __salt__['lowpkg.file_list']('httpd', 'apache')
```

This function does not return which files belong to which packages; all files are returned as one giant list (hence the *file_list* function name. However, This information is still returned inside of a dict, so that it can provide any errors to the user in a sane manner.

```
{'errors': ['package apache is not installed'],
  'files': ['/etc/httpd',
            '/etc/httpd/conf',
            '/etc/httpd/conf.d',
            '...SNIP...']}
```

file_dict

Lists all of the files installed by all packages specified. If not packages are specified, then all files for all known packages are returned.

```
installed = __salt__['lowpkg.file_dict']('httpd', 'apache', 'kernel')
```

Unlike *file_list*, this function will break down which files belong to which packages. It will also return errors in the same manner as *file_list*.

```
{'errors': ['package apache is not installed'],
 'packages': {'httpd': ['/etc/httpd',
                        '/etc/httpd/conf',
                        '...SNIP...'],
              'kernel': ['/boot/.vmlinuz-2.6.32-279.el6.x86_64.hmac',
                         '/boot/System.map-2.6.32-279.el6.x86_64',
                         '...SNIP...']}}
```

27.12 Pull Requests

Salt is a large software project with many developers working together. We encourage all Salt users to contribute new features, bug fixes and documentation fixes. For those who haven't contributed to a large software project before we encourage you to consider the following questions when preparing a pull request.

This isn't an exhaustive list and these aren't necessarily hard and fast rules, but these are things we consider when reviewing a pull request.

- Does this change work on all platforms? In cases where it does not, is an appropriate and easy-to-understand reason presented to the user? Is it documented as-such? Have we thought about all the possible ways this code might be used and accounted as best we can for them?
- Will this code work on versions of all Python we support? Will it work on future versions?
- Are Python reserved keywords used? Are variables named in a way that will make it easy for the next person to understand what's going on?
- Does this code present a security risk in any way? What is the worst possible thing that an attacker could do with this code? If dangerous cases are possible, is it appropriate to document them? If so, has this been done? Would this change pass muster with a professional security audit? Is it obvious to a person using this code what the risks are?
- Is it readable? Does it conform to our style guide? Is the code documented such that the next person who comes along will be able to read and understand it? Most especially, are edge-cases documented to avoid regressions? Will it be immediately evident to the next person who comes along why this change was made?
- If appropriate, has the person who wrote the code which is being modified been notified and included in the process?
- What are the performance implications of this change? Is there a more efficient way to structure the logic and if so, does making the change balance itself against readability in a sensible way? Do the performance characteristics of the code change based on the way it is being invoked (i.e., through an API or various command-line tools.) Will it be easy to profile this change if it might be a problem?
- Are caveats considered and documented in the change?
- Will the code scale? More critically, will it scale in *both* directions? Salt runs in data-centers and on Raspberry Pi installations in the Sahara. It needs to work on big servers and tiny devices.
- Is appropriate documentation written both in public-facing docs and in-line? How will the user know how to use this? What will they do if it doesn't work as expected? Is this something a new user will understand? Can a user know all they need to about this functionality by reading the public docs?
- Is this a change in behavior? If so, is it in the appropriate branch? Are deprecation warnings necessary? Have those changes been fully documented? Have we fully thought through what implications a change in behavior might have?
- How has the code been tested? If appropriate are there automated tests which cover this? Is it likely to regress? If so, how has the potential of that regression been mitigated? What is the plan for ensuring that this code works going forward?
- If it's asynchronous code, what is the potential for a race condition?
- Is this code an original work? If it's borrowed from another project or found online are the appropriate licensing/attribution considerations handled?
- Is the reason for the change fully explained in the PR? If not for review, this is necessary so that somebody in the future can go back and figure out why it was necessary.
- Is the intended behavior of the change clear? How will that behavior be known to future contributors and to users?
- Does this code handle errors in a reasonable way? Have we gone back through the stack as much as possible to make sure that an error cannot be raised that we do not account for? Are errors tested for as well as proper functionality?

- If the code relies on external libraries, do we properly handle old versions of them? Do we require a specific version and if so is this version check implemented? Is the library available on the same platforms that module in question claims to support? If the code was written and tested against a particular library, have we documented that fact?
- Can this code freeze/hang/crash a running daemon? Can it stall a state run? Are there infinite loops? Are appropriate timeouts implemented?
- Is the function interface well documented? If argument types can not be inferred by introspection, are they documented?
- Are resources such as file-handles cleaned-up after they are used?
- Is it possible that a reference-cycle exists between objects that will leak memory?
- Has the code been linted and does it pass all tests?
- Does the change fully address the problem or is it limited to a small surface area? By this, I mean that it should be clear that the submitter has looked for other cases in the function or module where the given case might also be addressed. If additional changes are necessary are they documented in the code as a FIXME or the PR and in Github as an issue to be tracked?
- Will the code throw errors/warnings/stacktraces to the console during normal operation?
- Has all the debugging been removed?
- Does the code log any sensitive data? Does it show sensitive data in process lists? Does it store sensitive data to disk and if so, does it do so in a secure manner? Are there potential race conditions in between writing the data to disk and setting the appropriate permissions?
- Is it clear from the solution that the problem is well-understood? How can somebody who has never seen the problem feel confident that this proposed change is the best one?
- What's hard-coded that might not need to be? Are we making sensible decisions for the user and allowing them to tune and change things where appropriate?
- Are utility functions used where appropriate? Does this change re-implement something we already have code for?
- Is the right thing being fixed? There are cases where it's appropriate to fix a test and cases where it's appropriate to fix the code that's under test. Which is best for the user? Is this change a shortcut or a solution that will be solid in the months and years to come?
- How will this code react to changes elsewhere in the code base? What is it coupled to and have we fully thought through how best to present a coherent interface to consumers of a given function or method?
- Does this PR try to fix too many bugs/problems at once?
- Should this be split into multiple PRs to make them easier to test and reason about?

27.13 Reporting Bugs

Salt uses GitHub to track open issues and feature requests.

To file a bug, please navigate to the new issue page for the Salt project.

In an issue report, please include the following information:

- The output of `salt --versions-report` from the relevant machines. This can also be gathered remotely by using `salt <my_tgt> test.versions_report`.

- A description of the problem including steps taken to cause the issue to occur and the expected behaviour.
- Any steps taken to attempt to remediate the problem.
- Any configuration options set in a configuration file that may be relevant.
- A reproduceable test case. This may be as simple as an SLS file that illustrates a problem or it may be a link to a repository that contains a number of SLS files that can be used together to re-produce a problem. If the problem is transitory, any information that can be used to try and reproduce the problem is helpful.
- [Optional] The output of each salt component (master/minion/CLI) running with the `-ldebug` flag set.

Note: Please be certain to scrub any logs or SLS files for sensitive data!

27.14 Salt Topology

Salt is based on a powerful, asynchronous, network topology using ZeroMQ. Many ZeroMQ systems are in place to enable communication. The central idea is to have the fastest communication possible.

27.14.1 Servers

The Salt Master runs 2 network services. First is the ZeroMQ PUB system. This service by default runs on port `4505` and can be configured via the `publish_port` option in the master configuration.

Second is the ZeroMQ REP system. This is a separate interface used for all bi-directional communication with minions. By default this system binds to port `4506` and can be configured via the `ret_port` option in the master.

27.14.2 PUB/SUB

The commands sent out via the salt client are broadcast out to the minions via ZeroMQ PUB/SUB. This is done by allowing the minions to maintain a connection back to the Salt Master and then all connections are informed to download the command data at once. The command data is kept extremely small (usually less than 1K) so it is not a burden on the network.

27.14.3 Return

The PUB/SUB system is a one way communication, so once a publish is sent out the PUB interface on the master has no further communication with the minion. The minion, after running the command, then sends the command's return data back to the master via the `ret_port`.

27.15 Developing Salt Tutorial

This tutorial assumes you have:

- a web browser
- a GitHub account (`<my_account>`)

- a command line (CLI)
- git
- a text editor

27.15.1 Fork

In your browser, navigate to the `saltstack/salt` GitHub repository.

Click on `Fork` (https://github.com/saltstack/salt/#fork-destination-box).

Note: If you have more than one GitHub presence, for example if you are a member of a team, GitHub will ask you into which area to clone Salt. If you don't know where, then select your personal GitHub account.

27.15.2 Clone

In your CLI, navigate to the directory into which you want clone the Salt codebase and submit the following command:

```
$ git clone https://github.com/<my_account>/salt.git
```

where `<my_account>` is the name of your GitHub account. After the clone has completed, add SaltStack as a second remote and fetch any changes from `upstream`.

```
$ cd salt
$ git remote add upstream https://github.com/saltstack/salt.git
$ git fetch upstream
```

For this tutorial, we will be working off from the `master` branch, which is the default branch for the SaltStack GitHub project. This branch needs to track `upstream/master` so that we will get all upstream changes when they happen.

```
$ git checkout master
$ git branch --set-upstream-to upstream/master
```

27.15.3 Fetch

Fetch any `upstream` changes on the `master` branch and sync them to your local copy of the branch with a single command:

```
$ git pull --rebase
```

Note: For an explanation on `pull` vs `pull --rebase` and other excellent points, see this article by Mislav Marohnić.

27.15.4 Branch

Now we are ready to get to work. Consult the sprint beginner bug list and select an execution module whose `__virtual__` function needs to be updated. I'll select the `alternatives` module.

Create a new branch off from `master`. Be sure to name it something short and descriptive.

```
$ git checkout -b virt_ret
```

27.15.5 Edit

Edit the file you have selected, and verify that the changes are correct.

```
$ vim salt/modules/alternatives.py
$ git diff
```

```
diff --git a/salt/modules/alternatives.py b/salt/modules/alternatives.py
index 1653e5f..30c0a59 100644
--- a/salt/modules/alternatives.py
+++ b/salt/modules/alternatives.py
@@ -30,7 +30,7 @@ def __virtual__():
         '''
         if os.path.isdir('/etc/alternatives'):
                 return True
-        return False
+        return (False, 'Cannot load alternatives module: /etc/alternatives dir not
↪found')


 def _get_cmd():
```

27.15.6 Commit

Stage and commit the changes. Write a descriptive commit summary, but try to keep it less than 50 characters. Review your commit.

```
$ git add salt/modules/alternatives.py
$ git commit -m 'modules.alternatives: __virtual__ return err msg'
$ git show
```

Note: If you need more room to describe the changes in your commit, run `git commit` (without the `-m`, message, option) and you will be presented with an editor. The first line is the commit summary and should still be 50 characters or less. The following paragraphs you create are free form and will be preserved as part of the commit.

27.15.7 Push

Push your branch to your GitHub account. You will likely need to enter your GitHub username and password.

```
$ git push origin virt_ret
Username for 'https://github.com': <my_account>
Password for 'https://<my_account>@github.com':
```

Note: If authentication over https does not work, you can alternatively setup ssh keys. Once you have done this, you may need add the keys to your git repository configuration

```
$ git config ssh.key ~/.ssh/<key_name>
```

where `<key_name>` is the file name of the private key you created.

27.15.8 Merge

In your browser, navigate to the new pull request page on the `saltstack/salt` GitHub repository and click on `compare across forks`. Select `<my_account>` from the list of head forks and the branch you are wanting to merge into `master` (`virt_ret` in this case).

When you have finished reviewing the changes, click `Create pull request`.

If your pull request contains only a single commit, the title and comment will be taken from that commit's summary and message, otherwise the branch name is used for the title. Edit these fields as necessary and click `Create pull request`.

Note: Although these instructions seem to be the official pull request procedure on github's website, here are two alternative methods that are simpler.

- If you navigate to your clone of salt, `https://github.com/<my_account>/salt`, depending on how old your branch is or how recently you pushed updates on it, you may be presented with a button to create a pull request with your branch.
- I find it easiest to edit the following URL:

  `https://github.com/saltstack/salt/compare/master...<my_account>:virt_ret`

27.15.9 Resources

GitHub offers many great tutorials on various aspects of the git- and GitHub-centric development workflow:

https://help.github.com/

There are many topics covered by the Salt Developer documentation:

https://docs.saltstack.com/en/latest/topics/development/index.html

The contributing documentation presents more details on specific contributing topics:

https://docs.saltstack.com/en/latest/topics/development/contributing.html

27.16 Modular Systems

When first working with Salt, it is not always clear where all of the modular components are and what they do. Salt comes loaded with more modular systems than many users are aware of, making Salt very easy to extend in many places.

The most commonly used modular systems are execution modules and states. But the modular systems extend well beyond the more easily exposed components and are often added to Salt to make the complete system more flexible.

27.16.1 Developing New Modules

Interactive Debugging

Sometimes debugging with `print()` and extra logs sprinkled everywhere is not the best strategy.

IPython is a helpful debug tool that has an interactive python environment which can be embedded in python programs.

First the system will require IPython to be installed.

```
# Debian
apt-get install ipython

# Arch Linux
pacman -Syu ipython2

# RHEL/CentOS (via EPEL)
yum install python-ipython
```

Now, in the troubling python module, add the following line at a location where the debugger should be started:

```
test = 'test123'
import IPython; IPython.embed_kernel()
```

After running a Salt command that hits that line, the following will show up in the log file:

```
[CRITICAL] To connect another client to this kernel, use:
[IPKernelApp] --existing kernel-31271.json
```

Now on the system that invoked `embed_kernel`, run the following command from a shell:

```
# NOTE: use ipython2 instead of ipython for Arch Linux
ipython console --existing
```

This provides a console that has access to all the vars and functions, and even supports tab-completion.

```
print(test)
test123
```

To exit IPython and continue running Salt, press `Ctrl-d` to logout.

Special Module Contents

These are things that may be defined by the module to influence various things.

__virtual__

__virtual_aliases__

__virtualname__

__init__

Called before `__virtual__()`

__proxyenabled__

grains and proxy modules

__proxyenabled__ as a list containing the names of the proxy types that the module supports.

__load__

__func_alias__

__outputter__

Dunder Dictionaries

Salt provides several special "dunder" dictionaries as a convenience for Salt development. These include `__opts__`, `__context__`, `__salt__`, and others. This document will describe each dictionary and detail where they exist and what information and/or functionality they provide.

The following dunder dictionaries are always defined, but may be empty

- `__context__`
- `__grains__`
- `__pillar__`
- `__opts__`

__opts__

Defined in: All modules

The `__opts__` dictionary contains all of the options passed in the configuration file for the master or minion.

Note: In many places in salt, instead of pulling raw data from the __opts__ dict, configuration data should be pulled from the salt *get* functions such as config.get, aka - `__salt__['config.get']('foo:bar')` The *get* functions also allow for dict traversal via the *:* delimiter. Consider using get functions whenever using `__opts__` or `__pillar__` and `__grains__` (when using grains for configuration data)

The configuration file data made available in the `__opts__` dictionary is the configuration data relative to the running daemon. If the modules are loaded and executed by the master, then the master configuration data is available, if the modules are executed by the minion, then the minion configuration is available. Any additional information passed into the respective configuration files is made available

__salt__

Defined in: Auth, Beacons, Engines, Execution, Executors, Outputters, Pillars, Proxies, Renderers, Returners, Runners, SDB, SSH Wrappers, State, Thorium

`__salt__` contains the execution module functions. This allows for all functions to be called as they have been set up by the salt loader.

```
__salt__['cmd.run']('fdisk -l')
__salt__['network.ip_addrs']()
```

Note: When used in runners or outputters, `__salt__` references other runner/outputter modules, and not execution modules.

__grains__

Filled in for: Execution, Pillar, Renderer, Returner, SSH Wrapper, State.

The `__grains__` dictionary contains the grains data generated by the minion that is currently being worked with. In execution modules, state modules and returners this is the grains of the minion running the calls, when generating the external pillar the `__grains__` is the grains data from the minion that the pillar is being generated for.

While `__grains__` is defined for every module, it's only filled in for some.

__pillar__

Filled in for: Execution, Renderer, Returner, SSH Wrapper, State

The `__pillar__` dictionary contains the pillar for the respective minion.

While `__pillar__` is defined for every module, it's only filled in for some.

__ext_pillar__

Filled in for: Pillar

The `__ext_pillar__` dictionary contains the external pillar modules.

__context__

During a state run the `__context__` dictionary persists across all states that are run and then is destroyed when the state ends.

When running an execution module `__context__` persists across all module executions until the modules are refreshed; such as when *`saltutil.sync_all`* or *`state.apply`* are executed.

A great place to see how to use `__context__` is in the cp.py module in salt/modules/cp.py. The fileclient authenticates with the master when it is instantiated and then is used to copy files to the minion. Rather than create a new fileclient for each file that is to be copied down, one instance of the fileclient is instantiated in the `__context__` dictionary and is reused for each file. Here is an example from salt/modules/cp.py:

```
if not 'cp.fileclient' in __context__:
    __context__['cp.fileclient'] = salt.fileclient.get_file_client(__opts__)
```

Note: Because __context__ may or may not have been destroyed, always be sure to check for the existence of the key in __context__ and generate the key before using it.

__utils__

Defined in: Cloud, Engine, Execution, File Server, Grain, Pillar, Proxy, Roster, Runner, SDB, State

__proxy__

Defined in: Beacon, Engine, Execution, Executor, Proxy, Renderer, Returner, State, Util

__runner__

Defined in: Engine, Roster, Thorium

Note: When used in engines, it should be called __runners__ (plural)

__executors__

Defined in: Executor

__ret__

Defined in: Proxy

__thorium__

Defined in: Thorium

__states__

Defined in: Renderers, State

__serializers__

Defined in: State

__sdb__

Defined in: SDB

Additional Globals

Defined for: Runners, Execution Modules, Wheels

- `__jid__`: The job ID
- `__user__`: The user
- `__tag__`: The jid tag
- `__jid_event__`: A `salt.utils.event.NamespacedEvent`.

`NamespacedEvent` defines a single method `fire_event`, that takes data and tag. The *Runner docs* has examples.

27.16.2 Configuration Options

A number of configuration options can affect the load process. This is a quick list of them:

- `autoload_dynamic_modules` (*Minion*)
- `cython_enable` (*Minion*, *Master*)
- `disable_modules` (*Minion*)
- `disable_returners` (*Minion*)
- `enable_zip_modules` (*Minion*)
- `extension_modules` (*Master*)
- `extmod_whitelist` (*Minion*, *Master*)
- `extmod_blacklist` (*Minion*, *Master*)
- `whitelist_modules` (*Minion*)
- `grains_dirs` (*Minion*)
- `module_dirs` (*Minion*, *Master*)
- `outputter_dirs` (*Minion*, *Master*)
- `providers` (*Minion*)
- `render_dirs` (*Minion*)
- `returner_dirs` (*Minion*)
- `runner_dirs` (*Master*)
- `states_dirs` (*Minion*)
- `utils_dirs` (*Minion*)

27.16.3 Loading Modules

Modules come primarily from several sources:

- The Salt package itself
- The Salt File Server
- The extmods directory
- Secondary packages installed

Using one source to override another is not supported.

The Salt Package

Salt itself ships with a large number of modules. These are part of the Salt package itself and don't require the user to do anything to use them. (Although a number of them have additional dependencies and/or configuration.)

The Salt File Server

The user may add modules by simply placing them in special directories in their *fileserver*.

The name of the directory inside of the file server is the directory name prepended by underscore, such as:

- `_grains`
- `_modules`
- `_states`

Modules must be synced before they can be used. This can happen a few ways, discussed below.

Note: Using saltenvs besides `base` may not work in all contexts.

Sync Via States

The minion configuration contains an option `autoload_dynamic_modules` which defaults to `True`. This option makes the state system refresh all dynamic modules when states are run. To disable this behavior set *`autoload_dynamic_modules`* to `False` in the minion config.

When dynamic modules are autoloaded via states, only the modules defined in the same saltenvs as the states currently being run.

Also it is possible to use the explicit `saltutil.sync_*` *`state functions`* to sync the modules (previously it was necessary to use the `module.run` state):

Sync Via the saltutil Module

The saltutil module has a number of functions that can be used to sync all or specific dynamic modules. The `saltutil.sync_*` *`execution functions`* and *`runner functions`* can be used to sync modules to minions and the master, respectively.

The extmods Directory

Any files places in the directory set by `extension_modules` settings (`minion`, *`master`*, default `/var/cache/salt/*/extmods`) can also be loaded as modules. Note that these directories are also used by the `saltutil.sync_*` functions (mentioned above) and files may be overwritten.

Secondary Packages

Third-party packages may also add modules to Salt if they are installed in the same system and Python environment as the Salt Minion or Master.

This is done via setuptools entry points:

```
setup(
    # ...
    entry_points={
        'salt.loader': [
            'module_dirs=spirofs.loader:module',
        ],
    },
    # ...
)
```

Note that these are not synced from the Salt Master to the Minions. They must be installed independently on each Minion.

27.16.4 Module Types

The specific names used by each loading method above are as follows. See sections below for a short summary of each of these systems.

Module Type	Salt Package Name	FS/Directory Name	Entry Point
Auth	`salt.auth` (*index*)	`auth`[1]	`auth_dirs`
Beacon	`salt.beacons` (*index*)	`beacons`	`beacons_dirs`
Cache	`salt.cache` (*index*)	`cache`	`cache_dirs`
Cloud	`salt.cloud.clouds` (*index*)	`clouds`	`cloud_dirs`
Engine	`salt.engines` (*index*)	`engines`	`engines_dirs`
Execution	`salt.modules` (*index*)	`modules`	`module_dirs`
Executor	`salt.executors` (*index*)	`executors`	`executor_dirs`
File Server	`salt.fileserver` (*index*)	`fileserver`	`fileserver_dirs`
Grain	`salt.grains` (*index*)	`grains`	`grains_dirs`
Log Handler	`salt.log.handlers` (*index*)	`log_handlers`	`log_handlers_dirs`
Matcher	`salt.matchers`	`matchers`	`matchers_dirs`
Metaproxy	`salt.metaproxy`	`metaproxy`[1]	`metaproxy_dirs`
Net API	`salt.netapi` (*index*)	`netapi`[1]	`netapi_dirs`
Outputter	`salt.output` (*index*)	`output`	`outputter_dirs`
Pillar	`salt.pillar` (*index*)	`pillar`	`pillar_dirs`
Proxy	`salt.proxy` (*index*)	`proxy`	`proxy_dirs`
Queue	`salt.queues` (*index*)	`queues`	`queue_dirs`
Renderer	`salt.renderers` (*index*)	`renderers`	`render_dirs`
Returner	`salt.returners` (*index*)	`returners`	`returner_dirs`
Roster	`salt.roster` (*index*)	`roster`	`roster_dirs`

Continued on next page

Table 1 – continued from previous page

Module Type	Salt Package Name	FS/Directory Name	Entry Point
Runner	`salt.runners` (*index*)	`runners`	`runner_dirs`
SDB	`salt.sdb` (*index*)	`sdb`	`sdb_dirs`
Serializer	`salt.serializers` (*index*)	`serializers`[1]	`serializers_dirs`
SPM pkgdb	`salt.spm.pkgdb`	`pkgdb`[1]	`pkgdb_dirs`
SPM pkgfiles	`salt.spm.pkgfiles`	`pkgfiles`[1]	`pkgfiles_dirs`
SSH Wrapper	`salt.client.ssh.wrapper`	`wrapper`[1]	`wrapper_dirs`
State	`salt.states` (*index*)	`states`	`states_dirs`
Thorium	`salt.thorium` (*index*)	`thorium`	`thorium_dirs`
Tokens	`salt.tokens`	`tokens`	`tokens_dirs`
Top	`salt.tops` (*index*)	`tops`	`top_dirs`
Util	`salt.utils`	`utils`	`utils_dirs`
Wheel	`salt.wheels` (*index*)	`wheel`	`wheel_dirs`

Note: While it is possible to import modules directly with the import statement, it is strongly recommended that the appropriate *dunder dictionary* is used to access them instead. This is because a number of factors affect module names, module selection, and module overloading.

Auth

The auth module system allows for external authentication routines to be easily added into Salt. The *auth* function needs to be implemented to satisfy the requirements of an auth module. Use the `pam` module as an example.

See *External Authentication System* for more about authentication in Salt.

Beacon

- *Writing Beacons*

Beacons are polled by the Salt event loop to monitor non-salt processes. See *Beacons* for more information about the beacon system.

Cache

The minion cache is used by the master to store various information about minions. See *Minion Data Cache* for more information.

Cloud

Cloud modules are backend implementations used by *Salt Cloud*.

Engine

Engines are open-ended services managed by the Salt daemon (both master and minion). They may interact with event loop, call other modules, or a variety of non-salt tasks. See *Salt Engines* for complete details.

---

[1] These modules cannot be loaded from the Salt File Server.

Execution

Execution modules make up the core of the functionality used by Salt to interact with client systems. The execution modules create the core system management library used by all Salt systems, including states, which interact with minion systems.

Execution modules are completely open ended in their execution. They can be used to do anything required on a minion, from installing packages to detecting information about the system. The only restraint in execution modules is that the defined functions always return a JSON serializable object.

Executor

Executors control how execution modules get called. The default is to just call them, but this can be customized.

File Server

The file server module system is used to create file server backends used by the Salt Master. These modules need to implement the functions used in the fileserver subsystem. Use the `gitfs` module as an example.

See *File Server Backends* for more information.

Grains

- *Writing Grains*

Grain modules define extra routines to populate grains data. All defined public functions will be executed and MUST return a Python dict object. The dict keys will be added to the grains made available to the minion.

See *Grains* for more.

Log Handler

Log handlers allows the logs from salt (master or minion) to be sent to log aggregation systems.

Matcher

Matcher modules are used to define the *minion targeting expressions*. For now, it is only possible to override the *existing matchers* (the required CLI plumbing for custom matchers is not implemented yet).

Metaproxy

Metaproxy is an abstraction layer above the existing proxy minion. It enables adding different types of proxy minions that can still load existing proxymodules.

Net API

Net API modules are the actual server implementation used by Salt API.

Output

The output modules supply the outputter system with routines to display data in the terminal. These modules are very simple and only require the *output* function to execute. The default system outputter is the `nested` module.

Pillar

External Pillars

Salt provides a mechanism for generating pillar data by calling external pillar interfaces. This document will describe an outline of an ext_pillar module.

Location

Salt expects to find your `ext_pillar` module in the same location where it looks for other python modules. If the `extension_modules` option in your Salt master configuration is set, Salt will look for a `pillar` directory under there and load all the modules it finds. Otherwise, it will look in your Python site-packages `salt/pillar` directory.

Configuration

The external pillars that are called when a minion refreshes its pillars is controlled by the `ext_pillar` option in the Salt master configuration. You can pass a single argument, a list of arguments or a dictionary of arguments to your pillar:

```
ext_pillar:
  - example_a: some argument
  - example_b:
    - argumentA
    - argumentB
  - example_c:
      keyA: valueA
      keyB: valueB
```

The Module

Imports and Logging

Import modules your external pillar module needs. You should first include generic modules that come with stock Python:

```
import logging
```

And then start logging. This is an idiomatic way of setting up logging in Salt:

```
log = logging.getLogger(__name__)
```

Finally, load modules that are specific to what you are doing. You should catch import errors and set a flag that the `__virtual__` function can use later.

```
try:
    import weird_thing
    EXAMPLE_A_LOADED = True
except ImportError:
    EXAMPLE_A_LOADED = False
```

Options

If you define an `__opts__` dictionary, it will be merged into the `__opts__` dictionary handed to the `ext_pillar` function later. This is a good place to put default configuration items. The convention is to name things `modulename.option`.

```
__opts__ = { 'example_a.someconfig': 137 }
```

Initialization

If you define an `__init__` function, it will be called with the following signature:

```
def __init__( __opts__ ):
    # Do init work here
```

Note: The `__init__` function is ran every time a particular minion causes the external pillar to be called, so don't put heavy initialization code here. The `__init__` functionality is a side-effect of the Salt loader, so it may not be as useful in pillars as it is in other Salt items.

__virtual__

If you define a `__virtual__` function, you can control whether or not this module is visible. If it returns `False` then Salt ignores this module. If it returns a string, then that string will be how Salt identifies this external pillar in its `ext_pillar` configuration. If you're not renaming the module, simply return `True` in the `__virtual__` function, which is the same as if this function did not exist, then, the name Salt's `ext_pillar` will use to identify this module is its conventional name in Python.

This is useful to write modules that can be installed on all Salt masters, but will only be visible if a particular piece of software your module requires is installed.

```
# This external pillar will be known as `example_a`
def __virtual__():
    if EXAMPLE_A_LOADED:
        return True
    return False
```

```
# This external pillar will be known as `something_else`
__virtualname__ = 'something_else'

def __virtual__():
    if EXAMPLE_A_LOADED:
        return __virtualname__
    return False
```

ext_pillar

This is where the real work of an external pillar is done. If this module is active and has a function called `ext_pillar`, whenever a minion updates its pillar this function is called.

How it is called depends on how it is configured in the Salt master configuration. The first argument is always the current pillar dictionary, this contains pillar items that have already been added, starting with the data from `pillar_roots`, and then from any already-ran external pillars.

Using our example above:

```
ext_pillar( id, pillar, 'some argument' )                   # example_a
ext_pillar( id, pillar, 'argumentA', 'argumentB' )          # example_b
ext_pillar( id, pillar, keyA='valueA', keyB='valueB' )    # example_c
```

In the `example_a` case, `pillar` will contain the items from the `pillar_roots`, in `example_b` `pillar` will contain that plus the items added by `example_a`, and in `example_c` `pillar` will contain that plus the items added by `example_b`. In all three cases, `id` will contain the ID of the minion making the pillar request.

This function should return a dictionary, the contents of which are merged in with all of the other pillars and returned to the minion. **Note**: this function is called once for each minion that fetches its pillar data.

```
def ext_pillar( minion_id, pillar, *args, **kwargs ):

    my_pillar = {'external_pillar': {}}

    my_pillar['external_pillar'] = get_external_pillar_dictionary()

    return my_pillar
```

You can call pillar with the dictionary's top name to retrieve its data. From above example, 'external_pillar' is the top dictionary name. Therefore:

```
salt '*' pillar.get external_pillar
```

You shouldn't just add items to `pillar` and return that, since that will cause Salt to merge data that already exists. Rather, just return the items you are adding or changing. You could, however, use `pillar` in your module to make some decision based on pillar data that already exists.

This function has access to some useful globals:

__opts__ A dictionary of mostly Salt configuration options. If you had an `__opts__` dictionary defined in your module, those values will be included.

__salt__ A dictionary of Salt module functions, useful so you don't have to duplicate functions that already exist. E.g. `__salt__['cmd.run']( 'ls -l' )` **Note**, runs on the *master*

__grains__ A dictionary of the grains of the minion making this pillar call.

Example configuration

As an example, if you wanted to add external pillar via the `cmd_json` external pillar, add something like this to your master config:

```
ext_pillar:
  - cmd_json: 'echo {\"arg\":\"value\"}'
```

Reminder

Just as with traditional pillars, external pillars must be refreshed in order for minions to see any fresh data:

```
salt '*' saltutil.refresh_pillar
```

Used to define optional external pillar systems. The pillar generated via the filesystem pillar is passed into external pillars. This is commonly used as a bridge to database data for pillar, but is also the backend to the libvirt state used to generate and sign libvirt certificates on the fly.

Proxy

Proxy Minions are a way to manage devices that cannot run a full minion directly.

Renderers

Renderers are the system used to render sls files into salt highdata for the state compiler. They can be as simple as the `py` renderer and as complex as `stateconf` and `pydsl`.

Returners

Returners are used to send data from minions to external sources, commonly databases. A full returner will implement all routines to be supported as an external job cache. Use the `redis` returner as an example.

Roster

The *Roster system* is used by Salt SSH to enumerate devices.

Runners

Runners are purely master-side execution sequences.

SDB

- *Writing SDB Modules*

SDB is a way to store data that's not associated with a minion. See *Storing Data in Other Databases*.

Serializer

Primarily used with `file.serialize`.

State

State modules are used to define the state interfaces used by Salt States. These modules are restrictive in that they must follow a number of rules to function properly.

Note: State modules define the available routines in sls files. If calling an execution module directly is desired, take a look at the *module* state.

SPM pkgdb

- *SPM Development Guide: Package Database*

pkgdb modules provides storage backends to the package database.

SPM pkgfiles

- *SPM Development Guide: Package Database*

pkgfiles modules handle the actual installation.

SSH Wrapper

Replacement execution modules for *Salt SSH*.

Thorium

Modules for use in the *Thorium* event reactor.

Tokens

Token stores for *External Authentication*. See the `salt.tokens` docstring for details.

Note: The runner to load tokens modules is *`saltutil.sync_eauth_tokens`*.

Tops

Tops modules are used to convert external data sources into top file data for the state system.

Util

Just utility modules to use with other modules via `__utils__` (see *Dunder Dictionaries*).

Wheel

The wheel system is used to manage master side management routines. These routines are primarily intended for the API to enable master configuration.

27.17 Salt Extend

`salt-extend` is a templating tool for extending SaltStack. If you're looking to add a module to SaltStack, then the `salt-extend` utility can guide you through the process.

You can use Salt Extend to quickly create templated modules for adding new behaviours to some of the module subsystems within Salt.

Salt Extend takes a template directory and merges it into a SaltStack source code directory.

27.17.1 Command line usage

See salt-extend

27.17.2 Choosing a template

The following templates are available:

module

Creates a new execution module within salt/modules/{{module_name}}.py

module_unit

Creates a new execution module unit test suite within tests/unit/modules/test_{{module_name}}.py

state

Creates a new state module within salt/states/{{module_name}}.py

state_unit

Creates a new state module unit test suite within tests/unit/states/test_{{module_name}}.py

27.17.3 Adding templates

1. Create a directory under <src>/templates
2. Create a file `template.yml` containing properties for

- `description` - a description of the template
- `questions` - a collection of additional questions to ask the user, the name of the item will be used as the key in the context dictionary within the jinja template.
  - `question` - The question to ask the user, as a string
  - `default` - (optional) the default value, can contain Jinja2 template syntax and has access to the default context properties

Example template.yml

```
description: "Execution module"
questions:
    depending_libraries:
        question: "What libraries does this module depend upon?"
    virtual_name:
        question: "What module virtual name to use?"
        default: "{{module_name}}"
```

3. Create the files within <src>/templates/<your template> to match the target

---

Note: File names can contain Jinja 2 template syntax, e.g. *'{{module_name}}.py}}'*

Example file in the template directory

```
print('Hello {{module_name}}')
__virtual__ = '{{__virtual_name__}}'
```

Default context properties

The default context provides the following properties

- `description` - A description of the template
- `short_description` - A short description of the module as entered by the user
- `version` - The version name of the next release
- `module_name` - The module name as entered by the user
- `release_date` - The current date in the format *YYYY-MM-DD*
- `year` - The current year in the format *YYYY*

As well as any additional properties entered from the questions section of `template.yml`

27.17.4 API

salt.utils.extend module

SaltStack Extend

A templating tool for extending SaltStack.

Takes a template directory and merges it into a SaltStack source code directory. This tool uses Jinja2 for templating.

This tool is accessed using *salt-extend*

codeauthor Anthony Shaw <anthonyshaw@apache.org>

`salt.utils.extend.`**`apply_template`**(*template_dir*, *output_dir*, *context*)

Apply the template from the template directory to the output using the supplied context dict.

Parameters

- **`src`** (`str`) – The source path

- **dst** (`str`) – The destination path
- **context** (`dict`) – The dictionary to inject into the Jinja template as context

`salt.utils.extend.`**`run`**(*extension=None*, *name=None*, *description=None*, *salt_dir=None*, *merge=False*, *temp_dir=None*)

A template factory for extending the salt ecosystem

Parameters

- **extension** (`str`) – The extension type, e.g. 'module', 'state', if omitted, user will be prompted
- **name** (`str`) – Python-friendly name for the module, if omitted, user will be prompted
- **description** (`str`) – A description of the extension, if omitted, user will be prompted
- **salt_dir** (`str`) – The targeted Salt source directory
- **merge** (`bool`) – Merge with salt directory, *False* to keep separate, *True* to merge trees.
- **temp_dir** (`str`) – The directory for generated code, if omitted, system temp will be used

27.18 Salt's Test Suite

Salt comes with a powerful integration and unit test suite allowing for the fully automated run of integration and/or unit tests from a single interface.

To learn the basics of how Salt's test suite works, be sure to check out the *Salt's Test Suite: An Introduction* tutorial.

27.18.1 Test Directory Structure

Salt's test suite is located in the `tests` directory in the root of Salt's codebase. The test suite is divided into two main groups:

- *Integration Tests*
- *Unit Tests*

Within each of these groups, the directory structure roughly mirrors the structure of Salt's own codebase. Notice that there are directories for `states`, `modules`, `runners`, `output`, and more in each testing group.

The files that are housed in the `modules` directory of either the unit or the integration testing factions contain respective integration or unit test files for Salt execution modules.

Note: Salt's test framework provides for the option to only run tests which correspond to a given file (or set of files), via the `--from-filenames` argument to `runtests.py`:

```
python /path/to/runtests.py --from-filenames=salt/modules/foo.py
```

Therefore, where possible, test files should be named to match the source files they are testing. For example, when writing tests for `salt/modules/foo.py`, unit tests should go into `tests/unit/modules/test_foo.py`, and integration tests should go into `tests/integration/modules/test_foo.py`.

However, integration tests are organized differently from unit tests, and this may not always be plausible. In these cases, to ensure that the proper tests are run for these files, they must be mapped in tests/filename_map.yml.

The filename map is used to supplement the test framework's filename matching logic. This allows one to ensure that states correspnding to an execution module are also tested when `--from-filenames` includes that execution module. It can also be used for those cases where the path to a test file doesn't correspond directly to the file which is being tested (e.g. the `shell`, `spm`, and `ssh` integration tests, among others). Both glob expressions and regular expressions are permitted in the filename map.

Important: Test modules which don't map directly to the source file they are testing (using the naming convention described above), **must** be added to the `ignore` tuple in `tests/unit/test_module_names.py`, in the `test_module_name_source_match` function. This unit test ensures that we maintain the naming convention for test files.

Integration Tests

The Integration section of Salt's test suite start up a number of Salt daemons to test functionality in a live environment. These daemons include two Salt Masters, one Syndic, and two Minions. This allows the Syndic interface to be tested and Master/Minion communication to be verified. All of the integration tests are executed as live Salt commands sent through the started daemons.

Integration tests are particularly good at testing modules, states, and shell commands, among other segments of Salt's ecosystem. By utilizing the integration test daemons, integration tests are easy to write. They are also SaltStack's generally preferred method of adding new tests.

The discussion in the *Integration vs. Unit* section of the *testing tutorial* is beneficial in learning why you might want to write integration tests vs. unit tests. Both testing arenas add value to Salt's test suite and you should consider adding both types of tests if possible and appropriate when contributing to Salt.

- *Integration Test Documentation*

Unit Tests

Unit tests do not spin up any Salt daemons, but instead find their value in testing singular implementations of individual functions. Instead of testing against specific interactions, unit tests should be used to test a function's logic as well as any `return` or `raises` statements. Unit tests also rely heavily on mocking external resources.

The discussion in the *Integration vs. Unit* section of the *testing tutorial* is useful in determining when you should consider writing unit tests instead of, or in addition to, integration tests when contributing to Salt.

- *Unit Test Documentation*

27.18.2 Running The Tests

There are requirements, in addition to Salt's requirements, which need to be installed in order to run the test suite. Install one of the lines below, depending on the relevant Python version:

```
pip install -r requirements/dev_python27.txt
pip install -r requirements/dev_python34.txt
```

To be able to run integration tests which utilizes ZeroMQ transport, you also need to install additional requirements for it. Make sure you have installed the C/C++ compiler and development libraries and header files needed for your Python version.

This is an example for RedHat-based operating systems:

```
yum install gcc gcc-c++ python-devel
pip install -r requirements/zeromq.txt
```

On Debian, Ubuntu or their derivatives run the following commands:

```
apt-get install build-essential python-dev
pip install -r requirements/zeromq.txt
```

This will install the latest `pycrypto` and `pyzmq` (with bundled `libzmq`) Python modules required for running integration tests suite.

Once all requirements are installed, use `runtests.py` script to run all of the tests included in Salt's test suite:

```
python tests/runtests.py
```

For more information about options you can pass the test runner, see the `--help` option:

```
python tests/runtests.py --help
```

An alternative way of invoking the test suite is available in `setup.py`:

```
./setup.py test
```

Running Test Subsections

Instead of running the entire test suite all at once, which can take a long time, there are several ways to run only specific groups of tests or individual tests:

- Run *unit tests only*: `python tests/runtests.py --unit-tests`
- Run unit and integration tests for states: `python tests/runtests.py --state`
- Run integration tests for an individual module: `python tests/runtests.py -n integration.modules.virt`
- Run unit tests for an individual module: `python tests/runtests.py -n unit.modules.virt_test`
- Run an individual test by using the class and test name (this example is for the `test_default_kvm_profile` test in the `integration.module.virt`): `python tests/runtests.py -n integration.module.virt.VirtTest.test_default_kvm_profile`

For more specific examples of how to run various test subsections or individual tests, please see the *Test Selection Options* documentation or the *Running Specific Tests* section of the *Salt's Test Suite: An Introduction* tutorial.

Running Unit Tests Without Integration Test Daemons

Since the unit tests do not require a master or minion to execute, it is often useful to be able to run unit tests individually, or as a whole group, without having to start up the integration testing daemons. Starting up the master, minion, and syndic daemons takes a lot of time before the tests can even start running and is unnecessary to run unit tests. To run unit tests without invoking the integration test daemons, simply run the `runtests.py` script with `--unit` argument:

```
python tests/runtests.py --unit
```

All of the other options to run individual tests, entire classes of tests, or entire test modules still apply.

Running Destructive Integration Tests

Salt is used to change the settings and behavior of systems. In order to effectively test Salt's functionality, some integration tests are written to make actual changes to the underlying system. These tests are referred to as "destructive tests". Some examples of destructive tests are changes may be testing the addition of a user or installing packages. By default, destructive tests are disabled and will be skipped.

Generally, destructive tests should clean up after themselves by attempting to restore the system to its original state. For instance, if a new user is created during a test, the user should be deleted after the related test(s) have completed. However, no guarantees are made that test clean-up will complete successfully. Therefore, running destructive tests should be done with caution.

Note: Running destructive tests will change the underlying system. Use caution when running destructive tests.

To run tests marked as destructive, set the `--run-destructive` flag:

```
python tests/runtests.py --run-destructive
```

Running Cloud Provider Tests

Salt's testing suite also includes integration tests to assess the successful creation and deletion of cloud instances using *Salt-Cloud* for providers supported by Salt-Cloud.

The cloud provider tests are off by default and run on sample configuration files provided in `tests/integration/files/conf/cloud.providers.d/`. In order to run the cloud provider tests, valid credentials, which differ per provider, must be supplied. Each credential item that must be supplied is indicated by an empty string value and should be edited by the user before running the tests. For example, DigitalOcean requires a client key and an api key to operate. Therefore, the default cloud provider configuration file for DigitalOcean looks like this:

```
digitalocean-config:
  driver: digitalocean
  client_key: ''
  api_key: ''
  location: New York 1
```

As indicated by the empty string values, the `client_key` and the `api_key` must be provided:

```
digitalocean-config:
  driver: digitalocean
  client_key: wFGEwgregeqw3435gDger
  api_key: GDE43t43REGTrkilg43934t34qT43t4dgegerGEgg
  location: New York 1
```

Note: When providing credential information in cloud provider configuration files, do not include the single quotes.

Once all of the valid credentials for the cloud provider have been supplied, the cloud provider tests can be run by setting the `--cloud-provider-tests` flag:

```
./tests/runtests.py --cloud-provider-tests
```

Running The Tests In A Docker Container

The test suite can be executed under a docker container using the `--docked` option flag. The docker container must be properly configured on the system invoking the tests and the container must have access to the internet.

Here's a simple usage example:

```
python tests/runtests.py --docked=ubuntu-12.04 -v
```

The full docker container repository can also be provided:

```
python tests/runtests.py --docked=salttest/ubuntu-12.04 -v
```

The SaltStack team is creating some containers which will have the necessary dependencies pre-installed. Running the test suite on a container allows destructive tests to run without making changes to the main system. It also enables the test suite to run under a different distribution than the one the main system is currently using.

The current list of test suite images is on Salt's docker repository.

Custom docker containers can be provided by submitting a pull request against Salt's docker Salt test containers repository.

27.18.3 Automated Test Runs

SaltStack maintains a Jenkins server to allow for the execution of tests across supported platforms. The tests executed from Salt's Jenkins server create fresh virtual machines for each test run, then execute destructive tests on the new, clean virtual machine.

SaltStack's Jenkins server continuously runs the entire test suite, including destructive tests, on an array of various supported operating systems throughout the day. Each actively supported branch of Salt's repository runs the tests located in the respective branch's code. Each set of branch tests also includes a pylint run. These branch tests help ensure the viability of Salt code at any given point in time as pull requests are merged into branches throughout the day.

In addition to branch tests, SaltStack's Jenkins server also runs tests on pull requests. These pull request tests include a smaller set of virtual machines that run on the branch tests. The pull request tests, like the branch tests, include a pylint test as well.

When a pull request is submitted to Salt's repository on GitHub, the suite of pull request tests are started by Jenkins. These tests are used to gauge the pull request's viability to merge into Salt's codebase. If these initial tests pass, the pull request can then merged into the Salt branch by one of Salt's core developers, pending their discretion. If the initial tests fail, core developers may request changes to the pull request. If the failure is unrelated to the changes in question, core developers may merge the pull request despite the initial failure.

As soon as the pull request is merged, the changes will be added to the next branch test run on Jenkins.

For a full list of currently running test environments, go to http://jenkins.saltstack.com.

Using Salt-Cloud on Jenkins

For testing Salt on Jenkins, SaltStack uses *Salt-Cloud* to spin up virtual machines. The script using Salt-Cloud to accomplish this is open source and can be found here: https://github.com/saltstack/salt/blob/master/tests/jenkins.py

27.18.4 Writing Tests

The salt testing infrastructure is divided into two classes of tests, integration tests and unit tests. These terms may be defined differently in other contexts, but for Salt they are defined this way:

- Unit Test: Tests which validate isolated code blocks and do not require external interfaces such as `salt-call` or any of the salt daemons.
- Integration Test: Tests which validate externally accessible features.

Salt testing uses unittest2 from the python standard library and MagicMock.

- *Writing integration tests*
- *Writing unit tests*

Naming Conventions

Any function in either integration test files or unit test files that is doing the actual testing, such as functions containing assertions, must start with `test_`:

```
def test_user_present(self):
```

When functions in test files are not prepended with `test_`, the function acts as a normal, helper function and is not run as a test by the test suite.

Submitting New Tests

Which branch of the Salt codebase should new tests be written against? The location of where new tests should be submitted depends largely on the reason you're writing the tests.

Tests for New Features

If you are adding new functionality to Salt, please write the tests for this new feature in the same pull request as the new feature. New features should always be submitted to the `master` branch.

If you have already submitted the new feature, but did not write tests in the original pull request that has already been merged, please feel free to submit a new pull request containing tests. If the feature was recently added to Salt's `master` branch, then the tests should be added there as well. However, if the feature was added to `master` some time ago and is already present in one or more release branches, please refer to the *Tests for Entire Files or Functions* section below for more details about where to submit tests for functions or files that do not already have tests.

Tests to Accompany a Bugfix

If you are writing tests for code that fixes a bug in Salt, please write the test in the same pull request as the bugfix. If you're unsure of where to submit your bugfix and accompanying test, please review the *Which Salt Branch?* documentation in Salt's *Contributing* guide.

Tests for Entire Files or Functions

Sometimes entire files in Salt are completely untested. If you are writing tests for a file that doesn't have any tests written for it, write your test against the earliest supported release branch that contains the file or function you're testing.

Once your tests are submitted in a pull request and is merged into the branch in question, the tests you wrote will be merged-forward by SaltStack core engineers and the new tests will propagate to the newer release branches. That way the tests you wrote will apply to all current and relevant release branches, and not just the `master` branch, for example. This methodology will help protect against regressions on older files in Salt's codebase.

There may be times when the tests you write against an older branch fail in the merge-forward process because functionality has changed in newer release branches. In these cases, a Salt core developer may reach out to you for advice on the tests in question if the path forward is unclear.

Note: If tests are written against a file in an older release branch and then merged forward, there may be new functionality in the file that is present in the new release branch that is untested.It would be wise to see if new functionality could use additional testing once the test file has propagated to newer release branches.

Test Helpers

Several Salt-specific helpers are available. A full list is available by inspecting functions exported in *tests.support.helpers*.

@expensiveTest – Designates a test which typically requires a relatively costly external resource, like a cloud virtual machine. This decorator is not normally used by developers outside of the Salt core team.

@destructiveTest – Marks a test as potentially destructive. It will not be run by the test runner unless the `-run-destructive` test is expressly passed.

@requires_network – Requires a network connection for the test to operate successfully. If a network connection is not detected, the test will not run.

@requires_salt_modules – Requires all the modules in a list of modules in order for the test to be executed. Otherwise, the test is skipped.

@requires_system_grains – Loads and passes the grains on the system as an keyword argument to the test function with the name *grains*.

@skip_if_binaries_missing(['list', 'of', 'binaries']) – If called from inside a test, the test will be skipped if the binaries are not all present on the system.

@skip_if_not_root – If the test is not executed as root, it will be skipped.

@with_system_user – Creates and optionally destroys a system user within a test case. See implementation details in *tests.support.helpers* for details.

@with_system_group – Creates and optionally destroys a system group within a test case. See implementation details in *tests.support.helpers* for details.

@with_system_user_and_group – Creates and optionally destroys a system user and group within a test case. See implementation details in *tests.support.helpers* for details.

27.19 Integration Tests

The Salt integration tests come with a number of classes and methods which allow for components to be easily tested. These classes are generally inherited from and provide specific methods for hooking into the running integration test environment created by the integration tests.

It is noteworthy that since integration tests validate against a running environment that they are generally the preferred means to write tests.

The integration system is all located under `tests/integration` in the Salt source tree. Each directory within `tests/integration` corresponds to a directory in Salt's tree structure. For example, the integration tests for the `test.py` Salt module that is located in `salt/modules` should also be named `test.py` and reside in `tests/integration/modules`.

27.19.1 Preparing to Write Integration Tests

This guide assumes that your Salt development environment is already configured and that you have a basic understanding of contributing to the Salt codebase. If you're unfamiliar with either of these topics, please refer to the *Installing Salt for Development* and the *Contributing* pages, respectively.

This documentation also assumes that you have an understanding of how to *run Salt's test suite*, including running the *test subsections*, and running a single integration test file, class, or individual test.

27.19.2 Best Practices

Integration tests should be written to the following specifications.

What to Test?

Since integration tests are used to validate against a running Salt environment, integration tests should be written with the Salt components, and their various interactions, in mind.

- Isolate testing functionality. Don't rely on the pass or failure of other, separate tests.
- Individual tests should test against a single behavior.
- Since it occasionally takes some effort to "set up" an individual test, it may be necessary to call several functions within a single test. However, be sure that once the work has been done to set up a test, make sure you are clear about the functionality that is being tested.

Naming Conventions

Test names and docstrings should indicate what functionality is being tested. Test functions are named `test_<fcn>_<test-name>` where `<fcn>` is the function being tested and `<test-name>` describes the behavior being tested.

In order for integration tests to get picked up during a run of the test suite, each individual test must be prepended with the `test_` naming syntax, as described above.

If a function does not start with `test_`, then the function acts as a "normal" function and is not considered a testing function. It will not be included in the test run or testing output.

The setUp and tearDown Functions

There are two special functions that can be utilized in the integration side of Salt's test suite: `setUp` and `tearDown`. While these functions are not required in all test files, there are many examples in Salt's integration test suite illustrating the broad usefulness of each function.

The `setUp` function is used to set up any repetitive or useful tasks that the tests in a test class need before running. For example, any of the `mac_*` integration tests should only run on macOS machines. The `setUp` function can be used to test for the presence of the `Darwin` kernel. If the `Darwin` kernel is not present, then the test should be skipped.

```
def setUp(self):
    '''
    Sets up test requirements
    '''
    os_grain = self.run_function('grains.item', ['kernel'])
    if os_grain['kernel'] not in 'Darwin':
        self.skipTest(
            'Test not applicable to \'{kernel}\' kernel'.format(
                **os_grain
            )
        )
```

The `setUp` function can be used for many things. The above code snippet is only one example. Another example might be to ensure that a particular setting is present before running tests that would require the setting.

The `tearDown` function is used to clean up after any tests. This function is useful for restoring any settings that might have been changed during the test run.

Note: The `setUp` and `tearDown` functions run before and after each test in the test class that the `setUp` and `tearDown` functions are defined.

Be sure to read the *Destructive vs Non-Destructive Tests* section when using any kind of destructive functions that might alter the system running the test suite in either the `setUp` or `tearDown` function definitions.

Testing Order

The test functions within a test class do not run in the order they were defined, but instead run in lexicographical order.

Note that if any `setUp` or `tearDown` functions are defined in the class, those functions will run before (for `setUp`) or after (for `tearDown`) each test case.

27.19.3 Integration Classes

The integration classes are located in `tests/integration/__init__.py` and can be extended therein. There are four classes available to extend:

- *ModuleCase*
- *ShellCase*
- *SSHCase*
- *SyndicCase*

ModuleCase

Used to define executions run via the master to minions and to call single modules and states. The available testing functions are:

run_function

Run a single salt function and condition the return down to match the behavior of the raw function call. This will run the command and only return the results from a single minion to verify.

run_state

Run the state.single command and return the state return structure.

minion_run

Run a single salt function on the 'minion' target and condition the return down to match the behavior of the raw function call.

ShellCase

Shell out to the scripts which ship with Salt. The testing functions are:

run_cp

Execute salt-cp. Pass in the argument string as it would be passed on the command line.

run_call

Execute salt-call, pass in the argument string as it would be passed on the command line.

run_cloud

Execute the salt-cloud command. Pass in the argument string as it would be passed on the command line.

run_key

Execute the salt-key command. Pass in the argument string as it would be passed on the command line.

run_run

Execute the salt-run command. Pass in the argument string as it would be passed on the command line.

run_run_plus

Execute the runner function the and return the return data and output in a dict

run_salt

Execute the salt command. Pass in the argument string as it would be passed on the command line.

run_script

Execute a salt script with the given argument string.

run_ssh

Execute the salt-ssh. Pass in the argument string as it would be passed on the command line.

SSHCase

Used to execute remote commands via salt-ssh. The available methods are as follows:

run_function

Run a single salt function via salt-ssh and condition the return down to match the behavior of the raw function call. This will run the command and only return the results from a single minion to verify.

SyndicCase

Used to execute remote commands via a syndic and is only used to verify the capabilities of the Salt Syndic. The available methods are as follows:

run_function

Run a single salt function and condition the return down to match the behavior of the raw function call. This will run the command and only return the results from a single minion to verify.

27.19.4 Examples

The following sections define simple integration tests present in Salt's integration test suite for each type of testing class.

Module Example via ModuleCase Class

Import the integration module, this module is already added to the python path by the test execution. Inherit from the `integration.ModuleCase` class.

Now the workhorse method `run_function` can be used to test a module:

```
import os
import tests.integration as integration


class TestModuleTest(integration.ModuleCase):
    '''
    Validate the test module
    '''
    def test_ping(self):
        '''
        test.ping
        '''
        self.assertTrue(self.run_function('test.ping'))

    def test_echo(self):
        '''
        test.echo
        '''
        self.assertEqual(self.run_function('test.echo', ['text']), 'text')
```

The fist example illustrates the testing master issuing a `test.ping` call to a testing minion. The test asserts that the minion returned with a `True` value to the master from the `test.ping` call.

The second example similarly verifies that the minion executed the `test.echo` command with the `text` argument. The `assertEqual` call maintains that the minion ran the function and returned the data as expected to the master.

Shell Example via ShellCase

Validating the shell commands can be done via shell tests:

```
import sys
import shutil
import tempfile

import tests.integration as integration

class KeyTest(integration.ShellCase):
    '''
    Test salt-key script
    '''

    _call_binary_ = 'salt-key'

    def test_list(self):
        '''
        test salt-key -L
        '''
        data = self.run_key('-L')
        expect = [
                'Unaccepted Keys:',
                'Accepted Keys:',
                'minion',
                'sub_minion',
                'Rejected:', '']
        self.assertEqual(data, expect)
```

This example verifies that the `salt-key` command executes and returns as expected by making use of the `run_key` method.

SSH Example via SSHCase

Testing salt-ssh functionality can be done using the SSHCase test class:

```
import tests.integration as integration

class SSHGrainsTest(integration.SSHCase):
'''
Test salt-ssh grains functionality
Depend on proper environment set by integration.SSHCase class
'''

def test_grains_id(self):
    '''
    Test salt-ssh grains id work for localhost.
    '''
    cmd = self.run_function('grains.get', ['id'])
    self.assertEqual(cmd, 'localhost')
```

Testing Event System via SaltMinionEventAssertsMixin

The fundamentially asynchronous nature of Salt makes testing the event system a challenge. The `SaltMinionEventAssertsMixin` provides a facility for testing that events were received on a minion event bus.

```
import tests.integration as integration
import salt.utils.event

class TestEvent(integration.SaltEventAssertsMixin):
    '''
    Example test of firing an event and receiving it
    '''

    def test_event(self):
        e = salt.utils.event.get_event('minion', sock_dir=self.minion_opts['sock_dir
↪'], opts=self.minion_opts)

        e.fire_event({'a': 'b'}, '/test_event')

        self.assertMinionEventReceived({'a': 'b'})
```

Syndic Example via SyndicCase

Testing Salt's Syndic can be done via the SyndicCase test class:

```
import tests.integration as integration

class TestSyndic(integration.SyndicCase):
    '''
    Validate the syndic interface by testing the test module
    '''
    def test_ping(self):
```

(continues on next page)

(continued from previous page)

```
        '''
        test.ping
        '''
        self.assertTrue(self.run_function('test.ping'))
```

This example verifies that a `test.ping` command is issued from the testing master, is passed through to the testing syndic, down to the minion, and back up again by using the `run_function` located with in the `SyndicCase` test class.

27.19.5 Integration Test Files

Since using Salt largely involves configuring states, editing files, and changing system data, the integration test suite contains a directory named `files` to aid in testing functions that require files. Various Salt integration tests use these example files to test against instead of altering system files and data.

Each directory within `tests/integration/files` contain files that accomplish different tasks, based on the needs of the integration tests using those files. For example, `tests/integration/files/ssh` is used to bootstrap the test runner for salt-ssh testing, while `tests/integration/files/pillar` contains files storing data needed to test various pillar functions.

The `tests/integration/files` directory also includes an integration state tree. The integration state tree can be found at `tests/integration/files/file/base`.

The following example demonstrates how integration files can be used with ModuleCase to test states:

```
# Import python libs
from __future__ import absolute_import
import os
import shutil

# Import Salt Testing libs
from tests.support.case import ModuleCase
from tests.support.paths import FILES, TMP
from tests.support.mixins import SaltReturnAssertsMixin

# Import salt libs
import salt.utils.files

HFILE = os.path.join(TMP, 'hosts')


class HostTest(ModuleCase, SaltReturnAssertsMixin):
    '''
    Validate the host state
    '''

    def setUp(self):
        shutil.copyfile(os.path.join(FILES, 'hosts'), HFILE)
        super(HostTest, self).setUp()

    def tearDown(self):
        if os.path.exists(HFILE):
            os.remove(HFILE)
        super(HostTest, self).tearDown()

    def test_present(self):
```

(continues on next page)

(continued from previous page)

```
        '''
        host.present
        '''
        name = 'spam.bacon'
        ip = '10.10.10.10'
        ret = self.run_state('host.present', name=name, ip=ip)
        self.assertSaltTrueReturn(ret)
        with salt.utils.files.fopen(HFILE) as fp_:
            output = fp_.read()
            self.assertIn('{0}\t\t{1}'.format(ip, name), output)
```

To access the integration files, a variable named `FILES` points to the `tests/integration/files` directory. This is where the referenced `host.present` sls file resides.

In addition to the static files in the integration state tree, the location `TMP` can also be used to store temporary files that the test system will clean up when the execution finishes.

27.19.6 Destructive vs Non-Destructive Tests

Since Salt is used to change the settings and behavior of systems, one testing approach is to run tests that make actual changes to the underlying system. This is where the concept of destructive integration tests comes into play. Tests can be written to alter the system they are running on. This capability is what fills in the gap needed to properly test aspects of system management like package installation.

Any test that changes the underlying system in any way, such as creating or deleting users, installing packages, or changing permissions should include the `@destructive` decorator to signal system changes and should be written with care. System changes executed within a destructive test should also be restored once the related tests have completed. For example, if a new user is created to test a module, the same user should be removed after the test is completed to maintain system integrity.

To write a destructive test, import, and use the destructiveTest decorator for the test method:

```
import tests.integration as integration
from tests.support.helpers import destructiveTest, skip_if_not_root

class DestructiveExampleModuleTest(integration.ModuleCase):
    '''
    Demonstrate a destructive test
    '''

    @destructiveTest
    @skip_if_not_root
    def test_user_not_present(self):
        '''
        This is a DESTRUCTIVE TEST it creates a new user on the minion.
        And then destroys that user.
        '''
        ret = self.run_state('user.present', name='salt_test')
        self.assertSaltTrueReturn(ret)
        ret = self.run_state('user.absent', name='salt_test')
        self.assertSaltTrueReturn(ret)
```

27.19.7 Cloud Provider Tests

Cloud provider integration tests are used to assess *Salt-Cloud*'s ability to create and destroy cloud instances for various supported cloud providers. Cloud provider tests inherit from the ShellCase Integration Class.

Any new cloud provider test files should be added to the `tests/integration/cloud/providers/` directory. Each cloud provider test file also requires a sample cloud profile and cloud provider configuration file in the integration test file directory located at `tests/integration/files/conf/cloud.*.d/`.

The following is an example of the default profile configuration file for Digital Ocean, located at: `tests/integration/files/conf/cloud.profiles.d/digitalocean.conf`:

```
digitalocean-test:
  provider: digitalocean-config
  image: Ubuntu 14.04 x64
  size: 512MB
```

Each cloud provider requires different configuration credentials. Therefore, sensitive information such as API keys or passwords should be omitted from the cloud provider configuration file and replaced with an empty string. The necessary credentials can be provided by the user by editing the provider configuration file before running the tests.

The following is an example of the default provider configuration file for Digital Ocean, located at: `tests/integration/files/conf/cloud.providers.d/digitalocean.conf`:

```
digitalocean-config:
  driver: digitalocean
  client_key: ''
  api_key: ''
  location: New York 1
```

In addition to providing the necessary cloud profile and provider files in the integration test suite file structure, appropriate checks for if the configuration files exist and contain valid information are also required in the test class's `setUp` function:

```
from tests.support.case import ShellCase
from tests.support.paths import FILES

class LinodeTest(ShellCase):
'''
Integration tests for the Linode cloud provider in Salt-Cloud
'''

def setUp(self):
    '''
    Sets up the test requirements
    '''
    super(LinodeTest, self).setUp()

    # check if appropriate cloud provider and profile files are present
    profile_str = 'linode-config:'
    provider = 'linode'
    providers = self.run_cloud('--list-providers')
    if profile_str not in providers:
        self.skipTest(
            'Configuration file for {0} was not found. Check {0}.conf files '
            'in tests/integration/files/conf/cloud.*.d/ to run these tests.'
            .format(provider)
        )
```

(continues on next page)

(continued from previous page)

```
    # check if apikey and password are present
    path = os.path.join(FILES,
                        'conf',
                        'cloud.providers.d',
                        provider + '.conf')
    config = cloud_providers_config(path)
    api = config['linode-config']['linode']['apikey']
    password = config['linode-config']['linode']['password']
    if api == '' or password == '':
        self.skipTest(
            'An api key and password must be provided to run these tests. Check '
            'tests/integration/files/conf/cloud.providers.d/{0}.conf'.format(
                provider
            )
        )
```

Repeatedly creating and destroying instances on cloud providers can be costly. Therefore, cloud provider tests are off by default and do not run automatically. To run the cloud provider tests, the `--cloud-provider-tests` flag must be provided:

```
./tests/runtests.py --cloud-provider-tests
```

Since cloud provider tests do not run automatically, all provider tests must be preceded with the `@expensiveTest` decorator. The expensive test decorator is necessary because it signals to the test suite that the `--cloud-provider-tests` flag is required to run the cloud provider tests.

To write a cloud provider test, import, and use the expensiveTest decorator for the test function:

```
from tests.support.helpers import expensiveTest

@expensiveTest
def test_instance(self):
    '''
    Test creating an instance on Linode
    '''
    name = 'linode-testing'

    # create the instance
    instance = self.run_cloud('-p linode-test {0}'.format(name))
    str = '        {0}'.format(name)

    # check if instance with salt installed returned as expected
    try:
        self.assertIn(str, instance)
    except AssertionError:
        self.run_cloud('-d {0} --assume-yes'.format(name))
        raise

    # delete the instance
    delete = self.run_cloud('-d {0} --assume-yes'.format(name))
    str = '            True'
    try:
        self.assertIn(str, delete)
    except AssertionError:
        raise
```

27.19.8 Adding New Directories

If the corresponding Salt directory does not exist within `tests/integration`, the new directory must be created along with the appropriate test file to maintain Salt's testing directory structure.

In order for Salt's test suite to recognize tests within the newly created directory, options to run the new integration tests must be added to `tests/runtests.py`. Examples of the necessary options that must be added can be found here: https://github.com/saltstack/salt/blob/master/tests/runtests.py. The functions that need to be edited are `setup_additional_options`, `validate_options`, and `run_integration_tests`.

27.20 Writing Unit Tests

27.20.1 Introduction

Like many software projects, Salt has two broad-based testing approaches – integration testing and unit testing. While integration testing focuses on the interaction between components in a sandboxed environment, unit testing focuses on the singular implementation of individual functions.

Unit tests should be used specifically to test a function's logic. Unit tests rely on mocking external resources.

While unit tests are good for ensuring consistent results, they are most useful when they do not require more than a few mocks. Effort should be made to mock as many external resources as possible. This effort is encouraged, but not required. Sometimes the isolation provided by completely mocking the external dependencies is not worth the effort of mocking those dependencies.

In these cases, requiring an external library to be installed on the system before running the test file is a useful way to strike this balance. For example, the unit tests for the MySQL execution module require the presence of the MySQL python bindings on the system running the test file before proceeding to run the tests.

Overly detailed mocking can also result in decreased test readability and brittleness as the tests are more likely to fail when the code or its dependencies legitimately change. In these cases, it is better to add dependencies to the test runner dependency state.

27.20.2 Preparing to Write a Unit Test

This guide assumes that your Salt development environment is already configured and that you have a basic understanding of contributing to the Salt codebase. If you're unfamiliar with either of these topics, please refer to the *Installing Salt for Development* and the *Contributing* pages, respectively.

This documentation also assumes that you have an understanding of how to *run Salt's test suite*, including running the *unit test subsection*, running the unit tests *without testing daemons* to speed up development wait times, and running a unit test file, class, or individual test.

27.20.3 Best Practices

Unit tests should be written to the following specifications.

What to Test?

Since unit testing focuses on the singular implementation of individual functions, unit tests should be used specifically to test a function's logic. The following guidelines should be followed when writing unit tests for Salt's test suite:

- Each `raise` and `return` statement needs to be independently tested.
- Isolate testing functionality. Don't rely on the pass or failure of other, separate tests.
- Test functions should contain only one assertion.
- Many Salt execution modules are merely wrappers for distribution-specific functionality. If there isn't any logic present in a simple execution module, consider writing an *integration test* instead of heavily mocking a call to an external dependency.

Mocking Test Data

A reasonable effort needs to be made to mock external resources used in the code being tested, such as APIs, function calls, external data either globally available or passed in through function arguments, file data, etc.

- Test functions should contain only one assertion and all necessary mock code and data for that assertion.
- External resources should be mocked in order to "block all of the exits". If a test function fails because something in an external library wasn't mocked properly (or at all), this test is not addressing all of the "exits" a function may experience. We want the Salt code and logic to be tested, specifically.
- Consider the fragility and longevity of a test. If the test is so tightly coupled to the code being tested, this makes a test unnecessarily fragile.
- Make sure you are not mocking the function to be tested so vigorously that the test return merely tests the mocked output. The test should always be testing a function's logic.

Mocking Loader Modules

Salt loader modules use a series of globally available dunder variables, `__salt__`, `__opts__`, `__pillar__`, etc. To facilitate testing these modules a mixin class was created, `LoaderModuleMockMixin` which can be found in `tests/support/mixins.py`. The reason for the existence of this class is because historically one would add these dunder variables directly on the imported module. This, however, introduces unexpected behavior when running the full test suite since those attributes would not be removed once we were done testing the module and would therefore leak to other modules being tested with unpredictable results. This is the kind of work that should be deferred to mock, and that's exactly what this mixin class does.

As an example, if one needs to specify some options which should be available to the module being tested one should do:

```
import salt.modules.somemodule as somemodule

class SomeModuleTest(TestCase, LoaderModuleMockMixin):

    def setup_loader_modules(self):
        return {
            somemodule: {
                '__opts__': {'test': True}
            }
        }
```

Consider this more extensive example from `tests/unit/modules/test_libcloud_dns.py`:

```
# Import Python Libs
from __future__ import absolute_import

# Import Salt Testing Libs
```

(continues on next page)

(continued from previous page)

```
from tests.support.mixins import LoaderModuleMockMixin
from tests.support.unit import TestCase
from tests.support.mock import (
    patch,
    MagicMock,
)
import salt.modules.libcloud_dns as libcloud_dns


class MockDNSDriver(object):
    def __init__(self):
        pass


def get_mock_driver():
    return MockDNSDriver()


@patch('salt.modules.libcloud_dns._get_driver',
       MagicMock(return_value=MockDNSDriver()))
class LibcloudDnsModuleTestCase(TestCase, LoaderModuleMockMixin):

    def setup_loader_modules(self):
        module_globals = {
            '__salt__': {
                'config.option': MagicMock(return_value={
                    'test': {
                        'driver': 'test',
                        'key': '2orgk34kgk34g'
                    }
                })
            }
        }
        if libcloud_dns.HAS_LIBCLOUD is False:
            module_globals['sys.modules'] = {'libcloud': MagicMock()}

        return {libcloud_dns: module_globals}
```

What happens in the above example is we mock a call to *__salt__['config.option']* to return the configuration needed for the execution of the tests. Additionally, if the `libcloud` library is not available, since that's not actually part of what's being tested, we mocked that import by patching `sys.modules` when tests are running.

Mocking Filehandles

Note: This documentation applies to the 2018.3 release cycle and newer. The extended functionality for `mock_open` described below does not exist in the 2017.7 and older release branches.

Opening files in Salt is done using `salt.utils.files.fopen()`. When testing code that reads from files, the `mock_open` helper can be used to mock filehandles. Note that is not the same `mock_open` as `unittest.mock.mock_open()` from the Python standard library, but rather a separate implementation which has additional functionality.

```
from tests.support.unit import TestCase
from tests.support.mock import (
    patch
    mock_open,
)

import salt.modules.mymod as mymod

class MyAwesomeTestCase(TestCase):

    def test_something(self):
        fopen_mock = mock_open(read_data='foo\nbar\nbaz\n')
        with patch('salt.utils.files.fopen', fopen_mock):
            result = mymod.myfunc()
            assert result is True
```

This will force any filehandle opened to mimic a filehandle which, when read, produces the specified contents.

Important: String Types

When running tests on Python 2, `mock_open` will convert any `unicode` types to `str` types to more closely reproduce Python 2 behavior (file reads are always `str` types in Python 2, irrespective of mode).

However, when configuring your read_data, make sure that you are using bytestrings (e.g. `b'foo\nbar\nbaz\n'`) when the code you are testing is opening a file for binary reading, otherwise the tests will fail on Python 3. The mocked filehandles produced by `mock_open` will raise a `TypeError` if you attempt to read a bytestring when opening for non-binary reading, and similarly will not let you read a string when opening a file for binary reading. They will also not permit bytestrings to be ”written” if the mocked filehandle was opened for non-binary writing, and vice-versa when opened for non-binary writing. These enhancements force test writers to write more accurate tests.

More Complex Scenarios

Multiple File Paths

What happens when the code being tested reads from more than one file? For those cases, you can pass `read_data` as a dictionary:

```
import textwrap

from tests.support.unit import TestCase
from tests.support.mock import (
    patch
    mock_open,
)

import salt.modules.mymod as mymod

class MyAwesomeTestCase(TestCase):

    def test_something(self):
        contents = {
            '/etc/foo.conf': textwrap.dedent('''\
```

(continues on next page)

(continued from previous page)

```
                foo
                bar
                baz
                '''),
            '/etc/b*.conf': textwrap.dedent('''\
                one
                two
                three
                '''),
        }
        fopen_mock = mock_open(read_data=contents)
        with patch('salt.utils.files.fopen', fopen_mock):
            result = mymod.myfunc()
            assert result is True
```

This would make `salt.utils.files.fopen()` produce filehandles with different contents depending on which file was being opened by the code being tested. `/etc/foo.conf` and any file matching the pattern `/etc/b*.conf` would work, while opening any other path would result in a `FileNotFoundError` being raised (in Python 2, an `IOError`).

Since file patterns are supported, it is possible to use a pattern of `'*'` to define a fallback if no other patterns match the filename being opened. The below two `mock_open` calls would produce identical results:

```
mock_open(read_data='foo\n')
mock_open(read_data={'*': 'foo\n'})
```

Note: Take care when specifying the `read_data` as a dictionary, in cases where the patterns overlap (e.g. when both `/etc/b*.conf` and `/etc/bar.conf` are in the `read_data`). Dictionary iteration order will determine which pattern is attempted first, second, etc., with the exception of `*` which is used when no other pattern matches. If your test case calls for specifying overlapping patterns, and you are not running Python 3.6 or newer, then an `OrderedDict` can be used to ensure matching is handled in the desired way:

```
contents = OrderedDict()
contents['/etc/bar.conf'] = 'foo\nbar\nbaz\n'
contents['/etc/b*.conf'] = IOError(errno.EACCES, 'Permission denied')
contents['*'] = 'This is a fallback for files not beginning with "/etc/b"\n'
fopen_mock = mock_open(read_data=contents)
```

Raising Exceptions

Instead of a string, an exception can also be used as the `read_data`:

```
import errno

from tests.support.unit import TestCase
from tests.support.mock import (
    patch
    mock_open,
)

import salt.modules.mymod as mymod

```

(continues on next page)

(continued from previous page)

```
class MyAwesomeTestCase(TestCase):

    def test_something(self):
        exc = IOError(errno.EACCES, 'Permission denied')
        fopen_mock = mock_open(read_data=exc)
        with patch('salt.utils.files.fopen', fopen_mock):
            mymod.myfunc()
```

The above example would raise the specified exception when any file is opened. The expectation would be that `mymod.myfunc()` would gracefully handle the IOError, so a failure to do that would result in it being raised and causing the test to fail.

Multiple File Contents

For cases in which a file is being read more than once, and it is necessary to test a function's behavior based on what the file looks like the second (or third, etc.) time it is read, just specify the the contents for that file as a list. Each time the file is opened, `mock_open` will cycle through the list and produce a mocked filehandle with the specified contents. For example:

```
import errno
import textwrap

from tests.support.unit import TestCase
from tests.support.mock import (
    patch
    mock_open,
)

import salt.modules.mymod as mymod

class MyAwesomeTestCase(TestCase):

    def test_something(self):
        contents = {
            '/etc/foo.conf': [
                textwrap.dedent('''\
                    foo
                    bar
                    '''),
                textwrap.dedent('''\
                    foo
                    bar
                    baz
                    '''),
            ],
            '/etc/b*.conf': [
                IOError(errno.ENOENT, 'No such file or directory'),
                textwrap.dedent('''\
                    one
                    two
                    three
                    '''),
            ],
        }
        fopen_mock = mock_open(read_data=contents)
```

(continues on next page)

(continued from previous page)

```
        with patch('salt.utils.files.fopen', fopen_mock):
            result = mymod.myfunc()
            assert result is True
```

Using this example, the first time `/etc/foo.conf` is opened, it will simulate a file with the first string in the list as its contents, while the second time it is opened, the simulated file's contents will be the second string in the list.

If no more items remain in the list, then attempting to open the file will raise a `RuntimeError`. In the example above, if `/etc/foo.conf` were to be opened a third time, a `RuntimeError` would be raised.

Note that exceptions can also be mixed in with strings when using this technique. In the above example, if `/etc/bar.conf` were to be opened twice, the first time would simulate the file not existing, while the second time would simulate a file with string defined in the second element of the list.

Note: Notice that the second path in the `contents` dictionary above (`/etc/b*.conf`) contains an asterisk. The items in the list are cycled through for each match of a given pattern (*not* separately for each individual file path), so this means that only two files matching that pattern could be opened before the next one would raise a `RuntimeError`.

Accessing the Mocked Filehandles in a Test

Note: The code for the `MockOpen`, `MockCall`, and `MockFH` classes (referenced below) can be found in `tests/support/mock.py`. There are extensive unit tests for them located in `tests/unit/test_mock.py`.

The above examples simply show how to mock `salt.utils.files.fopen()` to simulate files with the contents you desire, but you can also access the mocked filehandles (and more), and use them to craft assertions in your tests. To do so, just add an `as` clause to the end of the `patch` statement:

```
fopen_mock = mock_open(read_data='foo\nbar\nbaz\n')
with patch('salt.utils.files.fopen', fopen_mock) as m_open:
    # do testing here
    ...
    ...
```

When doing this, `m_open` will be a `MockOpen` instance. It will contain several useful attributes:

- **read_data** - A dictionary containing the `read_data` passed when `mock_open` was invoked. In the event that *multiple file paths* are not used, then this will be a dictionary mapping `*` to the `read_data` passed to `mock_open`.
- **call_count** - An integer representing how many times `salt.utils.files.fopen()` was called to open a file.
- **calls** - A list of `MockCall` objects. A `MockCall` object is a simple class which stores the arguments passed to it, making the positional arguments available via its `args` attribute, and the keyword arguments available via its `kwargs` attribute.

  ```
  from tests.support.unit import TestCase
  from tests.support.mock import (
      patch
      mock_open,
      MockCall,
  ```

(continues on next page)

(continued from previous page)

```
)

import salt.modules.mymod as mymod

class MyAwesomeTestCase(TestCase):

    def test_something(self):

        with patch('salt.utils.files.fopen', mock_open(read_data=b'foo\n')) as m_
↪open:
            mymod.myfunc()
            # Assert that only two opens attempted
            assert m_open.call_count == 2
            # Assert that only /etc/foo.conf was opened
            assert all(call.args[0] == '/etc/foo.conf' for call in m_open.calls)
            # Asser that the first open was for binary read, and the
            # second was for binary write.
            assert m_open.calls == [
                MockCall('/etc/foo.conf', 'rb'),
                MockCall('/etc/foo.conf', 'wb'),
            ]
```

Note that `MockCall` is imported from `tests.support.mock` in the above example. Also, the second assert above is redundant since it is covered in the final assert, but both are included simply as an example.

- **filehandles** - A dictionary mapping the unique file paths opened, to lists of `MockFH` objects. Each open creates a unique `MockFH` object. Each `MockFH` object itself has a number of useful attributes:
  - **filename** - The path to the file which was opened using `salt.utils.files.fopen()`
  - **call** - A `MockCall` object representing the arguments passed to `salt.utils.files.fopen()`. Note that this `MockCall` is also available in the parent `MockOpen` instance's **calls** list.
  - The following methods are mocked using `unittest.mock.Mock` objects, and Mock's built-in asserts (as well as the call data) can be used as you would with any other Mock object:
    - **.read()**
    - **.readlines()**
    - **.readline()**
    - **.close()**
    - **.write()**
    - **.writelines()**
    - **.seek()**
  - The read functions (**.read()**, **.readlines()**, **.readline()**) all work as expected, as does iterating through the file line by line (i.e. `for line in fh:`).
  - The **.tell()** method is also implemented in such a way that it updates after each time the mocked filehandle is read, and will report the correct position. The one caveat here is that **.seek()** doesn't actually work (it's simply mocked), and will not change the position. Additionally, neither **.write()** or **.writelines()** will modify the mocked filehandle's contents.
  - The attributes **.write_calls** and **.writelines_calls** (no parenthesis) are available as shorthands and correspond to lists containing the contents passed for all calls to **.write()** and **.writelines()**, respectively.

Examples

```
with patch('salt.utils.files.fopen', mock_open(read_data=contents)) as m_open:
    # Run the code you are unit testing
    mymod.myfunc()
    # Check that only the expected file was opened, and that it was opened
    # only once.
    assert m_open.call_count == 1
    assert list(m_open.filehandles) == ['/etc/foo.conf']
    # "opens" will be a list of all the mocked filehandles opened
    opens = m_open.filehandles['/etc/foo.conf']
    # Check that we wrote the expected lines ("expected" here is assumed to
    # be a list of strings)
    assert opens[0].write_calls == expected
```

```
with patch('salt.utils.files.fopen', mock_open(read_data=contents)) as m_open:
    # Run the code you are unit testing
    mymod.myfunc()
    # Check that .readlines() was called (remember, it's a Mock)
    m_open.filehandles['/etc/foo.conf'][0].readlines.assert_called()
```

```
with patch('salt.utils.files.fopen', mock_open(read_data=contents)) as m_open:
    # Run the code you are unit testing
    mymod.myfunc()
    # Check that we read the file and also wrote to it
    m_open.filehandles['/etc/foo.conf'][0].read.assert_called_once()
    m_open.filehandles['/etc/foo.conf'][1].writelines.assert_called_once()
```

Naming Conventions

Test names and docstrings should indicate what functionality is being tested. Test functions are named `test_<fcn>_<test-name>` where `<fcn>` is the function being tested and `<test-name>` describes the `raise` or `return` being tested.

Unit tests for `salt/.../<module>.py` are contained in a file called `tests/unit/.../test_<module>.py`, e.g. the tests for `salt/modules/fib.py` are in `tests/unit/modules/test_fib.py`.

In order for unit tests to get picked up during a run of the unit test suite, each unit test file must be prefixed with `test_` and each individual test must be prepended with the `test_` naming syntax, as described above.

If a function does not start with `test_`, then the function acts as a "normal" function and is not considered a testing function. It will not be included in the test run or testing output. The same principle applies to unit test files that do not have the `test_*.py` naming syntax. This test file naming convention is how the test runner recognizes that a test file contains unit tests.

Imports

Most commonly, the following imports are necessary to create a unit test:

```
from tests.support.unit import TestCase
```

If you need mock support to your tests, please also import:

```
from tests.support.mock import MagicMock, patch, call
```

27.20.4 Evaluating Truth

A longer discussion on the types of assertions one can make can be found by reading Python's documentation on unit testing.

27.20.5 Tests Using Mock Objects

In many cases, the purpose of a Salt module is to interact with some external system, whether it be to control a database, manipulate files on a filesystem or something else. In these varied cases, it's necessary to design a unit test which can test the function whilst replacing functions which might actually call out to external systems. One might think of this as "blocking the exits" for code under tests and redirecting the calls to external systems with our own code which produces known results during the duration of the test.

To achieve this behavior, Salt makes heavy use of the MagicMock package.

To understand how one might integrate Mock into writing a unit test for Salt, let's imagine a scenario in which we're testing an execution module that's designed to operate on a database. Furthermore, let's imagine two separate methods, here presented in pseduo-code in an imaginary execution module called 'db.py'.

```
def create_user(username):
    qry = 'CREATE USER {0}'.format(username)
    execute_query(qry)

def execute_query(qry):
    # Connect to a database and actually do the query...
```

Here, let's imagine that we want to create a unit test for the *create_user* function. In doing so, we want to avoid any calls out to an external system and so while we are running our unit tests, we want to replace the actual interaction with a database with a function that can capture the parameters sent to it and return pre-defined values. Therefore, our task is clear – to write a unit test which tests the functionality of *create_user* while also replacing 'execute_query' with a mocked function.

To begin, we set up the skeleton of our class much like we did before, but with additional imports for MagicMock:

```
# Import Salt Testing libs
from tests.support.unit import TestCase

# Import Salt execution module to test
from salt.modules import db

# Import Mock libraries
from tests.support.mock import MagicMock, patch, call

# Create test case class and inherit from Salt's customized TestCase
# Skip this test case if we don't have access to mock!
class DbTestCase(TestCase):
    def test_create_user(self):
        # First, we replace 'execute_query' with our own mock function
        with patch.object(db, 'execute_query', MagicMock()) as db_exq:

            # Now that the exits are blocked, we can run the function under test.
            db.create_user('testuser')

            # We could now query our mock object to see which calls were made
            # to it.
            ## print db_exq.mock_calls
```

(continues on next page)

(continued from previous page)

```
            # Construct a call object that simulates the way we expected
            # execute_query to have been called.
            expected_call = call('CREATE USER testuser')

            # Compare the expected call with the list of actual calls.  The
            # test will succeed or fail depending on the output of this
            # assertion.
            db_exq.assert_has_calls(expected_call)
```

27.20.6 Modifying `__salt__` In Place

At times, it becomes necessary to make modifications to a module's view of functions in its own `__salt__` dictionary. Luckily, this process is quite easy.

Below is an example that uses MagicMock's `patch` functionality to insert a function into `__salt__` that's actually a MagicMock instance.

```
def show_patch(self):
    with patch.dict(my_module.__salt__,
                    {'function.to_replace': MagicMock()}):
        # From this scope, carry on with testing, with a modified __salt__!
```

27.20.7 A Simple Example

Let's assume that we're testing a very basic function in an imaginary Salt execution module. Given a module called `fib.py` that has a function called `calculate(num_of_results)`, which given a `num_of_results`, produces a list of sequential Fibonacci numbers of that length.

A unit test to test this function might be commonly placed in a file called `tests/unit/modules/test_fib.py`. The convention is to place unit tests for Salt execution modules in `test/unit/modules/` and to name the tests module prefixed with `test_*.py`.

Tests are grouped around test cases, which are logically grouped sets of tests against a piece of functionality in the tested software. Test cases are created as Python classes in the unit test module. To return to our example, here's how we might write the skeleton for testing `fib.py`:

```
# Import Salt Testing libs
from tests.support.unit import TestCase

# Import Salt execution module to test
import salt.modules.fib as fib

# Create test case class and inherit from Salt's customized TestCase
class FibTestCase(TestCase):
    '''
    This class contains a set of functions that test salt.modules.fib.
    '''
    def test_fib(self):
        '''
        To create a unit test, we should prefix the name with `test_' so
        that it's recognized by the test runner.
        '''
```

(continues on next page)

(continued from previous page)

```
        fib_five = (0, 1, 1, 2, 3)
        self.assertEqual(fib.calculate(5), fib_five)
```

At this point, the test can now be run, either individually or as a part of a full run of the test runner. To ease development, a single test can be executed:

```
tests/runtests.py -v -n unit.modules.test_fib
```

This will report the status of the test: success, failure, or error. The `-v` flag increases output verbosity.

```
tests/runtests.py -n unit.modules.test_fib -v
```

To review the results of a particular run, take a note of the log location given in the output for each test:

```
Logging tests on /var/folders/nl/d809xbq577l3qrbj3ymtpbq80000gn/T/salt-runtests.log
```

27.20.8 A More Complete Example

Consider the following function from salt/modules/linux_sysctl.py.

```
def get(name):
    '''
    Return a single sysctl parameter for this minion

    CLI Example:

    .. code-block:: bash

        salt '*' sysctl.get net.ipv4.ip_forward
    '''
    cmd = 'sysctl -n {0}'.format(name)
    out = __salt__['cmd.run'](cmd)
    return out
```

This function is very simple, comprising only four source lines of code and having only one return statement, so we know only one test is needed. There are also two inputs to the function, the `name` function argument and the call to `__salt__['cmd.run']()`, both of which need to be appropriately mocked.

Mocking a function parameter is straightforward, whereas mocking a function call will require, in this case, the use of MagicMock. For added isolation, we will also redefine the `__salt__` dictionary such that it only contains `'cmd.run'`.

```
# Import Salt Libs
import salt.modules.linux_sysictl as linux_sysctl

# Import Salt Testing Libs
from tests.support.mixins import LoaderModuleMockMixin
from tests.support.unit import TestCase
from tests.support.mock import (
    MagicMock,
    patch,
)


class LinuxSysctlTestCase(TestCase, LoaderModuleMockMixin):
```

(continues on next page)

(continued from previous page)

```
    '''
    TestCase for salt.modules.linux_sysctl module
    '''

    def test_get(self):
        '''
        Tests the return of get function
        '''
        mock_cmd = MagicMock(return_value=1)
        with patch.dict(linux_sysctl.__salt__, {'cmd.run': mock_cmd}):
            self.assertEqual(linux_sysctl.get('net.ipv4.ip_forward'), 1)
```

Since `get()` has only one raise or return statement and that statement is a success condition, the test function is simply named `test_get()`. As described, the single function call parameter, `name` is mocked with `net.ipv4.ip_forward` and `__salt__['cmd.run']` is replaced by a MagicMock function object. We are only interested in the return value of `__salt__['cmd.run']`, which MagicMock allows us by specifying via `return_value=1`. Finally, the test itself tests for equality between the return value of `get()` and the expected return of `1`. This assertion is expected to succeed because `get()` will determine its return value from `__salt__['cmd.run']`, which we have mocked to return `1`.

27.20.9 A Complex Example

Now consider the `assign()` function from the same salt/modules/linux_sysctl.py source file.

```
def assign(name, value):
    '''
    Assign a single sysctl parameter for this minion

    CLI Example:

    .. code-block:: bash

        salt '*' sysctl.assign net.ipv4.ip_forward 1
    '''
    value = str(value)
    sysctl_file = '/proc/sys/{0}'.format(name.replace('.', '/'))
    if not os.path.exists(sysctl_file):
        raise CommandExecutionError('sysctl {0} does not exist'.format(name))

    ret = {}
    cmd = 'sysctl -w {0}="{1}"'.format(name, value)
    data = __salt__['cmd.run_all'](cmd)
    out = data['stdout']
    err = data['stderr']

    # Example:
    #    # sysctl -w net.ipv4.tcp_rmem="4096 87380 16777216"
    #    net.ipv4.tcp_rmem = 4096 87380 16777216
    regex = re.compile(r'^{0}\s+=\s+{1}$'.format(re.escape(name),
                                                  re.escape(value)))

    if not regex.match(out) or 'Invalid argument' in str(err):
        if data['retcode'] != 0 and err:
            error = err
        else:
```

(continues on next page)

(continued from previous page)

```
            error = out
        raise CommandExecutionError('sysctl -w failed: {0}'.format(error))
    new_name, new_value = out.split(' = ', 1)
    ret[new_name] = new_value
    return ret
```

This function contains two raise statements and one return statement, so we know that we will need (at least) three tests. It has two function arguments and many references to non-builtin functions. In the tests below you will see that MagicMock's `patch()` method may be used as a context manager or as a decorator. When patching the salt dunders however, please use the context manager approach.

There are three test functions, one for each raise and return statement in the source function. Each function is self-contained and contains all and only the mocks and data needed to test the raise or return statement it is concerned with.

```
# Import Salt Libs
import salt.modules.linux_sysctl as linux_sysctl
from salt.exceptions import CommandExecutionError

# Import Salt Testing Libs
from tests.support.mixins import LoaderModuleMockMixin
from tests.support.unit import TestCase
from tests.support.mock import (
    MagicMock,
    patch,
)


class LinuxSysctlTestCase(TestCase, LoaderModuleMockMixin):
    '''
    TestCase for salt.modules.linux_sysctl module
    '''

    @patch('os.path.exists', MagicMock(return_value=False))
    def test_assign_proc_sys_failed(self):
        '''
        Tests if /proc/sys/<kernel-subsystem> exists or not
        '''
        cmd = {'pid': 1337, 'retcode': 0, 'stderr': '',
               'stdout': 'net.ipv4.ip_forward = 1'}
        mock_cmd = MagicMock(return_value=cmd)
        with patch.dict(linux_sysctl.__salt__, {'cmd.run_all': mock_cmd}):
            self.assertRaises(CommandExecutionError,
                              linux_sysctl.assign,
                              'net.ipv4.ip_forward', 1)

    @patch('os.path.exists', MagicMock(return_value=True))
    def test_assign_cmd_failed(self):
        '''
        Tests if the assignment was successful or not
        '''
        cmd = {'pid': 1337, 'retcode': 0, 'stderr':
               'sysctl: setting key "net.ipv4.ip_forward": Invalid argument',
               'stdout': 'net.ipv4.ip_forward = backward'}
        mock_cmd = MagicMock(return_value=cmd)
        with patch.dict(linux_sysctl.__salt__, {'cmd.run_all': mock_cmd}):
```

(continues on next page)

(continued from previous page)

```
        self.assertRaises(CommandExecutionError,
                          linux_sysctl.assign,
                          'net.ipv4.ip_forward', 'backward')

@patch('os.path.exists', MagicMock(return_value=True))
def test_assign_success(self):
    '''
    Tests the return of successful assign function
    '''
    cmd = {'pid': 1337, 'retcode': 0, 'stderr': '',
           'stdout': 'net.ipv4.ip_forward = 1'}
    ret = {'net.ipv4.ip_forward': '1'}
    mock_cmd = MagicMock(return_value=cmd)
    with patch.dict(linux_sysctl.__salt__, {'cmd.run_all': mock_cmd}):
        self.assertEqual(linux_sysctl.assign(
            'net.ipv4.ip_forward', 1), ret)
```

27.21 SaltStack Git Policy

The SaltStack team follows a git policy to maintain stability and consistency with the repository.

The git policy has been developed to encourage contributions and make contributing to Salt as easy as possible. Code contributors to SaltStack projects DO NOT NEED TO READ THIS DOCUMENT, because all contributions come into SaltStack via a single gateway to make it as easy as possible for contributors to give us code.

The primary rule of git management in SaltStack is to make life easy on contributors and developers to send in code. Simplicity is always a goal!

27.21.1 New Code Entry

All new SaltStack code should be submitted against either the `develop` branch or a point release branch, depending on the nature of the submission. Please see the *Which Salt Branch?* section of Salt's *Contributing* documentation or the Release Branching section below for more information.

27.21.2 Release Branching

SaltStack maintains two types of releases, `Feature Releases` and `Point Releases` (also commonly referred to as `Bugfix Releases`. A feature release is managed by incrementing the first or second release point number, so 2015.5.5 -> 2015.8.0 signifies a feature release and 2015.8.0 -> 2015.8.1 signifies a point release.

Feature Release Branching

Each feature release is maintained in a dedicated git branch derived from the last applicable release commit on develop. All file changes relevant to the feature release will be completed in the develop branch prior to the creation of the feature release branch. The feature release branch will be named after the relevant numbers to the feature release, which constitute the first two numbers. This means that the release branch for the 2015.8.0 series is named 2015.8.

A feature release branch is created with the following command:

```
# git checkout -b 2015.8 # From the develop branch
# git push origin 2015.8
```

Point Releases

Each point release is derived from its parent release branch. Constructing point releases is a critical aspect of Salt development and is managed by members of the core development team. Point releases comprise bug and security fixes. Bug fixes can be made against a point release branch in one of two ways: the bug fix can be submitted directly against the point release branch, or an attempt can be made to back-port the fix to the point release branch.

Bug fixes should be made against the earliest supported release branch on which the bug is present. The Salt development team regularly merges older point release branches forward into newer point release branches. That way, the bug fixes that are submitted to older release branches can cascade up through all related release branches.

For more information, please see the *Which Salt Branch?* section of Salt's *Contributing* documentation.

Generally point releases are made every 2 months or if there is a security fix they can be made sooner.

The point release is designated by branching (ex 2019.2.1) and then tagging (v2019.2.1) from that newly created release branch when its determined the release is stable. From the tag point a new source tarball is generated and published to PyPI, and a release announcement is made.

27.22 Salt Conventions

27.22.1 Writing Salt Documentation

Salt's documentation is built using the Sphinx documentation system. It can be built in a large variety of output formats including HTML, PDF, ePub, and manpage.

All the documentation is contained in the main Salt repository. Speaking broadly, most of the narrative documentation is contained within the https://github.com/saltstack/salt/blob/master/doc subdirectory and most of the reference and API documentation is written inline with Salt's Python code and extracted using a Sphinx extension.

Style

The Salt project recommends the IEEE style guide as a general reference for writing guidelines. Those guidelines are not strictly enforced but rather serve as an excellent resource for technical writing questions. The NCBI style guide is another very approachable resource.

Point-of-view

Use third-person perspective and avoid "I", "we", "you" forms of address. Identify the addressee specifically e.g., "users should", "the compiler does", etc.

Active voice

Use active voice and present-tense. Avoid filler words.

Title capitalization

Document titles and section titles within a page should follow normal sentence capitalization rules. Words that are capitalized as part of a regular sentence should be capitalized in a title and otherwise left as lowercase. Punctuation can be omitted unless it aids the intent of the title (e.g., exclamation points or question marks).

For example:

```
This is a main heading
======================

Paragraph.

This is an exciting sub-heading!
--------------------------------

Paragraph.
```

Serial Commas

According to Wikipedia: In English punctuation, a serial comma or series comma (also called Oxford comma and Harvard comma) is a comma placed immediately before the coordinating conjunction (usually "and", "or", or "nor") in a series of three or more terms. For example, a list of three countries might be punctuated either as "France, Italy, and Spain" (with the serial comma), or as "France, Italy and Spain" (without the serial comma)."

When writing a list that includes three or more items, the serial comma should always be used.

Documenting modules

Documentation for Salt's various module types is inline in the code. During the documentation build process it is extracted and formatted into the final HTML, PDF, etc format.

Inline documentation

Python has special multi-line strings called docstrings as the first element in a function or class. These strings allow documentation to live alongside the code and can contain special formatting. For example:

```python
def my_function(value):
    '''
    Upper-case the given value

    Usage:

    .. code-block:: python

        val = 'a string'
        new_val = myfunction(val)
        print(new_val) # 'A STRING'

    :param value: a string
    :return: a copy of ``value`` that has been upper-cased
    '''
    return value.upper()
```

Specify a release for additions or changes

New functions or changes to existing functions should include a marker that denotes what Salt release will be affected. For example:

```
def my_function(value):
    '''
    Upper-case the given value

    .. versionadded:: 2014.7.0

    <...snip...>
    '''
    return value.upper()
```

For changes to a function:

```
def my_function(value, strip=False):
    '''
    Upper-case the given value

    .. versionchanged:: 2016.3.0
        Added a flag to also strip whitespace from the string.

    <...snip...>
    '''
    if strip:
        return value.upper().strip()
    return value.upper()
```

Adding module documentation to the index

Each module type has an index listing all modules of that type. For example: *execution modules*, *state modules*, *renderer modules*. New modules must be added to the index manually.

1. Edit the file for the module type: execution modules, state modules, renderer modules, etc.
2. Add the new module to the alphebetized list.
3. *Build the documentation* which will generate an `.rst` file for the new module in the same directory as the `index.rst`.
4. Commit the changes to `index.rst` and the new `.rst` file and send a pull request.

Cross-references

The Sphinx documentation system contains a wide variety of cross-referencing capabilities.

Glossary entries

Link to *glossary entries* using the term role. A cross-reference should be added the first time a Salt-specific term is used in a document.

```
A common way to encapsulate master-side functionality is by writing a
custom :term:`Runner Function`. Custom Runner Functions are easy to write.
```

Index entries

Sphinx automatically generates many kinds of index entries, but it is occasionally useful to manually add items to the index.

One method is to use the index directive above the document or section that should appear in the index.

```
.. index:: ! Event, event bus, event system
    see: Reactor; Event
```

Another method is to use the index role inline with the text that should appear in the index. The index entry is created and the target text is left otherwise intact.

```
Information about the :index:`Salt Reactor`
-------------------------------------------

Paragraph.
```

Documents and sections

Each document should contain a unique top-level label of the form:

```
.. _my-page:

My page
=======

Paragraph.
```

Unique labels can be linked using the ref role. This allows cross-references to survive document renames or movement.

```
For more information see :ref:`my-page`.
```

Note, the `:doc:` role should *not* be used to link documents together.

Modules

Cross-references to Salt modules can be added using Sphinx's Python domain roles. For example, to create a link to the `test.ping` function:

```
A useful execution module to test active communication with a minion is the
:py:func:`test.ping <salt.modules.test.ping>` function.
```

Salt modules can be referenced as well:

```
The :py:mod:`test module <salt.modules.test>` contains many useful
functions for inspecting an active Salt connection.
```

The same syntax works for all modules types:

```
One of the workhorse state module functions in Salt is the
:py:func:`file.managed <salt.states.file.managed>` function.
```

Settings

Individual settings in the Salt Master or Salt Minion configuration files are cross-referenced using two custom roles, `conf_master`, and `conf_minion`.

```
The :conf_minion:`minion ID <id>` setting is a unique identifier for a
single minion.
```

Documentation Changes and Fixes

Documentation changes and fixes should be made against the earliest supported release branch that the update applies to. The practice of updating a release branch instead of making all documentation changes against Salt's main, default branch, `develop`, is necessary in order for the docs to be as up-to-date as possible when the docs are built.

The workflow mentioned above is also in line with the recommendations outlined in Salt's *Contributing* page. You can read more about how to choose where to submit documentation fixes by reading the *Salt's Branch Topology* section.

For an explanation of how to submit changes against various branches, see the *Sending a GitHub pull request* section. Specifically, see the section describing how to `Create a new branch` and the steps that follow.

Building the documentation

1. Install Sphinx using a system package manager or pip. The package name is often of the form `python-sphinx`. There are no other dependencies.
2. Build the documentation using the provided Makefile or `.bat` file on Windows.

   ```
   cd /path/to/salt/doc
   make html
   ```

3. The generated documentation will be written to the `doc/_build/<format>` directory.
4. A useful method of viewing the HTML documentation locally is to start Python's built-in HTTP server:

   Python 3:

   ```
   cd /path/to/salt/doc/_build/html
   python3 -m http.server
   ```

   Python 2:

   ```
   cd /path/to/salt/doc/_build/html
   python -m SimpleHTTPServer
   ```

   Then pull up the documentation in a web browser at http://localhost:8000/.

27.22.2 Salt Formulas

Formulas are pre-written Salt States. They are as open-ended as Salt States themselves and can be used for tasks such as installing a package, configuring, and starting a service, setting up users or permissions, and many other common tasks.

All official Salt Formulas are found as separate Git repositories in the "saltstack-formulas" organization on GitHub:

https://github.com/saltstack-formulas

As a simple example, to install the popular Apache web server (using the normal defaults for the underlying distro) simply include the apache-formula from a top file:

```
base:
  'web*':
    - apache
```

Installation

Each Salt Formula is an individual Git repository designed as a drop-in addition to an existing Salt State tree. Formulas can be installed in the following ways.

Adding a Formula as a GitFS remote

One design goal of Salt's GitFS fileserver backend was to facilitate reusable States. GitFS is a quick and natural way to use Formulas.

1. *Install any necessary dependencies and configure GitFS*.
2. Add one or more Formula repository URLs as remotes in the `gitfs_remotes` list in the Salt Master configuration file:

   ```
   gitfs_remotes:
     - https://github.com/saltstack-formulas/apache-formula
     - https://github.com/saltstack-formulas/memcached-formula
   ```

   We strongly recommend forking a formula repository into your own GitHub account to avoid unexpected changes to your infrastructure.

   Many Salt Formulas are highly active repositories so pull new changes with care. Plus any additions you make to your fork can be easily sent back upstream with a quick pull request!
3. Restart the Salt master.

Beginning with the 2018.3.0 release, using formulas with GitFS is now much more convenient for deployments which use many different fileserver environments (i.e. saltenvs). Using the *all_saltenvs* parameter, files from a single git branch/tag will appear in all environments. See *here* for more information on this feature.

Adding a Formula directory manually

Formulas are simply directories that can be copied onto the local file system by using Git to clone the repository or by downloading and expanding a tarball or zip file of the repository. The directory structure is designed to work with `file_roots` in the Salt master configuration.

1. Clone or download the repository into a directory:

   ```
   mkdir -p /srv/formulas
   cd /srv/formulas
   git clone https://github.com/saltstack-formulas/apache-formula.git

   # or

   mkdir -p /srv/formulas
   ```

(continues on next page)

(continued from previous page)

```
cd /srv/formulas
wget https://github.com/saltstack-formulas/apache-formula/archive/master.tar.gz
tar xf apache-formula-master.tar.gz
```

2. Add the new directory to *`file_roots`*:

```
file_roots:
  base:
    - /srv/salt
    - /srv/formulas/apache-formula
```

3. Restart the Salt Master.

Usage

Each Formula is intended to be immediately usable with sane defaults without any additional configuration. Many formulas are also configurable by including data in Pillar; see the `pillar.example` file in each Formula repository for available options.

Including a Formula in an existing State tree

Formula may be included in an existing `sls` file. This is often useful when a state you are writing needs to `require` or `extend` a state defined in the formula.

Here is an example of a state that uses the epel-formula in a `require` declaration which directs Salt to not install the `python26` package until after the EPEL repository has also been installed:

```
include:
  - epel

python26:
  pkg.installed:
    - require:
      - pkg: epel
```

Including a Formula from a Top File

Some Formula perform completely standalone installations that are not referenced from other state files. It is usually cleanest to include these Formula directly from a Top File.

For example the easiest way to set up an OpenStack deployment on a single machine is to include the openstack-standalone-formula directly from a `top.sls` file:

```
base:
  'myopenstackmaster':
    - openstack
```

Quickly deploying OpenStack across several dedicated machines could also be done directly from a Top File and may look something like this:

```
base:
  'controller':
    - openstack.horizon
    - openstack.keystone
  'hyper-*':
    - openstack.nova
    - openstack.glance
  'storage-*':
    - openstack.swift
```

Configuring Formula using Pillar

Salt Formulas are designed to work out of the box with no additional configuration. However, many Formula support additional configuration and customization through *Pillar*. Examples of available options can be found in a file named `pillar.example` in the root directory of each Formula repository.

Using Formula with your own states

Remember that Formula are regular Salt States and can be used with all Salt's normal state mechanisms. Formula can be required from other States with *require* declarations, they can be modified using `extend`, they can made to watch other states with *The _in versions of requisites*.

The following example uses the stock apache-formula alongside a custom state to create a vhost on a Debian/Ubuntu system and to reload the Apache service whenever the vhost is changed.

```
# Include the stock, upstream apache formula.
include:
  - apache

# Use the watch_in requisite to cause the apache service state to reload
# apache whenever the my-example-com-vhost state changes.
my-example-com-vhost:
  file:
    - managed
    - name: /etc/apache2/sites-available/my-example-com
    - watch_in:
      - service: apache
```

Don't be shy to read through the source for each Formula!

Reporting problems & making additions

Each Formula is a separate repository on GitHub. If you encounter a bug with a Formula please file an issue in the respective repository! Send fixes and additions as a pull request. Add tips and tricks to the repository wiki.

Writing Formulas

Each Formula is a separate repository in the saltstack-formulas organization on GitHub.

Get involved creating new Formulas

The best way to create new Formula repositories for now is to create a repository in your own account on GitHub and notify a SaltStack employee when it is ready. We will add you to the Contributors team on the saltstack-formulas organization and help you transfer the repository over. Ping a SaltStack employee on IRC (`#salt` on Freenode), join the `#formulas` channel on the salt-slack or send an email to the salt-users mailing list.

There are a lot of repositories in that organization! Team members can manage which repositories they are subscribed to on GitHub's watching page: https://github.com/watching.

Members of the Contributors team are welcome to participate in reviewing pull requests across the Organization. Some repositories will have regular contributors and some repositories will not. As you get involved in a repository be sure to communicate with any other contributors there on pull requests that are large or have breaking changes.

In general it is best to have another Contributor review and merge any pull requests that you open. Feel free to at-mention other regular contributors to a repository and request a review. However, there are a lot of formula repositories so if a repository does not yet have regular contributors or if your pull request has stayed open for more than a couple days feel free to "selfie-merge" your own pull request.

Style

Maintainability, readability, and reusability are all marks of a good Salt sls file. This section contains several suggestions and examples.

```
# Deploy the stable master branch unless version overridden by passing
# Pillar at the CLI or via the Reactor.

deploy_myapp:
  git.latest:
    - name: git@github.com/myco/myapp.git
    - version: {{ salt.pillar.get('myapp:version', 'master') }}
```

Use a descriptive State ID

The ID of a state is used as a unique identifier that may be referenced via other states in *requisites*. It must be unique across the whole state tree (*it is a key in a dictionary*, after all).

In addition a state ID should be descriptive and serve as a high-level hint of what it will do, or manage, or change. For example, `deploy_webapp`, or `apache`, or `reload_firewall`.

Use `module.function` notation

So-called "short-declaration" notation is preferred for referencing state modules and state functions. It provides a consistent pattern of `module.function` shared between Salt States, the Reactor, Salt Mine, the Scheduler, as well as with the CLI.

```
# Do
apache:
  pkg.installed:
    - name: httpd

# Don't
apache:
```

(continues on next page)

(continued from previous page)

```
  pkg:
    - installed
    - name: httpd
```

Salt's state compiler will transform "short-decs" into the longer format *when compiling the human-friendly highstate structure into the machine-friendly lowstate structure*.

Specify the `name` parameter

Use a unique and permanent identifier for the state ID and reserve `name` for data with variability.

The *name declaration* is a required parameter for all state functions. The state ID will implicitly be used as `name` if it is not explicitly set in the state.

In many state functions the `name` parameter is used for data that varies such as OS-specific package names, OS-specific file system paths, repository addresses, etc. Any time the ID of a state changes all references to that ID must also be changed. Use a permanent ID when writing a state the first time to future-proof that state and allow for easier refactors down the road.

Comment state files

YAML allows comments at varying indentation levels. It is a good practice to comment state files. Use vertical whitespace to visually separate different concepts or actions.

```
# Start with a high-level description of the current sls file.
# Explain the scope of what it will do or manage.

# Comment individual states as necessary.
update_a_config_file:
  # Provide details on why an unusual choice was made. For example:
  #
  # This template is fetched from a third-party and does not fit our
  # company norm of using Jinja. This must be processed using Mako.
  file.managed:
    - name: /path/to/file.cfg
    - source: salt://path/to/file.cfg.template
    - template: mako

  # Provide a description or explanation that did not fit within the state
  # ID. For example:
  #
  # Update the application's last-deployed timestamp.
  # This is a workaround until Bob configures Jenkins to automate RPM
  # builds of the app.
  cmd.run:
    # FIXME: Joe needs this to run on Windows by next quarter. Switch these
    # from shell commands to Salt's file.managed and file.replace state
    # modules.
    - name: |
        touch /path/to/file_last_updated
        sed -e 's/foo/bar/g' /path/to/file_environment
    - onchanges:
      - file: a_config_file
```

Be careful to use Jinja comments for commenting Jinja code and YAML comments for commenting YAML code.

```
# BAD EXAMPLE
# The Jinja in this YAML comment is still executed!
# {% set apache_is_installed = 'apache' in salt.pkg.list_pkgs() %}

# GOOD EXAMPLE
# The Jinja in this Jinja comment will not be executed.
{# {% set apache_is_installed = 'apache' in salt.pkg.list_pkgs() %} #}
```

Easy on the Jinja!

Jinja templating provides vast flexibility and power when building Salt sls files. It can also create an unmaintainable tangle of logic and data. Speaking broadly, Jinja is best used when kept apart from the states (as much as is possible).

Below are guidelines and examples of how Jinja can be used effectively.

Know the evaluation and execution order

High-level knowledge of how Salt states are compiled and run is useful when writing states.

The default *`renderer`* setting in Salt is Jinja piped to YAML. Each is a separate step. Each step is not aware of the previous or following step. Jinja is not YAML aware, YAML is not Jinja aware; they cannot share variables or interact.

- Whatever the Jinja step produces must be valid YAML.
- Whatever the YAML step produces must be a valid *highstate data structure*. (This is also true of the final step for *any of the alternate renderers* in Salt.)
- Highstate can be thought of as a human-friendly data structure; easy to write and easy to read.
- Salt's state compiler validates the *highstate* and compiles it to low state.
- Low state can be thought of as a machine-friendly data structure. It is a list of dictionaries that each map directly to a function call.
- Salt's state system finally starts and executes on each "chunk" in the low state. Remember that requisites are evaluated at runtime.
- The return for each function call is added to the "running" dictionary which is the final output at the end of the state run.

The full evaluation and execution order:

```
Jinja -> YAML -> Highstate -> low state -> execution
```

Avoid changing the underlying system with Jinja

Avoid calling commands from Jinja that change the underlying system. Commands run via Jinja do not respect Salt's dry-run mode (`test=True`)! This is usually in conflict with the idempotent nature of Salt states unless the command being run is also idempotent.

Inspect the local system

A common use for Jinja in Salt states is to gather information about the underlying system. The `grains` dictionary available in the Jinja context is a great example of common data points that Salt itself has already gathered. Less common values are often found by running commands. For example:

```
{% set is_selinux_enabled = salt.cmd.run('sestatus') == '1' %}
```

This is usually best done with a variable assignment in order to separate the data from the state that will make use of the data.

Gather external data

One of the most common uses for Jinja is to pull external data into the state file. External data can come from anywhere like API calls or database queries, but it most commonly comes from flat files on the file system or Pillar data from the Salt Master. For example:

```
{% set some_data = salt.pillar.get('some_data', {'sane default': True}) %}

{# or #}

{% import_yaml 'path/to/file.yaml' as some_data %}

{# or #}

{% import_json 'path/to/file.json' as some_data %}

{# or #}

{% import_text 'path/to/ssh_key.pub' as ssh_pub_key %}

{# or #}

{% from 'path/to/other_file.jinja' import some_data with context %}
```

This is usually best done with a variable assignment in order to separate the data from the state that will make use of the data.

Light conditionals and looping

Jinja is extremely powerful for programmatically generating Salt states. It is also easy to overuse. As a rule of thumb, if it is hard to read it will be hard to maintain!

Separate Jinja control-flow statements from the states as much as is possible to create readable states. Limit Jinja within states to simple variable lookups.

Below is a simple example of a readable loop:

```
{% for user in salt.pillar.get('list_of_users', []) %}

{# Ensure unique state IDs when looping. #}
{{ user.name }}-{{ loop.index }}:
  user.present:
    - name: {{ user.name }}
```

(continues on next page)

(continued from previous page)

```
    - shell: {{ user.shell }}

{% endfor %}
```

Avoid putting a Jinja conditionals within Salt states where possible. Readability suffers and the correct YAML indentation is difficult to see in the surrounding visual noise. Parametrization (discussed below) and variables are both useful techniques to avoid this. For example:

```
{# ---- Bad example ---- #}

apache:
  pkg.installed:
    {% if grains.os_family == 'RedHat' %}
    - name: httpd
    {% elif grains.os_family == 'Debian' %}
    - name: apache2
    {% endif %}

{# ---- Better example ---- #}

{% if grains.os_family == 'RedHat' %}
{% set name = 'httpd' %}
{% elif grains.os_family == 'Debian' %}
{% set name = 'apache2' %}
{% endif %}

 apache:
  pkg.installed:
    - name: {{ name }}

{# ---- Good example ---- #}

{% set name = {
    'RedHat': 'httpd',
    'Debian': 'apache2',
}.get(grains.os_family) %}

 apache:
  pkg.installed:
    - name: {{ name }}
```

Dictionaries are useful to effectively "namespace" a collection of variables. This is useful with parametrization (discussed below). Dictionaries are also easily combined and merged. And they can be directly serialized into YAML which is often easier than trying to create valid YAML through templating. For example:

```
{# ---- Bad example ---- #}

haproxy_conf:
  file.managed:
    - name: /etc/haproxy/haproxy.cfg
    - template: jinja
    {% if 'external_loadbalancer' in grains.roles %}
    - source: salt://haproxy/external_haproxy.cfg
    {% elif 'internal_loadbalancer' in grains.roles %}
    - source: salt://haproxy/internal_haproxy.cfg
    {% endif %}
```

(continues on next page)

(continued from previous page)

```
    - context:
        {% if 'external_loadbalancer' in grains.roles %}
        ssl_termination: True
        {% elif 'internal_loadbalancer' in grains.roles %}
        ssl_termination: False
        {% endif %}

{# ---- Better example ---- #}

{% load_yaml as haproxy_defaults %}
common_settings:
  bind_port: 80

internal_loadbalancer:
  source: salt://haproxy/internal_haproxy.cfg
  settings:
    bind_port: 8080
    ssl_termination: False

external_loadbalancer:
  source: salt://haproxy/external_haproxy.cfg
  settings:
    ssl_termination: True
{% endload %}

{% if 'external_loadbalancer' in grains.roles %}
{% set haproxy = haproxy_defaults['external_loadbalancer'] %}
{% elif 'internal_loadbalancer' in grains.roles %}
{% set haproxy = haproxy_defaults['internal_loadbalancer'] %}
{% endif %}

{% do haproxy.settings.update(haproxy_defaults.common_settings) %}

haproxy_conf:
  file.managed:
    - name: /etc/haproxy/haproxy.cfg
    - template: jinja
    - source: {{ haproxy.source }}
    - context: {{ haproxy.settings | yaml() }}
```

There is still room for improvement in the above example. For example, extracting into an external file or replacing the if-elif conditional with a function call to filter the correct data more succinctly. However, the state itself is simple and legible, the data is separate and also simple and legible. And those suggested improvements can be made at some future date without altering the state at all!

Avoid heavy logic and programming

Jinja is not Python. It was made by Python programmers and shares many semantics and some syntax but it does not allow for abitrary Python function calls or Python imports. Jinja is a fast and efficient templating language but the syntax can be verbose and visually noisy.

Once Jinja use within an sls file becomes slightly complicated – long chains of if-elif-elif-else statements, nested conditionals, complicated dictionary merges, wanting to use sets – instead consider using a different Salt renderer, such as the Python renderer. As a rule of thumb, if it is hard to read it will be hard to maintain – switch to a format that is easier to read.

Using alternate renderers is very simple to do using Salt's "she-bang" syntax at the top of the file. The Python renderer must simply return the correct *highstate data structure*. The following example is a state tree of two sls files, one simple and one complicated.

`/srv/salt/top.sls`:

```
base:
  '*':
    - common_configuration
    - roles_configuration
```

`/srv/salt/common_configuration.sls`:

```
common_users:
  user.present:
    - names:
      - larry
      - curly
      - moe
```

`/srv/salt/roles_configuration`:

```
#!py
def run():
    list_of_roles = set()

    # This example has the minion id in the form 'web-03-dev'.
    # Easily access the grains dictionary:
    try:
        app, instance_number, environment = __grains__['id'].split('-')
        instance_number = int(instance_number)
    except ValueError:
        app, instance_number, environment = ['Unknown', 0, 'dev']

    list_of_roles.add(app)

    if app == 'web' and environment == 'dev':
        list_of_roles.add('primary')
        list_of_roles.add('secondary')
    elif app == 'web' and environment == 'staging':
        if instance_number == 0:
            list_of_roles.add('primary')
        else:
            list_of_roles.add('secondary')

    # Easily cross-call Salt execution modules:
    if __salt__['myutils.query_valid_ec2_instance']():
        list_of_roles.add('is_ec2_instance')

    return {
        'set_roles_grains': {
            'grains.present': [
                {'name': 'roles'},
                {'value': list(list_of_roles)},
            ],
        },
    }
```

Jinja Macros

In Salt sls files Jinja macros are useful for one thing and one thing only: creating mini templates that can be reused and rendered on demand. Do not fall into the trap of thinking of macros as functions; Jinja is not Python (see above).

Macros are useful for creating reusable, parameterized states. For example:

```
{% macro user_state(state_id, user_name, shell='/bin/bash', groups=[]) %}
{{ state_id }}:
  user.present:
    - name: {{ user_name }}
    - shell: {{ shell }}
    - groups: {{ groups | json() }}
{% endmacro %}

{% for user_info in salt.pillar.get('my_users', []) %}
{{ user_state('user_number_' ~ loop.index, **user_info) }}
{% endfor %}
```

Macros are also useful for creating one-off "serializers" that can accept a data structure and write that out as a domain-specific configuration file. For example, the following macro could be used to write a php.ini config file:

`/srv/salt/php.sls`:

```
php_ini:
  file.managed:
    - name: /etc/php.ini
    - source: salt://php.ini.tmpl
    - template: jinja
    - context:
        php_ini_settings: {{ salt.pillar.get('php_ini', {}) | json() }}
```

`/srv/pillar/php.sls`:

```
php_ini:
  PHP:
    engine: 'On'
    short_open_tag: 'Off'
    error_reporting: 'E_ALL & ~E_DEPRECATED & ~E_STRICT'
```

`/srv/salt/php.ini.tmpl`:

```
{% macro php_ini_serializer(data) %}
{% for section_name, name_val_pairs in data.items() %}
[{{ section_name }}]
{% for name, val in name_val_pairs.items() -%}
{{ name }} = "{{ val }}"
{% endfor %}
{% endfor %}
{% endmacro %}

; File managed by Salt at <{{ source }}>.
; Your changes will be overwritten.

{{ php_ini_serializer(php_ini_settings) }}
```

Abstracting static defaults into a lookup table

Separate data that a state uses from the state itself to increases the flexibility and reusability of a state.

An obvious and common example of this is platform-specific package names and file system paths. Another example is sane defaults for an application, or common settings within a company or organization. Organizing such data as a dictionary (aka hash map, lookup table, associative array) often provides a lightweight namespacing and allows for quick and easy lookups. In addition, using a dictionary allows for easily merging and overriding static values within a lookup table with dynamic values fetched from Pillar.

A strong convention in Salt Formulas is to place platform-specific data, such as package names and file system paths, into a file named `map.jinja` that is placed alongside the state files.

The following is an example from the MySQL Formula. The *`grains.filter_by`* function performs a lookup on that table using the `os_family` grain (by default).

The result is that the `mysql` variable is assigned to a *subset* of the lookup table for the current platform. This allows states to reference, for example, the name of a package without worrying about the underlying OS. The syntax for referencing a value is a normal dictionary lookup in Jinja, such as `{{ mysql['service'] }}` or the shorthand `{{ mysql.service }}`.

`map.jinja`:

```
{% set mysql = salt['grains.filter_by']({
    'Debian': {
        'server': 'mysql-server',
        'client': 'mysql-client',
        'service': 'mysql',
        'config': '/etc/mysql/my.cnf',
        'python': 'python-mysqldb',
    },
    'RedHat': {
        'server': 'mysql-server',
        'client': 'mysql',
        'service': 'mysqld',
        'config': '/etc/my.cnf',
        'python': 'MySQL-python',
    },
    'Gentoo': {
        'server': 'dev-db/mysql',
        'client': 'dev-db/mysql',
        'service': 'mysql',
        'config': '/etc/mysql/my.cnf',
        'python': 'dev-python/mysql-python',
    },
}, merge=salt['pillar.get']('mysql:lookup')) %}
```

Values defined in the map file can be fetched for the current platform in any state file using the following syntax:

```
{% from "mysql/map.jinja" import mysql with context %}

mysql-server:
  pkg.installed:
    - name: {{ mysql.server }}
  service.running:
    - name: {{ mysql.service }}
```

Organizing Pillar data

It is considered a best practice to make formulas expect **all** formula-related parameters to be placed under second-level `lookup` key, within a main namespace designated for holding data for particular service/software/etc, managed by the formula:

```
mysql:
  lookup:
    version: 5.7.11
```

Collecting common values

Common values can be collected into a *base* dictionary. This minimizes repetition of identical values in each of the `lookup_dict` sub-dictionaries. Now only the values that are different from the base must be specified by the alternates:

`map.jinja`:

```
{% set mysql = salt['grains.filter_by']({
    'default': {
        'server': 'mysql-server',
        'client': 'mysql-client',
        'service': 'mysql',
        'config': '/etc/mysql/my.cnf',
        'python': 'python-mysqldb',
    },
    'Debian': {
    },
    'RedHat': {
        'client': 'mysql',
        'service': 'mysqld',
        'config': '/etc/my.cnf',
        'python': 'MySQL-python',
    },
    'Gentoo': {
        'server': 'dev-db/mysql',
        'client': 'dev-db/mysql',
        'python': 'dev-python/mysql-python',
    },
},
merge=salt['pillar.get']('mysql:lookup'), base='default') %}
```

Overriding values in the lookup table

Allow static values within lookup tables to be overridden. This is a simple pattern which once again increases flexibility and reusability for state files.

The `merge` argument in *`filter_by`* specifies the location of a dictionary in Pillar that can be used to override values returned from the lookup table. If the value exists in Pillar it will take precedence.

This is useful when software or configuration files is installed to non-standard locations or on unsupported platforms. For example, the following Pillar would replace the `config` value from the call above.

```
mysql:
  lookup:
    config: /usr/local/etc/mysql/my.cnf
```

Note: Protecting Expansion of Content with Special Characters

When templating keep in mind that YAML does have special characters for quoting, flows, and other special structure and content. When a Jinja substitution may have special characters that will be incorrectly parsed by YAML care must be taken. It is a good policy to use the `yaml_encode` or the `yaml_dquote` Jinja filters:

```
{%- set foo = 7.7 %}
{%- set bar = none %}
{%- set baz = true %}
{%- set zap = 'The word of the day is "salty".' %}
{%- set zip = '"The quick brown fox . . ."' %}

foo: {{ foo|yaml_encode }}
bar: {{ bar|yaml_encode }}
baz: {{ baz|yaml_encode }}
zap: {{ zap|yaml_encode }}
zip: {{ zip|yaml_dquote }}
```

The above will be rendered as below:

```
foo: 7.7
bar: null
baz: true
zap: "The word of the day is \"salty\"."
zip: "\"The quick brown fox . . .\""
```

The *`filter_by`* function performs a simple dictionary lookup but also allows for fetching data from Pillar and overriding data stored in the lookup table. That same workflow can be easily performed without using `filter_by`; other dictionaries besides data from Pillar can also be used.

```
{% set lookup_table = {...} %}
{% do lookup_table.update(salt.pillar.get('my:custom:data')) %}
```

When to use lookup tables

The `map.jinja` file is only a convention within Salt Formulas. This greater pattern is useful for a wide variety of data in a wide variety of workflows. This pattern is not limited to pulling data from a single file or data source. This pattern is useful in States, Pillar and the Reactor, for example.

Working with a data structure instead of, say, a config file allows the data to be cobbled together from multiple sources (local files, remote Pillar, database queries, etc), combined, overridden, and searched.

Below are a few examples of what lookup tables may be useful for and how they may be used and represented.

Platform-specific information

An obvious pattern and one used heavily in Salt Formulas is extracting platform-specific information such as package names and file system paths in a file named `map.jinja`. The pattern is explained in detail above.

Sane defaults

Application settings can be a good fit for this pattern. Store default settings along with the states themselves and keep overrides and sensitive settings in Pillar. Combine both into a single dictionary and then write the application config or settings file.

The example below stores most of the Apache Tomcat `server.xml` file alongside the Tomcat states and then allows values to be updated or augmented via Pillar. (This example uses the BadgerFish format for transforming JSON to XML.)

`/srv/salt/tomcat/defaults.yaml`:

```
Server:
  '@port': '8005'
  '@shutdown': SHUTDOWN
  GlobalNamingResources:
    Resource:
      '@auth': Container
      '@description': User database that can be updated and saved
      '@factory': org.apache.catalina.users.MemoryUserDatabaseFactory
      '@name': UserDatabase
      '@pathname': conf/tomcat-users.xml
      '@type': org.apache.catalina.UserDatabase
  # <...snip...>
```

`/srv/pillar/tomcat.sls`:

```
appX:
  server_xml_overrides:
    Server:
      Service:
        '@name': Catalina
        Connector:
          '@port': '8009'
          '@protocol': AJP/1.3
          '@redirectPort': '8443'
          # <...snip...>
```

`/srv/salt/tomcat/server_xml.sls`:

```
{% import_yaml 'tomcat/defaults.yaml' as server_xml_defaults %}
{% set server_xml_final_values = salt.pillar.get(
    'appX:server_xml_overrides',
    default=server_xml_defaults,
    merge=True)
%}

appX_server_xml:
  file.serialize:
    - name: /etc/tomcat/server.xml
    - dataset: {{ server_xml_final_values | json() }}
    - formatter: xml_badgerfish
```

The *`file.serialize`* state can provide a shorthand for creating some files from data structures. There are also many examples within Salt Formulas of creating one-off "serializers" (often as Jinja macros) that reformat a data structure to a specific config file format. For example, look at the'Nginx vhosts'_ states or the php.ini file template.

Environment specific information

A single state can be reused when it is parameterized as described in the section below, by separating the data the state will use from the state that performs the work. This can be the difference between deploying *Application X* and *Application Y*, or the difference between production and development. For example:

`/srv/salt/app/deploy.sls`:

```
{# Load the map file. #}
{% import_yaml 'app/defaults.yaml' as app_defaults %}

{# Extract the relevant subset for the app configured on the current
   machine (configured via a grain in this example). #}
{% app = app_defaults.get(salt.grains.get('role')) %}

{# Allow values from Pillar to (optionally) update values from the lookup
   table. #}
{% do app_defaults.update(salt.pillar.get('myapp', {})) %}

deploy_application:
  git.latest:
    - name: {{ app.repo_url }}
    - version: {{ app.version }}
    - target: {{ app.deploy_dir }}

myco/myapp/deployed:
  event.send:
    - data:
        version: {{ app.version }}
    - onchanges:
      - git: deploy_application
```

`/srv/salt/app/defaults.yaml`:

```
appX:
  repo_url: git@github.com/myco/appX.git
  target: /var/www/appX
  version: master
appY:
  repo_url: git@github.com/myco/appY.git
  target: /var/www/appY
  version: v1.2.3.4
```

Single-purpose SLS files

Each sls file in a Formula should strive to do a single thing. This increases the reusability of this file by keeping unrelated tasks from getting coupled together.

As an example, the base Apache formula should only install the Apache httpd server and start the httpd service. This is the basic, expected behavior when installing Apache. It should not perform additional changes such as set the Apache configuration file or create vhosts.

If a formula is single-purpose as in the example above, other formulas, and also other states can `include` and use that formula with *Requisites and Other Global State Arguments* without also including undesirable or unintended side-effects.

The following is a best-practice example for a reusable Apache formula. (This skips platform-specific options for brevity. See the full apache-formula for more.)

```
# apache/init.sls
apache:
  pkg.installed:
    [...]
  service.running:
    [...]

# apache/mod_wsgi.sls
include:
  - apache

mod_wsgi:
  pkg.installed:
    [...]
    - require:
      - pkg: apache

# apache/conf.sls
include:
  - apache

apache_conf:
  file.managed:
    [...]
    - watch_in:
      - service: apache
```

To illustrate a bad example, say the above Apache formula installed Apache and also created a default vhost. The mod_wsgi state would not be able to include the Apache formula to create that dependency tree without also installing the unneeded default vhost.

Formulas should be reusable. Avoid coupling unrelated actions together.

Parameterization

Parameterization is a key feature of Salt Formulas and also for Salt States. Parameterization allows a single Formula to be reused across many operating systems; to be reused across production, development, or staging environments; and to be reused by many people all with varying goals.

Writing states, specifying ordering and dependencies is the part that takes the longest to write and to test. Filling those states out with data such as users or package names or file locations is the easy part. How many users, what those users are named, or where the files live are all implementation details that **should be parameterized**. This separation between a state and the data that populates a state creates a reusable formula.

In the example below the data that populates the state can come from anywhere – it can be hard-coded at the top of the state, it can come from an external file, it can come from Pillar, it can come from an execution function call, or it can come from a database query. The state itself doesn't change regardless of where the data comes from. Production data will vary from development data will vary from data from one company to another, however the state itself stays the same.

```
{% set user_list = [
    {'name': 'larry', 'shell': 'bash'},
    {'name': 'curly', 'shell': 'bash'},
```

(continues on next page)

(continued from previous page)

```
    {'name': 'moe', 'shell': 'zsh'},
] %}

{# or #}

{% set user_list = salt['pillar.get']('user_list') %}

{# or #}

{% load_json "default_users.json" as user_list %}

{# or #}

{% set user_list = salt['acme_utils.get_user_list']() %}

{% for user in list_list %}
{{ user.name }}:
  user.present:
    - name: {{ user.name }}
    - shell: {{ user.shell }}
{% endfor %}
```

Configuration

Formulas should strive to use the defaults of the underlying platform, followed by defaults from the upstream project, followed by sane defaults for the formula itself.

As an example, a formula to install Apache **should not** change the default Apache configuration file installed by the OS package. However, the Apache formula **should** include a state to change or override the default configuration file.

Pillar overrides

Pillar lookups must use the safe `get()` and must provide a default value. Create local variables using the Jinja `set` construct to increase readability and to avoid potentially hundreds or thousands of function calls across a large state tree.

```
{% from "apache/map.jinja" import apache with context %}
{% set settings = salt['pillar.get']('apache', {}) %}

mod_status:
  file.managed:
    - name: {{ apache.conf_dir }}
    - source: {{ settings.get('mod_status_conf', 'salt://apache/mod_status.conf') }}
    - template: {{ settings.get('template_engine', 'jinja') }}
```

Any default values used in the Formula must also be documented in the `pillar.example` file in the root of the repository. Comments should be used liberally to explain the intent of each configuration value. In addition, users should be able copy-and-paste the contents of this file into their own Pillar to make any desired changes.

Scripting

Remember that both State files and Pillar files can easily call out to Salt *execution modules* and have access to all the system grains as well.

```
{% if '/storage' in salt['mount.active']() %}
/usr/local/etc/myfile.conf:
  file:
    - symlink
    - target: /storage/myfile.conf
{% endif %}
```

Jinja macros to encapsulate logic or conditionals are discouraged in favor of *writing custom execution modules* in Python.

Repository structure

A basic Formula repository should have the following layout:

```
foo-formula
|-- foo/
|   |-- map.jinja
|   |-- init.sls
|   `-- bar.sls
|-- CHANGELOG.rst
|-- LICENSE
|-- pillar.example
|-- README.rst
`-- VERSION
```

See also:

template-formula

The template-formula repository has a pre-built layout that serves as the basic structure for a new formula repository. Just copy the files from there and edit them.

`README.rst`

The README should detail each available `.sls` file by explaining what it does, whether it has any dependencies on other formulas, whether it has a target platform, and any other installation or usage instructions or tips.

A sample skeleton for the `README.rst` file:

```
===
foo
===

Install and configure the FOO service.

**NOTE**

See the full `Salt Formulas installation and usage instructions
<https://docs.saltstack.com/en/latest/topics/development/conventions/formulas.html>`_.
```

(continues on next page)

(continued from previous page)

```
Available states
================

.. contents::
    :local:

``foo``
-------

Install the ``foo`` package and enable the service.

``foo.bar``
-----------

Install the ``bar`` package.
```

`CHANGELOG.rst`

The `CHANGELOG.rst` file should detail the individual versions, their release date and a set of bullet points for each version highlighting the overall changes in a given version of the formula.

A sample skeleton for the *CHANGELOG.rst* file:

`CHANGELOG.rst`:

```
foo formula
===========

0.0.2 (2013-01-01)

- Re-organized formula file layout
- Fixed filename used for upstart logger template
- Allow for pillar message to have default if none specified
```

Versioning

Formula are versioned according to Semantic Versioning, http://semver.org/.

Note: Given a version number MAJOR.MINOR.PATCH, increment the:

1. MAJOR version when you make incompatible API changes,
2. MINOR version when you add functionality in a backwards-compatible manner, and
3. PATCH version when you make backwards-compatible bug fixes.

Additional labels for pre-release and build metadata are available as extensions to the MAJOR.MINOR.PATCH format.

Formula versions are tracked using Git tags as well as the `VERSION` file in the formula repository. The `VERSION` file should contain the currently released version of the particular formula.

Testing Formulas

A smoke-test for invalid Jinja, invalid YAML, or an invalid Salt state structure can be performed by with the `state.show_sls` function:

```
salt '*' state.show_sls apache
```

Salt Formulas can then be tested by running each `.sls` file via `state.apply` and checking the output for the success or failure of each state in the Formula. This should be done for each supported platform.

27.22.3 SaltStack Packaging Guide

Since Salt provides a powerful toolkit for system management and automation, the package can be spit into a number of sub-tools. While packaging Salt as a single package containing all components is perfectly acceptable, the split packages should follow this convention.

Patching Salt For Distributions

The occasion may arise where Salt source and default configurations may need to be patched. It is preferable if Salt is only patched to include platform specific additions or to fix release time bugs. It is preferable that configuration settings and operations remain in the default state, as changes here lowers the user experience for users moving across distributions.

In the event where a packager finds a need to change the default configuration it is advised to add the files to the master.d or minion.d directories.

Source Files

Release packages should always be built from the source tarball distributed via pypi. Release packages should *NEVER* use a git checkout as the source for distribution.

Single Package

Shipping Salt as a single package, where the minion, master, and all tools are together is perfectly acceptable and practiced by distributions such as FreeBSD.

Split Package

Salt Should always be split in a standard way, with standard dependencies, this lowers cross distribution confusion about what components are going to be shipped with specific packages. These packages can be defined from the Salt Source as of Salt 2014.1.0:

Salt Common

The *salt-common* or *salt* package should contain the files provided by the salt python package, or all files distributed from the `salt/` directory in the source distribution packages. The documentation contained under the `doc/` directory can be a part of this package but splitting out a doc package is preferred. Since salt-call is the entry point to utilize the libs and is useful for all salt packages it is included in the salt-common package.

Name

- *salt* OR *salt-common*

Files

- *salt/**
- *man/salt.7*
- *scripts/salt-call*
- *tests/**
- *man/salt-call.1*

Depends

- *Python 2.6-2.7*
- *PyYAML*
- *Jinja2*

Salt Master

The *salt-master* package contains the applicable scripts, related man pages and init information for the given platform.

Name

- *salt-master*

Files

- *scripts/salt-master*
- *scripts/salt*
- *scripts/salt-run*
- *scripts/salt-key*
- *scripts/salt-cp*
- *pkg/<master init data>*
- *man/salt.1*
- *man/salt-master.1*
- *man/salt-run.1*
- *man/salt-key.1*
- *man/salt-cp.1*
- *conf/master*

Depends

- *Salt Common*
- *ZeroMQ* >= 3.2
- *PyZMQ* >= 2.10
- *PyCrypto*
- *M2Crypto*
- *Python MessagePack* (Messagepack C lib, or msgpack-pure)

Salt Syndic

The Salt Syndic package can be rolled completely into the Salt Master package. Platforms which start services as part of the package deployment need to maintain a separate *salt-syndic* package (primarily Debian based platforms).

The Syndic may optionally not depend on the anything more than the Salt Master since the master will bring in all needed dependencies, but fall back to the platform specific packaging guidelines.

Name

- *salt-syndic*

Files

- *scripts/salt-syndic*
- *pkg/<syndic init data>*
- *man/salt-syndic.1*

Depends

- *Salt Common*
- *Salt Master*
- *ZeroMQ* >= 3.2
- *PyZMQ* >= 2.10
- *PyCrypto*
- *M2Crypto*
- *Python MessagePack* (Messagepack C lib, or msgpack-pure)

Salt Minion

The Minion is a standalone package and should not be split beyond the *salt-minion* and *salt-common* packages.

Name

- *salt-minion*

Files

- *scripts/salt-minion*
- *pkg/<minion init data>*
- *man/salt-minion.1*
- *conf/minion*

Depends

- *Salt Common*
- *ZeroMQ* >= 3.2
- *PyZMQ* >= 2.10
- *PyCrypto*
- *M2Crypto*
- *Python MessagePack* (Messagepack C lib, or msgpack-pure)

Salt SSH

Since Salt SSH does not require the same dependencies as the minion and master, it should be split out.

Name

- *salt-ssh*

Files

- *scripts/salt-ssh*
- *man/salt-ssh.1*
- *conf/cloud**

Depends

- *Salt Common*
- *Python MessagePack* (Messagepack C lib, or msgpack-pure)

Salt Cloud

As of Salt 2014.1.0 Salt Cloud is included in the same repo as Salt. This can be split out into a separate package or it can be included in the salt-master package.

Name

- *salt-cloud*

Files

- *scripts/salt-cloud*
- *man/salt-cloud.1*

Depends

- *Salt Common*
- *apache libcloud* >= 0.14.0

Salt Doc

The documentation package is very distribution optional. A completely split package will split out the documentation, but some platform conventions do not prefer this. If the documentation is not split out, it should be included with the *Salt Common* package.

Name

- *salt-doc*

Files

- *doc/**

Optional Depends

- *Salt Common*
- *Python Sphinx*
- *Make*

27.22.4 Salt Release Process

The goal for Salt projects is to cut a new feature release every six months. This document outlines the process for these releases, and the subsequent bug fix releases which follow.

Feature Release Process

When a new release is ready to be cut, the person responsible for cutting the release will follow the following steps (written using the 2019.2.0 release as an example):

1. Create first public draft of release notes with major features.
2. Remove any deprecations for the upcoming release.
3. Notify salt-users and salt-announce google groups when the feature freeze branch creation will occur so they can try to get their work merged.
4. Create QA test plan. Review features planned for the release and determine if there is sufficient test coverage.
5. Ensure all required features are merged.
6. Complete one last merge forward from the previous branch.
7. Create feature release branch with the name of the release. (ex. fluorine)
8. Create issue to start the process of deprecating for the next feature release.
9. Create jenkins jobs to test the new feature release branch.
10. Inform salt-users and salt-announce google groups feature branch and freeze is complete.
11. Add new feature branch to salt-jenkins repo and the kitchen yaml file.
12. Fix tests failing in jenkins test runs.
13. Finalize QA test plan and add all required tests.
14. Run through a manual test run based off of the head of the feature branch.
15. Convert the feature release branch to the version number. For example (v2019.2) This is based off of the year and month that is planned to release.
16. Migrate both the jenkins test jobs and salt-jenkins repo to the new branch number.
17. Notify salt-users and salt-announce google groups of the new version branch number and migrate any PRs to the new branch.
18. Delete old feature release branch name (ex. fluorine)
19. Update all name references to version number in the docs. For example all fluorine references in the docs needs to be moved to v2019.2.0
20. Create RC release branch. (ex. 2019.2.0.rc)
21. Create new jenkins test jobs with new RC release branch
22. Notify salt-users and salt-announce google groups of the new RC branch.
23. Fix tests failing in jenkins test runs.
24. Review the release notes with major features.
25. Generate the new man pages for the release.
26. Create internal RC tag for testing.
27. Build latest windows, mac, ubuntu, debian and redhat packages.
28. Run manual and package tests against new RC packages.
29. Update release candidate docs with the new version. (ex. 2019.2.0rc1)
30. Push the internal tag live to salt's repo.

31. Publish release archive to pypi based off tag.
32. Push the RC packages live.
33. Announce new RC to salt-users and salt-announce google groups.
34. Triage incoming issues based on the new RC release.
35. Fix RC issues once they are categorized as a release blocker.
36. Depending on the issues found during the RC process make a decesion on whether to release based off the RC or go through another RC process, repeating the steps starting at ensuring the tests are not failing.
37. If a RC is categorized as stable, build all required packages.
38. Test all release packages.
39. Test links from repo.saltstack.com.
40. Update installation instructions with new release number at repo.saltstack.com.
41. Update and build docs to include new version (2019.2) as the latest.
42. Pre-announce on salt-users google group that we are about to update our repo.
43. Publish release (v2019.2.0) archive to pypi based off tag.
44. Publish all packages live to repo.
45. Publish the docs.
46. Create release at github
47. Update win-repo-ng with new salt versions.
48. Announce release is live to irc, salt-users, salt-announce and release slack community channel.

Maintenance and Bugfix Releases

Once a feature release branch has been cut from `develop`, the branch moves into a "feature freeze" state. The new release branch enters the `merge-forward` chain and only bugfixes should be applied against the new branch. Once major bugs have been fixed, a bugfix release can be cut:

1. Ensure all required bug fixes are merged.
2. Inform salt-users and salt-announce we are going to branch for the release.
3. Complete one last merge forward from the previous branch.
4. Create release branch with the version of the release. (ex. 2019.2.1)
5. Create jenkins jobs that test the new release branch.
6. Fix tests failing in jeknins test runs.
7. Run through a manual test run based off of the head of the branch.
8. Generate the new man pages for the release.
9. Create internal tag for testing.(ex v2019.2.1)
10. Build all release packages.
11. Run manual and package tests against new packages.
12. Update installation instructions with new release number at repo.saltstack.com.
13. Update and build docs to include new version. (ex. 2019.2.1)

14. Pre-announce on salt-users google groups that we are about to update our repo.
15. Push the internal tag live to salt's repo.
16. Publish release archive to pypi based off tag.
17. Push the packages live.
18. Publish release (v2019.2.1) archive to pypi based off tag.
19. Publish all packages live to repo.
20. Publish the docs.
21. Create release at github
22. Update win-repo-ng with new salt versions.
23. Announce release is live to irc, salt-users, salt-announce and release slack channel.

For more information about the difference between the `develop` branch and bugfix release branches, please refer to the *Which Salt Branch?* section of Salt's *Contributing* documentation.

27.22.5 Salt Coding Style

To make it easier to contribute and read Salt code, SaltStack has adopted Black as its code formatter. There are a few places where Black is silent, and this guide should be used in those cases.

Coding style is NEVER grounds to reject code contributions, and is never grounds to talk down to another member of the community (There are no grounds to treat others without respect, especially people working to improve Salt)!

Linting

Most Salt style conventions are codified in Salt's `.testing.pylintrc` file. Salt's pylint file has two dependencies: pylint and saltpylint. You can install these dependencies with `pip`:

```
pip install pylint
pip install saltpylint
```

The `.testing.pylintrc` file is found in the root of the Salt project and can be passed as an argument to the pylint program as follows:

```
pylint --rcfile=/path/to/salt/.testing.pylintrc salt/dir/to/lint
```

Note: There are two pylint files in the `salt` directory. One is the `.pylintrc` file and the other is the `.testing.pylintrc` file. The tests that run in Jenkins against GitHub Pull Requests use `.testing.pylintrc`. The `testing.pylintrc` file is a little less strict than the `.pylintrc` and is used to make it easier for contributors to submit changes. The `.pylintrc` file can be used for linting, but the `testing.pylintrc` is the source of truth when submitting pull requests.

Variables

Variables should be a minimum of three characters and should provide an easy-to-understand name of the object being represented.

When keys and values are iterated over, descriptive names should be used to represent the temporary variables.

Multi-word variables should be separated by an underscore.

Variables which are two-letter words should have an underscore appended to them to pad them to three characters.

Formatting Strings

All strings which require formatting should use the *.format* string method:

```
data = 'some text'
more = '{0} and then some'.format(data)
```

Make sure to use indices or identifiers in the format brackets, since empty brackets are not supported by python 2.6.

Please do NOT use printf formatting.

Docstring Conventions

When adding a new function or state, where possible try to use a `versionadded` directive to denote when the function, state, or parameter was added.

```
def new_func(msg=''):
    '''
    .. versionadded:: 0.16.0

    Prints what was passed to the function.

    msg : None
        The string to be printed.
    '''
    print(msg)
```

If you are uncertain what version should be used, either consult a core developer in IRC or bring this up when opening your *pull request* and a core developer will let you know what version to add. Typically this will be the next element in the periodic table.

Similar to the above, when an existing function or state is modified (for example, when an argument is added), then under the explanation of that new argument a `versionadded` directive should be used to note the version in which the new argument was added. If an argument's function changes significantly, the `versionchanged` directive can be used to clarify this:

```
def new_func(msg='', signature=''):
    '''
    .. versionadded:: 0.16.0

    Prints what was passed to the function.

    msg : None
        The string to be printed. Will be prepended with 'Greetings! '.

    .. versionchanged:: 0.17.1

    signature : None
        An optional signature.

    .. versionadded 0.17.0
```

(continues on next page)

(continued from previous page)

```
    '''
    print('Greetings! {0}\n\n{1}'.format(msg, signature))
```

Dictionaries

Dictionaries should be initialized using *{}* instead of *dict()*.

See here for an in-depth discussion of this topic.

Imports

Salt code prefers importing modules and not explicit functions. This is both a style and functional preference. The functional preference originates around the fact that the module import system used by pluggable modules will include callable objects (functions) that exist in the direct module namespace. This is not only messy, but may unintentionally expose code python libs to the Salt interface and pose a security problem.

To say this more directly with an example, this is *GOOD*:

```
import os

def minion_path():
    path = os.path.join(self.opts['cachedir'], 'minions')
    return path
```

This on the other hand is *DISCOURAGED*:

```
from os.path import join

def minion_path():
    path = join(self.opts['cachedir'], 'minions')
    return path
```

The time when this is changed is for importing exceptions, generally directly importing exceptions is preferred:

This is a good way to import exceptions:

```
from salt.exceptions import CommandExecutionError
```

Absolute Imports

Although absolute imports seems like an awesome idea, please do not use it. Extra care would be necessary all over salt's code in order for absolute imports to work as supposed. Believe it, it has been tried before and, as a tried example, by renaming `salt.modules.sysmod` to `salt.modules.sys`, all other salt modules which needed to import `sys` would have to also import `absolute_import`, which should be avoided.

Note: An exception to this rule is the `absolute_import` from `__future__` at the top of each file within the Salt project. This import is necessary for Py3 compatibility. This particular import looks like this:

```
from __future__ import absolute_import
```

This import is required for all new Salt files and is a good idea to add to any custom states or modules. However, the practice of avoiding absolute imports still applies to all other cases as to avoid a name conflict.

Code Churn

Many pull requests have been submitted that only churn code in the name of PEP 8. Code churn is a leading source of bugs and is **strongly discouraged**. While style fixes are encouraged they should be isolated to a single file per commit, and the changes should be legitimate, if there are any questions about whether a style change is legitimate please reference this document and the official PEP 8 (http://legacy.python.org/dev/peps/pep-0008/) document before changing code. Many claims that a change is PEP 8 have been invalid, please double check before committing fixes.

27.23 Salt code and internals

Reference documentation on Salt's internal code.

27.23.1 Contents

salt.aggregation

salt.utils.aggregation

This library makes it possible to introspect dataset and aggregate nodes when it is instructed.

Note: The following examples with be expressed in YAML for convenience's sake:

- !aggr-scalar will refer to Scalar python function
- !aggr-map will refer to Map python object
- !aggr-seq will refer for Sequence python object

How to instructs merging

This yaml document has duplicate keys:

```
foo: !aggr-scalar first
foo: !aggr-scalar second
bar: !aggr-map {first: foo}
bar: !aggr-map {second: bar}
baz: !aggr-scalar 42
```

but tagged values instruct Salt that overlapping values they can be merged together:

```
foo: !aggr-seq [first, second]
bar: !aggr-map {first: foo, second: bar}
baz: !aggr-seq [42]
```

Default merge strategy is keep untouched

For example, this yaml document still has duplicate keys, but does not instruct aggregation:

```
foo: first
foo: second
bar: {first: foo}
bar: {second: bar}
baz: 42
```

So the late found values prevail:

```
foo: second
bar: {second: bar}
baz: 42
```

Limitations

Aggregation is permitted between tagged objects that share the same type. If not, the default merge strategy prevails.

For example, these examples:

```
foo: {first: value}
foo: !aggr-map {second: value}

bar: !aggr-map {first: value}
bar: 42

baz: !aggr-seq [42]
baz: [fail]

qux: 42
qux: !aggr-scalar fail
```

are interpreted like this:

```
foo: !aggr-map{second: value}

bar: 42

baz: [fail]

qux: !aggr-seq [fail]
```

Introspection

TODO: write this part

`salt.utils.aggregation.`**`aggregate`**`(`*obj_a, obj_b, level=False, map_class=<class 'salt.utils.aggregation.Map'>, sequence_class=<class 'salt.utils.aggregation.Sequence'>*`)`

Merge obj_b into obj_a.

```
>>> aggregate('first', 'second', True) == ['first', 'second']
True
```

`class` `salt.utils.aggregation.`**`Aggregate`**

Aggregation base.

class `salt.utils.aggregation.`**`Map`**

Map aggregation.

`salt.utils.aggregation.`**`Scalar`**(*obj*)

Shortcut for Sequence creation

```
>>> Scalar('foo') == Sequence(['foo'])
True
```

class `salt.utils.aggregation.`**`Sequence`**

Sequence aggregation.

Exceptions

Salt-specific exceptions should be thrown as often as possible so the various interfaces to Salt (CLI, API, etc) can handle those errors appropriately and display error messages appropriately.

`salt.exceptions`	This module is a central location for all salt exceptions

salt.exceptions

This module is a central location for all salt exceptions

exception `salt.exceptions.`**`ArgumentValueError`**(*message=”, info=None*)

Used when an invalid argument was passed to a command execution

exception `salt.exceptions.`**`AuthenticationError`**(*message=”*)

If sha256 signature fails during decryption

exception `salt.exceptions.`**`AuthorizationError`**(*message=”*)

Thrown when runner or wheel execution fails due to permissions

exception `salt.exceptions.`**`CheckError`**(*message=”, info=None*)

Used when a check fails

exception `salt.exceptions.`**`CommandExecutionError`**(*message=”, info=None*)

Used when a module runs a command which returns an error and wants to show the user the output gracefully instead of dying

exception `salt.exceptions.`**`CommandNotFoundError`**(*message=”*)

Used in modules or grains when a required binary is not available

exception `salt.exceptions.`**`EauthAuthenticationError`**(*message=”*)

Thrown when eauth authentication fails

exception `salt.exceptions.`**`FileLockError`**(*message, time_start=None, *args, **kwargs*)

Used when an error occurs obtaining a file lock

exception `salt.exceptions.`**`FileserverConfigError`**(*message=”*)

Used when invalid fileserver settings are detected

exception `salt.exceptions.`**`GitLockError`**(*errno, message, *args, **kwargs*)

Raised when an uncaught error occurs in the midst of obtaining an update/checkout lock in salt.utils.gitfs.

NOTE: While this uses the errno param similar to an OSError, this exception class is *not* as subclass of OSError. This is done intentionally, so that this exception class can be caught in a try/except without being caught as an OSError.

exception `salt.exceptions.`**`GitRemoteError`**(*message=''*)
Used by GitFS to denote a problem with the existence of the "origin" remote or part of its configuration

exception `salt.exceptions.`**`InvalidConfigError`**(*message='', info=None*)
Used when the config is invalid

exception `salt.exceptions.`**`InvalidEntityError`**(*message='', info=None*)
Used when an entity fails validation

exception `salt.exceptions.`**`LoaderError`**(*message=''*)
Problems loading the right renderer

exception `salt.exceptions.`**`LoggingRuntimeError`**
Raised when we encounter an error while logging

exception `salt.exceptions.`**`MasterExit`**
Rise when the master exits

exception `salt.exceptions.`**`MinionError`**(*message=''*)
Minion problems reading uris such as salt:// or http://

exception `salt.exceptions.`**`MissingSmb`**(*message=''*)
Raised when no smb library is found.

exception `salt.exceptions.`**`NotImplemented`**(*message=''*)
Used when a module runs a command which returns an error and wants to show the user the output gracefully instead of dying

exception `salt.exceptions.`**`PkgParseError`**(*message=''*)
Used when of the pkg modules cannot correctly parse the output from the CLI tool (pacman, yum, apt, aptitude, etc)

exception `salt.exceptions.`**`PublishError`**(*message=''*)
Problems encountered when trying to publish a command

exception `salt.exceptions.`**`SaltCacheError`**(*message=''*)
Thrown when a problem was encountered trying to read or write from the salt cache

exception `salt.exceptions.`**`SaltClientError`**(*message=''*)
Problem reading the master root key

exception `salt.exceptions.`**`SaltClientTimeout`**(*message, jid=None, *args, **kwargs*)
Thrown when a job sent through one of the Client interfaces times out

Takes the `jid` as a parameter

exception `salt.exceptions.`**`SaltCloudConfigError`**(*message=''*)
Raised when a configuration setting is not found and should exist.

exception `salt.exceptions.`**`SaltCloudException`**(*message=''*)
Generic Salt Cloud Exception

exception `salt.exceptions.`**`SaltCloudExecutionFailure`**(*message=''*)
Raised when too much failures have occurred while querying/waiting for data.

exception `salt.exceptions.`**`SaltCloudExecutionTimeout`**(*message=''*)
Raised when too much time has passed while querying/waiting for data.

exception `salt.exceptions.`**`SaltCloudNotFound`**(*message=''*)
Raised when some cloud provider function cannot find what's being searched.

exception `salt.exceptions.`**`SaltCloudPasswordError`**(*message=''*)
Raise when virtual terminal password input failed

exception salt.exceptions.**SaltCloudSystemExit**(*message*, *exit_code=1*)

This exception is raised when the execution should be stopped.

exception salt.exceptions.**SaltConfigurationError**(*message=”*)

Configuration error

exception salt.exceptions.**SaltDaemonNotRunning**(*message=”*)

Throw when a running master/minion/syndic is not running but is needed to perform the requested operation (e.g., eauth).

exception salt.exceptions.**SaltDeserializationError**(*message=”*)

Thrown when salt cannot deserialize data.

exception salt.exceptions.**SaltException**(*message=”*)

Base exception class; all Salt-specific exceptions should subclass this

pack()

Pack this exception into a serializable dictionary that is safe for transport via msgpack

exception salt.exceptions.**SaltInvocationError**(*message=”*)

Used when the wrong number of arguments are sent to modules or invalid arguments are specified on the command line

exception salt.exceptions.**SaltMasterError**(*message=”*)

Problem reading the master root key

exception salt.exceptions.**SaltMasterUnresolvableError**(*message=”*)

Problem resolving the name of the Salt master

exception salt.exceptions.**SaltNoMinionsFound**(*message=”*)

An attempt to retrieve a list of minions failed

exception salt.exceptions.**SaltRenderError**(*message*, *line_num=None*, *buf=”*, *marker=’ <======================’*, *trace=None*)

Used when a renderer needs to raise an explicit error. If a line number and buffer string are passed, get_context will be invoked to get the location of the error.

exception salt.exceptions.**SaltReqTimeoutError**(*message=”*)

Thrown when a salt master request call fails to return within the timeout

exception salt.exceptions.**SaltRunnerError**(*message=”*)

Problem in runner

exception salt.exceptions.**SaltSyndicMasterError**(*message=”*)

Problem while proxying a request in the syndication master

exception salt.exceptions.**SaltSystemExit**(*code=0*, *msg=None*)

This exception is raised when an unsolvable problem is found. There’s nothing else to do, salt should just exit.

exception salt.exceptions.**SaltWheelError**(*message=”*)

Problem in wheel

exception salt.exceptions.**TemplateError**(*message=”*)

Used when a custom error is triggered in a template

exception salt.exceptions.**TimedProcTimeoutError**(*message=”*)

Thrown when a timed subprocess does not terminate within the timeout, or if the specified timeout is not an int or a float

exception salt.exceptions.**TimeoutError**(*message=”*)

Thrown when an opration cannot be completet within a given time limit.

exception salt.exceptions.**TokenAuthenticationError**(*message=”*)

Thrown when token authentication fails

exception `salt.exceptions.`**`VMwareApiError`**(*message=”, info=None*)
Used when representing a generic VMware API error

exception `salt.exceptions.`**`VMwareConnectionError`**(*message=”, info=None*)
Used when the client fails to connect to a either a VMware vCenter server or to a ESXi host

exception `salt.exceptions.`**`VMwareFileNotFoundError`**(*message=”, info=None*)
Used when representing a generic VMware error if a file not found

exception `salt.exceptions.`**`VMwareMultipleObjectsError`**(*message=”, info=None*)
Used when multiple objects were retrieved (and one was expected)

exception `salt.exceptions.`**`VMwareNotFoundError`**(*message=”, info=None*)
Used when a VMware object was not found

exception `salt.exceptions.`**`VMwareObjectExistsError`**(*message=”, info=None*)
Used when a VMware object already exists

exception `salt.exceptions.`**`VMwareObjectNotFoundError`**(*message=”, info=None*)
Used when a VMware object was not found

exception `salt.exceptions.`**`VMwareObjectRetrievalError`**(*message=”, info=None*)
Used when a VMware object cannot be retrieved

exception `salt.exceptions.`**`VMwarePowerOnError`**(*message=”, info=None*)
Used when error occurred during power on

exception `salt.exceptions.`**`VMwareRuntimeError`**(*message=”, info=None*)
Used when a runtime error is encountered when communicating with the vCenter

exception `salt.exceptions.`**`VMwareSaltError`**(*message=”, info=None*)
Used when a VMware object cannot be retrieved

exception `salt.exceptions.`**`VMwareSystemError`**(*message=”, info=None*)
Used when representing a generic VMware system error

exception `salt.exceptions.`**`VMwareVmCreationError`**(*message=”, info=None*)
Used when a configuration parameter is incorrect

exception `salt.exceptions.`**`VMwareVmRegisterError`**(*message=”, info=None*)
Used when a configuration parameter is incorrect

`salt.exceptions.`**`get_error_message`**(*error*)
Get human readable message from Python Exception

The Salt Fileserver and Client

Introduction

Salt has a modular fileserver, and multiple client classes which are used to interact with it. This page serves as a developer's reference, to help explain how the fileserver and clients both work.

Fileserver

The fileserver is not a daemon, so the fileserver and client are not a true server and client in the traditional sense. Instead, the fileserver is simply a class (`salt.fileserver.Fileserver`), located in salt/fileserver/__init__.py. This class has access to the configured fileserver backends via a loader instance, referenced as `self.servers`. When a request comes in from the fileclient, it will ultimately result in a `Fileserver` class function being run.

The functions in this class will run corresponding functions in the configured fileserver backends to perform the requested action. So, in summary:

1. A fileclient class makes a request…
2. which triggers the fileserver to run a function…
3. which runs a named function in each of the configured backends.

Not all of the functions will always execute on every configured backend. For instance, the `find_file` function in the fileserver will stop when it finds a match, so if it finds a match for the desired path in the first configured backend, it won't proceed and try to find the file in the next backend in the list.

Additionally, not all backends implement all functions in the `salt.fileserver.Fileserver` class. For instance, there is a function called `update`, which exists to update remote fileservers such as the `git`, `hg`, and `svn` backends. This action has no use however in the `roots` backend, so it is simply not implemented there, and thus the `roots` backend will be skipped if the `update` function is run on the fileserver.

Backends for the fileserver are located in salt/fileserver/ (the files not named `__init__.py`).

Fileclient

There are three fileclient classes:

salt.fileclient.RemoteClient

This client is used when *`file_client`* is set to `remote`. This is how minions request files from the master.

Functions in this client will craft a payload and send it to the master via the transport channel. This is the same way that the minion asks the minion to do other things, such as updating and requesting data from the mine. The payload will be a dictionary with a key called `cmd`, and other values as needed.

Payloads sent via the transport channel are processed my an MWorker instance on the master, and the MWorker's `_handle_aes()` function will execute the command. The command will be a function attribute of the `salt.master.AESFuncs` class. The AESFuncs class' `__setup_fileserver()` function instantiates a `salt.fileserver.Fileserver` instance and maps its member functions to AESFuncs attributes. This is what makes the fileserver functions available remotely. The result of the function is returned back through the transport channel to the minion.

Transporting files is done in chunks, the size of which is decided by the `file_buffer_size` config option. If you look at the `serve_file()` function in any of the fileserver backends, you can see how the `loc` value in the payload determines the offset so that an intermediate chunk of the file can be served. The RemoteClient's `get_file()` function will loop until the end of the file is reached, retrieving one chunk at a time.

salt.fileclient.FSClient

This client is used when *`file_client`* is set to `local`. This is how masterless minions request files.

This class inherits from the RemoteClient, but instead of using a transport channel (zmq, tcp, etc.), it uses a "fake" transport channel (`salt.fileserver.FSChan`), which implements its own `send()` function. Thus, when a function that the FSClient inherits from the RemoteClient runs `self.channel.send()`, it's actually calling `salt.fileserver.FSChan.send()`, which calls corresponding functions in the `salt.fileserver.Fileserver()` class. The result is that local file requests use the same code as remote file requests, they just bypass sending them through an actual transport channel and instead call them on the FSChan's Fileserver instance.

salt.fileclient.LocalClient

This client is now used exclusively by Pillar. This used to be used when *`file_client`* was set to `local`, but the `FSChan` class was written to allow minions with `file_client: local` to access the full set of backends. This class will probably be renamed at some point as it is often confused with `salt.client.LocalClient`.

The `cp` Module

Most of the user-facing interaction with the fileclient happens via the *`cp`* module. The functions in this module instantiate a fileclient instance (if one is not already saved to the `__context__` dunder) and run fileclient functions.

Updating the Fileserver

The master daemon spawns a process dedicated to routine maintenance tasks upon startup. This process runs an instance of `salt.master.Maintenance`, which loops forever, running a series of functions and then sleeping for a length of time determined by the *`loop_interval`* config option. One of the maintenance tasks is to update the fileserver, and it essentially runs `salt.fileserver.Fileserver.update()`, which as we know from above will run all configured backends' `update()` functions, if present. This is now remote fileservers like `git`, `hg`, and `svn` stay up-to-date.

For the local file_client (FSClient), since it does not interact with the master, upon spawning of its FSChan it will update the fileserver.

Salt opts dictionary

It is very common in the Salt codebase to see *opts* referred to in a number of contexts.

For example, it can be seen as __*opts*__ in certain cases, or simply as *opts* as an argument to a function in others.

Simply put, this data structure is a dictionary of Salt's runtime configuration information that's passed around in order for functions to know how Salt is configured.

When writing Python code to use specific parts of Salt, it may become necessary to initialize a copy of *opts* from scratch in order to have it available for a given function.

To do so, use the utility functions available in *salt.config*.

As an example, here is how one might generate and print an options dictionary for a minion instance:

```python
import salt.config
opts = salt.config.minion_config('/etc/salt/minion')
print(opts)
```

To generate and display *opts* for a master, the process is similar:

```python
import salt.config
opts = salt.config.master_config('/etc/salt/master')
print(opts)
```

Unicode in Salt

Though Unicode handling in large projects can often be complex, Salt adheres to several basic rules to help developers handle Unicode correctly.

(For a basic introduction to this problem, see Ned Batchelder's *excellent intoroduction to the topic <http://nedbatchelder.com/text/unipain/unipain.html>*.

Salt's basic workflow for Unicode handling is as follows:

1) Salt should convert whatever data is passed on CLI/API to Unicode. Internally, everything that Salt does should be Unicode unless it is printing to the screen or writing to storage.

2) Modules and various Salt pluggable systems use incoming data assuming Unicode.

   2.1) For Salt modules that query an API; the module should convert the data received from the API into Unicode.

   2.2) For Salt modules that shell out to get output; the module should convert data received into Unicode. (This does not apply if using the *cmd* execution module, which should handle this for you.

   2.3) For Salt modules which print directly to the console (not via an outputter) or which write directly to disk, a string should be encoded when appropriate. To handle this conversion, the global variable `__salt_system_encoding__` is available, which declares the locale of the system that Salt is running on.

3) When a function in a Salt module returns a string, it should return a `unicode` type in Python 2.

4) When Salt delivers the data to an outputter or a returner, it is the job of the outputter or returner to encode the Unicode before displaying it on the console or writing it to storage.

27.24 Salt Community Projects

This page contains links to Salt-related projects created by community members. If you come across a useful project please add it to the list!

27.24.1 Hubblestack

Hubble is a modular, open-source security compliance framework built on top of SaltStack. The project provides on-demand profile-based auditing, real-time security event notifications, automated remediation, alerting and reporting.

http://hubblestack.io/

27.24.2 alkali

alkali is a collections of SaltStack states and pillar data that provide just the basics for provisioning Linux instances that may be built upon. alkali is a starter kit of sorts, to help new users to SaltStack get up-and-running quickly with the most commonly used, core packages.

https://github.com/zulily/alkali

27.24.3 buoyant

buoyant leverages docker to provide an alternative to VM-centric SaltStack development environments. buoyant containers may be spun up nearly instantly, once an initial docker image has been built.

https://github.com/zulily/buoyant

27.24.4 Salt Sandbox

Salt Sandbox is a multi-VM Vagrant-based Salt development environment used for creating and testing new Salt state modules outside of your production environment. It's also a great way to learn firsthand about Salt and its remote execution capabilities.

https://github.com/elasticdog/salt-sandbox

27.24.5 Salt Vagrant Demo

A Salt Demo using Vagrant.

https://github.com/UtahDave/salt-vagrant-demo

CHAPTER

TWENTYEIGHT

RELEASE NOTES

See the version numbers page for more information about the version numbering scheme.

28.1 Latest Branch Release

Release Candidate

28.2 Previous Releases

28.2.1 Salt 3000 Release Notes - Codename Neon

Security Advisory

For historical reasons, Salt requires PyCrypto as a "lowest common denominator". However, PyCrypto is unmaintained and best practice is to manually upgrade to use a more maintained library such as PyCryptodome. See Issue #52674 and Issue #54115 for more info

New Versioning

The neon release has removed the date versioning. Going forward we will use a non-date based version schema beginning at 3000. The version will be MAJOR.PATCH. For a planned release containing features and/or bug fixes the MAJOR version will be incremented. Please review the approved SEP for further details.

The new versioning scheme is PEP 440 compliant, but distutils.StrictVersion will result in an error `invalid version number`. If using StrictVersion to compare Salt's versions, please use LooseVersion. There is also the packaging library you can use to compare versions. Another alternative is using the *`salt version module`*

Python 3.8 Not Supported

Please note that running Salt with Python 3.8 is currently not supported. It is recommended to not use a version higher than 3.7.

Vendored Tornado Code Base

The Tornado code base has been included as `salt.ext.tornado`. This was done to provide a consistent version of Tornado across all operating systems while we continue to work on upgrading Tornado to a recent version. New code that uses Tornado should import the module from `salt.ext.tornado` instead of importing the system version of Tornado.

Msgpack 1.0.0rc1 Incompatibility

Salt is currently incompatible with mgspack 1.0.0rc1. We recommend using versions < 1.0.0 . This impacts the Salt Master's ability to run commands against minions. Please see Issue 56007 for more details and updates.

Pip Installing Salt On Windows

This release will not install correctly on windows when using pip to install salt. This issue is addressed in PR 56099. The workaround for this issue is:

```
pip install "pywin32==224" "WMI==1.4.9"
```

Those using the Salt Windows Installer are not affected by this issue.

Saltcheck Updates

Available since 2018.3, the *`saltcheck module`* has been enhanced to:

- Support saltenv environments
- Associate tests with states by naming convention
- Adds report_highstate_tests function
- Adds empty and notempty assertions
- Adds skip keyword
- Adds print_result keyword
- Adds assertion_section keyword
- Use saltcheck.state_apply to run state.apply for test setup or teardown
- Changes output to display test time
- Works with salt-ssh

Saltcheck provides unittest like functionality requiring only the knowledge of salt module execution and yaml. Saltcheck uses salt modules to return data, then runs an assertion against that return. This allows for testing with all the features included in salt modules.

In order to run state and highstate saltcheck tests, a sub-folder in the state directory must be created and named `saltcheck-tests`. Tests for a state should be created in files ending in `*.tst` and placed in the `saltcheck-tests` folder. `tst` files are run through the salt rendering system, enabling tests to be written in yaml (or renderer of choice), and include jinja, as well as the usual grain and pillar information. Like states, multiple tests can be specified in a `tst` file. Multiple `tst` files can be created in the `saltcheck-tests` folder, and should be named the same as the associated state. The `id` of a test works in the same manner as in salt state files and should be unique and descriptive.

Usage

Example file system layout:

```
/srv/salt/apache/
    init.sls
    config.sls
    saltcheck-tests/
        init.tst
        config.tst
        deployment_validation.tst
```

Tests can be run for each state by name, for all `apache/saltcheck/*.tst` files, or for all states assigned to the minion in top.sls. Tests may also be created with no associated state. These tests will be run through the use of `saltcheck.run_state_tests`, but will not be automatically run by `saltcheck.run_highstate_tests`.

```
salt '*' saltcheck.run_state_tests apache,apache.config
salt '*' saltcheck.run_state_tests apache check_all=True
salt '*' saltcheck.run_highstate_tests
salt '*' saltcheck.run_state_tests apache.deployment_validation
```

Example Tests

```
{# will run the common salt state before further testing #}
setup_test_environment:
  module_and_function: saltcheck.state_apply
  args:
    - common
  pillar-data:
    data: value

{% for package in ["apache2", "openssh"] %}
{# or another example #}
{# for package in salt['pillar.get']("packages") #}
jinja_test_{{ package }}_latest:
  module_and_function: pkg.upgrade_available
  args:
    - {{ package }}
  assertion: assertFalse
{% endfor %}

validate_user_present_and_shell:
  module_and_function: user.info
  args:
    - root
  assertion: assertEqual
  expected-return: /bin/bash
  assertion_section: shell
  print_result: False

skip_test:
  module_and_function: pkg.upgrade_available
  args:
    - apache2
  assertion: assertFalse
  skip: True
```

Output Format Changes

Saltcheck output has been enhanced to display the time taken per test. This results in a change to the output format.

Previous Output:

```
local:
  |_
    ----------
    ntp:
        ----------
        ntp-client-installed:
            Pass
        ntp-service-status:
            Pass
  |_
    ----------
    TEST RESULTS:
        ----------
        Failed:
            0
        Missing Tests:
            0
        Passed:
            2
```

New output:

```
local:
  |_
    ----------
    ntp:
        ----------
        ntp-client-installed:
            ----------
            duration:
                1.0408
            status:
                Pass
        ntp-service-status:
            ----------
            duration:
                1.464
            status:
                Pass
  |_
    ----------
    TEST RESULTS:
        ----------
        Execution Time:
            2.5048
        Failed:
            0
        Missing Tests:
            0
        Passed:
            2
        Skipped:
```

(continues on next page)

(continued from previous page)

```
            0
```

Unless and onlyif Enhancements

The `unless` and `onlyif` requisites can now be operated with salt modules. The dictionary must contain an argument `fun` which is the module that is being run, and everything else must be passed in under the args key or will be passed as individual kwargs to the module function.

Note: Certain states have an `unless`/`onlyif` implementation that predates this feature, and may not work as expected:

- states.git
- states.cmd
- states.macpackage
- states.file
- states.docker_container

Examples:

```
check external ip address:
  http.query:
    - name: https://icanhazip.com
    - status: 200
    - onlyif:
      - fun: file.file_exists
        path: /usr/local/bin/whatever
```

is equivalent to

```
check another ip address:
  http.query:
    - name: https://icanhazip.com
    - status: 200
    - onlyif:
      - test -f /tmp/fnord.txt
```

Another example:

```
set mysql root password:
  debconf.set:
    - name: mysql-server-5.7
    - data:
        'mysql-server/root_password': {'type': 'password', 'value': {{pillar['mysql.
↪pass']}} }
    - unless:
      - fun: pkg.version
        args:
          - mysql-server-5.7
```

Keystore State and Module

A new *state* and *execution module* for manaing Java Keystore files is now included. It allows for adding/removing/listing as well as managing keystore files.

```
# salt-call keystore.list /path/to/keystore.jks changeit
local:
  |_
    ----------
    alias:
        hostname1
    expired:
        True
    sha1:
        CB:5E:DE:50:57:99:51:87:8E:2E:67:13:C5:3B:E9:38:EB:23:7E:40
    type:
        TrustedCertEntry
    valid_start:
        August 22 2012
    valid_until:
        August 21 2017
```

```
define_keystore:
  keystore.managed:
    - name: /tmp/statestore.jks
    - passphrase: changeit
    - force_remove: True
    - entries:
      - alias: hostname1
        certificate: /tmp/testcert.crt
      - alias: remotehost
        certificate: /tmp/512.cert
        private_key: /tmp/512.key
      - alias: stringhost
        certificate: |
          -----BEGIN CERTIFICATE-----
          MIICEjCCAX
          Hn+GmxZA
          -----END CERTIFICATE-----
```

XML Module

A new state and execution module for editing XML files is now included. Currently it allows for editing values from an xpath query, or editing XML IDs.

```
# salt-call xml.set_attribute /tmp/test.xml ".//actor[@id='3']" editedby "Jane Doe"
local:
    True
# salt-call xml.get_attribute /tmp/test.xml ".//actor[@id='3']"
local:
    ----------
    editedby:
        Jane Doe
    id:
        3
# salt-call xml.get_value /tmp/test.xml ".//actor[@id='2']"
```

(continues on next page)

(continued from previous page)

```
local:
    Liam Neeson
# salt-call xml.set_value /tmp/test.xml ".//actor[@id='2']" "Patrick Stewart"
local:
    True
# salt-call xml.get_value /tmp/test.xml ".//actor[@id='2']"
local:
    Patrick Stewart
```

```
ensure_value_true:
  xml.value_present:
    - name: /tmp/test.xml
    - xpath: .//actor[@id='1']
    - value: William Shatner
```

LGPO Execution Module

Multiple fixes were made to the *win_lgpo* to expand support for additional policies and improve performance. Issues with encoding and unsupported characters (smart-quotes, em-dash, etc) found in the ADML files as well as whitespace in some policies have been addressed.

Speed enhancements include:

- Caching the compiled xml from ADMX/ADML files in `__context__`
- Lowercasing all keys before compiling XML to remove an expensive XPath lookup

Additional functionality:

- Adds the *lgpo.get_policy* function that allows you to get the current settings for a single policy.
- Changes some policy names that were overly long

LGPO State Module

Multiple changes were made to the *win_lgpo* state to improve performance.

It now uses the *lgpo.get_policy* function instead of the *lgpo.get* to avoid reloading all settings on the machine at the beginning and end of each state run.

Supports the new shorter policy names but also attempts to resolve the older, longer policy names.

Win_network Salt Util

Now uses .NET libraries to obtain Interface information on Windows systems that support it. Unsupported systems fall back to the more expensive WMI calls. This is important as this data is gathered for grains and occurs each time a process is forked.

Jinja enhancements

Troubleshooting Jinja map files

A new *execution module* for `map.jinja` troubleshooting has been added.

Assuming the map is loaded in your formula SLS as follows:

```
{% from "myformula/map.jinja" import myformula with context %}
```

The following command can be used to load the map and check the results:

```
salt myminion jinja.load_map myformula/map.jinja myformula
```

The module can be also used to test `json` and `yaml` maps:

```
salt myminion jinja.import_yaml myformula/defaults.yaml

salt myminion jinja.import_json myformula/defaults.json
```

json_query filter

A port of Ansible *`json_query`* Jinja filter has been added. It allows making queries against JSON data using JMESPath language. Could be used to filter `pillar` data, `yaml` maps, and also useful with *`http_query`*.

Depends on the jmespath Python module.

Slot Syntax Updates

The slot syntax has been updated to support parsing dictionary responses and to append text.

```
demo dict parsing and append:
  test.configurable_test_state:
    - name: slot example
    - changes: False
    - comment: __slot__:salt:test.arg(shell="/bin/bash").kwargs.shell ~ /appended
```

```
local:
  ----------
        ID: demo dict parsing and append
  Function: test.configurable_test_state
      Name: slot example
    Result: True
   Comment: /bin/bash/appended
   Started: 09:59:58.623575
  Duration: 1.229 ms
   Changes:
```

Also, slot parsing is now supported inside of nested state data structures (dicts, lists, unless/onlyif args):

```
demo slot parsing for nested elements:
  file.managed:
    - name: /tmp/slot.txt
    - source: salt://slot.j2
    - template: jinja
    - context:
        # Slot inside of the nested context dictionary
        variable: __slot__:salt:test.echo(a_value)
    - unless:
      - fun: file.search
        args:
          # Slot as unless argument
```

(continues on next page)

(continued from previous page)

```
        - __slot__:salt:test.echo(/tmp/slot.txt)
        - "DO NOT OVERRIDE"
      ignore_if_missing: True
```

State Changes

- The *file.symlink* state was fixed to remove existing file system entries other than files, directories and symbolic links properly.
- The `onchanges` and `prereq` *requisites* now behave properly in test mode, due to removing `pchanges`.
- Added new *ssh_auth.manage* state to ensure only the specified ssh keys are present for the specified user.
- Added new *saltutil* state to use instead of `module.run` to more easily handle change.

Module Changes

- Added new `boto_ssm` module to set and query secrets in AWS SSM parameters.
- The *file.remove* module was fixed to remove file system entries other than files, directories and symbolic links properly.

Enhancements to Engines

Multiple copies of a particular Salt engine can be configured by including the `engine_module` parameter in the engine configuration.

```
engines:
  - production_logstash:
      host: production_log.my_network.com
      port: 5959
      proto: tcp
      engine_module: logstash
  - develop_logstash:
      host: develop_log.my_network.com
      port: 5959
      proto: tcp
      engine_module: logstash
```

- A new `fluent engine` has been added to export Salt events to fluentd.

  ```
  engines:
    - fluent
        host: localhost
        port: 24224
  ```

  ```
  <source>
    @type forward
    port 24224
  </source>
  <match saltstack.**>
    @type file
    path /var/log/td-agent/saltstack
  </match>
  ```

Enhancements to Beacons

Multiple copies of a particular Salt beacon can be configured by including the `beacon_module` parameter in the beacon configuration.

```
beacons:
  watch_importand_file:
    - files:
        /etc/important_file: {}
    - beacon_module: inotify
  watch_another_file:
    - files:
        /etc/another_file: {}
    - beacon_module: inotify
```

Enhancements to chroot

- New functions added to chroot `apply`, `sls`, and `highstate` that allow executing states in sls files or running apply/highstate inside of a chroot.

Minion-side ACL

Salt has had master-side ACL for the salt mine for some time, where the master configuration contained *mine_get* that specified which minions could request which functions. However, now you can specify which minions can access a function in the salt mine function definition itself (or when calling *`mine.send`*). This targeting works the same as the generic minion targeting as specified *here*. The parameters used are `allow_tgt` and `allow_tgt_type`. See also *the documentation of the Salt Mine*. Please note that if you want to use this new feature both your minion and masters will need to be on atleast version 3000.

Deprecations

Raet Deprecated

- The Raet transport has been deprecated. Please use the supported transport protocols tcp or zeromq.

Module Deprecations

- The hipchat module has been removed due to the service being retired. *`Google Chat`*, *`MS Teams`*, or *`Slack`* may be suitable replacements.
- The *`dockermod`* module has been changed as follows:
  - Support for the `tags` kwarg has been removed from the *`dockermod.resolve_tag`* function.
  - Support for the `network_id` kwarg has been removed from the *`dockermod.connect_container_to_network`* function. Please use `net_id` instead.
  - Support for the `name` kwarg has been removed from the *`dockermod.sls_build`* function. Please use `repository` and `tag` instead.
  - Support for the `image` kwarg has been removed from the following functions. In all cases, please use both the `repository` and `tag` options instead:
    - *`dockermod.build`*

    - *dockermod.commit*
    - *dockermod.import*
    - *dockermod.load*
    - *dockermod.tag*

- The heat module has removed the `enviroment` kwarg from the *heat.create_stack* and *heat.update_stack* functions due to a spelling error. Please use `environment` instead.
- The *ssh* execution module has been changed as follows:
  - Support for the `ssh.get_known_host` function has been removed. Please use the *ssh.get_known_host_entries* function instead.
  - Support for the `ssh.recv_known_host` function has been removed. Please use the *ssh.recv_known_host_entries* function instead.
- The :py:mod‘firewalld <salt.modules.firewalld>‘ module has been changed as follows:
  - The default setting for the `force_masquerade` option in the `firewalld.add_port <salt.module.firewalld.add_port()` function has changed from `True` to `False`.
  - Support for the `force_masquerade` option in the `firewalld.add_port_fwd <salt.module.firewalld.add_port_fwd()` function has been changed from `True` to `False`.

State Deprecations

- The hipchat state has been removed due to the service being retired. *MS Teams* or *Slack* may be suitable replacements.
- The cmd state module has removed the `quiet` kwarg from the *cmd.run* function. Please set `output_loglevel` to `quiet` instead.
- The heat state module has removed the `enviroment` kwarg from the *heat.deployed* function due to a spelling error. Please use `environment` instead.
- The :py:mod‘firewalld <salt.states.firewalld>‘ state has been changed as follows:
  - The default setting for the `prune_services` option in the *firewalld.present* function has changed from `True` to `False`.

Fileserver Deprecations

- The hgfs fileserver had the following config options removed:
  - The `hgfs_env_whitelist` config option has been removed in favor of `hgfs_saltenv_whitelist`.
  - The `hgfs_env_blacklist` config option has been removed in favor of `hgfs_saltenv_blacklist`.
- The svnfs fileserver had the following config options removed:
  - The `svnfs_env_whitelist` config option has been removed in favor of `svnfs_saltenv_whitelist`.
  - The `svnfs_env_blacklist` config option has been removed in favor of `svnfs_saltenv_blacklist`.
- The gitfs fileserver had the following config options removed:

  - The `gitfs_env_whitelist` config option has been removed in favor of `gitfs_saltenv_whitelist`.
  - The `gitfs_env_blacklist` config option has been removed in favor of `gitfs_saltenv_blacklist`.

Engine Removal

- The hipchat engine has been removed due to the service being retired. For users migrating to Slack, the *`slack`* engine may be a suitable replacement.

Returner Removal

- The hipchat returner has been removed due to the service being retired. For users migrating to Slack, the *`slack`* returner may be a suitable replacement.

Grain Deprecations

For `smartos` some grains have been deprecated. These grains have been removed.

- The `hypervisor_uuid` has been replaced with `mdata:sdc:server_uuid` grain.
- The `datacenter` has been replaced with `mdata:sdc:datacenter_name` grain.

Cloud Deprecations

- The nova cloud driver has been removed in favor of the openstack cloud driver.

Jinja Filter Deprecations

- The following jinja filters are set to be removed in the Aluminium release:
  - *`json_decode_dict`* in favor of *`tojson`*
  - *`json_decode_list`* in favor of *`tojson`*

Utils Deprecations

- All of the functions in salt.utils.__init__.py have been removed. These include:
  - *salt.utils.option*
  - *salt.utils.required_module_list*
  - *salt.utils.required_modules_error*
  - *salt.utils.get_accumulator_dir*. Please use `salt.state.get_accumulator_dir()` instead.
  - *salt.utils.fnmatch_multiple*. Please use `salt.utils.itertools.fnmatch_multiple()` instead.
  - *salt.utils.appendproctitle*. Please use `salt.utils.process.appendproctitle()` instead.
  - *salt.utils.daemonize*. Please use `salt.utils.process.daemonize()` instead.

- *salt.utils.daemonize_if.* Please use `salt.utils.process.daemonize_if()` instead.
- *salt.utils.reinit_crypto.* Please use `salt.utils.crypt.reinit_crypto()` instead.
- *salt.utils.pem_finger.* Please use `salt.utils.crypt.pem_finger()` instead.
- *salt.utils.to_bytes.* Please use `salt.utils.stringutils.to_bytes()` instead.
- *salt.utils.to_str.* Please use `salt.utils.stringutils.to_str()` instead.
- *salt.utils.to_unicode.* Please use `salt.utils.stringutils.to_unicode()` instead.
- *salt.utils.str_to_num.* Please use `salt.utils.stringutils.to_num()` instead.
- *salt.utils.is_quoted.* Please use `salt.utils.stringutils.is_quoted()` instead.
- *salt.utils.dequote.* Please use `salt.utils.stringutils.dequote()` instead.
- *salt.utils.is_hex.* Please use `salt.utils.stringutils.is_hex()` instead.
- *salt.utils.is_bin_str.* Please use `salt.utils.stringutils.is_binary()` instead.
- *salt.utils.rand_string.* Please use `salt.utils.stringutils.random()` instead.
- *salt.utils.contains_whitespace.* Please use `salt.utils.stringutils.contains_whitespace()` instead.
- *salt.utils.build_whitespace_split_regex.* Please use `salt.utils.stringutils.build_whitespace_split_regex()` instead.
- *salt.utils.expr_match.* Please use `salt.utils.stringutils.expr_match()` instead.
- *salt.utils.check_whitelist_blacklist.* Please use `salt.utils.stringutils.check_whitelist_blacklist()` instead.
- *salt.utils.check_include_exclude.*Please use `salt.utils.stringutils.check_include_exclude()` instead.
- *salt.utils.print_cli.*Please use `salt.utils.stringutils.print_cli()` instead.
- *salt.utils.clean_kwargs.*Please use `salt.utils.args.clean_kwargs()` instead.
- *salt.utils.invalid_kwargs.*Please use `salt.utils.args.invalid_kwargs()` instead.
- *salt.utils.shlex_split.*Please use `salt.utils.args.shlex_split()` instead.
- *salt.utils.arg_lookup.*Please use `salt.utils.args.arg_lookup()` instead.
- *salt.utils.argspec_report.*Please use `salt.utils.args.argspec_report()` instead.
- *salt.utils.split_input.*Please use `salt.utils.args.split_input()` instead.
- *salt.utils.test_mode.*Please use `salt.utils.args.test_mode()` instead.
- *salt.utils.format_call.*Please use `salt.utils.args.format_call()` instead.
- *salt.utils.which.*Please use `salt.utils.path.which()` instead.
- *salt.utils.which_bin.*Please use `salt.utils.path.which_bin()` instead.
- *salt.utils.path_join.*Please use `salt.utils.path.join()` instead.
- *salt.utils.check_or_die.*Please use `salt.utils.path.check_or_die()` instead.
- *salt.utils.sanitize_win_path_string.*Please use `salt.utils.path.sanitize_win_path()` instead.
- *salt.utils.rand_str.*Please use `salt.utils.hashutils.random_hash()` instead.
- *salt.utils.get_hash.*Please use `salt.utils.hashutils.get_hash()` instead.

- *salt.utils.is_windows.*Please use `salt.utils.platform.is_windows()` instead.
- *salt.utils.is_proxy.*Please use `salt.utils.platform.is_proxy()` instead.
- *salt.utils.is_linux.*Please use `salt.utils.platform.is_linux()` instead.
- *salt.utils.is_darwin.*Please use `salt.utils.platform.is_darwin()` instead.
- *salt.utils.is_sunos.*Please use `salt.utils.platform.is_sunos()` instead.
- *salt.utils.is_smartos.*Please use `salt.utils.platform.is_smartos()` instead.
- *salt.utils.is_smartos_globalzone.*Please use `salt.utils.platform.is_smartos_globalzone()` instead.
- *salt.utils.is_smartos_zone.*Please use `salt.utils.platform.is_smartos_zone()` instead.
- *salt.utils.is_freebsd.*Please use `salt.utils.platform.is_freebsd()` instead.
- *salt.utils.is_netbsd.*Please use `salt.utils.platform.is_netbsd()` instead.
- *salt.utils.is_openbsd.*Please use `salt.utils.platform.is_openbsd()` instead.
- *salt.utils.is_aix.*Please use `salt.utils.platform.is_aix()` instead.
- *salt.utils.safe_rm.*Please use `salt.utils.files.safe_rm()` instead.
- *salt.utils.is_empty.*Please use `salt.utils.files.is_empty()` instead.
- *salt.utils.fopen.*Please use `salt.utils.files.fopen()` instead.
- *salt.utils.flopen.*Please use `salt.utils.files.flopen()` instead.
- *salt.utils.fpopen.*Please use `salt.utils.files.fpopen()` instead.
- *salt.utils.rm_rf.*Please use `salt.utils.files.rm_rf()` instead.
- *salt.utils.mkstemp.*Please use `salt.utils.files.mkstemp()` instead.
- *salt.utils.istextfile.*Please use `salt.utils.files.is_text_file()` instead.
- *salt.utils.is_bin_file.*Please use `salt.utils.files.is_binary()` instead.
- *salt.utils.list_files.*Please use `salt.utils.files.list_files()` instead.
- *salt.utils.safe_walk.*Please use `salt.utils.files.safe_walk()` instead.
- *salt.utils.st_mode_to_octal.*Please use `salt.utils.files.st_mode_to_octal()` instead.
- *salt.utils.normalize_mode.*Please use `salt.utils.files.normalize_mode()` instead.
- *salt.utils.human_size_to_bytes.*Please use `salt.utils.files.human_size_to_bytes()` instead.
- *salt.utils.backup_minion.*Please use `salt.utils.files.backup_minion()` instead.
- *salt.utils.str_version_to_evr.*Please use `salt.utils.pkg.rpm.version_to_evr()` instead.
- *salt.utils.parse_docstring.*Please use `salt.utils.doc.parse_docstring()` instead.
- *salt.utils.compare_versions.*Please use `salt.utils.versions.compare()` instead.
- *salt.utils.version_cmp.*Please use `salt.utils.versions.version_cmp()` instead.
- *salt.utils.warn_until.*Please use `salt.utils.versions.warn_until()` instead.
- *salt.utils.kwargs_warn_until.*Please use `salt.utils.versions.kwargs_warn_until()` instead.
- *salt.utils.get_color_theme.*Please use `salt.utils.color.get_color_theme()` instead.

- *salt.utils.get_colors*.Please use `salt.utils.color.get_colors()` instead.
- *salt.utils.gen_state_tag*.Please use `salt.utils.state.gen_tag()` instead.
- *salt.utils.search_onfail_requisites*.Please use `salt.utils.state.search_onfail_requisites()` instead.
- *salt.utils.check_onfail_requisites*.Please use `salt.utils.state.check_onfail_requisites()` instead.
- *salt.utils.check_state_result*.Please use `salt.utils.state.check_result()` instead.
- *salt.utils.get_user*.Please use `salt.utils.user.get_user()` instead.
- *salt.utils.get_uid*.Please use `salt.utils.user.get_uid()` instead.
- *salt.utils.get_specific_user*.Please use `salt.utils.user.get_specific_user()` instead.
- *salt.utils.chugid*.Please use `salt.utils.user.chugid()` instead.
- *salt.utils.chugid_and_umask*.Please use `salt.utils.user.chugid_and_umask()` instead.
- *salt.utils.get_default_group*.Please use `salt.utils.user.get_default_group()` instead.
- *salt.utils.get_group_list*.Please use `salt.utils.user.get_group_list()` instead.
- *salt.utils.get_group_dict*.Please use `salt.utils.user.get_group_dict()` instead.
- *salt.utils.get_gid_list*.Please use `salt.utils.user.get_gid_list()` instead.
- *salt.utils.get_gid*.Please use `salt.utils.user.get_gid()` instead.
- *salt.utils.enable_ctrl_logoff_handler*.Please use `salt.utils.win_functions.enable_ctrl_logoff_handler()` instead.
- *salt.utils.traverse_dict*.Please use `salt.utils.data.traverse_dict()` instead.
- *salt.utils.traverse_dict_and_list*.Please use `salt.utils.data.traverse_dict_and_list()` instead.
- *salt.utils.filter_by*.Please use `salt.utils.data.filter_by()` instead.
- *salt.utils.subdict_match*.Please use `salt.utils.data.subdict_match()` instead.
- *salt.utils.substr_in_list*.Please use `salt.utils.data.substr_in_list()` instead.
- *salt.utils.is_dictlist*.Please use `salt.utils.data.is_dictlist()` instead.
- *salt.utils.repack_dictlist*.Please use `salt.utils.data.repack_dictlist()` instead.
- *salt.utils.compare_dicts*.Please use `salt.utils.data.compare_dicts()` instead.
- *salt.utils.compare_lists*.Please use `salt.utils.data.compare_lists()` instead.
- *salt.utils.decode_dict*.Please use `salt.utils.data.encode_dict()` instead.
- *salt.utils.decode_list*.Please use `salt.utils.data.encode_list()` instead.
- *salt.utils.exactly_n*.Please use `salt.utils.data.exactly_n()` instead.
- *salt.utils.exactly_one*.Please use `salt.utils.data.exactly_one()` instead.
- *salt.utils.is_list*.Please use `salt.utils.data.is_list()` instead.
- *salt.utils.is_iter*.Please use `salt.utils.data.is_iter()` instead.
- *salt.utils.isorted*.Please use `salt.utils.data.sorted_ignorecase()` instead.
- *salt.utils.is_true*.Please use `salt.utils.data.is_true()` instead.

  - *salt.utils.mysql_to_dict*.Please use `salt.utils.data.mysql_to_dict()` instead.
  - *salt.utils.simple_types_filter*.Please use `salt.utils.data.simple_types_filter()` instead.
  - *salt.utils.ip_bracket*.Please use `salt.utils.zeromq.ip_bracket()` instead.
  - *salt.utils.gen_mac*.Please use `salt.utils.network.gen_mac()` instead.
  - *salt.utils.mac_str_to_bytes*.Please use `salt.utils.network.mac_str_to_bytes()` instead.
  - *salt.utils.refresh_dns*.Please use `salt.utils.network.refresh_dns()` instead.
  - *salt.utils.dns_check*.Please use `salt.utils.network.dns_check()` instead.
  - *salt.utils.get_context*.Please use `salt.utils.stringutils.get_context()` instead.
  - *salt.utils.get_master_key*.Please use `salt.utils.master.get_master_key()` instead.
  - *salt.utils.get_values_of_matching_keys*.Please use `salt.utils.master.get_values_of_matching_keys()` instead.
  - *salt.utils.date_cast*.Please use `salt.utils.dateutils.date_cast()` instead.
  - *salt.utils.date_format*.Please use `salt.utils.dateutils.strftime()` instead.
  - *salt.utils.total_seconds*.Please use `salt.utils.dateutils.total_seconds()` instead.
  - *salt.utils.find_json*.Please use `salt.utils.json.find_json()` instead.
  - *salt.utils.import_json*.Please use `salt.utils.json.import_json()` instead.
  - *salt.utils.namespaced_function*.Please use `salt.utils.functools.namespaced_function()` instead.
  - *salt.utils.alias_function*.Please use `salt.utils.functools.alias_function()` instead.
  - *salt.utils.profile_func*.Please use `salt.utils.profile.profile_func()` instead.
  - *salt.utils.activate_profile*.Please use `salt.utils.profile.activate_profile()` instead.
  - *salt.utils.output_profile*.Please use `salt.utils.profile.output_profile()` instead.

salt.auth.Authorize Class Removal

- The salt.auth.Authorize Class inside of the *salt/auth/__init__.py* file has been removed and the *any_auth* method inside of the file *salt/utils/minions.py*. These method and classes were not being used inside of the salt code base.

28.2.2 Salt 3000.1 Release Notes

Version 3000.1 is a bugfix release for *3000*.

Statistics

- Total Merges: **53**
- Total Issue References: **15**
- Total PR References: **54**
- Contributors: **16** (Ch3LL, UtahDave, bryceml, cmcmarrow, dwoz, frogunder, garethgreenaway, lorengordon, mchugh19, oeuftete, raddessi, s0undt3ch, sjorge, terminalmage, twangboy, waynew)

Changelog for v3000..v3000.1

Generated at: 2020-03-27 16:48:41 UTC

- **PR** #56455: (s0undt3ch) Fix gitpython windows requirement @ *2020-03-27 16:31:57 UTC*
  - c5a700e01e Merge pull request #56455 from s0undt3ch/hotfix/gitpython
  - d9791c393f Revert and fix windows requirements
  - 4b573c1c94 Revert "Fix win deps"
- **PR** #56446: (frogunder) 3000.1 releasenotes updates @ *2020-03-24 20:28:23 UTC*
  - 7ba36325d9 Merge pull request #56446 from frogunder/releasenotes_3000.1_updates
  - 6b47f474af 3000.1 releasenotes updates
- **PR** #56435: (twangboy) Update mac build scripts @ *2020-03-24 19:47:40 UTC*
  - 4d2bc7004d Merge pull request #56435 from twangboy/update_mac_build_scripts
  - 5d244b0aa6 Merge branch 'master' into update_mac_build_scripts
  - 8fdf52a243 Update gitpython to 2.1.15
  - afcedc5232 Remove –ignore-installed
  - e6cc2a6192 Separate pyobjc requirements
  - 262ad2e98a Compile static requirements
  - 224f72d0f2 Update noxfile and .precommit
  - 68a36f2e37 Add req_ext.txt
  - 5851a5c2cd Roll back changes, fix dependencies
- **ISSUE** #56433: (Ch3LL) integration.states.test_pip_state.PipStateTest.test_pip_installed_specific_env failing on MAC (refs: #56436)
- **PR** #56436: (Ch3LL) Fix #56433 - test_pip_installed_specific_env @ *2020-03-24 19:47:20 UTC*
  - 8a8ae8ebe4 Merge pull request #56436 from Ch3LL/pip_custom_pypi
  - 55655ff96c Merge branch 'master' into pip_custom_pypi
- **PR** #56423: (Ch3LL) Update changelog with package fixes @ *2020-03-24 16:42:17 UTC*
  - 3a993d73a9 Merge pull request #56423 from Ch3LL/changelog_again
  - 963c16e1a7 update pr number for mac build changes in changelog
  - 83e22b77c7 Update changelog with package fixes
- **PR** #56417: (twangboy) Update mac build scripts @ *2020-03-23 18:45:34 UTC*
  - d2a5bd8add Merge pull request #56417 from twangboy/update_mac_build_scripts
  - 3bda8ddb82 Update noxfile and pre-commit
  - de58c52d66 Merge branch 'update_mac_build_scripts' of https://github.com/twangboy/salt into update_mac_build_scripts
    - e7f08d5349 Update static requirements
    - a53977de5b Merge branch 'update_mac_build_scripts' of github.com:twangboy/salt into update_mac_build_scripts
    - 04e5cde9dd *pkg/osx/req_ext.txt* no longer exists

    - be4a272d07 Update static requirements
  - 02dfe4119c Add pyobjc in its own requirements file
  - 6b2ac2be64 Remove pyobjc, enforce pip install location
  - 30ca5d04ec Remove cryptography, it gets installed by pyopenssl
  - fda8abf4a8 Update static requirements
  - 732ecc83c1 Update pre-commit config
  - 1fa9864e3d Don't set a version for pyopenssl
  - 461b198978 Update dependencies, remove req_ext.txt
- **PR** #56418: (Ch3LL) Ensure version.py included before we install @ *2020-03-23 18:27:46 UTC*
  - 74575a6993 Merge pull request #56418 from Ch3LL/egg_version
  - 85d7c784ef Fix integration setup egg test
  - a8a22a4040 Ensure version.py included before we install
  - 86fe450c82 Fix #56433 - test_pip_installed_specific_env
- **PR** #56403: (frogunder) update 3000.1 releasenotes @ *2020-03-17 23:50:21 UTC*
  - 249367b462 Merge pull request #56403 from frogunder/update_releasenotes_3000.1
  - 9d972c96e8 update 3000.1 releasenotes
- **PR** #56398: (Ch3LL) Add additional PRs to 3000.1 changelog @ *2020-03-17 18:08:15 UTC*
  - 0de5c1e136 Merge pull request #56398 from Ch3LL/change_3000.1
  - 79c337b3db Add additional PRs to 3000.1 changelog
- **PR** #56376: (twangboy) Fix win deps @ *2020-03-17 17:02:46 UTC*
  - 5ac09decb9 Merge pull request #56376 from twangboy/fix_win_deps
  - 6c83beeb9e Fix win deps
- **PR** #56378: (Ch3LL) Include _version.py if building wheel @ *2020-03-17 17:01:33 UTC*
  - e72a8d2cbc Merge pull request #56378 from Ch3LL/wheel_version
  - 22cccd2107 Use virtualenv test helper that already exists and fix setup.py
  - 293b1fddf2 cleanup code
  - ddfb065bfb add bdist_wheel test
  - fceff0287c ensure name is included in new version
  - 0cbf6d3246 Ensure SaltStackVersion attributes in _version.py correct
  - 39cdf5382d Include _version.py if building wheel
- **PR** #56387: (bryceml) update gpg key expiration date @ *2020-03-17 16:59:30 UTC*
  - 6a9326fb70 Merge pull request #56387 from bryceml/update_gpg_key
  - b74b26ccf3 update gpg key expiration date
- **PR** #55822: (cmcmarrow) fix_Indefinitely_code @ *2020-03-16 17:34:11 UTC*
  - fcee692c4a Merge pull request #55822 from cmcmarrow/win_task_repeat
  - e257fb2804 Merge branch 'master' into win_task_repeat

  - 07cada0f8f Merge branch 'master' into win_task_repeat
  - 4b80301338 Merge branch 'master' into win_task_repeat
  - 9df94569c4 Merge branch 'master' into win_task_repeat
  - 22a2d0b9ec Merge branch 'master' into win_task_repeat
  - dee9c134e7 Merge branch 'master' into win_task_repeat
  - beaf05d514 Update test_win_task.py
  - 6e923f75fc Merge branch 'master' into win_task_repeat
  - 5839da81ee add test
  - 2aa6338ed0 fix Indefinitely code
- **PR** #56373: (frogunder) update 3000.1 releasenotes @ *2020-03-13 18:58:43 UTC*
  - c11ef89200 Merge pull request #56373 from frogunder/releasenotes_3000.1
  - a5ca8b5277 Fix doc test errors
  - 47e483187e update 3000.1 releasenotes
- **PR** #56365: (Ch3LL) Update 3000.1 changelog @ *2020-03-13 17:21:02 UTC*
  - 62857a9f0e Merge pull request #56365 from Ch3LL/changelog_3000.1
  - 851d7d8fc7 Update 3000.1 changelog
- **PR** #56360: (Ch3LL) Use virtualenv 20.0.10 for macosx tests @ *2020-03-13 16:39:57 UTC*
  - a660e96fb2 Merge pull request #56360 from Ch3LL/mac_virtual
  - abda125086 Update static requirements files
  - dca3390c0c skip zcbuildout state test when virtualenv 20.0.0
  - fdeae1f0e1 Use virtualenv 20.0.10 for macosx tests
- **ISSUE** #56324: (kiemlicz) Cannot use Saltcheck module (refs: #56327)
- **PR** #56327: (mchugh19) keep cache_copied_files variable a list @ *2020-03-13 16:39:32 UTC*
  - d8fc07274c Merge pull request #56327 from mchugh19/56324
  - e1e2df2b1d Merge branch 'master' into 56324
  - 6d7572706d use includes for saltcheck integration test
  - 5b7073b01d keep cache_copied_files variable a list
- **PR** #56023: (cmcmarrow) add fix for bin_env @ *2020-03-13 16:39:09 UTC*
  - 2d31d61ab9 Merge pull request #56023 from cmcmarrow/binary_fix_pip_bin_env
  - f780fd013e Merge branch 'master' into binary_fix_pip_bin_env
  - d642c7c270 Merge branch 'master' into binary_fix_pip_bin_env
  - 82b39d26f8 Merge branch 'binary_fix_pip_bin_env' of https://github.com/cmcmarrow/salt into binary_fix_pip_bin_env
    - a5fa22528a Merge branch 'master' into binary_fix_pip_bin_env
  - 10ca0fa5d6 fix space
  - 22fff6c180 fix _pip_bin_env

  - d5425e6630 Merge branch 'master' into binary_fix_pip_bin_env
  - ceda3fde00 add full support for when missing kwargs or args
  - cf4ab215b5 add fix for bin_env
- **PR** #56310: (twangboy) Only process ADMX files when loading policies @ *2020-03-11 23:47:50 UTC*
  - 19bb6aae0c Merge pull request #56310 from twangboy/fix_lgpo_admx
  - 4bc5b05586 Fix some lint
  - ce282f9754 Mark destructive test
  - 8af09c31ca Add test case
  - 547d35cf4c Only process ADMX files when loading policies
- **PR** #56272: (twangboy) Properly resolve the policy name @ *2020-03-11 22:11:02 UTC*
  - 2d78931eaf Merge pull request #56272 from twangboy/fix_lgpo_names
  - 233ed12472 Add comments to helper function
  - a7369e41d7 Remove redundent code… some more
  - 1db3052b94 Consolidate duplicate code
  - b25b56299d Fix failing PY2 tests
  - 36a24ac28f Only load adml data once
  - bc13be6850 Add some tests
  - 2e9be6e461 Properly resolve the policy name
- **PR** #56358: (s0undt3ch) Fix version instantiation when minor is an empty string @ *2020-03-11 22:10:04 UTC*
  - c6c6e2e3d9 Merge pull request #56358 from s0undt3ch/hotfix/version-parsing
  - cb22e78502 Fix version instantiation when minor is an empty string
- **PR** #56160: (twangboy) Fix issue with existing reg_dword entries @ *2020-03-11 22:09:21 UTC*
  - b33047c574 Merge pull request #56160 from twangboy/fix_reg_dword
  - 80650c2a8c Make test more explicit
  - 67dd1c18c7 Fix issue with existing reg_dword entries
- **PR** #56337: (twangboy) Handle Adapter Type 53 and Undefined Types @ *2020-03-11 22:08:19 UTC*
  - 8f23706871 Merge pull request #56337 from twangboy/fix_win_network_type
  - 18a31a0b22 Merge branch 'master' into fix_win_network_type
  - d92914ad91 Merge branch 'master' into fix_win_network_type
  - 224629bdeb Handle Adapter Type 53 and Undefined Types
- **PR** #56060: (twangboy) Fix regex string for **Del and **DelVals @ *2020-03-11 22:05:43 UTC*
  - e551ff6a31 Merge pull request #56060 from twangboy/fix_lgpo_regex
  - ca260ef582 Use byte-strings in the test
  - 7c81874a5b Add some tests
  - 944b022f33 Fix regex string for **Del and **DelVals
- **ISSUE** #56131: (thusoy) Salt v3000 crashes with unless clause if pip is missing (refs: #56215)

- **PR** #56215: (dwoz) Fix for unless requisite when pip is not installed @ *2020-03-11 22:04:30 UTC*
  - c9bc9431c1 Merge pull request #56215 from dwoz/issue-56131
  - 5dd8f8eabe Fix linter
  - 7492c2f7c7 Remove crufty prints from test
  - 8ceaa6e6f3 Only remove pip relasted modules
  - 69b21e5e68 Add regression test for issue 56131
  - 9d23f55adc Fix for unless requisite when pip is not installed
- **PR** #56339: (twangboy) Fix win_dns_client when used with scheduler @ *2020-03-11 22:01:32 UTC*
  - 74b67ea741 Merge pull request #56339 from twangboy/fix_win_dns_client
  - d6a4fbe85d Fix bad test (lint)
  - 47bd67373f Add some tests
  - 995975649a Add missing import, improve docs
- **ISSUE** #56288: (lorengordon) win_lgpo: *lgpo.set* with secedit policy always generates *KeyError: u'lgpo.secedit_data'* (refs: #56290)
- **PR** #56290: (lorengordon) Ensures popping lgpo.secedit_data does not throw KeyError @ *2020-03-11 17:27:04 UTC*
  - 7701e8762d Merge pull request #56290 from lorengordon/pop-secdata
  - 62453bb6ce Merge branch 'master' into pop-secdata
  - 95d5cea9ff Merge branch 'master' into pop-secdata
  - 2979158a8b Tests the *if _secedits:* logic path in *lgpo.set_*
  - d1f776178c Ensures popping lgpo.secedit_data does not throw KeyError
- **ISSUE** #56119: (finalduty) Release notes for v3000 do not mention changes to *slspath* variable (refs: #56341)
- **PR** #56341: (dwoz) Revert "Don't remove one directory level from slspath" @ *2020-03-11 17:03:33 UTC*
  - 84c60708cd Merge pull request #56341 from dwoz/issue-56119
  - afe6e84c36 Clarify slspath documentation
  - 6dfc098fd1 Add debug for test asserts
  - 9fa9dab8b2 Fix linter warts
  - fd702bdd44 Add regression test for #56119
  - 3d3b673cec Revert "Don't remove one directory level from slspath"
- **PR** #56185: (terminalmage) Fix regression in service states with reload argument @ *2020-03-11 16:45:58 UTC*
  - 4f9813a49c Merge pull request #56185 from terminalmage/issue56167
  - 65b3f4c9a0 Merge branch 'master' into issue56167
  - 7b41a00b4f Merge branch 'master' into issue56167
  - 8f7034d946 Merge branch 'master' into issue56167
  - e9fbb634e1 Skip on OSX for now
  - 5996280241 Add __opts__ and __grains__ just in case
  - af3e841d08 Fix failing test

  - 586f21aedc Add functional test
  - 1afb9c10f0 Fix regression in service states with reload argument
- **PR** #56068: (s0undt3ch) Update the bootstrap script to latest version, v2020.02.24 @ *2020-03-11 16:44:23 UTC*
  - 74f8b2a926 Merge pull request #56068 from s0undt3ch/hotfix/update-bootstrap
  - ce83b190ed Merge branch 'master' into hotfix/update-bootstrap
  - ccd231d82d Merge branch 'master' into hotfix/update-bootstrap
  - 88a52f88ff Update the bootstrap script to latest version, v2020.02.24
  - 012fb5bc4b Update the bootstrap script to latest version, v2020.02.04
- **PR** #56321: (oeuftete) Tidy up formatting in boto_secgroup docs @ *2020-03-11 08:14:48 UTC*
  - 154257e2e9 Merge pull request #56321 from oeuftete/tidy-boto-secgroup-docs
  - 616f11b33e Merge branch 'master' into tidy-boto-secgroup-docs
  - 412bb4d62d Tidy up formatting in boto_secgroup docs
- **PR** #56336: (Ch3LL) Fix test_issue_2594_non_invalidated_cache test (update zope.interface) @ *2020-03-11 00:08:43 UTC*
  - 11d33e3d90 Merge pull request #56336 from Ch3LL/fix_56330
  - bbf37e090b Fix test_issue_2594_non_invalidated_cache test (update zope.interface)
- **PR** #56346: (frogunder) Update man pages to 3000.1 @ *2020-03-11 00:07:31 UTC*
  - a640bd30fc Merge pull request #56346 from frogunder/man_pages_3000_1
  - 6d7c1b6482 Update man pages to 3000.1
- **PR** #56099: (s0undt3ch) Fix Windows and macOS requirements handling in setup.py @ *2020-03-11 00:04:37 UTC*
  - 2f783d247e Merge pull request #56099 from s0undt3ch/hotfix/requirements
  - c19f4a3fae Merge branch 'master' into hotfix/requirements
  - 1e7bc8fc8f Also take into account macOS requirements
  - 7eef14952c Also include req_win.txt
- **PR** #56218: (raddessi) Changed StrictVersion checking of setuptools to LooseVersion @ *2020-03-10 23:59:54 UTC*
  - 7c4d879073 Merge pull request #56218 from raddessi/master.v3000-conda
  - 44556f6d57 Merge branch 'master' into master.v3000-conda
  - 566c03b786 Merge branch 'master' into master.v3000-conda
  - 82773a9799 Removed now-unused StrictVersion import from setup.py
  - 31bb0f7cd1 Changed StrictVersion checking of setuptools to LooseVersion
- **PR** #56128: (waynew) Update CHANGELOG with release date and unreleased. @ *2020-03-10 23:58:15 UTC*
  - acbd3556e5 Merge pull request #56128 from waynew/master
  - e3216db3e5 Update CHANGELOG with release date and unreleased.
- **PR** #55937: (twangboy) Update windows build scripts @ *2020-03-10 23:55:55 UTC*
  - 12140545ab Merge pull request #55937 from twangboy/update_deps

  - f00a504a48 Add back the pylauncher
  - 30b9c32356 Revert changes to req and req_win
  - 98dc0e970c Fix some warts in the build_env scripts
  - 8404141f65 Update dependencies
- **PR** #55906: (sjorge) smartos.vm_present could not handle nics with vrrp_vrid property @ *2020-03-10 23:54:44 UTC*
  - 485a47cdf1 Merge pull request #55906 from sjorge/smartos_vrrp
  - 5bd7dd009a Merge branch 'master' into smartos_vrrp
  - f77719c179 smartos state should handle vrrp config
- **ISSUE** #55185: (sjorge) salt.modules.pdbedit doesn't work on samba older than 4.8 (refs: #55894)
- **PR** #55894: (sjorge) #55185 pdbedit module should check for version 4.8.x or newer @ *2020-03-10 23:54:21 UTC*
  - 1fa8555360 Merge pull request #55894 from sjorge/pdbedit_55185
  - 9dc7b71122 Merge branch 'master' into pdbedit_55185
- **ISSUE** #56195: (lorengordon) Windows: Using inline powershell in args with *cmd.script* and *shell: powershell* (refs: #56197)
- **PR** #56197: (lorengordon) Allows use of inline powershell for cmd.script args @ *2020-03-10 23:52:47 UTC*
  - 3e57d58db2 Merge pull request #56197 from lorengordon/file-or-no-file
  - fcd1699f5e Allows use of inline powershell for cmd.script args
  - be2e67c0a0 Tests that powershell processes inline powershell in args
- **ISSUE** #53152: (jbeaird) daily highstate fails after 2019.2 upgrade (refs: #56149)
- **PR** #56149: (garethgreenaway) [master] Fix to scheduler for use of when and splay @ *2020-03-10 23:52:16 UTC*
  - 547c73e4cc Merge pull request #56149 from garethgreenaway/53152_fix_schedule_when_splay
  - 8f068f6f9b Fix for when using a combination of when and splay. Previously comparing the wrong value when determining if the job should be run and next_fire_time updated. This resulted in multiple job runs when *when* and *splay* were used together. Code updated and test updated to ensure only one run at the specific time. Skip eval tests is dateutil.parser is unavailable.
- **PR** #56345: (s0undt3ch) Bump Windows Py3 builds timeout to 10 hours @ *2020-03-10 20:43:41 UTC*
  - 192ce76a95 Merge pull request #56345 from s0undt3ch/hotfix/win-py3-timeouts
  - 11bdc38ae3 Bump Windows 2019 Py3 builds timeout to 10 hours
- **PR** #55888: (s0undt3ch) Disable codecov PR comments and status checks @ *2020-03-10 15:45:52 UTC*
  - a204906c80 Merge pull request #55888 from s0undt3ch/hotfix/coverage-reporting
  - 4b8dc8a586 Disable codecov PR comments and status checks
  - cb0f4dff87 Apply the suggestion given by the codecov team
- **ISSUE** #56177: (jodok) mysql states fail because conv is ‘‘ instead of None (refs: #56174)
- **ISSUE** #56170: (jeffdyke) mariadb socket access must be enabled before highstate - salt 3K still tries empty password (refs: #56174)
- **ISSUE** #56124: (ymasson) MySQL state and module broken after upgrade to 3000 (refs: #56174)
- **PR** #56174: (garethgreenaway) [master] MySQL module fixes @ *2020-03-10 04:03:23 UTC*

  - 3e913631bb Merge pull request #56174 from garethgreenaway/56124_mysql_module_state_fixes
  - fcc061368b Removing quotes from the plugin_status query. Updating tests to reflect changes.
  - 3dc66393b2 Adding better error reporting around plugins. Updating tests. Only attempt to delete a user if they exist.
  - 1337da1e4e Ensure _mysql_user_exists is using auth_socket. Updating mysql and mariadb chpass functions to ensure that the respective plugins are enabled before attempting to use them.
  - 34995ba4e8 Reworking the unix_socket code to support the differences between MySQL and MariaDB. Adding some functions to install, remove, and check the status of plugins which we can then use when adding users which will use the unix_socket & auth_socket plugins. Adding additional tests for these new functions as well as test to ensure the correct SQL is being generated when using passwordless and unix_socket options.
  - 5bfd67c13e Minor tweak to mysql module. Fixing failing tests.
  - e871a3ffd1 Various fixes to the mysql module to break out the handling of user management into different functions based on MySQL variant.
- **ISSUE** #56063: (terminalmage) [master] Traceback in esxi grain module on import (refs: #56094)
- **PR** #56094: (dwoz) Fix type error in TornadoImporter @ *2020-03-10 01:39:08 UTC*
  - 211c88bfbc Merge pull request #56094 from dwoz/fix_56063
  - 7b1632e8e3 Fix type error in TornadoImporter
- **PR** #56172: (Ch3LL) Only change mine data if using new allow_tgt feature @ *2020-03-10 01:34:27 UTC*
  - fb5252fc53 Merge pull request #56172 from Ch3LL/mine_g
  - f4c9c2a5cf Fix docs
  - 6c914caec8 Use different targeting for windows/linux
  - f6348127dc Only change mine data if using new allow_tgt feature
- **ISSUE** #56121: (githubcdr) salt-minion broken after upgrade to 3000 (refs: #56143)
- **ISSUE** #51854: (Oloremo) Fluorine: minion_pillar_cache: True leads to exception (refs: #56143, #52195)
- **PR** #56143: (waynew) Use encoding when caching pillar data @ *2020-03-10 01:33:37 UTC*
  - **PR** #52195: (waynew) Use encoding when caching pillar data (refs: #56143)
  - 8a8e9c9c5f Merge pull request #56143 from waynew/51854-minion-pillar-cache-exception
  - 58cc9488aa Merge branch 'master' into 51854-minion-pillar-cache-exception
- **PR** #56082: (Ch3LL) Fix saltversioninfo grain for new version @ *2020-03-10 01:32:11 UTC*
  - 9f27caa7d0 Merge pull request #56082 from Ch3LL/ver_grains
  - e6abd6d31b ensure full_info/noc_info work with new versioning
  - bcc520ccc4 Add saltversioninfo grains test
  - 510e149b87 Fix saltversioninfo grain for new version
- **PR** #56285: (UtahDave) Add missing colon. @ *2020-03-09 22:22:17 UTC*
  - 602ff3b9f1 Merge pull request #56285 from UtahDave/fix_f5_doc
  - 1034013831 Add missing colon.
- **PR** #56333: (Ch3LL) add pylint ignore in django returner @ *2020-03-09 20:42:42 UTC*

  - de5184a206 Merge pull request #56333 from Ch3LL/lint_fix
  - 6a213a429a add pylint ignore in django returner
  - 759290a055 Use encoding when caching pillar data
- **ISSUE** #56080: (sagetherage) Update release notes, pip download page and install docs on pycrpto (refs: #56095)
- **PR** #56095: (waynew) Provide security advisory for PyCrypto @ *2020-02-10 18:42:00 UTC*
  - 9adc2214c3 Merge pull request #56095 from waynew/crypto-warning
  - 484bc51f4b Add warnings to a couple of other places
  - 2711c04ca9 Provide security advisory for PyCrypto
- **PR** #56092: (twangboy) Add LGPO and Network PRs to changelogs @ *2020-02-10 18:17:28 UTC*
  - 1acd492bb9 Merge pull request #56092 from twangboy/waynew-update-changelog
  - 4e03620d4e Fix a docs issue
  - e7b64277e1 Add LGPO and Network PRs to changelogs
- **PR** #56115: (s0undt3ch) Add information about the pip install salt on windows issue. @ *2020-02-10 17:33:44 UTC*
  - 65d59b0ee9 Merge pull request #56115 from s0undt3ch/hotfix/release-notes
  - aeac9f36cd Add information about the pip install salt on windows issue.
  - a5179434e7 Merge branch 'master' into pdbedit_55185
  - 95d46d6cc8 #55185 pdbedit module should check for version 4.8.x or newer

28.2.3 Salt 3000.2 Release Notes

Version 3000.2 is a CVE-fix release for *3000*.

Security Fix

CVE-2020-11651

An issue was discovered in SaltStack Salt before 2019.2.4 and 3000 before 3000.2. The salt-master process ClearFuncs class does not properly validate method calls. This allows a remote user to access some methods without authentication. These methods can be used to retrieve user tokens from the salt master and/or run arbitrary commands on salt minions.

CVE-2020-11652

An issue was discovered in SaltStack Salt before 2019.2.4 and 3000 before 3000.2. The salt-master process ClearFuncs class allows access to some methods that improperly sanitize paths. These methods allow arbitrary directory access to authenticated users.

Known Issue

Part of the fix for CVE-2020-11651 added better validation of the methods allowed to be called by remote clients. Both AESFuncs and ClearFuncs now have an explicit list of methods that can be called. The name of one of these whitlisted methods on AESFuncs had a typo. The _minion_runner method should be minion_runner (without the underscore prefix). This typo breaks the publish module's runner method. Calling runners, for example:

```
salt minion publish.runner manage.down
```

Will not work, and you will receive and empty reply from the salt master.

This will be addressed in the Sodium release of Salt set for mid-June 2020.

28.2.4 Salt 3000.3 Release Notes

Version 3000.3 is a bug-fix release for *3000*.

Statistics

- Total Merges: **2**
- Total Issue References: **2**
- Total PR References: **2**
- Contributors: **2** (dwoz, frogunder)

Changelog for v3000.2..v3000.3

Generated at: 2020-05-06 02:53:12 UTC

- **PR** #57097: (frogunder) Update man_pages 3000.3 @ *2020-05-05 22:13:09 UTC*
- **ISSUE** #57027: (ecarson) [BUG] Master running 2019.2.4 or 3000.2 unable to synchronize files using saltutil.sync_all to 2017.7.1 minion due to CVE fix (refs: #57100)
- **ISSUE** #57016: (idontwanttosignin) [BUG] Requested method not exposed: minion_runner (refs: #57100)
- **PR** #57100: (dwoz) Address Issues in CVE Release @ *2020-05-05 22:09:25 UTC*

28.2.5 Salt 2019.2.0 Release Notes - Codename Fluorine

Python 2.7 Deprecation

In light of Python 2.7 reaching its End of Life (EOL) on Jan 1st 2020, Python 2 will be deprecated from SaltStack no earlier than the Sodium release, that is either the Sodium release or a later release. This decision is pending further community discussion.

Non-Backward-Compatible Change to YAML Renderer

In earlier releases, this was considered valid usage in Python 2, assuming that `data` was a list or dictionary containing keys/values which are `unicode` types:

```
/etc/foo.conf:
  file.managed:
    - source: salt://foo.conf.jinja
    - template: jinja
    - context:
        data: {{ data }}
```

One common use case for this is when using one of Salt's *custom Jinja filters* which return lists or dictionaries, such as the *ipv4* filter.

In Python 2, Jinja will render the `unicode` string types within the list/dictionary with the "u" prefix (e.g. `{u'foo': u'bar'}`). While not valid YAML, earlier releases would successfully load these values.

As of this release, the above SLS would result in an error message. To allow for a data structure to be dumped directly into your SLS file, use the tojson Jinja filter:

```
/etc/foo.conf:
  file.managed:
    - source: salt://foo.conf.jinja
    - template: jinja
    - context:
        data: {{ data|tojson }}
```

Another example where the new filter needs to be used is the following state example:

```
grafana_packages:
  pkg.installed:
  - names: {{ server.pkgs }}
```

This will fail when pkgs is a list or dictionary. You will need to update the state:

```
grafana_packages:
  pkg.installed:
  - names: {{ server.pkgs|tojson }}
```

This test case has also been tested with the `yaml` and `json` filters successfully.

Note: This filter was added in Jinja 2.9. However, fear not! The 2018.3.3 release added a `tojson` filter which will be used if this filter is not already present, making it available on platforms like RHEL 7 and Ubuntu 14.04 which provide older versions of Jinja.

Important: The *json_encode_dict* and *json_encode_list* filters do not actually dump the results to JSON. Since `tojson` accomplishes what those filters were designed to do, they are now deprecated and will be removed in the 3000 release. The `tojson` filter should be used in all cases where *json_encode_dict* and *json_encode_list* would have been used.

Ansible Playbook State and Execution Modules

Along with the including the `ansible modules` in the Oxygen release, running playbooks has been added in 2019.2.0 with the *playbooks function*. This also includes an *ansible playbooks state module* which can be used on a targeted host to run ansible playbooks, or used in an orchestration state runner.

```
install nginx:
  ansible.playbooks:
    - name: install.yml
    - git_repo: git://github.com/gtmanfred/playbook.git
    - git_kwargs:
        rev: master
```

The playbooks modules also includes the ability to specify a git repo to clone and use, or a specific directory can to used when running the playbook.

Network Automation

Beginning with this release, Salt provides much broader support for a variety of network operating systems, and features for configuration manipulation or operational command execution.

NetBox

Added in the previous release, 2018.3.0, the capabilities of the `netbox` Execution Module have been extended, with a much longer list of available features:

- `netbox.create_circuit`
- `netbox.create_circuit_provider`
- `netbox.create_circuit_termination`
- `netbox.create_circuit_type`
- `netbox.create_device`
- `netbox.create_device_role`
- `netbox.create_device_type`
- `netbox.create_interface`
- `netbox.create_interface_connection`
- `netbox.create_inventory_item`
- `netbox.create_ipaddress`
- `netbox.create_manufacturer`
- `netbox.create_platform`
- `netbox.create_site`
- `netbox.delete_interface`
- `netbox.delete_inventory_item`
- `netbox.delete_ipaddress`
- `netbox.get_circuit_provider`
- `netbox.get_interfaces`
- `netbox.get_ipaddresses`
- `netbox.make_interface_child`
- `netbox.make_interface_lag`
- `netbox.openconfig_interfaces`
- `netbox.openconfig_lacp`
- `netbox.update_device`
- `netbox.update_interface`

Besides this Execution Module, Salt users can load data directly from NetBox into the device Pillar, via the `netbox` External Pillar module.

Netmiko

Netmiko, the multi-vendor library to simplify Paramiko SSH connections to network devices, is now officially integrated into Salt. The network community can use it via the *netmiko* Proxy Module or directly from any Salt Minions, passing the connection credentials - see the documentation for the *netmiko* Execution Module.

Arista

Arista switches can now be managed running under the *pyeapi* Proxy Module, and execute RPC requests via the *pyeapi* Execution Module.

Cisco Nexus

While support for SSH-based operations has been added in the release codename Carbon (2016.11), the new *nxos_api* Proxy Module and *nxos_api* allow management of Cisco Nexus switches via the NX-API.

It is important to note that these modules don't have third party dependencies, therefore they can be used straight away from any Salt Minion. This also means that the user may be able to install the regular Salt Minion on the Nexus switch directly and manage the network devices like a regular server.

General-purpose Modules

The new *ciscoconfparse* Execution Module can be used for basic configuration parsing, audit or validation for a variety of network platforms having Cisco IOS style configuration (one space indentation), as well as brace-delimited configuration style.

The *iosconfig* can be used for various configuration manipulation for Cisco IOS style configuration, such as: *configuration cleanup*, *tree representation of the config*, etc.

NAPALM

Commit At and Commit Confirmed

Beginning with this release, NAPALM users are able to execute scheduled commits (broadly known as "commit at") and "commit confirmed" (revert the configuration change unless the user confirms by running another command). These features are available via the `commit_in`, `commit_at`, `revert_in`, or `revert_at` arguments for the *net.load_config* and *net.load_template* execution functions, or *netconfig.managed*.

The counterpart execution functions *net.confirm_commit*, or *net.cancel_commit*, as well as the State functions *netconfig.commit_cancelled*, or *netconfig.commit_confirmed* can be used to confirm or cancel a commit.

Please note that the commit confirmed and commit cancelled functionalities are available for any platform whether the network devices supports the features natively or not. However, be cautious and make sure you read and understand the caveats before using them in production.

Multiple Templates Rendered Simultaneously

The `template_name` argument of the *net.load_template* Execution and *netconfig.managed* State function now supports a list of templates. This is particularly useful when a very large Jinja template is split into multiple smaller and easier to read templates that can eventually be reused in other States. For example, the following syntax is not correct to manage the configuration of NTP and BGP simultaneously, using two different templates and changing the device configuration through one single commit:

```
manage_bgp_and_ntp:
  netconfig.managed:
    - template_name:
        - salt://templates/bgp.jinja
        - salt://templates/ntp.jinja
    - context:
        bpg: {{ pillar.bgp }}
        ntp: {{ pillar.ntp }}
```

Connection Re-establishment on Demand

Beginning with this release, any NAPALM command executed when running under a NAPALM Proxy Minion supports the `force_reconnect` magic argument.

Proxy Minions generally establish a connection with the remote network device at the time of the Minion startup and that connection is going to be used forever.

If one would need to execute a command on the device but is connecting using different parameters (due to various causes, e.g., unable to authenticate the user specified in the Pillar as the authentication system - say TACACS+ is not available, or the DNS resolver is currently down and would like to temporarily use the IP address instead, etc.), it implies updating the Pillar data and restarting the Proxy Minion process restart. In particular cases like that, you can pass the `force_reconnect=True` keyword argument, together with the alternative connection details, to enforce the command to be executed over a separate connection.

For example, if the usual command is `salt '*' net.arp`, you can use the following to connect using a different username instead:

```
salt '*' net.arp username=my-alt-usr force_reconnect=True
```

The same goes with any of the other configuration arguments required for the NAPALM connection - see *NAPALM proxy documentation*.

Configuration Replace Features

To replace various configuration chunks, you can use the new *net.replace_pattern* execution function, or the *netconfig.replace_pattern* State function. For example, if you want to update your configuration and rename a BGP policy referenced in many places, you can do so by running:

```
salt '*' net.replace_pattern OLD-POLICY-CONFIG new-policy-config
```

Similarly, you can also replace entire configuration blocks using the *net.blockreplace* function.

Configuration Save Features

The *net.save_config* function can be used to save the configuration of the managed device into a file. For the State subsystem, the *netconfig.saved* function has been added which provides a complete list of facilities when managing the target file where the configuration of the network device can be saved.

For example, backup the running configuration of each device under its own directory tree:

```
/var/backups/{{ opts.id }}/running.cfg:
  netconfig.saved:
    - source: running
    - makedirs: true
```

All the new network automation modules mentioned above are directly exposed to the NAPALM users, without requiring any architectural changes, just eventually install some requirements:

Junos

The features from the existing *junos* Execution Module are available via the following functions:

- *napalm.junos_cli*: Execute a CLI command and return the output as text or Python dictionary.
- *napalm.junos_rpc*: Execute an RPC request on the remote Junos device, and return the result as a Python dictionary, easy to digest and manipulate.
- *napalm.junos_install_os*: Install the given image on the device.
- *napalm.junos_facts*: The complete list of Junos facts collected by the `junos-eznc` underlying library.

Note: To be able to use these features, you muse ensure that you meet the requirements for the *junos* module. As `junos-eznc` is already a dependency of NAPALM, you will only have to install `jxmlease`.

Usage examples:

```
salt '*' napalm.junos_cli 'show arp' format=xml
salt '*' napalm.junos_rpc get-interface-information
```

Netmiko

The features from the newly added *netmiko* Execution Module are available as:

- *napalm.netmiko_commands*: Execute one or more commands to be execute on the remote device, via Netmiko, and return the output as a text.
- *napalm.netmiko_config*: Load a list of configuration command on the remote device, via Netmiko. The commands can equally be loaded from a local or remote path, and passed through Salt's template rendering pipeline (by default using `Jinja` as the template rendering engine).

Usage examples:

```
salt '*' napalm.netmiko_commands 'show version' 'show interfaces'
salt '*' napalm.netmiko_config config_file=https://bit.ly/2sgljCB
```

Arista pyeapi

For various operations and various extension modules, the following features have been added to gate functionality from the *pyeapi* module:

- *napalm.pyeapi_run_commands*: Execute a list of commands on the Arista switch, via the `pyeapi` library.

- *napalm.pyeapi_config*: Configure the Arista switch with the specified commands, via the `pyeapi` Python library. Similarly to *napalm.netmiko_config*, you can use both local and remote files, with or without templating.

Usage examples:

```
salt '*' napalm.pyeapi_run_commands 'show version' 'show interfaces'
salt '*' napalm.pyeapi_config config_file=salt://path/to/template.jinja
```

Cisco NX-API

In the exact same way as above, the user has absolute control by using the following primitives to manage Cisco Nexus switches via the NX-API:

- *napalm.nxos_api_show*: Execute one or more show (non-configuration) commands, and return the output as plain text or Python dictionary.
- *napalm.nxos_api_rpc*: Execute arbitrary RPC requests via the Nexus API.
- *napalm.nxos_api_config*: Configures the Nexus switch with the specified commands, via the NX-API. The commands can be loaded from the command line, or a local or remote file, eventually rendered using the templating engine of choice (default: `jinja`).

Usage examples:

```
salt '*' napalm.nxos_api_show 'show bgp sessions' 'show processes' raw_text=False
```

Ciscoconfparse

The following list of function may be handy when manipulating Cisco IOS or Junos style configurations:

- *napalm.config_filter_lines*: Return a list of detailed matches, for the configuration blocks (parent-child relationship) whose parent and children respect the regular expressions provided.
- *napalm.config_find_lines*: Return the configuration lines that match the regular expression provided.
- *napalm.config_lines_w_child*: Return the configuration lines that match a regular expression, having child lines matching the child regular expression.
- *napalm.config_lines_wo_child*: Return the configuration lines that match a regular expression, that don't have child lines matching the child regular expression.

Note: These functions require the `ciscoconfparse` Python library to be installed.

Usage example (find interfaces that are administratively shut down):

```
salt '*' napalm.config_lines_w_child 'interface' 'shutdown'
```

IOSConfig

For Cisco IOS style configuration, the following features have been added to the *napalm* Execution Module:

- *napalm.config_tree*: Transform Cisco IOS style configuration to structured Python dictionary, using the configuration of the interrogated network device.

- *napalm.config_merge_tree*: Return the merge tree of the configuration of the managed network device with a different configuration to be merged with (without actually loading any changes on the device).
- *napalm.config_merge_text*: Return the merge result (as text) of the configuration of the managed network device with a different configuration to be merged with.
- *napalm.config_merge_diff*: Return the merge diff after merging the configuration of the managed network device with a different configuration (without actually loading any changes on the device).

SCP

Reusing the already available connection credentials provided for NAPALM, the following features are now available:

- *napalm.scp_put*: Transfer files and directories to remote network device.
- *napalm.scp_get*: Transfer files and directories from remote network device to the localhost of the Minion.

PeeringDB

The *peeringdb* Execution Module is useful to gather information about other networks you can potentially peer with, and automatically establish BGP sessions, e.g., given just a specific AS number, the rest of the data (i.e., IP addresses, locations where the remote network is available, etc.) is retrieved from PeeringDB, and the session configuration is automated with minimal to no effort (typing the IP addresses manually can be both tedious and error prone).

New Docker Proxy Minion

Docker containers can now be treated as actual minions without installing salt in the container, using the new `docker proxy minion`.

This proxy minion uses the *docker executor* to pass commands to the docker container using *docker.call*. Any state module calls are passed through the corresponding function from the *docker* module.

```
proxy:
  proxytype: docker
  name: keen_proskuriakova
```

Terraform Salt-SSH Roster

You can now dynamically generate a Salt-SSH roster from the terraform resources defined with terraform-provider-salt.

This allows you to combine both terraform and Salt-SSH to provision and configure your hosts. See the *terraform roster* for an example on how to setup and use.

Grains Dictionary Passed into Custom Grains

Starting in this release, if a custom grains function accepts a variable named `grains`, the Grains dictionary of the already compiled grains will be passed in. Because of the non-deterministic order that grains are rendered in, the only grains that can be relied upon to be passed in are `core.py` grains, since those are compiled first.

More Precise `virtual` Grain

This release improves the accuracy of the `virtual` grain when running Salt in a nested virtualization environment (e.g. `systemd-nspawn` container inside a VM) and having `virt-what` installed.

Until now, the `virtual` grain was determined by matching against all output lines of `virt-what` instead of individual items which could lead to not quite precise results (e.g. reporting `HyperV` inside a `systemd-nspawn` container running within a Hyper-V-based VM.

Configurable Module Environment

Salt modules (states, execution modules, returners, etc.) now can have custom environment variables applied when running shell commands. This can be configured by setting a `system-environment` key either in Grains or Pillar. The syntax is as follows:

```
system-environment:
  <type>
    <module>:
      # Namespace for all functions in the module
      _:
        <key>: <value>

      # Namespace only for particular function in the module
      <function>:
        <key>: <value>
```

- `<type>` would be the type of module (i.e. `states`, `modules`, etc.).
- `<module>` would be the module's name.

> **Note:** The module name can be either the virtual name (e.g. `pkg`), or the physical name (e.g. `yumpkg`).

- `<function>` would be the function name within that module. To apply environment variables to *all* functions in a given module, use an underscore (i.e. `_`) as the function name. For example, to set the same environment variable for all package management functions, the following could be used:

```
system-environment:
  modules:
    pkg:
      _:
        SOMETHING: for_all
```

To set an environment variable in `pkg.install` only:

```
system-environment:
  modules:
    pkg:
      install:
        LC_ALL: en_GB.UTF-8
```

To set the same variable but only for SUSE minions (which use zypper for package management):

```
system-environment:
  modules:
    zypper:
```

(continues on next page)

(continued from previous page)

```
      install:
        LC_ALL: en_GB.UTF-8
```

Note: This is not supported throughout Salt; the module must explicitly support this feature (though this may change in the future). As of this release, the only modules which support this are the following `pkg` virtual modules:

- *`aptpkg`*
- *`yumpkg`*
- `zypper`

"Virtual Package" Support Dropped for APT

In APT, some packages have an associated list of packages which they provide. This allows one to do things like run `apt-get install foo` when the real package name is `foo1.0`, and get the right package installed.

Salt has traditionally designated as "virtual packages" those which are provided by an installed package, but for which there is no real package by that name installed. Given the above example, if one were to run a *`pkg.installed`* state for a package named `foo`, then *`pkg.list_pkgs`* would show a package version of simply `1` for package `foo`, denoting that it is a virtual package.

However, while this makes certain aspects of package management convenient, there are issues with this approach that make relying on "virtual packages" problematic. For instance, Ubuntu has four different mutually-conflicting packages for `nginx`:

- nginx-core
- nginx-full
- nginx-light
- nginx-extras

All four of these provide `nginx`. Yet there is an nginx package as well, which has no actual content and merely has dependencies on any one of the above four packages. If one used `nginx` in a *`pkg.installed`* state, and none of the above four packages were installed, then the nginx metapackage would be installed, which would pull in *nginx-core_*. Later, if `nginx` were used in a *`pkg.removed`* state, the nginx metapackage would be removed, leaving nginx-core installed. The result would be that, since *nginx-core_* provides *nginx_*, Salt would now see nginx as an installed virtual package, and the *`pkg.removed`* state would fail. Moreover, *nginx would not actually have been removed*, since nginx-core would remain installed.

Starting with this release, Salt will no longer support using "virtual package" names in `pkg` states, and package names will need to be specified using the proper package name. The *`pkg.list_repo_pkgs`* function can be used to find matching package names in the repositories, given a package name (or glob):

```
# salt myminion pkg.list_repo_pkgs 'nginx*'
myminion:
    ----------
    nginx:
        - 1.10.3-0ubuntu0.16.04.2
        - 1.9.15-0ubuntu1
    nginx-common:
        - 1.10.3-0ubuntu0.16.04.2
        - 1.9.15-0ubuntu1
    nginx-core:
```

(continues on next page)

(continued from previous page)

```
        - 1.10.3-0ubuntu0.16.04.2
        - 1.9.15-0ubuntu1
    nginx-core-dbg:
        - 1.10.3-0ubuntu0.16.04.2
        - 1.9.15-0ubuntu1
    nginx-doc:
        - 1.10.3-0ubuntu0.16.04.2
        - 1.9.15-0ubuntu1
    nginx-extras:
        - 1.10.3-0ubuntu0.16.04.2
        - 1.9.15-0ubuntu1
    nginx-extras-dbg:
        - 1.10.3-0ubuntu0.16.04.2
        - 1.9.15-0ubuntu1
    nginx-full:
        - 1.10.3-0ubuntu0.16.04.2
        - 1.9.15-0ubuntu1
    nginx-full-dbg:
        - 1.10.3-0ubuntu0.16.04.2
        - 1.9.15-0ubuntu1
    nginx-light:
        - 1.10.3-0ubuntu0.16.04.2
        - 1.9.15-0ubuntu1
    nginx-light-dbg:
        - 1.10.3-0ubuntu0.16.04.2
        - 1.9.15-0ubuntu1
```

Alternatively, the newly-added *pkg.show* function can be used to get more detailed information about a given package and help determine what package name is correct:

```
# salt myminion pkg.show 'nginx*' filter=description,provides
myminion:
    ----------
    nginx:
        ----------
        1.10.3-0ubuntu0.16.04.2:
            ----------
            Description:
                small, powerful, scalable web/proxy server
        1.9.15-0ubuntu1:
            ----------
            Description:
                small, powerful, scalable web/proxy server
    nginx-common:
        ----------
        1.10.3-0ubuntu0.16.04.2:
            ----------
            Description:
                small, powerful, scalable web/proxy server - common files
        1.9.15-0ubuntu1:
            ----------
            Description:
                small, powerful, scalable web/proxy server - common files
    nginx-core:
        ----------
        1.10.3-0ubuntu0.16.04.2:
```

(continues on next page)

(continued from previous page)

```
            ----------
            Description:
                nginx web/proxy server (core version)
            Provides:
                httpd, httpd-cgi, nginx
        1.9.15-0ubuntu1:
            ----------
            Description:
                nginx web/proxy server (core version)
            Provides:
                httpd, httpd-cgi, nginx
    nginx-core-dbg:
        ----------
        1.10.3-0ubuntu0.16.04.2:
            ----------
            Description:
                nginx web/proxy server (core version) - debugging symbols
        1.9.15-0ubuntu1:
            ----------
            Description:
                nginx web/proxy server (core version) - debugging symbols
    nginx-doc:
        ----------
        1.10.3-0ubuntu0.16.04.2:
            ----------
            Description:
                small, powerful, scalable web/proxy server - documentation
        1.9.15-0ubuntu1:
            ----------
            Description:
                small, powerful, scalable web/proxy server - documentation
    nginx-extras:
        ----------
        1.10.3-0ubuntu0.16.04.2:
            ----------
            Description:
                nginx web/proxy server (extended version)
            Provides:
                httpd, httpd-cgi, nginx
        1.9.15-0ubuntu1:
            ----------
            Description:
                nginx web/proxy server (extended version)
            Provides:
                httpd, httpd-cgi, nginx
    nginx-extras-dbg:
        ----------
        1.10.3-0ubuntu0.16.04.2:
            ----------
            Description:
                nginx web/proxy server (extended version) - debugging symbols
        1.9.15-0ubuntu1:
            ----------
            Description:
                nginx web/proxy server (extended version) - debugging symbols
    nginx-full:
```

(continues on next page)

(continued from previous page)

```
        ----------
        1.10.3-0ubuntu0.16.04.2:
            ----------
            Description:
                nginx web/proxy server (standard version)
            Provides:
                httpd, httpd-cgi, nginx
        1.9.15-0ubuntu1:
            ----------
            Description:
                nginx web/proxy server (standard version)
            Provides:
                httpd, httpd-cgi, nginx
    nginx-full-dbg:
        ----------
        1.10.3-0ubuntu0.16.04.2:
            ----------
            Description:
                nginx web/proxy server (standard version) - debugging symbols
        1.9.15-0ubuntu1:
            ----------
            Description:
                nginx web/proxy server (standard version) - debugging symbols
    nginx-light:
        ----------
        1.10.3-0ubuntu0.16.04.2:
            ----------
            Description:
                nginx web/proxy server (basic version)
            Provides:
                httpd, httpd-cgi, nginx
        1.9.15-0ubuntu1:
            ----------
            Description:
                nginx web/proxy server (basic version)
            Provides:
                httpd, httpd-cgi, nginx
    nginx-light-dbg:
        ----------
        1.10.3-0ubuntu0.16.04.2:
            ----------
            Description:
                nginx web/proxy server (basic version) - debugging symbols
        1.9.15-0ubuntu1:
            ----------
            Description:
                nginx web/proxy server (basic version) - debugging symbols
```

Minion Startup Events

When a minion starts up it sends a notification on the event bus with a tag that looks like this: `salt/minion/<minion_id>/start`. For historical reasons the minion also sends a similar event with an event tag like this: `minion_start`. This duplication can cause a lot of clutter on the event bus when there are many minions. Set `enable_legacy_startup_events: False` in the minion config to ensure only the `salt/minion/<minion_id>/start` events are sent.

The new *`enable_legacy_startup_events`* minion config option defaults to `True`, but will be set to default to `False` beginning with the Sodium release of Salt.

The Salt Syndic currently sends an old style `syndic_start` event as well. The syndic respects *`enable_legacy_startup_events`* as well.

Failhard changes

It is now possible to override a global failhard setting with a state-level failhard setting. This is most useful in case where global failhard is set to `True` and you want the execution not to stop for a specific state that could fail, by setting the state level failhard to `False`. This also allows for the use of `onfail*`-requisites, which would previously be ignored when a global failhard was set to `True`. This is a deviation from previous behavior, where the global failhard setting always resulted in an immediate stop whenever any state failed (regardless of whether the failing state had a failhard setting of its own, or whether any `onfail*`-requisites were used).

Pass Through Options to `file.serialize` State

This allows for more granular control over the way in which the dataset is serialized. See the documentation for the new `serializer_opts` and `deserializer_opts` options in the *`file.serialize`* state for more information.

`file.patch` State Rewritten

The `file.patch` state has been rewritten with several new features:

- Patch sources can now be remote files instead of only `salt://` URLs
- Multi-file patches are now supported
- Patch files can be templated

In addition, it is no longer necessary to specify what the hash of the patched file should be.

New no_proxy Minion Configuration

Pass a list of hosts using the `no_proxy` minion config option to bypass an HTTP proxy.

Note: This key does nothing unless proxy_host is configured and it does not support any kind of wildcards.

```
no_proxy: [ '127.0.0.1', 'foo.tld' ]
```

Changes to `slack` Engine

The output returned to Slack from functions run using this engine is now formatted using that function's proper outputter. Earlier releases would format the output in YAML for all functions except for when states were run.

Enhancements to `wtmp` Beacon

A new key, `action`, has been added to the events fired by this beacon, which will contain either the string `login` or `logout`. This will simplify reactors which use this beacon's data, as it will no longer be necessary to check the integer value of the `type` key to know whether the event is a login or logout.

Additionally, in the event that your platform has a non-standard `utmp.h`, you can now configure which type numbers indicate a login and logout.

See the *wtmp beacon documentation* for more information.

Deprecated and Removed Options

API Removed Arguments

Support for *LocalClient*'s `expr_form` argument has been removed. Please use `tgt_type` instead. This change was made due to numerous reports of confusion among community members, since the targeting method is published to minions as `tgt_type`, and appears as `tgt_type` in the job cache as well.

Those who are using the *LocalClient* (either directly, or implicitly via a *netapi module*) need to update their code to use `tgt_type`.

```
>>> import salt.client
>>> local = salt.client.LocalClient()
>>> local.cmd('*', 'cmd.run', ['whoami'], tgt_type='glob')
{'jerry': 'root'}
```

Minion Configuration Deprecated Option

The *master_shuffle* configuration option is deprecated as of the `2019.2.0` release. Please use the *random_master* option instead.

Module Removed Options

- The *napalm_network* module has been changed as follows:
    - Support for the `template_path` has been removed from *net.load_template* function. This is because support for NAPALM native templates has been dropped.
- The *pip* module has been changed as follows:
    - Support for the `no_chown` option has been removed from *pip.install* function.
- The *trafficserver* module has been changed as follows:
    - The `trafficserver.match_var` function was removed. Please use *trafficserver.match_metric* instead.
    - The `trafficserver.read_var` function was removed. Please use *trafficserver.read_config* instead.
    - The `trafficserver.set_var` function was removed. Please use *trafficserver.set_config* instead.
- The `win_update` module has been removed. It has been replaced by *win_wua*.
- The *win_wua* module has been changed as follows:

    - The `win_wua.download_update` and `win_wua.download_updates` functions have been removed. Please use *win_wua.download* instead.
    - The `win_wua.install_update` and `win_wua.install_updates` functions have been removed. Please use *win_wua.install* instead.
    - The `win_wua.list_update` function has been removed. Please use functions have been removed. Please use *win_wua.get* instead.
    - The `win_wua.list_updates` function has been removed. Please use functions have been removed. Please use `win_wua.list` instead.

Pillar Removed Option

- The *vault* external pillar has been changed as follows:
    - Support for the `profile` argument was removed. Any options passed up until and following the first `path=` are discarded.

Roster Removed Option

- The *cache* roster has been changed as follows:
    - Support for `roster_order` as a list or tuple has been removed. As of the `2019.2.0` release, `roster_order` must be a dictionary.
    - The `roster_order` option now includes IPv6 in addition to IPv4 for the `private`, `public`, `global` or `local` settings. The syntax for these settings has changed to `ipv4-*` or `ipv6-*`, respectively.

State Removed Modules and Options

- The `docker` state module has been removed
    - In *2017.7.0*, the states from this module were split into four separate state modules:
        - *docker_container*
        - *docker_image*
        - *docker_volume*
        - *docker_network*
    - The `docker` module remained, for backward-compatibility, but it has now been removed. Please update SLS files to use the new state names:
        - `docker.running` => *docker_container.running*
        - `docker.stopped` => *docker_container.stopped*
        - `docker.absent` => *docker_container.absent*
        - `docker.network_present` => *docker_network.present*
        - `docker.network_absent` => *docker_network.absent*
        - `docker.image_present` => *docker_image.present*
        - `docker.image_absent` => *docker_image.absent*
        - `docker.volume_present` => *docker_volume.present*

    - `docker.volume_absent` => *docker_volume.absent*

- The *docker_network* state module has been changed as follows:
  - The `driver` option has been removed from *docker_network.absent*. It had no functionality, as the state simply deletes the specified network name if it exists.
- The deprecated `ref` option has been removed from the *git.detached* state. Please use `rev` instead.
- The `k8s` state module has been removed in favor of the *kubernetes* state mdoule. Please update SLS files as follows:
  - In place of `k8s.label_present`, use *kubernetes.node_label_present*
  - In place of `k8s.label_absent`, use *kubernetes.node_label_absent*
  - In place of `k8s.label_folder_absent`, use *kubernetes.node_label_folder_absent*
- Support for the `template_path` option in the `netconfig.managed <salt.states.netconfig.managed()` state has been removed. This is because support for NAPALM native templates has been dropped.
- Support for the `no_chown` option in the `pip.insalled` state has been removed.
- The `trafficserver.set_var` state has been removed. Please use *trafficserver.config* instead.
- Support for the `no_chown` option in the :py:func'virtualenv.managed <salt.states.virtualenv.managed>' function has been removed.
- The `win_update` state module has been removed. It has been replaced by *win_wua*.
- Support for virtual packages has been removed from the py:mod:*pkg state <salt.states.pkg>*.

Utils Removed Options

The `cloud` utils module had the following changes:

- Support for the `cache_nodes_ip` function in `salt utils module` has been removed. The function was incomplete and non-functional.

The `vault` utils module had the following changes:

- Support for specifying Vault connection data within a 'profile' has been removed. Please see the *vault execution module* documentation for details on the new configuration schema.

Dependency Deprecations

Salt-Cloud has been updated to use the `pypsexec` Python library instead of the `winexe` executable. Both `winexe` and `pypsexec` run remote commands against Windows OSes. Since `winexe` is not packaged for every system, it has been deprecated in favor of `pypsexec`.

Salt-Cloud has deprecated the use `impacket` in favor of `smbprotocol`. This changes was made because `impacket` is not compatible with Python 3.

SaltSSH Major Updates

SaltSSH now works across different major Python versions. Python 2.7 ~ Python 3.x are now supported transparently. Requirement is, however, that the SaltMaster should have installed Salt, including all related dependencies for Python 2 and Python 3. Everything needs to be importable from the respective Python environment.

SaltSSH can bundle up an arbitrary version of Salt. If there would be an old box for example, running an outdated and unsupported Python 2.6, it is still possible from a SaltMaster with Python 3.5 or newer to access it. This feature requires an additional configuration in /etc/salt/master as follows:

```
ssh_ext_alternatives:
    2016.3:                      # Namespace, can be actually anything.
        py-version: [2, 6]       # Constraint to specific interpreter version
        path: /opt/2016.3/salt  # Main Salt installation
        dependencies:            # List of dependencies and their installation paths
          jinja2: /opt/jinja2
          yaml: /opt/yaml
          tornado: /opt/tornado
          msgpack: /opt/msgpack
          certifi: /opt/certifi
          singledispatch: /opt/singledispatch.py
          singledispatch_helpers: /opt/singledispatch_helpers.py
          markupsafe: /opt/markupsafe
          backports_abc: /opt/backports_abc.py
```

It is also possible to use several alternative versions of Salt. You can for instance generate a minimal tarball using runners and include that. But this is only possible, when such specific Salt version is also available on the Master machine, although does not need to be directly installed together with the older Python interpreter.

SaltSSH now support private key's passphrase. You can configure it by:

- *–priv-passwd* for salt-ssh cli
- *salt_priv_passwd* for salt master configure file
- *priv_passwd* for salt roster file

State Module Changes

`salt` State Module (used in orchestration)

The `test` option now defaults to None. A value of `True` or `False` set here is passed to the state being run and can be used to override a `test: True` option set in the minion's config file. In previous releases the minion's config option would take precedence and it would be impossible to run an orchestration on a minion with test mode set to True in the config file.

If a minion is not in permanent test mode due to the config file and the 'test' argument here is left as None then a value of `test=True` on the command-line is passed correctly to the minion to run an orchestration in test mode. At present it is not possible to pass `test=False` on the command-line to override a minion in permanent test mode and so the `test: False` option must still be set in the orchestration file.

`event.send` State

The *`event.send`* state does not know the results of the sent event, so returns changed every state run. It can now be set to return changed or unchanged.

`influxdb_user.present` Influxdb User Module State

The `password` parameter has been changed to `passwd` to remove the name collusion with the influxdb client configuration (`client_kwargs`) allowing management of users when authentication is enabled on the influxdb instance

Old behavior:

```
influxdb_user.present:
  - name: exampleuser
  - password: exampleuserpassword
  - user: admin
  - password: adminpassword
```

New behavior:

```
influxdb_user.present:
  - name: exampleuser
  - passwd: exampleuserpassword
  - user: admin
  - password: adminpassword
```

`winrepo_cache_expire_min` Windows Package Definitions Caching

The *`winrepo_cache_expire_min`* has been changed from 0 to 1800 (30 minutes) For example if you run highstate the package definitions are normally updated, however now if the package definitions are younger than *`winrepo_cache_expire_min`* (30 minutes) the package definitions will not be refreshed, reducing the amount of time taken to run a 2nd highstate. To get the old behaviour change the value back to 0 in the minion configuration file. This also effects the behaviour of other functions which default to refresh. The `pkg.refresh_db` will always refresh the package definitions.

LDAP External Authentication

freeipa `groupattribute` support

Previously, if Salt was using external authentication against a freeipa LDAP system it could only search for users via the `accountattributename` field. This release add an additional search using the `groupattribute` field as well. The original `accountattributename` search is done first then the `groupattribute` allowing for backward compatibility with previous Salt releases.

Jinja Include Relative Paths

When a jinja include template name begins with `./` or `../` then the import will be relative to the importing file.

Prior practices required the following construct:

```
{% from tpldir ~ '/foo' import bar %}
```

A more "natural" construct is now supported:

```
{% from './foo' import bar %}
```

Comparatively when importing from a parent directory - prior practice:

```
{% from tpldir ~ '/../foo' import bar %}
```

New style for including from a parent directory:

```
{% from '../foo' import bar %}
```

salt-api

salt-api Windows support

Previously, salt-api was was not supported on the Microsoft Windows platforms. Now it is! salt-api provides a RESTful interface to a running Salt system. It allows for viewing minions, runners, and jobs as well as running execution modules and runners of a running Salt system through a REST API that returns JSON. See *Salt-API* documentation. .. _Salt-API: https://docs.saltstack.com/en/latest/topics/netapi/index.html

Logging Changes

Include Job ID (JID) in Minion and Master Logs

The Job ID (JID) can now be optionally included in both the minion and master logs by including `jid` in either the `log_fmt_console` or `log_fmt_logfile` configuration option:

```
log_fmt_console: "[%(levelname)-8s] %(jid)s %(message)s"
```

The will cause the JID to be included in any log entries that are related to a particular Salt job. The JID will be included using the default format, `[JID: %(jid)s]` but can be overriden with the `log_fmt_jid` configuration item.

```
log_fmt_jid: "[JID: %(jid)s]"
```

Security

Windows runas changes

A password is no longer required with `runas` under normal circumstances. The password option is only needed if the minion process is run under a restricted (non-administrator) account. In the aforementioned case, a password is only required when using the `runas` argument to run command as a different user.

New Modules

Execution Modules

- *salt.modules.ciscoconfparse_mod*
- *salt.modules.jira*
- *salt.modules.google_chat*
- *salt.modules.iosconfig*
- *salt.modules.netmiko*
- *salt.modules.nxos_api*
- *salt.modules.peeringdb*
- *salt.modules.purefb*

- *salt.modules.pyeapi*

Pillar Modules

- *netbox*

Proxy Modules

- *salt.proxy.netmiko*
- *salt.proxy.nxos_api*
- *salt.proxy.pyeapi*

28.2.6 Salt 2019.2.1 Release Notes

Version 2019.2.1 is a bugfix release for *2019.2.0*.

Known Issues

- **ISSUE** #54751: Proxy minion fails to start. Fix planned in 2019.2.2 release.
- **ISSUE** #54762: URIs with IPv6 addresses are broken, preventing master-minion communication in IPv6-only environments. Fix planned in 2019.2.2 release.
- **ISSUE** #54776: Minion fails to start when it contains *ping_interval* in minion configuration file. Fix planned in 2019.2.2 release.
- **ISSUE** #54731: Returners (except for default salt master returner) not loading properly in Py3. Fix planned in 2019.2.2 release.
- **ISSUE** #54758: salt-call state.show_states gives "Passed invalid arguments" error when a sls defined in top.sls file is missing. Fix planned in 2019.2.2 release.
- **ISSUE** #54765: Jinja from import is broken. Fix planned in 2019.2.2 release.
- **ISSUE** #54771: Affects only Debian 10. pkgrepo.managed does not work if used with proxy (configured at salt-minion). No fix date available yet.
- **ISSUE** #54759: Deprecation warnings for pyzmq. No fix date available yet.

Change to YAML Renderer

```
/etc/foo.conf:
  file.managed:
    - source: salt://foo.conf.jinja
    - template: jinja
    - context:
        data: {{ data }}
```

In 2019.2.0, the above SLS will result in an error message following changes to the YAML renderer that now require the new Jinja filter *tojson*.

```
/etc/foo.conf:
  file.managed:
    - source: salt://foo.conf.jinja
    - template: jinja
    - context:
        data: {{ data|tojson }}
```

In 2019.2.1, we introduce a new configuration option for both the Salt master and Salt minion configurations to be able to support the older YAML renderer. Using the option `use_yamlloader_old` will allow the YAML renderer to function as before.

Statistics

- Total Merges: **541**
- Total Issue References: **70**
- Total PR References: **355**
- Contributors: **49** (Akm0d, Ch3LL, DmitryKuzmenko, Ethyling, FireGrace, KChandrashekhar, ScoreUnder, amendlik, aplanas, arsiesys, bbinet, bryceml, cbosdo, cdalvaro, chdamianos, cmcmarrow, cro, damianosSemmle, dmurphy18, doesitblend, dwoz, felippeb, frogunder, garethgreenaway, github-abcde, isbm, jfindlay, lomeroe, mattLLVW, mattp-, mirceaulinic, nicholasmhughes, rbthomp, rombert, rsmekala, s0undt3ch, sathieu, sbrennan4, sdodsley, simonflood, sjorge, soer7022, stratusjerry, tanlingyun2005, terminalmage, twangboy, waynew, weswhet, xuhcc)

Changelog for v2019.2.0..v2019.2.1

Generated at: 2019-09-09 19:56:22 UTC

- **ISSUE** #54429: (frogunder) [2019.2.1] Sudo user error (refs: #54433)
- **PR** #54433: (dwoz) Re-gen executors with proper arguments @ *2019-09-09 19:14:05 UTC*
  - 62fc61c Merge pull request #54433 from dwoz/sudo_minion
  - e08eb19 Merge branch '2019.2.1' into sudo_minion
- **PR** #54434: (frogunder) fix_docs @ *2019-09-09 18:55:28 UTC*
  - 76fac57 Merge pull request #54434 from frogunder/fix_doc
  - 2d22d73 fix_docs error
  - a266a5c fix_docs
  - db5464a Fix linter warts
  - 144a600 Re-gen executors with proper arguments
- **PR** #54424: (frogunder) remove in progress from release_notes 2019.2.1 @ *2019-09-05 23:11:03 UTC*
  - 4071dcb Merge pull request #54424 from frogunder/releasenotes_201921_new
  - b66436f Update 2019.2.1.rst
  - bcfef11 remove in progress from release_notes 2019.2.1
- **PR** #54416: (frogunder) Update man_pages for 2019.2.1 @ *2019-09-05 22:48:51 UTC*
  - c1587cc Merge pull request #54416 from frogunder/man_pages_201921

  - 6f7c4ee Update man_pages for 2019.2.1
- **PR** #54419: (frogunder) releasenotes 2019.2.1 @ *2019-09-05 22:48:42 UTC*
  - 3118e2f Merge pull request #54419 from frogunder/release_notes_201921
  - 2caee96 Update doc/topics/releases/2019.2.1.rst
  - ffabde2 Update 2019.2.1.rst
  - dd6426d Update doc/topics/releases/2019.2.1.rst
  - 9d4239c Update doc/topics/releases/2019.2.1.rst
  - 8c60ae4 Update doc/topics/releases/2019.2.1.rst
  - 72cb24c releasenotes 2019.2.1
- **PR** #54409: (dwoz) Point salt-call to correct multi-master configs @ *2019-09-04 20:08:24 UTC*
  - 04e28cf Merge pull request #54409 from dwoz/mm_log
  - e207de6 Point salt-call to correct multi-master configs
- **PR** #54396: (dwoz) Use different logging port for multimaster tests @ *2019-09-03 23:20:23 UTC*
  - 2c281e0 Merge pull request #54396 from dwoz/multimaster_logging
  - 104e0a0 Use different logging port for multimaster tests
- **ISSUE** #54256: (sleminov-tc) Empty salt api token file breaks maintenance process. (refs: #54325, #54324, #54388, #54376)
- **ISSUE** #37945: (gstachowiak) Missing exception handling in salt.master.Maintenance. Process never completes. (refs: #54325, #54324, #54388, #54376)
- **PR** #54388: (dwoz) 54256 re fix empty api token 2019.2.1 @ *2019-09-03 23:19:20 UTC*
  - **PR** #54325: (waynew) Delete bad API token files (refs: #54388)
  - **PR** #54324: (waynew) Delete bad API token files (refs: #54325, #54388)
  - 730a007 Merge pull request #54388 from dwoz/54256-re-fix-empty-api-token-2019.2.1
  - 80651d3 Fix linter
  - 982ed3d Delete bad API token files
- **ISSUE** #54256: (sleminov-tc) Empty salt api token file breaks maintenance process. (refs: #54325, #54324, #54388, #54376)
- **ISSUE** #37945: (gstachowiak) Missing exception handling in salt.master.Maintenance. Process never completes. (refs: #54325, #54324, #54388, #54376)
- **PR** #54376: (dwoz) Move tokens in place with an atomic operation @ *2019-09-03 20:40:50 UTC*
  - 11016ce Merge pull request #54376 from dwoz/tmp_token
  - 0b28c60 Fix linter complaints
  - 8e33343 Move tokens in place with an atomic operation
- **PR** #54386: (twangboy) Fix unit.utils.test_win_dacl (2019.2.1) @ *2019-09-03 16:57:25 UTC*
  - 5161203 Merge pull request #54386 from twangboy/fix_test_win_dacl_2019.2.1
  - b8fa7d3 Sort the perms
- **PR** #54355: (s0undt3ch) [2019.2.1] [GitPillar Tests] Have the started processes log instead of just writing to stdout/stderr @ *2019-08-29 23:26:50 UTC*

  - b4caaad Merge pull request #54355 from s0undt3ch/hotfix/test-daemons-2019.2.1
  - 561c0a2 Have the started processes log instead of just writing to stdout/stderr
  - c2a3115 Don't overwritethe stderr logger
- **PR** #54353: (cmcmarrow) adds @skipIf not windows @ *2019-08-29 19:33:11 UTC*
  - ce5b109 Merge pull request #54353 from cmcmarrow/2019_win_system_skip
  - 8e28074 adds @skipIf not windows
- **PR** #54338: (dwoz) Log server closes completely during shutdown @ *2019-08-29 17:14:05 UTC*
  - 94c03e5 Merge pull request #54338 from dwoz/logging_fixup
  - a3e227e Log server closes completely durring shutdown
- **PR** #54327: (garethgreenaway) [2019.2.1] Catch the AccessDenied exception and continue when running under Py3. @ *2019-08-29 16:59:34 UTC*
  - 490ee69 Merge pull request #54327 from garethgreenaway/more_test_program_fixes
  - 97b4d86 Catch the AccessDenied exception and continue when running under Py3.
- **PR** #54316: (Akm0d) Azure cleanup @ *2019-08-27 16:14:09 UTC*
  - bb911c3 Merge pull request #54316 from Akm0d/azure-cleanup
  - 2072773 pass longer timeout to msazure destroyinstance
  - 34eb477 Merge branch '2019.2.1' of https://github.com/saltstack/salt into azure-cleanup
  - 7ae48b1 Allow azure tests enough time to clean up interfaces
  - b45ae74 Clean up azure instances properly
- **PR** #54314: (garethgreenaway) [2019.2.1] Do not raise the exception in testprogram.py @ *2019-08-26 23:59:37 UTC*
  - 9334e4e Merge pull request #54314 from garethgreenaway/testprogram_access_denied_do_not_raise
  - a53bacb Do not raise the exception if a process can not be access, just continue
- **PR** #54305: (dwoz) Fix wart in test_process tests on windows @ *2019-08-26 17:59:12 UTC*
  - 71fb051 Merge pull request #54305 from dwoz/winfix
  - 4e68860 Fix wart in test_process tests on windows
- **PR** #54303: (dwoz) Fix wart in periodic callback interval @ *2019-08-24 20:04:35 UTC*
  - 4d80f0a Merge pull request #54303 from dwoz/periodic_cb
  - bf5f7de Fix wart in periodic callback interval
- **ISSUE** #54219: (KChandrashekhar) integration.shell.test_call.py and tests/integration/shell/test_proxy.py (refs: #54259)
- **ISSUE** #54153: (KChandrashekhar) Ubuntu 16.04 Proxy failures (refs: #54259)
- **PR** #54259: (dwoz) Call os.fork less to avoid race conditions @ *2019-08-24 02:12:42 UTC*
  - 5838851 Merge pull request #54259 from dwoz/less_forking
  - bf4d0dd Fix process list tests on windows py3
  - 316ec78 Allow more time for slow test
  - 9baf86f Do not remove attributes from process class

  - e7b6aa8 Clean up cruft
  - ed7103a Attempt to fix integration.shell.test_call test
  - 659c805 Change noise log to debug
  - 0660b1a Clean up typoes
  - 9f1fe42 Call os.fork less to avoid race conditions
- **ISSUE** #50814: (doesitblend) Minion continues trying down master (refs: #54239, #53417)
- **ISSUE** #49680: (doesitblend) Beacon events handled differently than event.send (refs: #53344, #54239)
- **ISSUE** #49663: (doesitblend) Beacon only fires to single random master in HA mode (refs: #53344, #54239)
- **PR** #54247: (DmitryKuzmenko) 2019.2.1 with multimaster bb fixes @ *2019-08-23 20:19:19 UTC*
  - **PR** #54239: (DmitryKuzmenko) Multimaster tests with bb fixes (refs: #54247)
  - **PR** #53848: (DmitryKuzmenko) Features/multimaster tests (refs: #54239)
  - **PR** #53417: (DmitryKuzmenko) Multimaster minion hang on fire_master (refs: #54239, #53848)
  - **PR** #53344: (DmitryKuzmenko) Run beacons on the only one minion instance. Return to all masters. (refs: #54239, #53848)
  - 531f3aa Merge pull request #54247 from DSRCorporation/2019.2.1_with_multimaster_bb_fixes
  - 62cd8cd Merge branch '2019.2.1' into 2019.2.1_with_multimaster_bb_fixes
- **PR** #54235: (github-abcde) Fix 46034 2019.2.1 @ *2019-08-22 22:32:09 UTC*
  - 993c341 Merge pull request #54235 from ogd-software/fix_46034-2019.2.1
  - 93bd30d Add alternative fix for "!" stomping Apparently (after watching Jenkins tests fail), what yaml.safe_load returns depends not on the version of salt, but on some other external dependency. Because of this, fix both possible return values.
  - c95dd4d Add test for this specific bugfix
  - efc29d8 Prevent yamlify_arg from stomping "!"
- **PR** #54282: (Akm0d) clean up renamed instances in the teardown @ *2019-08-22 22:13:51 UTC*
  - 5d1a4d0 Merge pull request #54282 from Akm0d/clean-up-renamed
  - 11c1117 Removed redundant parenthesis
  - 802235d don't delete shutting down ec2 instances
  - 3602942 reversed the order in which an instance was renamed
  - 5500ede renamed _alt_name function
  - 23beab4 clean up renamed instances in the teardown
- **PR** #54252: (s0undt3ch) [2019.2.1] More control on spun test deamons on git pillar tests @ *2019-08-22 22:13:08 UTC*
  - c181f5a Merge pull request #54252 from s0undt3ch/hotfix/git-pillar-2019.2.1
  - 4eb85fa Skip tests on RHEL family < 7
  - 64a1be8 Don't even go through the transport to get the system grains
  - d0f8efa Don't keep recreating the git repos. Restore from backup.
  - 6114e9d All prep work is done in *setUpClass*.

  - 5b030dd Daemons are now started/stopped on tests, not by salt
  - ed60f0d We don't actually need grains
  - 911446d Lock *uwsgi* requirement.
    - 805f0ac Merge branch '2019.2.1' into 2019.2.1_with_multimaster_bb_fixes
- **PR** #54214: (Ch3LL) Remove unnecessary logging from minion.py @ *2019-08-22 18:08:26 UTC*
  - b40cdbf Merge pull request #54214 from Ch3LL/fix_logging
  - 8fd24cf Remove unnecessary logging from minion.py
    - 9d28098 Merge branch '2019.2.1' into 2019.2.1_with_multimaster_bb_fixes
- **PR** #54277: (dwoz) Win runas plus @ *2019-08-21 23:59:16 UTC*
  - de77762 Merge pull request #54277 from dwoz/win_runas_plus
  - f719591 Update tests
  - d8749d9 Do not remove attributes from process class
  - ac5a2a4 Only close handle if it got created
- **PR** #54278: (Akm0d) Correctly detect when an ms_azure instance is deleted @ *2019-08-21 23:14:01 UTC*
  - 93cf40a Merge pull request #54278 from Akm0d/cloud-test-logging
  - 2dc1520 Merge branch '2019.2.1' into cloud-test-logging
  - 3867448 Fixed incorrect arguments
  - bdce5f4 Fix detecting msazure deletion with multiple tries
- **PR** #54263: (s0undt3ch) [2019.2.1] Exit test suite if test daemons fail to start @ *2019-08-21 22:06:52 UTC*
  - 94f77a0 Merge pull request #54263 from s0undt3ch/hotfix/test-daemons-2019.2.1
  - 655efd6 Rearrange logging termination
  - 28fa5d2 Prevent traceback on python interpreter shutdown
  - 9035731 Write to stderr a message with the exit code
  - 566c620 Exit the test suite if one of the test daemons fails to start
- **PR** #54270: (s0undt3ch) [2019.2.1] Speedup testing helpers decorators @ *2019-08-21 20:33:25 UTC*
  - 3b9fd44 Merge pull request #54270 from s0undt3ch/hotfix/requires-system-grains-2019.2.1
  - 9f99bff Speedup testing helpers decorators
- **ISSUE** #53931: (KChandrashekhar) integration.states.test_pip_state.PipStateTest.test_issue_6912_wrong_owner_requirements_f (refs: #54274)
- **PR** #54274: (felippeb) #53931 never fail on codecov curls @ *2019-08-21 18:12:19 UTC*
  - 2c4afa0 Merge pull request #54274 from felippeb/2019.2.1
  - 9c0b216 #53931 never fail on codecov curls
- **PR** #54264: (dwoz) Fix unit utils jinja when run on their own @ *2019-08-21 17:22:10 UTC*
  - b3c70c3 Merge pull request #54264 from dwoz/jinja_units
  - 10a6d53 Fix unit utils jinja when run on thier own
- **PR** #54266: (Akm0d) Fix incorrect types on failing cloud tests @ *2019-08-21 17:14:40 UTC*

  - 308ba55 Merge pull request #54266 from Akm0d/cloud_test_ultimate_fix
  - ff2125a Invert assertion to test for instance gone
  - 82136a8 test for ec2 instance shutting down
  - 283f631 run cloud –query if delete string is not available
  - 33e6b6d changed deprecated assertEquals
  - 41e6bc8 Cloud tests run correctly everywhere
  - 03a8b2c reverted literal_eval
- **PR** #54251: (Akm0d) reduced query calls to salt-cloud @ *2019-08-20 20:10:59 UTC*
  - 25d0b52 Merge pull request #54251 from Akm0d/cloud_test_optimization
  - 5fe6bfc Optimized salt-cloud queries
  - 43418a4 Merge pull request #7 from Ch3LL/cloud_provider_files
    - b93944b Remove joyent config cloud test files
    - bd37849 Rename provider azure-config to azurearm-config
  - e7c06cb Added longer delays for ec2 tests
  - 5ceb47e Correctly check for GCE and EC2 deletion
  - 52c178c Added pretty yaml formatting to cloud-test logs
  - 4fc5e9e re-fixed simultaneous GCE tests
  - 326e9f2 Merge pull request #6 from Ch3LL/cloud_provider_files
    - adb1464 fix pylint
    - 51ce819 Move provider config setup into Setup/Teardown class
  - e58b40a Merge pull request #5 from Ch3LL/cloud_provider_files
    - ab8adcc Only use the provider conf.d file we are testing
  - 76609ad Merge branch '2019.2.1' into cloud_test_optimization
  - f764bc0 Clean up expensive queries
  - b3d02df reduced query calls to salt-cloud
- **PR** #54240: (Ch3LL) Fix TypeError python3 in get_image_id ec2 cloud @ *2019-08-20 20:10:01 UTC*
  - ba81ca3 Merge pull request #54240 from Ch3LL/get_image_ec2
  - a3a39f3 Use ami variable instead in test
  - dc1ae52 Fix TypeError python3 in get_image_id
    - d6ad475 Merge branch '2019.2.1' into 2019.2.1_with_multimaster_bb_fixes
- **PR** #54246: (dwoz) Cut down on log noise @ *2019-08-19 17:20:03 UTC*
  - 8404ec2 Merge pull request #54246 from dwoz/log_noise
  - e572711 Cut down on log noise
- **PR** #54245: (dwoz) Fix the sshd pid file @ *2019-08-19 17:19:32 UTC*
  - 5b3e925 Merge pull request #54245 from dwoz/sshd_pid_file
  - df95347 Fix the sshd pid file

- **PR** #54237: (dwoz) Do not dup2 things without fileno @ *2019-08-18 17:13:29 UTC*
  - 268803b Merge pull request #54237 from dwoz/dup2
  - 2ebd22b Fix linter
  - 63c616a Do not dup2 things without fileno
    - 0b33498 Workaround for beacons.list_available slowness.
    - b7a1058 Minor: Better support of named multimaster tests.
    - a5cccdf Minor: remove unneded logging
    - 314bbb3 Support tcp transport in multimaster tests. Minor beacon update.
    - fcb8ff0 Properly set HAS_IPTABLES from __virtual__ return.
    - a75db01 Multimaster related fixup: don't run test engine on minions
    - 8a88542 Multimaster beacons fix: put list into a dict to fire on the bus
    - 790fb41 Inotify beacons test logging and cleanup update
    - d84c8851 Inotify beacon test fixes and improvements.
    - 472ffc0 Give minion more time to handle beacon creation.
    - 17ceee0 Fixed Lint errors
    - 0bcf53c Fixed the unicode and timing inotify test issues.
    - f17bcf3 Improved assertion error diagnostic for master down test.
    - db0bae8 Run multimaster as well tests when no specific group is set
    - 723f363 Ignore the multimaster minion subgroup test in module names test.
    - 612191c Fixed lint failures
    - c1efae6 Multimaster test for events handling on master disconnect
    - 1654fe8 Multimaster inotify beacons test
    - 65dc47c Multimaster tests environment and test module test.
    - 2c178b0 Moved annoying message to trace log level.
    - 3e497c1 Handle minion events asynchronously.
    - 909323a Tests fixes
    - 6dad18f Cleanup event object
    - 2bf55be Run beacons on the only one minion instance. Return to all masters.
- **PR** #54121: (Akm0d) Refactored cloud tests @ *2019-08-16 17:24:14 UTC*
  - 27d89bd Merge pull request #54121 from Akm0d/gce_cloud_test
  - 48c57b3 Try multiple times to detect deletion of an instance
  - 74300eb return a valid set
  - 2d63b87 Wait for an instance to be fully created if necessary
  - 83ad8b3 Merge branch '2019.2.1' of github.com:SaltStack/salt into gce_cloud_test
- **ISSUE** #54174: (dwoz) SignalHandlingMultiprocessingProcess bugs (refs: #54204)
- **PR** #54204: (dwoz) Fix signal handling in subprocesses @ *2019-08-16 00:46:40 UTC*

  - c5ee91c Merge pull request #54204 from dwoz/signal_proc
  - 09e5e49 Merge branch '2019.2.1' into signal_proc
- **ISSUE** #7745: (mgwilliams) yaml 'module' has no attribute 'Dumper' (refs: #54223)
- **PR** #54223: (dwoz) Test verify logs without a whole minion @ *2019-08-16 00:46:09 UTC*
  - 99a8fc5 Merge pull request #54223 from dwoz/7754_test_removal
  - 4be711e Fix up linter warts
  - 7573826 Remove crufty un-needed tests
  - effa77d Test verify logs without a whole minion
    - b40f41f Revert fallback to default signals change
    - 1d83b10 Fix signal handling in subprocesses
      - f90fe66 Reverted foo/bar creds per #46265
      - 3f56bad Reduced the amount of informtaion in debuggings tatements
      - de53bf0 Merge branch '2019.2.1' of github.com:SaltStack/salt into gce_cloud_test
- **PR** #54203: (dwoz) Run transport test loops in single thread @ *2019-08-15 20:10:20 UTC*
  - 8d440e0 Merge pull request #54203 from dwoz/transport_tests
  - 22b9d38 Run transsport test loops in single thread
    - f6180a6 Even better logging of errors
    - a15dc4b better formatting and error reporting
    - 51fbc2a better logging on instance create failure
    - 1f7dd52 removed hack fixes from gce
    - 31cdc7c WAR ROOM SKIP GCE TESTS
    - 7d603be hasattr cannot find parent properties starting with __
    - 686cab9 Separated assertInstanceDestroy and _destry_instance
    - 158a38c Revert "Removed subclass from cloud name"
    - b7cf43d Removed subclass from cloud name
    - 8c5eb18 fixed pylint error on format string
    - 59e20bc Merge branch '2019.2.1' of https://github.com/saltstack/salt into gce_cloud_test
- **PR** #54191: (garethgreenaway) [2019.2.1] Addition logging in testprogram used by integration.shell.test_minion.MinionTest.test_exit_status_correct_usage @ *2019-08-13 21:44:41 UTC*
  - a05018e Merge pull request #54191 from garethgreenaway/test_program_test_mininion
  - 8463f84 Adding some additional logging when the call to "proc_cmdline = proc.cmdline()" reults in a AccessDenied exception, so we can see what the process is and who the process is running as.
- **PR** #54202: (waynew) Only run one filter test @ *2019-08-13 21:33:32 UTC*
  - e5e9299 Merge pull request #54202 from waynew/shorter-jinja-ssh-tests
  - c333ac8 Only run one filter test
    - 5912f24 Switched to Azurearm over msazure
- **PR** #54172: (Ch3LL) Migrate from azure to azurearm tests @ *2019-08-13 20:07:34 UTC*

  - a13cb3e Merge pull request #54172 from Ch3LL/azurearm_tests
  - 465523f Migrate from azure to azurearm tests
- **ISSUE** #50535: (wyardley) salt errors with current azure-storage because of no version set (refs: #50567)
- **PR** #54173: (Ch3LL) Cherry Pick #50567 and #53238 into 2019.2.1 @ *2019-08-13 20:07:09 UTC*
  - **PR** #53238: (nicholasmhughes) Azure: Fixed ability to pass SSH key to Linux VMs (refs: #54173)
  - **PR** #50567: (rombert) azurefs: gracefully handle AttributeError (refs: #54173)
  - 39d0c59 Merge pull request #54173 from Ch3LL/azure_fixes
  - 9052eaf reverting log string handling
  - a1a61ee fixed ability to pass ssh key to Linux VMs
  - 4a141a1 azurefs: gracefully handle AttributeError
    - 6fa28de fixed underscore
      - c35379a Moved finding installer into class
    - 95484f8 Moved finding installer into class
    - a86ae34 moved installer finder to SetUp method
    - 933964a removed default user/pass from profitbricks test file
    - 2a2437d fixed dictionary access
    - b9ee411 Fixed broken tests
    - 80efd26 fixed misspelled provider information
    - 511779b removed unused imports
    - 1cd1c13 setUp all cloud tests the same way
    - d322408 Got rid of unnecessary delay
    - 98ba18c fixed pylint error
    - c34c17e Merge branch '2019.2.1' of https://github.com/saltstack/salt into gce_cloud_test
- **PR** #54178: (dwoz) Fix flaky set_computer_name in mac_system module @ *2019-08-12 18:26:40 UTC*
  - 19f4d5e Merge pull request #54178 from dwoz/mac_computer_name
  - e51100e Fix skip on non macos platforms
  - a3a9d92 Fix linter
  - c499d44 Revert unwanted flaky from non computer name tests
  - 941fcc8 Fix flaky set_computer_name in mac_system module
    - 90e8350 Merge branch '2019.2.1' of https://github.com/saltstack/salt into gce_cloud_test
- **PR** #54180: (dwoz) Point test-kitchen to my keepalive_maxcount branch @ *2019-08-12 18:18:26 UTC*
  - 6fe6799 Merge pull request #54180 from dwoz/keepalive_maxcount
  - dc9ee21 Add reminder about kitchen version
  - 19a1f1b Point test-kitchen to my keepalive_maxcount branch
    - 60f8351 Fixed failing Ec2 Tests
    - e2f085c removed py3-only function call

    - 4406c57 fixed lint errors, more descriptive fails
    - 6840d5f assert instances exist in unified way
    - 03da233 skip EOL joyent tests, secure delete instances
    - 3123bb1 Merge branch '2019.2.1' of https://github.com/saltstack/salt into gce_cloud_test
- **ISSUE** #53306: (doesitblend) Fully Qualify CMD on Windows Minions (refs: #53311, #54033)
- **PR** #54033: (twangboy) Backport #53311 to 2019.2.1 @ *2019-08-09 16:46:36 UTC*
  - **PR** #53311: (doesitblend) Add fully qualified cmd call (refs: #54033)
  - 9f5302d Merge pull request #54033 from twangboy/fix_win_service_2019.2.1
  - 2850fab Merge branch '2019.2.1' into fix_win_service_2019.2.1
  - 0e2a2b8 Merge branch '2019.2.1' into fix_win_service_2019.2.1
  - 16c704e Backport #53311 to 2019.2.1
- **PR** #53900: (twangboy) Fix inconsistent full names in LGPO @ *2019-08-09 09:15:41 UTC*
  - c0e49d9 Merge pull request #53900 from twangboy/fix_lgpo
  - e05abdf Merge branch '2019.2.1' into fix_lgpo
  - e4e71cd Remove duplicate test
  - 68c632b Add some tests
  - 3c34075 Fall back to slower method if new method not available
  - 0108bcd Avoid using the lowercase method in XPath
  - fcc0405 Fix telemetry issue, add some timining logs
- **PR** #52818: (twangboy) Add watchdog as a bundled dependency on Windows @ *2019-08-09 04:26:54 UTC*
  - 9eb2938 Merge pull request #52818 from twangboy/add_watchdog
  - 2658634 pre-commit
  - 35691f2 Add watchdog to Linux and OSX .in files
  - 5ef5969 Rremove opensuse 42 static files
  - ec285d0 Remove watchdog from the rest of the .in files
  - 86e64ae Remove unused import, remove Windows reference
  - b934d53 Remove watchdog from windows.in
  - cfaee96 Fix and unify watchdog beacon tests for all OS's
  - cc73204 Put Windows tests in their own class
  - 6fbf1a3 Remove unused import… lint
  - 832a1db Fix windowsisms
  - 4f47d50 Update static requirements
  - 762fe3b Remove some fluff
  - 946076c Add watchdog for tests
  - 02248dc Fix modified test to account for OS variances
  - 15ee479 Compile static requirements

  - 78a5723 Add watchdog to Linux and OSX .in files
  - 9a0f08a Update static requirements
  - 5cad13b Rremove opensuse 42 static files
  - 1f05fce Remove watchdog from the rest of the .in files
  - 121595a Remove unused import, remove Windows reference
  - c410a4b Remove watchdog from windows.in
  - 05a0845 Fix and unify watchdog beacon tests for all OS's
  - 200e230 Put Windows tests in their own class
  - 6cc8d96 Remove unused import… lint
  - a4abb72 Fix windowsisms
  - 005bed9 Update static requirements
  - c1bc627 Remove some fluff
  - 92c8d47 Add watchdog for tests
  - 9544624 Add watchdog as a bundled dependency on Windows
- **PR** #54143: (Akm0d) Don't call potentially missing shade library (#53734) @ *2019-08-09 01:05:26 UTC*
  - **PR** #53734: (Akm0d) Don't call potentially missing shade library (refs: #54143)
  - 5bef520 Merge pull request #54143 from Akm0d/openstack_fix
  - 3f58885 Merge branch '2019.2.1' into openstack_fix
  - 71f5f38 Don't call potentially missing shade library (#53734)
    - 29ee7ce removed print statements, optimized query
    - 14b7539 added back print statements for easy debug
    - 823b3b2 Removed print statements and broken re
    - 9696ed9 experiment with sublcass names
    - 2fa6157 more verbose instance names based on test
    - 2acb2df undercased non-const variable and added print statements
    - c5e875d Merge branch '2019.2.1' of https://github.com/saltstack/salt into gce_cloud_test
- **PR** #54001: (garethgreenaway) [2019.2.1] Porting metaproxy changes in #50183 to 2019.2.1 @ *2019-08-08 04:24:31 UTC*
  - **PR** #50183: (cro) Allow proxy minion types to be dynamically loaded (refs: #54001)
  - 584e3f2 Merge pull request #54001 from garethgreenaway/add_metaproxy_2019_2_1
  - 8353467 Merge branch 'add_metaproxy_2019_2_1' of github.com:garethgreenaway/salt into add_metaproxy_2019_2_1
    - 1892d86 Merge branch '2019.2.1' into add_metaproxy_2019_2_1
  - ab72e1b Adding unit.test_proxy_minion to ignore list
  - 09f6fed Removing match tests, functionality does not exist in 2019.2.1
  - 4af95e4 Adding tests to ensure _metaproxy_call to called when ProxyMinion objects are created.
  - d162a88 Adding salt/metaproxy/proxy.py and tests/unit/modules/test_match.py

  - 51726b1 POrting metaproxy changes in #50183 to 2019.2.1
- **ISSUE** #51008: (cdalvaro) cwd option not working with cmd.run and runas (refs: #52632, #54079)
- **PR** #54136: (ScoreUnder) Fix cmd.run on MacOS (rebased) @ *2019-08-08 04:20:31 UTC*
  - **PR** #54079: (ScoreUnder) Fix cmd.run on MacOS – wrong environment variables (refs: #54136)
  - **PR** #52632: (cdalvaro) Prevent shell injection as root with cmd.run on macOS (refs: #54079)
  - **PR** #47212: (weswhet) fix macOS running as user. (refs: #54079)
  - bf863d0 Merge pull request #54136 from ScoreUnder/2019.2.1-patch-macos-cmd.run
  - 1182eb9 Add integration test justifying strange use of braces after cd in cmd.run
  - 733d2e7 Fixes to ITs
  - da38993 Add integration tests for cwd/runas changes on MacOS
  - ab8b39c Fix environment in cmd.run runas on MacOS
    - c9cb18f Don't call potentially missing shade library (#53734)
    - 5ef8414 Added more verbose logging to cloud tests
    - 2f3942f better logging and fail messages
    - 9454f4c Fixed pylint errors and temporary print statements
    - fd13ce1 Further refactored cloud test code
    - b03cc48 Assume the instance exists and needs to be deleted
    - 0991ff5 fixed pylint errors and failed tests
    - 55e502a Merge branch '2019.2.1' of https://github.com/saltstack/salt into gce_cloud_test
- **PR** #53974: (Ch3LL) [2019.2.1] Add pygit2 requirement @ *2019-08-06 17:32:05 UTC*
  - e268b95 Merge pull request #53974 from Ch3LL/add_pygit2
  - 11864a0 Merge branch '2019.2.1' into add_pygit2
  - d1b1452 Merge branch '2019.2.1' into add_pygit2
  - a5698cc Merge branch '2019.2.1' into add_pygit2
  - 7dd97a6 Merge branch '2019.2.1' into add_pygit2
  - 358974a Skip Pygit2 tests on windows
  - bc64961 Use accurate file:/// URI on windows for gitfs tests
  - 894f958 Fix GitFS support for pygit2 >= 0.28.0
  - 4580030 [2019.2.1] Add pygit2 requirement
    - 4009bb5 fixed pylint errors
    - 459b16d put cloudtest base in separate file so that only one change needs to be made
    - 8e72335 Merge branch 'gce_cloud_test' of github.com:Akm0d/salt into gce_cloud_test
      - 0e2621e Merge branch '2019.2.1' into gce_cloud_test
    - 9f98b16 finished merge
- **PR** #54108: (dwoz) More robust azure instance deletion @ *2019-08-03 03:58:20 UTC*
  - 46f5e2e Merge pull request #54108 from dwoz/azure_test

  - 754b719 Fix linter
  - 773235d Fix missing time import
  - 5a8f2ec More robust azure instance deletion
- **PR** #54017: (dmurphy18) Allow for main thread having terminated pid, before ThreadPoolExecutor threads @ *2019-08-02 18:11:48 UTC*
  - 2cb5a0b Merge pull request #54017 from dmurphy18/fix_deb9_build90_tests
  - 2866520 Skip process kill tests on Windows
  - 5577f14 Clean up lint errors
  - 4ca709e Merge branch '2019.2.1' into fix_deb9_build90_tests
  - 264c767 Merge pull request #1 from dwoz/fix_deb9_build90_tests
    - 9bb9466 Add unit tests for weird pid does not exist cases
    - 64d9752 Add warning messages to help determine why pids do not exist
  - 406d382 Adjusted for review comments
  - 2d20fbe Adjusted try/except statements on process checking
  - e189177 Removed typos
  - 2e3ca43 Updated for review comments
  - 68c29af Allow for main thread having terminated pid, before ThreadPoolExecutor threads
    - e3158c8 refactored all cloud tests
    - 194e0c7 Refactored GCE cloud test
- **PR** #54080: (dmurphy18) Disabling random website tests till allow for quota usage @ *2019-07-31 21:16:54 UTC*
  - 4e2efcd Merge pull request #54080 from dmurphy18/u1804_py3_random_skip
  - 0660e6e Disabling random website tests till allow for quota usage
- **PR** #54063: (twangboy) Make the skip apply to any system missing crypt @ *2019-07-30 19:22:07 UTC*
  - dbbbcc9 Merge pull request #54063 from twangboy/fix_test_pycrypto_2019.2.1
  - 6fcf035 Make the skip apply to any system missing crypt
- **PR** #54050: (cmcmarrow) fixs integration terminate error @ *2019-07-30 13:38:55 UTC*
  - 4df6271 Merge pull request #54050 from cmcmarrow/integration_terminate_fix
  - ebb07f4 Merge branch '2019.2.1' into integration_terminate_fix
- **PR** #54057: (dmurphy18) Added support for is_fedora and skip Minion test test_issue_7754 on Fedora @ *2019-07-29 22:41:25 UTC*
  - 9d1bd63 Merge pull request #54057 from dmurphy18/fix_fedora30_test
  - 30f3bda Added support for is_fedora and skip Minion test test_issue_7754
  - b3293a9 Merge branch '2019.2.1' into integration_terminate_fix
- **ISSUE** #53948: (KChandrashekhar) integration.shell.test_call.CallTest.test_issue_2731_masterless (refs: #54040)
- **ISSUE** #2731: (cwood) Masterless Broken in 0.10.5 (refs: #54040)
- **PR** #54040: (waynew) Remove dead test @ *2019-07-29 17:23:45 UTC*

  - 5d3bcd7 Merge pull request #54040 from waynew/remove-dead-test
  - e9a5a57 Remove dead test
    - ba3e867 Merge branch '2019.2.1' into integration_terminate_fix
- **PR** #54051: (twangboy) Skip get time test @ *2019-07-29 16:54:50 UTC*
  - 70ffcec Merge pull request #54051 from twangboy/fix_test_system
  - ceaba05 Skip get time test
- **PR** #54038: (Ch3LL) Pyton3 digitial ocean test fix: to_str on key @ *2019-07-29 16:54:19 UTC*
  - 4aed833 Merge pull request #54038 from Ch3LL/do_py3_fix
  - f7346db Pyton3 digitial ocean test fix: to_str on key
    - 05cd93f fixs integration terminate error
    - 69c3106 fixs integration terminate error
- **PR** #53735: (twangboy) Fix Windows tests in test_system @ *2019-07-26 22:38:28 UTC*
  - 3cedacd Merge pull request #53735 from twangboy/fix_test_system
  - a9e9c97 Add timeouts and account for those in assert
  - d3a160e Merge branch '2019.2.1' into fix_test_system
  - f95f0e7 Merge branch '2019.2.1' into fix_test_system
  - 87e02b8 Use setUpClass
  - 6a685bd Fix some tests in the Windows Class
- **PR** #53953: (Ch3LL) Send SIGTERM to webserver in teardown of gitfs tests @ *2019-07-25 21:12:42 UTC*
  - cc1cda1 Merge pull request #53953 from Ch3LL/improve_git_test
  - 347ea1e Use Sigkill and add time.sleep befor check
  - 637bf95 Merge branch '2019.2.1' into improve_git_test
  - c0be147 Send SIGTERM webserver during teardown of gitfs tests
  - b776c0c Check if gitfs server fails to setup for tests
- **PR** #53999: (Ch3LL) Generate new key each time for digital ocean key test @ *2019-07-25 21:07:45 UTC*
  - 2324167 Merge pull request #53999 from Ch3LL/fix_digital_ocean
  - b0b6e3b Generate new key each time for digital ocean key test
- **PR** #53970: (garethgreenaway) [2019.2.1] Adding a WAR ROOM Skip for test_directory_clean_require_with_name @ *2019-07-25 21:03:49 UTC*
  - 32fec66 Merge pull request #53970 from garethgreenaway/osx_skip_test_directory_clean_require_with_name
  - 362b84a Adding a WAR ROOM Skip for test_directory_clean_require_with_name
- **PR** #54003: (dwoz) War room skip for tcp build @ *2019-07-24 22:52:18 UTC*
  - 213dfff Merge pull request #54003 from dwoz/skip_test
  - 6cdb8fa War room skip for tcp build
- **PR** #53897: (cmcmarrow) patches salt grains locale_info decode error @ *2019-07-23 13:19:00 UTC*
  - 048c097 Merge pull request #53897 from cmcmarrow/timezone_fix

  - d9e402d Merge branch '2019.2.1' into timezone_fix
- **PR** #53920: (s0undt3ch) [2019.2.1] Update CI pipelines. Remove old jenkins cruft. @ *2019-07-20 12:33:01 UTC*
  - 9a846b4 Merge pull request #53920 from s0undt3ch/hotfix/ci-pipelines-2019.2.1
  - b09963b Update CI pipelines. Remove old jenkins cruft.
- **PR** #53728: (garethgreenaway) [2019.2.1] Disabling test_get_set_computer_name on OS X and Py3 @ *2019-07-19 17:18:33 UTC*
  - 15accef Merge pull request #53728 from garethgreenaway/2019_2_1_mac_system_disable_test_get_set_computer_name
  - a6d853c Merge branch '2019.2.1' into 2019_2_1_mac_system_disable_test_get_set_computer_name
  - 5d537ed Missing six import. Updating skip message.
  - ac6dccd Disabling test_get_set_computer_name on OS X and Py3.
- **PR** #53913: (garethgreenaway) [2019.2.1] skip test_issue_2594_non_invalidated_cache on MacOS @ *2019-07-19 17:17:36 UTC*
  - a54ec89 Merge pull request #53913 from garethgreenaway/disable_test_issue_2594_non_invalidated_cache_macos
  - c65c5d5 Skip the test test_issue_2594_non_invalidated_cache on MacOS where it is flakey.
- **PR** #53902: (twangboy) Skip tests that modify date or time @ *2019-07-18 17:53:31 UTC*
  - b091eb7 Merge pull request #53902 from twangboy/skip_dt
  - 76cf936 Skip tests that modify date or time
- **PR** #53901: (s0undt3ch) [2019.2.1] Don't fail when combining coverage files @ *2019-07-18 17:46:32 UTC*
  - 22b7b1a Merge pull request #53901 from s0undt3ch/hotfix/wrap-coverage-combine
  - b459a48 Don't fail when combining coverage files
- **PR** #53542: (dwoz) Fix leak of SaltMessageClient instances when using tcp transport @ *2019-07-18 17:44:44 UTC*
  - 84e798a Merge pull request #53542 from dwoz/tcp_leak
  - 3daeb4f Update TCP pipelines
  - 52f9556 Merge branch '2019.2.1' into tcp_leak
  - f5313fd Fix tcp message client test
  - 3419bf5 Merge branch '2019.2.1' into tcp_leak
  - 3125cd3 Merge branch '2019.2.1' into tcp_leak
  - 9a33582 Merge branch '2019.2.1' into tcp_leak
  - 7eb5d41 Merge branch '2019.2.1' into tcp_leak
  - 481372b Fix torando loop thread issue
  - aff15d4 Fix linter wart
  - 9606db7 Comment the tcp message client test better
  - dcf576f Add unit test for tcp message client close method
  - 4420556 Download artifacts for tcp jobs
  - e6a09dd Merge remote-tracking branch 'origin/tcp_leak' into tcp_leak
    - f484f3f Merge branch '2019.2.1' into tcp_leak

  - 4a37234 Add tcp suffix to github notifications
  - 4a2848b Add transport PR tests
  - 9a4b407 Fix leak of SaltMessageClient instances when using tcp transport
    - 592f3fe undo test
    - bc550ad wip pytest test david
    - 75571e4 fixs timezone decode error
    - d041660 fixes timezone decode error
    - 1ddd1a9 fixs timezone decode error
    - e75dafa patches timezone grain
    - eaca473 patches salt grains locale_info decode error
- **PR** #53873: (bryceml) increase sleep time between kitchen create failures to account for ap… @ *2019-07-16 23:46:09 UTC*
  - 61e9efd Merge pull request #53873 from bryceml/2019.2.1-increase-sleep
  - aae0bf6 increase sleep time between kitchen create failures to account for api limits
- **PR** #53750: (twangboy) Fix memory error when the test suite cleans up (2019.2.1) @ *2019-07-16 23:00:07 UTC*
  - 317d9af Merge pull request #53750 from twangboy/fix_helpers
  - fa0e9c3 Merge branch '2019.2.1' into fix_helpers
- **PR** #53851: (dwoz) Master stats revert @ *2019-07-16 16:05:38 UTC*
  - 6c2b3a5 Merge pull request #53851 from dwoz/master_stats_revert
  - 0bc72e4 Revert "Merge pull request #53822 from dwoz/master_stats_test"
  - 4553ba7 Revert "Merge pull request #53829 from saltstack/fix_stats_2019.2.1"
- **PR** #53829: (dwoz) Fix stats on windows @ *2019-07-12 01:03:47 UTC*
  - 5064027 Merge pull request #53829 from saltstack/fix_stats_2019.2.1
  - 9f2d20d Fix stats on windows
- **PR** #53826: (dmurphy18) WAR ROOM test skip till rewritten to allow for dnf on RHEL 8 and F30 @ *2019-07-12 00:39:11 UTC*
  - 971eda3 Merge pull request #53826 from dmurphy18/fedora30_fixes
  - 34b261d WAR ROOM test skip till rewritten to allow for dnf on RHEL 8 and Fedora 30
- **PR** #53822: (dwoz) Enable master stats for tests @ *2019-07-11 23:37:55 UTC*
  - ba33d76 Merge pull request #53822 from dwoz/master_stats_test
  - d2b8315 Enable master stats for tests
  - 3471422 Fix memory error when the test suite cleans up
- **PR** #53591: (twangboy) Fix whitelist errors in *test_boto_** state tests on Windows @ *2019-07-07 05:45:21 UTC*
  - 1756156 Merge pull request #53591 from twangboy/fix_boto_tests
  - ba8ba26 Merge branch '2019.2.1' into fix_boto_tests
- **ISSUE** #53532: (dafyddj) win_lgpo.py: crash caused by empty presentation text element (refs: #53662)
- **PR** #53688: (twangboy) Merge Forward #53662 (2019.2.1) @ *2019-07-07 05:44:25 UTC*

  - **PR** #53662: (lomeroe) Update win_lgpo (refs: #53688)
  - 464464c Merge pull request #53688 from twangboy/mf_53662
  - c19dc97 Merge branch '2019.2.1' into mf_53662
- **ISSUE** #52391: (rsmekala) Port Junos-related bug fixes from develop to 2019.2 (refs: #52401)
- **PR** #52401: (rsmekala) Port Junos-related bug fixes from develop to 2019.2 @ *2019-07-06 21:50:51 UTC*
  - **PR** #51164: (rsmekala) Updates to salt-junos modules (refs: #52401)
  - 1a76e00 Merge pull request #52401 from rsmekala/2019.2.1
  - 345938d Merge branch '2019.2.1' into 2019.2.1
  - 326f9f4 Merge branch '2019.2.1' into 2019.2.1
  - c11a004 Merge branch '2019.2.1' into 2019.2.1
  - 33b45d2 Merge branch '2019.2.1' into 2019.2.1
  - af66fac Merge branch '2019.2.1' into 2019.2.1
  - 211b169 Ported relevant fixes from unit.modules.test_junos from develop to 2019.2
  - 0c51bc9 Ported relevant fixes from states.junos from develop to 2019.2
  - a966ad1 Ported relevant fixes from proxy.junos from develop to 2019.2
  - 263e9f5 Ported relevant fixes from modules.junos from develop to 2019.2
    - b9033b9 Merge branch '2019.2.1' into mf_53662
    - 75e740e Merge forward #53662
      - 8ace391 Merge branch '2019.2.1' into fix_boto_tests
- **PR** #53585: (twangboy) Fix *test_winrepo* on Windows @ *2019-07-06 15:15:47 UTC*
  - 056f596 Merge pull request #53585 from twangboy/fix_test_winrepo
  - c1f8cba Merge branch '2019.2.1' into fix_test_winrepo
- **PR** #53590: (twangboy) Fix *test status* on Windows @ *2019-07-06 15:15:13 UTC*
  - db950f2 Merge pull request #53590 from twangboy/fix_test_status
  - 67750b7 Merge branch '2019.2.1' into fix_test_status
  - b667045 Merge branch '2019.2.1' into fix_test_status
  - 9b4338a Merge branch '2019.2.1' into fix_test_status
  - 2735b19 Add __grains__
  - bb12da9 Load grains properly
    - 91da774 Merge branch '2019.2.1' into fix_test_winrepo
- **PR** #53556: (twangboy) Fixes an issue with line endings in the jinja renderer @ *2019-07-05 23:06:01 UTC*
  - becaf12 Merge pull request #53556 from twangboy/fix_test_pillar
  - ac405d1 Merge branch '2019.2.1' into fix_test_pillar
- **PR** #53557: (twangboy) Fix test_pydsl on Windows @ *2019-07-05 22:36:40 UTC*
  - 7a5111b Merge pull request #53557 from twangboy/fix_test_pydsl
  - d9df951 Merge branch '2019.2.1' into fix_test_pydsl

    - 1619c68 Merge branch '2019.2.1' into fix_test_pydsl
    - 2c3e91d Fix test_pydsl on Windows
      - 137f4a4 Merge branch '2019.2.1' into fix_test_pillar
      - 4a96252 Fixes an issue with line endings in the jinja renderer
        - c8c8bc0 Merge branch '2019.2.1' into fix_test_winrepo
        - 8373865 Merge branch '2019.2.1' into fix_test_winrepo
        - a818396 Split out the tests, patch test=True
        - 62041eb Merge branch '2019.2.1' into fix_boto_tests
- **PR** #53653: (s0undt3ch) [2019.2.1] Log which address failed to resolve @ *2019-07-05 18:29:31 UTC*
  - 8f05226 Merge pull request #53653 from s0undt3ch/hotfix/fix-nox-bypass-2019.2.1
  - f44253c Merge branch '2019.2.1' into hotfix/fix-nox-bypass-2019.2.1
  - 0ead7fe Log which address failed to resolve
- **PR** #53725: (s0undt3ch) [2019.2.1] Archive the kitchen logs @ *2019-07-05 18:19:50 UTC*
  - f750f44 Merge pull request #53725 from s0undt3ch/features/f30-reqs-2019.2.1
  - 118fbf7 Improve slack message
  - 62a2ee8 Archive the kitchen logs
- **PR** #53689: (twangboy) Merge Forward #52593 (2019.2.1) @ *2019-07-05 14:58:10 UTC*
  - **PR** #52593: (twangboy) Update setup.py (refs: #53689)
  - 0c6009f Merge pull request #53689 from twangboy/fix_setup_2019.2.1
  - c6f3da5 Merge branch '2019.2.1' into fix_setup_2019.2.1
- **PR** #53690: (twangboy) Merge Forward #52065 (2019.2.1) @ *2019-07-05 14:55:12 UTC*
  - **PR** #52065: (twangboy) Use the dism binary that matches system architecture (refs: #53690)
  - 54c4220 Merge pull request #53690 from twangboy/fix_win_dism_2019.2.1
  - fae9f1a Merge branch '2019.2.1' into fix_win_dism_2019.2.1
- **PR** #53719: (s0undt3ch) [2019.2.1] PR Pipeline Enhancements @ *2019-07-04 20:29:06 UTC*
  - 78c45eb Merge pull request #53719 from s0undt3ch/features/f30-reqs-2019.2.1
  - 76b99fd Enhance lint report
  - 45f19af Use milestones to abort previous, still running, builds, on new builds
- **PR** #53697: (s0undt3ch) [2019.2.1] Add Fedora 30 requirements files @ *2019-07-04 04:26:02 UTC*
  - 3db4ddb Merge pull request #53697 from s0undt3ch/features/f30-reqs-2019.2.1
  - 36198cd Delete Opensuse 42 static requirements
  - 4413626 Delete Fedora 28 static requirements
  - 41809e5 Add Fedora 30 requirements files
- **PR** #53680: (Ch3LL) Pytest 5.0 contextmanager str: call value on ExceptionInfo objects @ *2019-07-04 00:12:56 UTC*
  - f5c5da4 Merge pull request #53680 from Ch3LL/pytest_5_changes

  - b5e2b0b Merge branch '2019.2.1' into pytest_5_changes
- **PR** #53682: (twangboy) Fix compare issue in lgpo state module @ *2019-07-03 04:43:22 UTC*
  - 727843c Merge pull request #53682 from twangboy/fix_lock_1740_lgpo
  - 62450db Merge branch '2019.2.1' into fix_lock_1740_lgpo
  - f6276a3 Fix compare issue in lgpo state module
    - 110b953 Pytest 5.0 contextmanager str: call value on ExceptionInfo objects
    - b71b655 Merge forward #52065
    - 023c47d Fix typo
    - 43dd7b3 Merge Forward #52593
- **PR** #53678: (twangboy) Fix LGPO when string object is None @ *2019-07-02 23:47:05 UTC*
  - 491bfa8 Merge pull request #53678 from twangboy/fix_lock_1688_lgpo
  - 7b05bf8 Merge branch '2019.2.1' into fix_lock_1688_lgpo
- **PR** #53220: (twangboy) Don't remove the pythonwin directory (2019.2.1) @ *2019-07-02 22:43:43 UTC*
  - 4a4a91b Merge pull request #53220 from twangboy/fix_pywin32_2019.2.1
  - 748bf1c Merge branch '2019.2.1' into fix_pywin32_2019.2.1
- **PR** #53614: (Ch3LL) Increase flaky attempts on mac tests using systemsetup @ *2019-07-02 21:33:56 UTC*
  - c2befe2 Merge pull request #53614 from Ch3LL/mac_flaky
  - 5460031 Merge branch '2019.2.1' into mac_flaky
  - 7dcaaf0 Increase flaky attempts on mac tests using systemsetup
- **PR** #53624: (Ch3LL) Allow yaml list notation for nodegroup expansion @ *2019-07-02 21:32:43 UTC*
  - 02461cb Merge pull request #53624 from Ch3LL/nodegroup_group_list
  - 475d904 Merge branch '2019.2.1' into nodegroup_group_list
  - fbb15cd Add nodegroup list test
  - ee59d39 Merge branch '2019.2.1' into nodegroup_group_list
  - d236bd4 Allow yaml list notation for nodegroup expansion
- **PR** #53562: (Ch3LL) Move create key call into try/except in Digital Ocean key test @ *2019-07-02 16:08:08 UTC*
  - 8857dbd Merge pull request #53562 from Ch3LL/do_key_test
  - 315eb35 Merge branch '2019.2.1' into do_key_test
  - 75ac708 Merge branch '2019.2.1' into do_key_test
  - 128ba07 Move create key call into try/except in Digital Ocean key test
    - 22d4a3a Merge branch '2019.2.1' into fix_pywin32_2019.2.1
    - 6bb6df7 Merge branch '2019.2.1' into fix_pywin32_2019.2.1
    - 534c984 Merge branch '2019.2.1' into fix_pywin32_2019.2.1
    - 382c637 Merge branch '2019.2.1' into fix_pywin32_2019.2.1
    - fb4090d update py3 script
    - 0ea70ba Don't remove pythonwin directory

      - ba37276 Use string_types instead of text_types
      - f6d0084 Add some unit tests, raise error on non-string types
      - 35ed8be Create function for encoding string values
      - c4c1082 Return encoded null when string value is None
- **PR** #53627: (dmurphy18) Cherry pick pr 53370 from 2018.3 into 2019.2.1 @ *2019-06-28 19:54:13 UTC*
  - 164aaeb Merge pull request #53627 from dmurphy18/cherry-pick-pr-53370
  - 450d23d Merge branch '2019.2.1' into cherry-pick-pr-53370
- **ISSUE** #53411: (cro) Events can grow stale when event_listen_queue is set. (refs: #53587, #53412)
- **PR** #53587: (cro) Forward port from 2018.3 Add event_listen_queue_max_seconds to fix #53411 @ *2019-06-28 16:49:46 UTC*
  - **PR** #53412: (cro) Stale events 53411 (refs: #53587)
  - 6559e4c Merge pull request #53587 from cro/53412-2019.2.1
  - 9e69bd4 Merge branch '2019.2.1' into 53412-2019.2.1
  - 949a026 Merge branch '2019.2.1' into 53412-2019.2.1
  - f5d81c9 Merge branch '2019.2.1' into 53412-2019.2.1
  - d74c313 Merge branch '2019.2.1' into 53412-2019.2.1
  - b3c3f52 Make doc build test pass.
  - 8001130 Add event_listen_queue_max_seconds to fix #53411
    - 34e38d7 Merge branch '2019.2.1' into cherry-pick-pr-53370
- **ISSUE** #53283: (Ch3LL) integration.states.test_file.FileTest.test_directory_broken_symlink (refs: #53295)
- **PR** #53295: (DmitryKuzmenko) Recurse kwarg of state.directory state shall be a list or None @ *2019-06-27 23:20:10 UTC*
  - 14efced Merge pull request #53295 from DSRCorporation/bugs/test_file_recurse_set
  - 2927b78 Merge branch '2019.2.1' into bugs/test_file_recurse_set
  - 9c01670 Merge branch '2019.2.1' into bugs/test_file_recurse_set
  - 571d82a Merge branch '2019.2.1' into bugs/test_file_recurse_set
  - ee8f0e3 Merge branch '2019.2.1' into bugs/test_file_recurse_set
  - 013e04c Recurse kwarg of state.directory state shall be a list of None
- **ISSUE** #52926: (waynew) integration.states.test_beacon.BeaconStateTestCase.test_present_absent - Beacon "diskusage" is not available (refs: #53493, #53466)
- **ISSUE** #52245: (twangboy) integration.states.test_beacon.BeaconStateTestCase.test_present_absent (refs: #53493, #53466)
- **PR** #53466: (dwoz) More robust beacon state test @ *2019-06-27 23:17:01 UTC*
  - 75175bb Merge pull request #53466 from dwoz/test_pres_abs
  - f24bf06 Use new pipeline format
  - a74580b Clean up cruft
  - 68958f5 Do not create duplicate minion ids

  - c47eeb4 Add amazon 2 to PR tests
  - dc90b47 More robust beacon state test
    - 95b1819 Initial working tests for gpg fixes and import keys, signing
    - b2c9ae2 Update access to str/bytes with to_unicode/to_bytes for Python 3
- **PR** #53609: (s0undt3ch) CI Pipelines. Timeout after getting a node. Report exit code. @ *2019-06-27 16:09:54 UTC*
  - 582ac5f Merge pull request #53609 from s0undt3ch/hotfix/fix-nox-bypass-2019.2.1
  - 024be84 CI Pipelines. Timeout after getting a node. Report exit code.
- **PR** #53574: (s0undt3ch) [2019.2.1] Minor fixes/adjustments to the new CI pipelines @ *2019-06-25 18:48:25 UTC*
  - f568796 Merge pull request #53574 from s0undt3ch/hotfix/fix-nox-bypass-2019.2.1
  - e85e2f4 Minor fixes/adjustments to the new CI pipelines
- **PR** #53584: (dwoz) Add unit tests for recent SaltCacheLoader changes @ *2019-06-24 23:40:43 UTC*
  - **PR** #53563: (twangboy) SaltCacheLoader does not create multiple FileClients (refs: #53584)
  - 77b7fc4 Merge pull request #53584 from dwoz/client_cache
  - e9b61dc Add unit tests for recent SaltCacheLoader changes
    - 5558a0a Add reg to the whitelist
- **PR** #53563: (twangboy) SaltCacheLoader does not create multiple FileClients (refs: #53584) @ *2019-06-22 23:23:03 UTC*
  - 6ed6c31 Merge pull request #53563 from twangboy/py3_windows_hang
  - 369720e Allow the file_client to be overridden
  - f6c592c Don't instantiate the file_client every time
- **PR** #53432: (garethgreenaway) [2019.2.1] Various fixes for 2019.2.1 to ensure tests pass on Mac OS X @ *2019-06-21 13:47:41 UTC*
  - c606952 Merge pull request #53432 from garethgreenaway/2019_2_1_mac_fixes
  - 1974f11 Disabling a couple more tests in integration.modules.test_cp that hang on OS X and Py2.
  - 7721151 Fixing lint.
  - 1d4e228 Updating various skips for tests to only skip if OS is OS X and Python is Python2.
  - 01dafdc Skipping integration.modules.test_cp.CPModuleTest.test_get_file_str_https on OS X
  - c02db20 Disabling integration.modules.test_state.StateModuleTest.test_parallel_state_with_long_tag
  - 8788877 Skip integration.states.test_file.FileTest.test_issue_11003_immutable_lazy_proxy_sum and integration.states.test_pip_state.PipStateTest.test_22359_pip_installed_unless_does_not_trigger_warnings when running on OS X.
  - cf76027 Removing wrong import of skipIf
  - b10e1e7 Moving the skipIf for OS X from the beacon state tests to the renderer state tests.
  - d2fef9a Disabling beacon state tests on OS X for the time being.
  - 1741bb0 Adding some timeouts to see if it helps the tests pass on OS X.
  - fd0ba0a Ensure the user and group used by @with_system_user_and_group are consistent.
  - e03ab81 Using salt.utils.path.which to find false, on OSX it lives under /usr/bin/

  - 353f9d4 Fixing the beacons.reset function. Once the reset has taken place in beacons/__init__.py we need to fire an event back to complete the loop and ensure that everything worked as expected.
  - ee3cbc7 fix to how the depends decorator works. Only run the dependancy commands for the module we're checking.
  - 8440176 Fixing a log issue that pops up after test_gen_thin_compression_fallback_py3 on "OS X, need to ensure that salt.utils.thin.os.close is mocked.
  - 9767ddd Format for the sqlite3 databse used for the assistive information changed in Mojave, additional columns added.
  - 9c8a7e6 Fixing a bug when the roots fileserver and the location is a symlink to another location. This fix ensures that when fsroot is referenced we are using the real path and not the symlink path.
- **PR** #53526: (s0undt3ch) [2019.2.1] Update pipelines to work on old and new jenkins @ *2019-06-20 15:29:13 UTC*
  - 59e2a1f Merge pull request #53526 from s0undt3ch/hotfix/fix-nox-bypass-2019.2.1
  - 24d6d09 Update pipelines to work on old and new jenkins
- **PR** #53210: (Ch3LL) Cherry-Pick #52787 into 2019.2.1 @ *2019-06-20 13:33:13 UTC*
  - **PR** #52787: (garethgreenaway) [2018.3] Fixes to test_pip state sls files (refs: #53210)
  - f5f80af Merge pull request #53210 from Ch3LL/cp-52787
  - f27c434 Merge branch '2019.2.1' into cp-52787
- **PR** #53467: (twangboy) Check valid username first (fixes failing symlink test) @ *2019-06-19 17:04:30 UTC*
  - fd6cb35 Merge pull request #53467 from twangboy/fix_test_win_file
  - 92950c5 Merge branch '2019.2.1' into fix_test_win_file
  - 6a21edb Merge branch '2019.2.1' into fix_test_win_file
  - 31ce1fb Fix some lint, skip some tests
  - 6de4db6 Monkeypatch in the class instead of globally
  - d663a1f Fix the failing BlockReplace test cases on Windows
  - 299f88f Update test_managed_contents
  - 81110e5 Fix test_file_copy_make_dirs that was failing on Linux
  - 8d0529d Fix some lint
  - 4a9c020 Fix issues with win_runas
  - bceffa1 Remove privs message
  - 54be0a6 Verify username early in win_runas
- **PR** #53475: (Ch3LL) Add pypsexec requirement for cloud tests @ *2019-06-19 13:30:25 UTC*
  - 3230078 Merge pull request #53475 from Ch3LL/add_pypsexec
  - ab59a55 Merge branch '2019.2.1' into add_pypsexec
  - f954363 Merge branch '2019.2.1' into add_pypsexec
  - 8f7cbfc Merge branch '2019.2.1' into add_pypsexec
  - fa22b39 Add pypsexec requirement for cloud tests
- **PR** #53491: (Ch3LL) Update test_gen_hash for macosx @ *2019-06-19 13:16:04 UTC*
  - bdd7c2c Merge pull request #53491 from Ch3LL/crypt_test_mac

  - 85e96bd Merge branch '2019.2.1' into crypt_test_mac
  - 700338e Merge branch '2019.2.1' into crypt_test_mac
  - 4ce7fb1 Merge branch '2019.2.1' into crypt_test_mac
  - 832ded6 Merge branch '2019.2.1' into crypt_test_mac
  - 775b8c2 Merge branch '2019.2.1' into crypt_test_mac
  - 0832b49 Update test_gen_hash for macosx
- **PR** #53494: (Ch3LL) Always delete digital ocean key for test_key_management test @ *2019-06-19 13:14:51 UTC*
  - 8c2e952 Merge pull request #53494 from Ch3LL/do_cloud_test
  - 5053eab Merge branch '2019.2.1' into do_cloud_test
  - 64a3e8d Merge branch '2019.2.1' into do_cloud_test
  - 45c0c43 Merge branch '2019.2.1' into do_cloud_test
  - 3826c56 Allows delete digital ocean key for test_key_management test
    - 6b6dc66 Merge branch '2019.2.1' into cp-52787
- **PR** #53434: (weswhet) Update documentation for newer mac_service module. @ *2019-06-19 01:15:19 UTC*
  - 2e3778b Merge pull request #53434 from weswhet/patch-2
  - 89398a9 Update salt.modules.service.rst
  - b9c1b1a Update salt.modules.service.rst
  - a962a64 Merge branch '2019.2.1' into patch-2
- **PR** #53498: (amendlik) Fix broken documentation links @ *2019-06-18 23:44:05 UTC*
  - f72ebba Merge pull request #53498 from amendlik/links
  - e360a70 Fix broken documentation links for service virtual module
  - 56f65ec Fix broken documentation links for pkg virtual module
    - 010a2c5 Update documentation for newer mac_service module.
      - 1bbdc7f Merge branch '2019.2.1' into cp-52787
- **PR** #53514: (Ch3LL) Revert logging changes from #53492 @ *2019-06-18 20:49:04 UTC*
  - **PR** #53492: (dwoz) Fix syndic connection when using tcp transport (refs: #53514)
  - 0793272 Merge pull request #53514 from Ch3LL/disable_py3_logging
  - 8c8f0ac import six runtests log handler
  - f442b33 Revert "Enable logging for test runs on py3"
    - b62be16 Merge branch '2019.2.1' into cp-52787
- **PR** #53485: (s0undt3ch) [2019.2.1] Workaround nox's install only flag @ *2019-06-18 09:35:34 UTC*
  - 98285f9 Merge pull request #53485 from s0undt3ch/hotfix/fix-nox-bypass-2019.2.1
  - 69e1d84 Workaround nox's install only flag
    - 8622bba Merge branch '2019.2.1' into cp-52787
- **PR** #53369: (Akm0d) Added refs to AIX module documentation @ *2019-06-17 20:29:01 UTC*
  - ff7370e Merge pull request #53369 from Akm0d/aix_docs

  - 2ece253 Merge branch '2019.2.1' into aix_docs
  - f2eda45 Merge branch '2019.2.1' into aix_docs
  - 3e793ac Merge branch '2019.2.1' into aix_docs
  - e800641 Merge branch '2019.2.1' into aix_docs
  - 0ef9892 Merge branch '2019.2.1' into aix_docs
  - 65cb718 Merge branch '2019.2.1' into aix_docs
  - 32f4d7e Merge branch '2019.2.1' into aix_docs
  - a59f45e Merge branch '2019.2.1' into aix_docs
  - 29f89a4 Merge branch '2019.2.1' into aix_docs
  - e74345c Merge branch '2019.2.1' into aix_docs
  - 28fbd11 Merge branch '2019.2.1' into aix_docs
  - 60129f0 Merge branch 'aix_docs' of github.com:Akm0d/salt into aix_docs
    - a1c4abc Merge branch '2019.2.1' into aix_docs
    - e2c9fcc Merge branch '2019.2.1' into aix_docs
  - fd197d3 Merge branch '2019.2.1' of https://github.com/saltstack/salt into aix_docs
  - 6d42cf7 Merge branch 'aix_docs' of github.com:Akm0d/salt into aix_docs
    - fae6045 Merge branch '2019.2.1' into aix_docs
  - 97145b0 Merge branch '2019.2.1' of https://github.com/saltstack/salt into aix_docs
  - 6d16343 Updated 'used for' description of aixpkg
  - 6092683 Added aixpkg to index
  - 2867d07 Added refs to AIX module documentation
    - 458fe9f Merge branch '2019.2.1' into cp-52787
- **PR** #53461: (waynew) Update file.chattr @ *2019-06-14 21:16:52 UTC*
  - 8df7684 Merge pull request #53461 from waynew/fix-chattr-problems
  - a8d8174 Merge branch '2019.2.1' into fix-chattr-problems
- **ISSUE** #52926: (waynew) integration.states.test_beacon.BeaconStateTestCase.test_present_absent - Beacon "diskusage" is not available (refs: #53493, #53466)
- **ISSUE** #52245: (twangboy) integration.states.test_beacon.BeaconStateTestCase.test_present_absent (refs: #53493, #53466)
- **PR** #53493: (dwoz) Cherry-pick and merge beacon event change @ *2019-06-14 20:36:10 UTC*
  - cfe866d Merge pull request #53493 from dwoz/cherry_pick_test_fix
  - fb104bb Cherry-pick and merge beacon event change
- **PR** #53492: (dwoz) Fix syndic connection when using tcp transport (refs: #53514) @ *2019-06-14 19:23:41 UTC*
  - a1f4136 Merge pull request #53492 from dwoz/tcp_syndic_fix
  - 17c983b Fix linter
  - 9339425 Enable logging for test runs on py3
  - f1b65d1 Fix syndic connection when using tcp transport

- **PR** #53437: (twangboy) Fix failing symlink test @ *2019-06-12 22:53:32 UTC*
  - e852596 Merge pull request #53437 from twangboy/fix_test_win_file
  - 8c30dbd Add priv info to error message
  - a7d41a8 Make them non-destructive tests
  - d8bede0 Remove duplicate test
  - 89aaf2e Remove duplicate test
  - 184ec4a Skip test on linux
  - c97ea99 Mark it a destructive test
  - 9da7090 Fix failing symlink test
- **PR** #53408: (s0undt3ch) [2019.2.1] Fix static requirements @ *2019-06-12 22:52:33 UTC*
  - a92836b Merge pull request #53408 from s0undt3ch/hotfix/fix-pkg-reqs-2019.2.1
  - 19629a0 Fix *unit.templates.test_jinja.TestCustomExtensions.test_http_query*
  - 5aee99b Take into account the packaging requirements for OSX
  - b9618f6 Take into account the packaging requirements for windows
  - ff63ae4 Stop compiling static TCP requirements.
    - 62e590b Update file.chattr
- **PR** #53165: (Ch3LL) Update ami's used for windows cloud tests @ *2019-06-11 15:54:41 UTC*
  - ab9fe46 Merge pull request #53165 from Ch3LL/ami_window_cloud
  - 1d0b0a6 Merge branch '2019.2.1' into ami_window_cloud
  - a77a05c Merge branch '2019.2.1' into ami_window_cloud
  - a1204dc Merge branch '2019.2.1' into ami_window_cloud
  - d15c8f2 Merge branch '2019.2.1' into ami_window_cloud
  - 941778f Update ami's used for windows cloud tests
  - **PR** saltstack/salt#53332: (s0undt3ch) Non optional full test runs for 2019.2.1 (refs: #53431)
- **PR** #53431: (dwoz) Revert "Non optional full test runs for 2019.2.1" @ *2019-06-10 21:44:30 UTC*
  - d806b58 Merge pull request #53431 from saltstack/revert-53332-hotfix/full-test-run-2019.2.1
  - 701218c Revert "Non optional full test runs for 2019.2.1"
- **ISSUE** #52174: (amendlik) file.stat function not working under Python 3 (refs: #53430)
  - **PR** #53430: (Akm0d) Cherry-pick #52174 into 2019.2.1
    - 7811971 Merge branch '2019.2.1' into cp-52787
- **PR** #53389: (bryceml) only keep last 10 builds of every pr on jenkins to reduce disk usage @ *2019-06-10 17:18:48 UTC*
  - 6b800a2 Merge pull request #53389 from bryceml/2019.2.1
  - 05368a1 Merge branch '2019.2.1' into 2019.2.1
- **PR** #53332: (s0undt3ch) Non optional full test runs for 2019.2.1 @ *2019-06-10 15:51:41 UTC*
  - fe18c40 Merge pull request #53332 from s0undt3ch/hotfix/full-test-run-2019.2.1

    - 2372733 Merge branch '2019.2.1' into hotfix/full-test-run-2019.2.1
    - f8bd47b Merge branch '2019.2.1' into hotfix/full-test-run-2019.2.1
    - 45c3c06 Non optional full test runs for 2019.2.1
        - 928b05f only fetch pr target branch instead of all branches to save time and disk space
        - ccc6106 only keep last 10 builds of every pr on jenkins to reduce disk usage
            - b970bde Merge branch '2019.2.1' into cp-52787
- **PR** #53287: (twangboy) Fix win system @ *2019-06-06 23:35:47 UTC*
    - d0810d7 Merge pull request #53287 from twangboy/fix_win_system
    - 0fc88fb Add TypeError
    - 5b4160d Try NumberOfEnabledCores
    - 49cbfd4 Add a note about skipping unavailable items
    - 044b56b Fix get_system_info for older versions of Windows
- **ISSUE** #52173: (amendlik) file.directory fails to check directory permissions correctly (refs: #53385)
    - **PR** #53385: (Akm0d) Check dir_mode recursively in file.directory
- **PR** #53386: (dwoz) Fix missing import @ *2019-06-06 20:45:59 UTC*
    - cf98b83 Merge pull request #53386 from dwoz/missing_import
    - a23d6d9 Fix missing import
- **ISSUE** #49559: (zyguy) Salt-cloud - proxmox driver - returns AttributeError: 'generator' object has no attribute 'next' (refs: #53240)
    - **PR** #53240: (FireGrace) change .next() to next() py2=>py3 leftover
- **ISSUE** #53274: (Ch3LL) integration.states.test_pkg failing on amazon 1 py2 (refs: #53323)
- **PR** #53323: (dmurphy18) Fix for issue #53274, test on Amazon Linux 1 @ *2019-06-06 17:31:18 UTC*
    - 18991f9 Merge pull request #53323 from dmurphy18/fix_53274
    - 6814852 Merge branch '2019.2.1' into fix_53274
    - bb6c97a Merge branch '2019.2.1' into fix_53274
    - b84833d Merge branch '2019.2.1' into fix_53274
    - 7085f36 Fixed pylint issue on PR not showing when run pylint locally
    - 1200031 Updated to use package bash-doc if Amazon Linux 1 after review comments
    - b2a4a5e Fix for issue #53274, test on Amazon Linux 1
    - **PR** #53356: (Akm0d) Updated paramiko to version 2.2.3 for python3.7 support
- **ISSUE** #53137: (bryceml) update doc pr's to be built using python3 and sphinx 2.0.1 (refs: #53273)
- **PR** #53273: (s0undt3ch) [2019.2.1] Switch docs building under Py3 @ *2019-06-05 10:01:03 UTC*
    - 1cf57e9 Merge pull request #53273 from s0undt3ch/hotfix/py3-nox-docs-2019.2.1
    - 98653c0 Allow docs to be built on Python >= 3.5, not just Python 3.6
    - c54f06f Have sphinx turn errors into warnings
    - 07f4327 Switch docs building under Py3

- **PR** #53361: (s0undt3ch) [2019.2.1] Only make a new log record if it's a dictionary. @ *2019-06-05 09:56:15 UTC*
  - dfd36a2 Merge pull request #53361 from s0undt3ch/hotfix/fix-mp-logging-2019.2.1
  - d70d61f Fully revert 962b11687
- **ISSUE** #53171: (twangboy) integration.states.test_network.NetworkTest.test_managed (refs: #53351)
- **PR** #53351: (waynew) Add checks for Amazon Linux to ip module @ *2019-06-04 19:25:58 UTC*
  - 5efb670 Merge pull request #53351 from waynew/53171-fix-network-managed
  - e801afe Add checks for Amazon Linux to ip module
- **PR** #53242: (s0undt3ch) [2019.2.1] newer msgpack ipc fixes @ *2019-06-03 23:39:09 UTC*
  - **PR** #52934: (twangboy) Update msgpack calls for newer msgpack (refs: #53242)
  - **PR** #52755: (dwoz) Fix non raw msg pack msg decoding (refs: #53242)
  - **PR** #52488: (terminalmage) Fix deprecation warning in msgpack >= 0.5.2 (2018.3) (refs: #52755)
  - **PR** #52487: (terminalmage) Fix deprecation warning in msgpack >= 0.5.2 (develop) (refs: #53242)
  - ce5d79a Merge pull request #53242 from s0undt3ch/hotfix/msgpack-ipc-2019.2.1
  - d27a524 Merge branch '2019.2.1' into hotfix/msgpack-ipc-2019.2.1
- **PR** #53289: (cmcmarrow) stops pylint E1120 from showing @ *2019-06-01 04:57:27 UTC*
  - df448c2 Merge pull request #53289 from cmcmarrow/test_kubernetes_lint_failures_silenced
  - 03eacac Merge branch '2019.2.1' into test_kubernetes_lint_failures_silenced
- **PR** #53304: (s0undt3ch) [2019.2.1] Refactor Jenkins PR pipelines to download artifacts after timeout @ *2019-05-31 17:10:19 UTC*
  - 6fac22b Merge pull request #53304 from s0undt3ch/hotfix/jenkins-pipelines-refactor-2019.2.1
  - 6e778ef Refactor Jenkins PR pipelines to download artifacts after timeout
- **PR** #53297: (s0undt3ch) [2019.2.1] The TCP transport needs the exact same requirements as the ZeroMQ one @ *2019-05-30 19:42:29 UTC*
  - d359513 Merge pull request #53297 from s0undt3ch/hotfix/tcp-transport-tests
  - bec7fa5 The TCP transport needs the exact same requirements as the ZeroMQ one
  - f93e7d3 stops pylint E1120 from showing
  - cbe6423 stops pylint E1120 from showing
  - c5a5d43 test_kubernetes_lint_failures_silenced
  - b838395 test_kubernetes_lint_failures_silenced
  - 997d0a8 stops pylint E1120 from showing
    - 5542fa8 Add the missing, and required, top pillar file.
    - 0ae2ef0 Under Py2 we still want raw to be set to True
    - 714d663 Re-submit #52934 fixed after being reverted in #52755
    - 243b512 Update msgpack calls for newer msgpack
      - 0f9077b Merge branch '2019.2.1' into cp-52787
- **PR** #53264: (s0undt3ch) [2019.2.1] Minion blackout tests - Add the missing, and required, top pillar file @ *2019-05-29 07:30:45 UTC*

  - 50e31ec Merge pull request #53264 from s0undt3ch/2019.2.1
  - acabb70 Merge branch '2019.2.1' into 2019.2.1
- **PR** #52966: (s0undt3ch) Always run the full test suite on the 2019.2.1 release branch @ *2019-05-28 15:43:12 UTC*
  - 749c626 Merge pull request #52966 from s0undt3ch/hotfix/full-test-run
  - 247c461 Always run the full test suite on the 2019.2.1 release branch
- **PR** #53138: (frogunder) vultr cloudtest fix @ *2019-05-28 14:34:22 UTC*
  - a7afd31 Merge pull request #53138 from frogunder/fix_vultr_cloudtest
  - 9f4550c vultr cloudtest fix
    - 912b9b3 Add the missing, and required, top pillar file.
      - 9d21b75 Merge branch '2019.2.1' into cp-52787
- **PR** #53258: (s0undt3ch) [2019.2.1] Fix multiprocessing logging queue dict changing during iteration errors @ *2019-05-27 19:26:27 UTC*
  - d011beb Merge pull request #53258 from s0undt3ch/2019.2.1
  - ad01dd2 More robust minion blackout tests
  - a0346d1 Try harder to get the minion grains
  - 36717bd Try and address the test flakyness
  - ce07d8c Add more information when the assertion fails
  - b99e914 Attempt to fix mine tests
  - fca981c One more known to return None
  - 951df48 One more known to return None
  - eb5fd3e Fix *integration.modules.test_mine.MineTest.test_get* for sub_minion
  - 26314f5 Fix multiprocessing logging queue dict changing during iteration errors
- **PR** #53153: (s0undt3ch) [2019.2.1] Add Amazon Linux 2018.3 and 2 static requirements @ *2019-05-27 12:02:20 UTC*
  - b69e080 Merge pull request #53153 from s0undt3ch/2019.2.1
  - 166067c Use *currentBuild.resultIsBetterOrEqualTo* instead
  - cefdd90 Stop error messages about missing roster file on syndic master
  - 6bf8f46 Don't complain when attempting to close sockets at this stage
  - e3f3cc9 Add Amazon Linux 2018.3 and 2 static requirements
  - 872acc0 Switch the ubuntu-14.04 exception with amzn-1
  - **PR** saltstack/salt#52934: (twangboy) Update msgpack calls for newer msgpack (refs: #53235)
- **PR** #53235: (s0undt3ch) Revert "Update msgpack calls for newer msgpack" @ *2019-05-24 13:35:20 UTC*
  - 14aeeea Merge pull request #53235 from saltstack/revert-52934-fix_msgpack
  - 3295aea Revert "Update msgpack calls for newer msgpack"
- **PR** #53131: (s0undt3ch) [2019.2.1] Update to salt-bootstrap v2019.05.20 @ *2019-05-24 07:45:57 UTC*
  - 6923427 Merge pull request #53131 from s0undt3ch/hotfix/update-bootstrap-2019.2.1
  - 4cce4f4 Merge branch '2019.2.1' into hotfix/update-bootstrap-2019.2.1

  - 6751ee1 Merge branch '2019.2.1' into hotfix/update-bootstrap-2019.2.1
  - 6988e07 Update to salt-bootstrap v2019.05.20
- **PR** #52934: (twangboy) Update msgpack calls for newer msgpack (refs: #53242) @ *2019-05-23 23:18:23 UTC*
  - a61db20 Merge pull request #52934 from twangboy/fix_msgpack
  - f02a12e Merge branch '2019.2.1' into fix_msgpack
  - 7e2cd34 Merge branch '2019.2.1' into fix_msgpack
  - bc9ce8e Merge branch '2019.2.1' into fix_msgpack
  - 0abd1ec Update msgpack calls for newer msgpack
    - 3366d59 update pylint exceptoin
    - 6c452b3 removing unwanted commits from this branch
    - e0b0ba8 Handling in flaky when maximum number of attempts raised and the exception should be raised. Different approaches depending on Py2 vs Py3.
    - 434dcda Disabling two tests when using Python 3 and OS X
    - 5784e80 Disabling integration.shell.test_minion.MinionTest.test_issue_7754 test on OS X because it hangs the test suite.
    - 78aeb61 Dropping the version check for InstallationError down to anything 1.0 or greater. Removing the test that simulates versions of pip below 1.0.
    - 4ec90c2 Adding an jinja if statement to only the python parameter if the result from get_python_executable is a valid value. Maintaining backwards compatibilty to run tests without Nox.
- **PR** #53192: (twangboy) Skip *test_gen_hash* test on Windows @ *2019-05-23 05:33:13 UTC*
  - 8a57270 Merge pull request #53192 from twangboy/skip_test_gen_hash
  - 758d020 Skip test on Windows
- **PR** #53157: (twangboy) Fix *unit.modules.test_win_file.WinFileTestCase.test_issue_52002_check_file_remove_symlink* @ *2019-05-22 22:32:26 UTC*
  - 446e70c Merge pull request #53157 from twangboy/fix_test_win_file_symlink
  - 5672076 Merge branch '2019.2.1' into fix_test_win_file_symlink
- **PR** #53141: (Ch3LL) Check for all non-word characters when calling secure_password @ *2019-05-22 17:24:54 UTC*
  - 6fbe9aa Merge pull request #53141 from Ch3LL/shadow_fed_fix
  - a3c4066 Merge branch '2019.2.1' into shadow_fed_fix
- **PR** #53161: (Ch3LL) Add HAS_REQUIRED_CRYPTO var for m2crypto in joyent @ *2019-05-22 13:36:07 UTC*
  - fb29512 Merge pull request #53161 from Ch3LL/joyent_m2crypto
  - 08d03e0 Add HAS_REQUIRED_CRYPTO var for m2crypto in joyent
  - 16ef3d2 Merge branch '2019.2.1' into shadow_fed_fix
  - 09ff867 Check for all non-word when calling secure_password
    - 334c93b Elevate token before creating symlink
- **PR** #53073: (Ch3LL) salt-ssh: python binary exists before version check @ *2019-05-20 22:36:34 UTC*

  - 85e9b2f Merge pull request #53073 from Ch3LL/ssh_py3_log
  - 383e781 Improve logging in salt-ssh gen_thin
  - 4371434 salt-ssh: python binary exists before version check
- **PR** #52957: (Ch3LL) Set default_flow_style=None in yaml.dump calls @ *2019-05-16 19:42:35 UTC*
  - bd02ea6 Merge pull request #52957 from Ch3LL/yaml_flow_style
  - 5f6581a Merge branch ’2019.2.1’ into yaml_flow_style
- **PR** #53072: (Ch3LL) Backport #52754 into 2019.2.1 @ *2019-05-16 19:29:22 UTC*
  - **PR** #52754: (garethgreenaway) [2018.3] Fixes to multiprocessing queue when using MacOS (refs: #53072)
  - f410346 Merge pull request #53072 from Ch3LL/bp-52754-2019.2.1
  - 13e5e55 The maximum for the multiprocessing queue on MacOS is 32767, so if we running on MacOS then we use that maximum.
- **ISSUE** #52817: (waynew) unit.modules.test_telegram.TelegramModuleTest.test_post_message (refs: #52972)
- **PR** #52972: (garethgreenaway) [2019.2.1] Fix to unit.modules.test_telegram @ *2019-05-15 07:50:13 UTC*
  - 06fa91b Merge pull request #52972 from garethgreenaway/52817_unit_modules_test_telegram_telegrammoduletest_test_p
  - cea7131 Removing global declaration.
  - 5bf9a95 Fixing lint.
  - 4d3c46d fixing broken unit.modules.test_telegram test.
- **PR** #53025: (Ch3LL) Change package name for suse pkg tests @ *2019-05-15 07:41:44 UTC*
  - d340cbc Merge pull request #53025 from Ch3LL/suse15_pkg_test
  - 0a213ee Change package name for suse pkg tests
- **PR** #53020: (Ch3LL) Change ssh tops log message to debug @ *2019-05-14 16:56:42 UTC*
  - c56fbb0 Merge pull request #53020 from Ch3LL/ssh_tops
  - cbbc764 Merge branch ’2019.2.1’ into ssh_tops
- **PR** #52973: (twangboy) Fix *integration.states.test_pip_state.PipStateTest.test_issue_2028_pip_installed_state* on Windows @ *2019-05-13 20:56:28 UTC*
  - ce099aa Merge pull request #52973 from twangboy/fix_test_pip_state
  - 21838a1 Merge branch ’2019.2.1’ into fix_test_pip_state
- **PR** #52986: (Ch3LL) Revert ”Make sure –run-expensive runtests.py arg works” @ *2019-05-13 19:38:17 UTC*
  - 055d374 Merge pull request #52986 from Ch3LL/expensive_revert
  - 368c012 Revert ”Make sure –run-expensive runtests.py arg works”
  - cce263e Merge branch ’2019.2.1’ into fix_test_pip_state
  - 351d415 Return path to Python instead of None
    - c0538d2 Change ssh tops log message to debug
    - e317186 Merge branch ’2019.2.1’ into yaml_flow_style
- **PR** #52968: (s0undt3ch) [2019.2.1] Salt >= 2018.3.x supports Py3.7 @ *2019-05-12 21:44:57 UTC*
  - ee05da5 Merge pull request #52968 from s0undt3ch/2019.2.1
  - 73c3726 Salt >= 2018.3.x supports Py3.7

- **ISSUE** #50310: (xuhcc) acme.cert falsely reports changes (refs: #50400)
- **PR** #52796: (Ch3LL) Backport #50400 and #50402 into 2019.2.1 @ *2019-05-10 19:30:12 UTC*
  - **PR** #50402: (xuhcc) Fix typo in salt.modules.acme (refs: #52796)
  - **PR** #50400: (xuhcc) Fix incorrect change reporting in acme.cert (refs: #50402, #52796)
  - cae51c2 Merge pull request #52796 from Ch3LL/bp-50400
  - 0c2e3e7 Merge branch '2019.2.1' into bp-50400
  - 1268e3e Merge branch '2019.2.1' into bp-50400
  - 09fa9e3 Fix typo
  - 9be5c0c Fix incorrect change reporting in acme.cert
    - 9a27fb3 Merge branch '2019.2.1' into yaml_flow_style
- **PR** #52770: (twangboy) Fix test_file_managed_should_fall_back_to_binary on Windows @ *2019-05-09 20:35:02 UTC*
  - 20cccd8 Merge pull request #52770 from twangboy/fix_test_file
  - ac27e69 Merge branch '2019.2.1' into fix_test_file
  - 30a5093 Merge branch '2019.2.1' into fix_test_file
  - c08b10d Merge branch '2019.2.1' into fix_test_file
  - 340ee23 Pass encoding to _validate_str_list
    - 0d0e354 Fix yamldumper test for both py2/py3
    - 20bc954 Update docs for yamldumper test
    - bf648e5 Update tests/unit/utils/test_yamldumper.py
    - 70d578a Set default_flow_style=None in yaml.dump calls
- **PR** #52943: (Ch3LL) Fix elasticsearch state module: allow user to define empty aliases @ *2019-05-09 15:46:38 UTC*
  - 4437764 Merge pull request #52943 from Ch3LL/fix_elastisearch
  - 451fb7e Merge branch '2019.2.1' into fix_elastisearch
  - ebff9b9 Fix elasticsearch state module: allow user to define empty aliases
- **PR** #52952: (s0undt3ch) [2019.2.1] More nox changes and test fixes @ *2019-05-09 15:03:07 UTC*
  - 9b290b6 Merge pull request #52952 from s0undt3ch/merge-forward/2018.3-to-2019.2.1-cherry
  - a06d7ce We need to call the decorator
  - 77185af Partial revert since this is a class method
  - f5c5771 Lint fixes
  - 3e43c87 Revert "fix compare_versions method"
  - 0e139bf Fix *GOLDEN_IMAGES_CI_BRANCH* value
  - 12a9408 Add Py3.7 static requirements
  - 8e505fb Move tests to existing test modules
  - 4681609 *tests/unit/config/__init__.py* -> *tests/unit/config/test_config.py*
  - b1dc7f6 Pin to *moto<=1.3.7* because of https://github.com/spulec/moto/pull/1952

  - 0ffe7ca We no longer test againt Ubuntu 14.04 which reached EOL
  - 33743fc Don't traceback when trying to close a closed socket
  - da2e147 Always cleanup the changed environ
  - 087c74d No globals in tests, specially dictionaries!
  - a814fc0 Fix underscore wart (and linter)
  - 74b8f5f Fix deprecation warning in msgpack >= 0.5.2
  - 962b116 Try to address dict changing during iteration
  - c58af57 Tweak codeclimate settings for less noise
  - 5bb7869 fix linter
  - 0b81841 Cherry-pick test fix
  - 99f9e1c fix compare_versions method
- **PR** #52853: (s0undt3ch) [2019.2.1] Bring nox into 2019.2.1 @ *2019-05-09 09:52:42 UTC*
  - a510edf Merge pull request #52853 from s0undt3ch/merge-forward/2018.3-to-2019.2.1-cherry
  - a8f7b46 Test is not destructive.
  - 63663cf We no longer test againt Ubuntu 14.04 which reached EOL
  - 62574ee Typo
  - e3e3761 Proper english in requirements comments
  - e9f6a12 *identical_signature_wrapper* adds *__wrapped__* to function globals
  - 10283b4 Upload coverage reports on full test runs
  - 6e0b61b Less flaky mine tests
  - 3709ed7 msgpack returns string_types
  - a36b15e Add regression test
  - ce782ff Fix non raw msg pack msg decoding
  - f193f0f Update Issue Template: questions removed
  - 4417095 Fix underscore wart (and linter)
  - e6c2ae4 Fix deprecation warning in msgpack >= 0.5.2
  - 6375944 Add ubuntu1804 to pr tests
  - 8fdb04d SIGKILL is not alwasy available use a local variable
  - 4d6b8da Fix linter
  - 759bccc Fix test_doc on windows by using grep yay!
  - 0ce086d Fix remaining failures on ubuntu 1404 and add to PR tests
  - fdf80fd Change py3 windows timeout from 6 to 8 hours
  - b0aa287 Fix missing class wart
  - 21ecb74 fix linter
  - 62c0f9f Skip tests when no libcloud
  - 1d92290 Fix unused import

- 72cdc91 Fix linter
- 238fd0f Fix broken pip state
- 89533ba Fix *NOX_ENV_NAME*
- 7cfc9e8 Fix *NOX_ENV_NAME*
- 2087c91 Fix *NOX_ENV_NAME*
- 2f1aff5 Rename kitchen-centos6-py3 to kitchen-centos6-py2
- 1eea990 Fix linter
- 52ca668 Skip tests when no libcloud
- 2cf4b98 Add centos-6-py2 and debain 8,9 py2 and 3 to PR tests
- 522599d Dont fail just because some random process died
- 34cef86 Ignore super not called
- b7ace9c Fix linter
- aceb6d3 Skip libcloud unit tests when no libcloud
- f73420b Install mock on Python < 3.6
- 28fbde1 Add *moto* to the windows dependencies
- 8cbecc1 Pin kubernetes to <4.0
- 2acb0f3 Make sure –run-expensive runtests.py arg works
- 22cae20 Remove ref restriciton from kitche-salt
- 67095df Add 'runFull' build parameter for PR tests
- 12d8d35 Stop w32time before and start after testing
- 7f1c22c Skip tests that fail on Windows
- f4ae97f Upgrade etcd to > 0.4.2
- 381f5fe Limit and reduce the ammount of log records sent over the wire
- 0c94b5d More entries to ignore
- c3a21f5 As a script, not as a module
- 8948b69 Include *COVERAGE_FILE* as an env variable.
- 0af561e Each generated script is now prepared for code coverage
- 294d6f4 Use the system's path separator
- e52ab87 Always combine and generate the XML coverage report
- 666ca9f Fix proxy minion startup issue on the test suite
- 7838ace *impacket* does not support Py3
- ba8d0fe Allow *–install-only* to gather the required information
- eed7d68 Create nox lint virtualenvs before running them
- e9e57cf Syndic roster not in 2017.7.9
- 111c63a4d Create the roster even when not running ssh
- 11c02a0 Fix unit.test_loader.LoaderGlobalsTest.test_states

- 8694db1 Fix windows tests
- 82e9ac5 Avoid race condition in even assertions
- 9443451 Revert part of 927219c since it is not a fix
- 5fee762 Fix timeout logic
- 517650f check timeout when queue empty
- 7c0a9af fix linter
- 961dc40 Ignore missing variable in six module
- 0f5a7f0 Fix typo
- 6c8c418 Do not limit event assert to first event
- 3042292 Wait longer for ping reaction
- 8c10f5f Fix xml block causing docs to fail
- 6e5768a Add a nox session for the Jenkins tornado jobs
- 43321e8 Add a cloud nox session for both runtests and pytest
- b6b4e95 Lint cleanup
- 0670614 Add tests for wraps
- 1670b5d Use functools.wraps with decorators
- 5ae263c Update static requirements for 2018.3 branch
- 3482c4d Disable code coverage uploads.
- 0e5bc67 Lint fixes
- 337c737 Update CI jobs to point to 2018.3
- 72e8603 Skip test if required cypto libs are not available
- e6bc9f8 Default to Cryptodome, fallback to PyCrypto
- b2b0764 Be aware of the different path separators
- 7e79b18 We need to make sure the virtualenv path entry is removed when searching
- ff6d3c6 We don't need to be root to run these tests
- 5965ab4 Disable progress bars on pip installs
- f86a44e Use the real python executable path when running within a virtualenv
- 3d407e8 Add a helper to return the path to the real pytohn executable
- ea9d246 We must also provide *venv_bin* when running within a virtualenv
- 31e91b0 Fix pip tests, in particular under windows
- 493f493 Don't fail if pkg_resources is not importable
- a1eb400 Specify the SHA we need
- 0b01f21 Let's fully unload pip
- 0b6f848 Fix docker entries
- 870b899 Remove unused argument
- 1656cb4 Disable re-runing failed tests for now

- 6db4141 Also ignore multiprocessing coverage files
- 83dc97d Actually write the lint reports.
- 5f97270 Try to make sure some pip internal functions are always present in sys.modules after reloads
- 0090c55 Try and address the fact that some python packages are only provided by the distro
- 4a072fe These tests aren't destructive
- 5c7a956 Don't fail the build if the issue was sending a slack notification
- 361ef6e Don't try to change ownership on non existing paths
- 7b17352 More insight(context) on failure reports
- 93b6b95 Lock supervisor to 3.3.5, last kown working(passing tests) version
- 51114aa Watch out for too long shebang's
- d97fe82 The output is important to know why it failed.
- 9fef385 Strings and integers are not comparable under Python 3
- d308dce Be verbose in a CI context
- 8b7105e Fix failed tests re-run logic
- d6d8db1 Blacklist *enum34* on Py>=3.4. Update requirements.
- 1cd00c0 Update PR CI jobs for nox
- 19a83d5 Blacklist pycrypto and install pycryptodome instead
- 91376b3 Ignore the generated docs archive
- 207b83a Update compiled requirements
- 535e7a7 Add TCP transport nox sessions
- 680b3e5 Compile requirements for the TCP transport
- 49a1ebe Throw error when running either on Py<2.7 or Py>=3.7
- 10528b9 Don't repeat filed tests on missing/empty failed test files file
- 4886809 We currently only support running lint under Py2.7
- a76981b Remove dead code
- 7b2c1ce Don't require six
- a27ed35 Add nox env to build docs
- e1b3f1d Lock docs python requirements
- c8de644 Fix lint issues under tests
- 0372718 Fix lint issues on salt
- 9eab9f4 Add nox session/env/target to run lint against Salt and it's test suite
- 123f771 Lock lint requirements
- 8df33ad Back to *+apache-libcloud==1.0.0* and skip windows for now
- 43e3ae6 Bump apache-libcloud requirement because 1.0.0 fails to install on windows
- d5f7813 Remove requiremens for platforms which aren't tested under Py3
- 8f02ee9 Recompile requirements

  - 66626f2 Remove pycryptodome from compiled requirements files
  - 75693e5 Use pycryptodomex on windows and PyCrypto on the rest
  - 4a9c19c Update(fix) requirements
  - 33a3467 We don't test OSX nor Windows under Py3.4
  - fd44fce We want a more verbose pip-compile output
  - c2ae2d5 We now compile requirements for each of the supported minor version
  - 6ee1260 Static requirements are now placed on *py<major>.<minor>* subdirectories
  - 3504804 The required crypto library is pulled in from zeromq.txt or raet.txt
  - 3ffe03c PyZMQ is pulled in from zeromq.txt
  - 7c2fe16 Instead of quoting session parameters, proxy nox sessions
  - d3ae77b Add separate crypto sessions
  - 0399620 Parametrize the transport
  - d33fd03 Update/Generate static requirements to new layout
  - 52ac533 Add pre-commit config to generate static requirements
  - 3cfcb13 Be aware of the new static requirements layout
  - dc7e16a Don't lock the docker requirement. It's not locked on the other platforms
  - 5476ba2 Kubernetes 3.0.0 does include the requirements files. pip-compile chokes on that
  - 79a8261 Previously generated requirements were py2 only
  - 1e29666 IOFLO is pulled in from the raet requirements
  - 4945e35 Fix pylint on 2017.7.9
  - 036bf68 Rerun failed tests
  - 03a9601 Ignore *.nox* directory
  - 50cb630 We have long files
  - db2392f Update CodeClimate settings
  - 899c80e Fix integration.client.test_kwarg test timeouts
- **ISSUE** #52836: (Ch3LL) unit.utils.test_args.ArgsTestCase.test_argspec_report failing (refs: #52852)
- **PR** #52852: (garethgreenaway) [2019.2.1] Fixes to unit.utils.test_args.test_argspec_report @ *2019-05-07 19:25:48 UTC*
  - 77355e2 Merge pull request #52852 from garethgreenaway/52836_test_argspec_report_failing
  - eee5a38 Merge branch '2019.2.1' into 52836_test_argspec_report_failing
- **PR** #52845: (Ch3LL) Backport #52780 into 2019.2.1 @ *2019-05-07 16:27:21 UTC*
  - **PR** #52780: (sbrennan4) Handle ipv6 in _netlink_tool_remote_on (refs: #52845)
  - dcc5fde Merge pull request #52845 from Ch3LL/bp-52780
  - dfff09b Merge branch '2019.2.1' into bp-52780
- **PR** #52851: (Ch3LL) Reload matcher loader when ext_pillar_first set @ *2019-05-07 13:12:07 UTC*
  - 2e4f296 Merge pull request #52851 from Ch3LL/matcher_ext_pillar

  - bdcae5d Add docs for new reload kwarg
  - fc70884 Reload matcher loader when ext_pillar_first set
  - 8ac74e2 Merge branch '2019.2.1' into bp-52780
  - 226c7ce Merge branch '2019.2.1' into bp-52780
  - c1eae84 Add test
  - 69fe65b Use rsplit
    - 8fdc48c Removing create_autospec and having salt.utils.args.argspec_report run against the _test_spec function directly. Depending on the python version, create_autospec gives different results and cause the test to fail. The test is now more accurate at testing the arguments for the function.
- **PR** #52842: (s0undt3ch) *libcrypto.OpenSSL_version_num* might be a callable @ *2019-05-06 17:00:37 UTC*
  - cc7a0d1 Merge pull request #52842 from s0undt3ch/2019.2.1
  - a944017 *libcrypto.OpenSSL_version_num* might be a callable
- **PR** #52839: (twangboy) Add support for EC2 @ *2019-05-04 17:28:54 UTC*
  - b625a6a Merge pull request #52839 from twangboy/fix_test_grain
  - 1c9b372 Add support for EC2
- **ISSUE** #51982: (arsiesys) topic (minion_id) need to be encoded before being hash and sent using zeroMQ (refs: #51983)
- **PR** #52767: (Ch3LL) [2019.2.1] Add tests to PR #51983 @ *2019-05-03 08:35:27 UTC*
  - **PR** #51983: (arsiesys) encode topic before using hashlib (refs: #52767)
  - 6002939 Merge pull request #52767 from Ch3LL/zmq_filter_test
  - 1ee8f37 Merge branch '2019.2.1' into zmq_filter_test
  - d74b6ee Merge branch '2019.2.1' into zmq_filter_test
  - 3210274 Fix linter warnings
  - e7c9d6e increase timeout when gathering results
  - 784786d Add zmq_filter unit tests
  - 805c5ee encode topic before using hashlib
- **PR** #52827: (twangboy) Add pymssql dependency for Windows builds @ *2019-05-03 08:21:40 UTC*
  - 5996932 Merge pull request #52827 from twangboy/add_pymssql
  - 6f0b8e2 Add pymssql dependency for Windows builds
- **ISSUE** #52717: (Ch3LL) debian9 tests segmentation fault (refs: #52797)
- **PR** #52797: (garethgreenaway) [2019.2.1] Only run the libcrypto init if less than OpenSSL 1.1.0 @ *2019-05-03 05:12:31 UTC*
  - e3918c5 Merge pull request #52797 from garethgreenaway/52717_debian_tests_segmentation_fault
  - 7ab0961 Merge branch '52717_debian_tests_segmentation_fault' of github.com:garethgreenaway/salt into 52717_debian_tests_segmentation_fault
    - f8d816e Merge branch '2019.2.1' into 52717_debian_tests_segmentation_fault
- **PR** #52771: (twangboy) Fix *test_gpg* tests on Windows @ *2019-05-02 13:30:18 UTC*

  - b68544f Merge pull request #52771 from twangboy/fix_test_gpg
  - c7f45c2 Pass encoding to stringutils.to_unicode
- **PR** #52783: (Ch3LL) Fix boto_apigateway tests for PyYAML 5.1 @ *2019-05-02 13:26:34 UTC*
  - 316d05b Merge pull request #52783 from Ch3LL/fix_boto_yaml
  - cde29cb Fix boto_apigateway tests for PyYAML 5.1
  - 6ffbca5 Fixing lint.
  - e24f2f8 Only run the libcrypto init if less than OpenSSL 1.1.0.
- **PR** #52769: (twangboy) Remove domain grain from tests for Windows @ *2019-04-30 21:14:26 UTC*
  - 3509465 Merge pull request #52769 from twangboy/fix_test_core
  - 0049f85 Remove domain grain from tests
- **PR** #52768: (twangboy) Remove pchanges… again… @ *2019-04-30 20:24:27 UTC*
  - 30dc14b Merge pull request #52768 from twangboy/fix_test_win_dacl
  - 123b607 Remove pchanges… again…
- **PR** #52753: (twangboy) Skip test_gen_thin_compression_fallback_py3 @ *2019-04-30 03:36:27 UTC*
  - 3e46f51 Merge pull request #52753 from twangboy/skip_test_gen_thin
  - 811b381 Skip test_gen_thin_compression_fallback_py3
- **ISSUE** #52721: (Ch3LL) unit.cloud.clouds.test_ec2 test failing (refs: #52736)
- **PR** #52736: (dwoz) Skip password decryption test when no libraries available @ *2019-04-29 17:07:56 UTC*
  - 6bfca94 Merge pull request #52736 from dwoz/ec2test
  - 04ad027 Merge branch '2019.2.1' into ec2test
- **PR** #52624: (tanlingyun2005) fix TypeError: argument of type int is not iterable @ *2019-04-29 13:15:42 UTC*
  - 9a1ed78 Merge pull request #52624 from tanlingyun2005/2019.2.1
  - 2c6867e add test case to tests/unit/cli/test_batch.py
  - cb7742e Merge branch '2019.2.1' into 2019.2.1
  - c943900 Merge branch '2019.2.1' into 2019.2.1
  - d4abddd fix TypeError: argument of type int is not iterable
  - ed908b3 Merge branch '2019.2.1' into 2019.2.1
  - 463b60e Merge branch '2019.2.1' into 2019.2.1
  - 65ee219 Merge branch '2019.2.1' of https://github.com/tanlingyun2005/salt into 2019.2.1
    - f71168d Merge branch '2019.2.1' into 2019.2.1
  - 509c797 bug fix salt/cli/batch.py
  - 211915c fix salt/cli/batch.py TypeError: argument of type int is not iterable
  - 070ae84 fix TypeError: argument of type int is not iterable
    - b770f96 Skip test requiring crypto when none available
    - e52b390 Warn when password decrypton requested but not possible
- **PR** #52696: (Ch3LL) Backport #50087 into 2019.2.1 @ *2019-04-25 20:33:13 UTC*

  - **PR** #50087: (rbthomp) Fix applying of attributes for returner rawfile_json (refs: #52696)
  - 806307a Merge pull request #52696 from Ch3LL/bp-50087
  - 66e97e7 Fix applying of attributes for returner rawfile_json
- **PR** #52659: (twangboy) Fix issues with the win_file tests @ *2019-04-22 23:34:36 UTC*
  - 174f558 Merge pull request #52659 from twangboy/fix_test_win_file
  - 915c780 Fix issues with the win_file tests
- **PR** #52655: (dwoz) Parse chattr version and fix test case @ *2019-04-22 01:46:29 UTC*
  - d1a61a6 Merge pull request #52655 from dwoz/cron_test_fix
  - e69fcc5 Fix typo
  - 859d088 Merge remote-tracking branch 'origin/cron_test_fix' into cron_test_fix
    - 7f7bb90 Merge branch '2019.2.1' into cron_test_fix
- **PR** #52601: (Ch3LL) Cherry-Pick #52415 into 2019.2.1 @ *2019-04-21 19:33:03 UTC*
  - **PR** #52415: (Ch3LL) Backport #49832 into 2018.3 (refs: #52601)
  - **PR** #49832: (terminalmage) Replace pchanges with changes to make onchanges/prereq work in test mode (refs: #52415)
  - f7d823c Merge pull request #52601 from Ch3LL/cp-52415
  - ecd6802 Merge branch '2019.2.1' into cp-52415
  - 538d5cf Remove pchanges from win_dacl
  - 81b865f use same newfile message on linux for windows file state
  - 93bdd08 Update dict correctly in file state
  - f3c7f27 remove pchanges for windows file modules
  - fb3b75d Fix tests for pchanges backport into 2018.3
  - 0f296bb Update kernelpkg test to reflect pchanges removal
  - 885d6ff Update file.touch unit tests to reflect addition of changes in test mode
  - 4bc5fd0 Add integration tests for test mode onchanges/prereq
  - ed214c4 Make it possible to use prereq with test and saltmod state mods
  - ceb3f4d Add repack_state_returns to TestCase
  - 982e693 Add exception logging in flaky decorator
  - 0ba0ddf Add test mode changes to file.touch state
  - 6db2beb Replace "pchanges" with "changes" to fix onchanges/prereq requisites
    - 169c2d5 Chattr version py3 fix
    - cf88c27 Fix linter issues
    - 0ea007d Parse chattr version and fix test case
- **ISSUE** #52508: (Ch3LL) integration.modules.test_publish test failures (refs: #52645)
- **PR** #52645: (dwoz) Fix salt.modules.publish and salt.states.x509 tests @ *2019-04-20 08:17:19 UTC*
  - af4f204 Merge pull request #52645 from dwoz/test_fixes

  - 9b6f9f9 Clean up cruft
  - c3971a5 Fix salt.modules.publish and salt.states.x509 tests
- **PR** #52629: (dwoz) Fix event assertion race condition @ *2019-04-19 19:03:20 UTC*
  - **PR** #52540: (dwoz) Fix race condition in event assertion (refs: #52629)
  - 0a4d2d9 Merge pull request #52629 from dwoz/ping_wait_2019.2.1
  - c001022 Fix linter
  - 423f6f9 Fix vent assertion race condition
- **PR** #52619: (dwoz) Remove unused method from pytest engine @ *2019-04-18 23:45:35 UTC*
  - 4859e6c Merge pull request #52619 from dwoz/pytest_engine_2019.2.1
  - 105784a Remove unused method from pytest engine
- **PR** #52614: (Ch3LL) [2019.2.1] Update integration.modules.test_network.NetworkTest.test_network_ping… @ *2019-04-18 21:54:28 UTC*
  - 77943e5 Merge pull request #52614 from Ch3LL/cp-test_network-2019.2.1
  - 27c79d2 Update integration.modules.test_network.NetworkTest.test_network_ping test address
- **ISSUE** #32245: (tkwilliams) git_pillar should permit automatic mapping of environment to branch a-la gitfs (refs: #50768)
- **PR** #52615: (twangboy) Bring 51661 into 2019.2.1 @ *2019-04-18 21:53:35 UTC*
  - **PR** #51661: (sathieu) git_pillar: Fix all_saltenvs on base env (refs: #52615)
  - **PR** #51597: (sathieu) git_pillar: Fix all_saltenvs on base env (refs: #51661)
  - **PR** #50768: (sathieu) git_pillar: Add support for all_saltenvs parameter (refs: #51597, #51661)
  - c6b4986 Merge pull request #52615 from twangboy/gitfs_2019.2.1
  - d952674 Merge branch '2019.2.1' into gitfs_2019.2.1
- **ISSUE** #52547: (cbosdo) virt.purge throws a libvirt exception on Xen hypervisor (refs: #52548)
- **PR** #52548: (cbosdo) Virt purge fix @ *2019-04-18 20:07:36 UTC*
  - 8ee15a9 Merge pull request #52548 from cbosdo/virt-purge-fix
  - 6ca8f4b Try/except undefineFlags() as this operation is not supported on bhyve
    - 5a0464b Bring 51661 into 2019.2.1
- **PR** #52591: (Ch3LL) [2019.2.1] Update test_schema to mirror the new ValidationErrors in 3.0.0 @ *2019-04-18 16:34:22 UTC*
  - e03aed5 Merge pull request #52591 from Ch3LL/fix_jsonschema_2019.2
  - e09be842 Update test_schema to mirror the new ValidationErrors in 3.0.0
- **ISSUE** #52525: (lomeroe) deserializer_opts are not passed the serializer in file.serialize (refs: #52526)
- **PR** #52526: (lomeroe) Fix use of deserializer_opts in file.serialize @ *2019-04-17 14:08:52 UTC*
  - 21d6365 Merge pull request #52526 from lomeroe/deserializer_opts_fix
  - ea0520c update test for serialization opts
  - 84ff03a Correct attempting to use the serializer_name to pull deserializer_opts when deserializer_name is used as the key in the deserializer_options dict.

- **ISSUE** #49147: (furgerf) Log filled with "Exception occurred while Subscriber handling stream: Already reading" (refs: #51963)
- **PR** #52570: (DmitryKuzmenko) Simplify IPCClient and prevent corrupt messages (Turn 2) 2019.2 @ *2019-04-17 14:03:02 UTC*
  - **PR** #52564: (DmitryKuzmenko) Simplify IPCClient and prevent corrupt messages (Turn 2) (refs: #52570)
  - **PR** #52445: (dwoz) Simplify IPCClient and prevent corrupt messages (refs: #52564)
  - **PR** #51963: (DmitryKuzmenko) Allow multiple instances of IPCMessageSubscriber in one process (refs: #52564)
  - e522ffe Merge pull request #52570 from DSRCorporation/bugs/revert_51963_2019.2
  - fabbcac A couple of race conditions fixes and a test update.
  - 9c85734 Drop singleton from IPCClient
  - 7cf6d54 Revert "Support parallel work of multiple IPCMEssageSubscribers in one process"
  - 7f26e76 Revert "Update doc conf with the new import *tornado.queues*"
  - 69e9416 Revert "Minor: Fix typo in docstring"
- **ISSUE** #52449: (Ch3LL) integration.cloud.clouds.test_gce.GCETest.test_instance_extra failure (refs: #52551)
- **PR** #52551: (garethgreenaway) [2019.2] Marking the two tests in test_gce as flaky @ *2019-04-17 14:00:39 UTC*
  - b173406 Merge pull request #52551 from garethgreenaway/52449_test_gce_random_failures
  - 28af717 Marking the two tests in test_gce as flaky.
- **ISSUE** #51842: (mattLLVW) salt.auth.django docs use wrong model name (refs: #52553)
- **PR** #52553: (mattLLVW) fix: #51842 @ *2019-04-17 13:59:45 UTC*
  - 8352362 Merge pull request #52553 from mattLLVW/fix_django_auth_docs
  - 3baeedf fix: #51842
- **ISSUE** #52134: (sploenix) Regression: Include list of pillar files under different keys in 2019.2 (refs: #52490)
- **ISSUE** #22063: (jeanpralo) Wildcard inside top.sls file for pillar (refs: #52490)
- **PR** #52490: (dwoz) Fix pillar include regression @ *2019-04-17 13:54:39 UTC*
  - 9faa49c Merge pull request #52490 from dwoz/issue_52134
  - 876dd18 Fix merge wart
  - 2ed650b Merge remote-tracking branch 'saltstack/2019.2' into issue_52134
  - b0af11f Run include tests when pillar source changes
  - 307cec6 Fix pillar tests
  - 1a33bde import __future__.absolute_import
  - 5dc5de9 Use file encoding and add docstring
  - e84cd41 Fix pillar include wart
  - c79f496 Add pillar include tests
  - 7745242 Revert "Fix #22063: pillar wildcard support include"
- **PR** #52552: (twangboy) Bring 52170 into 2019.2 @ *2019-04-17 13:54:17 UTC*
  - **PR** #52170: (twangboy) Fix issue when task doesn't have delay_random defined (refs: #52552)

  - b0c0237 Merge pull request #52552 from twangboy/fix_win_task_2019.2
  - fd19cca Remove some warts in the docs
  - deb0b10 Fix some lint
  - a279d45 Final doc fixes
  - 58f0cd2 Fix docs... attempt 6
  - a66716e Fix docs... attempt 5
  - f6f3381 Fix docs... attempt 4
  - 18cfc15 Fix docs... attempt 3
  - c691e0d More doc fixes
  - e01077b Fix docs issue... maybe...
  - b649495 Honor 80 character line limit
  - eba4507 Bring 52170 into 2019.2
- **PR** #52528: (Ch3LL) [2019.2] Merge forward from 2018.3 to 2019.2 @ *2019-04-17 13:53:46 UTC*
  - cc0cd5d Merge pull request #52528 from Ch3LL/merge-2019.2
  - 87cf385 increase timeout on test_state for windows
  - 3442202 Increase timeout for test_kwarg tests
  - 2c3ff2f Merge branch '2019.2' into merge-2019.2
- **PR** #52534: (twangboy) Add TLS support up to TLS 1.2 @ *2019-04-13 22:56:41 UTC*
  - e91240b Merge pull request #52534 from twangboy/tls_support_2019.2
  - 9ec54c4 Add TLS support up to TLS 1.2
    - 474efa1 Fix pylint and state test failure
    - 048a82e Merge branch '2019.2' into merge-2019.2
- **PR** #52536: (twangboy) Bring #52191 into 2019.2 branch @ *2019-04-12 21:31:23 UTC*
  - **PR** #52191: (twangboy) Fix issue where version doesn't detect when check_remote=True (refs: #52536)
  - 2c42e4d Merge pull request #52536 from twangboy/fix_choco_2019.2
  - b394ad5 Bring #52191 into 2019.2 branch
  - 795a7f6 Fix windows tests - increase timeout
  - c70772c Fix linux state tests- add new comment
  - 2941567 Merge branch '2018.3' into '2019.2'
  - 262516c Merge pull request #52209 from twangboy/fix_choco_retcode
    - 7aec2d3 Merge branch '2018.3' into fix_choco_retcode
    - bfdc47b Handle new enhanced retcode 2
  - dd7a4ba Merge pull request #52046 from twangboy/use_reg_timezone
    - fc8e970 Merge branch '2018.3' into use_reg_timezone
    - d0987e6 Merge branch '2018.3' into use_reg_timezone
    - 1bf0cbc Merge branch '2018.3' into use_reg_timezone

    - b466f8a Merge branch '2018.3' into use_reg_timezone
    - 34bdae5 Merge branch '2018.3' into use_reg_timezone
    - 9a8afa6 Fix tests, add null byte test
    - 59b1d4f Revert back to using reg to get timezone
  - 0d09608 Merge pull request #52396 from dmurphy18/fix_aix_ssh
    - 8bfa340 Correct typo
    - 6d98577 Altered code to support salt-ssh on AIX
  - 576478e Merge pull request #52473 from Ch3LL/bp-49437
    - 781385f Merge branch '2018.3' into bp-49437
    - 3a3114d Fix merge conflict wart
    - bbfd412 Merge branch '2018.3' into bp-49437
    - 48ee82d Merge branch '2018.3' into bp-49437
    - f66d474 Add support to avoid calling refresh_db in opkg.del_repo
  - d830bc6 Merge pull request #52493 from dwoz/ping_reaction_test
    - 8b05e77 Merge branch '2018.3' into ping_reaction_test
    - 7083d21 Merge branch '2018.3' into ping_reaction_test
    - 7c963c1 Merge branch '2018.3' into ping_reaction_test
    - 75ddeb3 fix linter
    - e3e1d2e fix reactor ping test
  - 9b7a0d1 Merge pull request #52505 from bloomberg/2018_zmq_uri
    - b682ec5 transport.zmq: fix bug introduced by b7df7e75cf2
  - 452bbeb Merge pull request #52512 from dwoz/supervisord_2018.3
    - cd5cd7d The exit code matters.
  - 53228ae Merge pull request #52475 from Ch3LL/bp-49464
    - a891fd3 Make opkg.del_repo be compatible with pkrepo state module
  - 6e79e24 Merge pull request #52500 from terminalmage/invalid-archive-source
    - e9d175b Fix traceback when invalid source passed to archive.extracted
  - 2df74ca Merge pull request #52476 from bloomberg/win_cmd_2018
    - 0e0c42e 2018 backport: modules.cmdmod: handle windows environ better
  - 4dfb2f8 Merge pull request #52474 from Ch3LL/bp-49984
    - c8d547e Use prune option in Pygit2 provider when fetching
  - d699485 Merge pull request #52393 from waynew/52087-mine-delete-test-enhancements
    - e3f5be3 Use uncommon item in mine delete test
  - 89bd258 Merge pull request #52445 from dwoz/ipc_fixes_2018.3
    - 1bdaf29 Ensure exceptions in service future are handled
    - c7ad732 Use six.reraise for py3 compatability

    - 29999b0 Close message service on subscriber close
    - 47203f7 Fix ipc unit tests
    - 2b35437 Fix linter issues
    - 503cdd2 Remove IPCClient singleton
    - b374034 Re-raise queued exceptions with traceback

- **ISSUE** #51879: (whytewolf) 2019.2.0 binary pillar unicode error returns. (refs: #52334)
- **PR** #52334: (waynew) 51879 fix binary pillar return error @ *2019-04-12 03:45:17 UTC*
  - 6eb2bce Merge pull request #52334 from waynew/51879-fix-binary-pillar-return-error
  - ead856e Merge branch '2019.2' into 51879-fix-binary-pillar-return-error
  - 2aa971d lint cleanup
  - 28c2945 keep_pillar not needed
  - fb010c0 Add binary pillar to the docs
  - f2aebf9 Assume file contents are binary
  - 2b8c782 Test gpg render with replace newlines
  - c4b385b Allow binary pillar data
  - 604b671 Return binary data from gpg renderer
  - 32aafab Ignore vscode and fix swap ignore
- **ISSUE** #52041: (arsiesys) 2019.2 influxdb_retention_policy.present broken when used extra arguments (refs: #52043)
- **PR** #52043: (arsiesys) fix missing client_args in influxdb module @ *2019-04-12 03:43:03 UTC*
  - 2d59705 Merge pull request #52043 from arsiesys/fix_influxdb_state_rp
  - e617c5b Merge branch '2019.2' into fix_influxdb_state_rp
  - 7661860 fix missing client_args in influxdb module
- **PR** #52146: (twangboy) Add missing symlink test @ *2019-04-12 00:51:16 UTC*
  - 1b969c3 Merge pull request #52146 from twangboy/fix_test_symlink_2019.2
  - 8e1c882 Merge branch '2019.2' into fix_test_symlink_2019.2
  - a429542 Merge branch '2019.2' into fix_test_symlink_2019.2
  - 401c253 Merge branch '2019.2' into fix_test_symlink_2019.2
  - cce6200 Fixing lint
  - c7cb009 Merge branch '2019.2' into fix_test_symlink_2019.2
  - 3eae05a Merge branch '2019.2' into fix_test_symlink_2019.2
  - 83bed46 Add missing symlink test
- **PR** #52341: (cbosdo) virt.pool_running: fix pool start @ *2019-04-12 00:14:52 UTC*
  - 7a1b8ca Merge pull request #52341 from cbosdo/virt-state-fixes
  - 89f0bd8 Merge branch '2019.2' into virt-state-fixes
  - 25b9681 virt.pool_running: fix pool start

  - 30981d2 Remove unneeded kwargs in virt states
  - 0871c02 virt: allow defining the VM type and arch when creating it
  - 1c65d25 Updating running domains in virt.running
  - 13d7819 Add missing virt states unit tests
  - 0681d86 virt.running support for all virt.init options
  - c6a444b Add test case for virt.running
  - 2db7a98 Let virt running state provide errors
- **ISSUE** #52350: (Ch3LL) Re-Add Python 2 unicode string literals in YAML renderer (refs: #52427)
- **PR** #52427: (garethgreenaway) [2019.2] Support for old yaml render @ *2019-04-12 00:06:37 UTC*
  - 6040282 Merge pull request #52427 from garethgreenaway/52350_readd_and_gate_unicode_string_literal_support
  - 82f010a Merge branch '2019.2' into 52350_readd_and_gate_unicode_string_literal_support
  - ae81fee Merge branch '2019.2' into 52350_readd_and_gate_unicode_string_literal_support
  - b618d44 Merge branch '2019.2' into 52350_readd_and_gate_unicode_string_literal_support
  - b191bc0 Merge branch '2019.2' into 52350_readd_and_gate_unicode_string_literal_support
  - 38a9818 Merge branch '2019.2' into 52350_readd_and_gate_unicode_string_literal_support
  - 2767984 Removing logging entry
  - 71cd303 Adding missing yamlloader_old.py
  - f848aa4 Adding support back in with, a gated configuration option, for the old YAML Loader.
- **ISSUE** #51865: (magenbrot) salt.match.search_by resulting in 'None' since 2019.2 (refs: #52234)
- **PR** #52234: (garethgreenaway) [2019.2] Fixes to matchers when used in pillar @ *2019-04-12 00:05:48 UTC*
  - 05ba7c5 Merge pull request #52234 from garethgreenaway/51865_match_search_by_master
  - 1760245 Merge branch '2019.2' into 51865_match_search_by_master
  - d50f011 Merge branch '2019.2' into 51865_match_search_by_master
  - b0cbb60 Merge branch '2019.2' into 51865_match_search_by_master
  - 799a08b Merge branch '2019.2' into 51865_match_search_by_master
  - 21891d6 Merge branch '2019.2' into 51865_match_search_by_master
  - a2173d7 Fixing a typo where attempting to get a list from __opts__ instead of an individual item
  - 6a5b5b2 swapping out if…else approach for __opts__.get approach.
  - aa2c626 With the change/addition of the matcher subsystem in 2019.2, the match.search_by when used in pillar broke when targetting the minion that was also the salt master. This was caused by the id in __opts__ being used in all cases. This change updates the glob_match function to use the preserved minion_id of the master if it is available so that targeting works as expected.
- **PR** #52160: (cbosdo) Virt whitespace fix @ *2019-04-12 00:04:25 UTC*
  - 80d7aea Merge pull request #52160 from cbosdo/virt-whitespace-fix
  - fd1f539 Merge branch '2019.2' into virt-whitespace-fix
  - b297e7d virt: handle whitespaces in VM names
- **ISSUE** #51832: (nocturo) pillar include stopped working (refs: #52008)

- **PR** #52008: (waynew) Allow / in pillar includes @ *2019-04-12 00:02:40 UTC*
  - b551bbd Merge pull request #52008 from waynew/51832-re-allow-slash-includes
  - ee3115f Allow leading dots and / as pillar separators
  - 29c676d Ensure _closing exists
  - eb517e1 Swap '/' for '.' when matching
- **ISSUE** #51821: (OrangeDog) 'comment' is an invalid keyword argument for 'test.fail_without_changes' (refs: #51937)
- **PR** #51937: (garethgreenaway) [2019.2] Fix to test state functions to allow arguments that mod_watch uses @ *2019-04-11 23:57:44 UTC*
  - d3159cb Merge pull request #51937 from garethgreenaway/51821_ensure_result_comment_changes_valid_arguments
  - 86c63b5 Merge branch '2019.2' into 51821_ensure_result_comment_changes_valid_arguments
  - 0966d61 Merge branch '2019.2' into 51821_ensure_result_comment_changes_valid_arguments
  - 5527dff Ensure the comment, changes, and result are valid arguments for various test state functions as they are valid arguments for mod_watch.
- **ISSUE** #51818: (syphernl) salt.nacl.dec: expected str, bytearray, or unicode (refs: #51913)
- **PR** #51913: (garethgreenaway) [2019.2] Fixes to utils/nacl.py @ *2019-04-11 23:56:57 UTC*
  - 7be0428 Merge pull request #51913 from garethgreenaway/51818_fix_when_sk_is_empty
  - 55d7488 Merge branch '2019.2' into 51818_fix_when_sk_is_empty
  - b2128e8 Merge branch '2019.2' into 51818_fix_when_sk_is_empty
  - cb01a1b Merge branch '2019.2' into 51818_fix_when_sk_is_empty
  - 1a1e333 Merge branch '51818_fix_when_sk_is_empty' of github.com:garethgreenaway/salt into 51818_fix_when_sk_is_empty
    - 102f0a3 Merge branch '2019.2' into 51818_fix_when_sk_is_empty
    - 7fd04ec Merge branch '2019.2' into 51818_fix_when_sk_is_empty
  - dd23069 Adding tests for NACL changes.
  - c314f51 Correctly handle the situation when with the secret key or public key values are empty.
- **ISSUE** #52116: (lordcirth) host.present always returns None when in test=True (refs: #52399)
- **PR** #52399: (waynew) Bring host.present in line with Salt's test conventions @ *2019-04-11 23:32:28 UTC*
  - d6f0971 Merge pull request #52399 from waynew/52116-fix-host-present-fail
  - 8440ae1 Merge branch '2019.2' into 52116-fix-host-present-fail
  - dd002fd Fix linter
  - 9d1b2a0 Only return None on host.present changes when test
- **PR** #52477: (mattp-) 2019.2 backport #52472 modules.cmdmod: handle windows environ better @ *2019-04-11 23:13:34 UTC*
  - **PR** #52472: (mattp-) modules.cmdmod: handle windows environ better (refs: #52476, #52477)
  - 7c709cf Merge pull request #52477 from bloomberg/win_cmd_2019
  - 736a437 2019.2 bacpkport: modules.cmdmod: handle windows environ better
- **ISSUE** #49661: (ephreal) Issues starting salt-master or salt-minion with openssl-1.1.1-1 (refs: #51655)

- **PR** #52423: (bbinet) Remove unused salt.crypt import @ *2019-04-11 21:44:41 UTC*
  - **PR** #51655: (garethgreenaway) [2017.7] Removing unused salt.crypt imports (refs: #52423)
  - b0d3c76 Merge pull request #52423 from bbinet/unused-salt-crypt
  - dc330e7 Merge branch '2019.2' into unused-salt-crypt
  - 4b26dea Merge branch '2019.2' into unused-salt-crypt
  - 9607c38 Also remove unused salt.crypt import in pillar/__init__.py
  - 49f2631 Remove unused salt.crypt import
- **PR** #52506: (mattp-) 2019.2: transport.zmq: fix bug introduced by b7df7e75cf2 @ *2019-04-11 21:12:51 UTC*
  - 4bb9185 Merge pull request #52506 from bloomberg/2019_zmq_uri
  - 1e9dd7f transport.zmq: fix bug introduced by b7df7e75cf2
- **ISSUE** #52462: (Ch3LL) integration.states.test_supervisord test failing (refs: #52513, #52512)
- **PR** #52513: (dwoz) [2019.2] Fix supervisord @ *2019-04-11 21:09:17 UTC*
  - 1532642 Merge pull request #52513 from dwoz/supervisord_2019.2
  - dc9ace6 The exit code matters.
- **PR** #52496: (dwoz) Fix wart in IPC merge forward @ *2019-04-11 05:19:02 UTC*
  - **PR** #52482: (dwoz) Ipc fixes 2019.2 (refs: #52496)
  - 978084d Merge pull request #52496 from dwoz/ipc_wart
  - 9805f38 Fix wart in IPC merge forward
- **PR** #52482: (dwoz) Ipc fixes 2019.2 (refs: #52496) @ *2019-04-10 19:53:58 UTC*
  - 791b646 Merge pull request #52482 from dwoz/ipc_fixes_2019.2
  - f45d29b Remove un-needed test
  - 0141b7f Fix merge wart
  - 82150f0 Remove un-used import
  - c9ec8b1 Ensure exceptions in service future are handled
  - 25f5a90 Use six.reraise for py3 compatability
  - 6d80789 Fix ipc unit tests
  - c80da32 Fix linter issues
  - 024b4b6 Remove IPCClient singleton
  - f1d0e02 Re-raise queued exceptions with traceback
- **ISSUE** #51932: (tomlaredo) [REGRESSION] –output-diff doesn't display changes anymore with test=True since v2019.2.0 (refs: #51992)
- **PR** #51992: (garethgreenaway) [2019.2] Swapping pchanges for changes in file state. @ *2019-04-09 19:38:57 UTC*
  - 3173673 Merge pull request #51992 from garethgreenaway/51932_show_diff_when_test_True
  - 7c4abd6 changes needs to be a dictionary.
  - ac950b3 Merge branch '2019.2' into 51932_show_diff_when_test_True
- **PR** #52413: (Ch3LL) [2019.2] Merge forward from 2018.3 to 2019.2 @ *2019-04-08 16:19:49 UTC*

  - bd3ad7b Merge pull request #52413 from Ch3LL/merge-2019.2
  - 30783f4 Merge branch '2019.2' into merge-2019.2
- **PR** #52151: (simonflood) Fix typo in actual 2019.2.0 release notes @ *2019-04-08 13:36:28 UTC*
  - **PR** #51861: (simonflood) Fix typo in 2019.2.0 release notes (refs: #52151)
  - a18abf2 Merge pull request #52151 from simonflood/patch-2
  - a4a5521 Merge branch '2019.2' into patch-2
  - 3d31102 Fix typo in actual 2019.2.0 release notes
    - 16733da Merge branch '2019.2' into merge-2019.2
    - fe13214 Merge branch '2018.3' into '2019.2'
      - b6028b9 Merge pull request #52403 from bdrung/fix-test-conditions
      - e74f78f Skip ExtendTestCase if templates directory is missing
      - 0473683 Skip SampleConfTest if sample conf directories are missing
      - e5a755d Merge pull request #52250 from twangboy/fix_grains
      - a77ec81 Merge branch '2018.3' into fix_grains
      - b405391 Refer to the windowsdomain grain
      - e103561 Add some documentation about the domain grain on Windows
      - c874831 Don't set the domain grain to windowsdomain
      - 65eb461 Add a test for windows grains
      - 173d1e7 Fix the domain grain on Windows
      - 0ce0c04 Merge branch '2019.2' into 51932_show_diff_when_test_True
- **ISSUE** #51869: (eimantaszd) x509.sign_remote_certificate not working after upgrade to 2019.2.0 (refs: #52381)
- **PR** #52381: (dwoz) Fix issue #51869 and add cert signing test @ *2019-04-05 20:36:11 UTC*
  - 6bae227 Merge pull request #52381 from dwoz/issue_51869
  - cd78485 Merge branch '2019.2' into issue_51869
- **ISSUE** #21927: (jfindlay) network.traceroute intermittently stacktraces (refs: #52370)
- **PR** #52370: (jfindlay) Fix network.traceroute exec module function @ *2019-04-03 16:39:20 UTC*
  - c1063cf Merge pull request #52370 from jfindlay/traceroute
  - 2f49231 modules.network: update unit test
  - 5b64fc0 modules.network: log and skip problematic traceroute lines
  - e33f5c1 modules.network: standardize util check
- **PR** #52397: (garethgreenaway) [2019.2] Merge forward from 2018.3 to 2019.2 @ *2019-04-02 22:12:45 UTC*
  - 3675f1e Merge pull request #52397 from garethgreenaway/merge-2019.2
  - 28e76b1 Merge branch '2018.3' into merge-2019.2
  - 28f947e Merge pull request #52213 from garethgreenaway/beacon_state_dont_save_test_true
    - 7f83b4b Swapping out if state looking for test in opts for a __opts__.get
    - e1cf599 Merge branch '2018.3' into beacon_state_dont_save_test_true

    - b751122 Merge branch '2018.3' into beacon_state_dont_save_test_true
    - a901ec4 Don't save beacons when test=True
  - 10d041b Merge pull request #52344 from twangboy/os_release_embedded
    - 74ab4d3 Use old way to get osrelease if new way fails
  - e88d3ba Merge pull request #52345 from garethgreenaway/52197_incron_cannot_have_comments
    - 070837a Merge branch '2018.3' into 52197_incron_cannot_have_comments
    - 8461608 Updating the incron module, state module and tests to remove use of comments.
      - 855f31a Remove un-used file
      - affd9b8 Fix setup/teardown methods
      - 215bf93 Refresh after modifying tmp pillars
      - 95c3aba Add config for listener
      - 5d231f4 remove unused import
      - 8c1b1db Use tmp pillar for signing policies
      - a0e2458 fix unused import
      - 7f5f7dd Fix requisite path
      - 5630498 Fix signing policies path
      - 0bd0826 Fix issue #51869 and add cert signing test
- **PR** #52347: (Ch3LL) [2019.2] Merge forward from 2018.3 to 2019.2 @ *2019-03-28 16:41:53 UTC*
  - 27e6cbf Merge pull request #52347 from Ch3LL/merge-2019.2
  - 63962b5 Merge branch '2018.3' into '2019.2'
  - bda3ca2 Merge pull request #51749 from terminal-labs/salt-cloud-map-alt-update
    - 9ca0f72 Changed where to mention the map files compatibility for minion configuration options. This is a result of @Ch3LL suggestion on the PR itself for this doc change.
    - 364ef06 Added mention of map file alternative for minion configuration options.
  - a743fff Merge pull request #52113 from twangboy/fix_lgpo
    - bb4a704 Merge branch '2018.3' into fix_lgpo
    - 7186405 Merge branch '2018.3' into fix_lgpo
    - 7656331 Change default language from 1033 to en_US
  - 7040643 Merge pull request #52172 from garethgreenaway/51959_fix_acl_present_output
    - 8c53890 Merge branch '2018.3' into 51959_fix_acl_present_output
    - 0bdde94 Merge branch '2018.3' into 51959_fix_acl_present_output
    - c5337e1 Merge branch '2018.3' into 51959_fix_acl_present_output
    - 889660f Fixing lint.
    - bfdb669 Updating the reverse octal lookup dictionary. Updating tests.
    - 24c907b Adding additional permissions to the lookup.
    - 4b99afa Fixing the output when there are changes for the ACL state so the permissions are shown and not the octal number.

- 45d6cad Merge pull request #52283 from dmurphy18/fix_cpe_name_amzn2
  - cbe5f7d Update to allow for malformed CPE_NAME from some OS's
- 14a6b9d Merge pull request #52306 from terminal-labs/replace-ping-with-version
  - c04ba6e Changed a text artifact from the ping function to supporting text for the version function.
  - 0b1eb00 Replaced *test.ping* with *test.version* Result of issue #52277
- 7b2b521 Merge pull request #51963 from DSRCorporation/bugs/49147_ipc_subscriber
  - 37aeba3 Minor: Fix typo in docstring
  - cc46e9c Merge branch '2018.3' into bugs/49147_ipc_subscriber
  - 23733bd Regression test for parallel IPCMessageSubscriber support
  - 2bc2a7d Merge branch '2018.3' into bugs/49147_ipc_subscriber
  - 684bf58 Update doc conf with the new import *tornado.queues*
  - 008cf49 Merge branch '2018.3' into bugs/49147_ipc_subscriber
  - 01e9a3d Merge branch '2018.3' into bugs/49147_ipc_subscriber
  - 710ab50 Support parallel work of multiple IPCMEssageSubscribers in one process
- 87bb513 Fix failing symlink test (#52145)
- 8255901 document *regex_replace* Jinja filter (#52326)
- 3211397 Merge pull request #52310 from garethgreenaway/merge-2018.3
  - 9324b83 Merge branch '2018.3' into merge-2018.3
- 71e9b66 Merge pull request #52312 from terminal-labs/zone-clarification
  - ae0e18f Merge branch '2018.3' into zone-clarification
  - e8c8dba Added in an explaination of the –zone flag approved in #52251
    - 4908ed3 Merge branch '2017.7' into merge-2018.3
    - b6a0161 Merge remote-tracking branch 'upstream/2017.7.9' into 2017.7
    - 1f99e28 Let's not include raet by default
    - 4306779 Merge branch '2017.7.9' into 2017.7
    - 36ed50d Lock to unittest-xml-reporting 2.2.1
    - 3e6cb79 Merge branch '2017.7.9' into 2017.7
    - 57348ca Add static requirements for Arch linux
    - cc6fb46 Additionally ignore files in nox virtualenvs and CI artifacts directories
    - bf9c55e Previously have setuptools-git installed if ioflo is to be installed
    - 973e1ca Merge remote-tracking branch 'upstream/2017.7.9' into 2017.7
    - 981335a Revert "Avoid a traceback on tornado.testing test classes"
    - f48ba6a Merge remote-tracking branch 'upstream/2017.7.9' into 2017.7
    - b039cec Windows now has a static requirements file and no longer needs hacks
    - a86af2b Merge pull request #52226 from dwoz/jenkins_build
    - 8c0f46b Use windows state to bootstrap windows builds

            - 7263956 Lock coverage and xml-unittest-reporting versions
            - dce4ffa Avoid a traceback on tornado.testing test classes
- **PR** #52314: (twangboy) Add the ability to pass a timeout to beacons @ *2019-03-27 19:11:13 UTC*
    - d3f4c9d Merge pull request #52314 from twangboy/fix_beacon_tests
    - a5fa99a Add missing **kwargs
    - 7e88d04 Add the ability to pass a timeout to beacons
- **PR** #52311: (garethgreenaway) [2019.2] Merge forward from 2018.3 to 2019.2 @ *2019-03-26 18:31:32 UTC*
    - bad4e47 Merge pull request #52311 from garethgreenaway/merge-2019.2
    - 11cd485 Merge branch '2018.3' into merge-2019.2
    - 582d306 Merge pull request #52307 from Akm0d/doc_fix
        - 4da0fd6 Fix x509 inconsistent file name in state example
    - 2cdf48e Merge pull request #52276 from terminal-labs/replace-github-dead-link
        - a2fbbe0 Merge branch '2018.3' into replace-github-dead-link
    - 8fa0e51 Merge pull request #52290 from Akm0d/doc_fix
        - ca00e4b Merge branch '2018.3' of github.com:saltstack/salt into doc_fix
        - 8e1ec15 Add missing master/minion docs for log_rotate*
            - 08728cf Merge branch '2018.3' into replace-github-dead-link
    - d2fd84f Merge pull request #51724 from clinta/docker-create-with-ip
        - c933663 Merge branch '2018.3' into docker-create-with-ip
    - 838a7aa Merge pull request #51753 from terminal-labs/docs-sphinx-ref-deadlink-fix
        - f2b6abd Merge branch '2018.3' into docs-sphinx-ref-deadlink-fix
        - 9857f53 Merge branch '2018.3' into docs-sphinx-ref-deadlink-fix
        - 8a11d27 Merge branch '2018.3' into docs-sphinx-ref-deadlink-fix
        - 842ec84 Merge branch '2018.3' into docs-sphinx-ref-deadlink-fix
        - 9275207 Merge branch '2017.7' into docs-sphinx-ref-deadlink-fix
        - 2330c2a makefile had an old sphinx-doc link, updated to new general front page of sphinx docs.
        - 8d70e3b Replaced all bad sphinx links with ones from the new sphinx doc site.
        - ebdffcd replaced two python sphinx deadlinks with sphinx new relevant documentation pages.
            - a4e3e34 Merge branch '2018.3' into docker-create-with-ip
    - efb9ec0 Merge pull request #52215 from garethgreenaway/bp-52212
        - 65857ef Merge branch '2018.3' into bp-52212
        - a96bf24 Back-port PR #52212 to 2018.3
            - c6e553c add networking_config to container_create for custom network_mode
            - 5c99dbd add test to expose ip not added to custom network #51723
    - 679c4b2 Merge pull request #52269 from garethgreenaway/merge-2018.3
        - ee58560 Merge branch '2018.3' into merge-2018.3

- 418e543 Merge pull request #52261 from dwoz/update_kitchen
  - a87c6ea Use windows state to bootstrap windows builds
  - 1aa1bd8 Merge branch '2017.7' into merge-2018.3
  - 2962e94 Lock coverage and xml-unittest-reporting versions
  - 595303b Merge remote-tracking branch 'upstream/2017.7.9' into 2017.7
    - 55e150c Proper setup and teardown
    - b9b6009 Don't run distro on windows
    - 3cb5c82 Merge pull request #52208 from s0undt3ch/2017.7.9
  - b14dbc9 Merge branch '2017.7.9' into 2017.7
    - 6a789a1 Update static requirements
    - cee9480 Lock PyYAML to < 5.1
    - a4f241d The kitchen nox verifier already takes care of the sinlge *-v* flag.
    - 0cb641e Fix static requirements search bad logic
    - 45a3e4a *distro* reports Opensuse 15 as Opensuse Leap 15
    - de7bb24 Ubuntu 18.04 locked requirements files
    - bc08e3c Ubuntu 16.04 locked requirements files
    - 0d353d2 Ubuntu 14.04 locked requirements files
    - 1186122 Opensuse 42 locked requirements files
    - fa6888b Opensuse 15 locked requirements files
    - b6578b8 Fedora 29 locked requirements files
    - 777dbb0 Fedora 28 locked requirements files
    - 4388fa9 Debian 9 locked requirements files
    - 2f9030f Debian 8 locked requirements files
    - 1b18ec5 CentOS 7 locked requirements files
    - 6e9f0c0 CentOS 6 locked requirements
    - 7415846 Start handling static(and platform specific) requirements files
    - ea4d370 Allow overriding some requirements based on distro information
    - 17017a1 Merge pull request #51361 from olipovch/cloud-sync-after-install
    - 8fde3a0 Merge branch '2017.7' into cloud-sync-after-install
    - ab00151 Fix salt-cloud sync_after_install functionality
  - ff54f53 The kitchen nox verifier already takes care of the sinlge *-v* flag.
  - a2aa708 Fix static requirements search bad logic
  - 621b047 *distro* reports Opensuse 15 as Opensuse Leap 15
  - bc00a26 Ubuntu 18.04 locked requirements files
  - d5ed9e1 Ubuntu 16.04 locked requirements files
  - 03879bf Ubuntu 14.04 locked requirements files

    - c3d7a18 Opensuse 42 locked requirements files
    - b29d986 Opensuse 15 locked requirements files
    - 73154d0 Fedora 29 locked requirements files
    - bf36cd4 Fedora 28 locked requirements files
    - 52cc5bb Debian 9 locked requirements files
    - cb4a823 Debian 8 locked requirements files
    - 2128a79 CentOS 7 locked requirements files
    - 4a29e93 CentOS 6 locked requirements
    - 0d96e64 Start handling static(and platform specific) requirements files
  - 0fc82bd Merge pull request #52193 from Ch3LL/bp-47005
    - 1a54fd1 Merge branch '2018.3' into bp-47005
  - 0f3cd2a Merge pull request #52194 from Ch3LL/bp-52149
    - bff3f3c Merge branch '2018.3' into bp-52149
    - 7e459a1 example names parameter in file.managed state
      - f68caa6 Merge branch '2018.3' into bp-47005
  - 85be178 Merge pull request #52201 from max-arnold/rename-beacon-validate
    - e1e9a2b Merge branch '2018.3' into rename-beacon-validate
  - 0383ac3 Merge pull request #52189 from Ch3LL/merge-2018.3
    - 9972980 Merge branch '2018.3' into merge-2018.3
    - 79962ab Merge branch '2017.7' into '2018.3'
      - fd6068e Allow overriding some requirements based on distro information
      - 659ec11 Merge remote-tracking branch 'upstream/2017.7.9' into 2017.7
      - 18bbbec We need to nest require *req_win.txt*
      - 956a84c Merge pull request #51995 from s0undt3ch/2017.7
      - 0bf9d62 Merge remote-tracking branch 'upstream/2017.7.9' into 2017.7
      - baf776b Merge pull request #51974 from s0undt3ch/features/nox-2017.7
      - ee4df3a Lint fixes
      - 56f841f Windows needs special treatment
      - c3cb447 Requirements fixes
      - 6a67531 We need a fixed winrm-fs so that dotfiles are also included
      - 15c6a53 use kitchen-salt from my fork for now
      - b745332 Merge pull request #51800 from dwoz/typo_fix
      - 5bbe5e0 Fix up docstring
      - 9f3c583 Fix typo in docstring
      - 75acce1 Rename obsolete beacon validation function
      - d1700f3 Merge branch '2018.3' into bp-47005

    - 03f228d Merge pull request #52112 from Ch3LL/proxy_gs
        - 7ad4758 Update device names in proxy getting started docs
            - 25ae7c8 remove freebsd-upstream reference
            - b70c45c Remove salt FreeBSD repo which is now 2 years out of date
            - 62e5404 github issue tracker link and sends you to the url for listed issues.
            - ccf6a06 Replaced a dead link for github issue tracker with another github link for the issue tracker and added a link to show more options for closing issues in commit messages.
- **PR** #52190: (Ch3LL) [2019.2] Merge forward from 2018.3 to 2019.2 @ *2019-03-23 01:09:33 UTC*
    - 3ac6dbd Merge pull request #52190 from Ch3LL/merge-2019.2
    - 68c02b9 Merge branch '2019.2' into merge-2019.2
- **PR** #52240: (soer7022) Fix Typos "swtich" to "switch" @ *2019-03-23 00:08:50 UTC*
    - f5fc807 Merge pull request #52240 from soer7022/patch-2
    - d130832 Merge branch '2019.2' into patch-2
    - 3b6fdf8 Merge branch '2019.2' into patch-2
    - 5db7a3f Fix Typos "swtich" to "switch"
        - 34d601e Merge branch '2019.2' into merge-2019.2
- **ISSUE** #52036: (ghost) rsync.synchronized: "An exception occurred in this state: 'Changes' should be a dictionary." (refs: #52040)
- **PR** #52040: (garethgreenaway) [2019.2] Fix to rsync state @ *2019-03-21 19:08:19 UTC*
    - a00245e Merge pull request #52040 from garethgreenaway/52036_rsync_state_no_changes_when_error
    - 09d1672 Merge branch '2019.2' into 52036_rsync_state_no_changes_when_error
- **PR** #51962: (Ch3LL) Update 2019.2.0 release notes with other tojson example @ *2019-03-21 16:37:43 UTC*
    - 5387821 Merge pull request #51962 from Ch3LL/rn_update
    - f63bae9 Merge branch '2019.2' into rn_update
- **PR** #52262: (dwoz) Use windows state to bootstrap windows builds @ *2019-03-20 22:12:25 UTC*
    - 0fe7f28 Merge pull request #52262 from dwoz/update_kitcen_2019.2
    - 49d529d Use windows state to bootstrap windows builds
    - e2c3f93 Merge branch '2019.2' into rn_update
    - 164398f Merge branch '2019.2' into rn_update
    - c04b2a0 Merge branch '2019.2' into rn_update
    - 9ca6e50 Update doc/topics/releases/2019.2.0.rst
    - 052f193 add additional formatting to 2019.2.0 release notes
    - a6ca59a Update 2019.2.0 release notes with other tojson example
        - 84cd46c Merge branch '52036_rsync_state_no_changes_when_error' of github.com:garethgreenaway/salt into 52036_rsync_state_no_changes_when_error
            - 4a126b3 Merge branch '2019.2' into 52036_rsync_state_no_changes_when_error
        - d5d3117 Removing debugging statements

- 0fb859d Adding a test for the rsync state
- 4c678e6 When the retcode is non-zero, there are no changes so should include the default empty dictionary.
- c6ba2e6 Fix lint for test_win_file
- 9ada8d9 Merge branch '2018.3' into '2019.2'
- f8c0266 Allow overriding some requirements based on distro information
- c58c775 Merge pull request #52035 from Ch3LL/bp-52012
  - 37666bd Merge branch '2018.3' into bp-52012
- 54ff3fe Merge pull request #51935 from Ch3LL/py3_doc
  - b217f94 Merge branch '2018.3' into py3_doc
  - 605bfaf Update futures dependency for python2 only
  - 072cd84 Update supported python versions in installation docs
- 892cdae Merge pull request #52038 from terminalmage/docs
  - 4d82146 Minor documentation tweak for file.replace state
- a7a9754 Merge pull request #51896 from twangboy/win_disks
  - 057d415 Merge branch '2018.3' into win_disks
  - 0d419e3 Fix some lint
  - aa61dab Add tests
  - 9f6ec53 Add all attached drives to the disks grain
- 48d298c Merge pull request #51918 from twangboy/fix_compat
  - a791901 Fix compat, add tests
  - 2837bb9 Fix _compat for py3
- 02328a6 Merge pull request #52030 from Ch3LL/bp-52004
  - b4e3183 Fixed comparison of *None* with constants
- aa82fcc Merge pull request #52029 from Ch3LL/bp-52003
  - fcac9f3 Skip Test on Non-Windows systems
  - f42d17d Derive from LoaderModuleMockMixin
  - 5052017 Define __opts__ dict
  - b7b571e Define __opts__ dict
  - ee3ad08 Define __opts__ dict
  - 64d8451 Use default file module
  - 886ef19 Fix name of makedirs_ method
  - 4cc5f77 Check for path to be a symlink
- 5ff1725 Merge pull request #52028 from Ch3LL/bp-50384
  - 6493eb0 Fix issue #50381
  - f2b4ec8 Don't log debug messages during close

- 7ab1281 Merge pull request #51936 from gtmanfred/2018.3
  - faa3471 Merge branch '2018.3' into 2018.3
- 2ebfa22 Merge pull request #51967 from twangboy/add_compat_tests
  - 2056578 Merge branch '2018.3' into add_compat_tests
  - e85f004 Merge branch '2018.3' into add_compat_tests
  - ea69062 Fix lint
  - eb984f5 Add unit tests for _compat.py
- c8e2971 Merge pull request #51980 from Ch3LL/merge-2018.3
  - 9adc19e Merge branch '2017.7' into '2018.3'
  - bfaa6a9 Merge pull request #51895 from twangboy/win_update_2017.7
  - b2cb1eb Merge branch '2017.7' into win_update_2017.7
  - 7d6280d Merge pull request #51875 from dwoz/avoid_ssl_race
  - e99654d Merge branch '2017.7' into avoid_ssl_race
  - c0bb1f7 Merge pull request #51850 from waynew/cherry-pick-ping-fix
  - 6da495f Fix linter
  - f78b360 fix reactor ping test
  - ad7a9ec Merge pull request #51893 from twangboy/gate_zfs_2017.7
  - e0d58eb Gate zfs module on Windows
  - f758a10 Avoid race condition when initializing OpenSSL
  - 0f125db Add __virtual__ to win_update
  - 864d4be Merge pull request #51770 from Ch3LL/doc_conf_2017
  - 25d9ca2 Update release versions for the 2017.7 branch
  - 3cf3099 Merge pull request #51774 from terminal-labs/py3-server-docs-update
  - 1f00c89 Added in python 3 server according to the consensus from #51748
  - bfb9d87 added the python3 command for startign the python http server.
  - b4153d3 Merge pull request #51757 from twangboy/fix_case_2017.7
  - 6b0b5d2 Fix failing py3 tests
  - feb29f3 Make recurse and directory case sensitive
- 2c4dff6 Merge pull request #51905 from aplanas/fix_mount
  - ac688df mount: fix extra -t parameter
- 4b2c8cc Merge pull request #51815 from garethgreenaway/51673_multiple_file_sections_with_excludes
  - 58ef51d Merge branch '2018.3' into 51673_multiple_file_sections_with_excludes
  - 8c4485d Merge branch '2018.3' into 51673_multiple_file_sections_with_excludes
  - b55a894 Fixing some code that was not working correctly with Python 3.

      - 63da8be Fixing a bug that caused excludes to not work as expected when multiple file options were in place, the excludes of the last file section were being used over any previous ones. Including a test to ensure excludes work as expected when multiple file sections are in place.
    - 1bf3fc8 Merge pull request #51867 from Oloremo/mysql-module-fix-for-python3
      - dfcfebd Merge branch '2018.3' into mysql-module-fix-for-python3
      - b6d217c Merge branch '2018.3' into mysql-module-fix-for-python3
      - 9745b0a Fixed typo
      - 8149e9e Merge branch '2018.3' into mysql-module-fix-for-python3
      - 58d8a2e Merge branch '2018.3' into mysql-module-fix-for-python3
      - f964962 Added handling of empty server_version
      - 58791e9 Decode mysql_version for python3 compatibility
    - 7f05abf Merge pull request #51908 from garethgreenaway/51862_icinga_fixes
      - 233dbaf Adding __salt__ to utils/icinga so cmd.run_all is available.
      - 1e7490c update kubernetesmod in index.rst
      - 1f3be1c fix import in test
      - 8572c17 rename kubernetes module to kubernetesmod
- **PR** #52032: (Ch3LL) Add azure state and execution modules to doc index @ *2019-03-11 15:46:49 UTC*
  - 7d6b53c Merge pull request #52032 from Ch3LL/fix_azure_docs
  - 8a8a5fa Add azure module ref pages and fix doc errors
  - 6a1d06a fix additional azure state module doc errors
  - 65d8f27 Update doc failures in azure modules
  - 92dbe2a Merge branch '2019.2' into fix_azure_docs
- **PR** #52037: (Ch3LL) Backport #51201 into 2019.2 @ *2019-03-07 19:45:18 UTC*
  - **PR** #51201: (terminalmage) Lint: Fix str-format-in-logging, len-as-condition, etc. (2) (refs: #52037)
  - 96935c9 Merge pull request #52037 from Ch3LL/bp-51201
  - 1fb0697 Remove develop modules not needed in 2019.2
  - e3adc2f Lint: Fix str-format-in-logging, len-as-condition, etc. (2)
- **PR** #52034: (terminalmage) fileclient: Fix traceback when template file cannot be cached @ *2019-03-07 19:41:24 UTC*
  - 75398e4 Merge pull request #52034 from terminalmage/fix-get-template
  - 744dfd9 fileclient: Fix traceback when template file cannot be cached
- **PR** #51984: (Ch3LL) [2019.2] Merge forward from 2018.3 to 2019.2 @ *2019-03-07 17:58:42 UTC*
  - 7af4f5c Merge pull request #51984 from Ch3LL/merge-2019.2
  - 54ea40a Merge branch '2019.2' into merge-2019.2
  - 227ddbb Fix linter
  - a168913 Merge branch '2018.3' into '2019.2'

- c68744a Merge pull request #51975 from s0undt3ch/features/nox-2018.3
  - af4ae91 Lint fixes
  - 33955c9 Windows needs special treatment
  - d4f899f Requirements fixes
  - e550c60 We need a fixed winrm-fs so that dotfiles are also included
  - ada1ef3 use kitchen-salt from my fork for now
- a689177 Merge pull request #51807 from Ch3LL/2018.3.5_rn
  - 2c0b85c Add 2018.3.5 release notes
- 6aee2c4 Merge pull request #51806 from twangboy/fix_beacon
  - 2bf6179 Fix some lint
  - 0f38eea Add lowercase test
  - 1459985 Make sure drive letter is uppercase for re compare
  - 7cdc660 Add some tests
  - 48ae8c6 Fix disk.usage beacon on Windows
- da815ba Merge pull request #51966 from twangboy/use_winapi_com
  - a38300b Use winapi.com instead of coinitialize
- 7673c69 Merge pull request #51920 from twangboy/add_zabbix
  - e2d496c Add zabbix to the Windows builds
- 77c2796 Merge pull request #51802 from garethgreenaway/merge-2018.3.4
  - 2e20939 Merge branch '2018.3' into merge-2018.3.4
  - 769f002 Merge branch '2018.3.4' into merge-2018.3.4
  - b53a843 Merge pull request #51713 from dwoz/gemstestagain_2018.3.4
  - f2c6631 Merge branch '2018.3.4' into gemstestagain_2018.3.4
  - 1c82d24 Fix ruby gems tests, again
- 496525d Merge pull request #51930 from Ch3LL/release_docs
  - 9a68136 Update release documentation
- a027c97 Merge pull request #51931 from Ch3LL/bp_51298
  - 873cabd Roll back some changes to _compat.py
  - 1fcb0ff Fix edge case when minion ID is a 16-character string
- 1fa2072 Merge pull request #51892 from twangboy/gate_zfs
  - fbc2c0c Gate zfs grain on Windows
- 89b0991 Merge pull request #51836 from stratusjerry/backport_49039_2018.3
  - 35fdc4c Update win_pkg.py
- c1ab63d Merge pull request #51771 from Ch3LL/doc_conf_2018
  - 904beb3 [2018.3] change build_type and release in doc/conf.py
  - 3efcea5 Update release versions for the 2018.3 branch

        - 98f1b03 Merge pull request #51775 from garethgreenaway/merge-2017.7
            - 92ffe03 fixing lint
            - 73b601d Merge branch '2017.7' into merge-2017.7
            - e04f735 Merge pull request #51502 from ogd-software/51501-states-boto3_route53.py-fails-to-validate-ALIAS-A-records
            - e69ce9d Merge branch '2017.7' into 51501-states-boto3_route53.py-fails-to-validate-ALIAS-A-records
            - 5093406 Implement PEP8 E711 styling fix
            - 7640e90 Ensure boto3_route53 correctly validates existing ALIAS records
            - ff633b0 Ensure boto3_route53 survives attempts to validate ALIAS records
            - 9c128aa Merge pull request #51732 from s0undt3ch/features/nox-2017.7
            - 328989d Add initial nox configuration. Tell *tox*'ers to use *nox*'.
            - 99ce48d Merge pull request #51712 from dwoz/gemstestagain
            - 0d3b227 Fix ruby gems tests, again
            - 18452c4 Merge pull request #51528 from twangboy/backport_51471
            - 8c79f02 Merge branch '2017.7' into backport_51471
            - 20a7ee5 Backport 51471
            - ee80ce3 Merge pull request #51655 from garethgreenaway/49661_starting_salt_master_openssl
            - f647b95 Merge branch '2017.7' into 49661_starting_salt_master_openssl
            - 1831e64 Removing options from libcrypto.OPENSSL_init_crypto so it will simply use the defaults.
            - 8d0f801 removing salt.crypt from payload.py which is not used and causing salt-key to break.
            - a5a54c4 Removing unused import which is causing salt-master to not start when using openssl 1.1.1-1.
            - c9c082c Fix azurearm field list doc error
            - af0537f Add azure state and execution modules to doc index

- **PR** #52004: (chdamianos) Fixed comparison of *None* with constants (refs: #52030) @ *2019-03-06 17:41:52 UTC*
    - 01bcff0 Merge pull request #52004 from chdamianos/fix-constants-comparison
    - 6fd60af Fixed comparison of *None* with constants
- **PR** #51981: (bryceml) Fix typo in 2019.2.0 release notes @ *2019-03-05 23:59:33 UTC*
    - 8d6bb5d Merge pull request #51981 from bryceml/2019.2
    - ecf0ef0 Fix typo in 2019.2.0 release notes
        - a65e621 Adding an additional note warning that the file will not be changed because test=True, but could be changed by other states.
        - 6c0dcf2 Swapping pchanges for changes in file state.
- **ISSUE** #51825: (OrangeDog) schedule.disable doesn't work in 2019.2.0 (refs: #51855)

- **PR** #51855: (garethgreenaway) [2019.2] Global enable/disable was being ignored @ *2019-03-05 19:40:34 UTC*
  - c3105df Merge pull request #51855 from garethgreenaway/51825_scheduled_global_enabled
  - 301e773 Merge branch '51825_scheduled_global_enabled' of github.com:garethgreenaway/salt into 51825_scheduled_global_enabled
    - ae7ab06 Merge branch '2019.2' into 51825_scheduled_global_enabled
    - ff13833 Merge branch '2019.2' into 51825_scheduled_global_enabled
  - 9e7e4ed Fixing broken add_job test.
  - 72466a5 Global enable/disable was being ignored
- **ISSUE** #51745: (fayak) Systemd update 241 on archlinux breaks systemd.py version() function (refs: #51946)
- **PR** #51946: (Ethyling) Fix retrieve systemd version using regex @ *2019-03-05 19:20:19 UTC*
  - be28623 Merge pull request #51946 from Ethyling/fix-systemd-version
  - b557707 Add test for systemd version from git describe
  - 94809d0 Fix retrieve systemd version using regex
- **ISSUE** #51816: (jkrauz) pillar_ldap broken in 2019.2.0 (refs: #51964)
- **PR** #51964: (dwoz) [2019.2] Make sure ldap attrs are string types on python 2 @ *2019-03-05 19:18:30 UTC*
  - c310022 Merge pull request #51964 from dwoz/issue_51816
  - a879140 Merge branch '2019.2' into issue_51816
  - 0e61cf3 Fix linter errors/warnings
  - 171217a Revert debug logging
  - 8c641c6 make LDAP attr defaults string types on py2
- **PR** #51976: (s0undt3ch) [2019.2] Nox fixes @ *2019-03-05 18:56:49 UTC*
  - cb817f1 Merge pull request #51976 from s0undt3ch/features/nox-2019.2
  - 0b4e68c Lint fixes
  - 257b3fc Windows needs special treatment
  - b298da0 Requirements fixes
  - 4f45a7e We need a fixed winrm-fs so that dotfiles are also included
  - fa70af2 use kitchen-salt from my fork for now
- **PR** #51761: (sdodsley) Add new module to release notes for 2019.2.0 @ *2019-03-05 09:44:56 UTC*
  - **PR** #51681: (sdodsley) Add new module to release notes for 2019.2.0 (refs: #51761)
  - f867e6e Merge pull request #51761 from sdodsley/cheery_pick_fb_releasenotes
  - 80192c8 Add new module to release notes for 2019.2.0
- **PR** #51799: (s0undt3ch) Catch attribute error @ *2019-03-05 09:38:08 UTC*
  - 08e3956 Merge pull request #51799 from s0undt3ch/2019.2
  - 342244a Merge branch '2019.2' into 2019.2
  - 3e8a17e Catch attribute error
- **PR** #51810: (Ch3LL) Add 2019.2.1 release notes @ *2019-03-05 09:26:47 UTC*

  - ff5d7d6 Merge pull request #51810 from Ch3LL/2019.2.1_rn
  - aaee5fa Add 2019.2.1 release notes
- **ISSUE** #51824: (OrangeDog) schedule.present state not idempotent in 2019.2.0 (refs: #51849)
- **PR** #51849: (garethgreenaway) [2019.2] Fix to schedule.list function to filter out attributes that are None @ *2019-03-01 19:56:17 UTC*
  - **PR** #51647: (dwoz) [develop] increase batch test timeout on windows (refs: #51849)
  - 3732004 Merge pull request #51849 from garethgreenaway/51824_splay_breaking_schedule_present
  - a0af441 Merge branch '2019.2' into 51824_splay_breaking_schedule_present
- **PR** #51878: (Ch3LL) Add new terraform salt-ssh roster feature to release notes @ *2019-02-27 19:18:49 UTC*
  - 3fdb564 Merge pull request #51878 from Ch3LL/terraform_rn
  - 502d197 fix spelling error in release notes
  - 4d53b41 Add new terraform salt-ssh roster feature to release notes
- **PR** #51837: (stratusjerry) Update win_pkg.py @ *2019-02-26 16:51:35 UTC*
  - **PR** #49039: (stratusjerry) Update win_pkg.py (refs: #51836, #51837)
  - e63c872 Merge pull request #51837 from stratusjerry/backport_49039_2019.2
  - 531b23d Update win_pkg.py
  - 39b5645 Splay defaulting to None internally, broke schedule.present causing it to always report differences when a state is run. Updating the schedule.list function to check if an attribute is None and remove it.
- **PR** #51772: (Ch3LL) Update release versions for the 2019.2 branch @ *2019-02-25 14:16:40 UTC*
  - 27dd9fb Merge pull request #51772 from Ch3LL/doc_conf_2019
  - f28cc9b [2019.2] change build_type and release in doc/conf.py
  - 4c670e2 Update release versions for the 2019.2 branch
- **PR** #51776: (garethgreenaway) [2019.2] Merge forward from 2018.3 to 2019.2 @ *2019-02-24 22:08:58 UTC*
  - 45425a4 Merge pull request #51776 from garethgreenaway/merge-2018.3
  - 186306a reverting change in release notes, adding change to doc/ref/modules/all/index.rst.
  - 1ef140a Fixing lint.
  - 6173578 Fixing lint errors
  - 38dcab9 fixing lint
  - e37fc20 Merge branch '2018.3' into merge-2018.3
  - 7e3e71c Merge pull request #51756 from twangboy/fix_case_sensitive
    - 4c55040 Fix failing py3 tests
    - 64f45f8 Make recurse and directory case sensitive
  - e75fc1c Merge pull request #51747 from ymasson/fix_mysql_grants
    - f614dd7 Fix escaping for special charaters
  - b1f7e85 Merge pull request #51387 from chrillux/make-binarydata-output-possible
    - 010393e Make binary data output possible

- 3475813 Merge pull request #51694 from assafShapira/2018.3
    - 895b97a Merge branch '2018.3' into 2018.3
    - bc2d27a fix minion bootstrapping issue "/usr/lib/python2.7/site-packages/salt/utils/cloud.py", line 472 Configuration value u'provider' needs to be set
- c7136cb Merge pull request #51738 from garethgreenaway/nftables_cleanup
    - 8201d1e Fixing lint.
    - d681064 Merge branch '2018.3' into nftables_cleanup
    - e37458e Fixing a couple tests.
    - ed1b743 Additional cleanup. Adding additional documentation for state module.
    - dc4afbb More updates to nftables module, state modules, and unit tests.
    - 3ec96e3 initial work to cleanup the nftables module, state module, and the unit tests.
- 2121049 Merge pull request #51733 from s0undt3ch/features/nox-2018.3
    - 97532c2 Add initial nox configuration. Tell *tox*'ers to use *nox*'.
- f4adf33 Merge pull request #51728 from gtmanfred/virtualnames
    - 61b4390 fix modulenames from virtualnames tests
- d681fb9 Merge pull request #51721 from dwoz/solarisipstests
    - 3263d31 Merge remote-tracking branch 'saltstack/2018.3' into solarisipstests
- fb0a538 Merge pull request #51676 from Ch3LL/merge-2018.3.4
    - 8111479 Merge branch '2018.3' into merge-2018.3.4
- e08f08d Merge pull request #50812 from gtmanfred/virtualnames
    - d59952f Merge branch '2018.3' into virtualnames
    - 33bb5bf fix use of virtualname
        - 2ca1253 Merge branch 'merge-2018.3.4' of github.com:Ch3LL/salt into merge-2018.3.4
        - 6d6910a Merge branch '2018.3' into merge-2018.3.4
        - b5a6265 Fix ruby gems tests, again
        - 05f2797 Merge branch '2018.3' into merge-2018.3.4
        - 9f0999f Merge branch '2018.3' into merge-2018.3.4
        - 139daa8 Merge branch '2018.3.4' into 2018.3
        - 9741289 Merge pull request #51646 from dwoz/batch_exit_2018.3.4
        - a8f25bb increase batch test timeout on windows
        - 9caafec Skip solarisips when not on solaris
- 31c6785 Merge pull request #51714 from dwoz/gemstestagain_2018.3
    - fab74bb Fix ruby gems tests, again
- 39aadfb Merge pull request #51689 from gtmanfred/2018.3
    - 8c35d51 fix import
    - 6a55cfdf specify minimum version of shade module

  - a475b70 Merge pull request #51469 from s0undt3ch/hotfix/fix-suse-locale-docker
    - 14f49b5 Don't prevent *get_locale* from working when running under docker
  - a59585d Merge pull request #51671 from Ch3LL/merge-2018.3
    - 64322db Fix is_windows utils call
    - a5c8df0 Fix pylint
    - 547889b Merge branch '2017.7' into '2018.3'
      - bcc13db Merge pull request #51562 from gilbrechbuhler/fix_rabbitmq_policy_change_check
      - ed2b1e1 Fix RabbitMQ policy definition update check
      - cafaa98 Merge pull request #51648 from dwoz/increase_wait
      - e2377a9 Merge branch '2017.7' into increase_wait
      - e3d28d9 Merge pull request #51643 from dwoz/batch_exit
      - 42d1d02 increase batch test timeout on windows
      - 7c2d7a4 Increase minion wait for slow boxes
      - b5ae53a Merge pull request #51631 from Ch3LL/lint-2017
      - c0aa4dc Fix linter by ignoring async await check.
      - 396622c Merge branch '2017.7' into lint-2017
      - ba8d0ea Merge pull request #51547 from s0undt3ch/2017.7
      - 8e009e0 Allow NOT to install pip dependencies
      - 30ce509 Merge pull request #51182 from lmf-mx/darwin-plist-update
      - 4297c0d Merge branch '2017.7' into darwin-plist-update
      - 68ac07f Merge pull request #51442 from dwoz/multi_master
      - 40cb4db Merge branch '2017.7' into multi_master
      - 2dc5171 Hanlde multi_master failover when daemonized
      - 8df083c Merge pull request #51608 from dwoz/wait_minions_2017.7
      - c55359f fix linter
      - 4702e22 Wait for minions to be pingable before starting tests
      - 5afa0a1 Merge branch '2017.7' into darwin-plist-update
      - f46e8be Fix service name in minion plist
      - 1da9cd9 Fix pylint on 2017.7
      - 39122ca Merge pull request #51589 from dwoz/script_fix_2017.7
      - 8697ce7 Disable pylint checks, only for 2017.7
      - e5972f5 Use the code directory instead of cwd for python path
  - b8764fd Merge pull request #51680 from Ch3LL/bp-50336
    - 3502dbb states.gpg: fix missing existing keys; modules.gpg: fix set trust level command

- **PR** #51734: (s0undt3ch) [2019.2] Add initial nox configuration. Tell *tox*'ers to use *nox*'. @ *2019-02-21 17:39:29 UTC*

  - 1894082 Merge pull request #51734 from s0undt3ch/features/nox-2019.2
  - 01b74c0 Add initial nox configuration. Tell *tox*'ers to use *nox*'.
- **PR** #51707: (isbm) Bugfix: pillar refresh @ *2019-02-21 15:12:49 UTC*
  - 74db589 Merge pull request #51707 from isbm/isbm-pillar-refresh-fix
  - 269ae76 Refresh matchers and beacons on pillar refresh
- **PR** #51674: (Ch3LL) [2019.2] Merge forward from 2018.3 to 2019.2 @ *2019-02-20 17:35:07 UTC*
  - 82cbd98 Merge pull request #51674 from Ch3LL/merge-2019.2
  - 4171271 Fix pylint, remove unused import
  - 3f14619 Remove skipif and change expected return for solarisips test
  - d3660d5 Merge branch '2019.2' into merge-2019.2
- **PR** #51715: (dwoz) [2019.2] Fix ruby gems tests, again @ *2019-02-19 21:44:21 UTC*
  - 7a82c65 Merge pull request #51715 from dwoz/gemstestagain_2019.2
  - 95ce288 Fix ruby gems tests, again
  - 043efa0 Skip solarisips when not on solaris
  - 7ed0d7d Fix ruby gems tests, again
  - b46eac3 Merge branch '2019.2' into merge-2019.2
- **PR** #51561: (isbm) SaltSSH: fix delay when creating thin TAR @ *2019-02-19 10:15:47 UTC*
  - 0bd2d6e Merge pull request #51561 from isbm/isbm-saltssh-thintar-delay-fix
  - 2b02975 Give up on sync for Py3 because of the way tests for py2 are written here
  - f79c81a Update UT mocking, adjusting for temporary file
  - 2f90054 Use temporary filename instead of output directly to the "thin.tgz"
  - d841164 Copy over the result, cleanup afterwards
  - c6792ee Add a function that reserves temporary filename for further tarball generation
- **PR** #51678: (Ch3LL) [2019.2] Merge forward from 2019.2.0.rc1 to 2019.2 @ *2019-02-17 19:02:33 UTC*
  - 5f9764a Merge pull request #51678 from Ch3LL/merge-2019.2.0.rc
  - f8372ca Merge branch '2019.2' into merge-2019.2.0.rc
- **ISSUE** #51503: (garethhowell) smartos.image_present fails with "does not exist" (refs: #51383)
- **ISSUE** #51351: (garethhowell) Exception using smartos.vm_present to create docker zone (refs: #51383)
- **PR** #51679: (Ch3LL) Backport #51383 into 2019.2 @ *2019-02-16 00:31:06 UTC*
  - **PR** #51383: (sjorge) imgadm module should deal with invalid images better (refs: #51679)
  - 2036c8a Merge pull request #51679 from Ch3LL/bp-51383
  - e4971fc imgadm module should deal with invalid images better
  - 58f2407 Merge branch '2019.2.0.rc1' into 2019.2
    - 6292f86 Revert "Remove custom grains module from 2018.3 branch"
    - 92bd6b9 Fix salt.utils.is_windows call
    - ffa2f5f Merge branch '2018.3' into '2019.2'

- a8898f6 Merge pull request #51635 from lomeroe/lgpo_boolean_element_fix
  - b97c2b2 Merge branch '2018.3' into lgpo_boolean_element_fix
  - 5a98fb9 Merge branch '2018.3' into lgpo_boolean_element_fix
- abd3760 Merge pull request #51610 from garethgreenaway/51266_schedule_enable_disable_break_save
  - fbacb35 Merge branch '2018.3' into 51266_schedule_enable_disable_break_save
- 6765eea Merge pull request #51649 from dwoz/increase_wait_2018.3
  - 9c964ef Merge branch '2018.3' into increase_wait_2018.3
- 852a6be Merge pull request #51644 from dwoz/batch_exit_2018.3
  - 1087d10 increase batch test timeout on windows
  - 918030e Increase minion wait for slow boxes
  - 3d52b26 Merge branch '2018.3' into 51266_schedule_enable_disable_break_save
- dadf414 Merge pull request #51637 from twangboy/backport_50887_2018.3
  - d8d2b86 backport #50887
- a55e8f6 Merge pull request #51626 from twangboy/fix_lgpo_2018.3
  - b2c9c3b Use makedirs instead of mkdir
  - 2be2f42 Merge branch '2018.3' into 51266_schedule_enable_disable_break_save
- 0f1a411 Merge pull request #51187 from menglong81/fix-broken-thing
  - 298ad5c Merge branch '2018.3' into fix-broken-thing
  - d879d18 fix broken network.py
  - 76770f3 ss commmand replace to netstat command
  - 241707e ss commmand replace to netstat command
  - deeefc7 Merge branch '2018.3' into 51266_schedule_enable_disable_break_save
- 7ed5189 Merge pull request #51613 from twangboy/fix_51566
  - 1206a94 Merge branch '2018.3' into fix_51566
- 8b06d06 Merge pull request #51630 from twangboy/add_tests_51534
  - ddc017c Merge branch '2018.3' into add_tests_51534
- c4bbe1a Merge pull request #51177 from nullify005/51069-ri-and-rdoc-removed
  - 6b1a201 Merge branch '2018.3' into 51069-ri-and-rdoc-removed
  - f339608 Merge branch '2018.3' into 51069-ri-and-rdoc-removed
  - e586370 Merge branch '2018.3' into 51069-ri-and-rdoc-removed
  - 13d3288 Fix lint issues
  - 5de5e64 Update tests
  - 22eda16 Fix lint violations
  - 2e6252c Support rubygems 3 cli param changes
  - 9e3a01b Use existing file instead of tzutil
  - 30c1832 Use tempfile to get temp dir

  - e3d3c5a Add tests for issue 51309
  - 1168141 Add test for explicit lists
  - 84f5d63 Merge branch '2018.3' into fix_51566
  - 7376cb9 Handle explicit lists properly
  - 28fc070 Merge branch '2018.3' into 51266_schedule_enable_disable_break_save
  - 0760133 adding integration.scheduler.test_helpers to various files to by pass the test_module_name_source_match check.
  - 1552fba lint cleanup
  - 086066b Adding a test to ensure _get_schedule which is used by the save functionality works when there is an enabled attribute.
  - 1c6d4fb When looping through the schedule to remove hidden attributes, we should only do that if the item in question is a dictionary. It could be the attribute that determines if the entire scheudle is enabled or disable.
  - 5ed8eb6 Properly create the value for a "True" boolean element item
- f1e29c8 Merge pull request #51548 from s0undt3ch/2018.3
  - 611065e Merge branch '2018.3' into 2018.3
- 00a6246 Merge pull request #51622 from brejoc/2018.3-empty-job-workaround
  - d263410 Prevents crash when there is no job entry
  - 5342305 Allow NOT to install pip dependencies
- fdb13a3 Merge pull request #51523 from lomeroe/solarisips_fixes
  - e8e91e3 Merge branch '2018.3' into solarisips_fixes
  - b7bfe17 lint fixes
  - cdaccf4 add tests to validate command being sent to cmd.run_all
  - eaa229d add unit test
  - 432428d Don't assume that each item in "pkgs" is a dict (as packed by pkg.installed state).
- e3d5fbe Merge pull request #51577 from garethgreenaway/merge-2017.7
  - e060b97 Merge branch '2018.3' into merge-2017.7
  - 3eee038 Fixing a couple issues that did not merge properly.
  - 25b2ca6 Merge branch '2018.3' into merge-2017.7
  - c337fce Merge branch '2018.3' into merge-2017.7
  - f9d6828 Fixing missed merge.
  - a041775 Merge branch '2017.7' into merge-2017.7
  - 61889ba Merge pull request #51452 from terminalmage/bp-50963
  - 54d2e02 Merge branch '2017.7' into bp-50963
  - 0c989f0 Merge pull request #51437 from twangboy/fix_raise_2017.7
  - ecd1637 Merge branch '2017.7' into fix_raise_2017.7
  - f3c9cd4 Add missing raise statement

- d06526c Allow for kwargs to be used in object initialization
- 0e760b5 pip states: Use case-insensitive dictionaries for pip.list return
- 5ca9f82 Add a CaseInsensitiveDict implementation
- 2aaa9f9 pip.installed: Fix traceback when _find_key doesn't return a match
- 67b7d28 Merge pull request #51370 from dwoz/setup_py_req_fix
- e1e06a0 Fix python 3 path for shell test cases
- 09cd5fc Merge branch '2017.7' into setup_py_req_fix
- 0de189d Handle pipenv like version restrictions
- e9b3d94 Merge pull request #51367 from dwoz/cloud_config_2017
- 3f4c5a3e Merge branch 'cloud_config_2017' of github.com:dwoz/salt into cloud_config_2017
- 8b8d41a Merge branch '2017.7' into cloud_config_2017
- 7bb4ceb Fix python 3 path for shell test cases
- a2bbf4d Fix linter issues
- 004d9b8 Cloud provider and profile configs are merged
- 3a21afa Coverage on windows chokes with the tox env var *COVERAGE_PROCESS_START*
- 1c62e27 Switch required PyTest dependency
- b80320d Merge pull request #51402 from dwoz/win_test_fix
- 21359e6 Merge branch '2017.7' into win_test_fix
- 1199530 Merge pull request #51397 from dwoz/gemstest
- 43ac36e Use gems.github.com no longer exists, use gemcutter.org
- bac652f Add testing path to run_script commands
- f511ad2 Merge pull request #51227 from astronouth7303/doc-modules-additional-globals
- cbe1e17 Merge branch '2017.7' into doc-modules-additional-globals
- 2a9bf17 Merge pull request #51312 from s0undt3ch/features/tox-runtests
- 3840af8 "Tell" coverage to track subprocesses.
- f67a529 Remove unused imports
- 15ab061 Show objects on assertion failure
- 0281fc5 Don't fail the test because of order
- 9a31be7 Don't call *.keys()* on dictionaries
- 97ced3a Add *pytest-salt-from-filenames* to pytest requirements
- 8d2c4d2 Remove duplicate dependency
- 5eec144 Add *.coveragerc*
- aa108d5 Run tests from tox, wether runtests or pytest
- 80a3826 Update TODO
- 1945c30 Document the client-defined dunders

        - 5bb9b32 Merge pull request #51393 from garethgreenaway/various_documentation_fixes
            - 8bb7535 Merge branch '2018.3' into various_documentation_fixes
            - c15406a Merge branch '2018.3' into various_documentation_fixes
            - 33bb4e4 Merge branch '2018.3' into various_documentation_fixes
            - 3582f6e Fixing the underline.
            - a8f42a0 Various documentation fixes.
        - 884981f Merge pull request #51465 from ezh/50556
            - a3ee70c fix #50556 state.orchestrate_show_sls
        - 231d6e2 Merge pull request #51541 from bdrung/silence-linux_distribution-deprecation-warning
            - c4feea1 Merge branch '2018.3' into silence-linux_distribution-deprecation-warning
            - e1dcbb5 Silence linux_distribution deprecation warning
        - f87d5ec Merge pull request #51558 from bdrung/test-zypp-plugin
            - c34967d Merge branch '2018.3' into test-zypp-plugin
        - edb1093 Merge pull request #51609 from dwoz/wait_minions_2018.3
            - d794edb fix linter
            - d6720d2 Wait for minions to be pingable before starting tests
        - 8d79359 Merge pull request #51599 from terminalmage/bp-51510
            - ed194c7 grains: assign os_family Debian to Debian derivative TurnKey Linux
        - 24aa083 Merge pull request #51598 from terminalmage/add-grain-option-documentation
            - 4ee9440 Improve ec2 pillar documentation
            - 9c02104 Add link to metadata grain module docs
            - 997ce26 Add documentation for the metadata_server_grains config option
        - d89c489 Merge pull request #51491 from terminalmage/issue51268
            - 550e9ab Don't log shell warning if output_loglevel is quiet
            - 7678c28 Do not load zyppnotify file on module import
        - c6dfded Merge pull request #51588 from dwoz/script_fix_2018.3
            - cbac390 Use the code directory instead of cwd for python path

- **PR** #51658: (dwoz) [2019.2] Let memcopy make the null bytes, better py3 support @ *2019-02-15 05:42:37 UTC*
    - **PR** #51586: (garethgreenaway) [2019.2] Merge forward from 2018.3 to 2019.2 (refs: #51658)
    - 1ea6f4b Merge pull request #51658 from dwoz/pip_test_fix
    - c811a70 Let memcopy make the null bytes, better py3 support
- **PR** #51650: (dwoz) [2019.2] Increase minion wait for slow boxes @ *2019-02-14 16:31:59 UTC*
    - 22a04c1 Merge pull request #51650 from dwoz/increase_wait_2019.2
    - ad294f3 Merge branch '2019.2' into increase_wait_2019.2
- **PR** #51645: (dwoz) [2019.2] increase batch test timeout on windows @ *2019-02-14 16:29:26 UTC*
    - a0982f7 Merge pull request #51645 from dwoz/batch_exit_2019.2

  - 1a882d7 Merge branch '2019.2' into batch_exit_2019.2

- **PR** #51586: (garethgreenaway) [2019.2] Merge forward from 2018.3 to 2019.2 (refs: #51658) @ *2019-02-14 16:25:29 UTC*
  - 705a62d Merge pull request #51586 from garethgreenaway/merge-2018.3
  - 09fb3e7 Merge branch '2019.2' into merge-2018.3
  - 95c505a Merge branch '2019.2' into merge-2018.3
  - 4abd9d0 Merge branch '2019.2' into merge-2018.3
  - 6c99cb1 Merge branch '2018.3' into merge-2018.3
    - 41e25a6 Merge pull request #51580 from garethgreenaway/merge-2018.3.4
      - 84e15ce Fixing lint
      - e891b3f Merge branch '2018.3.4' into merge-2018.3.4
      - d0cad3e Merge pull request #51553 from twangboy/update_libsodium
      - 4177f64 Update url to libsodium for mac builds
      - 564a65a Merge pull request #51544 from twangboy/fix_test_ssh_2018.3.4
      - 7771632 Fix ssh on Windows
      - b4299a3 Merge pull request #51532 from frogunder/new_releasenotes
      - 09faf0b Merge branch '2018.3.4' into new_releasenotes
      - c6bfe53 2018.3.4 releasenotes
      - c37bdb7 Merge pull request #51534 from twangboy/fix_51309
      - 3ad95bc Convert data to string using salt util
      - 5b41af3 Merge pull request #51530 from garethgreenaway/48204_mysql_module_handle_mariadb_versions
      - cdf5298 Later versions of MariaDB still support "ALL PRIVILEGES"
      - d31c902 Updating the mysql module to handle MariDB versions.
      - f3f345c Merge pull request #51529 from frogunder/releasenotes
      - fb678ee fix escaped asterisks
      - 6a4f0da 2018.3.4 releasenotes
      - a37fb7d Merge pull request #51525 from twangboy/backport_51095_2018.3.4
      - b6acc92 Backport 51095
      - 602ddc8 Merge pull request #51520 from frogunder/man_pages
      - 0cbe3f8 Update man pages for 2018.3.4
      - 1f981c6 Merge pull request #51429 from yosnoop/replace-noclass-with-nocl-for-dig
      - 5231e73 Merge branch '2018.3.4' into replace-noclass-with-nocl-for-dig
      - 14e6fb7 In case the test machine can't resolve
      - 95e8a2f The +nocl is supported much wider than +noclass
      - a00d708 Test dig options for salt.utils.dns.lookup
      - 209ef32 Merge pull request #51441 from Ch3LL/bp-51366

- 5bc466e Fix python 3 path for shell test cases
- 5351e5c Add testing path to run_script commands
- 2d8a364 Fix linter issues
- 9b8718a Cloud provider and profile configs are merged
- 90a8682 Merge pull request #51458 from twangboy/add_py2_warning
- 05522c0 Merge branch '2018.3.4' into add_py2_warning
- b492021 Merge pull request #51460 from twangboy/update_installer_mac_2018.3.4
- 3c92688 Merge branch '2018.3.4' into update_installer_mac_2018.3.4
- 892d98c Merge pull request #51471 from twangboy/fix_test_win_dns_client
- c9f6b01 Merge branch '2018.3.4' into fix_test_win_dns_client
- a05305d Merge pull request #51462 from ShantonRU/fix-insecure-sql-queries-in-mysql-module-1
- d3ef2cb Merge branch '2018.3.4' into fix-insecure-sql-queries-in-mysql-module-1
- f47dda6 Fix improper use of socket authentication in mysql.user_chpass
- ed62a2f Fix insecure SQL queries in mysql.user_chpass
- 33922c9 Fix the failing dns test on Windows
- 48936c2 Merge pull request #51445 from terminalmage/fix-custom-grains-traceback
- 191235d Remove custom grains module from 2018.3 branch
- 72fd1a2 Roll back icon change
- 7dd1704 Add welcome_py2.rtf
- c93e9e3 Add Py2 Warning to installer package
- b658900 Add Py2 Warning for Py2 Installer
- a880e42 Merge pull request #51438 from twangboy/fix_raise_2018.3.4
- 5830fad Add missing raise statement
- 28b898c Merge pull request #51432 from Ch3LL/bp-51324
- bd41ff0 Fixing various bits in x509 module and unit test to ensure tests are passing on python2 & python3
- 01717d2 Merge pull request #51377 from twangboy/fix_auditpol_2018.3.4
- 19e5181 Fix issue when directory doesn't exist
- 43a9d2f Merge pull request #51338 from Ch3LL/bp-51207
- 345ae67 Backport PR# 50771 to 2018.3 branch
- 3dcd11d Merge pull request #51337 from Ch3LL/bp-51170
- 4e85505 Merge branch '2018.3.4' into bp-51170
- 7656aa5 Merge pull request #51340 from Ch3LL/bp-51252
- e74a6c6 Fix py3 managed.file test failure
- 6f2c7cf Remoe un-needed log statement
- 824950c Fix 50221 regression test on Windows

    - 1840af9 peserve newlines from pillar data for file.managed
    - e90ca1d Merge pull request #51290 from s0undt3ch/hotfix/fix-network
    - da3aace Merge branch '2018.3.4' into hotfix/fix-network
    - 04f16b8 Fix tests. We no longer get *ipaddress* instances back.
    - c3fa82a Fix minion start issue
    - c893dc4 Fix saltmod roster test
    - b61854f Roster defines the roster system not a file
    - 484bab4 Fix linter
    - 5e2db2f Honor roster file for ssh orchestrations
    - c8db7e3 Merge pull request #51317 from terminalmage/fix-pr-50417
    - 4be2891 Merge branch '2018.3.4' into fix-pr-50417
    - e246fdb Merge pull request #51311 from twangboy/fix_test_file.test_managed
    - f4177e0 Remove mode parameter on Windows
    - 4c4ded3 Fix regression in dynamic pillarenv
    - 40b20126 Merge pull request #51292 from Ch3LL/bp-51226
    - aba114e Merge branch '2018.3.4' into bp-51226
    - fc0bec5 Merge pull request #51295 from Ch3LL/bp-51279
    - 5c4a7a4 Merge branch '2018.3.4' into bp-51279
    - dacd1ea Merge pull request #51293 from Ch3LL/bp-51233-2
    - a155fd7 Accounting for when permissions are not returned from file.check_perms, eg. windows systems.
    - 53ba068 When test=True and there are permissions changes, report it.
    - d2eba77 Remove network.py patch
    - d431c71 Add __context__ support
    - de6c13d Handle spaces in fstab opts, similar fix to #39593
    - 1abffb2 Merge pull request #51278 from dwoz/issue51248
    - c614198 Use pillar roots validation method
    - 0d2b0c6 Fix linter warnings
    - 5b98895 Fix wart in pillar_roots validation
  - f7d50b1 Merge pull request #50784 from sathieu/dynamic_file_roots
    - 7ca2aa4 Allow dynamic file_roots
  - 5e840a5 Merge pull request #51527 from garethgreenaway/bp-51475
    - 801abf2 Fix Elasticsearch retuner
  - 1f6da6a Merge pull request #50921 from terminalmage/issue49894
    - a35c023 Don't use a decorator that assumes /etc/apt/sources.list.d exists
    - 6a62e79 Add integration test for architecture support

    - 67ae5a5 Add support for architectures for APT pkgrepo.managed states
- 3f17776 Actually, remove python-ldap. It needs system deps. Have salt-jenkins do it.
- afcb6bd Fix the *ldap* pacakge name in requirements/tests.txt
- 588cb87 Merge pull request #51454 from terminalmage/fix-deprecation-warning
    - 529cce0 Fix deprecation warning in nested outputter
- 13de0b8 Merge pull request #50963 from terminalmage/issue50942
    - c84be3f Allow for kwargs to be used in object initialization
    - 2bf93fd pip states: Use case-insensitive dictionaries for pip.list return
    - 792f054 Add a CaseInsensitiveDict implementation
    - 1410164 pip.installed: Fix traceback when _find_key doesn't return a match
- 60953b7 Merge pull request #51366 from dwoz/cloud_config_merge
    - ae351eb Fix python 3 path for shell test cases
    - 48062c8 Merge branch '2018.3' into cloud_config_merge
    - eeea289 Add testing path to run_script commands
    - 542d857 Fix linter issues
    - a8af3ba Cloud provider and profile configs are merged
- 9ab2ec6 Coverage on windows chokes with the tox env var *COVERAGE_PROCESS_START*
- 4994c38 Switch required PyTest dependency
- 89710a7 Merge pull request #51401 from dwoz/run_script_win
    - 77fdf35 Add testing path to run_script commands
- 1438a62 Merge pull request #51389 from terminalmage/proxy-fixes
    - 3cec85e Normalize results if test proxy is enabled
- 8acd9e5 Merge pull request #51368 from s0undt3ch/hotfix/tests
    - f4e1692 Fix test
- ab60782 Merge pull request #51319 from terminalmage/issue51255
    - 04eb66b Merge branch '2018.3' into issue51255
- fdf2cf4 Merge pull request #51250 from yosnoop/fix-salt-utils-dns-lookup
    - 0ddfad1 Merge branch '2018.3' into fix-salt-utils-dns-lookup
    - 9fb4abe Merge branch '2018.3' into fix-salt-utils-dns-lookup
    - 67e4131 Merge branch '2018.3' into fix-salt-utils-dns-lookup
    - 9a34bcd Unit test for lookup with servers option
    - 9498fdc Fix _lookup_host to make it work when server specified
    - af867e1 Remove excessive whitespaces
    - 0d074a4 Fix a comment
    - 1363d2b Proper way to specify _server_ to query with _host_
    - 7b2facc Fix broken lookup in salt/utils/dns.py

    - 73528a5 Merge pull request #51339 from dwoz/aws_role_backport
      - 156f532 Backport aws role support
    - 7908875 Merge pull request #51341 from Ch3LL/bp-51299
      - e3080f1 Add sleep before sending stop over UDP
    - 326425c Merge pull request #51352 from garethgreenaway/fixing_service_beacon_documentation
      - 021fe85 Fixing the documentation for the service beacon. If not options are provided for the service then there needs to be an empty dictionary.
    - 14a84e8 Merge pull request #51333 from Ch3LL/merge-2018.3
      - 27d7a89 Merge branch '2018.3' into merge-2018.3
    - 31b921f Merge pull request #51304 from terminalmage/issue51270
      - 36673fb Merge branch '2018.3' into issue51270
    - 2c3aef7 Merge pull request #51313 from s0undt3ch/features/tox-runtests-2018.3
      - 2eafc5c "Tell" coverage to track subprocesses.
      - f6babeb Remove unused imports
      - ff6824c Don't fail the test because of order
      - 4d12ccb Show objects on assertion failure
      - ee2f10a Add *pytest-salt-from-filenames* to pytest requirements
      - 4f4d019 Remove duplicate dependency
      - 1d4b9f5 Add *.coveragerc*
      - 78c6d68 Run tests from tox, wether runtests or pytest
      - c02757d gitfs: Fix use of deprecated pygit2 function
      - ae62ebc Merge branch '2017.7' into '2018.3'
      - 9309639 Merge pull request #51305 from s0undt3ch/2017.7
      - 09b20df Add 2017.7 codecov config
      - 4fa2a02 Merge pull request #51267 from ethframe/fix-file-directory
      - c108825 Fix missing file error in file.directory.
      - ad17ac5 Fix mocks to reflect changes to git.latest state
      - c817213 Don't try to run git.config_get_regexp from nonexistant cwd
      - f8c3e44 git.latest: add auth to merge/reset calls when LFS used with SSH auth
      - 4b11d6b increase batch test timeout on windows
      - a5af9c5 Increase minion wait for slow boxes

- **PR** #51549: (s0undt3ch) [2019.2] Allow NOT to install pip dependencies @ *2019-02-13 18:16:52 UTC*
  - e7ee04e Merge pull request #51549 from s0undt3ch/2019.2
  - d2125aa Allow NOT to install pip dependencies
- **PR** #51595: (mirceaulinic) Fix ipaddress imports @ *2019-02-13 17:33:08 UTC*
  - c96464b Merge pull request #51595 from mirceaulinic/mircea/port-824e0e5

  - 91bcb4f Merge branch '2019.2' into mircea/port-824e0e5
  - 84fc76d Fix ipaddress imports
- **PR** #51517: (damianosSemmle) Fixed string format index out of range error @ *2019-02-13 08:18:38 UTC*
  - e3c9a3e Merge pull request #51517 from damianosSemmle/fix-string-format-index
  - c0959e2 Merge branch '2019.2' into fix-string-format-index
- **PR** #51581: (garethgreenaway) [2019.2] Merge forward from 2019.2.0rc1 to 2019.2 @ *2019-02-13 04:11:15 UTC*
  - 82500bb Merge pull request #51581 from garethgreenaway/merge-2019.2.0rc1
  - b00e7e8 Merge branch '2019.2' into merge-2019.2.0rc1
- **PR** #51607: (dwoz) [2019.2] Wait for minions to be pingable before starting tests @ *2019-02-13 00:48:47 UTC*
  - edb5e94 Merge pull request #51607 from dwoz/wait_minions_2019.2
  - ab0a6a3 fix linter
  - f2e4a30 Wait for minions to be pingable before starting tests
- **PR** #51584: (dwoz) [2019.2] Use the code directory instead of cwd for python path @ *2019-02-12 18:43:46 UTC*
  - 1e0275e Merge pull request #51584 from dwoz/script_fixes
  - dbf4356 Use the code directory instead of cwd for python path
  - cd24006 Merge branch '2019.2' into merge-2019.2.0rc1
  - 6e55164 Merge branch '2019.2.0.rc1' into merge-2019.2.0rc1
    - ac73745 Merge branch '2019.2' into fix-string-format-index
- **PR** #51583: (dwoz) Add testing path to run_script commands @ *2019-02-11 21:38:51 UTC*
  - f446c52 Merge pull request #51583 from dwoz/scriptfix
  - e49ba86 Add testing path to run_script commands
  - 788f7e0 Fixed string format index out of range error
- **PR** #51405: (aplanas) Documentation: fix typo in "equivalent" @ *2019-02-01 21:11:12 UTC*
  - c3e27be Merge pull request #51405 from aplanas/backport_49669
  - 58279b4 Documentation: fix typo in "equivalent"
- **PR** #51406: (aplanas) states_pt3: fix rST link format @ *2019-02-01 21:10:43 UTC*
  - 5c26652 Merge pull request #51406 from aplanas/backport_49670
  - 204432d Merge branch '2019.2' into backport_49670
- **PR** #51426: (aplanas) mount: cache blkid information @ *2019-02-01 21:09:51 UTC*
  - fb70bd3 Merge pull request #51426 from aplanas/backport_51135
  - ac790a6 Merge branch '2019.2' into backport_51135
  - 66743d6 mount: cache blkid information
- **PR** #51425: (aplanas) cmdmod: add sysfs into the chroot @ *2019-02-01 21:09:00 UTC*
  - 2e8bf05 Merge pull request #51425 from aplanas/backport_51094
  - 9193106 cmdmod: add sysfs into the chroot
- **PR** #51419: (aplanas) file: update attributes for lsattr and chattr @ *2019-02-01 21:04:54 UTC*

  - d7a47f0 Merge pull request #51419 from aplanas/backport_50607
  - ab8a8b8 file: update attributes for lsattr and chattr
- **PR** #51414: (aplanas) service: SUSE is not based on sysvinit anymore @ *2019-02-01 21:02:33 UTC*
  - 7e020dd Merge pull request #51414 from aplanas/backport_50396
  - b5cff17 service: SUSE is not based on sysvinit anymore
- **PR** #51408: (aplanas) parted: fix the ordering of list command @ *2019-02-01 21:00:19 UTC*
  - **PR** #50473: (aplanas) parted: support variable length output for print (refs: #51408)
  - **PR** #49804: (aplanas) parted: fix the ordering of list command (refs: #51408)
  - 9f87dda Merge pull request #51408 from aplanas/backport_49804
  - 467daf2 parted: support variable length output for print
  - da2e632 parted: fix the ordering of list command
- **ISSUE** #50097: (aplanas) lowpkg.diff refers to a package name, but a package file is required (refs: #51410)
- **PR** #51410: (aplanas) Fix lowpkg.diff documentation and parameter name @ *2019-02-01 20:57:50 UTC*
  - 7e14499 Merge pull request #51410 from aplanas/backport_50126
  - d38622e Fix lowpkg.diff documentation and parameter name
    - e61a077 states_pt3: fix rST link format
- **PR** #51404: (aplanas) blockdev: fix url from comment @ *2019-01-30 20:05:54 UTC*
  - a16461b Merge pull request #51404 from aplanas/backport_49668
  - da2e30d blockdev: fix url from comment
- **PR** #51407: (aplanas) parted: fix _validate_partition_boundary @ *2019-01-30 20:05:02 UTC*
  - 756c367 Merge pull request #51407 from aplanas/backport_49803
  - b004b33 parted: fix _validate_partition_boundary
- **PR** #51435: (damianosSemmle) Fixed not raised exceptions @ *2019-01-30 18:32:57 UTC*
  - 1d45664 Merge pull request #51435 from damianosSemmle/fix-raise-exception
  - 7a9ff59 Fixed not raised exceptions
- **PR** #51336: (Ch3LL) [2019.2] Merge Forward from 2019.2.0.rc1 to 2019.2 @ *2019-01-27 23:55:08 UTC*
  - d4b9093 Merge pull request #51336 from Ch3LL/merge-2019.2.0rc1
  - 62a20b1 Remove unused import
  - 2284ea5 Merge branch '2019.2' into merge-2019.2.0rc1
  - 18bf236 Merge branch '2019.2.0.rc1' into 2019.2
- **PR** #51334: (Ch3LL) [2019.2] Merge forward from 2018.3 to 2019.2 @ *2019-01-27 23:49:32 UTC*
  - 02b43a0 Merge pull request #51334 from Ch3LL/merge-2019.2
  - 25ec653 Merge branch '2018.3' into '2019.2'
    - f62cc11 Merge pull request #51324 from garethgreenaway/1257_something_something_bytes_argh_python2
      - bd7072f Merge branch '2018.3' into 1257_something_something_bytes_argh_python2
    - b44cd84 Merge pull request #51323 from garethgreenaway/network_dns_check_test

    - 055eadc Merge branch '2018.3' into network_dns_check_test
- be57a0f Merge pull request #51321 from terminalmage/issue51256
    - d9c4462 Fix 500 error when using wheel_async
    - c847f54 Swap passed addr for string rather than ipaddress object.
    - 87f3972 removing another debugging log.
    - ad64efb Use MagicMock
    - 172ad2e removing debugging
    - 5812eae Adding a test to ensure dns_check works as expected.
    - 0e542c5 Fixing various bits in x509 module and unit test to ensure tests are passing on python2 & python3
- 85e7ac6 Merge pull request #51306 from s0undt3ch/2018.3
    - 5f0f296 Add 2018.3 codecov config
- 5a398a3 Merge pull request #51207 from twangboy/fix_test_pkg
    - d4ab6d8 Merge branch '2018.3' into fix_test_pkg
- 7a97cb4 Merge pull request #51252 from dwoz/issue-50221
    - f96828a Merge branch '2018.3' into issue-50221
- 253f5e4 Merge pull request #51264 from gtmanfred/2018.3
    - f3ebd32 Merge branch '2018.3' into 2018.3
- aacc047 Merge pull request #51274 from twangboy/fix_network
    - 344bf65 Fix tests. We no longer get *ipaddress* instances back.
    - 345aa5b Fix minion start issue
    - f553be7 tuples don't have pop
    - 55ea80e don't shadow builtin
    - 6148c50 an argument is not always needed to be passed to open a file
- 8a89cb2 Merge pull request #51279 from twangboy/backport_lgpo
    - 06b1545 Remove network.py patch
    - 2356f25 Add __context__ support
    - a5255f0 Merge remote-tracking branch 'saltstack/2018.3' into issue-50221
- 6796e65 Merge pull request #51276 from garethgreenaway/merge-2018.3
    - 44ddc43 Merge branch '2018.3' into merge-2018.3
- 0adde9b Merge pull request #51220 from sathieu/artifactory-classifier-2018.3
    - a85f146 fixes #31179 by appling the same logic as used in the nexus module
- f6a7cad Merge pull request #51225 from astronouth7303/loader-docs-2018.3
    - 6e7da95 Add the tokens system.
    - 7e2838c Add reference to the execution module docs
    - f30f9e8 Remove search system

    - f0cc183 Fileserver modules can be loaded from the fileserver.
  - 1a2f680 Merge pull request #51239 from garethgreenaway/51208_file_manage_escaped_double_quotes
    - 4251c0c Merge branch '2018.3' into 51208_file_manage_escaped_double_quotes
    - 6a9daa7 Merge branch '2018.3' into 51208_file_manage_escaped_double_quotes
    - 0bf468c Fixing lint. Using a constant intead of looking for the condition in the string.
    - a233dbc Ensuring we can handle a string that contains single quote + an escaped double quote.
  - d614127 Merge pull request #51261 from OrangeDog/patch-2
    - d809b9b Avoid double-counting errors and ignore test runs
    - 6ca7deb Remove unused host parameter and average/disaster keys
    - 2d44527 Just let zabbix_sender use the config file
    - 6404e2c Fix fallback command so it's actually run
    - b79a55a Fix KeyError in Zabbix returner
    - 0971035 Merge branch '2018.3' into merge-2018.3
    - 243dc06 Merge branch '2017.7' into merge-2018.3
    - acfb7d8 Update pytest requirements
    - c19479b Move all of PyTest requirements to its own requiremnts file
    - 39e633f Merge pull request #51246 from garethgreenaway/bp-50358
    - c33c558 Backport #50358 to 2017.7
    - c1ea10f Fix py3 managed.file test failure
    - 537114c Remoe un-needed log statement
    - ac978f0 Fix 50221 regression test on Windows
    - 65f38b5 peserve newlines from pillar data for file.managed
    - 464d813 Merge branch '2018.3' into fix_test_pkg
  - 276d621 Merge pull request #51197 from Ch3LL/tag_manager
    - b9a0705 Merge branch '2018.3' into tag_manager
    - 210238b Merge branch '2018.3' into tag_manager
    - 827d6c3 Merge branch '2018.3' into tag_manager
    - fcdfe69 Add Google Tag Manager to Docs
  - c8e70a9 Merge pull request #51206 from terminalmage/fix-reload
    - 8e23a6c Add missing import
    - 31ca7c6 Fix usage of reload for PY3 compatibility
  - 41ae390 Merge pull request #51231 from terminalmage/issue51056
    - 4a61477 Clarify documentation for the the gitfs "all_saltenvs" config param
    - 0574476 Merge branch '2018.3' into fix_test_pkg
    - b91b2c6 Backport PR# 50771 to 2018.3 branch

- **PR** #51314: (s0undt3ch) [2019.2] Allow running runtests.py using tox @ *2019-01-25 18:56:01 UTC*
  - 95607d1 Merge pull request #51314 from s0undt3ch/features/tox-runtests-2019.2
  - 39fb981 "Tell" coverage to track subprocesses.
  - f27589a Remove unused imports
  - 3000d20 Fix tests. We no longer get *ipaddress* instances back.
  - 8fcf235 Fix minion start issue
  - a26ba7f Don't fail the test because of order
  - 138ca2e Show objects on assertion failure
  - 840b3d2 Add *pytest-salt-from-filenames* to pytest requirements
  - 716bda4 Remove duplicate dependency
  - 6d62156 Add *.coveragerc*
  - 6bb56ce Run tests from tox, wether runtests or pytest
- **PR** #51307: (s0undt3ch) Add 2019.2 codecov config @ *2019-01-24 13:34:25 UTC*
  - b348034 Merge pull request #51307 from s0undt3ch/2019.2
  - 495aa5c Add 2019.2 codecov config
- **PR** #51280: (garethgreenaway) [2019.2] Merge forward from 2019.2.0rc1 to 2019.2 @ *2019-01-23 07:08:32 UTC*
  - d224096 Merge pull request #51280 from garethgreenaway/merge-2019.2-rc
  - e78ede2 Merge branch '2019.2' into merge-2019.2-rc
  - a425de6 Merge branch '2019.2.0.rc1' into merge-2019.2-rc
- **PR** #51277: (garethgreenaway) [2019.2] Merge forward from 2018.3 to 2019.2 @ *2019-01-23 07:01:29 UTC*
  - 8596ee4 Merge pull request #51277 from garethgreenaway/merge-2019.2
  - f05e754 Merge branch '2019.2' into merge-2019.2
- **ISSUE** #51273: (isbm) How to properly fix API? (refs: #51285)
- **PR** #51285: (garethgreenaway) [2019.2] Add metadata to accepted keyword arguments @ *2019-01-22 23:07:44 UTC*
  - 8cca51b Merge pull request #51285 from garethgreenaway/51273_metadata_auth_keyword_argument
  - bd82d80 Ensure metadata is an accepted keyword argument for authentication functions.
  - d0d5726 Merge branch '2018.3' into merge-2019.2
  - a6a9040 Merge pull request #51179 from terminalmage/fix-regex-chars
    - 9951b8d Fix possible oversight in nodegroup modifications
  - d5acd5b Merge pull request #51170 from dwoz/issue51158
    - b69c002 Fix saltmod roster test
    - 2594f24 Roster defines the roster system not a file
    - 05836b3 Merge branch '2018.3' into issue51158
    - bf47f6f Fix linter
    - 6223596 Honor roster file for ssh orchestrations

  - 5543a53 Update pytest requirements
  - f958df5 Move all of PyTest requirements to its own requiremnts file
  - 8f8ebbc Merge pull request #51226 from garethgreenaway/51195_handle_spaces_in_fstab_opts
    - 1902ce8 Merge branch '2018.3' into 51195_handle_spaces_in_fstab_opts
  - e3a9e99 Merge pull request #51233 from Ch3LL/bp-51105
    - eef49e8 Merge branch '2018.3' into bp-51105
  - e53e9b1 Merge pull request #51230 from Ch3LL/merge-2018.3
    - 0ef7675 Merge branch '2017.7' into '2018.3'
    - 4a12c5f Merge pull request #50852 from mirceaulinic/fix-capirca
      - fa91267 Merge branch '2017.7' into fix-capirca
      - 71913ff Merge branch '2017.7' into fix-capirca
      - 6fc3484 Lint?
      - 3d56750 Extra newline for lint
      - 5176cd8 Improve the Capirca and related modules for ACL config generation
    - 3f4759b Merge pull request #50923 from astronouth7303/patch-2
      - 4f631a6 Document {% include %}
    - 210e817 Merge pull request #51111 from s0undt3ch/2017.7
      - 08e0ae7 Update to salt-bootstrap v2019.01.08
      - c3520aa Accounting for when permissions are not returned from file.check_perms, eg. windows systems.
      - cd80548 When test=True and there are permissions changes, report it.
      - 89345eb Handle spaces in fstab opts, similar fix to #39593
- **ISSUE** #50538: (sjorge) salt.modules.pdbedit broken on Samba 4.9.x (refs: #50540, #51221)
- **PR** #51221: (Ch3LL) Backport #50538 into 2019.2 @ *2019-01-18 22:42:36 UTC*
  - **PR** #50540: (sjorge) Fix pdbedit module on samba 4.9 (Fixes #50538)
  - 0ccd5c3 Merge pull request #51221 from Ch3LL/bp-50540
  - 223e7c0 Merge branch '2019.2' into bp-50540
- **PR** #51232: (Ch3LL) [2019.2] Merge forward from 2018.3 to 2019.2 @ *2019-01-18 20:51:37 UTC*
  - 3edaf18 Merge pull request #51232 from Ch3LL/merge-2019.2
  - 3b966f3 Merge branch '2019.2' into merge-2019.2
  - ffe23fa Merge branch '2018.3' into '2019.2'
    - 26008ef Merge pull request #51212 from sathieu/file_roots_slash
      - b769f18 Allow slash in file_roots envs
      - f9437e2 Test for file_roots envs with slash leads to "No such file or directory"
    - 5916282 Merge pull request #51209 from garethgreenaway/bp-51124
      - 82acce4 Handle encoding when opening file for both py2/py3

- 094bd33 Merge pull request #51184 from twangboy/fix_auditpol
  - 2e68d35 directly call cmd.run_all
- ac71922 Merge pull request #51191 from Ch3LL/bp-50918
  - 7300d84 Skip debian tests on Windows
  - 4622433 Fix *test_debian_ip* on Windows
- e7f53a3 Merge pull request #51144 from jgleissner/2018.3-azurearm-py3
  - 3178f1a msazure.py: remove unused imports
  - 99a4b9d azurearm: fix list_locations
  - aebe34b azurearm: fix show_instance function
  - d80669f msazure: fix object_to_dict for python 3
- 1206f92 Merge pull request #51171 from twangboy/fix_crypt
  - 6caa93b Add random sleep time to avoid overloading Windows
- a96deed Merge pull request #51174 from jpsv/2018.3
  - 055823e Update opsgenie.py
  - 501c8df Merge branch '2018.3' into 2018.3
  - 220aa8a Update opsgenie.py
  - 5dc21cc Update to from V1 API to V2
  - f93ce7c Update to V2 API
- 563d487 Merge pull request #51010 from amendlik/ldap-groups
  - cd336ed Merge branch '2018.3' into ldap-groups
  - 94f8ee4 Update unit tests for auth.ldap
  - 08e36d0 Allow unauthenticated bind for listing LDAP groups
- 6dc3752 Merge pull request #51112 from s0undt3ch/2018.3
  - 4f1d846 Update to salt-bootstrap v2019.01.08
- b19c990 Merge pull request #51097 from terminalmage/fix-keyerror-raise
  - a3c0b49 Fix poorly-written test
  - c0ff7b1 Fix more incorrect exception raises
  - f11dbcc Fix improper KeyError raise
- 8920a86 Merge pull request #51169 from twangboy/fix_git_state
  - f2d1180 More descriptive error when missing GitPython or PyGit2
- 58eede6 Merge pull request #51173 from dwoz/issue51160
  - 11cc7b3 Remove un-needed test
  - 0f87812 Merge develop, Revert exception handling
  - 7365a1a Pass bytes to idna
- 83c38c6 Revert "wordsmithing"
- a40739c wordsmithing

- b348ace Make sure blackout tests clean up after themselves. Properly.
- 68a12d5 These should not be considered destructive tests
- 475404d fix docstring quote style
- 71c3574 Revert "backport test improvements from develop"
- e32d219 backport test improvements from develop Bogus random test failures forklifted from develop at 414bfe61a4fbe6f84be32242924f23c7126c24d6
- 42654d8 socket.connect needs a tuple TypeError https://docs.python.org/2/library/socket.html RTFD "Note This method has historically accepted a pair of parameters for AF_INET addresses instead of only a tuple. This was never intentional and is no longer available in Python 2.0 and later. "
- 27254fc syntax error bad parentheses
- 24bb94f Don't ip_bracket addresses returned by check_dns.
- 02b825e expose error in tests
- 0a92c46 Need more detail on test failure
- 3574698 ip_address does not need str() and parse_host_port handles ipv6 or ipv4
- 7a6bc57 avoid TypeError by not constructing an ip_address from an ip_address
- 35410dd try to debug error in tests (revert me)
- 2f8ca47 wordsmithing
- 1dd1f7d parsing logic error
- 98041c1 minimize diff
- c92a247 don't try to handle/log test exception
- eb60586 remove bad extra test assertion inside exception handler
- e1e5d87 checking for truthiness better than len
- a984a43 typo in ipaddress.IPv6Address()
- 5fc6c60 clear up lint, disambiguation
- 8c6dfea lint pep8 whitespace
- f9594b8 don't test invalid combination
- de823fd fix good_host_ports iteration
- c5a5b34 test assertion correction
- e169e0d fixup rebase merge goof
- d28bab6 fix test setup
- 42c710e lint pep8
- 15c6deb fix horrible mistakes
- c092baf fix indentation doh
- 9080745 debug ipaddress.ip_address TypeError
- 8bd82f4 debug ipaddress.ip_address TypeError
- 080ab70 fix parse_host_port() parse error on hostname only arg

- 571603c coverage for master host:port ipv4 and ipv6 config value support fix test_parse_host_port bad values lint
- f14ff16 fix minion zmq connecting to master configured as IPv6 address
- 9d72f13 pylint E741 triggers error in salt-pylint E8741 (unhandled)
- 2052838 avoid error on (redundant) is_ip check
- 0eaeb1e avoid error on (redundant) is_ip check
- f8bd08e fix parse_host_port() is_ip call
- 89519fe fix parse_host_port() parse error on hostname only arg
- 1417528 fix parse_host_port() bad exception on hostname only arg
- 194a024 coverage for master host:port ipv4 and ipv6 config value support fix test_parse_host_port bad values lint
- df73388 fix minion zmq connecting to master configured as IPv6 address
- 06854cf Merge pull request #51147 from Ch3LL/merge-2018.3
  - d39d8b7 Merge branch '2018.3' into merge-2018.3
- b79c09e Merge pull request #51161 from amendlik/long-hostname
  - 6fcc4ad Merge branch '2018.3' into long-hostname
- 1424734 Merge pull request #51159 from rkrieger/fix-issue-51067
  - 1440ccf Prevent crash by testing for 'manufacturer' in osdata on OpenBSD
  - 1a4e307 Ignore exceptions when generating the minion ID
- 5e37326 Merge pull request #51146 from terminalmage/fix-vsphere-tuple-issue
  - 5d60022 Fix invalid assignment in vsphere module
- fad8621 Merge pull request #51154 from twangboy/auditpol
  - 9064376 Add support for Advanced Audit policies
- f7a100c Merge pull request #51151 from Ch3LL/bp_51061
  - ffdae27 When writing output to stdout we want to ensure that the data is a string not bytes. Under py2 the salt.utils.data.encode function results in a string but under py3 the result is a bytestring. Swapping out salt.utils.data.encode for salt.utils.stringutils.to_str.
- 3a50a57 Merge pull request #51150 from Ch3LL/bp-49508
  - 7ae53ad Do not silently ignore errors.
  - deb0134 Convert to string before sending via 'publish'.
  - 651c551 Use to_str salt.utils when writing to a file.
- 8bb589d Merge pull request #51152 from garethgreenaway/50433_handle_grants_better
  - f7caa4d Fixing lint
  - e8c8c0f Adding some tests to ensure "ALL PRIVILEGES" is handled correctly in 8.0 and 5.6
  - b4bfd9f Add additional grants. Adding logic to handle when ALL or ALL PRIVILEGES is passed for the grant, including some logic to handle the fact that the grants are split when show grants is run for a particular user.
  - 74edfd9 Fix pylint

        - 81a3d47 Merge branch '2017.7' into '2018.3'
        - 8a76a19 Merge pull request #50858 from jasonarewhy/bugfix-2017.7/ubuntu-networking-searchdomain
        - 3395a3d Merge branch '2017.7' into bugfix-2017.7/ubuntu-networking-searchdomain
        - 69d25aa update debian_ip test
        - a6b896c get searchdomain in Ubuntu 12+
    - dda1ebb Merge pull request #51133 from 5uper5hoot/issue-#50849
        - ec14e13 Fixes mysql server version comparison.
    - 9d4321d Merge pull request #50878 from max-arnold/output-docs
        - d7403da Fix profile outputter docs
    - ce87504 Merge pull request #51076 from OrangeDog/patch-1
        - 568e4fb Revert add -> check
        - 4d7ebe7 Typos in index_template_present documentation
    - d08abf4 Merge pull request #51081 from garethgreenaway/merge-2018.3
        - 027804a Merge branch '2018.3' into merge-2018.3
    - 335088b Merge pull request #51080 from garethgreenaway/50966_nxos_ensure_kwargs_list
    - 230a66b Merge branch '2018.3' into 50966_nxos_ensure_kwargs_list
    - 629dc25 Ensure kwargs is a list before we attempt to loop through. Similar fix to https://github.com/rallytime/salt/commit/430c462f34eb4eedf2384e06fbee8dc19b8026f6 in the proxy module.
    - a20791b Merge branch '2018.3' into merge-2018.3
    - 9386aab Merge branch '2017.7' into merge-2018.3
    - 998feca Merge pull request #51030 from terminalmage/issue38502
    - 6dbd81f Add unit test to ensure we don't diff bare repos
    - 3c4b243 Merge branch '2019.2' into bp-50540
- **PR** #51234: (KChandrashekhar) Added a note in Release notes about py 2.7 EOL @ *2019-01-18 15:53:51 UTC*
    - e6ab913 Merge pull request #51234 from KChandrashekhar/2019.2-docs
    - e4afc81 Addressing Megan's comment to fix the py2.7 deprecation message in release notes
    - 8ec0296 Added a note in Release notes about py 2.7 EOL
- **PR** #51198: (Ch3LL) [2019.2] Merge forward from 2019.2.0.rc1 to 2019.2 @ *2019-01-17 21:23:34 UTC*
    - 644ef8c Merge pull request #51198 from Ch3LL/merge-2019.2-rc
    - 810c611 Merge branch '2019.2.0.rc1' into 2019.2
    - 3f61e77 Fix pdbedit module on samba 4.9 (Fixes #50538)
- **PR** #51148: (Ch3LL) [2019.2] Merge forward from 2018.3 to 2019.2 @ *2019-01-14 22:49:02 UTC*
    - 12901c3 Merge pull request #51148 from Ch3LL/merge-2019.2
    - 1d7329a Fix pylint
    - 5271636 Fixes mysql server version comparison.

  - b53f171 Ensure kwargs is a list before we attempt to loop through. Similar fix to https://github.com/rallytime/salt/commit/430c462f34eb4eedf2384e06fbee8dc19b8026f6 in the proxy module.
  - d0f2d15 Revert add -> check
  - 437d8b5 Typos in index_template_present documentation
  - 3bf189e Add unit test to ensure we don't diff bare repos
  - 2f8e391 Fix profile outputter docs
- **PR** #51100: (dwoz) Fix typo in method name (2019.2) @ *2019-01-08 18:12:23 UTC*
  - 4c2d76d Merge pull request #51100 from dwoz/cloud_typo
  - 60f53d0 Fix typo in method name
- **PR** #51099: (terminalmage) Fix improper exception raises (2019.2) @ *2019-01-08 17:55:53 UTC*
  - 39b28f4 Merge pull request #51099 from terminalmage/fix-exception-raise
  - 2a25530 Fix improper exception raise
- **PR** #51084: (s0undt3ch) [fluorine] Cleanup the singleton instances map. @ *2019-01-07 22:03:29 UTC*
  - 8827193 Merge pull request #51084 from s0undt3ch/merge-singleton-last-ref-close-fluorine
  - 979fbac Merge branch '2019.2' into merge-singleton-last-ref-close-fluorine
  - 63f051a Merge branch '2019.2' into merge-singleton-last-ref-close-fluorine
  - bd6c46d Cleanup the singleton instances map.

28.2.7 Salt 2019.2.2 Release Notes

Version 2019.2.2 is a bugfix release for *2019.2.0*.

Statistics

- Total Merges: **26**
- Total Issue References: **12**
- Total PR References: **26**
- Contributors: **13** (Akm0d, Ch3LL, Oloremo, OrlandoArcapix, bryceml, dhiltonp, dwoz, frogunder, garethgreenaway, javierbertoli, pizzapanther, s0undt3ch, twangboy)

Changelog for v2019.2.1..v2019.2.2

Generated at: 2019-10-11 20:54:15 UTC

- **PR** #54919: (twangboy) Add missing docs for win_wusa state and module (2019.2.1) @ *2019-10-11 18:28:11 UTC*
  - 7d253bc Merge pull request #54919 from twangboy/update_docs
  - 57ff199 Add docs for win_wusa
- **ISSUE** #54941: (UtahDave) Pillar data is refreshed for EVERY salt command in 2019.2.1 and 2019.2.2 (refs: #54942)
- **PR** #54942: (dwoz) Fix for 54941 pillar_refresh regression @ *2019-10-11 18:27:31 UTC*
  - 2f817bc Merge pull request #54942 from dwoz/fix-54941

  - cb5d326 Add a test for 54941 using test.ping
  - 348d1c4 Add regression tests for issue 54941
  - 766f3ca Initial commit of a potential fix for 54941
- **PR** #54897: (bryceml) update version numbers to be correct @ *2019-10-05 01:59:14 UTC*
  - f783108 Merge pull request #54897 from bryceml/2019.2.1_fix_docs
  - e9a2a70 update version numbers to be correct
- **PR** #54894: (bryceml) 2019.2.1 fix docs @ *2019-10-04 22:31:26 UTC*
  - 3233663 Merge pull request #54894 from bryceml/2019.2.1_fix_docs
  - 2456aaa Porting PR #52948 to 2019.2.1
  - 94a1e3b Porting PR #52752 to 2019.2.1
  - c7b7474 modifying saltconf ads
  - d48057b add new saltconf ads
- **PR** #54858: (frogunder) remove in progress from releasenotes 2019.2.2 @ *2019-10-02 20:42:59 UTC*
  - 4b06eca Merge pull request #54858 from frogunder/releasenotes_remove2019.2.2
  - a697abd remove in progress from releasenotes 2019.2.2
- **PR** #54854: (frogunder) releasenotes 2019.2.2 @ *2019-10-02 18:58:21 UTC*
  - aaf2d1c Merge pull request #54854 from frogunder/release_notes_2019.2.2
  - a41dc59 Update 2019.2.2.rst
  - 9bea043 releasenotes 2019.2.2
- **PR** #54852: (frogunder) Update man pages for 2019.2.2 @ *2019-10-02 18:27:07 UTC*
  - 10d433f Merge pull request #54852 from frogunder/man_pages_2019.2.2
  - 92bc4b2 Update man pages for 2019.2.2
- **PR** #54845: (s0undt3ch) Remove debug print @ *2019-10-02 17:38:21 UTC*
  - 8ca6b20 Merge pull request #54845 from s0undt3ch/hotfix/event-return-fix-2019.2.1
  - 3937890 Remove debug print
- **ISSUE** #54755: (Reiner030) 2019.2.1/2019.2.0 pip failures even when not using pip (refs: #54826)
- **PR** #54826: (dwoz) Fix issue 54755 and add regression tests @ *2019-10-01 20:07:46 UTC*
  - 9e3914a Merge pull request #54826 from dwoz/issue_54755
  - 0bad9cb Handle locals and globals separatly
  - bcbe9a2 Only purge pip when needed
  - d2f98ca Fix issue 54755 and add regression tests
- **PR** #54830: (frogunder) Add known issues to 2019.2.1 release notes @ *2019-10-01 16:23:30 UTC*
  - ba569d0 Merge pull request #54830 from frogunder/update_relasenotes_2019.2.1
  - 8cdb27b Update 2019.2.1.rst
  - 14f955c Add known issues to 2019.2.1 release notes
- **ISSUE** #54521: (Oloremo) [Regression] Failhard, batch and retcodes (refs: #54806)

- **PR** #54806: (Oloremo) [Regression] Batch with failhard fix @ *2019-10-01 14:51:47 UTC*
    - 433b6fa Merge pull request #54806 from Oloremo/failhard-batch-fix-2019.2.1
    - 6684793 Merge branch '2019.2.1' into failhard-batch-fix-2019.2.1
    - 3e0e928 Added tests for cli and runner
    - 2416516 Made batch work properly with failhard in cli and runner
- **ISSUE** #54820: (OrangeDog) schedule.present not idempotent when scheduler disabled (refs: #54828)
- **PR** #54828: (garethgreenaway) [2019.2.1] Fix global disabling code in scheduler @ *2019-10-01 09:27:49 UTC*
    - ed94aa5 Merge pull request #54828 from garethgreenaway/54820_fix_schedule_disabled_job_enabled_bug
    - be15a28 Rework code that handles individual jobs being disabled and scheduler being globally being disabled. Previously disabling the schedule would result in individual jobs being disabled when they were run through eval. This change does not change schedule items.
- **PR** #54778: (Akm0d) fix broken salt-cloud openstack query @ *2019-10-01 09:23:27 UTC*
    - 435b40c Merge pull request #54778 from Akm0d/master_openstack_query_fix
    - ba4ba2a fixed pylint errors in openstack test
    - d9a8517 Added openstack tests for openstack –query fix
    - 59214ad Fallback to image id if we don't have an image name
    - 3a42a4d fixed pylint error
    - 0074d18 created unit tests for openstack
    - 4255e3e Merge branch '2019.2.1' of https://github.com/saltstack/salt into HEAD
    - 1c2821b Return a configured provider, not a bool
    - c585550 fix broken salt-cloud openstack query
- **ISSUE** #54762: (margau) 2019.2.1: Breaks Minion-Master Communication (refs: #54823, #54784, #54807)
- **PR** #54823: (dhiltonp) ip_bracket can now accept ipv6 addresses with brackets @ *2019-10-01 01:13:34 UTC*
    - 93b1c4d Merge pull request #54823 from dhiltonp/maybe-bracket
    - faa1d98 ip_bracket can now accept ipv6 addresses with brackets
- **ISSUE** #54762: (margau) 2019.2.1: Breaks Minion-Master Communication (refs: #54823, #54784, #54807)
- **PR** #54807: (dwoz) Fix pip state pip >=10.0 and <=18.0 @ *2019-09-30 09:20:14 UTC*
    - **PR** #54772: (OrlandoArcapix) Fix import of pip modules (refs: #54807)
    - b61b30d Merge pull request #54807 from dwoz/patch-2
    - 664806b Add unit test for pip state fix
    - e637658 Revert change to pip version query
    - 42810a2 Fix import of pip modules
- **ISSUE** #54741: (kjkeane) Schedulers Fail to Run (refs: #54799)
- **PR** #54799: (garethgreenaway) Fix to scheduler when job without a time element is run with schedule.run_job @ *2019-09-30 00:19:43 UTC*
    - 4ee1ff6 Merge pull request #54799 from garethgreenaway/54741_run_job_fails_without_time_element

  - 44caa81 Merge branch '54741_run_job_fails_without_time_element' of github.com:garethgreenaway/salt into 54741_run_job_fails_without_time_element
    - 3ae4f75 Merge branch '2019.2.1' into 54741_run_job_fails_without_time_element
  - 8afd2d8 Removing extra, unnecessary code.
  - 549cfb8 Fixing test_run_job test to ensure the right data is being asserted. Updating unit/test_module_names.py to include integration.scheduler.test_run_job.
  - 7d716d6 Fixing lint.
  - ec68591 If a scheduled job does not contains a time element parameter then running that job with schedule.run_job fails with a traceback because data['run'] does not exist.
- **PR** #54785: (Ch3LL) Fix state.show_states when sls file missing in top file @ *2019-09-30 00:00:34 UTC*
  - b90c3f2 Merge pull request #54785 from Ch3LL/fix_show_states
  - 96540be Clean up files after state.show_states test
  - ad265ae Fix state.show_states when sls file missing
- **ISSUE** #54768: (paul-palmer) 2019.2.1 Some Jinja imports not found (refs: #54780)
- **ISSUE** #54765: (awerner) 2019.2.1: Jinja from import broken (refs: #54780)
- **PR** #54780: (dwoz) Fix masterless jinja imports @ *2019-09-29 22:12:48 UTC*
  - b9459e6 Merge pull request #54780 from dwoz/fix-masterless-jinja-imports
  - 5d873cc Merge branch '2019.2.1' into fix-masterless-jinja-imports
  - e901a83 Add regression tests for jinja import bug
  - 3925bb7 Fix broken jinja imports in masterless salt-call
- **ISSUE** #54776: (javierbertoli) Setting *ping_interval* in salt-minion's config (version 2019.2.1) prevents it from starting (refs: #54777)
- **PR** #54777: (javierbertoli) Fix minion's remove_periodic_callback() @ *2019-09-29 21:33:53 UTC*
  - 4c240e5 Merge pull request #54777 from netmanagers/2019.2.1
  - 459c790 Merge branch '2019.2.1' into 2019.2.1
- **PR** #54805: (bryceml) improve lint job @ *2019-09-29 21:24:05 UTC*
  - 83f8f5c Merge pull request #54805 from bryceml/2019.2.1_update_lint_salt
  - ffa4ed6 improve lint job
  - fa1a767 Merge branch '2019.2.1' into 2019.2.1
- **ISSUE** #54751: (jnmatlock) NXOS_API Proxy Minions Error KeyError: 'proxy.post_master_init' after upgrading to 2019.2.1 (refs: #54783)
- **PR** #54783: (garethgreenaway) Ensure metaproxy directory is included in sdist @ *2019-09-29 02:17:23 UTC*
  - 6b43fbe Merge pull request #54783 from garethgreenaway/54751_fixing_missing_metaproxy_directory
  - 67d9938 Merge branch '2019.2.1' into 54751_fixing_missing_metaproxy_directory
  - a35e609 Adding __init__.py to metaproxy directory so that metaproxy is included when running setup.py.
- **ISSUE** #54762: (margau) 2019.2.1: Breaks Minion-Master Communication (refs: #54823, #54784, #54807)

- **PR** #54784: (dhiltonp) fix dns_check to return uri-compatible ipv6 addresses, add tests @ *2019-09-28 08:36:51 UTC*
  - 7912b67 Merge pull request #54784 from dhiltonp/ipv46
  - 042a101 Merge branch '2019.2.1' into ipv46
- **PR** #54779: (frogunder) Add 2019.2.2 release notes @ *2019-09-27 17:45:46 UTC*
  - 2f94b44 Merge pull request #54779 from frogunder/releasenotes_2019.2.2
  - 67f564b Add 2019.2.2 release notes
  - ac6b54f Merge branch '2019.2.1' into ipv46
  - 93ebd09 update mock (py2) from 2.0.0 to 3.0.5
  - 37bcc4c fix dns_check to return uri-compatible ipv6 addresses, add tests
    - dd86c46 Merge pull request #1 from waynew/pull/54777-callback-typo
      - a57f7d0 Add tests
    - c19d0b0 Fix minion's remove_periodic_callback()
- **PR** #54731: (pizzapanther) Fix returners not loading properly @ *2019-09-26 17:24:27 UTC*
  - 46bec3c Merge pull request #54731 from pizzapanther/not-so-__new__-and-shiny
  - bdf24f4 Make sure we tests salt-master's *event_return* setting
  - 5499518 remove unnecessary import
  - 3f8a382 fix module import
  - 0746aa7 remove __new__ method since it was removed from parent class
- **PR** #54706: (bryceml) 2019.2.1 ruby @ *2019-09-23 16:00:27 UTC*
  - e2b86bf Merge pull request #54706 from bryceml/2019.2.1_ruby
  - 168a6c1 switch to ruby 2.6.3

28.2.8 Salt 2019.2.3 Release Notes

Version 2019.2.3 is a CVE-fix release for *2019.2.0*.

Security Fix

CVE-2019-17361

With the Salt NetAPI enabled in addition to having a SSH roster defined, unauthenticated access is possible when specifying the client as SSH. Additionally, when the raw_shell option is specified any arbitrary command may be run on the Salt master when specifying SSH options.

28.2.9 Salt 2019.2.4 Release Notes

Version 2019.2.4 is a CVE-fix release for *2019.2.0*.

Security Fix

CVE-2020-11651

An issue was discovered in SaltStack Salt before 2019.2.4 and 3000 before 3000.2. The salt-master process ClearFuncs class does not properly validate method calls. This allows a remote user to access some methods without authentication. These methods can be used to retrieve user tokens from the salt master and/or run arbitrary commands on salt minions.

CVE-2020-11652

An issue was discovered in SaltStack Salt before 2019.2.4 and 3000 before 3000.2. The salt-master process ClearFuncs class allows access to some methods that improperly sanitize paths. These methods allow arbitrary directory access to authenticated users.

Known Issue

Part of the fix for CVE-2020-11651 added better validation of the methods allowed to be called by remote clients. Both AESFuncs and ClearFuncs now have an explicit list of methods that can be called. The name of one of these whitlisted methods on AESFuncs had a typo. The _minion_runner method should be minion_runner (without the underscore prefix). This typo breaks the publish module's runner method. Calling runners, for example:

```
salt minion publish.runner manage.down
```

Will not work, and you will receive and empty reply from the salt master.

This will be addressed in the Sodium release of Salt set for mid-June 2020.

28.2.10 Salt 2019.2.5 Release Notes

Version 2019.2.5 is a bug-fix release for *2019.2.0*.

Statistics

- Total Merges: **2**
- Total Issue References: **2**
- Total PR References: **2**
- Contributors: **2** (dwoz, frogunder)

Changelog for v2019.2.4..v2019.2.5

Generated at: 2020-05-05 22:43:12 UTC

- **PR** #57096: (frogunder) Update man_pages 2019.2.5 @ *2020-05-05 22:10:46 UTC*
    - 6877b7259a Merge pull request #57096 from frogunder/man_pages_2019.2.5
    - 58ea351a59 Update man_pages 2019.2.5
- **ISSUE** #57027: (ecarson) [BUG] Master running 2019.2.4 or 3000.2 unable to synchronize files using saltutil.sync_all to 2017.7.1 minion due to CVE fix (refs: #57090)
- **ISSUE** #57016: (idontwanttosignin) [BUG] Requested method not exposed: minion_runner (refs: #57090)
- **PR** #57090: (dwoz) Address Issues in CVE Release @ *2020-05-05 22:09:25 UTC*

  - 8fe0f66f94 Merge pull request #57090 from dwoz/bugs_n_stuff
  - f3e8590bac Describe SEPs
  - aa1a9d340d Update hardening doc to mention 4505/4506
  - ca303f7c0c Add link to salt-announce to documentation
  - c63253ef9c Address issues in cve release

28.2.11 Salt 2018.3.0 Release Notes - Codename Oxygen

Warning: If you are using Jinja to dump lists or dictionaries in your SLS files, this will now cause errors in Python 2 since Jinja does not produce YAML-compatible output when strings in the data structures contain unicode types. The dictionary must be passed through a Jinja filter to produce YAML-compatible strings.

The below is an example of invalid SLS:

```
/etc/foo.conf:
  file.mangaged:
    - source: salt://foo.conf
    - template: jinja
    - defaults: {{ mydict }}
```

To make it valid, use either one of Salt's own `json` or `yaml` filters:

```
/etc/foo.conf:
  file.mangaged:
    - source: salt://foo.conf
    - template: jinja
    - defaults: {{ mydict | json }}
```

Unicode/Python 3 Compatibility Improvements

This release fixes a number of nagging issues with Unicode strings in Salt under Python 2 (ex. `'ascii' codec can't decode byte 0xd0`). For best results, use a UTF-8 locale (such as by setting the `LANG` environment variable to one which supports UTF-8. For example `en_US.UTF-8`, `de_DE.UTF-8`, `ru_RU.UTF-8`, `C.UTF-8`).

Additionally, a number of Python 3 compatibility fixes have been made, many of them having to do with file I/O and str/bytes mismatches.

We continue to work toward improving both Unicode and Python 3 compatibility and welcome any feedback.

Lots of Docker Improvements

Much Improved Support for Docker Networking

The *docker_network.present* state has undergone a full rewrite, which includes the following improvements:

Full API Support for Network Management

The improvements made to input handling in the *docker_container.running* state for 2017.7.0 have now been expanded to *docker_network.present*. This brings with it full support for all tunable configuration arguments.

Custom Subnets

Custom subnets can now be configured. Both IPv4 and mixed IPv4/IPv6 networks are supported. See *here* for more information.

Network Configuration in `docker_container.running` States

A long-requested feature has finally been added! It is now possible to configure static IPv4/IPv6 addresses, as well as links and labels. See *here* for more information.

Note: While the `containers` argument to *`docker_network.present`* will continue to be supported, it will no longer be the recommended way of ensuring that a container is attached to a network.

Improved Handling of Images from Custom Registries

Rather than attempting to parse the tag from the passed image name, Salt will now resolve that tag down to an image ID and use that ID instead.

Important: Due to this change, there are some backward-incompatible changes to image management. See below for a full list of these changes.

Backward-incompatible Changes to Docker Image Management

Passing image names to the following functions must now be done using separate `repository` and `tag` arguments:

- *`docker.build`*
- *`docker.commit`*
- *`docker.import`*
- *`docker.load`*
- *`docker.tag`*
- *`docker.sls_build`*

Additionally, the `tag` argument must now be explicitly passed to the *`docker_image.present`* state, unless the image is being pulled from a docker registry.

State and Execution Module Support for `docker run` Functionality

The *`docker_container.running`* state is good for containers which run services, but it is not as useful for cases in which the container only needs to run once. The `start` argument to *`docker_container.running`* can be set to `False` to prevent the container from being started again on a subsequent run, but for many use cases this is not sufficient. Therefore, the *`docker.run_container`* remote-execution function was added. When used on the Salt CLI, it will return information about the container, such as its name, ID, exit code, and any output it produces.

State support has also been added via the *docker_container.run* state. This state is modeled after the *cmd.run* state, and includes arguments like `onlyif`, `unless`, and `creates` to control whether or not the container is run.

Full API Support for `docker.logs`

This function now supports all of the functions that its Docker API counterpart does, allowing you to do things like include timestamps, and also suppress stdout/stderr, etc. in the return.

start Argument Added to `docker.create` Function

This removes the need to run *docker.start* separately when creating containers on the Salt CLI.

```
salt myminion docker.create image=foo/bar:baz command=/path/to/command start=True
```

Use SaltSSH Minions like regular Master-Minions

The Master process can now also call SSH minions as if they were connected to the master using ZeroMQ. By setting `enable_ssh_minions: True` in the master config file, the master will create a Salt SSH client process which connects to the minion and returns the output for the `salt` CLI to use like a regular minion. This can be used anywhere the LocalClient is used.

Exceptions Raised for Authentication/Authorization Errors

When sending `publish` commands via `master.py` and `masterapi.py` and an authorization or authentication problem is encountered, Salt will now raise the appropriate exceptions instead of returning an empty string: `''`.

The reasoning behind this change is to make it easier to debug various scenarios surrounding authentication and authorization issues more effectively.

Comparison Operators in Package Installation

Salt now supports using comparison operators (e.g. `>=1.2.3`) when installing packages on minions which use *yum/dnf* or *apt*. This is supported both in the *pkg.installed* state and in the `pkg.install` remote execution function.

Master Tops Changes

When both *Master Tops* and a *Top File* produce SLS matches for a given minion, the matches were being merged in an unpredictable manner which did not preserve ordering. This has been changed. The top file matches now execute in the expected order, followed by any master tops matches that are not matched via a top file.

To make master tops matches execute first, followed by top file matches, set the new *master_tops_first* minion config option to `True`.

Several Jinja Filters Renamed

The following Jinja filters (originally added in 2017.7.0) have been renamed due to the fact that they were inaccurately named when initially added. The original names will be supported until the 3000 release of Salt.

- `rand_str` renamed to *`random_hash`*
- `jinja_decode_dict` renamed to `jinja_encode_dict`
- `jinja_decode_list` renamed to `jinja_encode_list`

Return Codes for Runner/Wheel Functions

When using *orchestration*, runner and wheel functions used to report a `True` result if the function ran to completion without raising an exception. It is now possible to set a return code in the `__context__` dictionary, allowing runner and wheel functions to report that they failed. Here's some example pseudocode:

```
def myrunner():
    ...
    do stuff
    ...
    if some_error_condition:
        __context__['retcode'] = 1
    return result
```

Variable Update Intervals for Fileserver Backends

Prior to this release, fileservers would be updated as part of a dedicated "maintenance" process, in which various routine maintenance tasks were performed. This tied the update interval to the *`loop_interval`* config option, and also forced all fileservers to update at the same interval.

2018.3.0 adds the following configuration options for the various fileserver backends:

- *`roots_update_interval`*
- *`azurefs_update_interval`*
- *`gitfs_update_interval`*
- *`hgfs_update_interval`*
- *`minionfs_update_interval`*
- *`s3fs_update_interval`*
- *`svnfs_update_interval`*

These allow for update intervals to be set for each individual backend. The default value for each of these is 60 seconds.

In addition, for *GitFS* it is also possible to apply intervals to individual remotes. See *here* for examples.

Note: git_pillar does not yet support variable update intervals, this is targeted for the next feature release (2019.2.0).

LDAP via External Authentication Changes

In this release of Salt, if LDAP Bind Credentials are supplied, then these credentials will be used for all LDAP access except the first authentication when a job is submitted. The first authentication will use the user's credentials as passed on the CLI. This behavior is to accommodate certain two-factor authentication schemes where the authentication token can only be used once.

In previous releases the bind credentials would only be used to determine the LDAP user's existence and group membership. The user's LDAP credentials were used from then on.

Stormpath External Authentication Removed

Per Stormpath's announcement, their API will be shutting down on 8/17/2017 at noon PST so the Stormpath external authentication module has been removed.

https://stormpath.com/oktaplusstormpath

New (Proxy) Minion Configuration Options

To be able to connect the Minion to the Master using a certain source IP address or port, the following options have been added:

- *`source_interface_name`*
- *`source_address`*
- *`source_ret_port`*
- *`source_publish_port`*

`environment` config option renamed to `saltenv`

The *`environment`* config option predates referring to a salt fileserver environment as a **saltenv**. To pin a minion to a single environment for running states, one would use *`environment`*, but overriding that environment would be done with the `saltenv` argument. For consistency, *`environment`* is now simply referred to as *`saltenv`*. There are no plans to deprecate or remove *`environment`*, if used it will log a warning and its value will be used as *`saltenv`*.

`lock_saltenv` config option added

If set to `True`, this option will prevent a minion from allowing the `saltenv` argument to override the value set in *`saltenv`* when running states.

Failed Minions for State/Function Orchestration Jobs Added to Changes Dictionary

For orchestration jobs which run states (or run remote execution functions and also use a *fail function* to indicate success or failure), minions which have `False` results were previously included as a formatted string in the comment field of the return for that orchestration job. This made the failed returns difficult to *parse programatically*. The failed returns in these cases are now included in the changes dictionary, making for much easier parsing.

Grains

- `fc_wwn` - Show all fibre channel world wide port names for a host, must be enabled with *fibre_channel_grains*
- `iscsi_iqn` - Show the iSCSI IQN name for a host
- `swap_total` - Show the configured swap_total for Linux, *BSD, OS X and Solaris/SunOS
- `virtual`:
  - identifies reports KVM and VMM hypervisors when running an OpenBSD guest
  - for detecting Solaris Logical Domains (LDOMs) running on T-Series SPARC hardware. The `virtual_subtype` grain is populated as a list of domain roles.

Salt Minion Auto-discovery

Using auto-discovery, the Salt Minion now no longer needs to be configured against a specific DNS name or IP address of a Master.

For this feature Salt Master now requires port 4520 for UDP broadcast packets to be opened and the Salt Minion be able to send UDP packets to the same port.

Configuration

By default, automatic discovery is disabled.

Warning: Due to the current limitations that will be changing in a future release, before you turn on auto-discovery, make sure your network is secured and trusted.

Auto-discovery is configured on Master and Minion. Both of them are configured via the `discovery` option as follows:

Master configuration

To use the default configuration, which accepts any minion, simply set `discovery` to True:

```
discovery: true
```

A sub-option called *mapping* allows auto-discovery to help find the proper Master. The mapping contains an arbitrary set of key/value pairs, which the Minion configuration can target. By default, no mappings are set.

Example:

```
discovery:
  mapping:
    description: SES 5.0
    node: 1
```

It is also possible to change the port used from the default of `4520`, by setting a `port` option under the Master's `discovery` configuration:

```
discovery:
  port: 4567
```

Note: When using a port number other than the default, the Minion's `discovery` configuration must *also* have a port specified, otherwise the Minion will still attempt to contact the Master on port `4520`.

Minion configuration

In addition to the `mapping` and `port` options, the following additional options are available to Minions:

- `attempts` - This option specifies how many broadcast requests should be sent to the network, waiting for any Master response. Each attempt takes a couple of seconds, so raising this value may result in a slower Minion startup. Note that, on a properly-configured network, autodiscovery should succeed on the first attempt. By default, this value is set to `3`.
- `match` - This option can be set to either `all` or `any`, and it determines how the values configured in `mapping` are matched. If set to `all`, then all of the key/value pairs in the Minion's `mapping` must match a given Master. If set to `any` (the default), then any match to a key/value mapping will constitute a match.
- `pause` - The interval in seconds between attempts (default: 5).
- `fibre_channel_grains` - Enables the `fc_wwn` grain. (Default: False)
- `iscsi_grains` - Enables the `iscsi_iqn` grain. (Default: False)

Connection to a type instead of DNS

By now each Minion was connecting to a Master by DNS or IP address. From now on it is possible also to connect to a _type_ of a Master. For example, in a network there are three different Masters, each corresponds for a particular niche or environment or specific role etc. The Minion is supposed to connect only to one of those Masters that is described appropriately.

To achieve such an effect, each `/etc/salt/master` configuration should have a `discovery` option, which should have a `mapping` element with arbitrary key/value pairs. The same configuration should be on the Minion, so then when mapping matches, Minion recognises Master as its connection target.

Example for Master configuration (`/etc/salt/master`):

```
discovery:
  mapping:
    description: SES 5.0
    node: 1
```

The example above describes a system that is running a particular product, where `description` is an arbitrary key and `SES 5.0` is just a string. In order to match exactly this Master, the following configuration at Minion should be present:

```
discovery:
  match: all  # Can be "all" or "any"
  mapping:
    description: SES 5.0
    node: 1
```

Notice `match` criteria is set to `all`. This would mean that from all found Masters select only that, which `description` is set to `SES 5.0` _and_ `node` is set to `1`. All other Masters will be ignored.

Limitations

This feature has a couple of _temporary_ limitations that are subject to change in the future:

- Only one Master on the network is supported. Currently the Minion cannot select which Master out of few the same to choose. This will change to choosing the Master that is least loaded.
- Minions will accept _any_ master that matches connection criteria without any particular security applied (priv/pub key check, signature, fingerprint etc). That implies that administrator is expected to know his network and make sure it is clean.

New Modules

- *`salt.modules.purefa`*

New NaCl Renderer

A new renderer has been added for encrypted data.

New support for Cisco UCS Chassis

The salt proxy minion now allows for control of Cisco USC chassis. See the `cimc` modules for details.

New support for Cassandra v3

The `cassandra_cql` module now supports Cassandra v3 which has changed its internal schema to define keyspaces and columns.

New salt-ssh roster

A new roster has been added that allows users to pull in a list of hosts for salt-ssh targeting from a `~/.ssh` configuration. For full details, please see the `sshconfig` roster.

New GitFS Features

Two new features which affect how GitFS maps branches/tags to fileserver environments (i.e. `saltenvs`) have been added:

1. It is now possible to completely turn off Salt's default mapping logic (aside from the mapping of the `base` saltenv). This can be triggered using the new *`gitfs_disable_saltenv_mapping`* config option.

   Note: When this is disabled, only the `base` saltenv and any configured using *per-saltenv configuration parameters* will be available.

2. The types of refs which Salt will use as saltenvs can now be controlled. In previous releases, branches and tags were both mapped as environments, and individual commit SHAs could be specified as saltenvs in states (and when caching files using *`cp.cache_file`*). Using the new *`gitfs_ref_types`* config option, the types of refs which are used as saltenvs can be restricted. This makes it possible to ignore all tags and use branches only, and also to keep SHAs from being made available as saltenvs.

Additional output modes

The `state_output` parameter now supports `full_id`, `changes_id` and `terse_id`. Just like `mixed_id`, these use the state ID as name in the highstate output. For more information on these output modes, see the docs for the *`Highstate Outputter`*.

Windows

Python Version

Python 2 Windows API was design when Windows did not support Unicode. Windows now supports Unicode however to keep backwards compatibility Python 2 Windows API has not been changed. Python 3 Windows API supports Unicode. Salt Python 3 installer is the recommend choice for users who need characters other than Non-ASCII (7bit) characters.

`pkg` Execution module changes

Significant changes have been made to the *`win_pkg`* execution module. Users should test this release against their existing package sls definition files. These changes are also in 2016.11.9 & 2017.7.3.

- `pkg.list_available` no longer defaults to refreshing the winrepo meta database.
- `pkg.install` without a `version` parameter no longer upgrades software if the software is already installed. Use `pkg.install version=latest` or in a state use `pkg.latest` to get the old behavior.
- `pkg.list_pkgs` now returns multiple versions if software installed more than once.
- `pkg.list_pkgs` now returns 'Not Found' when the version is not found instead of '(value not set)' which matches the contents of the sls definitions.
- `pkg.remove()` will wait up to 3 seconds (normally about a second) to detect changes in the registry after removing software, improving reporting of version changes.
- `pkg.remove()` can remove `latest` software, if `latest` is defined in sls definition.
- Documentation was update for the execution module to match the style in new versions, some corrections as well.
- All install/remove commands are prefix with cmd.exe shell and cmdmod is called with a command line string instead of a list. Some sls files in saltstack/salt-winrepo-ng expected the commands to be prefixed with cmd.exe (i.e. the use of &).
- Some execution module functions results, now behavour more like their Unix/Linux versions.

Installer

Changes to config handling

Behavior with existing configuration has changed. With previous windows installers the existing config was used and the master and minion id could be modified via the installer. It was problematic in that it didn't account for configuration that may be defined in the `minion.d` directory. This change gives you the option via a drop-down list to use one of the following:

- Default Config: Use the config that comes with the installer
- Existing Config: Use the current config without changes

- Custom Config: Select a custom config using the file picker

The existing config option will only be available if the installer detects an existing config. If there is an existing config, and you choose `Default` or `Custom`, the existing config will be deleted, including the `minion.d` directory, and replaced by your selection.

The `Default Config` and `Custom Config` options will allow you to modify the Master and the Minion ID. `Existing Config` will leave the existing configuration unchanged.

These settings can be defined on the command line using the following switches:

- `/default-config`
- `/custom-config=C:\Path\To\Custom\Config\minion`

If neither option is passed and there is an existing config, the default is to use the existing config. If there is no existing config (new install) the default config will be used.

Multi-master configuration

The installer now has the ability to apply a multi-master configuration either from the GUI or the command line. The `master` field in the GUI can accept either a single master or a comma-separated list of masters. The command-line switch (`/master=`) can accept the same.

Command-line help

The Windows installer will now display command-line help when a help switch (`/?`) is passed.

New utils module `salt.utils.pkg.win`

A new utils module has been added, which gathers information about windows installed software. This is currently not used by any salt execution module or state at this time. Users are encouraged to run this and report any issues. Running the command with the `detail` option will be useful for anyone developing windows package definitions. With salt installed in the default location the following command will print the help message.

```
chcp 65001
c:\salt\bin\python.exe c:\salt\bin\lib\site-packages\salt\utils\pkg\win.py
c:\salt\bin\python.exe c:\salt\bin\lib\site-packages\salt\utils\pkg\win.py detail␣
↪system
```

Salt Cloud Features

OpenStack Revamp

The OpenStack Driver has been rewritten mostly from scratch. Salt is now using the *shade driver <https://docs.openstack.org/shade/latest/>*.

With this, the `nova` driver is being deprecated.

`openstack driver`

There have also been several new modules and states added for managing OpenStack setups using shade as well.

`keystone` `keystone role grant` `keystone group` `keystone role` `keystone service` `keystone user` `keystone domain` `keystone project` `keystone endpoint` `glance`

glance_image neutron neutron subnet neutron secgroup neutron secgroup rule neutron network

Pre-Flight Commands

Support has been added for specified "preflight commands" to run on a VM before the deploy script is run. These must be defined as a list in a cloud configuration file. For example:

```
my-cloud-profile:
  provider: linode-config
  image: Ubuntu 16.04 LTS
  size: Linode 2048
  preflight_cmds:
    - whoami
    - echo 'hello world!'
```

These commands will run in sequence **before** the bootstrap script is executed.

New salt-cloud Grains

When salt-cloud creates a new minion, it will now add grain information to the minion configuration file, identifying the resources originally used to create it.

The generated grain information will appear similar to:

```
grains:
  salt-cloud:
    driver: ec2
    provider: my_ec2:ec2
    profile: ec2-web
```

The generation of salt-cloud grains can be suppressed by the option `enable_cloud_grains: 'False'` in the cloud configuration file.

Upgraded Saltify Driver

The salt-cloud Saltify driver is used to provision machines which are not controlled by a dedicated cloud supervisor (such as typical hardware machines) by pushing a salt-bootstrap command to them and accepting them on the salt master. Creation of a node has been its only function and no other salt-cloud commands were implemented.

With this upgrade, it can use the salt-api to provide advanced control, such as rebooting a machine, querying it along with conventional cloud minions, and, ultimately, disconnecting it from its master.

After disconnection from ("destroying" on) one master, a machine can be re-purposed by connecting to ("creating" on) a subsequent master.

New Vagrant Driver

The salt-cloud Vagrant driver brings virtual machines running in a limited environment, such as a programmer's workstation, under salt-cloud control. This can be useful for experimentation, instruction, or testing salt configurations.

Using salt-api on the master, and a salt-minion running on the host computer, the Vagrant driver can create (`vagrant up`), restart (`vagrant reload`), and destroy (`vagrant destroy`) VMs, as controlled by salt-cloud profiles which designate a `Vagrantfile` on the host machine.

The master can be a very limited machine, such as a Raspberry Pi, or a small VagrantBox VM.

Python PyWinRM Module

Versions of `pywinrm>=0.2.1` are finally able to disable validation of self signed certificates. *Here* for more information.

DigitalOcean

The DigitalOcean driver has been renamed to conform to the company name. The new driver name is `digitalocean`. The old name `digital_ocean` and a short one `do` will still be supported through virtual aliases, this is mostly cosmetic.

Azure Cloud

The azure sdk used for the `azurearm` cloud driver now depends on `azure-cli>=2.0.12`

New `saltclass` pillar/master_tops modules

This module clones the behaviour of reclass (http://reclass.pantsfullofunix.net/), without the need of an external app, and add several features to improve flexibility. Saltclass lets you define your nodes from simple `yaml` files (`.yml`) through hierarchical class inheritance with the possibility to override pillars down the tree.

Features

- Define your nodes through hierarchical class inheritance
- Reuse your reclass data with minimal modifications
  - applications => states
  - parameters => pillars
- Use Jinja templating in your yaml definitions
- Access to the following Salt objects in Jinja
  - `__opts__`
  - `__salt__`
  - `__grains__`
  - `__pillars__`
  - `minion_id`
- Chose how to merge or override your lists using ^ character (see examples)
- Expand variables ${} with possibility to escape them if needed ${} (see examples)
- Ignores missing node/class and will simply return empty without breaking the pillar module completely - will be logged

An example subset of data is available here: http://git.mauras.ch/salt/saltclass/src/master/examples

Terms usable in yaml files	Description
classes	A list of classes that will be processed in order
states	A list of states that will be returned by master_tops function
pillars	A yaml dictionary that will be returned by the ext_pillar function
environment	Node saltenv that will be used by master_tops

A class consists of:

- zero or more parent classes
- zero or more states
- any number of pillars

A child class can override pillars from a parent class. A node definition is a class in itself with an added `environment` parameter for `saltenv` definition.

Class Names

Class names mimic salt way of defining states and pillar files. This means that `default.users` class name will correspond to one of these:

- `<saltclass_path>/classes/default/users.yml`
- `<saltclass_path>/classes/default/users/init.yml`

Saltclass Tree

A saltclass tree would look like this:

```
<saltclass_path>
├── classes
│   ├── app
│   │   ├── borgbackup.yml
│   │   └── ssh
│   │       └── server.yml
│   ├── default
│   │   ├── init.yml
│   │   ├── motd.yml
│   │   └── users.yml
│   ├── roles
│   │   ├── app.yml
│   │   └── nginx
│   │       ├── init.yml
│   │       └── server.yml
│   └── subsidiaries
│       ├── gnv.yml
│       ├── qls.yml
│       └── zrh.yml
└── nodes
    ├── geneva
    │   └── gnv.node1.yml
    ├── lausanne
    │   ├── qls.node1.yml
    │   └── qls.node2.yml
    │
```

(continues on next page)

(continued from previous page)

```
    ├── node127.yml
    └── zurich
        ├── zrh.node1.yml
        ├── zrh.node2.yml
        └── zrh.node3.yml
```

Examples

`<saltclass_path>/nodes/lausanne/qls.node1.yml`

```
environment: base

classes:
{% for class in ['default'] %}
  - {{ class }}
{% endfor %}
  - subsidiaries.{{ __grains__['id'].split('.')[0] }}
```

`<saltclass_path>/classes/default/init.yml`

```
classes:
  - default.users
  - default.motd

states:
  - openssh

pillars:
  default:
    network:
      dns:
        srv1: 192.168.0.1
        srv2: 192.168.0.2
        domain: example.com
    ntp:
      srv1: 192.168.10.10
      srv2: 192.168.10.20
```

`<saltclass_path>/classes/subsidiaries/gnv.yml`

```
pillars:
  default:
    network:
      sub: Geneva
      dns:
        srv1: 10.20.0.1
        srv2: 10.20.0.2
        srv3: 192.168.1.1
        domain: gnv.example.com
    users:
      adm1:
        uid: 1210
        gid: 1210
        gecos: 'Super user admin1'
        homedir: /srv/app/adm1
      adm3:
```

(continues on next page)

(continued from previous page)

```
        uid: 1203
        gid: 1203
        gecos: 'Super user adm
```

Variable expansions:

Escaped variables are rendered as is - `${test}`

Missing variables are rendered as is - `${net:dns:srv2}`

```
pillars:
  app:
  config:
    dns:
      srv1: ${default:network:dns:srv1}
      srv2: ${net:dns:srv2}
    uri: https://application.domain/call?\${test}
    prod_parameters:
      - p1
      - p2
      - p3
  pkg:
    - app-core
    - app-backend
```

List override:

Not using ^ as the first entry will simply merge the lists

```
pillars:
  app:
    pkg:
      - ^
      - app-frontend
```

Known limitation

Currently you can't have both a variable and an escaped variable in the same string as the escaped one will not be correctly rendered - '${xx}' will stay as is instead of being rendered as '${xx}'

Lists of comments in state returns

State functions can now return a list of strings for the `comment` field, as opposed to only a single string. This is meant to ease writing states with multiple or multi-part comments.

Beacon configuration changes

In order to remain consistent and to align with other Salt components such as states, support for configuring beacons using dictionary based configuration has been deprecated in favor of list based configuration. All beacons have a validation function which will check the configuration for the correct format and only load if the validation passes.

avahi_announce

Old behavior:

```
beacons:
  avahi_announce:
    run_once: True
    servicetype: _demo._tcp
    port: 1234
    txt:
      ProdName: grains.productname
      SerialNo: grains.serialnumber
      Comments: 'this is a test'
```

New behavior:

```
beacons:
  avahi_announce:
    - run_once: True
    - servicetype: _demo._tcp
    - port: 1234
    - txt:
        ProdName: grains.productname
        SerialNo: grains.serialnumber
        Comments: 'this is a test'
```

bonjour_announce

Old behavior:

```
beacons:
  bonjour_announce:
    run_once: True
    servicetype: _demo._tcp
    port: 1234
    txt:
      ProdName: grains.productname
      SerialNo: grains.serialnumber
      Comments: 'this is a test'
```

New behavior:

```
beacons:
  bonjour_announce:
    - run_once: True
    - servicetype: _demo._tcp
    - port: 1234
    - txt:
        ProdName: grains.productname
        SerialNo: grains.serialnumber
        Comments: 'this is a test'
```

btmp

Old behavior:

```
beacons:
  btmp: {}
```

New behavior:

```
beacons:
  btmp: []
```

glxinfo

Old behavior:

```
beacons:
  glxinfo:
    user: frank
    screen_event: True
```

New behavior:

```
beacons:
  glxinfo:
    - user: frank
    - screen_event: True
```

haproxy

Old behavior:

```
beacons:
    haproxy:
        - www-backend:
            threshold: 45
            servers:
                - web1
                - web2
        - interval: 120
```

New behavior:

```
beacons:
  haproxy:
    - backends:
        www-backend:
          threshold: 45
          servers:
            - web1
            - web2
    - interval: 120
```

inotify

Old behavior:

```
beacons:
  inotify:
```

(continues on next page)

(continued from previous page)

```
    /path/to/file/or/dir:
        mask:
          - open
          - create
          - close_write
        recurse: True
        auto_add: True
        exclude:
          - /path/to/file/or/dir/exclude1
          - /path/to/file/or/dir/exclude2
          - /path/to/file/or/dir/regex[a-m]*$:
        regex: True
    coalesce: True
```

New behavior:

```
beacons:
  inotify:
    - files:
        /path/to/file/or/dir:
          mask:
            - open
            - create
            - close_write
          recurse: True
          auto_add: True
          exclude:
            - /path/to/file/or/dir/exclude1
            - /path/to/file/or/dir/exclude2
            - /path/to/file/or/dir/regex[a-m]*$:
          regex: True
    - coalesce: True
```

journald

Old behavior:

```
beacons:
  journald:
    sshd:
      SYSLOG_IDENTIFIER: sshd
      PRIORITY: 6
```

New behavior:

```
beacons:
  journald:
    - services:
        sshd:
          SYSLOG_IDENTIFIER: sshd
          PRIORITY: 6
```

load

Old behavior:

```
beacons:
  load:
    1m:
      - 0.0
      - 2.0
    5m:
      - 0.0
      - 1.5
    15m:
      - 0.1
      - 1.0
    emitatstartup: True
    onchangeonly: False
```

New behavior:

```
beacons:
  load:
    - averages:
        1m:
          - 0.0
          - 2.0
        5m:
          - 0.0
          - 1.5
        15m:
          - 0.1
          - 1.0
    - emitatstartup: True
    - onchangeonly: False
```

log

Old behavior:

```
beacons:
    log:
      file: <path>
      <tag>:
        regex: <pattern>
```

New behavior:

```
beacons:
    log:
      - file: <path>
      - tags:
          <tag>:
            regex: <pattern>
```

network_info

Old behavior:

```
beacons:
  network_info:
    - eth0:
        type: equal
        bytes_sent: 100000
        bytes_recv: 100000
        packets_sent: 100000
        packets_recv: 100000
        errin: 100
        errout: 100
        dropin: 100
        dropout: 100
```

New behavior:

```
beacons:
  network_info:
    - interfaces:
        eth0:
          type: equal
          bytes_sent: 100000
          bytes_recv: 100000
          packets_sent: 100000
          packets_recv: 100000
          errin: 100
          errout: 100
          dropin: 100
          dropout: 100
```

network_settings

Old behavior:

```
beacons:
  network_settings:
    eth0:
      ipaddr:
      promiscuity:
        onvalue: 1
    eth1:
      linkmode:
```

New behavior:

```
beacons:
  network_settings:
    - interfaces:
        - eth0:
            ipaddr:
            promiscuity:
              onvalue: 1
```

(continues on next page)

(continued from previous page)

```
        - eth1:
            linkmode:
```

proxy_example

Old behavior:

```
beacons:
  proxy_example:
    endpoint: beacon
```

New behavior:

```
beacons:
  proxy_example:
    - endpoint: beacon
```

ps

Old behavior:

```
beacons:
  ps:
    - salt-master: running
    - mysql: stopped
```

New behavior:

```
beacons:
  ps:
    - processes:
        salt-master: running
        mysql: stopped
```

salt_proxy

Old behavior:

```
beacons:
  salt_proxy:
    - p8000: {}
    - p8001: {}
```

New behavior:

```
beacons:
  salt_proxy:
    - proxies:
        p8000: {}
        p8001: {}
```

sensehat

Old behavior:

```
beacons:
  sensehat:
    humidity: 70%
    temperature: [20, 40]
    temperature_from_pressure: 40
    pressure: 1500
```

New behavior:

```
beacons:
  sensehat:
    - sensors:
        humidity: 70%
        temperature: [20, 40]
        temperature_from_pressure: 40
        pressure: 1500
```

service

Old behavior:

```
beacons:
  service:
    salt-master:
    mysql:
```

New behavior:

```
beacons:
  service:
    - services:
        nginx:
            onchangeonly: True
            delay: 30
            uncleanshutdown: /run/nginx.pid
```

sh

Old behavior:

```
beacons:
  sh: {}
```

New behavior:

```
beacons:
  sh: []
```

status

Old behavior:

```
beacons:
  status: {}
```

New behavior:

```
beacons:
  status: []
```

telegram_bot_msg

Old behavior:

```
beacons:
  telegram_bot_msg:
    token: "<bot access token>"
    accept_from:
      - "<valid username>"
    interval: 10
```

New behavior:

```
beacons:
  telegram_bot_msg:
    - token: "<bot access token>"
    - accept_from:
      - "<valid username>"
    - interval: 10
```

twilio_txt_msg

Old behavior:

```
beacons:
  twilio_txt_msg:
    account_sid: "<account sid>"
    auth_token: "<auth token>"
    twilio_number: "+15555555555"
    interval: 10
```

New behavior:

```
beacons:
  twilio_txt_msg:
    - account_sid: "<account sid>"
    - auth_token: "<auth token>"
    - twilio_number: "+15555555555"
    - interval: 10
```

wtmp

Old behavior:

```
beacons:
  wtmp: {}
```

New behavior:

```
beacons:
  wtmp: []
```

New requisites available in state compiler

require_any

The use of `require_any` demands that one of the required states executes before the dependent state. The state containing the `require_any` requisite is defined as the dependent state. The states specified in the `require_any` statement are defined as the required states. If at least one of the required state's execution succeeds, the dependent state will then execute. If all of the executions by the required states fail, the dependent state will not execute.

watch_any

The state containing the `watch_any` requisite is defined as the watching state. The states specified in the `watch_any` statement are defined as the watched states. When the watched states execute, they will return a dictionary containing a key named "changes".

If the "result" of any of the watched states is `True`, the watching state *will execute normally*, and if all of them are `False`, the watching state will never run. This part of `watch` mirrors the functionality of the `require` requisite.

If the "result" of any of the watched states is `True` *and* the "changes" key contains a populated dictionary (changes occurred in the watched state), then the `watch` requisite can add additional behavior. This additional behavior is defined by the `mod_watch` function within the watching state module. If the `mod_watch` function exists in the watching state module, it will be called *in addition to* the normal watching state. The return data from the `mod_watch` function is what will be returned to the master in this case; the return data from the main watching function is discarded.

If the "changes" key contains an empty dictionary, the `watch` requisite acts exactly like the `require` requisite (the watching state will execute if "result" is `True`, and fail if "result" is `False` in the watched state).

onchanges_any

The `onchanges_any` requisite makes a state only apply one of the required states generates changes, and if one of the watched state's "result" is `True`. This can be a useful way to execute a post hook after changing aspects of a system.

onfail_any

The `onfail_any` requisite allows for reactions to happen strictly as a response to the failure of at least one other state. This can be used in a number of ways, such as executing a second attempt to set up a service or begin to execute a separate thread of states because of a failure.

The `onfail_any` requisite is applied in the same way as `require_any` and `watch_any`.

Basic Slots support in state compiler

Slots extend the state syntax and allows you to do things right before the state function is executed. So you can make a decision in the last moment right before a state is executed.

Slot syntax looks close to the simple python function call. Here is a simple example:

```
copy-some-file:
  file.copy:
    - name: __slot__:salt:test.echo(text=/tmp/some_file)
    - source: __slot__:salt:test.echo(/etc/hosts)
```

Read more *here*.

Cryptographic layer changes

M2Crypto is coming back. We are making the crypto backend modular but in this release M2Crypto is enabled if it's importable by Python. If not Cryptodome or PyCrypto is used as it was in the previous releases. M2Crypto is used in the same way as PyCrypto so there would be no compatibility issues, different nodes could use different backends.

NaCL Module and Runner changes

In addition to argument changes in both the NaCL module and runner for future removal in the 3000 release, the default "box_type" has changed from `secretbox` to `sealedbox`. SecretBox is data encrypted using private key `sk` and Sealedbox is encrypted using public key `pk`.

`utils` functions reorganized into separate modules

The Salt utility functions from `salt.utils` (typically used by those developing extension modules for Salt) have been moved into different modules, grouped logically based on their functionality. The old function names will continue to work until the `3000` release of Salt (due around Q1 2019).

The renamed functions are:

- `salt.utils.appendproctitle`: use `salt.utils.process.appendproctitle` instead.
- `salt.utils.daemonize`: use `salt.utils.process.daemonize` instead.
- `salt.utils.daemonize_if`: use `salt.utils.process.daemonize_if` instead.
- `salt.utils.reinit_crypto`: use `salt.utils.crypt.reinit_crypto` instead.
- `salt.utils.pem_finger`: use `salt.utils.crypt.pem_finger` instead.
- `salt.utils.to_bytes`: use `salt.utils.stringutils.to_bytes` instead.
- `salt.utils.to_str`: use `salt.utils.stringutils.to_str` instead.
- `salt.utils.to_unicode`: use `salt.utils.stringutils.to_unicode` instead.
- `salt.utils.str_to_num`: use `salt.utils.stringutils.to_num` instead.
- `salt.utils.is_quoted`: use `salt.utils.stringutils.is_quoted` instead.
- `salt.utils.dequote`: use `salt.utils.stringutils.dequote` instead.
- `salt.utils.is_hex`: use `salt.utils.stringutils.is_hex` instead.

- `salt.utils.is_bin_str`: use `salt.utils.stringutils.is_bin_str` instead.
- `salt.utils.rand_string`: use `salt.utils.stringutils.random` instead.
- `salt.utils.contains_whitespace`: use `salt.utils.stringutils.contains_whitespace` instead.
- `salt.utils.build_whitespace_split_regex`: use `salt.utils.stringutils.build_whitespace_split_regex` instead.
- `salt.utils.expr_match`: use `salt.utils.stringutils.expr_match` instead.
- `salt.utils.check_whitelist_blacklist`: use `salt.utils.stringutils.check_whitelist_blacklist` instead.
- `salt.utils.check_include_exclude`: use `salt.utils.stringutils.check_include_exclude` instead.
- `salt.utils.print_cli`: use `salt.utils.stringutils.print_cli` instead.
- `salt.utils.clean_kwargs`: use `salt.utils.args.clean_kwargs` instead.
- `salt.utils.invalid_kwargs`: use `salt.utils.args.invalid_kwargs` instead.
- `salt.utils.shlex_split`: use `salt.utils.args.shlex_split` instead.
- `salt.utils.arg_lookup`: use `salt.utils.args.arg_lookup` instead.
- `salt.utils.argspec_report`: use `salt.utils.args.argspec_report` instead.
- `salt.utils.split_input`: use `salt.utils.args.split_input` instead.
- `salt.utils.test_mode`: use `salt.utils.args.test_mode` instead.
- `salt.utils.format_call`: use `salt.utils.args.format_call` instead.
- `salt.utils.which`: use `salt.utils.path.which` instead.
- `salt.utils.which_bin`: use `salt.utils.path.which_bin` instead.
- `salt.utils.path_join`: use `salt.utils.path.join` instead.
- `salt.utils.check_or_die`: use `salt.utils.path.check_or_die` instead.
- `salt.utils.sanitize_win_path_string`: use `salt.utils.path.sanitize_win_path` instead.
- `salt.utils.rand_str`: use `salt.utils.hashutils.random_hash` instead.
- `salt.utils.get_hash`: use `salt.utils.hashutils.get_hash` instead.
- `salt.utils.is_windows`: use `salt.utils.platform.is_windows` instead.
- `salt.utils.is_proxy`: use `salt.utils.platform.is_proxy` instead.
- `salt.utils.is_linux`: use `salt.utils.platform.is_linux` instead.
- `salt.utils.is_darwin`: use `salt.utils.platform.is_darwin` instead.
- `salt.utils.is_sunos`: use `salt.utils.platform.is_sunos` instead.
- `salt.utils.is_smartos`: use `salt.utils.platform.is_smartos` instead.
- `salt.utils.is_smartos_globalzone`: use `salt.utils.platform.is_smartos_globalzone` instead.
- `salt.utils.is_smartos_zone`: use `salt.utils.platform.is_smartos_zone` instead.
- `salt.utils.is_freebsd`: use `salt.utils.platform.is_freebsd` instead.

- `salt.utils.is_netbsd`: use `salt.utils.platform.is_netbsd` instead.
- `salt.utils.is_openbsd`: use `salt.utils.platform.is_openbsd` instead.
- `salt.utils.is_aix`: use `salt.utils.platform.is_aix` instead.
- `salt.utils.safe_rm`: use `salt.utils.files.safe_rm` instead.
- `salt.utils.is_empty`: use `salt.utils.files.is_empty` instead.
- `salt.utils.fopen`: use `salt.utils.files.fopen` instead.
- `salt.utils.flopen`: use `salt.utils.files.flopen` instead.
- `salt.utils.fpopen`: use `salt.utils.files.fpopen` instead.
- `salt.utils.rm_rf`: use `salt.utils.files.rm_rf` instead.
- `salt.utils.mkstemp`: use `salt.utils.files.mkstemp` instead.
- `salt.utils.istextfile`: use `salt.utils.files.is_text_file` instead.
- `salt.utils.is_bin_file`: use `salt.utils.files.is_binary` instead.
- `salt.utils.list_files`: use `salt.utils.files.list_files` instead.
- `salt.utils.safe_walk`: use `salt.utils.files.safe_walk` instead.
- `salt.utils.st_mode_to_octal`: use `salt.utils.files.st_mode_to_octal` instead.
- `salt.utils.normalize_mode`: use `salt.utils.files.normalize_mode` instead.
- `salt.utils.human_size_to_bytes`: use `salt.utils.files.human_size_to_bytes` instead.
- `salt.utils.backup_minion`: use `salt.utils.files.backup_minion` instead.
- `salt.utils.str_version_to_evr`: use `salt.utils.pkg.rpm.version_to_evr` instead.
- `salt.utils.parse_docstring`: use `salt.utils.doc.parse_docstring` instead.
- `salt.utils.compare_versions`: use `salt.utils.versions.compare` instead.
- `salt.utils.version_cmp`: use `salt.utils.versions.version_cmp` instead.
- `salt.utils.warn_until`: use `salt.utils.versions.warn_until` instead.
- `salt.utils.kwargs_warn_until`: use `salt.utils.versions.kwargs_warn_until` instead.
- `salt.utils.get_color_theme`: use `salt.utils.color.get_color_theme` instead.
- `salt.utils.get_colors`: use `salt.utils.color.get_colors` instead.
- `salt.utils.gen_state_tag`: use `salt.utils.state.gen_tag` instead.
- `salt.utils.search_onfail_requisites`: use `salt.utils.state.search_onfail_requisites` instead.
- `salt.utils.check_state_result`: use `salt.utils.state.check_result` instead.
- `salt.utils.get_user`: use `salt.utils.user.get_user` instead.
- `salt.utils.get_uid`: use `salt.utils.user.get_uid` instead.
- `salt.utils.get_specific_user`: use `salt.utils.user.get_specific_user` instead.
- `salt.utils.chugid`: use `salt.utils.user.chugid` instead.
- `salt.utils.chugid_and_umask`: use `salt.utils.user.chugid_and_umask` instead.
- `salt.utils.get_default_group`: use `salt.utils.user.get_default_group` instead.

- `salt.utils.get_group_list`: use `salt.utils.user.get_group_list` instead.
- `salt.utils.get_group_dict`: use `salt.utils.user.get_group_dict` instead.
- `salt.utils.get_gid_list`: use `salt.utils.user.get_gid_list` instead.
- `salt.utils.get_gid`: use `salt.utils.user.get_gid` instead.
- `salt.utils.enable_ctrl_logoff_handler`: use `salt.utils.win_functions.enable_ctrl_logoff_handler` instead.
- `salt.utils.traverse_dict`: use `salt.utils.data.traverse_dict` instead.
- `salt.utils.traverse_dict_and_list`: use `salt.utils.data.traverse_dict_and_list` instead.
- `salt.utils.filter_by`: use `salt.utils.data.filter_by` instead.
- `salt.utils.subdict_match`: use `salt.utils.data.subdict_match` instead.
- `salt.utils.substr_in_list`: use `salt.utils.data.substr_in_list` instead.
- `salt.utils.is_dictlist`: use `salt.utils.data.is_dictlist`.
- `salt.utils.repack_dictlist`: use `salt.utils.data.repack_dictlist` instead.
- `salt.utils.compare_dicts`: use `salt.utils.data.compare_dicts` instead.
- `salt.utils.compare_lists`: use `salt.utils.data.compare_lists` instead.
- `salt.utils.decode_dict`: use `salt.utils.data.encode_dict` instead.
- `salt.utils.decode_list`: use `salt.utils.data.encode_list` instead.
- `salt.utils.exactly_n`: use `salt.utils.data.exactly_n` instead.
- `salt.utils.exactly_one`: use `salt.utils.data.exactly_one` instead.
- `salt.utils.is_list`: use `salt.utils.data.is_list` instead.
- `salt.utils.is_iter`: use `salt.utils.data.is_iter` instead.
- `salt.utils.isorted`: use `salt.utils.data.sorted_ignorecase` instead.
- `salt.utils.is_true`: use `salt.utils.data.is_true` instead.
- `salt.utils.mysql_to_dict`: use `salt.utils.data.mysql_to_dict` instead.
- `salt.utils.simple_types_filter`: use `salt.utils.data.simple_types_filter` instead.
- `salt.utils.ip_bracket`: use `salt.utils.zeromq.ip_bracket` instead.
- `salt.utils.gen_mac`: use `salt.utils.network.gen_mac` instead.
- `salt.utils.mac_str_to_bytes`: use `salt.utils.network.mac_str_to_bytes` instead.
- `salt.utils.refresh_dns`: use `salt.utils.network.refresh_dns` instead.
- `salt.utils.dns_check`: use `salt.utils.network.dns_check` instead.
- `salt.utils.get_context`: use `salt.utils.stringutils.get_context` instead.
- `salt.utils.get_master_key`: use `salt.utils.master.get_master_key` instead.
- `salt.utils.get_values_of_matching_keys`: use `salt.utils.master.get_values_of_matching_keys` instead.
- `salt.utils.date_cast`: use `salt.utils.dateutils.date_cast` instead.
- `salt.utils.date_format`: use `salt.utils.dateutils.strftime` instead.

- `salt.utils.total_seconds`: use `salt.utils.dateutils.total_seconds` instead.
- `salt.utils.find_json`: use `salt.utils.json.find_json` instead.
- `salt.utils.import_json`: use `salt.utils.json.import_json` instead.
- `salt.utils.namespaced_function`: use `salt.utils.functools.namespaced_function` instead.
- `salt.utils.alias_function`: use `salt.utils.functools.alias_function` instead.
- `salt.utils.profile_func`: use `salt.utils.profile.profile_func` instead.
- `salt.utils.activate_profile`: use `salt.utils.profile.activate_profile` instead.
- `salt.utils.output_profile`: use `salt.utils.profile.output_profile` instead.

Deprecations

Configuration Option Deprecations

- The `requests_lib` configuration option has been removed. Please use `backend` instead.

Profitbricks Cloud Updated Dependency

The minimum version of the `profitbrick` python package for the `profitbricks` cloud driver has changed from 3.0.0 to 3.1.0.

Execution Module Deprecations

- The `blockdev` execution module has been removed. Its functions were merged with the *`disk`* module.
- The *`lxc`* execution module has been changed as follows:
  - The `dnsservers` option to *`lxc.cloud_init_interface`* no longer defaults to `4.4.4.4` and `8.8.8.8`.
  - The `dns_via_dhcp` option to *`lxc.cloud_init_interface`* now defaults to `True` instead of `False`.
- The *`win_psget`* module has been changed as follows:
  - The `psget.psversion` function was removed. Please use *`cmd.shell_info`* instead.
- The *`win_service`* module (which provides the `service` module on Windows platforms) has been changed as follows:
  - The `config` function was removed. Please use *`service.modify`* instead.
  - The following arguments to the *`service.create`* function have been renamed:
    - `binpath` has been renamed to `bin_path`
    - `depend` has been renamed to `dependencies`
    - `DisplayName` has been renamed to `display_name`
    - `error` has been renamed to `error_control`
    - `group` has been renamed to `load_order_group`

    - `obj` has been renamed to `account_name`
    - `password` has been renamed to `account_password`
    - `start` has been renamed to `start_type`
    - `type` has been renamed to `service_type`
- The *nacl* module has been changed as follows:
  - The following arguments have been renamed in the *nacl.keygen*, *nacl.enc*, and *nacl.dec*:
    - `key_file` has been renamed to `sk_file`
    - `key` has been renamed to `sk`

Runner Deprecations

- The *manage* runner has been changed as follows:
  - `root_user` argument was removed from the *manage.bootstrap* function. Please use `salt-ssh` roster entries for the host instead.
- The *nacl* runner has been changed as follows:
  - The following arguments have been renamed in the *nacl.keygen*, *nacl.enc*, and *nacl.dec*:
    - `key_file` has been renamed to `sk_file`
    - `key` has been renamed to `sk`

State Deprecations

- In the `archive` state, the `tar_options` and `zip_options` options were removed. Please use `options` instead.
- The `cmd` state had the following changes:
  - The `user` and `group` options were removed from the following functions (please use `runas` instead):
    - *cmd.run*
    - *cmd.script*
    - `cmd.wait`
    - `cmd.wait_script`
- In the *file* states, the `show_diff` option was removed in all states where it was previously supported. Please use `show_changes` instead.

Grain Deprecations

- For `smartos`, some grains have been deprecated. These grains will be removed in 3000:
  - The `hypervisor_uuid` grain has been replaced with `mdata:sdc:server_uuid`
  - The `datacenter` grain has been replaced with `mdata:sdc:datacenter_name`

Pillar Deprecations

The legacy configuration for `git_pillar` has been removed. Please use the new configuration for `git_pillar` which was added in 2015.8.0, which is documented *here*.

Utils Deprecations

- In `salt.utils.cloud`, the `fire_event` function now requires a `sock_dir` argument. It was previously optional.

Other Miscellaneous Deprecations

- In `version.py`, the `rc_info` function was removed. Please use `pre_info` instead.
- Warnings for moving away from the `env` option were removed. `saltenv` should be used instead. The removal of these warnings does not have a behavior change. Only the warning text was removed.

Minion Blackout

During a blackout, minions will not execute any remote execution commands, except for *`saltutil.refresh_pillar`*. Previously, support was added so that blackouts are enabled using a special pillar key, `minion_blackout` set to `True` and an optional pillar key `minion_blackout_whitelist` to specify additional functions that are permitted during blackout. This release adds support for using this feature in the grains as well, by using special grains keys `minion_blackout` and `minion_blackout_whitelist`.

Sentry Log Handler

Configuring sentry raven python client via `project`, `servers`, `public_key and ``secret_key` is deprecated and won't work with sentry clients > 3.0. Instead, the `dsn` config param must be used.

RAET transport

We haven't been doing development on RAET for quite some time and decided that 2018.3.0 is the time to announce the deprecation. RAET support will be removed in 3000. Please consider to move to `zeromq` or `tcp` transport instead of `raet`.

28.2.12 Salt 2018.3.1 Release Notes

Version 2018.3.1 is a bugfix release for *2018.3.0*.

Statistics

- Total Merges: **525**
- Total Issue References: **74**
- Total PR References: **255**

- Contributors: 55 (Ch3LL, DmitryKuzmenko, Giandom, Kimol, L4rS6, LukeCarrier, OrlandoArcapix, TamCore, The-Loeki, UtahDave, aesposito91, bbinet, bdrung, boltronics, bosatsu, clan, corywright, damon-atkins, dincamihai, dmurphy18, dnABic, douglasjreynolds, dwoz, edgan, ejparker12, esell, ezh, femnad, folti, garethgreenaway, gtmanfred, isbm, jasperla, johnj, mateiw, mcalmer, mirceaulinic, morganwillcock, opdude, pcn, pruiz, psagers, psyer, rallytime, robinro, s0undt3ch, samodid, shengis, skjaro, tankywoo, terminalmage, twangboy, vutny, yannj-fr, zmedico)

Warning: If you are using Jinja to dump lists or dictionaries in your SLS files, this will now cause errors in Python 2 since Jinja does not produce YAML-compatible output when strings in the data structures contain unicode types. The dictionary must be passed through a Jinja filter to produce YAML-compatible strings.

The below is an example of invalid SLS:

```
/etc/foo.conf:
  file.mangaged:
    - source: salt://foo.conf
    - template: jinja
    - defaults: {{ mydict }}
```

To make it valid, use either one of Salt's own `json` or `yaml` filters:

```
/etc/foo.conf:
  file.mangaged:
    - source: salt://foo.conf
    - template: jinja
    - defaults: {{ mydict | json }}
```

Tornado 5.0 Support for Python 2 Only

Tornado 5.0 moves to using asyncio for all python3 versions. Because of this and changes in asyncio between python 3.4 and 3.5 to only be able to use one ioloop, which requires some rearchitecting, support for tornado 5.0 and python3 versions of salt has been delayed to a later release.

For now, to use tornado 5.0, the python 2 version of salt must be used.

Changes to Slack Engine pillars

When using `groups_pillar_name` for the slack engine, the engine should be used as part of a salt-minion process running on the master. This will allow the minion to have pillars assigned to it, and will still allow the engine to create a LocalClient connection to the master ipc sockets to control environments.

Changes to Automatically Updating the Roster File

In `2018.3.0` salt-ssh was configured to automatically update the flat roster file if a minion was not found for salt-ssh. This was decided to be undesireable as a default. The `--skip-roster` flag has been removed and replaced with `--update-roster`, which will enable salt-ssh to add minions to the flat roster file. This behavior can also be enabled by setting `ssh_update_roster: True` in the master config file.

Changes to `file.blockreplace` State

The `append_newline` argument was added to this state. Additionally, to improve idempotence, if the string represented by `marker_end` is found in the middle of the line, the content preceding the marker will be removed when the block is replaced. This allows one to remove `append_newline: False` from the SLS and have the block properly replaced if the end of the content block is immediately followed by the `marker_end` (i.e. no newline before the marker).

Note: This will require changes to your SLS if your `marker_end` does not include the very beginning of the content you want to keep.

See the *`file.blockreplace`* state documentation for further information.

Changelog for v2018.3.0..v2018.3.1

Generated at: 2018-06-06 17:43:01 UTC

- **ISSUE** #47955: (frogunder) 2018.3.1 Creating Windows machine in Amazon using salt-cloud fails. (refs: #47989)
- **PR** #47998: (rallytime) Back-port #47989 to 2018.3.1 @ *2018-06-06 17:08:04 UTC*
  - **PR** #47989: (dwoz) Properly decode password from aws using m2crypto (refs: #47998)
  - 605463ca0d Merge pull request #47998 from rallytime/bp-47989
  - 1b7e9ac2d3 Lint fixes
  - 0545152ddd Properly decode password from aws using m2crypto
- **PR** #47965: (Ch3LL) Add PR 47924 from 2018.3 branch @ *2018-06-06 13:54:09 UTC*
  - dbc798ac68 Merge pull request #47965 from Ch3LL/gitpy_mac_3.1
  - bf608abd44 Catch all exceptions in git import for salt.utils.gitfs
- **PR** #47973: (terminalmage) salt.modules.testinframod: fix TypeError invoking types.FunctionType @ *2018-06-06 13:53:46 UTC*
  - 864d640633 Merge pull request #47973 from terminalmage/fix-testinfra
  - 4518c89484 Lint: Remove unused six import
  - c6816b2149 salt.modules.testinframod: fix TypeError invoking types.FunctionType
- **ISSUE** #47236: (MorphBonehunter) x509.private_key_managed broken after upgrade to 2018.3.0 (refs: #47957)
- **PR** #47967: (rallytime) Back-port #47957 to 2018.3.1 @ *2018-06-06 13:53:28 UTC*
  - **PR** #47957: (garethgreenaway) [2018.8] Ensure x509 passphrase is a string (refs: #47967)
  - 5ddcfff420 Merge pull request #47967 from rallytime/bp-47957
  - 9a55579af1 removing unnecessary change
  - 329b2e5956 Ensuring that when a passphrase is passed in, it is returned as a string from the passphrase callback.
- **PR** #47902: (Ch3LL) Remove In Progress for 2018.3.1 Release Notes @ *2018-05-30 18:26:49 UTC*
  - 9c964fdbce Merge pull request #47902 from Ch3LL/rn_in_progress
  - f560a151cd Remove In Progress for 2018.3.1 Release Notes
- **PR** #47897: (Ch3LL) Add changelog to 2018.3.1 release notes @ *2018-05-30 15:04:42 UTC*
  - ea7b4fdc08 Merge pull request #47897 from Ch3LL/rn_2018
  - e27ee273a7 Add == line to changelog line for release notes

  - 61e56d275d Add changelog to 2018.3.1 release notes
- **ISSUE** #47784: (jpsv) win_lgpo.py line 5368; AttributeError: 'OrderedDict' object has no attribute 'lower' (refs: #47848)
- **PR** #47848: (twangboy) Fix some major issues with the LGPO module @ *2018-05-30 13:37:32 UTC*
  - f15e636d5e Merge pull request #47848 from twangboy/fix_47784
  - 98facf8dc8 Remove log.debug statement in __virtual__
  - f037fa4064 Fix some major issues with the LGPO module
- **PR** #47881: (gtmanfred) quote python_version in requirements.txt @ *2018-05-29 21:12:05 UTC*
  - 92b8c4c08e Merge pull request #47881 from gtmanfred/2018.3.1
  - 3d874b5529 quote python_version in requirements.txt
- **PR** #47874: (gtmanfred) Tornado 5.0 is only supported on python 2 for now @ *2018-05-29 19:45:44 UTC*
  - 705bf8172d Merge pull request #47874 from gtmanfred/2018.3.1
  - 13f920415a add tornado5 note to 2018.3.1
  - aeacd2b749 allow tornado 5.0 to be installed only for python2
- **PR** #47820: (Ch3LL) Remove output_loglevel in mac_system module @ *2018-05-25 13:10:36 UTC*
  - 09e8c5f0cd Merge pull request #47820 from Ch3LL/mac_system
  - 362414e53b Remove output_loglevel in mac_system module
- **PR** #47798: (rallytime) Back-port #47776 to 2018.3.1 @ *2018-05-23 15:10:43 UTC*
  - **PR** #47776: (garethgreenaway) [2018.3] Fixes to failing _before_connect tests (refs: #47798)
  - 7e314c26c8 Merge pull request #47798 from rallytime/bp-47776
  - ae881547d2 Fixing unit.test_minion.MinionTestCase.test_beacons_before_connect and unit.test_minion.MinionTestCase.test_scheduler_before_connect.
- **PR** #47782: (rallytime) Back-port #47775 to 2018.3.1 @ *2018-05-22 20:56:37 UTC*
  - **PR** #47775: (gtmanfred) catch UnsupportedOperation with AssertionError (refs: #47782)
  - 9c610da0bc Merge pull request #47782 from rallytime/bp-47775
  - bab9c966c5 catch UnsupportedOperation with AssertionError
- **PR** #47770: (rallytime) Back-port #47769 to 2018.3.1 @ *2018-05-22 17:27:20 UTC*
  - **PR** #47769: (gtmanfred) skip test that breaks test suite (refs: #47770)
  - 4adf10b20b Merge pull request #47770 from rallytime/bp-47769
  - 3cfb95c7bc skip test that breaks test suite
- **PR** #47724: (terminalmage) 2 master_tops/ext_nodes fixes @ *2018-05-21 15:59:04 UTC*
  - bbe8e62a98 Merge pull request #47724 from terminalmage/master_tops_fixes
  - 48b8c5acd1 Merge branch '2018.3.1' into master_tops_fixes
  - 89b3070d4c Change deprecation warning to debug logging
  - ceb6e10f87 Fix spurious "Malformed request" error
- **ISSUE** #47484: (whytewolf) Windows: pkg.latest state not updating packages. (refs: #47702)
- **PR** #47739: (rallytime) Back-port #47702 to 2018.3.1 @ *2018-05-21 15:37:03 UTC*

  - **PR** #47702: (damon-atkins) State pkg.latest called win pkg.install with list of pkgs and the required versions (refs: #47739)
  - 97d6fe7434 Merge pull request #47739 from rallytime/bp-47702
  - f79da64bb0 Update is_windows path to use *platform*
  - f04b19b5b6 Ensure targeted_pkgs always contains value for non-windows.
  - 14659f9cad Adjusted based on feed back.
  - 9f18f7cdf5 Whitespace lint issues
  - 2a29b28ee6 pkg.install execution module on windows ensures the software package is installed when no version is specified, it does not upgrade the software to the latest. This is per the design. pkg.latest must provide the versions to install to pkg.install
- **PR** #47730: (rallytime) Back-port #47700 to 2018.3.1 @ *2018-05-21 15:36:16 UTC*
  - **PR** #47700: (yannj-fr) fix roots modification time check (refs: #47730)
  - cfbe0ba73e Merge pull request #47730 from rallytime/bp-47700
  - 9bc35b88ea fix roots modification time check
- **PR** #47727: (Ch3LL) Fix salt.utils.versions.warn_until spelling @ *2018-05-21 13:41:00 UTC*
  - 3614d3d83a Merge pull request #47727 from Ch3LL/spelling
  - 47a8de5b73 Fix salt.utils.versions.warn_until spelling
- **PR** #47736: (Ch3LL) mac_utils test: patch __salt__['cmd.run*'] @ *2018-05-21 13:38:59 UTC*
  - bb45cdaefe Merge pull request #47736 from Ch3LL/fix_util_mac_test
  - ee90c779a8 mac_utils test: patch __salt__['cmd.run*']
- **PR** #47641: (gtmanfred) fix _create_stream and tornado 5.0 @ *2018-05-18 14:25:36 UTC*
  - 43930f8bac Merge pull request #47641 from gtmanfred/2018.3.1
  - 037fd92f59 fix pylint
  - 75d42d8963 Fix last test for tornado
  - a046512287 allow using tornado 5.0
  - 05e651f038 fix _create_stream and tornado 5.0
- **ISSUE** #47532: (edgan) roster auto-add feature in salt-ssh-2018.3.0 (refs: #47541)
- **PR** #47541: (gtmanfred) switch skip-roster to update-roster @ *2018-05-18 13:29:50 UTC*
  - 9f926bcd1a Merge pull request #47541 from gtmanfred/2018.3
  - 8c5c780292 switch skip-roster to update-roster
- **PR** #47719: (rallytime) Back-port #47692 to 2018.3.1 @ *2018-05-18 13:22:02 UTC*
  - **PR** #47692: (dwoz) Default windows to m1.small for ec2-classic (refs: #47719)
  - a963f1b558 Merge pull request #47719 from rallytime/bp-47692
  - 1d9f247fb7 Default windows to m1.small for ec2-classic
- **PR** #47706: (Ch3LL) Add cmd._run_all_quiet to mac_utils and __utils__ in mac_service @ *2018-05-18 01:11:46 UTC*
  - c9108893ab Merge pull request #47706 from Ch3LL/mac_service_util

  - 3611af699f remove added space
  - 9921caa143 fix pylint
  - 317e41d3c0 use cmd._run_quiet and cmd._run_all_quiet instead of importing minion_mods in __salt__
  - a78652515a Add __salt__ to mac_utils and __utils__ in mac_service
- **PR** #47664: (rallytime) Back-port #47645 to 2018.3.1 @ *2018-05-15 18:25:27 UTC*
  - **PR** #47645: (Ch3LL) query the pip path for test test_issue_2087_missing_pip (refs: #47664)
  - fb3bf1ff3e Merge pull request #47664 from rallytime/bp-47645
  - 0a732d8e66 query the pip path for test test_issue_2087_missing_pip
- **PR** #47647: (rallytime) Back-port #47601 and #47643 to 2018.3.1 @ *2018-05-15 14:07:54 UTC*
  - **PR** #47643: (dwoz) Remove unwanted file (refs: #47647)
  - **PR** #47601: (dwoz) Skip tests when we can not use runas (refs: #47647)
  - 9039fee104 Merge pull request #47647 from rallytime/bp-47601-and-47643-2018.3.1
  - 7214fe17c8 Fix typo
  - 506dceed17 Remove unwanted file
  - b6a21dfda3 use ignore-undefined-variable
  - 2429f9fe8a Ignore pylint WindowsError
  - 2d63682fea Better doc string
  - ec2adff699 Skip tests when we can not use runas
- **PR** #47596: (rallytime) Back-port #47568 to 2018.3.1 @ *2018-05-10 22:09:09 UTC*
  - **PR** #47568: (terminalmage) salt.serializers.yaml/yamlex: remove invalid multi_constructor (refs: #47596)
  - 17b5265d95 Merge pull request #47596 from rallytime/bp-47568
  - ecf5dc8b9f Add exception logging on serialize/deserialize exceptions
  - 9659b19819 salt.serializers.yaml/yamlex: remove invalid multi_constructor
- **PR** #47595: (rallytime) Back-port #47569 to 2018.3.1 @ *2018-05-10 22:08:53 UTC*
  - **PR** #47569: (Ch3LL) Update salt.utils.path mock in virtual core test (refs: #47595)
  - c4c400f3e9 Merge pull request #47595 from rallytime/bp-47569
  - 0763f96458 update salt.utils.platform path for virt core test
  - 718252c1ef Update salt.utils.path mock in virtual core test
- **PR** #47599: (rallytime) Back-port #47570 to 2018.3.1 @ *2018-05-10 22:06:44 UTC*
  - **PR** #47570: (gtmanfred) Update dependency to msgpack (refs: #47599)
  - ec7de14be0 Merge pull request #47599 from rallytime/bp-47570
  - 9334c03da9 Update dependency to msgpack
- **PR** #47571: (rallytime) [2018.3.1] Update man pages @ *2018-05-10 16:21:57 UTC*
  - 2a10d92669 Merge pull request #47571 from rallytime/man-pages
  - ade5e9f664 [2018.3.1] Update man pages

- **PR** #47550: (pcn) Fixes a bad deletion I did that only surfaced in 2018.3 @ *2018-05-09 13:36:33 UTC*
    - 85284caaf9 Merge pull request #47550 from pcn/fix-disable-term-protect-in-2018.3
    - d58a56877c Fixes a bad deletion I did that only surfaced in 2018.3
- **ISSUE** #47553: (douglasjreynolds) Unicode version error in lxc (refs: #47554)
- **PR** #47554: (douglasjreynolds) Converted unicode str version to a LooseVersion; matching line 2080. @ *2018-05-09 13:34:13 UTC*
    - f9083ff77e Merge pull request #47554 from douglasjreynolds/lxc_unicode_fix
    - e6bce581c6 Converted unicode str version to _LooseVersion to match line 2080.
- **PR** #47518: (Ch3LL) Fix 47364: ensure we are not caching zfs.is_supported @ *2018-05-09 13:29:07 UTC*
    - fe4e79f1de Merge pull request #47518 from Ch3LL/zfs_support
    - d19fef963e remove unnecessary patch in zfs.is_supported test
    - 58c4f29f96 Fix 47364: ensure we are not caching zfs.is_supported
- **PR** #47159: (terminalmage) Fix for whitelist/blacklist checking for non-list iterables @ *2018-05-08 20:43:51 UTC*
    - 332e9f13a6 Merge pull request #47159 from terminalmage/whitelist_blacklist-iter-fix
    - ca936de372 Treat empty whitelist/blacklist as no whitelist/blacklist
    - bcccaf2621 Raise a TypeError when invalid input passed to check_whitelist_blacklist
    - 2ae510ff2b Fix comment in test
    - 17398efcf7 Fix for whitelist/blacklist checking for non-list iterables
- **PR** #47514: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-05-08 18:36:54 UTC*
    - 21809ddc02 Merge pull request #47514 from rallytime/merge-2018.3
    - e2616b605f Update the pip tests to use the parsing syntax generated in PR #47196
    - b13b59791f Remove double instance of adding *–format=json* in pip module
    - 2ad60c7e81 Lint: remove duplicate function in helpers.py
    - 75480158b3 Lint: cur_version should just be pip_version
    - 5565d5e9b1 Update old utils paths with new utils paths
    - 786076ac03 Merge branch '2017.7' into '2018.3'
        - 611ca1fc03 Merge pull request #47476 from gtmanfred/2017.7
            - 1f91a85587 specify cache dir for pip install
            - 99e150e09c check for kitchen-vagrant gem before loading windows tests
        - 7c3f2c56da Merge pull request #47412 from twangboy/fix_47125
            - c9bab0b8e3 Merge branch '2017.7' into fix_47125
            - 2600e404d5 Fix overly long line
            - 5c8db05769 Fix issue where the cwd was being removed
        - 4846e957c4 Merge pull request #47467 from twangboy/cleanup_settings
            - 9d498293b1 Remove unused settings, update NSIS
        - da9871d36b Merge pull request #47196 from twangboy/fix_47024

      - 14ee5537b9 Add @with_tempdir helper
      - 6c3b5fa6fa Fix typo
      - f031710af2 Merge branch '2017.7' into fix_47024
      - 7c46d9d0d4 Fix integration.modules.test_pip
      - 22ac81df63 Fix integration.modules.test_pip
      - 57d98224d4 Merge pull request #9 from terminalmage/twangboy/fix_47024
      - 37a13d8004 Update pip unit tests to reflect changes
      - 7f86779be0 Lint fix
      - c48d8f4f61 DRY and other fixes in pip module
      - b1117896a0 Change from global variable to __context__``
      - 3e6e524eca Fix some tests``
      - c94f0f20e4 Fix lint error
      - fd47b21530 Fix merge conflict
    - e8c4524bae Merge pull request #47455 from Ch3LL/unreleased_rn
      - b6d0cc2ab7 Add In Progress Warning for 2017.7.6 Release Notes
    - 2c7a4b6179 Merge pull request #47459 from gtmanfred/2017.7
      - d228e72477 update ubuntu-rolling to 18.04
    - 64a64c0ed7 Merge pull request #47462 from terminalmage/docs
      - 6d7803ece0 Fix docs build on Sphinx 1.7+
    - 6cd0d31c03 Merge pull request #47438 from lomeroe/double_admx_test
      - 4902f1e2ba check if a policy has either an enabled value or enabled list entry or a disabled value or disabled list entry when determining the state of the policy
    - ed69821d19 Merge pull request #47433 from s0undt3ch/2017.7
      - 5abadf25d6 Add missing requirements files not commited in #47106
- **ISSUE** #47443: (skylerberg) Input validation does not raise SaltInvocationError in win_dsc.py (refs: #47505)
- **PR** #47516: (rallytime) Back-port #47505 to 2018.3 @ *2018-05-08 13:32:33 UTC*
  - **PR** #47505: (dwoz) Raise proper invocation errors (refs: #47516)
  - 9559ac7679 Merge pull request #47516 from rallytime/bp-47505
  - 7c60e4071e Raise proper invocation errors
- **ISSUE** #47502: (psagers) service.enable (and .disable) destroys /etc/rc.conf on FreeBSD (refs: #47503)
- **PR** #47515: (rallytime) Back-port #47503 to 2018.3 @ *2018-05-08 13:32:03 UTC*
  - **PR** #47503: (psagers) Fix #47502: Remove an extraneous (accidentally introduced?) call to rstrip() (refs: #47515)
  - bf79acfbc8 Merge pull request #47515 from rallytime/bp-47503
  - 821dbb88a0 Fix #47502: Remove an extraneous (accidentally introduced?) call to rstrip.
- **ISSUE** #47511: (joesusecom) sshconfig salt-ssh roster is missing in the documentation (refs: #47531)
- **PR** #47531: (gtmanfred) add ssh config doc for rosters @ *2018-05-07 22:26:30 UTC*

  - 779b3ed056 Merge pull request #47531 from gtmanfred/2018.3
  - 92ded7162c add ssh config doc for rosters
- **PR** #47520: (rallytime) Cleanup weird spaces @ *2018-05-07 19:50:58 UTC*
  - 95b2f9db30 Merge pull request #47520 from rallytime/cleanup-spaces
  - e9cb080a00 Cleanup weird spaces
- **PR** #47495: (dwoz) Fix crufty nssm.exe reference @ *2018-05-07 19:12:49 UTC*
  - 05fc52f124 Merge pull request #47495 from dwoz/uninstall_wart
  - caa36c9064 Merge branch '2018.3' into uninstall_wart
- **ISSUE** #47322: (masau) lxc clone not working (refs: #47494)
- **PR** #47494: (ejparker12) Fixed lxc.clone unhandled exception in salt/modules/lxc.py @ *2018-05-07 19:03:58 UTC*
  - 3cc7d3ae7c Merge pull request #47494 from ejparker12/fix-lxc-clone
  - e0e2c9782d Fixed lxc.clone unhandled exception in salt/modules/lxc.py
- **ISSUE** #47496: (mateiw) salt-ssh –extra-filerefs doesn't include any files if no refs in state files (refs: #47497)
- **PR** #47497: (mateiw) Fix salt-ssh –extra-filerefs to include files even if no refs in states to apply @ *2018-05-07 19:02:50 UTC*
  - adde83f639 Merge pull request #47497 from mateiw/2018.3-fix-ssh-extra-files-refs-issue-47496
  - d67239aae7 –extra-filerefs include files even if no refs in states to apply
- **ISSUE** #47404: (shengis) Localized version of yum breaks pkg.install (refs: #47441)
- **PR** #47441: (shengis) Fix _run to reset LANGUAGE env variable @ *2018-05-07 18:29:25 UTC*
  - 34b1b1ee53 Merge pull request #47441 from shengis/fix-run-env-reset
  - 62fc16b721 Merge branch '2018.3' into fix-run-env-reset
  - 3b02b0bdc1 Merge branch '2018.3' into fix-run-env-reset
  - ee2ab38c8c Fix _run to reset LANGUAGE env variable
- **ISSUE** #47479: (whytewolf) win_task.info on py3 throwing error, but works in py2 (refs: #47507)
- **PR** #47507: (gtmanfred) fix win_task for py3 @ *2018-05-07 17:41:21 UTC*
  - 17cfd4f7cf Merge pull request #47507 from gtmanfred/2018.3
  - 19db39f402 fix win_task for py3
- **PR** #47472: (terminalmage) salt.utils.hashutils: Fix UnicodeEncodeError in several funcs @ *2018-05-07 13:31:07 UTC*
  - a4c2df8fb2 Merge pull request #47472 from terminalmage/hashutils
  - 7266c9984d salt.utils.hashutils: Fix UnicodeEncodeError in several funcs
- **PR** #47485: (gtmanfred) add openstack modules to doc index.rst @ *2018-05-07 13:11:42 UTC*
  - 8b0a370189 Merge pull request #47485 from gtmanfred/2018.3
  - c86163d79f add openstack modules to doc index.rst
    - 3557fc5fa6 Fix crufty nssm.exe reference
- **PR** #47482: (gtmanfred) add all autodoc for new salt openstack modules @ *2018-05-04 21:03:38 UTC*
  - 8df37f734a Merge pull request #47482 from gtmanfred/2018.3

  - 1f65d5cb73 add all autodoc for new salt openstack modules
- **PR** #47447: (dwoz) Fix failing test due to windows console encoding @ *2018-05-04 16:41:29 UTC*
  - d20ca15c5d Merge pull request #47447 from dwoz/strv
  - 8c01773833 Use the same non decodable bytes for all tests
  - 983881a2a1 Add bytes that will not decode using cp1252
- **PR** #47466: (dwoz) bytes file that decodes the same utf-8 and cp1252 @ *2018-05-04 15:54:24 UTC*
  - 8c5b30b541 Merge pull request #47466 from dwoz/randbytes
  - fd9bc06aab bytes file that decodes the same utf-8 and cp1252
- **ISSUE** #46660: (mruepp) top file merging same does produce conflicting ids with gitfs (refs: #47354, #46751)
- **PR** #47465: (rallytime) Back-port #47354 to 2018.3 @ *2018-05-04 13:06:04 UTC*
  - **PR** #47354: (folti) fix forward port of #46751 (refs: #47465)
  - **PR** #46751: (folti) top file merging strategy 'same' works again (refs: #47354)
  - 3658604c43 Merge pull request #47465 from rallytime/bp-47354
  - 3df6fa7990 fix forward port of #46751
- **PR** #47435: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-05-04 13:05:32 UTC*
  - fa293f8fac Merge pull request #47435 from rallytime/merge-2018.3
  - be0731da5f Add skipIfs back in for rest_tornado tests
  - fd98ee3dc1 Lint: Add missing blank line
  - 561718b20b Update old is_windows utils path to new utils path
  - a94cdf8a0d Merge branch '2017.7' into '2018.3'
    - 7ae3497b0c Merge pull request #47429 from gtmanfred/2017.7
      - 8ae32033cc server_list_min should use state, not status
    - 2f5fc4ecc5 Merge pull request #47399 from isbm/isbm-zeromq17-deprecationwarning-2017.7.2-v2
      - a36e49fd27 fix pylint
      - 98b5629b36 Fix imports
      - d94c0f0152 Remove unnecessary variable
      - 8e377b5653 Lintfix: E0203 and attribute access
      - 2aab70b1b8 Install ZMQ handler if <15 version
      - 296c589f4b Use ZMQ switch utility in the integration tests
      - ab5fa34d7c Use ZMQ_VERSION_INFO constant everywhere
      - 43b5558b82 Add trace logging on ZMQ sockets communication
      - 164204a9fe Remove duplicate code for ZMQ monitor handling
      - 834b1e4ff0 Remove obsolete ZMQIOLoop direct instance
      - 1c90cbdb3c Remove an empty line
      - ef2e0acd66 Add logging on ZMQ socket exception
      - 38ceed371d Lintfix: ident

  - 1ece6a5f52 Lintfix: line too long
  - 4e650c0b44 Remove code duplicate by reusing utilities functions
  - 57da54b676 Fix imports
  - 948368e9a1 Add libzmq version info builder
  - 0b4a17b859 Update log exception message
  - 116e1809fc Put a message alongside the exception to the logs
  - 4bc43124b7 Remove unnecessary ZMQ import and check for its presence
  - 05f4d40269 Use utility for ZMQ import handling in SSH client
  - 457ef7d9a5 Use utility for ZMQ import handling in flo/zero
  - 08dee6f5bd Use utility for ZMQ import handling
  - e2a353cfb0 Remove unnecessary ZMQ extra-check for cache utils
  - c8f2cc271d Remove unnecessary ZMQ extra-check for master utils
  - 3940667bb9 Remove old ZMQ import handling
  - f34a53e029 Use ZMQ utility for version check
  - cbb26dcb28 Use ZMQ installer for master
  - 453e83210a Add ZMQ version build
  - af9601e21d Use ZMQ importer utility in async
  - d50b2b2023 Incorporate tornado-5 fixes
  - 1fd9af0655 Add ZMQ backward-compatibility tornado installer for older versions
  - ad4b40415c Add one place for handling various ZMQ versions and IOLoop classes
- b14e974b5f Merge pull request #47343 from Ch3LL/win_srv_test
  - 2173b6f549 ensure we are enabling/disabling before test
  - d58be06751 Add additionatl service module integration tests and enable for windows
- b0f3fb577f Merge pull request #47375 from terminalmage/issue47310
  - fa2bea52bb Remove extra blank line to appease linter
  - f8ab2be81c Add debug logging if we fail to detect virtual packages
  - 67c4fc56ac Warn on use of virtual packages in pkg.installed state
- 56235032f4 Merge pull request #47415 from kstreee/fix-local-client-tgt-bug
  - b8d37e0a1e To add a test case for the syndic environment, copies the test case which was written by @mattp- that was already merged into develop branch, related pr is #46692.
  - 4627bad1fd Realizes 'tgt' field into actual minions using ckminions to subscribe results of the minions before publishing a payload.
- d65ceaee03 Merge pull request #47286 from baniobloom/vpc_peering_connection_name_fix
  - a968965087 Merge branch '2017.7' into vpc_peering_connection_name_fix
- 8a5d4437bb Merge pull request #47270 from meaksh/2017.7-fix-retcode-on-schedule-utils
  - d299cf3385 Merge branch '2017.7' into 2017.7-fix-retcode-on-schedule-utils
  - b6da600fff Initialize __context__ retcode for functions handled via schedule util module

    - 5b51075384 Merge pull request #47371 from rallytime/fix-47264
      - a43485b49c Fix ”of pass” typo in grains.delval docs: change to ”or pass”
    - a86e53be66 Merge pull request #47389 from dwoz/moregittestfix
      - 67745c1362 Older GitPython versions will not have close
    - a5367eaf63 Merge pull request #47388 from dwoz/test_pip_fix
      - eb26321e8b Fix missing import
    - 9b59b991c2 Merge pull request #47380 from gtmanfred/2017.7
      - 93d1445ec1 add io_loop handling to runtests engine
    - 37822c0cbb Merge pull request #47384 from dwoz/test_pip_fix
      - a37a9da1fb Fix py2 version of pip test
    - eefd96732e Merge pull request #47382 from dwoz/gitfs_tests
      - 1570708fac Close the repo and fix multiple tests
    - 57c75ff660 Merge pull request #47369 from terminalmage/ldap_pillar
      - 085883ae2d Return an empty dict if no search_order in ldap ext_pillar config file
    - bcc66dd9bf Merge pull request #47363 from DSRCorporation/bugs/replace_exc_info_with_exception
      - 3f7b93a23c Tornado5.0: Future.exc_info is dropped
    - bcef34f7e1 Merge pull request #47334 from terminalmage/ldap_pillar
      - 0175a8687c pillar_ldap: Fix cryptic errors when config file fails to load
      - 65c3ba7ff1 Remove useless documentation
      - 5d67cb27de Remove unncessary commented line
      - 8de3d41adb fixed vpc_peering_connection_name option
- **PR** #47464: (dwoz) Skip tests not applicable to windows @ *2018-05-04 13:04:38 UTC*
  - 51d21afd4f Merge pull request #47464 from dwoz/skiP_syslog_tests
  - ca9393b7fb Skip tests not applicable to windows
- **PR** #47456: (dwoz) Sysname returns text type @ *2018-05-04 02:57:50 UTC*
  - 3219430dcc Merge pull request #47456 from dwoz/sysname
  - 559ee1961f Sysname returns text type
- **PR** #47458: (Ch3LL) Add In Progress Warning for 2018.3.1 Release Notes @ *2018-05-03 20:40:46 UTC*
  - f3918514a7 Merge pull request #47458 from Ch3LL/unreleased_rn_2018
  - 6a261e5e3a Add In Progress Warning for 2018.3.1 Release Notes
- **PR** #47448: (dwoz) Fix missing import in test suite @ *2018-05-03 14:30:23 UTC*
  - 9fbdcbe994 Merge pull request #47448 from dwoz/transport_import
  - 7e04eb82e1 Fix missing import in test suite
- **ISSUE** #47260: (mew1033) disable_saltenv_mapping not working as expected (refs: #47410)
- **PR** #47410: (terminalmage) gitfs: Fix identification of base env when saltenv mapping is disabled @ *2018-05-03 14:12:27 UTC*

  - 157a32af7f Merge pull request #47410 from terminalmage/issue47260
  - 3ab332ad0e Update tests to reflect bugfix
  - 7b8127f336 gitfs: Fix identification of base env when saltenv mapping is disabled
- **PR** #47413: (dmurphy18) Repobuild improvements for Ubuntu 18.04 lack of gpg2 and better error checking @ *2018-05-02 16:21:31 UTC*
  - 091e4cf9a6 Merge pull request #47413 from saltstack/repobuild_improv
  - c064032110 Removed extra spaces for pylint
  - 20c50b3331 Minor cleanup due to review comments
  - c143b359e9 Update for Ubuntu 18.04 lack of gpg2 and enhanced error checking
- **PR** #47216: (twangboy) Reg docs @ *2018-05-02 13:33:27 UTC*
  - 5e5774fd37 Merge pull request #47216 from twangboy/reg_docs
  - 0beeb58b16 Fix lint, add bytes
  - bad441f8dc Fix some lint'
  - af5139c2ff Add additional examples
  - 24df6ec1b7 Additional docs formatting
  - ff46b27a60 Update reg docs, fix formatting issues
- **PR** #47417: (gtmanfred) revert instantiating a Caller Client in the engine @ *2018-05-01 18:58:06 UTC*
  - 63baf4c4f8 Merge pull request #47417 from gtmanfred/slack
  - 5c8ea7f506 Update slack.py
  - ee8a5eeb10 revert instantiating a Caller Client in the engine
- **ISSUE** #45790: (bdarnell) Test with Tornado 5.0b1 (refs: #46066, #47106, #47433)
- **PR** #47368: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-05-01 18:56:20 UTC*
  - **PR** #47106: (DmitryKuzmenko) Tornado50 compatibility fixes (refs: #47374, #47368, #47433)
  - **PR** #46002: (isbm) Pyzmq 17.0.0 proper handling (refs: #47374, #47368)
  - 0bdfaa5ffe Merge pull request #47368 from rallytime/merge-2018.3
  - 46806e595b Update test assertion comment for pip pkgs
  - d9d24de49e Lint: Add missing import
  - c7b73d132e Merge branch '2017.7' into '2018.3'
    - 31db8ca7ad Merge pull request #47347 from dwoz/test_mysql_fix_again
      - add78fb618 Fix linter warnings
      - 2644cc7553 Fix linter nits
      - 799c601184 Proper fix for mysql tests
  - fefc0cc3ca Update old utils paths to use new utils paths
  - 13e8124031 Merge branch '2017.7' into '2018.3'
    - e573236848 Merge pull request #47359 from gtmanfred/2017.7
      - 6214ed8133 add mention of the formulas channel to the formulas docs

- 629503b2a8 Merge pull request #47317 from dwoz/threadshutdown
  - 6db2a0e4d3 Log exceptions at exception level
  - d4ae787595 Do not join a thread that is stopped
- aacd5cefe3 Merge pull request #47304 from cachedout/test_cli_timeout_arg
  - 85025af83c Pass timeout to salt CLI for tests
- 55534fb659 Merge pull request #47311 from Ch3LL/firewall_windows
  - 4e16c18c16 Add firewall module windows tests to whitelist
  - 4b2fc4ec66 Add windows firewall execution modules integration tests
- 1667375a80 Merge pull request #47348 from dwoz/no_symlinks
  - 94a70e847a Ignore gitfs tests when symlinks not enabled
- dac04261b5 Merge pull request #47342 from dwoz/test_mysql_fix
  - 7496f4c5a8 Fix mysql test cases
- 34e78ef564 Merge pull request #47341 from dwoz/inet_pton_fix
  - 85451f48d4 Fix python 3 support for inet_pton function
- e4779f3246 Merge pull request #47339 from dwoz/ssh_key_test_fix
  - e37a93a1ca Remove redundent close call
  - b2ae5889b7 Close the temporary file handle
  - 9f7f83a975 Use salt.utils.fopen for line ending consistancy
- b221860151 Merge pull request #47335 from dwoz/pip_test_fix
  - dcb6a22c00 Remove un-needed string-escape
- 1c527bfd3a Merge pull request #47331 from dwoz/py3_wingroup_fix
  - cc154ef857 Do not encode usernames
- 708078b152 Merge pull request #47329 from cachedout/frank_credit
  - 33c0644ac4 Credit Frank Spierings
- a545e55543 Merge pull request #47281 from Ch3LL/system_test
  - c9181a75a6 Add destructivetest decorator on tests
  - 0d0c8987fc Add win_system integration module tests
- b64d930df0 Merge pull request #47283 from Ch3LL/ntp_test
  - ced7f86546 Add windows ntp integration module tests
- 910aff910f Merge pull request #47314 from Ch3LL/net_mac_test
  - 67beb1451c Skip netstat test on macosx as its not supported
- 0549ef7c16 Merge pull request #47307 from rallytime/bp-47257
  - 6c5b2f92bc Role is not a list but a dictionary
- d6ff4689f6 Merge pull request #47312 from rallytime/update-bootstrap-release
  - 765cce06a2 Update bootstrap script to latest release: 2018.04.25
- e0765f5719 Merge pull request #47279 from dwoz/py3_build_fix

    - 21dc1bab91 Pep-8 line endings
    - 717abedaf7 Fix comman wart
    - 4100dcd64c Close might get called more than once
    - dbe671f943 Stop socket before queue on delete
    - 9587f5c69e Silence pylint import-error for six.moves
    - 4b0c7d3b34 Fix typo
    - 05adf7c2b1 Use six.moves for queue import
    - fe340778fa Gracefully shutdown worker threads
- 44f19b2f94 Merge pull request #47113 from jfindlay/iptables_state
    - 8bd08012ee modules,states.iptables support proto for policy ext
- b7a6206330 Merge pull request #47302 from Ch3LL/dead_code
    - daa68b4877 Add virtual grains test for core grains
    - a59dd2785d Remove dead code in core grains file for virt-what
- e29362acfc Merge pull request #47303 from baniobloom/bug_fix_doc
    - b97c9df5f3 added clarity on how to figure out what is the oldest supported main release branch
- 0d9d55e013 Merge pull request #47106 from DSRCorporation/bugs/tornado50
    - 39e403b18d Merge branch '2017.7' into bugs/tornado50
    - 6706b3a2d1 Run off of a temporary config
    - d6873800d5 Allow running pytest>=3.5.0
    - 2da3983740 Tornado 5.0 compatibility fixes
- 2e014f4746 Merge pull request #47271 from gtmanfred/amazon
    - 8a53908908 Do not load rh_service module when booted with systemd
    - e4d1d5bf11 Revert "support amazon linux 2 for service module"
- 599b0ed1e9 Merge pull request #47246 from cloudflare/fix-44847-2017.7
    - ad80028104 This way, we can pass flags such as `debug` into the state, but also `test`.
- 4e2e1f0719 Merge pull request #47220 from benediktwerner/fix-pip-2017.7
    - 0197c3e973 Fix pip test
    - 34bf66c09f Fix pip.installed with pip>=10.0.0
- 92e606251f Merge pull request #47272 from rallytime/reg-windows-codeowners
    - 9445af0185 Add windows tests and reg module/state to CODEOWNERS file for team-windows
- 9dca5c0221 Merge pull request #47252 from rallytime/codeowners-fixes
    - 204b6af92b Fix the matching patterns in the CODEOWNERS file to use fnmatch patterns
- 3de1bb49c8 Merge pull request #47177 from fpicot/fix_47173_pkg_normalize
    - 149f846f34 fix normalize parameter in pkg.installed
- 10e30515dc Merge pull request #47251 from Ch3LL/pub_fix_rn

        - fa4c2e6575 Update Docs to remove unnecessary + sign
    - bb7850a431 Merge pull request #47249 from Ch3LL/pub_fix_rn
        - 24dea24b7e Add CVE number to 2016.3.6 Release
    - 56933eb0b2 Merge pull request #47227 from pruiz/pruiz/zfs-dataset-present-slow-2017.7
        - fded61f19b Fix issue #47225: avoid zfs.filesystem_present slowdown when dataset has lots of snapshots
    - 9825065048 Merge pull request #47167 from smitty42/vbox-skd-fix
        - 5de53139cd Merge branch '2017.7' into vbox-skd-fix
    - 976f031170 Merge pull request #47213 from dwoz/py3win
        - ad9c7f63f0 Fix coverate on py3 windows builds
        - 91252bac95 Adding updates for python3 compatibility and new virtualbox SDK version support.
    - cebcd6d069 Merge pull request #47197 from dwoz/testfix
        - 25803c9176 Move process target to top level module namespace
    - d4269c2b70 Merge pull request #47193 from Ch3LL/network_test
        - bbf9987c19 Add network module integration tests
    - c777248a78 Merge pull request #47189 from Ch3LL/autoruns
        - 6a88bedb7a Add autoruns to windows whitelist
        - e9e4d4af70 Add autoruns.list integration test for Windows

- **PR** #47403: (rallytime) Back-port #47356 to 2018.3 @ *2018-05-01 15:19:06 UTC*
    - **PR** #47356: (robinro) Fix sysctl translate (refs: #47403)
    - 4e6870305c Merge pull request #47403 from rallytime/bp-47356
    - 9b682bc48e Fix sysctl translate
- **PR** #47407: (terminalmage) Reduce severity of missing X_update_interval key @ *2018-05-01 15:18:46 UTC*
    - 7e0cdd6145 Merge pull request #47407 from terminalmage/update-interval-log
    - abc592bfff Reduce severity of missing X_update_interval key
- **ISSUE** #47042: (valentin2105) [ERROR] Unable to manage file: 'utf8' codec can't decode byte (refs: #47061)
- **PR** #47405: (terminalmage) Fix file.get_diff regression in 2018.3 branch @ *2018-05-01 15:16:46 UTC*
    - **PR** #47061: (terminalmage) Fix diffing binary files in file.get_diff (refs: #47405)
    - 1377942bcc Merge pull request #47405 from terminalmage/binary-diff
    - 89ddb08026 Use a lambda instead of defining a one-line function
    - b79ff04fda Remove no-longer-used enumerate
    - e03b865359 Add unit test for file.get_diff
    - 5bdc9e9bd5 Fix UnboundLocalError in file.get_diff
- **ISSUE** #47325: (robertodocampo) docker_container.running creates containers using the image ID as the image name (refs: #47367)
- **PR** #47367: (terminalmage) Start docker containers with image name instead of ID @ *2018-04-30 18:46:13 UTC*

  - c267e6083e Merge pull request #47367 from terminalmage/issue47325
  - 798134caa3 Add regression test for creating images with image name insead of ID
  - 4ed47e839c Start docker containers with image name instead of ID
- **ISSUE** #47006: (cedwards) marathon & fx2 grain modules cause master and minion failure (refs: #47401)
- **PR** #47401: (gtmanfred) fix proxy virtual checks for marathon and fx2 @ *2018-04-30 18:44:46 UTC*
  - 3bb00cbb55 Merge pull request #47401 from gtmanfred/proxy
  - 99f9231759 fix proxy virtual checks for marathon and fx2
- **PR** #47397: (rallytime) Add 2018.3.1 Release Notes @ *2018-04-30 14:44:38 UTC*
  - c160fe36ce Merge pull request #47397 from rallytime/2018.3.1-release-notes
  - 3b40cdad2a Add 2018.3.1 Release Notes
- **ISSUE** #45790: (bdarnell) Test with Tornado 5.0b1 (refs: #46066, #47106, #47433)
- **PR** #47374: (DmitryKuzmenko) tornado50 merge forward for 2018.3 @ *2018-04-29 16:29:12 UTC*
  - **PR** #47106: (DmitryKuzmenko) Tornado50 compatibility fixes (refs: #47374, #47368, #47433)
  - **PR** #46002: (isbm) Pyzmq 17.0.0 proper handling (refs: #47374, #47368)
  - 3400f829c4 Merge pull request #47374 from DSRCorporation/bugs/tornado50-2018.3
  - 400999c54f fix pylint
  - 47b6d409d1 add io_loop handling to runtests engine
  - fd074fdb7d use salt.utils.zeromq
  - 4ae33c5d9a Run off of a temporary config
  - 7938b4906e Allow running pytest>=3.5.0
  - 34058c181e Tornado 5.0 compatibility fixes
- **ISSUE** #47124: (mchugh19) Vault module problem in 2018.3.0 (refs: #47379)
- **PR** #47379: (dwoz) Properly encode messages when creating/validating signatures with m2crypto @ *2018-04-28 08:38:23 UTC*
  - 2afe4bee95 Merge pull request #47379 from dwoz/m2crypto_regression
  - 068f2d430d Always sign and verify bytes
  - 7810ebaba9 Add sign regression tests
  - f4441c3a1c Adding regression test for 47124
- **PR** #47277: (morganwillcock) Fix minion crash on NetBSD @ *2018-04-27 15:02:21 UTC*
  - 7390b72808 Merge pull request #47277 from morganwillcock/netbsdswap
  - 0bcb1a079a Merge branch '2018.3' into netbsdswap
  - 30478e8c9c Use swapctl for NetBSD
- **PR** #47320: (twangboy) Change from NSSM to SSM @ *2018-04-27 14:37:50 UTC*
  - 2b7c7ef704 Merge pull request #47320 from twangboy/win_ssm
  - 5549d83aae Use ssm instead of nssm
- **PR** #47308: (rallytime) Back-port #47287 to 2018.3 @ *2018-04-27 13:50:49 UTC*

  - **PR** #47287: (esell) convert unicode ssh pass to str for azure (refs: #47308)
  - b6df5facce Merge pull request #47308 from rallytime/bp-47287
  - 5f392a23fe convert unicode ssh pass to str for azure
- **ISSUE** #47324: (rlschilperoort) archive.extracted keep and/or keep_source not working (refs: #47332)
- **PR** #47332: (garethgreenaway) [2018.3] Removing duplicate code from state/archive.py @ *2018-04-27 13:12:51 UTC*
  - efa3aab800 Merge pull request #47332 from garethgreenaway/47324_archive_extracted_keep_keep_source
  - cc10bfec6b Removing redundant code which is prevening keep & keep_source from being set.
- **PR** #47326: (The-Loeki) Some Redis fixes @ *2018-04-26 17:12:47 UTC*
  - 245d62ca16 Merge pull request #47326 from The-Loeki/redis-cache-sockets
  - d86fbe5bdd redis_return: add unix_socket_path to docs
  - ee9f533765 redis_cache: document UNIX socket access
  - 5337558a5a redis_return: Let redis handle pool creation, add UNIX socket support
  - c90f83b0f9 redis_return: cluster_mode default to False in __virtual__ to prevent KeyError stacktraces
  - 71e3286829 redis_return: Fix code blocks in docs
  - e6605f1c78 redis_cache fix code blox in docs
  - 40e67747ee redis_cache: add socket to options
- **PR** #47319: (dwoz) Skip unix group tests on windows. @ *2018-04-26 15:59:35 UTC*
  - 27a438f0ff Merge pull request #47319 from dwoz/skip_tests
  - d9442d043e Skip tests not applicable to windows
- **PR** #47293: (dwoz) The grp module is not available on windows @ *2018-04-25 20:22:34 UTC*
  - 057f668788 Merge pull request #47293 from dwoz/win_build_fix
  - 0386216005 Fix sneaky indention
  - 082b8d0b3d Use salt.utils.platform
  - cc2538e08f The grp modules is not available on windows
- **ISSUE** #46862: (kivoli) Setting locale.system fails in 2018.3 (refs: #47280, #46869)
- **PR** #47280: (gtmanfred) make sure not to send invalid information @ *2018-04-25 17:46:45 UTC*
  - fff4f8c1a5 Merge pull request #47280 from gtmanfred/localectl
  - 7c212cbb2d fix pylint
  - 6754787e8e update localemod tests
  - 9075070573 make sure not to send invalid information
- **ISSUE** #46977: (gtmanfred) [2018.3.0] Backwards compatibilty breaking change in 2018.3.0 (refs: #47038)
- **PR** #47038: (garethgreenaway) [2018.3] fix to fileclient.py @ *2018-04-25 14:57:04 UTC*
  - 205701dcbe Merge pull request #47038 from garethgreenaway/46977_fixing_fileclient_forward_compatibilty
  - ba01d2133a Updating version.py to include Magnesium.

  - 10c823dd79 The _ext_nodes master function has been renamed to _master_tops. To ensure compatibility when using older Salt masters we continue to pass the function as _ext_nodes until the Magnesium release.
- **ISSUE** #47059: (OrlandoArcapix) Some states incorrectly return None instead of an empty dict when there are no changes (refs: #47060)
- **ISSUE** #46985: (OrlandoArcapix) grafana4_user.present and grafana4_org.present states not working in 2018.3.0 (refs: #47048)
- **PR** #47060: (OrlandoArcapix) Return an empty dict for 'changes' instead of 'None' @ *2018-04-25 14:55:24 UTC*
  - **PR** #47048: (OrlandoArcapix) Issue46985 fix grafana4 state (refs: #47060)
  - 89daf4fdc7 Merge pull request #47060 from OrlandoArcapix/Issue47059-return_dict_from_state
  - 5378e4fd07 Update grafana_datasource test to check for empty dict being returned on no changes, rather than None
  - f115452653 Return an empty dict for 'changes' instead of 'None'
- **ISSUE** #47089: (syphernl) UnicodeDecodeError: 'ascii' codec can't decode byte 0xc3 in position 404: ordinal not in range(128) (refs: #47153)
- **PR** #47153: (terminalmage) salt.modules.ssh: properly encode/decode I/O @ *2018-04-25 14:53:51 UTC*
  - 10cc0d312b Merge pull request #47153 from terminalmage/issue47089
  - bdb52797f8 salt.modules.ssh: properly encode/decode I/O
- **ISSUE** #47199: (tkaehn) Targeting by list (-L) broken for minions behind syndic? (refs: #47275)
- **PR** #47275: (terminalmage) Fix false failure events sent when using syndic @ *2018-04-25 13:56:47 UTC*
  - b5d64f1a70 Merge pull request #47275 from terminalmage/issue47199
  - 8012ad12f8 Fix false failure events sent when using syndic
- **ISSUE** #47267: (skjaro) Problem with beacon diskusage on windows platform in 2018.3 (refs: #47284)
- **PR** #47284: (skjaro) Fix beacon diskusage documentation for the new beahavior mentioned in issue #47267 @ *2018-04-25 13:52:30 UTC*
  - 6215a995d8 Merge pull request #47284 from skjaro/beacon_diskusage_doc_fix
  - fcc042aa5f Fix beacon documentation for the new beahavior mentioned in issue #47267
- **PR** #47291: (bosatsu) Fix proxy minion beacon doc @ *2018-04-25 13:42:36 UTC*
  - 3ef4fe6ed2 Merge pull request #47291 from bosatsu/fix-proxy-minion-beacon-doc
  - 01980b4c43 Fix topics/releases/2018.3.0.rst to include correct example of proxy_example beacon yaml configuration.
  - 9682e26eec Fix topics/proxyminion/beacon.rst to include correct example of salt_proxy beacon yaml configuration.
- **ISSUE** #47239: (bosatsu) Unable to load salt_proxy beacon on minion in 2018.3.0 (refs: #47255)
- **PR** #47255: (garethgreenaway) [2018.3] Fixes to salt_proxy beacon and beacon tests @ *2018-04-25 13:41:51 UTC*
  - ea2d68b865 Merge pull request #47255 from garethgreenaway/47239_fixes_to_salt_proxy_beacon
  - a2a8d78cb0 Fixing status beacon tests.
  - c87d6cae23 Ensure the salt_proxy is returning the correct tuple when the configuration is valid. Update various beacon unit tests to ensure they are testing the results of the validate function for a True result.

- **PR** #47292: (dwoz) Fix decorator wart @ *2018-04-25 04:25:23 UTC*
    - **PR** #47290: (dwoz) Run cache_master test in tmp dir (refs: #47292)
    - 19f9e8258f Merge pull request #47292 from dwoz/cp_fix_again
    - 7d045eb235 Fix decorator wart
- **PR** #47285: (dwoz) Fix reg grains test @ *2018-04-25 00:16:56 UTC*
    - da532aa1ac Merge pull request #47285 from dwoz/core_test_fix
    - 884f4c1829 Fix extra space
    - 8a9027c0c9 Fix reg grains test
- **PR** #47290: (dwoz) Run cache_master test in tmp dir (refs: #47292) @ *2018-04-24 23:37:21 UTC*
    - f591cff643 Merge pull request #47290 from dwoz/test_cp_fix
    - 5ff51affbd Run cache_master test in tmp dir
- **ISSUE** #47092: (syphernl) [2018.3.0] pkg.installed breaks with virtual packages (refs: #47250)
- **ISSUE** #38838: (Zorlin) Failing to remove nginx (refs: #44455)
- **PR** #47250: (terminalmage) Fix virtual package detection @ *2018-04-24 19:22:24 UTC*
    - **PR** #44455: (samodid) Fix for #38838 (refs: #47250)
    - 6d323aa8f0 Merge pull request #47250 from terminalmage/issue47092
    - b8630a70be Fix virtual package detection
- **ISSUE** #47225: (pruiz) zfs.filesystem_present takes forever on a dataset with lots (10k+) of snapshots (refs: #47228, #47227, #47226)
- **PR** #47228: (pruiz) Fix issue #47225: avoid zfs.filesystem_present slowdown when dataset has lots of snapshots (2018.3 branch) @ *2018-04-24 13:35:21 UTC*
    - **PR** #47226: (pruiz) Fix issue #47225: avoid zfs.filesystem_present slowdown when dataset has lots of snapshots (refs: #47228, #47227)
    - 428e915d6a Merge pull request #47228 from pruiz/pruiz/zfs-dataset-present-slow-2018.3
    - cfbf136ab2 Fix issue #47225: avoid zfs.filesystem_present slowdown when dataset has lots of snapshots
- **ISSUE** #46943: (Auha) Slack.Engine could not start (refs: #47262, #47109)
- **PR** #47262: (garethgreenaway) [2018.3] Fixes to targeting in Slack engine @ *2018-04-24 13:18:36 UTC*
    - 0b836106b9 Merge pull request #47262 from garethgreenaway/slack_engine_target_fix
    - bcdef641e8 Removing target and tgt_type from the cmdline that is passed along to Salt, the target is used else where and including it in the cmdline causes problem when it is passed along. Adding an additional test to ensure we are getting the right targt.
- **ISSUE** #47047: (Giandom) Pillars aren't evaluated when alias is passed in Slack Engine (refs: #47142)
- **PR** #47142: (garethgreenaway) [2018.3] pillar and output formatting fixes to Slack engine @ *2018-04-23 19:55:07 UTC*
    - 2ed4b38b02 Merge pull request #47142 from garethgreenaway/47047_passing_pillar_to_slack_aliases
    - 6f183e1d80 Initial commmit for unit/engines/test_slack_engine
    - a2840fc230 Only include the rest of the cmdline if the cmd is an alias.

  - e846df7409 Fixing a bug when passing pillar values to aliases for the Slack engine. Cleaned up the formatting of the results, color codes don't translate well into Slack output. For any state runs, eg. highstate. apply, sls, we run the output through the highstate formater. For anything else run it though the yaml outputer. Running it though highstate causes errors when the output does match what the highstate output is expecting.

- **PR** #47245: (terminalmage) Ensure we pass hexid as bytes when zmq_filtering enabled @ *2018-04-23 16:54:57 UTC*
  - 42a0e655dc Merge pull request #47245 from terminalmage/zeromq-bytes
  - a7accc0548 Ensure we pass hexid as bytes when zmq_filtering enabled
- **PR** #47242: (aesposito91) PY3 fix for zeromq setsockopt @ *2018-04-23 16:38:09 UTC*
  - 73525d1460 Merge pull request #47242 from aesposito91/2018.3
  - b225351e6d Update napalm_syslog.py
- **ISSUE** #47117: (prashanthtuttu) Napalm / Capirca Issue (refs: #47241)
- **PR** #47241: (mirceaulinic) Fix the imports into the netacl execution and state modules @ *2018-04-23 14:56:32 UTC*
  - b78295aee9 Merge pull request #47241 from cloudflare/fix-47117
  - 26c5583264 #47117: fix the napalm imports in the netacl state module
  - 48396467c1 #47117: fix the napalm imports in the netacl execution module
- **PR** #47219: (garethgreenaway) [2018.3] Fixing a backward compatibility issue with vault module & runner @ *2018-04-23 14:10:19 UTC*
  - 88557ea991 Merge pull request #47219 from garethgreenaway/vault_backward_compatibility
  - 1758081ffe When using the vault module on a 2018.3 minion against a 2017.7 master, the 2018.3 minion is expecting a verify element in the results from the Salt runner on the master. The runner in 2017.7 did not include a verify element, which results in an error. This change accounts for this by using the default in 2018.3 which is not to verify if not configured.
- **PR** #47186: (dmurphy18) backport of issue 46933, updated ZFS handling to Salt 2018.3.x @ *2018-04-23 14:07:06 UTC*
  - 370feadbd2 Merge pull request #47186 from dmurphy18/zfs_backport_46933
  - 283359d315 Corrected typo in comma-seprated and 2018.3.0 -> 2018.3.1
  - b7f8d5a22f Replace use of Fluorine with 2018.3.0 for comma-separated warnings
  - 3f30ab2ed6 ZFS backport of 46933 to 2018.3.1
- **PR** #47217: (twangboy) Remove installation of pywin32 from setup.py @ *2018-04-23 13:32:54 UTC*
  - bf3a67d11b Merge pull request #47217 from twangboy/fix_setup
  - eb3d45bb08 Remove installation of pywin32 from setup.py
- **PR** #47195: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-04-20 19:25:30 UTC*
  - 8e21703f13 Merge pull request #47195 from rallytime/merge-2018.3
  - f90fd8c663 Test fix: file strings must be unicode in master config
  - bee4948df1 Lint: use full path for event utils function
  - 120c5446b7 Update old utils paths to new utils paths
  - 4718d31e53 Merge branch '2017.7' into '2018.3'

    - 65f344e371 Merge pull request #47184 from Ch3LL/status_test
      - 25a84428b8 Add status module integration modules tests for Windows
    - 965600ad6c Merge pull request #47163 from rallytime/jenkins-autodoc
      - 0039395017 Updage jenkins module autodocs to use jenkinsmod name instead
    - 0a43dde5fc Merge pull request #47185 from twangboy/add_tests
      - 345daa0423 Add additional integration tests to whitelist
    - 1a600bb9a4 Merge pull request #47172 from dwoz/cover_without_admin
      - cadd759727 Use warnings to warn user
      - 144c68e214 Allow non admin name based runs on windows
    - d5997d2301 Merge pull request #47110 from kstreee/fix-misusing-of-timeout
      - 0624aee0ed Fixes misusing of the timeout option.
    - 87ca2b4003 Merge pull request #40961 from terminalmage/issue40948
      - 6ba66cca41 Fix incorrect logic in exception check
      - fed5041c5f Make error more specific to aid in troubleshooting
      - 8c67ab53b4 Fix path in log message
      - 3198ca8b19 Make error more explicit when PKI dir not present for salt-call
    - f5e63584d4 Merge pull request #47134 from Ch3LL/user_win_test
      - e7c9bc4038 Add user integration tests for windows OS
    - da2f6a3fac Merge pull request #47131 from gtmanfred/cli
      - 1b1c29bf62 add __cli for master processes
    - 9b8e6ffb8c Merge pull request #47129 from rallytime/bp-47121
      - 11da526b21 add ImportError
      - bd0c23396c fix pip.req import error in pip 10.0.0
    - eb5ac51a48 Merge pull request #47102 from gtmanfred/2017.7
      - 3dc93b310b fix tests
      - 8497e08f8e fix pip module for 10.0.0
      - 4c07a3d1e9 fix other tests
      - b71e3d8a04 dont allow using no_use_wheel for pip 10.0.0 or newer
    - c1dc42e67e Merge pull request #47037 from twangboy/fix_dev_scripts
      - 990a24d7ed Fix build_env scripts

- **ISSUE** #46906: (whytewolf) Windows failure with PR 46541 (refs: #47168)
- **PR** #47168: (gtmanfred) fix metadata grain for py3 and windows @ *2018-04-20 19:07:50 UTC*
  - a56eb7e05d Merge pull request #47168 from gtmanfred/metadata
  - 396f7906e3 fix metadata grain for py3 and windows
- **ISSUE** #46918: (AmbicaY) napalm/capirca issue (refs: #47202)
- **PR** #47202: (mirceaulinic) Fix #46918: add the TTL field @ *2018-04-20 14:34:09 UTC*

  - 6135b76e2c Merge pull request #47202 from cloudflare/fix-46918
  - 1e74141cc0 Fix #46918
- **ISSUE** #47150: (srkunze) [Regression] ip_to_host and SSH._expand_target require missing reverse-lookup (refs: #47191)
- **PR** #47191: (terminalmage) salt-ssh: Do not attempt to match host/ip to minion ID if reverse lookup fails @ *2018-04-20 14:20:05 UTC*
  - 7f1115e611 Merge pull request #47191 from terminalmage/issue47150
  - 95a6f075cb Add debug logging when ip_to_host fails
  - 45696e622b salt-ssh: Do not attempt to match host/ip to minion ID if reverse lookup fails
- **PR** #47122: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-04-19 20:44:18 UTC*
  - 1947ffdf56 Merge pull request #47122 from rallytime/merge-2018.3
  - 878fa06134 Test fix: remove tornado testing lib from class
  - a40f007962 lint: get_context is in stringutils.py now
  - 3416e398c6 Update old utils paths references to use new paths
  - 94c2a12be6 Merge branch '2017.7' into '2018.3'
    - 6a4c0b8a1a Merge pull request #47108 from dwoz/async_test_fix
      - 3d85e30ce5 AsyncTestCase is required for AsyncEventPublisher
    - 03892eaf0b Merge pull request #47068 from cachedout/catch_value_error_socket_test
      - 7db5625632 Catch an operation on a closed socket in a test
    - 1ea2885ec2 Merge pull request #47065 from dwoz/jinja_test_fix
      - 673cd31c65 Merge branch '2017.7' into jinja_test_fix
    - 5293b5b5ca Merge pull request #47077 from dwoz/test_state_fix
      - 444da3f893 Fix py3 wart (chr vs bytesstring)
      - e8acca01c2 Fix failing state test by normalizing line endings
    - ca967de5da Merge pull request #47067 from gtmanfred/2017.7
      - f913a7859c use the recommended opennebula lookup method
    - 7fddad6cd9 Merge pull request #47064 from dwoz/roots_tests_fix
      - 25fd7c0694 fix py3 wart, encode os.linesep
      - d79f1a1961 Fix fileserver roots tests
    - 977c6939c4 Merge pull request #47069 from cachedout/match_timeout_arg
      - b8990f5258 Pass the timeout variable to the CLI when calling salt in tests
    - 2c4c19c622 Merge pull request #47074 from dwoz/ignore_artifacts
      - c3941efad0 Kitchn should ignore artifacts directory
    - c484c0bd71 Merge pull request #47055 from bloomberg/GH-47000
      - 8af3f5b874 GH-47000: add proper handling of full_return in cmd_subset
    - f3496030cc Merge pull request #47039 from twangboy/win_fix_winrm_script
      - 6635b9003f Fix winrm powershell script

        - 46fa2c04de Fix py3 os.linesep wart
        - 3c565d7e54 Use salt.utils.fopen
        - aa965310f1 Clean up cruft
        - efc9866580 Jinja test fixes
- **PR** #47162: (terminalmage) Partial backport of #47161 to 2018.3 branch @ *2018-04-19 19:28:47 UTC*
    - **PR** #47161: (terminalmage) Fix failing pillar unit test (refs: #47162)
    - 291cca7ed8 Merge pull request #47162 from terminalmage/bp-47161
    - d185f97a47 mocked file_roots and pillar_roots should be dicts
- **ISSUE** #47081: (sjorge) file.directory with recursion fails if there are non-ascii characters in the path (refs: #47165)
- **PR** #47165: (terminalmage) Make sure a str type is passed to os.walk @ *2018-04-19 14:59:16 UTC*
    - 2ee8006da3 Merge pull request #47165 from terminalmage/issue47081
    - 9e29acb477 Make sure a str type is passed to os.walk
- **PR** #47070: (terminalmage) Use decorators for temp files/dirs in test suite @ *2018-04-19 14:01:48 UTC*
    - 6257862bbb Merge pull request #47070 from terminalmage/with_tempdir
    - 048728d2b7 Remove unused imports
    - 879c557264 Use decorators for temp files/dirs in test suite
- **PR** #47155: (mcalmer) Fix patchinstall for yumpkg @ *2018-04-18 19:24:17 UTC*
    - b46365614b Merge pull request #47155 from mcalmer/fix-patchinstall
    - 382afba457 fix invalid string compare
    - 8c19368938 provide kwargs to pkg_resource.parse_targets required to detect advisory type
- **ISSUE** #47042: (valentin2105) [ERROR] Unable to manage file: 'utf8' codec can't decode byte (refs: #47061)
- **PR** #47061: (terminalmage) Fix diffing binary files in file.get_diff (refs: #47405) @ *2018-04-18 18:52:10 UTC*
    - 13ae1a2413 Merge pull request #47061 from terminalmage/issue47042
    - 87f6cefea3 Rewrite flaky utf8 state to make it easier to troubleshoot
    - df6e535f05 Fix diffing binary files in file.get_diff
- **PR** #47058: (terminalmage) Fix calls to file.lsattr when lsattr is not installed @ *2018-04-18 16:30:12 UTC*
    - cba0f13cd9 Merge pull request #47058 from terminalmage/lsattr
    - eeb067e910 Fix calls to file.lsattr when lsattr is not installed
- **ISSUE** #46929: (noelmcloughlin) 2018.3 regression file.managed.context parsing (refs: #47104)
- **PR** #47104: (terminalmage) yamlloader: Properly handle colons in inline dicts @ *2018-04-18 16:22:47 UTC*
    - b96ce23b3f Merge pull request #47104 from terminalmage/issue46929
    - 33bf6643cd Add additional test for plain scalars
    - 508659b682 yamlloader: Properly handle colons in inline dicts
- **ISSUE** #46887: (julientravelaer) ldap.managed broken with 2018.3.0 (refs: #47029)
- **ISSUE** #46859: (cheribral) pillar_ldap causing TypeError exceptions in python-ldap with unicode objects (refs: #47029)

- **PR** #47076: (terminalmage) pillar_ldap: Load config options as str types @ *2018-04-18 16:16:22 UTC*
  - **PR** #47029: (terminalmage) ldapmod.py/ldap3.py: Force modlist for search/modify/etc. to be str types (refs: #47076)
  - c12697b173 Merge pull request #47076 from terminalmage/issue46859
  - c06c859caf pillar_ldap: Load config options as str types
- **PR** #47107: (twangboy) Fix issues with reg state, add tests @ *2018-04-18 15:53:02 UTC*
  - 50bd885ec7 Merge pull request #47107 from twangboy/fix_46932
  - ae8ab2ab1a Fix tests for py3, enable tearDown
  - 3cf4ac1475 Add integration tests for reg state
  - cc259b146f Cast vdata to appropriate type in reg state
- **ISSUE** #46909: (epelc) Binary *contents_pillar* with file.managed raises UnicodeDecodeError (refs: #47041)
- **PR** #47041: (terminalmage) Force null bytes to be str types @ *2018-04-18 14:08:25 UTC*
  - d6c59696be Merge pull request #47041 from terminalmage/issue46909
  - e4182715be Special check specifically for bytes types
  - ee90dd5d95 Merge branch '2018.3' into issue46909
  - 0e99343a7f Use the same way of defining contents in both file.managed states
  - 5741d287b5 Move back to using null byte check for contents
  - 8e214c9fa9 file.managed: Add test to ensure binary contents work
  - 7b7dc94610 Use salt.utils.stringutils.is_binary to check if contents are binary
  - e3c969da81 PY3: Ensure binary contents work with file.managed
  - 5d98a8bedd Make salt.utils.stringutils.to_binary work for bytestrings
  - 1024000369 Force null bytes to be str types
- **PR** #47007: (twangboy) Fix some issues with the win_servermanager module @ *2018-04-17 20:57:04 UTC*
  - 9a9f6524f8 Merge pull request #47007 from twangboy/fix_46968
  - 432db7c6ec Lint: Remove unused import
  - 10341e8f8b Remove erroneous pop statement
  - 56582f293a Remove redundant try/except block from state`
  - 6ad2427279 Remove unnecessary try/except blocks
  - 92eeaa51bd Put some error checking in the shell command
- **ISSUE** #46943: (Auha) Slack.Engine could not start (refs: #47262, #47109)
- **PR** #47109: (garethgreenaway) [2018.3] fixes to Slack engine @ *2018-04-17 13:56:27 UTC*
  - a52137ee36 Merge pull request #47109 from garethgreenaway/46943_slack_engine_fixes
  - 02baa76595 Fixing a bug that occured when a comment was added to a message sent to Slack by Salt. Also making *slack_engine:groups_pillar* optional.
- **PR** #47045: (tankywoo) Fix ba7d00f5 for gentoo pkg.installed method @ *2018-04-17 13:55:45 UTC*
  - 6c16a34c44 Merge pull request #47045 from tankywoo/fix-gentoo-pkg-installed
  - 551f4e10cf Fix ba7d00f5 for gentoo pkg.installed

- **PR** #47053: (clan) handle jinja error in <module> level @ *2018-04-16 22:47:54 UTC*
  - 86c7cfef56 Merge pull request #47053 from clan/jinja-error
  - a847466946 handle jinja error in <module> level
- **PR** #47062: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-04-16 19:58:32 UTC*
  - 7bfa608e9f Merge pull request #47062 from rallytime/merge-2018.3
  - 59f5880e72 lint fix
  - 1ddf8c584b Update old utils files to new new utils files path
  - 28a79ebba4 Merge branch '2017.7' into '2018.3'
    - 1700a10ebe Merge pull request #46326 from kstreee/fix-client-local
      - 0f358a9c9e Fixes a timing bug of saltnado's client local.
    - c3c00316c5 Merge pull request #46913 from lomeroe/2017_7-fix46877
      - 369a0645ed move exception for clarity
      - 32ce5bfda5 Use configparser serializer object to read psscript.ini and script.ini startup/shutdown script files.
    - 9e37cfc9d6 Merge pull request #47025 from terminalmage/fix-server_id-windows
      - cb0cf89ed3 Fix server_id grain in PY3 on Windows
    - 2e193cfb45 Merge pull request #47027 from rallytime/bp-44508
      - 8e72f362f4 Add priority field to support the latest capirca.
      - 112f92baab Add priority field to support the latest capirca.
    - 385fe2bc1e Merge pull request #47020 from rallytime/bp-46970
      - 9373dff52b Update test_pkgrepo.py
      - 13cf9eb5b1 Removing debugging.
      - a61a8593e5 Removing suse from pkgrepo comments tests. the pkgrepo functions in SUSE pkg module do not support comments.
- **PR** #47066: (terminalmage) Fix regression in handling of environment/saltenv @ *2018-04-16 19:57:12 UTC*
  - fa27e64a33 Merge pull request #47066 from terminalmage/issue46979
  - 5c4c0468ad Fix regression in handling of environment/saltenv
- **PR** #47051: (rallytime) Simplify LooseVersion check in __*virtual*__ check in mac_assistive module @ *2018-04-13 19:43:33 UTC*
  - 8761b81a69 Merge pull request #47051 from rallytime/fix-lint
  - d52b3689d9 Simplify LooseVersion check in __*virtual*__ check in mac_assistive module
- **PR** #47057: (corywright) Fix copy/paste typo in minionfs tutorial @ *2018-04-13 19:43:01 UTC*
  - bbb8018b55 Merge pull request #47057 from corywright/fix-minionfs-whitelist-docs
  - 9b7ee97d12 Fix copy/paste typo in minionfs tutorial
- **ISSUE** #46931: (anlutro) file.managed diff is switched when using template in salt-ssh 2018.3 (refs: #47046)
- **PR** #47046: (clan) switch order of file to be diffed @ *2018-04-13 13:40:13 UTC*
  - d5afa4a2c5 Merge pull request #47046 from clan/file_diff

  - bb58605c54 switch order of file to be diffed
- **ISSUE** #46985: (OrlandoArcapix) grafana4_user.present and grafana4_org.present states not working in 2018.3.0 (refs: #47048)
- **PR** #47048: (OrlandoArcapix) Issue46985 fix grafana4 state (refs: #47060) @ *2018-04-13 13:34:29 UTC*
  - ec9251ecd3 Merge pull request #47048 from OrlandoArcapix/Issue46985-fix-grafana4-state
  - 259d747414 Remove accidentally added copy of a file
  - 6c8c3da74d Return an empty dict instead of 'None' from grafana4 states
- **PR** #47017: (opdude) Don't encode a unicode string @ *2018-04-13 13:31:33 UTC*
  - d8c4c221cf Merge pull request #47017 from Unity-Technologies/hotfix/pip_windows
  - 838670f626 Don't encode a unicode string
- **ISSUE** #46917: (boltronics) mysql_grants.present broken with *database: somedatabase.** (refs: #46919)
- **PR** #47019: (rallytime) Back-port #46919 to 2018.3 @ *2018-04-12 19:43:01 UTC*
  - **PR** #46919: (boltronics) Replace failing is and is not tests with == and != (refs: #47019)
  - 5b7544eaa0 Merge pull request #47019 from rallytime/bp-46919
  - 6837d6c138 Replace failing is and is not tests with == and !=
- **ISSUE** #46887: (julientravelaer) ldap.managed broken with 2018.3.0 (refs: #47029)
- **ISSUE** #46859: (cheribral) pillar_ldap causing TypeError exceptions in python-ldap with unicode objects (refs: #47029)
- **PR** #47029: (terminalmage) ldapmod.py/ldap3.py: Force modlist for search/modify/etc. to be str types (refs: #47076) @ *2018-04-12 19:41:29 UTC*
  - ac2d54d78a Merge pull request #47029 from terminalmage/issue46859
  - ab6314247b ldapmod.py/ldap3.py: Force modlist for search/modify/etc. to be str types
  - 7691dee4ed Add to_str option to decode funcs
- **ISSUE** #46868: (tjyang) 2017.7.4 to 2018.3.0 upgrade issue: Salt request timed out. The master is not responding (refs: #46930)
- **PR** #46930: (dwoz) Clean up bad public key headers @ *2018-04-12 18:57:37 UTC*
  - e6e07720fa Merge pull request #46930 from dwoz/crptodomekeyfix
  - f2e484ed54 Merge branch '2018.3' into crptodomekeyfix
  - e1995a92ee Fix verify signature test
  - 0ba32118d9 Add test for bad public key without m2crypto
  - a44c356233 Clean up bad public key headers
- **ISSUE** #46951: (Giandom) Slack engine error using aliases: TypeError unhashable type (refs: #47008)
- **PR** #47008: (garethgreenaway) [2018.3] Fixing aliases in slack engine @ *2018-04-12 15:24:40 UTC*
  - 0e43becc12 Merge pull request #47008 from garethgreenaway/46951_fixing_slack_engine_aliases
  - dc2a72d44f Fixing aliases in slack engine
- **ISSUE** #46947: (Giandom) Slack engine groups error (refs: #47009)
- **PR** #47009: (garethgreenaway) [2018.3] fixes to slack engine documentation @ *2018-04-12 15:20:54 UTC*

  - c33de7c82d Merge pull request #47009 from garethgreenaway/46947_slack_documentation_update_catch_non_dicts
  - f0fadbb4ce Fixing indention for slack documention. Updating try..except to ensure we catch when groups aren't dicts.
- **PR** #47023: (rallytime) Back-port #46997 to 2018.3 @ *2018-04-12 15:05:24 UTC*
  - **PR** #46997: (LukeCarrier) Fix respository (=> repository) typo in sls_build (refs: #47023)
  - **PR** #44638: (terminalmage) Many improvements to docker network and container states (refs: #46997)
  - 68d17c71f1 Merge pull request #47023 from rallytime/bp-46997
  - c2c60f4ffc Fix respository (=> repository) typo in sls_build
- **PR** #47026: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-04-12 14:39:41 UTC*
  - 9cf3c6406a Merge pull request #47026 from rallytime/merge-2018.3
  - ba70df9d62 Use msgpack utils for loads call, import msgpack for UnpackValueError
  - 34a478dfe5 Update old fopen path with new utils files path
  - 590c7fc13f Merge branch '2017.7' into '2018.3'
    - 8f994e7cf9 Merge pull request #46539 from jfoboss/patch-1
      - 6890122e41 Merge pull request #1 from twangboy/pull_46539
      - 19c3fadbe5 Fix unit test for win_ntp
      - 826a8d3099 Fixing #46504
    - 74d70e95a5 Merge pull request #46999 from gtmanfred/2017.7
      - 791af8f6ce switch pip test package
    - 8adaf7f526 Merge pull request #46023 from bloomberg/parallel-orch
      - 0ac0b3ca29 Merge branch '2017.7' into parallel-orch
    - 39d65a39cf Merge pull request #46613 from myinitialsarepm/fix_puppet.fact_and_puppet.facts
      - 44ecd13abc Update tests to use cmd.run_all
      - 7d7d40f541 Merge branch '2017.7' into fix_puppet.fact_and_puppet.facts
      - 0ce1520bd0 Merge branch '2017.7' into fix_puppet.fact_and_puppet.facts
      - 69e1f6f681 Fix puppet.fact and puppet.facts to use stdout.
      - 3d5e69600b address lint issues raised by @isbm
      - a9866c7a03 fix parallel mode py3 compatibility
      - 6d7730864a removing prereq from test orch
      - 6c8a25778f add integration test to runners/test_state to exercise parallel
      - 2c86f16b39 cherry-pick cdata KeyError prevention from #39832
      - 26a96e8933 record start/stop duration for parallel processes separately
      - e4844bdf2b revisit previous join() behavior in check_requisites
      - f00a359cdf join() parallel process instead of a recursive sleep
      - 6e7007a4dc add parallel support for orchestrations
- **PR** #47021: (garethgreenaway) [2018.3] Fixing integration.modules.test_state_jinja_filters.StateModuleJinjaFiltersTest.test_path_ @ *2018-04-12 13:12:39 UTC*

  - d3be828696 Merge pull request #47021 from garethgreenaway/920_state_module_jinja_filters_test_test_path_which
  - 2ccf2c5fe0 Fixing test_path_which to check that the filter is available rather than results.
- **PR** #47022: (corywright) Add auth.file module to auth documentation page @ *2018-04-11 21:11:10 UTC*
  - 66e8445b82 Merge pull request #47022 from corywright/add-auth-file-module-to-docs
  - bd0918fc40 Add auth.file module to auth documentation page
- **PR** #45774: (twangboy) Fix __virtual__ issue in mac_system.py @ *2018-04-11 14:26:13 UTC*
  - 12ecfdee93 Merge pull request #45774 from twangboy/mac_add_service_util
  - 5796696617 Fix tests for Py3
  - 7b40218790 Fix lint, remove sentence from docstring
  - 781880f0fc Add _available_services function for testing
  - 6080633613 Add assert_called_with
  - 1bf70b2033 Add more tests for available_services
  - b429fc3e74 Add tests for mac_utils
  - b5f67130cc Used *args and **kwargs
  - ed061617a2 Fix unicode_literal issue in mac_assistive
  - 82e17e5fc8 Fix args/kwargs
  - 455146500a Move some functions into mac_utils
  - 125586264b Add utilsmac_service.py
- **ISSUE** #46953: (cskowronnek) salt-cloud azurearm [ERROR] There was a profile error: Parameter 'subscription_id' must be str. (refs: #47012)
- **PR** #47012: (terminalmage) Azure: ensure subscription_id is a str type @ *2018-04-11 13:57:08 UTC*
  - 79347f108a Merge pull request #47012 from terminalmage/issue46953
  - 5192622a32 Azure: ensure subscription_id is a str type
- **PR** #46526: (Ch3LL) Add tests for new source_* minion options @ *2018-04-10 19:56:45 UTC*
  - 6503bf8dfa Merge pull request #46526 from Ch3LL/ip_conf
  - c01180ff47 Patch ZMQ versions for master_uri test
  - da38f332a5 Change comment and salt.utils.network import
  - e972ebdf1a Add for new source_* minion options
- **PR** #46993: (L4rS6) Fix: tuple instead of string @ *2018-04-10 17:07:59 UTC*
  - 03907d3fce Merge pull request #46993 from L4rS6/fix-broken-keystone-auth/2018.3
  - e33ba1b3d5 Fix: tuple instead of string
- **PR** #46990: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-04-10 17:07:33 UTC*
  - ffaee26540 Merge pull request #46990 from rallytime/merge-2018.3
  - ccc5bad2df Merge branch '2017.7' into merge-2018.3
    - ba5421d988 Merge pull request #46991 from gtmanfred/windows
      - 98588c1dc5 use saltstack salt-jenkins

  - 2f1cf3e511 Merge branch '2017.7' into '2018.3'
    - 00c4067585 Merge pull request #46975 from gtmanfred/windows
      - 1f69c0d7f8 make sure windows outputs xml junit files
      - 4a2ec1bbb3 support new versions of winrm-fs
      - b9efec8526 remove libnacl on windows
      - 2edd5eaf9e fix path
      - b03e272e44 windows work
    - 3cf2353e41 Merge pull request #46945 from vutny/doc-faq-fix-jinja
      - bfdf54e61d [DOC] Fix Jinja block in FAQ page
    - fc2f728665 Merge pull request #46925 from terminalmage/fix-file.patch-docstring
      - 97695657f0 Remove reference to directory support in file.patch state
    - eef6c518e1 Merge pull request #46900 from rallytime/bp-46801
      - 6a41e8b457 rename jenkins to jenkinsmod
    - 71839b0303 Merge pull request #46899 from rallytime/bp-45116
      - b92f908da4 fix adding parameters to http.query from sdb yaml
- **PR** #46339: (DmitryKuzmenko) SSH State test failures @ *2018-04-10 17:06:51 UTC*
  - a34b92ae82 Merge pull request #46339 from DSRCorporation/bugs/ssh_state_test_failures
  - bd98c49dc7 Merge branch '2018.3' into bugs/ssh_state_test_failures
  - 6fdc458a7f Increase timeout for run_run in ShellCase
  - 8e60cccdfb Give background task more chance to start.
  - e0b6878fac One more useful assert for better test results.
  - 92a6c43c73 More logging and assertion fixes. Extended ssh ops timeout.
  - 6ebdd17ac4 Advanced logging in the failing SSH State tests.
- **PR** #46989: (Ch3LL) Fix redis cache log debug line @ *2018-04-10 16:35:12 UTC*
  - 9924100c44 Merge pull request #46989 from Ch3LL/redis_log
  - 6160bc06c6 Fix redis cache log debug line
- **ISSUE** #46834: (oeuftete) strftime filter not found in 2018.3.0 (refs: #46848)
- **ISSUE** #46668: (anlutro) Jinja2 filter strftime stopped working in salt-ssh 2018.3 (refs: #46848, #46744)
- **PR** #46848: (garethgreenaway) [2018.8] salt-ssh jinja filters tests @ *2018-04-10 16:19:51 UTC*
  - c6431936cb Merge pull request #46848 from garethgreenaway/testing_jinja_filters_avaiable_via_salt_ssh
  - 5fcda3eff8 Merge branch '2018.3' into testing_jinja_filters_avaiable_via_salt_ssh
  - 0adfee9b11 Updating a couple tests. Fixing check_whitelist_blacklist to work with PY3 when non-iterables are passed. Adding warning about lst_avg results being wrong and future updates in Neon.
  - f3f42146ca Removing expected from strftime and hashsum tests since the results are always different and we are only concerned about the filter being available.
  - 860234c045 Fixing lint.
  - 0891c6b580 fixing docstring

  - c8945e4b2e cleaning up some imports.
  - 0599759e5b cleaning up some test doc strings.
  - dceda5eb88 Moving all jinja filter tests into support/jinja_filters.py. Updaitng integration/ssh/test_jinja_filters.py to use those tests. Adding integration/modules/test_state_jinja_filters.py to also use the common jinja filter tests.
  - 07d7e3ca01 Adding a new integration test and corresponding state files to test availabilty of jinja filters when using salt-ssh.
- **ISSUE** #46880: (liquidgecka) rabbitmq_policy broken in 2018.3.0 (refs: #46973)
- **PR** #46973: (rallytime) New "apply_to" kwarg in rabbitmq module should be added at the end @ *2018-04-10 14:42:32 UTC*
  - **PR** #41233: (dnABic) added parameter apply_to for rabbitmq policy (refs: #46973)
  - fbbcb7584c Merge pull request #46973 from rallytime/fix-46880
  - 8ce21f982c New "apply_to" kwarg in rabbitmq module should be added at the end
- **ISSUE** #46934: (d601) GPG encrypted binary data in pillars breaks in 2018.3.0 (refs: #46966)
- **PR** #46966: (terminalmage) Fix traceback when attempting to decode binary data to unicode @ *2018-04-10 14:08:35 UTC*
  - 58f59cfbff Merge pull request #46966 from terminalmage/issue46934
  - df43ffdb8f salt.payload.Serial: fix traceback when unpacking binary blob
  - 40a49358c9 gpg renderer: fix tranceback when decrypted ciphertext contains binary data
  - 17a88f6a71 Include exc_info in pillar render errors to aid in troubleshooting
- **ISSUE** #46881: (SynPrime) Cron.file - source file not found (refs: #46944)
- **PR** #46944: (garethgreenaway) [2018.3] cron.file with salt source URL @ *2018-04-10 13:34:03 UTC*
  - e33e792e2a Merge pull request #46944 from garethgreenaway/46881_Cron_file_source_file_not_found
  - 438aafeb03 Adding kwargs to calls into file module functions
  - 14d12b1d6b Remove unused imports. Gating tests so they do not run on Windows
  - 623d96f21a Adding dummy cron file for integration/states/test_cron
  - c8e01871d6 Adding an integration test to test cron.file.
  - ddc55d8f9b Fixing bug that made cron.file unable to use a file via a Salt URL.
- **PR** #46937: (gtmanfred) enable_ssh_minions does not work with subset yet @ *2018-04-07 02:54:56 UTC*
  - 08e8782f76 Merge pull request #46937 from gtmanfred/2018.3
  - 3fb75e903c enable_ssh_minions does not work with subset yet
- **PR** #46936: (gtmanfred) don't copy __pycache__ or .pyc files for kitchen @ *2018-04-06 19:15:46 UTC*
  - ac4e7cd73f Merge pull request #46936 from gtmanfred/2018.3
  - 91474878fa don't copy __pycache__ or .pyc files for kitchen
- **ISSUE** #46659: (stamak) [salt-cloud] [new oxygen openstack driver] no public_ips and floating_ips in salt-cloud output (refs: #46912)
- **PR** #46912: (gtmanfred) pull latest vm data after building for openstack shade driver @ *2018-04-06 13:46:42 UTC*

  - 8105fd9715 Merge pull request #46912 from gtmanfred/openstack
  - 5ef538f8ad pull latest vm data after building for openstack shade driver
- **PR** #46908: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-04-05 21:27:03 UTC*
  - 735ea12960 Merge pull request #46908 from rallytime/merge-2018.3
  - 102e966512 Remove redundant section in log setup
  - 177c686b52 Update old utils paths to new utils paths
  - 0a297e7319 Merge branch '2017.7' into '2018.3'
    - d0f5b43753 Merge pull request #44926 from frogunder/whitelisted_acl
      - 18e460fc30 Merge branch '2017.7' into whitelisted_acl
      - 1ad4d7d988 fix assert errors
      - e6a56016df update test
      - 19a2244cb7 whitelist_acl_test
    - 7d822f9cec Merge pull request #46464 from gtmanfred/orchestration
      - 637cdc6b7b fix pylint
      - 0151013ddb document *cli* option for cmd_subset
      - 4a3ed6607d add test for subset in orchestration
      - 3112359dd6 fix salt subset in orchestrator
    - 805ed1c964 Merge pull request #46879 from dwoz/cloudtestfix
      - dc54fc53c3 Fix multiple typos causing tests to fail
    - f70f6de282 Merge pull request #46647 from twangboy/win_fix_test_grains
      - c179388b0e Fix the tear down function in integration.modules.test_grains.GrainsAppendTestCase
    - 91c078ce12 Merge pull request #46756 from nages13/bugfix-grain-virtual_subtype
      - 781f5030a4 Merge branch 'bugfix-grain-virtual_subtype' of https://github.com/nages13/salt into bugfix-grain-virtual_subtype
      - cd1ac4b7f9 Merge branch '2017.7' into bugfix-grain-virtual_subtype
      - 0ace76c0e7 Merge branch '2017.7' into bugfix-grain-virtual_subtype
      - 9eb6f5c0d0 Merge branch '2017.7' into bugfix-grain-virtual_subtype
      - 73d6d9d365 Merge branch '2017.7' into bugfix-grain-virtual_subtype
      - a4a17eba6a Merge branch '2017.7' into bugfix-grain-virtual_subtype
      - bf5034dbdb Merge branch '2017.7' into bugfix-grain-virtual_subtype
      - 8d12770951 Merge branch '2017.7' into bugfix-grain-virtual_subtype
      - 7e704c0e81 Moved down container check code below hypervisors to validate containers type running in virtual environment. Fixes #46754 & #43405
      - 710f74c4a6 fix grains['virtual_subtype'] to show Docker on xen kernels
    - 058bbed221 Merge pull request #46799 from garethgreenaway/46762_prereq_shenanigans_tests
      - 13875e78cf Fixing documention string for test.

        - 3d288c44d4 Fixing test documentation
        - 6cff02ef6a Adding tests for #46788
    - d9770bf3f8 Merge pull request #46867 from terminalmage/unicode-logging-normalization
        - 7652688e83 Backport string arg normalization to 2017.7 branch
    - 9eb98b1f6e Merge pull request #46770 from twangboy/fix_46433
        - 89af0a6222 Merge branch '2017.7' into fix_46433
        - 67b4697578 Remove unused import (ling)
        - 9302fa5ab0 Clean up code comments
        - b383b9b330 Change the order of SID Lookup
    - 9c776cffb7 Merge pull request #46839 from gtmanfred/tupletarget
        - 3b7208ce27 match tuple for targets as well
    - 7db251dc11 Merge pull request #46845 from rallytime/bp-46817
        - 36a0f6d8ca address filehandle/event leak in async run_job invocations
    - e3d17ab7bc Merge pull request #46847 from dwoz/missing-strdup
        - 55845f4846 strdup from libc is not available on windows
    - f2dd79f9c4 Merge pull request #46776 from gtmanfred/2017.7
        - edc1059ee0 fix shrinking list in for loop bug

- **PR** #46853: (terminalmage) Add back date_format filter @ *2018-04-05 20:33:50 UTC*
  - 9a47afc33b Merge pull request #46853 from terminalmage/date_format_filter
  - 266d13a665 Add back date_format filter
- **PR** #46882: (jasperla) Backport #46280 #46849 #46852 to 2018.3 @ *2018-04-05 14:29:12 UTC*
  - **PR** #46852: (jasperla) fix creating a nic tag on a link with double 0 in the MAC (refs: #46882)
  - **PR** #46849: (jasperla) Unbreak creating etherstubs on SmartOS (refs: #46882)
  - **PR** #46280: (jasperla) Remove unneeded checks for binaries in SmartOS modules (refs: #46882)
  - a064a3e695 Merge pull request #46882 from jasperla/smartos/backports
  - 47a66975ff fix creating a nic tag on a link with double 0 in the MAC
  - a3cb0e576e Unbreak creating etherstubs on SmartOS
  - e703254990 Remove unneeded checks for binaries in SmartOS modules
- **PR** #46873: (terminalmage) Attempt UTF-8 first when decoding/encoding @ *2018-04-05 14:16:28 UTC*
  - 4e5e291c99 Merge pull request #46873 from terminalmage/utf8-first
  - cf28eb74aa Don't log command when output_loglevel == 'quiet'
  - f59cee28db Remove hacky workarounds to get encode/decode tests to pass on Windows
  - 76e5d81bb4 Remove hacky workaround to get Windows to decode deserialized data properly
  - 0b5729e58a Remove hacky workaround to get git state/exec module to work properly on Windows
  - 22ff48518f Attempt UTF-8 first when decoding/encoding
- **ISSUE** #43499: (tyeapple) zmq setsockopt need to adapt python3 (refs: #46874)

- **PR** #46878: (terminalmage) Backport #46874 to 2018.3 @ *2018-04-05 13:26:04 UTC*
  - **PR** #46874: (johnj) Use bytestrings for PY3 compatibility when running setsockopt for zmq.SUBSCRIBE (refs: #46878)
  - 1518762465 Merge pull request #46878 from terminalmage/bp-46874
  - d9511d04d4 #43499, zmq setsockopt need to adapt python3
- **ISSUE** #46862: (kivoli) Setting locale.system fails in 2018.3 (refs: #47280, #46869)
- **PR** #46869: (gtmanfred) Always return dictionary for _localectl_status @ *2018-04-05 13:25:14 UTC*
  - 67894e3ee9 Merge pull request #46869 from gtmanfred/2018.3
  - 1496e985f7 fix pylint
  - 75425dfd20 fix tests for localemod
  - 2d7c7b5e33 Always return dictionary for _localectl_status
- **PR** #46870: (mirceaulinic) Correct the documentation for two new proxy modules @ *2018-04-04 21:48:41 UTC*
  - 58c8ff18e2 Merge pull request #46870 from cloudflare/proxy-doc
  - f4b6184476 Corect and add the cimc proxy module to autodoc
  - a99bc202b9 Correct & add Panos to autodoc
- **PR** #46729: (terminalmage) Performance improvement/error catching in expr_match @ *2018-04-04 20:25:57 UTC*
  - d7e4b9d755 Merge pull request #46729 from terminalmage/expr_match
  - 70cfafe299 Add test case
  - 250039b11f Restore original variable name
  - ae0f112a49 Log an exception when non-string val/expr passed to expr_match
  - dac42a672b Performance improvement/error catching in expr_match
- **PR** #46872: (terminalmage) Backport #46863 to 2018.3 @ *2018-04-04 19:04:40 UTC*
  - **PR** #46863: (TamCore) fixed top function which was broken since commit 002aa88a97e (refs: #46872)
  - e0b383afb5 Merge pull request #46872 from terminalmage/bp-46863
  - be284e5b99 Add skipIf when older mock present
  - db8faaee56 Add unit tests for ext_nodes master_tops module
  - ee437f7cbf fixed top function which was broken since commit 002aa88a97e
- **PR** #46850: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-04-04 18:07:44 UTC*
  - 5c76d98d1a Merge pull request #46850 from rallytime/merge-2018.3
  - a0fcd5c053 Fix test_cp failure: forgot to add tgt to test when @with_tempfile is present
  - d0202cab72 Resolve bad merge: there should only be one test_get_file_from_env_in_url test
  - e28f71b418 Lint: use full salt utils path
  - 4ad50bbdee Update old utils paths to new paths
  - 893196d3e6 Merge branch '2017.7' into '2018.3'
    - 1941426218 Merge pull request #46838 from gtmanfred/npm
      - bff61dd291 use http registry for npm

- e544254e7b Merge pull request #46823 from rallytime/fix-42312
  - dafa820f93 Improve __virtual__ checks in sensehat module
- 37f6d2de35 Merge pull request #46641 from skizunov/develop3
  - c624aa4827 Make LazyLoader thread safe
- 989508b100 Merge pull request #46837 from rallytime/merge-2017.7
  - 8522c1d634 Merge branch '2016.11' into '2017.7'
  - 3e844ed1df Merge pull request #46739 from rallytime/2016.11_update_version_doc
  - 4d9fc5cc0f Update release versions for the 2016.11 branch
- 307e7f35f9 Merge pull request #46740 from rallytime/2017.7_update_version_doc
  - 7edf98d224 Update 2018.3.0 information and move branch from "latest" to "previous"
  - 5336e866ac Update release versions for the 2017.7 branch
- ebf5dd276f Merge pull request #46783 from twangboy/fix_46680
  - da5ce25ef3 Fix unit tests on Linux
  - b7f4f377cd Add space I removed
  - f1c68a09b5 Fix network.managed test=True on Windows
- f652f25cc1 Merge pull request #46821 from rallytime/fix-mantest-failures
  - 209a8029c3 Fix the new test failures from the mantest changes
- c460f62081 Merge pull request #46800 from lomeroe/2017_7-46627
  - 2bee383e9d correct create list item value names if the valuePrefix attribute does not exist on the list item, the value is the value name, other wise, the valuename a number with the valuePrefix prepended to it
- df26f2641e Merge pull request #46675 from dwoz/inspectlib-tests
  - d39f4852d8 Handle non-zero status exception
  - 83c005802b Handle cases where git can not be found
  - 628b87d5c4 Skip test when git symlinks are not configured
- 4083e7c460 Merge pull request #46815 from terminalmage/bp-46809
  - 71d5601507 Fix sharedsecret authentication
- 3bac9717f4 Merge pull request #46769 from dwoz/wincloudtest
  - eabc234e5d Fix config override name
  - 5c22a0f88d Use aboslute imports
  - 810042710d Set default cloud test timeout back to 500 seconds
  - 5ac89ad307 Use winrm_verify_ssl option causing tests to pass
  - 71858a709c allow not verifying ssl winrm saltcloud
  - ba5f11476c Adding windows minion tests for salt cloud
- f1be939763 Merge pull request #46786 from twangboy/fix_46757
  - b0053250ff Remove int(), just return -1
  - 7d56126d74 Fixes some lint

        - 49b3e937da Return int(-1) when pidfile contains invalid data
    - 89bf24b15c Merge pull request #46814 from terminalmage/bp-46772
        - a9f26f2ab8 avoid breaking if AutoRemove is not found
        - 97779c965d fix container removal if auto_remove was enabled
    - 5ea4ffbdb6 Merge pull request #46813 from terminalmage/event-debug-log
        - 5d6de3a2eb Get rid of confusing debug logging
    - e533b7182d Merge pull request #46766 from twangboy/win_fix_test_git
        - 5afc66452c Remove unused/redundant imports
        - 88fd72c52c Use with_tempfile decorator where possible
    - 69d450db84 Merge pull request #46778 from terminalmage/salt-jenkins-906
        - bbfd35d3ea Replace flaky SPM man test
    - c935ffb740 Merge pull request #46788 from garethgreenaway/46762_prereq_shenanigans
        - fa7aed6424 Ensure failed tags are added to self.pre.
    - 395b7f8fdc Merge pull request #46655 from dwoz/pyobjects-46350
        - 5aabd442f2 Fix up import and docstring syntax
        - 62d64c9230 Fix missing import
        - 18b1730320 Skip test that requires pywin32 on *nix platforms
        - 45dce1a485 Add reg module to globals
        - 09f9322981 Fix pep8 wart
        - 73d06f664b Fix linter error
        - 009a8f56ea Fix up environ state tests for Windows
        - b4be10b8fc Fixing cleanUp method to restore environment
    - af45c49c42 Merge pull request #46632 from dwoz/file-recurse-36802
        - 44db77ae79 Fix lint errors and typo
        - cb5619537f Only change what is essential for test fix
        - eb822f5a12 Fix file.recurse w/ clean=True #36802
    - 6e9f504ed1 Merge pull request #46751 from folti/2017.7
        - 7058f10381 same top merging strategy works again
    - d3623e0815 Merge pull request #46691 from Ch3LL/win_group_test
        - 7cda825e90 Add groupadd module integration tests for Windows
    - 14ab50d3f4 Merge pull request #46696 from dwoz/win_test_client
        - ec4634fc06 Better explanation in doc strings
        - d9ae2abb34 Fix splling in docstring
        - b40efc5db8 Windows test client fixes

- **PR** #46851: (rallytime) Back-port #46844 to 2018.3 @ *2018-04-04 18:04:59 UTC*
    - **PR** #46844: (UtahDave) Fix warning format in 2018.3.0 release notes (refs: #46851)

  - b808ba7049 Merge pull request #46851 from rallytime/bp-46844
  - ab2cceа1af Quick grammar fix in 2018.3.0 release notes
  - af7bad3c7f Fix warning format in 2018.3.0 release notes
- **ISSUE** #46864: (femnad) Attribute Error When Invoking Vault Module Method (refs: #46865)
- **PR** #46865: (femnad) Fix Log Line for Vault Token Generation Debug Line @ *2018-04-04 14:52:00 UTC*
  - ea56778e03 Merge pull request #46865 from femnad/fix-log-in-vault-runner
  - 01a5b88e7b Fix Log Line for Vault Token Generation Debug Line
- **PR** #46836: (rallytime) [2018.3] Merge forward from 2018.3.0rc1 to 2018.3 @ *2018-04-03 16:54:53 UTC*
  - a0e168ccee Merge pull request #46836 from rallytime/merge-2018.3
  - e75ba1f502 Merge branch '2018.3.0rc1' into '2018.3'
    - 39235715e6 Merge pull request #46792 from damon-atkins/patch-1
    - db5b9464e6 provided an example
    - 41e3e1e253 Update windows information in release notes
    - 99447fbf49 Added more windows information
    - d4241006f2 Update 2018.3.0.rst Windows Items, Group topics
- **ISSUE** #46808: (ezh) Sharedsecret authentication is broken (refs: #46809)
- **PR** #46809: (ezh) Fix sharedsecret authentication (refs: #46815) @ *2018-04-03 16:53:24 UTC*
  - 4a358217a0 Merge pull request #46809 from ezh/2018.3-sharedsecret
  - 20db8f03f7 Merge branch '2018.3' into 2018.3-sharedsecret
  - 9df6d18ec7 Fix sharedsecret authentication
- **PR** #46820: (rallytime) [2018.3] Update the latest release information for docs @ *2018-04-03 14:36:31 UTC*
  - 1519d7d895 Merge pull request #46820 from rallytime/2018.3_update_version_doc
  - 274f8ee0dd [2018.3] Update the latest release information for docs
- **PR** #46731: (rallytime) Back-port #46024 to 2018.3 @ *2018-04-02 19:00:42 UTC*
  - **PR** #46024: (zmedico) Trivial bug fixes for tagify and fire_args functions (refs: #46731)
  - 07f1141722 Merge pull request #46731 from rallytime/bp-46024
  - ee4ee5b619 fire_args: fix UnboundLocalError: local variable 'tag'
  - 4ce2c21824 tagify: handle integer suffix list
- **ISSUE** #46779: (anlutro) salt-ssh 2018.3 states with "runas" fail with "Environment could not be retrieved for User" (refs: #46796)
- **PR** #46796: (terminalmage) Fix regression introduced in merge-forward @ *2018-04-02 18:10:22 UTC*
  - **PR** #46503: (psyer) Fixes stdout user environment corruption (refs: #46796)
  - 4f31c1062d Merge pull request #46796 from terminalmage/issue46779
  - f8f9d045ac Add regression test
  - e0e4e19ba3 Include extra troubleshooting information
  - dcb0c67309 Fix regression introduced in merge-forward

- **PR** #46690: (dincamihai) Fix unicode handling in pkg.info_installed @ *2018-03-29 14:10:48 UTC*
  - 4609a7dd85 Merge pull request #46690 from dincamihai/2018.3
  - 980adf8253 Fix unicode handling in pkg.info_installed
- **PR** #46746: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-03-28 21:13:07 UTC*
  - e5b3c8fa91 Merge pull request #46746 from rallytime/merge-2018.3
  - e8864b7b0b Merge branch '2017.7' into '2018.3'
    - 1222bdbc00 Merge pull request #46732 from rallytime/bp-46032
      - bf0b962dc0 Workaroung python bug in traceback.format_exc()
    - 50fe1e9480 Merge pull request #46749 from vutny/doc-deprecate-copr
      - a1cc55da3d [DOC] Remove mentions of COPR repo from RHEL installation page
    - bd1e8bcc7d Merge pull request #46734 from terminalmage/busybox
      - 6502b6b4ff Make busybox image builder work with newer busybox releases
    - c09c6f819c Merge pull request #46742 from gtmanfred/2017.7
      - fd0e649d1e only use npm test work around on newer versions
    - 3b6d5eca88 Merge pull request #46743 from Ch3LL/mac_auth
      - 4f1c42c0e3 Workaround getpwnam in auth test for MacOSX
  - d0278345fc Update old utils paths to new utils paths
  - e312efb5e7 Merge branch '2017.7' into '2018.3'
    - b548a3e742 Merge pull request #46171 from amaclean199/fix_mysql_grants_comparison
      - 97db3d9766 Merge branch '2017.7' into fix_mysql_grants_comparison
      - 0565b3980e Merge branch '2017.7' into fix_mysql_grants_comparison
      - 8af407173d Merge branch '2017.7' into fix_mysql_grants_comparison
      - 00d13f05c4 Fix mysql grant comparisons by stripping both of escape characters and quotes. Fixes #26920
    - 554400e067 Merge pull request #46709 from vutny/doc-faq-minion-master-restart
      - d0929280fc [DOC] Update FAQ about Salt self-restarting
    - 3f21e9cc65 Merge pull request #46503 from psyer/fix-cmd-run-env-corrupt
      - e8582e80f2 Python 3-compatibility fix to unit test
      - 27f651906d Merge pull request #1 from terminalmage/fix-cmd-run-env-corrupt
      - 172d3b2e04 Allow cases where no marker was found to proceed without raising exception
      - 35ad828ab8 Simplify the marker parsing logic
      - a09f20ab45 fix repr for the linter
      - 4ee723ac0f Rework how errors are output
      - dc283940e0 Merge branch '2017.7' into fix-cmd-run-env-corrupt
      - a91926561f Fix linting problems
      - e8d3d017f9 fix bytes or str in find command

    - 0877cfc38f Merge branch '2017.7' into fix-cmd-run-env-corrupt
    - 86176d1252 Merge branch '2017.7' into fix-cmd-run-env-corrupt
    - 3a7cc44ade Add python3 support for byte encoded markers
    - 09048139c7 Do not show whole env in error
    - ed94700255 fix missing raise statement
    - 15868bc88c Fixes stdout user environment corruption
- ac2a6616a7 Merge pull request #46432 from twangboy/win_locales_utf8
    - affa35c30d Revert passing encoding
    - a0ab27ef15 Merge remote-tracking branch 'dw/win_locales_utf8' into win_locales_utf8
    - 9f95c50061 Use default SLS encoding, fall back to system encoding
    - 6548d550d0 Use salt.utils.to_unicode
    - 8c0164fb63 Add ability to specify encoding in sdecode
    - 2e7985a81c Default to utf-8 on Windows
    - 8017860dcc Use salt.utils.to_unicode
    - c10ed26eab Add ability to specify encoding in sdecode
    - 8d7e2d0058 Default to utf-8 on Windows
- fadc5e4ba4 Merge pull request #46669 from terminalmage/pillar-merge-order
    - b4a1d34b47 Add option to return to pre-2017.7.3 pillar include merge order
- b90f0d1364 Merge pull request #46711 from terminalmage/wildcard-versions-info
    - fc7d16f1af Add performance reminder for wildcard versions
- 6c80d90bb6 Merge pull request #46693 from dwoz/test_smtp_return
    - 5bf850c67f File and Pillar roots are dictionaries
- 9a6bc1418c Merge pull request #46543 from dafenko/fix-add-saltenv-pillarenv-to-pillar-item
    - 6d5b2068aa Merge branch '2017.7' into fix-add-saltenv-pillarenv-to-pillar-item
    - 5219377313 Merge branch '2017.7' into fix-add-saltenv-pillarenv-to-pillar-item
    - b7d39caa86 Merge branch '2017.7' into fix-add-saltenv-pillarenv-to-pillar-item
    - 25f1074a85 Add docstring for added parameters
    - 973bc13955 Merge branch '2017.7' into fix-add-saltenv-pillarenv-to-pillar-item
    - 164314a859 Merge branch '2017.7' into fix-add-saltenv-pillarenv-to-pillar-item
    - 267ae9f633 Fix missing saltenv and pillarenv in pillar.item
- f776040e25 Merge pull request #46679 from vutny/doc-state-pkg
    - 4a730383bf [DOC] Correct examples in *pkg* state module
- 47409eaa6e Merge pull request #46646 from twangboy/win_fix_test_local_cache
    - 8d93156604 Fix *unit.returners.test_local_cache* for Windows
- 0c2dce0416 Merge pull request #46649 from terminalmage/issue46595
    - e82a1aa1ec Make server_id consistent on Python 3

- 4e7466a21c Merge pull request #46588 from UtahDave/no_crash_winshell
  - b7842a1777 Update error message.
  - 95dfdb91ca Don't stacktrace when salt-ssh w/o saltwinshell
- 33af3cfc7c Merge pull request #46631 from rallytime/update-pillar-unit-tests
  - 0f728186aa Fix pillar unit test failures: file_roots and pillar_roots environments should be lists
- d329e7af78 Merge pull request #46640 from terminalmage/file.copy-docs
  - 480c5f8faa Clarify the docs for the file.copy state
- ff40590c06 Merge pull request #46642 from vutny/doc-cloud-index
  - 51e6aa54a1 [DOC] Unify cloud modules index header
- 83ed40c06a Merge pull request #46619 from rallytime/merge-2017.7
  - bcbddf5d07 Merge branch '2017.7.5' into '2017.7'
  - 19bb725698 Merge pull request #46612 from Ch3LL/7.5_rn
  - 6076bfa2ee Add changelog to 2017.7.5 release
  - 31c78aef11 Merge pull request #46572 from dmurphy18/update_xxxbuild
  - c87511570d Merge branch '2017.7.5' into update_xxxbuild
  - cdd768fa4d Merge pull request #46577 from gtmanfred/2017.7.5
  - 78cbf7b5cd Fix npm issue
  - c76f7eb028 enable debug logging on the minionlog
  - e6682c660c Merge pull request #46551 from terminalmage/salt-jenkins-885
  - 703b5e7e65 Change versionadded to show that 2018.3.0 will not have this function
  - 010d260d06 Rewrite failing Suse pkg integration test
  - f3f5dec239 zypper.py: fix version argument being ignored
  - 214f2d6ad3 Add pkg.list_repo_pkgs to zypper.py
  - 0a541613f2 Additon of -sa flag to allow for revision numbers other than -0 or -1
  - bd62699ccb Merge pull request #46563 from gtmanfred/2017.7.5
  - 8d5ab72983 virtualenv version too old for python3.6
  - 2916708124 Merge pull request #46561 from gtmanfred/2017.7.5
  - 2c39ac6dfb disable verbose
  - ee3bff6e32 Merge pull request #46537 from rallytime/bp-46529
  - 289c7a228f retry if there is a segfault
  - 1271536a89 Merge pull request #46519 from rallytime/man-pages-2017.7.5
  - 782a5584f5 Update man pages for 2017.7.5
- df12135439 Merge pull request #46584 from twangboy/lgpo-46568
  - 661017104b Detect disabled reg_multi_sz elements properly
- 2fd3aa487c Merge pull request #46624 from twangboy/win_fix_installer

      - fa0b0efe46 Fix some installer script inconsistencies
    - f038e3c452 Merge pull request #46571 from garethgreenaway/46552_onfail_and_require
      - 152c43c843 Accounting for a case when multiple onfails are used along with requires. Previously if you have multiple states using 'onfail' and two of those states using a 'require' against the first one state, the last two will run even if the 'onfail' isn't met because the 'require' is met because the first state returns true even though it didn't excute. This change adds an additional hidden variable that is used when checking requisities to determine if the state actually ran.
    - 2677330e19 Merge pull request #46520 from gtmanfred/2017.7
      - caefedc095 make sure utils is empty for pickling for windows
      - 2883548e6b pass utils to the scheduler for reloading in modules
    - 7bc3c2e588 Merge pull request #46531 from terminalmage/issue44299
      - b70c3389da Fix case where no comments specified
      - ce391c53f4 Add regression test for #44299
      - c3e36a6c94 Fix regression in yumpkg._parse_repo_file()
      - f0c79e3da3 Slight modification to salt.utils.pkg.rpm.combine_comments()
    - b80edb5d26 Merge pull request #46567 from dwoz/runtest-n-wart
      - 3b6901e19d Honor named tests when running integration suites
    - 1dcd22e767 Merge pull request #46580 from twangboy/win_update_docs_dism
      - d52b99d7a3 Clarify some issues with msu files in win_dism.py
    - 0a68c22332 Merge pull request #46541 from gtmanfred/metadata
      - 19bd1d9db5 handle user-data for metadata grains

- **ISSUE** #46668: (anlutro) Jinja2 filter strftime stopped working in salt-ssh 2018.3 (refs: #46848, #46744)
- **PR** #46744: (garethgreenaway) [2018.3] Ensure salt.utils.dateutils is available for templates via salt-ssh @ *2018-03-28 21:09:46 UTC*
  - ef68df7f3a Merge pull request #46744 from garethgreenaway/46668_jinja2_filter_strftime_unavailable
  - 0b30955c00 Including salt.utils.dateutils so various jinja_filters are available when using salt-ssh.
- **ISSUE** #46334: (sjorge) [2018.3.0rc1] Stacktrace on call to nacl.dec (refs: #46426)
- **PR** #46720: (rallytime) Bump deprecation notices in nacl module & runner to Neon @ *2018-03-27 21:15:46 UTC*
  - **PR** #46426: (garethgreenaway) [2018.3.0rc1] fixes to nacl module & runner (refs: #46639, #46720)
  - 65bb37effd Merge pull request #46720 from rallytime/bump-nacl-deprecation
  - 5102c0310c Bump deprecation notices in nacl module & runner to Neon
- **PR** #46733: (rallytime) [2018.3] Merge forward from 2018.3.0rc1 to 2018.3 @ *2018-03-27 18:46:43 UTC*
  - c83d9e66fe Merge pull request #46733 from rallytime/merge-2018.3
  - 00d4eb26f3 Merge branch '2018.3.0rc1' into '2018.3'
- **PR** #46565: (twangboy) Create reg salt util (2018.3) @ *2018-03-26 22:03:33 UTC*
  - 0faced1d54 Merge pull request #46565 from twangboy/win_fix_cmd_powershell_2018.3
  - 5ee64e9b0e Fix lint (spelling error)

  - 0de54ed953 Additional tests
  - fc9ecd75e2 Skip unit.state.test_reg unless on Windows
  - aa98bdf250 Fix some lint
  - e0d201a96f Make sure the docs are correct for the tests
  - f15f92318d Add tests for salt.utils.win_reg
  - f7112b19a2 Submit #46527 agains 2018.3
- **ISSUE** #46334: (sjorge) [2018.3.0rc1] Stacktrace on call to nacl.dec (refs: #46426)
- **PR** #46639: (terminalmage) Use the correct path for nacl certificates in Windows @ *2018-03-26 19:20:10 UTC*
  - **PR** #46426: (garethgreenaway) [2018.3.0rc1] fixes to nacl module & runner (refs: #46639, #46720)
  - dd52368f90 Merge pull request #46639 from terminalmage/nacl-default-path
  - 2f7660fe35 Use the correct path for nacl certificates in Windows
- **PR** #46416: (dincamihai) Fix cp.push empty file @ *2018-03-26 17:52:47 UTC*
  - 2efef52a3e Merge pull request #46416 from dincamihai/fix-cp.push-empty-file
  - 536ba0fa1e Fix cp.push empty file
- **PR** #46643: (mcalmer) fix docker return @ *2018-03-26 15:52:31 UTC*
  - 84579e7652 Merge pull request #46643 from mcalmer/fix-docker-return
  - 3ceb63f607 fix checking test results
  - af64632bf3 add unit test for failed login
  - 0fc7989236 make it possible to use login, pull and push from module.run and detect errors
- **PR** #46650: (Ch3LL) Mirror libnacl imports in test from the nacl module @ *2018-03-26 14:48:40 UTC*
  - c67afbeb36 Merge pull request #46650 from Ch3LL/nacl_test
  - 9fef8bc431 Mirror libnacl imports in test from the nacl runner
  - f11d58a8e9 Mirror libnacl imports in test from the nacl module
- **PR** #46645: (terminalmage) Add Unicode / Python 3 update to 2018.3.0 release notes @ *2018-03-26 14:43:53 UTC*
  - 03b58a01cf Merge pull request #46645 from terminalmage/release-notes
  - 986c7bcdae Rewrite unicode/py3 section
  - 064bc83276 Add Unicode / Python 3 update to 2018.3.0 release notes
- **ISSUE** #46150: (whytewolf) With chocolately.version some packages don't work with check_remote=True (refs: #46661)
- **PR** #46661: (Kimol) Chocolatey - Lowered name of local and remote packages before comparing versions. @ *2018-03-26 14:35:39 UTC*
  - 308c9ddfc3 Merge pull request #46661 from Kimol/2018.3-fix_chocolatey_check_remote_packages
  - 52581e7918 Removed trailing whitespace
  - 123a86947c Chocolatey - Added lowering local packages for unifing both local and remote names to lowercase for comparison.
  - 4be1a991c2 Lowered name of available packages before comparing with local packages
- **PR** #46569: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 (refs: #46631) @ *2018-03-21 20:57:04 UTC*

- 2e1f7c37f7 Merge pull request #46569 from rallytime/merge-2018.3
- 46ba72fb1c Fix pillar unit test failures: file_roots and pillar_roots environments should be lists
- fe2d46dd0c Better merge conflict resolution for setup.py windows changes
- 8886b61576 Update old utils paths to new paths
- 8d1e1e7f94 Merge branch '2017.7' into '2018.3'
  - 048b2ba3f6 Merge pull request #46547 from garethgreenaway/46427_service_module_cumulus
    - edd0b11447 Merge branch '2017.7' into 46427_service_module_cumulus
    - ea3c16080e Disable the *service* module on Cumulus since it is using systemd.
  - 98e3260b9a Merge pull request #46548 from Ch3LL/profit_test
    - db96c4e72e check for foo,bar username,password set in profitbrick config
  - 79f2a76609 Merge pull request #46549 from Ch3LL/dimension_test
    - bb338c464c Fix dimensionsdata test random_name call
  - 083846fe0e Merge pull request #46529 from gtmanfred/kitchen
    - 50d6e2c7be retry if there is a segfault
  - 5cc11129f1 Merge pull request #46511 from rallytime/bp-45769
    - a8ffceda53 Surpress boto WARNING during decode, reference: https://github.com/boto/boto/issues/2965
  - 0e90c8ca6f Merge pull request #46493 from terminalmage/issue46207
    - f06ff68f10 salt-call: don't re-use initial pillar if CLI overrides passed
  - b11a8fc8e0 Merge pull request #46450 from gtmanfred/salt_runner
    - 7974ff7264 load grains for salt.cmd runner
  - 22d753364b Merge pull request #46337 from gtmanfred/2017.7
    - d6d9e36359 add tests for names and listen/listen_in
    - 3f8e0db572 let listen_in work with names
    - 7161f4d4df fix listen to be able to handle names
  - b7191b8782 Merge pull request #46413 from meaksh/2017.7-explore-result-in-depth
    - 885751634e Add new unit test to check state.apply within module.run
    - 9f19ad5264 Rename and fix recursive method
    - 1476ace558 Fix Python3 and pylint issue
    - 726ca3044d Explore 'module.run' response to catch the 'result' in depth
  - 02a79a2014 Merge pull request #46496 from gtmanfred/kitchen
    - da002f78d0 include virtualenv path for py3 windows
    - fe2efe03ea remove duplicate setup
  - 5c4c182d75 Merge pull request #46330 from bdrung/fix_kubernetes_test_create_deployments
    - 5008c53c44 Fix ValueError for template in AppsV1beta1DeploymentSpec
  - c7e05d3ff4 Merge pull request #46482 from rongshengfang/fix-keyerror-in-instance_present

    - ed8c83e89a Fix KeyError in salt/states/boto_ec2.py when an EIP is being associated to an existing instance with the instance_present state.
- 573d51afec Merge pull request #46463 from terminalmage/mock-2.0
    - b958b4699c Update requirements files to depend on mock>=2.0.0
- a154d35fc7 Merge pull request #46422 from rallytime/bp-46300
    - 829dfde8e8 Change stringutils path to old utils path for 2017.7
    - 91db2e0782 Python 3 support
- 2afaca17a1 Merge pull request #46320 from mcalmer/warn-kubernetes
    - c493ced415 add warning about future config option change
- c7f95581e3 Merge pull request #46449 from bdrung/make-doc-theme-configurable
    - 4a5da2d144 Make documentation theme configurable
- 10ce0e9e20 Merge pull request #46162 from rallytime/team-suse-zypper-owner
    - 13a295a3b7 Add *pkg* and *snapper* to team-suse
    - 35c7b7b0d3 Add btrfs, xfs, yumpkg, and kubernetes file to team-suse
    - 485d777ac0 Add team-suse to CODEOWNERS file for zypper files
- cac096b311 Merge pull request #46434 from gtmanfred/highstate_return
    - d18f1a55a7 fix pylint
    - 9e2c3f7991 split return key value correctly
- 7dd71101ce Merge pull request #46455 from whytewolf/Issue_44452_unicode_cloud
    - 5fe474b1a8 .format remove fix for #44452
- 4c8d9026d3 Merge pull request #46428 from twangboy/win_fix_reqs
    - e7ab97cc17 Remove six as a hard dep for Salt
    - cc67e5c2ef Set six to 1.11.0
- e834d9a63b Merge pull request #46454 from gtmanfred/kitchen
    - b8ab8434a5 fix windows for kitchen
- 2886dca88f Merge pull request #46452 from gtmanfred/spm_cache_dir
    - 169cf7a4e2 make spm cache_dir instead of all cachedirs
- a188984cd9 Merge pull request #46446 from bdrung/fix-typos
    - 7e6e80be87 heat: Fix spelling mistake of environment
    - a3c54b50f6 Fix various spelling mistakes
- e35fc5263c Merge pull request #46309 from bdrung/dynamic-pillarenv
    - 584b451fd1 Support dynamic pillar_root environment
- 35fe9827fe Merge pull request #46430 from terminalmage/issue44032
    - f9f187e915 Improve reliability/idempotence of file.blockreplace state
- 2bad0a21c0 Merge pull request #46429 from twangboy/win_fix_snmp
    - 8995a9b8de Fix problem with __virtual__ in win_snmp

    - 93a572f229 Merge pull request #46100 from jfindlay/resolv_scope
      - d5561bedaf tests.unit.grains.core add scoped IPv6 nameserver
      - 4e2e62d508 salt.utils.dns parse scope param for ipv6 servers
    - 5acc1d5c54 Merge pull request #46420 from bdrung/2017.7
      - e48c13d9e0 Fix SSH client exception if SSH is not found
    - ca6a76e317 Merge pull request #46379 from angeloudy/2017.7
      - 3acb59c74c Merge branch '2017.7' into 2017.7
      - d971e0c08b Fix indent
      - 269514683f Update http.py
      - 908c040ac3 Update http.py
      - 51ba3c135b Update http.py
      - 14aba24111 fix bytes-object required error in python 3
    - 73f9233557 Merge pull request #46404 from gtmanfred/kitchen
      - c56baa95a8 clone .git for the version tests
      - 3620611b5b fix unhold package for debian
      - 5219f7d2ba fix minion log path
    - ca28cfd4e4 Merge pull request #46310 from twangboy/win_update_installer_build
      - bcf8b19566 Update the installer build
    - decccbeca3 Merge pull request #46316 from twangboy/win_fix_dsc
      - 2042d33d59 Fix issues with the DSC module

- **PR** #46620: (rallytime) [2018.3] Merge 2018.3.0rc1 into 2018.3 @ *2018-03-20 22:45:00 UTC*
  - 8cdd56b9dc Merge pull request #46620 from rallytime/merge-2018.3.0rc1-into-2018.3
  - b03cda3cea Merge branch '2018.3.0rc1' into '2018.3'
- **PR** #46606: (Ch3LL) add autodoc topics for infoblox state modules @ *2018-03-19 21:35:46 UTC*
  - 2d2fe22ae2 Merge pull request #46606 from Ch3LL/infoblox_docs
  - 6eab6a7dc4 add autodoc topics for infoblox state modules
- **PR** #46540: (s0undt3ch) Some missing *isinstance* checks. @ *2018-03-15 16:17:19 UTC*
  - 1191d5b379 Merge pull request #46540 from s0undt3ch/2018.3
  - fa1d668774 Some missing *isinstance* checks. Committed again through a PR.
- **PR** #46513: (rallytime) [2018.3] Merge forward from 2018.3.0rc1 to 2018.3 @ *2018-03-15 15:58:59 UTC*
  - 5429438e4b Merge pull request #46513 from rallytime/merge-2018.3
  - aa760334a1 Merge branch '2018.3.0rc1' into '2018.3'
- **ISSUE** #43208: (mitar) Prevent user.present to change uid and gid of existing user (refs: #46502)
- **PR** #46502: (terminalmage) user.present: don't change uid/gid unless explicitly told to @ *2018-03-13 14:25:20 UTC*
  - 3e073c7e8a Merge pull request #46502 from terminalmage/issue43208

  - 4106840deb user.present: don't change uid/gid unless explicitly told to
- **PR** #46398: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-03-12 20:25:19 UTC*
  - 7cdb00ca9c Merge pull request #46398 from rallytime/merge-2018.3
  - d22e5ba442 Merge fix: return back *wb+* mode in *crypt.gen_keys*.
  - c7dddaf8ce Lint: Use log variable, not logger.
  - ca1860cd91 Use new get_umask function in mask calls in virt.py
  - 19ec7b6de1 Update old utils paths with new utils paths
  - d83727fdf9 Merge branch '2017.7' into '2018.3'
    - 95586678c3 Merge pull request #46394 from Ch3LL/mac_doc
      - 158add6661 change oxdownload to oxdownload-{python_version}
      - 21aa848c89 Add mac py2 and py3 packages to mac installation docs
    - 07b5d09ac1 Merge pull request #46338 from rallytime/fix-44831
      - 90771da999 Remove cmd.wait deprecation reference in docs
    - 3849e7a085 Merge pull request #46333 from danlsgiga/issue-42438
      - 3b13f37b44 Revert changes in the code and change docs instead
      - 38114a65d8 Fixes color parameter mismatch and handles 204 responses correctly
    - a8f2f1b063 Merge pull request #46322 from terminalmage/issue44935
      - 85ac6a9893 yamlify_arg: don't treat leading dashes as lists
    - da5c282cb2 Merge pull request #46327 from samilaine/fix-vmware-cloud-fqdn
      - 4b8dfb326f Modify the way a FQDN is handled in the vmware cloud provider.
    - 78c45d3786 Merge pull request #46318 from terminalmage/squelch-warnings
      - 5889b36646 Skip type-checking for several gitfs/git_pillar/winrepo params
    - bb0d6fc263 Merge pull request #46312 from gtmanfred/2017.7
      - 749ae580ed add module_dirs to salt ssh thin tarball
    - 88b5f7383d Merge pull request #46242 from redbaron4/fix-46127
      - 06dba51617 Make changes from review
      - 727ebe1056 Merge branch '2017.7' into fix-46127
      - 08d1ee8baf Fix Python3 test errors
      - aa9d709015 Pass env_vars to pip.freeze
    - a0716643e4 Merge pull request #46265 from Ch3LL/profit_cloud
      - d4893eab4c Add username/password to profitbricks conf for cloud tests
    - ed7bffa7e0 Merge pull request #46306 from rallytime/bp-46256
      - 6439bce4a8 Don't install msgpack 0.5.5
    - 8c2c4e3316 Merge pull request #46208 from terminalmage/audit-umask-usage
      - 9c92aadce8 Disable blacklisted-function check for legitimate uses
      - 58a11aaa26 Disable pylint check in salt-ssh shim

        - ecadf67659 Blacklist os.umask
        - 31b1d98fcb Replace direct use of os.umask with use of existing context manager
        - 82ce546e18 Prevent failed os.makedirs from leaving modified umask in place
      - 978e869490 Merge pull request #46293 from eliasp/2017.7-44624-py3-compat
        - 2e08b0d9c8 Fix Python3 comparison *TypeError* in *salt.modules.upstart*
      - bee4a66d0c Merge pull request #46264 from terminalmage/issue46128
        - 68000b7211 Fix incorrect merge conflict resolution
      - 1e0b3aa348 Merge pull request #46296 from vutny/doc-pillar-get
        - 1faa8331e1 [DOC] Add missing params to *pillar.get* docstring
      - c490a50452 Merge pull request #45874 from GwiYeong/2017.7-local-client-hotfix
        - 949aefc82b Merge branch '2017.7' into 2017.7-local-client-hotfix
        - 45d663f435 fix for local client timeout bug
      - 8e8a3a2897 Merge pull request #46261 from rallytime/merge-2017.7
        - 8256ae5ee5 Merge branch '2016.11' into '2017.7'
        - 140ef4d6b9 Merge pull request #46253 from rallytime/doc-banners
        - 07ed8c7db3 Update docbanner for SaltConf18
        - 9fe86ee520 Merge pull request #46179 from wedge-jarrad/cifs-remount-fix
        - 9ca25c4313 Add credentials and secretfile to mount.mounted mount_invisible_keys

- **PR** #46421: (bdrung) Skip SSHPasswordTests if ssh binary is not found @ *2018-03-09 16:21:02 UTC*
  - 9c089aa4de Merge pull request #46421 from bdrung/skip-ssh-tests-if-ssh-is-missing
  - 3d6f658309 Skip SSHPasswordTests if ssh binary is not found
- **PR** #46453: (bdrung) Fix various spelling mistakes in 2018.3 @ *2018-03-09 14:48:33 UTC*
  - **PR** #46446: (bdrung) Fix various typos (refs: #46453)
  - 4cbfde5839 Merge pull request #46453 from bdrung/fix-typos-2018.3
  - 3d37eca847 Fix various spelling mistakes
- **ISSUE** #44032: (PhilippeAB) blockreplace marker_end isn't applied with newline (refs: #46430)
- **PR** #46437: (terminalmage) Improve reliability/idempotence of file.blockreplace state (2018.3 branch) @ *2018-03-08 15:38:53 UTC*
  - **PR** #46430: (terminalmage) Improve reliability/idempotence of file.blockreplace state (refs: #46437)
  - a43d999fb8 Merge pull request #46437 from terminalmage/issue44032-2018.3
  - 4798187035 Improve reliability/idempotence of file.blockreplace state (2018.3 branch)
- **PR** #46328: (dincamihai) Fix openscap push @ *2018-03-07 17:51:41 UTC*
  - 0c66507aff Merge pull request #46328 from dincamihai/2018.3.0rc1
  - b5e508f339 Fix openscap push
- **PR** #46174: (twangboy) Fix a unicode issue with the git module on Windows @ *2018-03-06 18:53:53 UTC*
  - 82cb2ea5a0 Merge pull request #46174 from twangboy/win_fix_test_git_2

  - 80e3a47dd4 Add output_encoding argument to git state, and add docs
  - 661a0687ec Fix git utf-8 issues for Windows
- **PR** #46235: (twangboy) Fix *unit.modules.test_ssh* for Windows @ *2018-03-05 20:39:44 UTC*
  - 7690cf8564 Merge pull request #46235 from twangboy/win_fix_test_ssh
  - 9ea02d7045 Use write instead of writelines for Windows
- **PR** #46332: (terminalmage) Update the merge-forward docs to reference the 2018.3 branch @ *2018-03-05 19:39:56 UTC*
  - c4f366cdd9 Merge pull request #46332 from terminalmage/merge-forward-docs
  - 0411845cec Update the merge-forward docs to reference the 2018.3 branch
- **PR** #46307: (rallytime) [2018.3] Merge forward from 2018.3.0rc1 to 2018.3 @ *2018-03-03 12:56:07 UTC*
  - 241611aca5 Merge pull request #46307 from rallytime/merge-2018.3
  - c9fa21f62c Merge branch '2018.3.0rc1' into '2018.3'
- **PR** #46314: (terminalmage) Merge 2017.7 branch into 2018.3 @ *2018-03-03 12:54:27 UTC*
  - 30c34f0c62 Merge pull request #46314 from terminalmage/merge-2017.7-2018.3
  - 61ab47ee70 Merge branch '2017.7' into merge-2017.7-2018.3
    - 88a3166589 Merge pull request #46276 from terminalmage/issue44046
      - a14d4daf8c salt.utils.docker.translate_input: operate on deepcopy of kwargs
    - da60399b8f Merge pull request #46183 from oeuftete/fix-docker-container-running-host-config-ulimits
      - 5b09644429 Sort lists from Ulimits before comparing
      - 0b80f02226 Update old dockerng doc ref
    - 509429f08c Merge pull request #46260 from terminalmage/git_pillar
      - b1ce2501fd Normalize global git_pillar/winrepo config items
    - a97a3e6fb0 Merge pull request #46101 from jfindlay/openrc_ret
      - 2eef3c65a6 tests.unit.modules.gentoo_service add retcode arg
      - 81ec66fd8b modules.gentoo_service handle stopped retcode
    - 1a17593c05 Merge pull request #46254 from rallytime/enterprise-banner
      - f5fae3dedf Update enterprise banner
    - 8c50ff32bd Merge pull request #46250 from terminalmage/runner-docs
      - 91b4895087 Add documentation to the fileserver runner
    - 53067cca43 Merge pull request #46243 from racker-markh/fix-openstack-private-network-issue
      - 50c1e140f0 Don't check deny private_ips already in the original list of private_ips
    - 15405c8760 Merge pull request #46239 from terminalmage/issue46109
      - 586d8b0dcf archive.extracted: don't check source file when if_missing path exists
- **ISSUE** #33177: (robnagler) pillar.stack should not continue after errors (refs: #46287)
- **PR** #46287: (bbinet) Update PillarStack stack.py to latest upstream version @ *2018-03-02 21:39:52 UTC*
  - 194b0317ac Merge pull request #46287 from bbinet/upstream-pillarstack

  - b14b6f2c95 Update PillarStack stack.py to latest upstream version
- **PR** #46227: (Ch3LL) Mock file_client call in smtp return test @ *2018-02-28 22:12:22 UTC*
  - 7382654c70 Merge pull request #46227 from Ch3LL/smtp_file_client
  - 280dc9a2b6 Mock file_client call in smtp return test
- **PR** #46232: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-02-28 19:16:37 UTC*
  - 123625213e Merge pull request #46232 from rallytime/merge-2018.3
  - 04f24c1794 Lint: fix from a bad merge
  - aad61c77bd Update old utils paths to new paths
  - 7243baf2c0 Merge branch '2017.7' into '2018.3'
    - 633e1208e4 Merge pull request #46221 from terminalmage/salt-jenkins-854
      - 0eb012659c Fix hanging tests in integration suite
    - 7917277345 Merge pull request #46214 from vutny/formulas-readme-formatting
      - d702846961 [DOC] Replace *note* rST block for GitHub
    - a2e099b744 Merge pull request #46203 from Ch3LL/7.5_release
      - 6ddf3246ce Add 2017.7.5 Release Notes File
    - 973b227818 Merge pull request #46201 from rallytime/merge-2017.7
      - 9ac2101baa Merge branch '2016.11' into '2017.7'
      - a4c5417d23 Merge pull request #46132 from rallytime/2016.11_update_version_doc
      - d2196b6df3 Update release versions for the 2016.11 branch
    - 89cf2e5061 Merge pull request #46139 from bdrung/os-grains
      - 0b445f2a37 tests: Add unit tests for _parse_os_release()
      - f6069b77ed Fix osfinger grain on Debian
      - 8dde55a761 tests: Add os_grains test cases for Debian
      - ff02ab9937 tests: Add Ubuntu 17.10 (artful) os_grains test case
      - 77d5356aba Fix incorrect oscodename grain on Ubuntu
      - 7e62dc9fd2 tests: Support reading os-release files from disk
      - a92ec0db1b Make _parse_os_release() always callable
      - eee1fe5b38 tests: Dissolve _run_ubuntu_os_grains_tests
      - 1d6ef731fe tests: Deduplicate _run_os_grains_tests()
    - c8c71e75ca Merge pull request #46133 from rallytime/2017.7_update_version_doc
      - 0ed338e643 Update release versions for the 2017.7 branch
    - 390d592aa6 Merge pull request #46185 from terminalmage/issue46124
      - 3b58dd0da0 gitfs: Fix detection of base env when its ref is also mapped to a different env
    - 705caa8cca Merge pull request #46148 from rallytime/merge-2017.7
      - 25deebf7a6 Merge branch '2017.7.3' into '2017.7'
      - b5b083fd26 Merge pull request #46074 from Ch3LL/update-7.4

      - 8d0eeeb059 Update 2017.7.4 Release Notes with new fixes
      - 32f3d00e44 Merge pull request #46066 from rallytime/pin-tornado
      - 6dc1a3b9dc Pin tornado version in requirements file
      - 85761ee650 Merge pull request #46036 from terminalmage/issue43769
      - e2140d9a84 Mock the ssh.key_is_encrypted utils func
      - 169924b3fe Move ssh.key_is_encrypted to a utils module temporarily
      - 54f4d78f7a Only keep ssh.py in the Windows installer
      - 5f04531e1b Keep ssh state and execution modules in the installer
      - f2b69f703d git.latest: Fix regression with identity file usage
    - 10a47dcbc4 Merge pull request #46137 from damon-atkins/2017.7_fix_ec2_pillar2
      - 99e7f6a7d3 update ec2 pillar arguments with better names
    - d74cb14557 Merge pull request #46145 from terminalmage/issue46004
      - 467ff841cd pillarenv argument should default to None and not the value from opts
      - 2a185855ea Better solution for fixing the opts munging in pillar.show_pillar runner
      - e2c4702e0c Update tests to reflect changes to the SaltCacheLoader
      - f9301fcc34 Document behavior when orchestration runnner invoked with non-orch states
      - 9644579cd0 Instantiate the SaltCacheLoader's fileclient in the __init__
      - f9a6c86e21 salt.runners.pillar.show_pillar: don't modify master opts
      - e0940a9fc4 Properly detect use of the state.orch alias and add orch jid to kwargs
- **ISSUE** #42932: (bobrik) cmd.run with bg: true doesn't fail properly (refs: #46172, #45932)
- **PR** #46172: (The-Loeki) cmdmod: reimplementation of #45932 for Oxygen @ *2018-02-28 19:14:26 UTC*
  - **PR** #45932: (The-Loeki) Fix cmd run_all bg error (refs: #46172)
  - **PR** #39980: (vutny) [2016.3] Allow to use *bg* kwarg for *cmd.run* state function (refs: #46172, #45932)
  - 20d869c228 Merge pull request #46172 from The-Loeki/fix_cmd_run_all_bg_oxygen
  - 3ecf5018d0 Merge branch '2018.3' into fix_cmd_run_all_bg_oxygen
  - b5315e817b Merge branch '2018.3' into fix_cmd_run_all_bg_oxygen
  - beabf4f06b cmdmod: reimplementation of #45932 for Oxygen
- **PR** #46238: (terminalmage) Don't allow salt.utils.files.fopen() to open stdin/stdout/stderr @ *2018-02-28 19:08:23 UTC*
  - 687575b582 Merge pull request #46238 from terminalmage/fds-in-fopen
  - fe1527a3c4 Don't allow salt.utils.files.fopen() to open stdin/stdout/stderr
- **PR** #46219: (twangboy) Fix *unit.modules.test_network* for Windows @ *2018-02-28 15:45:02 UTC*
  - 3da5dcb313 Merge pull request #46219 from twangboy/win_fix_inet_pton
  - 46f1d2cc09 Use six.text_type instead of six.u
- **PR** #46228: (twangboy) Fix *unit.modules.test_pip* for Windows @ *2018-02-28 15:37:49 UTC*
  - 44343f8063 Merge pull request #46228 from twangboy/win_fix_test_pip

  - 415821eee9 Fix encoding issue
- **PR** #46198: (rallytime) [2018.3] Merge forward from 2018.3.0rc1 to 2018.3 @ *2018-02-27 15:17:51 UTC*
  - adc8950bbe Merge pull request #46198 from rallytime/merge-2018.3
  - 1b4dc71930 Lint fix
  - 776f2ea5d7 Merge branch '2018.3.0rc1' into '2018.3'
- **ISSUE** #45849: (Epiclemonaid) XenServer Provisioning errors out on this line. removing it succeeds. (refs: #46168)
- **PR** #46168: (gtmanfred) driver and provider should be specified @ *2018-02-26 16:17:13 UTC*
  - 06d2dff3ac Merge pull request #46168 from gtmanfred/2018.3
  - ac99bd26db driver and provider should be specified
- **PR** #46161: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-02-26 15:29:39 UTC*
  - 605e5eff73 Merge pull request #46161 from rallytime/merge-2018.3
  - 69ac94baca Update utils paths
  - cffbf52c10 Lint fix: remove extra line
  - 79bed6cff1 Merge branch '2017.7' into '2018.3'
    - 0398ce0482 Merge pull request #46135 from rallytime/bp-46088
      - 57a60f62a3 fix kernel subpackages install bug
    - 1fcbbd1e02 Merge pull request #46136 from rallytime/bp-46115
      - 0a481d707f update digitalocean salt-cloud driver
    - 11e5e8eb86 Merge pull request #45911 from twangboy/win_fix_lgpo_unicode
      - bcde5cc625 Update log statement
      - e9fa53d3b7 Change the Invalid Data Message
      - c818d4b791 Convert reg values to unicode for debug
    - 524a6a72a0 Merge pull request #46123 from gtmanfred/2017.7
      - 8d36730ef7 If no pubkey is passed in openmode fail
    - e48fa58012 Merge pull request #46131 from vutny/doc-formula-formatting
      - d8fb051e44 [DOC] Fix code-blocks for reStructuredText
    - 6cea44ee95 Merge pull request #46118 from rallytime/bp-44603
      - 2a2c23c66b Fix acme state to correctly return on test
    - 16c382b55b Merge pull request #46121 from rallytime/merge-2017.7
      - 4c2f504a85 Merge branch '2016.11' into '2017.7'
      - e197a0fbc5 Merge pull request #46076 from rallytime/bp-46066
      - b94d73c53e Pin tornado version in requirements file
      - c72c1bde5f Merge pull request #46093 from wedge-jarrad/contributing-doc-typo
      - 5a0fe104f7 Fix contributing doc typo
      - 3cb83ea87e Merge pull request #45992 from bgridley/fix-routes-present-state

    - 679787699c Add vpc_peering_connection_id to describe_route_tables route_keys
    - 8a60635da0 Merge pull request #46000 from terminalmage/issue45910
    - 8cf13325ee salt.states.reg.present: Prevent traceback when reg data is binary
    - 1f44e285dc Merge pull request #46011 from terminalmage/fix-solaris-runas
    - 8ee0a3a28b Move Solaris USER workaround up a bit
    - 13cdb52690 cmdmod.py: runas workaround for platforms that don't set a USER env var
    - 30fb8f7be0 Merge pull request #45467 from twangboy/win_exclude_hidden
    - ea41215646 Make the regex pattern less greedy
    - 6d223cffa7 Add tip about passing bogus saltenv
    - 1282ae3a93 Skip hidden first
    - 437a457911 Skip hidden dirs in genrepo
    - 87dc554dc3 Add final updates to docs
    - 3646d5c897 Fix some docs formatting, add some warnings
    - 35c81faf5a Log the source_dir when caching the files
    - 91c3da8dfd Improve docs for pkg.refresh_db
    - 4803d92707 Add some documentation
    - 08b82e0875 Fix lint error, use raw
    - 2f712691cf Exclude hidden directories in pkg.refresh_db
- b92346645b Merge pull request #46107 from amendlik/yumpkg-assumeyes
    - 8d9a432fb2 Add –assumeyes to yum/dnf commands in yumpkg.refresh_db
- 14fe423e0c Merge pull request #46094 from kstreee/fix-memory-leak
    - 48080a1bae Fixes memory leak, saltclients should be cleaned after used.
    - aba00805f4 Adds set_close_callback function to removes stream instance after closed from a set streams.
- 320c2037e1 Merge pull request #46097 from vutny/fix-https-link
    - 2062fd0e5c [DOC] Put https link to the formulas doc page
- 0eb137fb4e Merge pull request #46103 from bdrung/2017.7
    - dd3f936557 Fix skipping Kubernetes tests if client is not installed
- c3a938e994 Merge pull request #46070 from Ch3LL/fix-doc-dns
    - 2a5d855d97 add required arg to dns_check jinja doc example
- 01042e9d77 Merge pull request #46067 from rallytime/bp-45994
    - a07bb48726 Correct formatting for lint
    - e8678f633d Fix Comment being None not '' and inject quotes into the TXT ChangeRecords
- 5e0e2a30e2 Merge pull request #45932 from The-Loeki/fix_cmd_run_all_bg
    - f83da27ca5 Merge branch '2017.7' into fix_cmd_run_all_bg
    - 771758fbca Merge branch '2017.7' into fix_cmd_run_all_bg

        - c54fcf7a2d cmd: move separate DRY logging blocks into _run, prevent logging on bg=True, don't use_vt on bg
        - ebb1f81a9b cmd run: when running in bg, force ignore_retcode=True
    - 45ace39961 Merge pull request #46062 from vutny/pg-user-state-fix-typo
        - a5fbe4e95e Fix typo in postgres_user.present state function
    - edcb64de76 Merge pull request #45763 from twangboy/win_fix_path_rehash
        - b9a2bc7b29 Fix hyperlinks
        - 29912adc15 Move the test_rehash test to test_win_functions
        - adc594c183 Remove duplicate link
        - e84628c1eb Add some comments to the code
        - d50d5f582f Add additional info to docs for *broadcast_setting_change*
        - 3a54e09cd9 Rename setting to message
        - a3f9e99bc0 Change to a generic function to broadcast change
        - 79299361c3 Create refresh_environment salt util
        - 967b83940c Fix rehash function
    - a46fbc546c Merge pull request #46042 from jfindlay/file_tree_doc
        - 0ba4954a4b salt.pillar.file_tree revise module documentation
        - 3c6a5bf967 salt.pillar.file_tree provide better debug info
        - bb1cdc451e salt.pillar.file_tree no stack trace when nodegroups undefined
    - de86126dd8 Merge pull request #46013 from rallytime/bp-45598
        - 2ea3fef543 No lazy logging
        - f427b0febc Change formatting style of logging lines per review
        - ebb244396b Patch around ResourceRecords needing to be present for AliasTarget entries to work

- **PR** #46160: (rallytime) Mark 2 tests as flaky @ *2018-02-23 19:10:06 UTC*
  - 05b771bfd7 Merge pull request #46160 from rallytime/flaky-tests
  - 49e49ae51b Mark 2 tests as flaky
- **PR** #46006: (dincamihai) Remove obsolete unicode handling in pkg.info_installed @ *2018-02-22 19:22:36 UTC*
  - 9b2bc1982c Merge pull request #46006 from dincamihai/oxygen.rc1
  - 99079fc442 Remove obsolete unicode handling in pkg.info_installed
- **PR** #46078: (rallytime) [oxygen] Merge forward from oxygen.rc1 to oxygen @ *2018-02-20 21:49:04 UTC*
  - 93dab45307 Merge pull request #46078 from rallytime/merge-oxygen
  - 2d0f81fd1b Merge branch 'oxygen.rc1' into 'oxygen'
- **ISSUE** #45938: (edgan) zookeeper.present state doesn't deal with an existing zode with no ACL specified (refs: #46043)
- **PR** #46071: (rallytime) Back-port #46043 to oxygen @ *2018-02-16 19:56:36 UTC*
  - **PR** #46043: (edgan) Allow zookeeper znode creation to not require an ACL (refs: #46071)

  - 8d99c3b8fe Merge pull request #46071 from rallytime/bp-46043
  - b82c8bd630 Allow zookeeper znode creation to not require an ACL
- **PR** #46056: (Ch3LL) Fix mac_assistive module not loading @ *2018-02-16 14:57:46 UTC*
  - 5a31422432 Merge pull request #46056 from Ch3LL/ver_mac
  - e44f5133c5 Fix mac_assistive module not loading
- **PR** #46041: (rallytime) [oxygen] Merge forward from 2017.7 to oxygen @ *2018-02-16 14:55:51 UTC*
  - cdca28f5da Merge pull request #46041 from rallytime/merge-oxygen
  - e060a74fd8 Merge branch '2017.7' into 'oxygen'
    - 07e5735471 Merge pull request #46016 from rallytime/bp-45826
      - 1916e5c4a4 Fix selinux.fcontext_policy_present for Centos 6
    - a1f4092811 Merge pull request #46015 from rallytime/bp-45785
      - ef6ffb1492 Resolve linting errors
      - 8047066c46 Remove unused import
      - 8f7c45935a Add tests for salt.modules.selinux.fcontext_get_policy
      - bafb7b4e6e Ensure parsed fields are stripped
      - a830a6e819 m/selinux.fcontext_get_policy allow long filespecs
    - 96097c037e Merge pull request #46012 from rallytime/bp-45462
      - 9f76836a6c emit port cli version, variants as separate args
    - 1279924f5f Merge pull request #45991 from terminalmage/fix-duplicate-extra-opts
      - 916766f651 yumpkg: Fix a couple issues with _get_extra_opts
    - 8b9adc258e Merge pull request #46017 from rallytime/merge-2017.7
      - a06645ce71 Merge branch '2017.7.3' into '2017.7'
      - 6d534c6e7e Merge pull request #46009 from Ch3LL/rn_7.4
      - ac0baf4b34 Add 2017.7.4 Release Notes with PRs
      - ca76a0b328 Merge pull request #45981 from gtmanfred/2017.7.3
      - 0d448457dc apparently local is not set by default
      - 2a92f4bc16 use local config for vault when masterless
      - 6530649dbc Merge pull request #45953 from rallytime/bp-45928-2017.7.3
      - 85363189d1 Fixing vault when used with pillar over salt-ssh
      - fb378cebb0 Merge pull request #45934 from rallytime/bp-45902
      - bb83e8b345 Add regression test for issue 45893
      - cdda66d759 Remove duplicated section in docstring and fix example
      - 4b6351cda6 Check the effective saltenv for cached archive
      - 0d74151c71 Merge pull request #45935 from rallytime/bp-45742
      - 6a0b5f7af3 Removed the chained copy
      - ad1150fad4 list.copy() is not compatible with python 2.7

- d20ff89414 Merge pull request #45988 from rallytime/bp-45797
  - 953a400d79 follow symlinks
- b18087cee0 Merge pull request #45711 from bdrung/fix-unicode-tests
  - b6181b5ed6 Fix Unicode tests when run with LC_ALL=POSIX
- 5271fb1d40 Merge pull request #45878 from damon-atkins/2017.7_fix_ec2_pillar
  - 0e74025714 Merge branch '2017.7' into 2017.7_fix_ec2_pillar
  - b4d0b23891 py3 fix
  - 75d9e20d8a Add ignoring 'terminated', 'stopped' instances, to improve changes of a single match
  - 0093472a37 added tag_key_list and tag_key_sep to create ec2_tags_list
  - afb3968aa7 ec2_pillar could not find instance-id, resolved. add support to use any tag to compare minion id against.
- cf367dbd04 Merge pull request #45942 from terminalmage/issue45679-2017.7
  - 89cbd72a0d Don't try to sort ports when translating docker input
  - 9cd47b39dd Fix incorrect translation of docker port_bindings -> ports
- dae41de7a8 Merge pull request #45959 from rallytime/state-doc-update
  - 6f781cb95d A couple of grammar updates for the state compiler docs
- 007214f7bf Merge pull request #45908 from DimensionDataResearch/fix/issue/45884
  - 1a75786b5a Fix linter warnings.
  - 82ec0b589c Revert to using salt.utils.cloud.is_public_ip.
  - 9b6b01873b Fix violations reported by flake8.
  - a2bc155c73 Use __utils__['cloud.'] instead of salt.cloud.utils.
  - 98907a32cb Ensure 'auth' parameter is correctly passed to dimensiondata driver.
  - de26b03e2c Fix copy/paste bug in dimensiondata provider integration test.
  - 6b1b6be427 Add integration tests for dimensiondata cloud provider.
  - f6ea9fed7d Ensure that event data provided by the dimensiondata driver is serialisable.
- efcbfa868c Merge pull request #45985 from garethgreenaway/2017_7_fixing_mac_tests_again
  - 7b8dc14433 Missing *format* in the call to write.
- bf03abd07c Merge pull request #45958 from garethgreenaway/backport-fixing_mactests_queue_full
  - 25dffaae91 Backporting #45935
- bab365d6c6 Merge pull request #45949 from rallytime/merge-2017.7
  - f51687e903 Merge branch '2016.11' into '2017.7'
  - 7779fea7ba Merge pull request #45940 from dmurphy18/fix_aix_cmdmod
  - dd2788419b Fix use of 'su' for AIX to use '-'
- 7fd00ec752 Merge pull request #45928 from garethgreenaway/45915_fixing_vault_pillar_for_salt_ssh

            - 259e60e5d4 Fixing vault when used with pillar over salt-ssh
        - 9d14ad9ccf Merge pull request #45925 from terminalmage/fix-spelling
            - 7a143fe454 Fix spelling error in docstring

- **PR** #45972: (mcalmer) move log_file option to changeable defaults @ *2018-02-15 18:57:24 UTC*
    - 057e895faf Merge pull request #45972 from mcalmer/allow-salt-ssh-define-log_file
    - f89a20bf3e move log_file option to changeable defaults
- **PR** #46007: (rallytime) [oxygen] Merge forward from oxygen.rc1 to oxygen @ *2018-02-13 18:50:09 UTC*
    - d4377d4678 Merge pull request #46007 from rallytime/merge-oxygen
    - d6c2d0693a Merge branch 'oxygen.rc1' into 'oxygen'
- **PR** #45944: (mirceaulinic) Add NetBox module autodoc @ *2018-02-13 00:01:48 UTC*
    - 069f790b3c Merge pull request #45944 from cloudflare/netbox-autodoc
    - ed69b987cf Add NetBox module autodoc
- **PR** #45984: (garethgreenaway) [oxygen] Missing *format* in the call to write. @ *2018-02-12 19:06:04 UTC*
    - 2a6285d313 Merge pull request #45984 from garethgreenaway/fixing_mac_tests_again
    - ae7791d30b Missing *format* in the call to write.
- **PR** #45922: (rallytime) [oxygen] Merge forward from 2017.7 to oxygen @ *2018-02-09 20:24:26 UTC*
    - 88f481a3df Merge pull request #45922 from rallytime/merge-oxygen
    - 9c49c8d47c Remove extra patch
    - b96f4cf8ad Remove duplicate import in cmdmod.py
    - 34ecdffa71 Replace old utils paths with new paths
    - d80547e0b8 Merge branch '2017.7' into 'oxygen'
    - 0cbe93cd69 Merge pull request #45920 from rallytime/merge-2017.7
        - e4e4744218 Merge branch '2016.11' into '2017.7'
        - 27ff82f996 Merge pull request #45864 from rallytime/release-note-fix
            - 104a24f244 Remove extraneous] in release notes for 2016.11.9
        - 5fa010de2b Merge pull request #45787 from rallytime/2016.11.9_docs
            - a38d4d44fa [2016.11] Bump latest and previous versions
    - 643a8a5278 Merge pull request #45814 from gtmanfred/2017.7
        - d8eec9aa97 fix cookies dict size changing in http.query
    - 3a3f87c16d Merge pull request #45877 from rallytime/new-release-notes
        - f937e8ba81 Add release notes file for 2017.7.4 release
    - 1c3cc00670 Merge pull request #45904 from rallytime/bp-41017
        - 80c56cdcea Fixed typo in pkg state documentation
    - 317d35bd15 Merge pull request #45907 from terminalmage/fix-grains-backport
        - 6cf7e50cc4 Fix backport of grains fix
    - dade5f0cab Merge pull request #45906 from rallytime/bp-45548

    - 1befa7386c Update x509.py
  - 82c473a1fe Merge pull request #45902 from terminalmage/issue45893
    - 9d200efc26 Add regression test for issue 45893
    - 1468f1d0ff Remove duplicated section in docstring and fix example
    - 6cc5cd9b8a Check the effective saltenv for cached archive
  - fdedde3cfb Merge pull request #45862 from rallytime/bp-45830
    - 1024856f9a Wrapping the put_nowait in a try…except and catching the exception when the multiprocessing queue is full. This situation is happening when running the full testing suite on MacOS where the queue limit is 32767 vs on Linux where the queue limit is unlimited.
  - 43a45b42c3 Merge pull request #45779 from The-Loeki/patch-3
    - 8575ae3d52 Merge branch '2017.7' into patch-3
    - 47cf00d88e SSH shell shim: Don't use $() for optimal support
  - cca997d0da Merge pull request #45788 from rallytime/2017.7.3_docs
    - d5faf6126b [2017.7] Bump latest and previous versions
  - 746206cebe Merge pull request #45842 from rallytime/bp-45827
    - c631598a87 Fix traceback in disks grains when /sys/block not available
  - 900aadcd67 Merge pull request #45721 from garethgreenaway/44978_show_duration_when_no_state_run
    - 359265869f Adding a couple tests to ensure that duration is included in state run results even when states do not run.
    - 912347abc3 Include the duration when a state does not run, for example when the *onchanges* requisite is not met.
  - 80a2d009b4 Merge pull request #45517 from kstreee/fix-mkdir
    - 24d41f2451 Fixes base dir making logic to ensure not raising the exception when base directory already exists.
  - 7a4b1b2e77 Merge pull request #45835 from kstreee/fix-missing-return-statement
    - 68c7f3dcba Adds a missing return statement.
  - 0a04f118c2 Merge pull request #45840 from rallytime/bp-45603
    - 9653363131 Fix for duplicate entries with pkrepo.managed
  - bd2178cd5f Merge pull request #45716 from ciiqr/fix_cmd_script_quoting
    - 217791079b some code cleanup (lint errors and escape_argument as _cmd_quote)
    - 1c29bc5a3d fixed quoting of script path in cmd.script
  - 272f912c7c Merge pull request #45719 from bdrung/fix-python3-sphinx-build
    - 179e8fbe73 doc: Do not mock non-existing __qualname__ attribute
    - 971e59ebe2 Drop enforcing new-style object for SaltYamlSafeLoader
  - fc04336c3b Merge pull request #45764 from mchugh19/2017.7
    - 0a7f1a4d75 English better
    - 37e067c7b5 support amazon linux 2 for service module
- **PR** #45861: (rallytime) [oxygen] Merge forward from oxygen.rc1 to oxygen @ *2018-02-08 13:39:59 UTC*

  - 048c18ea42 Merge pull request #45861 from rallytime/merge-oxygen
  - 6d812ac192 Merge branch 'oxygen.rc1' into 'oxygen'

- **PR** #45852: (Giandom) fix-missing-highstate-module-import @ *2018-02-05 15:02:39 UTC*
  - 1bd38fb3b7 Merge pull request #45852 from Giandom/fix-missing-highstate-module-import
  - dc5a8f9233 fix-missing-highstate-module-import
- **PR** #45829: (rallytime) [oxygen] Merge forward from 2017.7 to oxygen @ *2018-02-02 20:20:32 UTC*
  - 5f54ce7b5f Merge pull request #45829 from rallytime/merge-oxygen
  - 34a17819ca Add opts to salt.utils.jid.gen_jid call in minion.py
  - 79d071df9c Merge branch '2017.7' into 'oxygen'
  - f234bf52f4 Merge pull request #45756 from roaldnefs/fix-grafana4-documentation
    - 92979c0b57 Fix grafana4 states documentation
  - 685b683db5 Merge pull request #45801 from rallytime/merge-2017.7
    - 26e992e011 Merge branch '2016.11' into '2017.7'
    - 746386d04c Merge pull request #45794 from vutny/doc-file-state-examples
      - ddfeae6a29 [DOC] Fix code-block rST directive in file state module
    - abc9ece214 Merge pull request #45780 from vutny/doc-pkgrepo-zypper
    - f80c7d8d69 [DOC] Add missing gpgautoimport for pkgrepo.managed
  - c7d319f3bc Merge pull request #45802 from rallytime/merge-2017.7-from-2017.7.3
    - eb48513ba0 Merge branch '2017.7.3' into '2017.7'
    - 1439da8d76 Merge pull request #45755 from terminalmage/issue45743
    - 8af1251c59 salt.crypt: Ensure message is encoded before signing
  - 96e9232cc2 Merge pull request #45761 from gtmanfred/2017.7
  - 280767ed57 generate a jid for cache_jobs on the minion
- **PR** #45819: (Giandom) oxygen-added-highstate-output-to-slack-engine @ *2018-02-01 18:38:42 UTC*
  - 3471796c51 Merge pull request #45819 from Giandom/oxygen-added-highstate-output-to-slack-engine
  - 1af8899a9d oxygen-added-highstate-output-to-slack-engine

28.2.13 Salt 2018.3.2 Release Notes

Version 2018.3.2 is a bugfix release for *2018.3.0*.

The `2018.3.2` release contains only a small number of fixes, which are detailed below.

This release fixes two critical issues.

The first is Issue #48038, which is a critical bug that occurs in a multi-syndic setup where the same job is run multiple times on a minion.

The second issue is #48130. This bug appears in certain setups where the Master reports a Minion time-out, even though the job is still running on the Minion.

Both of these issues have been fixed with this release.

Statistics

- Total Merges: **7**
- Total Issue References: **2**
- Total PR References: **10**
- Contributors: **4** (cro, garethgreenaway, gtmanfred, rallytime)

> **Warning:** If you are using Jinja to dump lists or dictionaries in your SLS files, this will now cause errors in Python 2 since Jinja does not produce YAML-compatible output when strings in the data structures contain unicode types. The dictionary must be passed through a Jinja filter to produce YAML-compatible strings.
>
> The below is an example of invalid SLS:
>
> ```
> /etc/foo.conf:
>   file.mangaged:
>     - source: salt://foo.conf
>     - template: jinja
>     - defaults: {{ mydict }}
> ```
>
> To make it valid, use either one of Salt's own `json` or `yaml` filters:
>
> ```
> /etc/foo.conf:
>   file.mangaged:
>     - source: salt://foo.conf
>     - template: jinja
>     - defaults: {{ mydict | json }}
> ```

Changelog for v2018.3.1..v2018.3.2

Generated at: 2018-06-17 19:17:16 UTC

- **ISSUE** #48130: (rmarchei) Minion timeouts with 2018.3.1 (refs: #48158)
- **PR** #48158: (gtmanfred) always listen when gathering job info @ *2018-06-17 19:04:03 UTC*
  - 521e926458 Merge pull request #48158 from gtmanfred/2018.3.2
  - cecf564433 always listen when gathering job info
- **PR** #48138: (rallytime) Update man pages for 2018.3.2 @ *2018-06-14 21:22:34 UTC*
  - f154545aff Merge pull request #48138 from rallytime/man-pages-2018.3.2
  - 8c340134f5 Update man pages for 2018.3.2
- **PR** #48137: (gtmanfred) [2018.3.2] bootstrap kitchen branch tests with 2017.7.6 @ *2018-06-14 21:20:28 UTC*
  - b49271b76d Merge pull request #48137 from gtmanfred/2018.3.2
  - 6128519e8b bootstrap kitchen branch tests with 2017.7.6
- **PR** #48129: (rallytime) Add release notes for 2018.3.2 @ *2018-06-14 15:48:36 UTC*
  - 21aaf1cbc4 Merge pull request #48129 from rallytime/release-notes-2018.3.2
  - 0b13be0111 Add release notes for 2018.3.2
- **PR** #48100: (rallytime) Back-port #48014 to 2018.3.2 @ *2018-06-14 12:54:52 UTC*
  - **PR** #48014: (cro) Find job pause (refs: #48100)

  - 36b99ae80a Merge pull request #48100 from rallytime/bp-48014
  - 77feccc5c4 Lint: Add blank line
  - 159b052962 One more case where returner doesn't respond
  - 91b45b4cc4 Catch two cases when a returner is not able to be contacted–these would throw a stacktrace.
- **PR** #48099: (rallytime) Back-port #47915 to 2018.3.2 @ *2018-06-14 12:54:23 UTC*
  - **PR** #47915: (garethgreenaway) [2018.3] state runner pause resume kill (refs: #48099)
  - 40c1bfdec9 Merge pull request #48099 from rallytime/bp-47915
  - 3556850058 fixing typo in alias_function call.
  - 4b0ff496fa Some fixes to the set_pause and rm_pause function in the state runner, renaming to in line with the functions in the state module. Including aliases to previous names for back-ward compatibility. Including a soft_kill function to kill running orchestration states. A new test to test soft_kill functionality.
- **ISSUE** #48038: (austinpapp) jobs are not dedup'ing minion side (refs: #48075)
- **PR** #48097: (rallytime) Back-port #48075 to 2018.3.2 @ *2018-06-14 12:52:44 UTC*
  - **PR** #48075: (garethgreenaway) [2017.7] Ensure that the shared list of jids is passed (refs: #48097)
  - 074a97dcfa Merge pull request #48097 from rallytime/bp-48075
  - e4c719b55f Ensure that the shared list of jids is passed when creating the Minion. Fixes an issue when minions are pointed at multiple syndics.

28.2.14 Salt 2018.3.3 Release Notes

Version 2018.3.3 is a security and bugfix release for *2018.3.0*.

Statistics

- Total Merges: **548**
- Total Issue References: **69**
- Total PR References: **341**
- Contributors: **55** (Ch3LL, FedericoCeratto, KaiSforza, L4rS6, Lutseslav, The-Loeki, Vaelatern, admd, aesposito91, asenci, astorath, azelezni, babs, bbczeuz, bbinet, brejoc, cro, daa, dmurphy18, dubb-b, dwoz, eliasp, ezh, garethgreenaway, gtmanfred, isbm, jeduardo, kt97679, kuetrzi, linoplt, lomeroe, lusche, mateiw, max-arnold, maxim-sermin, meaksh, mmulqueen, morganwillcock, mtorromeo, nullify005, paulcollinsiii, pritam-baral, rallytime, rares-pop, rmarchei, rosscdh, sizgiyaev, sjorge, t0fik, terminalmage, travispaul, twangboy, vinian, weswhet, zerthimon)

Warning: If you are using Jinja to dump lists or dictionaries in your SLS files, this will now cause errors in Python 2 since Jinja does not produce YAML-compatible output when strings in the data structures contain unicode types. The dictionary must be passed through a Jinja filter to produce YAML-compatible strings.

The below is an example of invalid SLS:

```
/etc/foo.conf:
  file.mangaged:
    - source: salt://foo.conf
    - template: jinja
    - defaults: {{ mydict }}
```

To make it valid, use either one of Salt's own `json` or `yaml` filters. Another option would be to use Jinja's *tojson* filter.

```
/etc/foo.conf:
  file.mangaged:
    - source: salt://foo.conf
    - template: jinja
    - defaults: {{ mydict | tojson }}
```

Security Fix

CVE-2018-15751 Remote command execution and incorrect access control when using salt-api.

CVE-2018-15750 Directory traversal vulnerability when using salt-api. Allows an attacker to determine what files exist on a server when querying /run or /events.

Credit and thanks for discovery and responsible disclosure: nullbr4in, xcuter, koredge, loupos, blackcon, Naver Business Platform

Changes to win_timezone

Improves timezone detection by using the pytz module.

`timezone.get_offset` and `timezone.get_zonecode` now work properly.

Adds `timezone.list` to list supported timezones in either Windows or Unix format.

New Jinja Filter

The `tojson` filter (from Jinja 2.9 and later) has been ported to Salt, and will be used when this filter is not available. This allows older LTS releases such as CentOS 7 and Ubuntu 14.04 to use this filter.

You can use this filter any time you wish to dump a list or dictionary into an SLS file, to ensure that the result is able to be loaded by the YAML renderer. For example:

```
foo:
  bar.baz:
    - some_arg: {{ mydict | tojson }}
```

MacOSX escape characters with runas

You are now required to escape quotes when using the runas argument with the cmd module on macosx.

Example:

```
cmd.run 'echo '\''h=\"baz\"'\''' runas=macuser
```

Changelog for v2018.3.2..v2018.3.3

Generated at: 2018-09-21 17:45:27 UTC

- **PR** #49662: (dwoz) Fix another bad filename reference in whitelist @ *2018-09-14 22:20:49 UTC*

  - 9d8cc0b3f4 Merge pull request #49662 from dwoz/2018.3.3
  - e109023013 Fix another bad filename reference in whitelist
- **PR** #49655: (dwoz) Fix windows test whitelist errors @ *2018-09-14 20:34:56 UTC*
  - 6391560d57 Merge pull request #49655 from dwoz/2018.3.3
  - 8a4946478e Fix windows test whitelist errors
- **PR** #49641: (rallytime) Back-port #49632 to 2018.3.3 @ *2018-09-13 16:46:02 UTC*
  - **PR** #49632: (garethgreenaway) [2018.3] Fixing integration.states.test_file.FileTest.test_directory_max_depth (refs: #49641)
  - 3fb3ffdb37 Merge pull request #49641 from rallytime/bp-49632
  - d11a400825 Fixing failing test under python 3.7 causaed by changes to how os.makedirs sets initial permissions.
- **PR** #49633: (garethgreenaway) [2018.3.3] Moving test_build_whitespace_split_regex to TestBuildWhitespaceRegex @ *2018-09-13 06:57:01 UTC*
  - 0096cf10b5 Merge pull request #49633 from garethgreenaway/moving_test_into_correct_class
  - 370de07617 Lint: Add extra blank line
  - 27b93fcc68 Moving the test_build_whitespace_split_regex test into the TestBuildWhitespaceRegex class.
- **PR** #49594: (rallytime) Back-port #49580 to 2018.3.3 @ *2018-09-10 19:59:41 UTC*
  - **PR** #49580: (garethgreenaway) [2018.3] Fixing tests for Python 3.7 (refs: #49594)
  - e3a14e3535 Merge pull request #49594 from rallytime/bp-49580
  - 41a2586fc0 Add file coding line at top of file
  - 7df3bebf53 Fixing lint.
  - 5fee38d1db Fixes various tests that were failing under python 3.7.
- **PR** #49589: (rallytime) Update old utils paths to use new utils paths @ *2018-09-10 16:51:31 UTC*
  - 39f9c9c952 Merge pull request #49589 from rallytime/utils-paths
  - 5de2245c11 Update old utils paths to use new utils paths
- **PR** #49550: (rallytime) Back-port #49548 to 2018.3.3 @ *2018-09-07 00:36:05 UTC*
  - **PR** #49548: (garethgreenaway) [2018.3] Disabling State boto tests for Python 3.7+ (refs: #49550)
  - 202da7a94f Merge pull request #49550 from rallytime/bp-49548
  - 180692ccee Disable various boto tests when run under python 3.7 because of //github.com/spulec/moto/issues/1706. which was causing the test suite to hang on unit tests. This PR is disabling the tests in the test_boto_vpc state tests.
- **PR** #49542: (twangboy) Update openssl @ *2018-09-06 16:11:34 UTC*
  - cae2d61568 Merge pull request #49542 from twangboy/fix_osx_build_3
  - fe02b2276f Add 1.0.2p shasum file
  - 5f06dc2762 Fix issues with osx build scripts on 2018.3.3
- **PR** #49536: (rallytime) Back-port #49524 to 2018.3.3 @ *2018-09-06 16:00:00 UTC*
  - **PR** #49524: (garethgreenaway) [2018.3] Disable boto tests under 3.7 (refs: #49536)
  - d9f09da5d9 Merge pull request #49536 from rallytime/bp-49524

  - 9e7203e08a Disable various boto tests when run under python 3.7 because of //github.com/spulec/moto/issues/1706. which was causing the test suite to hang on unit tests.
- **PR** #49535: (Ch3LL) Skip test_virt and pip_state requirements tests on macosx @ *2018-09-06 15:59:38 UTC*
  - cb934bf0b6 Merge pull request #49535 from Ch3LL/skip_pip_mac
  - 50237e9daf Skip test_virt and pip_state requirements tests on macosx
- **PR** #49499: (rallytime) Pin CherryPy version to < 18.0.0 in requirements files for PY2 @ *2018-09-04 18:52:44 UTC*
  - 87d3dfe085 Merge pull request #49499 from rallytime/pin-cherrypy-2018.3.3
  - 9e274335a3 Pin CherryPy version to < 18.0.0 in requirements files for PY2
- **ISSUE** saltstack/salt-jenkins#1075: (Ch3LL) [2018.3.3] arch python3 tests do not finish (refs: #49303, #49451)
- **PR** #49467: (rallytime) Back-port #49451 to 2018.3.3 @ *2018-08-31 17:38:09 UTC*
  - **PR** #49451: (gtmanfred) Handle thread shutdown on system exit (refs: #49467)
  - 39fdacc434 Merge pull request #49467 from rallytime/bp-49451
  - b891a0a8d3 add lock for proxy minion process too
  - 72519878c0 start thread in try block
  - b878f01662 use finally instead of catching baseexception
  - de98be6093 use rlock so blocking can be passed in py2
  - d346b42332 import Callable from collections.abc for python3.7
  - d7a410070a Handle thread shutdown on system exit
- **PR** #49468: (rallytime) Back-port #49291 and #49331 to 2018.3.3 @ *2018-08-31 17:37:30 UTC*
  - **PR** #49331: (dwoz) Use salt.utils to ensure string type (refs: #49468)
  - **PR** #49291: (dwoz) Add dedent that sets line endings (refs: #49468)
  - 37d1455d69 Merge pull request #49468 from rallytime/bp-49291-and-49331
  - 944f8e96c8 Use salt.utils to ensure string type
  - 6c92ed2021 Fix review nits
  - 0e18b157e3 Re-factor dedent to fix warts
  - b5034067f8 Use salt.utils.to_* functions
  - 6399d035a4 Add dedent that sets line endings
- **PR** #49449: (rallytime) Mark status test as flaky @ *2018-08-30 18:10:44 UTC*
  - 0cda22e7a9 Merge pull request #49449 from rallytime/flaky-test
  - 0f322bb39f Mark status test as flaky
- **PR** #49444: (rallytime) Back-port #49299 to 2018.3.3 @ *2018-08-30 18:10:21 UTC*
  - **PR** #49299: (dwoz) Work around cmd.run unicode issues in test for now (refs: #49444)
  - bcc5f1a7c1 Merge pull request #49444 from rallytime/bp-49299
  - b8c5a5bb91 Fix string formatting wart in file state tests
  - 19756022be Fix wart in file state test
  - ba68388342 Work around cmd.run unicode issues in test for now

- **PR** #49448: (rallytime) Back-port #49400 to 2018.3.3 @ *2018-08-30 18:04:29 UTC*
  - **PR** #49400: (rallytime) Mark pillar refresh test as flaky (refs: #49448)
  - 38713e2db9 Merge pull request #49448 from rallytime/bp-49400
  - b953fe0079 Mark pillar refresh test as flaky
- **PR** #49446: (rallytime) Back-port #49356 to 2018.3.3 @ *2018-08-30 18:04:01 UTC*
  - **PR** #49356: (dwoz) Fix tests that use timed_subprocess for py3 (refs: #49446)
  - d6ddcab351 Merge pull request #49446 from rallytime/bp-49356
  - 8022b0c3d6 Fix tests that use timed_subprocess for py3
- **PR** #49445: (rallytime) Back-port #49192 to 2018.3.3 @ *2018-08-30 18:03:44 UTC*
  - **PR** #49192: (dwoz) Test fixes flaky test and unicode environment key/value (refs: #49445)
  - 1a67956c0f Merge pull request #49445 from rallytime/bp-49192
  - 20148d4438 Test fixes
- **PR** #49443: (rallytime) Back-port #49197 to 2018.3.3 @ *2018-08-30 18:03:10 UTC*
  - **PR** #49197: (dwoz) File state line ending fixes (refs: #49443)
  - 33f59d44de Merge pull request #49443 from rallytime/bp-49197
  - 5fe821978e File state line ending fixes
- **PR** #49442: (rallytime) Back-port #49180 and related fixes to 2018.3.3 @ *2018-08-30 18:02:54 UTC*
  - **PR** #49186: (dwoz) Fix typo in #49180 (refs: #49442)
  - **PR** #49180: (dwoz) Cherry-pick test fixes (refs: #49442, #49186)
  - **PR** #49167: (dwoz) Fix remaining file state integration tests (py3) (refs: #49173, #49442)
  - 60758059c3 Merge pull request #49442 from rallytime/bp-49180
  - e155568957 Fix typo
  - 608a1ae7ba Account for file renames
  - e8e6a46a2b Fix directory unit test
  - 2f865c398e Fix is_windows checks
  - 6460f7f217 Account for normalized dirs in unit tests
  - f4b7101a35 Simplify dict keys lookup
  - 942b68bfc8 Fix remaining file state integration tests (py3)
- **PR** #49441: (rallytime) Back-port #49240 to 2018.3.3 @ *2018-08-30 18:02:40 UTC*
  - **PR** #49240: (dwoz) file state test fixes (refs: #49362, #49441)
  - 1f4906346a Merge pull request #49441 from rallytime/bp-49240
  - 22ed452479 Work around listdir encoding issues on py2 windows
  - f5be275835 file state test fixes
- **PR** #49440: (rallytime) Back-port #49258 to 2018.3.3 @ *2018-08-30 18:02:25 UTC*
  - **PR** #49258: (gtmanfred) flaky tests are flaky (refs: #49440)
  - cc27b67a37 Merge pull request #49440 from rallytime/bp-49258

  - 0191af1423 flaky tests are flaky yo
  - 720b671dda mark orchestration state tests as flaky
- **ISSUE** #48880: (damntoken) Can't run cmd.run with UTF-8 chars as arguments / parameters. With custom module. (refs: #49322)
- **PR** #49368: (rallytime) Back-port #49322 to 2018.3.3 @ *2018-08-28 17:15:15 UTC*
  - **PR** #49322: (dwoz) Encode shell commands explicitly. (refs: #49368)
  - af80e64569 Merge pull request #49368 from rallytime/bp-49322
  - 238853b9ec Encode shell commands explicitly.
- **PR** #49363: (rallytime) Back-port #49245 to 2018.3.3 @ *2018-08-28 17:14:18 UTC*
  - **PR** #49245: (dwoz) Skip grep unit tests on windows (refs: #49363)
  - 0fee3e8786 Merge pull request #49363 from rallytime/bp-49245
  - cc606509d0 Fix is_windows call - use the right path
  - 5488fbea38 Skip grep unit tests on windows
- **PR** #49361: (rallytime) Back-port #49244 to 2018.3.3 @ *2018-08-28 17:12:58 UTC*
  - **PR** #49244: (dwoz) Test fixes: unit.fileserver.test_gitfs (refs: #49361)
  - dbcd2fc726 Merge pull request #49361 from rallytime/bp-49244
  - e0909d3a25 Simplify by using to_unicode helper
  - 4723c69092 Older GitPython version do not have a close method
  - d5fecba716 Fix up fileserver.test_gitfs tests on windows
  - 4b688f6347 Remove unicode filenames on windows python 2
- **PR** #49362: (rallytime) Back-port #49240 to 2018.3.3 @ *2018-08-28 17:00:25 UTC*
  - **PR** #49240: (dwoz) file state test fixes (refs: #49362, #49441)
  - b4a1e1d365 Merge pull request #49362 from rallytime/bp-49240
  - 16ca5b9694 Work around listdir encoding issues on py2 windows
  - 7650208dbc file state test fixes
- **PR** #49365: (rallytime) Back-port #49270 to 2018.3.3 @ *2018-08-28 16:56:23 UTC*
  - **PR** #49270: (dwoz) Add async helper to test_sock_path_len (refs: #49365)
  - 9ca9a775ff Merge pull request #49365 from rallytime/bp-49270
  - 1b3f37a8b8 Add async helper to test_sock_path_len
- **PR** #49364: (rallytime) Back-port #49243 to 2018.3.3 @ *2018-08-28 16:55:55 UTC*
  - **PR** #49243: (dwoz) Revert newline translation change (refs: #49364)
  - bbff57da16 Merge pull request #49364 from rallytime/bp-49243
  - 5db77c6229 Revert newline translation change
- **PR** #49347: (Ch3LL) [2018.3.3] Backport #49345 @ *2018-08-28 01:54:00 UTC*
  - **PR** #49345: (gtmanfred) upgrade including linux kernels (refs: #49347)
  - 74b78835b3 Merge pull request #49347 from Ch3LL/bp_49345_2018.3.3

  - 7bf5ba83c8 upgrade including linux kernels
- **PR** #49323: (Ch3LL) Skip nonexistent branch test for git versions <1.7.10 @ *2018-08-25 19:52:48 UTC*
  - be6691d91b Merge pull request #49323 from Ch3LL/skip_git
  - ee3d32f74e Skip nonexistent branch for git versions <1.7.10
- **ISSUE** #32737: (Lothiraldan) No support for compound matcher in external auth configuration (refs: #49236)
- **PR** #49313: (rallytime) Back-port #49236 to 2018.3.3 @ *2018-08-24 20:59:16 UTC*
  - **PR** #49236: (terminalmage) Allow compound matching in eauth config expressions (refs: #49313)
  - 64d7b0e4c6 Merge pull request #49313 from rallytime/bp-49236
  - 1a5ef996e3 Add 'minion_data_cache: True' to mocked opts for minions unit tests
  - 549f5d5a86 Allow compound matching in eauth config expressions
- **ISSUE** saltstack/salt-jenkins#1075: (Ch3LL) [2018.3.3] arch python3 tests do not finish (refs: #49303, #49451)
- **PR** #49311: (rallytime) Back-port #49303 to 2018.3.3 @ *2018-08-24 17:48:23 UTC*
  - **PR** #49303: (gtmanfred) use os._exit instead of sys.exit when daemonizing (refs: #49311)
  - 7a89a4c8aa Merge pull request #49311 from rallytime/bp-49303
  - 3fe1387751 use os._exit instead of sys.exit when daemonizing
- **PR** #49294: (Ch3LL) Move run_function call from __init__ to setup @ *2018-08-24 12:41:35 UTC*
  - 7bb356f11e Merge pull request #49294 from Ch3LL/fed_28_tests
  - 0a5d44a3db Move run_function call from __init__ to setup
- **PR** #49302: (twangboy) Fix installer 2018.3.3 @ *2018-08-24 12:41:14 UTC*
  - a607f9332f Merge pull request #49302 from twangboy/fix_installer_2018.3.3
  - 68fd37575e Fix erroneous NSSM reference
  - 103f2c289e Remove delete vcredist line
  - 007a16638e Bring installer updates from 2017.7.8 to 2018.3.3
- **PR** #49241: (terminalmage) Don't silently catch SystemExit @ *2018-08-22 12:57:58 UTC*
  - bc0b4ac513 Merge pull request #49241 from terminalmage/salt-jenkins-1078
  - 08d144f2c7 Don't silently catch SystemExit
  - c0fdb818f7 Don't use a bare except!
- **PR** #49239: (Ch3LL) Use yaml's safe_dump in windows ec2 tests @ *2018-08-22 08:48:13 UTC*
  - 0fb9ccf60a Merge pull request #49239 from Ch3LL/win_yaml_test
  - f5b42dbb24 import salt.utils.yaml
  - 67290eaff7 Use yaml's safe_dump in windows ec2 tests
- **PR** #49182: (terminalmage) Fix hanging syndic test @ *2018-08-18 12:10:32 UTC*
  - 134f125b96 Merge pull request #49182 from terminalmage/salt-jenkins-1078
  - a2d2cd317b Fix hanging syndic test
- **PR** #49172: (Ch3LL) [2018.3.3] cherry pick #49118 @ *2018-08-17 20:54:05 UTC*
  - **PR** #49118: (dwoz) Multiple fixes for integration.states.test_file (refs: #49172)

  - **PR** #49088: (dwoz) Multiple file state test fixes (refs: #49118)
  - **PR** #49087: (dwoz) Filter out scheme's that are not valid (refs: #49118)
  - b3a247bfbb Merge pull request #49172 from Ch3LL/bp-49118
  - ce5e17bdbd update is_windows salt.utils to correct path
  - 3fef112409 Multiple fixes for integration.states.test_file
- **PR** #49173: (Ch3LL) [2018.3.3] cherry pick #49167 @ *2018-08-17 20:30:27 UTC*
  - **PR** #49167: (dwoz) Fix remaining file state integration tests (py3) (refs: #49173, #49442)
  - 85ffc8db87 Merge pull request #49173 from Ch3LL/bp-49167
  - a1a298a13c Simplify dict keys lookup
  - 3d26affa10 Fix remaining file state integration tests (py3)
- **PR** #49171: (Ch3LL) [2018.3.3] cherry pick #49103 @ *2018-08-17 20:23:32 UTC*
  - **PR** #49103: (dwoz) Install the launcher so we can execute py files (refs: #49171)
  - ee54ea5f73 Merge pull request #49171 from Ch3LL/bp-49103
  - 05a2b91fb2 Install the launcher so we can execute py files
- **PR** #49132: (Ch3LL) [2018.3.3] backport PR #49062 @ *2018-08-17 14:51:50 UTC*
  - **PR** #49062: (weswhet) fix memoize on available macOS services (refs: #49132)
  - 58034c9dc3 Merge pull request #49132 from Ch3LL/bp-49062
  - 990fdb6a52 decorator link fix, updating context names, as well as updating macutils tests for latest changes
  - 3ab5d282be fixing an issue with memoize on macOS services, switching to useing __context__ instead
- **PR** #49142: (Ch3LL) Remove -Z script_arg for cloud tests @ *2018-08-16 16:12:59 UTC*
  - 01f8f83cf0 Merge pull request #49142 from Ch3LL/rm_z_arg
  - a1ef6a88a6 Remove -Z script_arg for cloud tests
- **PR** #49137: (Ch3LL) [2018.3.3] Update bootstrap script to latest release (2018.08.15) @ *2018-08-15 19:49:22 UTC*
  - a03828884b Merge pull request #49137 from Ch3LL/bootstrap_2018.3.3
  - eb9a612096 [2018.3.3] Update bootstrap script to latest release (2018.08.15)
- **PR** #49110: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-08-14 17:11:25 UTC*
  - b412bff534 Merge pull request #49110 from rallytime/merge-2018.3
  - 4649f60209 Update old utils paths with new paths
  - 49c2a784bb Merge branch '2017.7' into '2018.3'
    - d46e23f4c2 Merge pull request #49109 from rallytime/merge-2017.7
      - 3c5527f300 Merge branch '2017.7.8' into '2017.7'
      - 01cf2c71ff Merge pull request #49051 from rallytime/bp-49046
      - fc0817cb35 The osfullname grain differs when using Python2 vs Python3, swapping this out for the "OS" grain which is consistent.
    - 8954dd269f Merge pull request #49090 from dwoz/file_line_fix
      - 0dca634e3c Search for (but do not include) CRLF line ending

- b99edc3dce Merge pull request #49088 from dwoz/file_hash_diffs
  - 054afb8197 Multiple file state test fixes
- 274977b6c4 Merge pull request #49087 from dwoz/url_file_test_fixes
  - 632b65f975 Filter out scheme's that are not valid
- db5fb3232a Merge pull request #49086 from rallytime/new-doc-img
  - 90954203cb Update the DOCBANNER image for saltconf
- af6ec1dce2 Merge pull request #49045 from twangboy/fix_43164
  - 589456d08f Fix lint errors
  - e79243566d Add rallytime's suggestions
  - d1ae6b3d6d Fix docs for the registry module and state
- a840fea1a1 Merge pull request #49083 from rallytime/bootstrap-2017.7
  - 358e14cdac [2017.7] Update bootstrap script to latest release
- 9db6cd5654 Merge pull request #49059 from twangboy/fix_37984
  - 7ed45b5b00 Remove import
  - 7fb1edb469 Fix docs to clarify uptime output on Windows
- f9db72f00c Merge pull request #49061 from saltstack/revert-48982-new_logo_2017.7
  - 98a8da728b Revert "New logo 2017.7 (WIP)"
- bbcd1869ec Merge pull request #48982 from newwebash/new_logo_2017.7
  - 5c1c311f77 Update Salt Conf ad
  - 72dc63c426 Merge branch '2017.7' of https://github.com/saltstack/salt into new_logo_2017.7
  - 9f4e78a7b9 Revert "Update saltconf ad"
  - 3ab8cdb882 Update saltconf ad

- **ISSUE** #48665: (dgengtek) salt-api auth ldap generates a valid token when using bindpw and an invalid request (refs: #48901)
- **PR** #48901: (garethgreenaway) [2018.3] fix to auth/ldap.py @ *2018-08-14 16:10:19 UTC*
  - e78fc0e0eb Merge pull request #48901 from garethgreenaway/48665_auth_ldap_valid_token_failed_auth
  - d4e4f2e803 Fixing a typo in a comment.
  - 2eb167ea42 Fixing issue when a valid token is generated even when invalid user credentials are passed. This change verifies that the binddn credentials are valid, then verifies that the username & password (if not None) are also valid.
- **ISSUE** #49081: (frogunder) Fluorine - I see error/traceback when running minion in debug mode (refs: #49085)
- **PR** #49099: (rallytime) Back-port #49085 to 2018.3 @ *2018-08-14 12:42:51 UTC*
  - **PR** #49085: (gtmanfred) fix async call to process manager (refs: #49099)
  - 484c1e0123 Merge pull request #49099 from rallytime/bp-49085
  - 024d9cb843 fix async call to process manager
- **ISSUE** #49018: (Ch3LL) add MasterPillarUtil tests (refs: #49034)
- **PR** #49071: (rallytime) Back-port #49034 to 2018.3 @ *2018-08-13 20:15:31 UTC*

  - **PR** #49034: (garethgreenaway) [fluorine] Adding tests for MasterPillarUtil (refs: #49071)
  - bc033da677 Merge pull request #49071 from rallytime/bp-49034
  - 8108a4d31a Adding some tests for the grains, pillar and mine functions in the cache runner. These will also ensure that the relevant functions in salt.utils.master.MasterPillarUtil are functioning properly.
- **PR** #49077: (rallytime) Back-port #49075 to 2018.3 @ *2018-08-13 20:00:00 UTC*
  - **PR** #49075: (gtmanfred) fix last async issue (refs: #49077)
  - 90c2f026b3 Merge pull request #49077 from rallytime/bp-49075
  - 5e07b8306b fix last async issue
- **PR** #49096: (rallytime) Update the DOCBANNER image for saltconf @ *2018-08-13 19:59:39 UTC*
  - 6942ef1102 Merge pull request #49096 from rallytime/new-doc-img-2018.3
  - 75080705ce Update the DOCBANNER image for saltconf
- **PR** #49055: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-08-13 19:20:11 UTC*
  - 0e1ed7b923 Merge pull request #49055 from rallytime/merge-2018.3
  - 83a15cb623 Add optimization_order config to mocked opts
  - ea6883ee55 Update old utils paths to new utils paths
  - 201031fa8a Merge branch '2017.7' into '2018.3'
    - 0d2a495378 Merge pull request #49050 from rallytime/merge-2017.7
      - ed6fb8b739 Merge branch '2017.7.8' into '2017.7'
      - 189e28691f Merge pull request #49044 from Ch3LL/vultr_test
      - 0dedfae4d1 increase timeout on vultr cloud integration test
      - c606a32cf2 Merge pull request #49042 from rallytime/flaky-jinja-test
      - a43d9b4ba6 Mark a jinja template test as flaky
      - 6415b6f73b Merge pull request #49041 from Ch3LL/ec2_fix
      - cf7f2459b8 [2017.7.8] backport PR #48212
      - 1db036406b Merge pull request #49030 from rallytime/update-client-tests
      - f08ee6c6ae Update netapi client tests
    - 70be9ac9b1 Merge pull request #49046 from garethgreenaway/1022_fixing_test_failing_py3_Debian_test
      - fa2339bb91 The osfullname grain differs when using Python2 vs Python3, swapping this out for the "OS" grain which is consistent.
    - 69c9b0afb1 Merge pull request #49036 from cspeidel/doc-fix-netyang
      - f7ac085eb5 minor doc fix in netyang.py
    - c5f3fd6b2b Merge pull request #48997 from AVeenstra/patch-1
      - 0e535f2c8c Unpacking dict with the six iterator
      - 5e82685b89 Python 3 related fix in highstate_return.py
    - c06a3cf531 Merge pull request #49021 from rallytime/merge-2017.7
      - ca82b3d57a Merge branch '2017.7.8' into merge-2017.7

      - f73ba21bc7 Merge pull request #49024 from rallytime/fix-deb-test
      - a9c16d9137 Fix test error in test_compiler jinja check
      - c6f8429e41 Merge branch '2017.7.8' into '2017.7'
      - 6b6d68c615 Merge pull request #49020 from rallytime/fix-grains-test
      - f72a3ac6be Make grains integration test more robust
      - 92d6c25c7f Merge pull request #49002 from rallytime/skip-jinja-deep-error-test
      - 23b66ef8bb Skip test_jinja_deep_error on Debian 8
      - 41d9f11eb3 Merge pull request #48999 from rallytime/fix-boto-test
      - d0136b1be5 Update expected return value in boto test
    - 2ddba6391a Merge pull request #49010 from Ch3LL/univention
      - c09bce9afe Add univention OS to debian os_family mapping
    - 41bd36842e Merge pull request #48636 from terminalmage/loader-fixes
      - 0441cd56ef Add optimization_order to mocking
      - 2256fad320 Process pycache files after .py files
      - a78663a301 Lint
      - 0728b5f8b6 Add note about Python 3.5+ only support for optimization_order
      - 764969ce08 Add a test to confirm that .py files are still loaded correctly
      - 0f60beb6b6 Fix bad copypasta
      - fbcd142ea2 Add configuration docs for optimization_order config option
      - 8af2d580f3 Only compile the suffix_order/map once per LazyLoader instance
      - 4b95e5f313 Don't put __pycache__ dir in the file list
      - e7c10196da Quiet the byte compiling for PY2
      - 9338370477 Add unit tests for loader optimization levels
      - c3622933c1 PY3: Support different optimization levels
      - 6fc8da5bab Add optimization_order config option with default value
      - dfe423a1e0 Remove commented-out log message

- **PR** #49084: (rallytime) [2018.3] Update bootstrap script to latest release @ *2018-08-13 18:14:42 UTC*
  - db7ec46875 Merge pull request #49084 from rallytime/bootstrap-2018.3
  - 28dc8ce954 [2017.7] Update bootstrap script to latest release
- **PR** #49052: (isbm) Python 3.7 support (backport 2018.3) @ *2018-08-13 13:06:51 UTC*
  - b0d5acbe0d Merge pull request #49052 from isbm/isbm-python37-support-2018.3
  - 4386a9ca1f Merge branch '2018.3' into isbm-python37-support-2018.3
  - bc85a5fa98 Fix configuration setting
  - 47078a300f Remove async keyword, moving it into the kwargs.
  - 31dccc4fdf fix unicode literals
  - 990936992c Keep runner API unchanged

  - 46bafcafef Fix nag-message
  - 5c887ac2ae Support original API
  - 2c22e794d4 Use kwargs instead of directly named parameters
  - 7d095491d9 Update docstring
  - 9807e8dbb7 Add 'async' backward compatibility
  - e8608aa9be Revert api call: it is about "functionname_async" suffix.
  - 0543578336 Deprecate 'async' parameter in Mandrill API
  - 1107de0dec Lintfix: PEP8 requires two empty lines
  - 2a18e335af Fix function signatures in Cassandra module
  - 603f94e2a3 Cleanup docstrings at module level
  - b9718d3a09 Fix log error/info/warning and exception messages
  - 6e77aff69a Fix local opts from CLI
  - 493e48ddb5 Remove internal variables/properties with the reserved words
  - be07f64bc4 Change internal function signatures to avoid reserved word
  - 7d095e0b26 Rename async function to asynchronous
  - 35eaebb8a4 Fix docstrings
  - 7a597f19b5 Fix comments
  - 69920366ae Fix CLI config
  - 173f3d7aa8 Fix docstring typo
  - b7da571624 Fix imports
  - 7ec3954bef Rename module to full wording
- **ISSUE** #48557: (whytewolf) file.line in python3 on windows 2012 r2 is adding extra CR line endings. (refs: #49026)
- **PR** #49026: (dwoz) Fix file.line line endings @ *2018-08-13 13:05:43 UTC*
  - 338ecb70ef Merge pull request #49026 from dwoz/issue_48557
  - a4d22fda2f Merge remote-tracking branch 'origin/issue_48557' into issue_48557
    - 702e76cfc9 Merge branch '2018.3' into issue_48557
  - 6cf93e8fc4 Skip newline in binary mode
  - 2cf80c1595 Fix linter errors
  - baf291b4c8 Fix file.line line endings
- **PR** #49038: (gtmanfred) disable enable_ssh_minions to see if it is slowing down 2018.3 tests @ *2018-08-10 18:07:59 UTC*
  - 602fed7806 Merge pull request #49038 from gtmanfred/slowtest
  - 6bc44c91c4 remove localhost from tests
  - 7b168a5aac disable enable_ssh_minions to see if it is slowing down 2018.3 tests
- **ISSUE** #48996: (jils2013) file.get_diff not work on version:2018.3.2 (refs: #49033)
- **PR** #49033: (terminalmage) Fix file.get_diff for remote files @ *2018-08-09 21:06:53 UTC*

  - 4eeb75f028 Merge pull request #49033 from terminalmage/issue48996
  - 163aea71c8 Lint
  - d6e5038022 Fix file.get_diff for remote files
- **ISSUE** #48856: (travispaul) Salt fails to start on NetBSD 8 (refs: #48926)
- **PR** #48926: (travispaul) Handle ifconfig output differently for NetBSD >= 8.0 @ *2018-08-09 20:07:44 UTC*
  - b24c96a292 Merge pull request #48926 from travispaul/fix-netbsd-8-new-ifconfig
  - d59b6d8269 Add unit test for NetBSD 8 ifconfig changes
  - 80f8a667d1 Handle ifconfig output differently for NetBSD >= 8.0
- **PR** #48803: (dmurphy18) Support for execution modules and states mount on AIX @ *2018-08-09 17:51:55 UTC*
  - 7d6b9ed0a5 Merge pull request #48803 from dmurphy18/aix_filesystems
  - 92818f816b Fixes for testing
  - 77dd7a1743 Adjust unit tests
  - eaed033cde Updated for review comments and adjusted locking when writing files
  - 5bf9e6085b Update due to review comments
  - 9fb5641dc7 Mount unit tests for modules and states for AIX
  - b28f427432 Save off work after laptop issue
  - e5c2741fe7 module and states mount support for AIX
  - b701e16ad0 Further updates to support for mount on AIX
  - 210076276f Initial support for handling /etc/filesystems on AIX
- **ISSUE** #48496: (icy) Salt-key unable to delete the certificate Test=True (refs: #48929)
- **PR** #48929: (terminalmage) 2 salt-key fixes @ *2018-08-09 17:50:07 UTC*
  - 2a38905a8a Merge pull request #48929 from terminalmage/issue48496
  - 7ac10f9eac 2 salt-key fixes
- **ISSUE** #47481: (whytewolf) Oxygen: task.add_trigger with start_date stating strptime needs a string not int. (refs: #49022)
- **PR** #49022: (dwoz) Document time format idiosyncrasies @ *2018-08-09 17:47:12 UTC*
  - bb9d23bfa1 Merge pull request #49022 from dwoz/47481_docs
  - 313a3d93d6 Document time format idiosyncrasies
- **PR** #48932: (twangboy) Fix pkg.install when pkg already installed @ *2018-08-09 17:45:51 UTC*
  - 9b6a9ff4f1 Merge pull request #48932 from twangboy/fix_win_repo
  - 075ea29d29 remove .lower(), fix debug messages
  - d7c2f476ac Remove current: version for latest as well
  - 522ac26459 Fix typo in code comment
  - 47b2898a85 return empty dict on no changes
  - 6532706d2f Make the tests run on Linux
  - cfe55a391a Add tests for pkg.install output

    - 8ec058f498 Clarify code comment
    - 8af2cfd54a Fix issues where current is not returned

- **PR** #49011: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-08-09 17:34:18 UTC*
    - d7b7a92ef6 Merge pull request #49011 from rallytime/merge-2018.3
    - 02670969d0 Apply fix to ssh init file that was there before merge
    - 7e1f7915e4 Revert "Allow for not being prompted to supply a password to deploy keys to a minion with salt-ssh"
    - 00416d54d1 Revert "add key-deploy test"
    - c26fa0d6c4 Mark orchestration test as expensive
    - fceb6d8d6f Update old utils paths to use new paths
    - 3e6445a9d6 Merge branch '2017.7' into '2018.3'
        - b49eeca6e9 Merge pull request #49003 from rallytime/boto-test-2017.7
            - 1a23bb233f Update expected return value in boto test
        - 51eed1fdfb Merge pull request #48988 from rallytime/merge-2017.7
            - df8699e2e7 Merge branch '2017.7.8' into '2017.7'
            - a1e54634dc Merge pull request #48976 from rallytime/tornado
            - 0bd838ab6c Skip unreliable tornado tests
            - 22713be9c1 Merge pull request #48979 from rallytime/bp-48959
            - 03aa0e49b0 Merge pull request #48970 from Ch3LL/back_48962
            - 3ce1b8a3c9 Update the elif block to only be true for versions below Debian 9.
            - e6cea5e3c7 Merge pull request #48968 from rallytime/man-pages
            - 64fe3be41a Update man pages for 2017.7.8 release
        - 10fd4661ff Merge pull request #48978 from gtmanfred/2017.7
            - 6108363f82 clean up gemfile
        - 5b2423e527 Merge pull request #48959 from rallytime/flaky-tests
            - aaf986d728 Mark one grains test as flaky & convert to pytest notation
            - e7e5abcf48 Mark 2 matcher tests as flaky
        - 79994ecab4 Merge pull request #48962 from garethgreenaway/1022_test_service_disable_debian_part_deux
            - 9e71551b36 Update the elif block to only be true for versions below Debian 9.
        - 1a1bda00cb Merge pull request #48960 from dwoz/block_replace_tests
            - 94ac2b4fc7 Multiple block replace test fixes
        - 93b862f350 Merge pull request #48957 from whytewolf/beacons_log_doc_change
            - a77fd16869 Update salt.beacons.log to reflect that re module is used for matching.
        - 0245cffb07 Merge pull request #48955 from terminalmage/service-systemd
            - 23f87bd536 Don't load service.py if minion is running a non-sysvinit init system
        - 848d583438 Merge pull request #48950 from KaiSforza/kitchenfix-2017.7

      - 5242cb143a Added a quote to kitchen Jenkinsfiles
    - 928d688d65 Merge pull request #48943 from rallytime/flaky-tests
      - 668da57ab9 Mark some shell and runner integration tests as flaky
    - cd42510d3a Merge pull request #48940 from rallytime/bp-48852
      - fa4ef92e79 Record all the artifacts from the build
    - 43649a68be Merge pull request #48935 from garethgreenaway/1045_test_pkg_015_installed_held_centos
      - 0bb10107b6 Merge branch '2017.7' into 1045_test_pkg_015_installed_held_centos
      - 24d5e6a22f Fixing the test_pkg_015_installed_held test to be able to successfully run on CentOS
    - 2421e2a570 Merge pull request #47100 from gtmanfred/ssh
      - 5b443af7ae add key-deploy test
      - a131c9beeb Allow for not being prompted to supply a password to deploy keys to a minion with salt-ssh
    - d541bd6446 Merge pull request #48891 from damon-atkins/2017.7_win_pkg.list_pkgs_not_found
      - a4af1dbfb1 Fix win_pkg issues introduced Jan 2018. If DisplayVersion does not exist it should return version as "Not Found"
    - 5f6a56f5dc Merge pull request #48896 from rallytime/bp-48730
      - 57aa204c9d Merge branch '2017.7' into bp-48730
      - 4995922584 Forgot variable in signature
      - 0503bc18b6 Fix batch install using pkgng
    - 0c64bba865 Merge pull request #48933 from garethgreenaway/1022_debian_8_failing_service-test
      - 280d1d2ad2 Fixing failing test, integration.modules.test_service.ServiceModuleTest.test_service_disable_does on Debian 8 and higher.
    - 0c3d2c6a09 Merge pull request #48922 from rallytime/cache-doc-error
      - 8ca89df7e8 Update backticks on job_cache docs
    - 8d1fc4f8e5 Merge pull request #48866 from Ch3LL/cmd_win_tests
      - 905da13653 Merge branch '2017.7' into cmd_win_tests
    - 57d58e7541 Merge pull request #48920 from rallytime/bp-48904-2017.7
      - a55f92954a No rehashing in parallel
      - 3be11e06fe Add docs for new escape kwarg
      - 391bb8a411 use a specific path for just the cmd._run call
      - 62c66ba489 make sure we lower the check on shell
      - 9312a993a5 Add cmd module integration tests for windows and fix space in path issue
- **ISSUE** #48123: (c-wicklein) file.directory with recurse fails when it encounters a broken symbolic link (refs: #48985)
- **PR** #48985: (garethgreenaway) [2018.3] Fix to salt/modules/file.py @ *2018-08-09 15:03:09 UTC*
  - 3325b7d4c0 Merge pull request #48985 from garethgreenaway/48123_file_directory_recurse_fails_broken_symlink

  - 10c4eca206 Merge branch '2018.3' into 48123_file_directory_recurse_fails_broken_symlink
  - a404cc030f Fixing the issue when using the file.directory state with recurse if the directory contains a broken symbolic link. This fix adds an additional conditional, is_link, before running lsattr since lsattr does not work on symlinks and causes issues when that symlink is broken.
- **ISSUE** #47695: (AmbicaY) Continuous error in the proxy minion logs (refs: #49019)
- **PR** #49019: (garethgreenaway) [2018.3] Fix to scheduler when global enabled key is present @ *2018-08-09 13:06:26 UTC*
  - d353c02a8c Merge pull request #49019 from garethgreenaway/47695_fixing_scheduler_bug_when_enabled_is_present
  - aff1b8f6d4 Lint.
  - 8935c08141 Fixing a bug that occurs if the "enabled" key is present in the scheduler items dictionary. Adding a test to ensure scheduler runs as expected when that key is present.
- **PR** #49023: (The-Loeki) Salt SSH appends IdentityFile=agent-forwarding @ *2018-08-09 12:55:59 UTC*
  - a56bc7ffb9 Merge pull request #49023 from The-Loeki/patch-1
  - 8b53571c70 Salt SSH appends IdentityFile=agent-forwarding
- **PR** #48981: (Ch3LL) Add warning to mac runas docs about escaping characters @ *2018-08-08 19:43:43 UTC*
  - 4590494b50 Merge pull request #48981 from Ch3LL/mac_runas_quote
  - 8269b55b84 remove unnecessary spaces in cmdmod.py docs
  - 757daf7d7e add runas macosx warning in 2018.3.3 release notes
  - 4e9e985b14 Add warning to mac runas docs about escaping characters
- **PR** #49004: (rallytime) Port #48999 to 2018.3 @ *2018-08-08 15:32:46 UTC*
  - **PR** #48999: (rallytime) Update expected return value in boto test (refs: #49004, #49003)
  - b87bf905c2 Merge pull request #49004 from rallytime/boto-test-2018.3
  - 9f0b9a1073 Update expected return value in boto test
- **PR** #48987: (twangboy) Fix issue with win_iis.create_cert_binding @ *2018-08-08 13:29:06 UTC*
  - 630f61e625 Merge pull request #48987 from twangboy/fix_48786
  - 86d0836f90 Fix issue with iterating over None
- **ISSUE** #48777: (jonasgit) file.recurse fails on file contents encoding not utf-8 (refs: #48934)
- **PR** #48934: (terminalmage) Properly handle latin-1 encoding in file diffs @ *2018-08-07 21:02:24 UTC*
  - ab1a713bc3 Merge pull request #48934 from terminalmage/issue48777
  - 52c64e4d51 Fix to_unicode test
  - e2d19f40b6 Only try latin-1 from get_diff instead of by default
  - d39fa889f3 Add stringutils.get_diff to mocks
  - 5b191c9120 Fix incorrect use of __salt__ when __utils__ is needed
  - 53ba10ad5f Skip pylint false-positives
  - f14f4dae22 Add unit test for latin-1 fallback, multi-encoding
  - 906644a80f PY3 scoping fix
  - 726dd4331f Add integration test for latin-1 file diffs

  - 2dd1f31d23 Use BASE_FILES instead of redundant STATE_DIR
  - 612ffb5fe8 Use new get_diff helper in file module
  - c632265802 Make to_unicode/to_str/to_bytes helpers attempt latin-1
  - 2a0cb49b01 Add get_diff to salt.utils.stringutils
- **ISSUE** #47766: (zerthimon) salt-cloud: openstack driver: crash on instance creation (refs: #48956)
- **PR** #48956: (gtmanfred) if booted from volume, use string from image @ *2018-08-07 16:39:45 UTC*
  - bad995462c Merge pull request #48956 from gtmanfred/openstack
  - 4cb1636c4b if booted from volume, use string from image
- **ISSUE** #48306: (davidscholberg) Documentation update for custom returners used for master job cache (refs: #48319)
- **PR** #48319: (gtmanfred) don't break older returners right now @ *2018-08-06 13:19:43 UTC*
  - 71f587edd7 Merge pull request #48319 from gtmanfred/jid
  - 084e7f0ef4 Merge branch '2018.3' into jid
  - 60661b4cd0 document minions required on save_load
  - a8c243071d document passing in minions
  - d49381c0b6 don't break save_load backwards compat
  - cfcacf953a don't break older returners right now
- **PR** #48941: (rallytime) Back-port #48912 to 2018.3 @ *2018-08-06 13:09:38 UTC*
  - **PR** #48912: (maxim-sermin) make jboss7.py compatible with 2018.3.0 and later (refs: #48941)
  - 1623e53ef4 Merge pull request #48941 from rallytime/bp-48912
  - 3f55b2f89e make jboss7.py compatible with 2018.3.0 and later
- **PR** #48897: (rallytime) Back-port #48863 to 2018.3 @ *2018-08-05 19:15:51 UTC*
  - **PR** #48863: (admd) Decode file contents for python2 as well (refs: #48897)
  - 070af9d925 Merge pull request #48897 from rallytime/bp-48863
  - 0ee1940232 Merge branch '2018.3' into bp-48863
  - f40b966d99 decode file contents for python2 as well
- **PR** #48928: (Ch3LL) fix test_runas integration test for macosx @ *2018-08-05 14:24:13 UTC*
  - abd7f1312d Merge pull request #48928 from Ch3LL/mac_runas
  - 3d6455dbcd remove unecessary comment in setup
  - 8e30db0217 move destructivetest to testname
  - 708dec8740 add destructivetest to setup and teardown for runas test
  - c0730ff968 fix test_runas integration test for macosx
- **PR** #48899: (Ch3LL) remove base_top_file file in teardown and add sleep @ *2018-08-05 14:23:42 UTC*
  - dbd300ff70 Merge pull request #48899 from Ch3LL/unit_state
  - 0272cadff8 Merge branch '2018.3' into unit_state
  - 5896c7fc36 remove base_top_file file in teardown and add sleep

- **PR** #48898: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-08-05 14:23:19 UTC*
  - ffa1fcc682 Merge pull request #48898 from rallytime/merge-2018.3
  - 1f093cef4a Merge branch '2018.3' into merge-2018.3
  - b3a5aa49d7 Merge branch '2017.7' into '2018.3'
    - e79ccb35b1 Merge pull request #48876 from Vaelatern/make-network-learning-optional
      - a4905b0e5d Make IP_LEARNING optional
    - f8bfab4f9f Merge pull request #48890 from Ch3LL/cmd_test
      - acda4ed9ab add sleep in test_cmd integration test
    - 52722f6ded Merge pull request #48885 from rallytime/flaky-tests-shadow
      - da871a2d57 Mark shadow module integration test as flaky
    - bf0895656e Merge pull request #48884 from rallytime/flaky-tests-matchers
      - 38d9eae537 Merge branch '2017.7' into flaky-tests-matchers
    - a567666938 Merge pull request #48868 from terminalmage/fix-loader-race
      - 5f1169b9a2 Fix race when SIGTERM/SIGINT received while lazyloading a module
    - 0ca0b6f2f2 Merge pull request #48883 from terminalmage/salt-jenkins-1023
      - c61f75cb50 Fix failing git worktree tests
      - 3660dff13c Separate compound matcher tests into individual tests
    - a84f5155a1 Merge pull request #48869 from Ch3LL/mac_shell_tests
      - 3734b1ec89 Catch socket.error exception in testprogram
    - 7f56b8bf44 Merge pull request #48867 from rallytime/skip-tornado-test
      - 7320aa9104 Skip unreliable tornado test.
- **PR** #48893: (Ch3LL) handle when ca_certs is None in utils.http @ *2018-08-03 20:30:48 UTC*
  - 848e26ed0b Merge pull request #48893 from Ch3LL/http_tests
  - 16d251f7ff Merge branch '2018.3' into http_tests
  - 5674dddb2a handle when ca_certs is None in utils.http
- **PR** #48903: (Vaelatern) Wrap ElementTree.tostring to make strs, not bytes @ *2018-08-03 19:12:34 UTC*
  - **PR** #48877: (Vaelatern) Wrap ElementTree.tostring to make strs, not bytes (refs: #48903)
  - f1adf56017 Merge pull request #48903 from Vaelatern/improve-salt-cloud-python3-virt
  - c45447fe0b Wrap ElementTree.tostring to make strs, not bytes
- **PR** #48900: (Ch3LL) skip getpwnam check on mac in unit test_cmdmod @ *2018-08-03 16:38:51 UTC*
  - d23471262d Merge pull request #48900 from Ch3LL/cmd_unit
  - b82dd708be skip getpwnam check on mac in unit test_cmdmod
- **PR** #48921: (rallytime) Back-port #48904 to 2018.3 @ *2018-08-03 15:17:15 UTC*
  - **PR** #48904: (KaiSforza) No rehashing in parallel (refs: #48921, #48920)
  - d622a49b58 Merge pull request #48921 from rallytime/bp-48904
  - b504625f61 No rehashing in parallel

- **PR** #48877: (Vaelatern) Wrap ElementTree.tostring to make strs, not bytes (refs: #48903) @ *2018-08-02 19:28:56 UTC*
  - 1e150923aa Merge pull request #48877 from Vaelatern/improve-salt-cloud-libvirt-python3-libvirt
  - fb7885315c Wrap ElementTree.tostring to make strs, not bytes
- **PR** #48824: (rallytime) Bump deprecation in win_servermanager state to Neon @ *2018-08-02 18:01:34 UTC*
  - 1fb7d9431b Merge pull request #48824 from rallytime/win_servermanager_deprecations
  - 6ef5412528 Bump deprecation in win_servermanager state to Neon
- **ISSUE** #37512: (ChristianBeer) What's the precedence if multiple master configurations are specified? (refs: #48888)
- **PR** #48888: (terminalmage) Explictly document the configuration override priority @ *2018-08-02 16:57:18 UTC*
  - ec8e07e8ce Merge pull request #48888 from terminalmage/issue37512
  - 7dce7cde14 Explictly document the configuration override priority
- **PR** #48871: (dwoz) Remove unicode key pairs from environ after test @ *2018-08-01 22:33:41 UTC*
  - d4bb3a0963 Merge pull request #48871 from dwoz/test_cleanup
  - f2e15e7f1d Do not re-define tearDown
  - 4f8a191a69 Remove unicode key pairs from environ after test
- **PR** #48843: (isbm) Prevent u'something' to appear in help info. @ *2018-08-01 20:38:58 UTC*
  - bec79e83a3 Merge pull request #48843 from isbm/isbm-log-level-names-fix-2
  - a63686180d Merge branch '2018.3' into isbm-log-level-names-fix-2
  - 783c96ac72 Prevent u'something' to appear in help info.
- **PR** #48855: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-08-01 17:44:59 UTC*
  - a8376b537a Merge pull request #48855 from rallytime/merge-2018.3
  - 44998c208e Lint: Fix undefined logger variable
  - 92c9317a25 Update old utils paths to use new paths
  - 15bfba7143 Merge branch '2017.7' into '2018.3'
    - 05f2d65de3 Merge pull request #48853 from rallytime/bp-48850
      - 3c33ee554a Skip tests in integration.shell.test_master
    - 8a1285239a Merge pull request #48426 from garethgreenaway/46689_fixing_pkg_held_when_package_is_installed
      - 9b0f5dd212 Fixing identation, removing some unnecessary conditionals.
      - 727964ab55 One last cleanup.
      - 11cb86e6eb General cleanup in pkg state, reducing duplicate code. Fixing the requires_salt_modules decorator, sys.doc was returning too much information for the event to handle. This change specifically calls sys.doc with the module name.
      - 16fb6ae635 Make sure pkg.hold and pkg.unhold are available before running the test.
      - 998651102d Fixing a situation when a package is already installed via salt or manually and a state attempts to set that package to be held. Previously the holding/unholding logic was only being run against packages that were being installed. This change moves the holding

logic outside and runs it against all desired packages. Adding a new test to test holding logic.

- c8e69431ff Merge pull request #47734 from OrlandoArcapix/Issue47689-pip-state-performance
  - 662bd1f780 Merge branch '2017.7' into Issue47689-pip-state-performance
  - 66936b4f41 Changed string comparison in pip test to match new confirmation string - ref PR #47734.
  - bb5939d6ef Merge branch '2017.7' into Issue47689-pip-state-performance
  - d6a49ae41c Merge branch '2017.7' into Issue47689-pip-state-performance
  - d4083fc9d1 Merge branch 'Issue47689-pip-state-performance' of github.com:OrlandoArcapix/salt into Issue47689-pip-state-performance
  - 779b5fa785 Merge branch '2017.7' into Issue47689-pip-state-performance
  - f3653349ab Removed whitespaces at end of added comments lines
  - db11f2ff4b Merge branch '2017.7' into Issue47689-pip-state-performance
  - eac0178de2 Ref: #47689 - document additional kwarg passed to pip._check_if_installed function
  - 0d19803106 Merge branch '2017.7' into Issue47689-pip-state-performance
  - d3678bf2f3 #47689 fix lint errors
  - 4fec8f6bcc #47698 improve run-speed of pip package state checks by only loading the current package list once when checking multiple packages
- 83a5b3cc47 Merge pull request #48844 from AVeenstra/fix-python3-incompatibility
  - f238779a62 Merge branch '2017.7' into fix-python3-incompatibility
  - 6b1805afc6 Fixed Python 3 incompatibility in methods in nilrt_ip and debian_ip.
- 136ff6735a Merge pull request #48662 from slaws/fix-retention-schedule-48637
  - 3c8f5f5b3d pylint fix
  - 5539eff39e Excluding relative dirs in state.file.retention_schedule
- 1fc04f281b Merge pull request #48840 from gtmanfred/docs
  - 8d2d268c4a fix links in pkg doc.
- 0a19f845ea Merge pull request #48834 from gtmanfred/slsutil
  - f9441d2bef read output of stringio if it is readable
- 2e00939a6e Merge pull request #48788 from Ch3LL/timezone_windows
  - de95a6a215 add unused import to timezone test file
  - 22e424859e add unused import to import
  - 0840fc3117 disable pylint import error
  - f09d1a2c7e Add timzeon windows integration tests and fix get_zone

- **ISSUE** #48659: (dstoliker) file.grep with glob (*) in path produces file not found error (refs: #48830)
- **PR** #48830: (garethgreenaway) [2018.3] Fixes to file.grep @ *2018-07-31 21:30:04 UTC*
  - e23ba01cd2 Merge pull request #48830 from garethgreenaway/48659_file_grep_glob_fix
  - 0791b0a03c FileGrepTestCase tests back to 2018.3

  - 2e01c55e7e Fixing a bug that prevents specifying wildcards for filenames.
- **PR** #48814: (isbm) dmidecode race conditions check @ *2018-07-31 20:48:29 UTC*
  - 9fda70b3d0 Merge pull request #48814 from isbm/isbm-2018.3-smbios-bugfix
  - 5f1141dc05 Merge branch '2018.3' into isbm-2018.3-smbios-bugfix
  - b00ee5feef Update clean clause
  - b1b2e9c222 Remove multiple returns and combine logic to just one clause
  - fd77f760ee Rephrase explanatory comment
  - 1031e06443 Remove unnecessary code
  - c379b7e4ed Get rid of global variable
- **PR** #48804: (Ch3LL) Use brew path from which cmd in mac_brew module @ *2018-07-31 20:24:33 UTC*
  - 3d16a63ff2 Merge pull request #48804 from Ch3LL/mac_brew
  - e818c752b3 Merge branch '2018.3' into mac_brew
  - 41e3d17f29 Use brew path from which cmd in mac_brew module
- **PR** #48836: (dwoz) Fix unicode directory listing on py2 @ *2018-07-31 20:22:53 UTC*
  - 950c1014aa Merge pull request #48836 from dwoz/unicode_names_py2
  - 47e158b9f0 Optomize if statement
  - 9b462394b2 Fix unicode directory listing on py2
- **PR** #48847: (terminalmage) Update file.blockreplace docs to reflect changed functionality @ *2018-07-31 18:11:30 UTC*
  - baf8c5784a Merge pull request #48847 from terminalmage/issue48695
  - e2bdf7fb92 Update file.blockreplace docs to reflect changed functionality
- **PR** #48587: (twangboy) Fix lgpo issue on Py3 @ *2018-07-31 18:05:58 UTC*
  - 644a1f4ff8 Merge pull request #48587 from twangboy/fix_lgpo
  - a42621c817 Write file in normal mode
  - bfcbdeca3b Fix lgpo issue on Py3
- **PR** #48796: (Ch3LL) Remove fake su used in integration tests @ *2018-07-31 17:08:54 UTC*
  - 2bacc23481 Merge pull request #48796 from Ch3LL/remove_su
  - d3b7f2cb18 Remove fake su used in integration tests
- **ISSUE** #38310: (ghost) Unable to checkout external pillar git repo using its git tag (refs: #48689)
- **PR** #48689: (linoplt) Fix ext_pillar remote checkout using tag (pygit2) @ *2018-07-31 16:15:58 UTC*
  - 00d06bda76 Merge pull request #48689 from linoplt/fix_38310_pygit2_checkout_ext_pillar_remote_using_tag
  - eb4361dcc2 Merge branch '2018.3' into fix_38310_pygit2_checkout_ext_pillar_remote_using_tag
  - c20977e3bc Merge branch '2018.3' into fix_38310_pygit2_checkout_ext_pillar_remote_using_tag
  - b0157c215b Fix ext_pillar remote checkout using tag (pygit2)
- **PR** #48838: (sizgiyaev) Fixed: added additional return code 200 for succeeded vault api request @ *2018-07-31 16:14:22 UTC*
  - 5e14e36195 Merge pull request #48838 from sizgiyaev/fix-vault-policy-state

  - 02d09c6281 Fixed: added additional return code 200 for succeeded api request
- **PR** #48714: (dwoz) Always transfer bytes from fileserver roots @ *2018-07-30 22:37:13 UTC*
  - 69f81214d1 Merge pull request #48714 from dwoz/state_echo_fix
  - ce3ed43dab Always use unix line endings
  - d5e60090bc Merge pull request #2 from terminalmage/state_echo_fix
    - 52fc1c955d Simplify file contents in roots fileserver test
    - b6f73e8e57 Remove trailing whitespace on what should be a blank line
  - 95329acb1e Fileserver transfers bytes
  - aa34a80997 Always trasfer bytes from fileserver roots
- **PR** #48822: (Ch3LL) Fix salt-ssh state.sls_id TypeError key must be a string @ *2018-07-30 20:29:29 UTC*
  - 16ca4ec2eb Merge pull request #48822 from Ch3LL/ssh_2018_id
  - 01f6a15da0 Fix salt-ssh state.sls_id TypeError key must be a string
- **ISSUE** #47999: (arthurlogilab) Carbon returner failing when run through scheduler on log.trace (refs: #48757)
- **PR** #48791: (rallytime) Back-port #48757 to 2018.3 @ *2018-07-27 21:30:44 UTC*
  - **PR** #48757: (kuetrzi) Update carbon_return.py (refs: #48791)
  - 488f6716d4 Merge pull request #48791 from rallytime/bp-48757
  - f440ebe91f Update carbon_return.py
- **PR** #48799: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-07-27 21:30:13 UTC*
  - fae29f4257 Merge pull request #48799 from rallytime/merge-2018.3
  - 487458ad4b Update old utils paths to use new utils paths
  - 436510796f Merge branch '2017.7' into '2018.3'
    - 8e61f3dce0 Merge pull request #48742 from frogunder/45014
      - f08058b043 Merge branch '45014' of https://github.com/frogunder/salt into 45014
      - 757fde70c6 Merge branch '2017.7' into 45014
      - 33c20c1ec0 fix tests
      - a7a914060d add windows integration tests for states.file
    - d74f47b340 Merge pull request #48795 from rallytime/fix-48417
      - 5123b17ffd Update linting docs to contain .testing.pylintrc use
    - bbea9ae936 Merge pull request #48789 from rallytime/bp-48783
      - 682a05bebe Threshold was wrong
      - d4ca0e3a97 test: except for OSError only
      - 4547231909 Only run lint checks against changed files
    - 41464d4b39 Merge pull request #48731 from zer0def/virt-runner-init-args
      - a1fa081ad0 Documentation to missing parameters in virt.init runner.
      - 365ebdf539 Fixed *enable_vnc* runner arg being passed into *seed_cmd* module arg in *virt.init*.
    - 169afea16a Merge pull request #48749 from Ch3LL/logo_docs

        - 73b1fc29f0 Update Saltstack Logo banner on docs.saltstack.com

- **ISSUE** #48020: (calvinhp) mine.get not returning data in a state.orchestrate sls (refs: #48765)
- **PR** #48765: (FedericoCeratto) [2018.3] Fix mine.get not returning data @ *2018-07-27 18:01:06 UTC*
  - bd67d2a805 Merge pull request #48765 from FedericoCeratto/fix_mine.get_not_returning_data
  - 7183a6e0e8 Workaround for #48020
- **PR** #48747: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-07-26 18:14:34 UTC*
  - 6978ce5d1d Merge pull request #48747 from rallytime/merge-2018.3
  - 4122da40b5 Merge branch '2017.7' into '2018.3'
    - 781c6a6c36 Merge pull request #48743 from rallytime/saltconf-ad-2017.7
      - 6df8fd8652 Update DOCBANNER with new SaltConf18 image
    - ce4e22224e Merge pull request #48720 from Ch3LL/mac_file_path
      - 01d25fae19 Fix test_managed_source_hash_indifferent_case on macosx to correct tmp path
    - 4b9f037d43 Merge pull request #48719 from Ch3LL/service_mac_state
      - f44a2fc349 Enable service if disabled before running state service tests
    - b2431eab10 Merge pull request #48715 from rallytime/flaky-tests
      - 7332cce567 [2017.7] Mark some tornado tests as flaky
    - 801eae3b8d Merge pull request #48672 from frogunder/45012
      - 0747f2e58a add service enabled test
- **ISSUE** #48676: (djneades) salt-ssh should not target wrong minion from roster file as a result of reverse-DNS lookups (refs: #48771)
- **PR** #48771: (gtmanfred) only do reverse dns lookup on ips for salt-ssh @ *2018-07-26 15:41:40 UTC*
  - 5ea43817ab Merge pull request #48771 from gtmanfred/2018.3
  - 535fb8f8e7 only do reverse dns lookup on ips for salt-ssh
- **PR** #48752: (garethgreenaway) [2018.3] Fix when state file is integers @ *2018-07-25 15:03:02 UTC*
  - f2ef2d3ef6 Merge pull request #48752 from garethgreenaway/fix_state_file_all_integers
  - 790801c67e Fixing a case where the state module would fail if the state file being passed was all integers. Added a new tests for this edge case.
- **ISSUE** #46896: (Poil) Proxy + file.managed => Comment: Failed to cache xxx invalid arguments to setopt (refs: #48754)
- **PR** #48754: (lomeroe) send proxy/ca_cert parameters as strings (not unicode) to tornado httpclient @ *2018-07-25 14:55:42 UTC*
  - 030c921914 Merge pull request #48754 from lomeroe/fix-tornado-proxy
  - d33056704b pass proxy_user, proxy_password, proxy_host, and ca_certs through salt.utils.stringutils.to_str()
- **PR** #48769: (Ch3LL) Update Saltstack Logo banner on docs.saltstack.com @ *2018-07-25 14:47:19 UTC*
  - 1eee4b136b Merge pull request #48769 from Ch3LL/update_doc_2018.3
  - 0cb4bac798 Update Saltstack Logo banner on docs.saltstack.com
- **PR** #48760: (dwoz) Multiple windows test fixes @ *2018-07-25 11:55:27 UTC*

  - 19afa3b023 Merge pull request #48760 from dwoz/test_file_fixes
  - a89019e956 Multiple windows test fixes
- **PR** #48753: (dwoz) Finally fix prepend for real @ *2018-07-25 00:07:15 UTC*
  - e6dace3959 Merge pull request #48753 from dwoz/fix_prepend
  - 0d1e8ab3f8 Merge branch '2018.3' into fix_prepend
- **PR** #48756: (dwoz) os.uname is not available on py2 windows @ *2018-07-24 23:38:32 UTC*
  - 8aef6d9960 Merge pull request #48756 from dwoz/core_grains_fix
  - b7a37ecf11 os.uname is not available on py2 windows
- **ISSUE** #48536: (whytewolf) diskusage beacon does not recognize tmpfs disks in linux. (refs: #48718)
- **PR** #48718: (garethgreenaway) [2018.3] beacon diskusage fixes @ *2018-07-24 17:45:26 UTC*
  - 062fe7cccf Merge pull request #48718 from garethgreenaway/48536_beacon_diskusage_fixes
  - 38a65e8fb3 Fixing test_diskusage_nomatch and test_diskusage_match_regex
  - 6ecca166ff Updating psutil.disk_partitions to pull in all mounts not just the physical ones. Check to see if the mount point from the configuration ends with a $ (regular expression end of line) if not we add one in to ensure that a simple / does not end up matching all mount points. Updating tests accordingly.
- **PR** #48711: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-07-24 17:38:58 UTC*
  - e873621009 Merge pull request #48711 from rallytime/merge-2018.3
  - ea5c063237 Update old utils paths to use new paths
  - e740d3b208 Merge branch '2017.7' into '2018.3'
    - 689c231d2b Merge pull request #48691 from Ch3LL/win_repo_pkg_test
      - 4b7d6d80c5 Remove unnecessary jinja in curl.sls file
      - 2bedadfadb Add windows pkg module integration tests
    - 83e4bba916 Merge pull request #48635 from nbraud/acme
      - 3673bae9de modules/acme: explicitely ignore the *perms* return value
      - 1800a231e8 Fixup some schema expectations
      - 8c718cb417 acme: Make the private key mode configurable
      - 917dea6761 modules/acme: Use file.check_perms ret-morphing powers
      - d2241ceb2d module/acme: Do not exit early when the certificate already exists
      - 98af0db826 modules/acme: Set the private key filemode to 0640
    - 85991680c8 Merge pull request #48345 from twangboy/fix_48169
      - ead19725b6 Merge branch '2017.7' into fix_48169
      - 653fbcb383 Fix some docs errata
      - 8898e5ff11 Add warn_until Fluorine
      - 707906ac15 Fix unit.state.test_powercfg
      - 32c5014eb6 Fix unit tests
      - 0d9c56e540 Add some more logging
      - b20453de9f Use minutes to set

  - 7dc7eb11c2 Fix documentation to denote seconds when setting
- 96447ce541 Merge pull request #48656 from Ch3LL/windows_ip_mod
  - 90c3f568b1 Add windows ip module integration tests
- 0c5fff3dc8 Merge pull request #48638 from twangboy/fix_vcredist
  - e30d17099d Use goto instead of if statement
  - e1042fa084 Remove vcredist for Py3
- 1ebd96d909 Merge pull request #48664 from Ch3LL/srv_disable_mac
  - f1fbfad387 move the disable call up in try block
  - f60d21bda4 reverse assertion order for service disable test
  - 3727d1b3b9 switch try/except to match 2018.3
  - fb953c2369 [2017.7] Fix service.disabled test for macosx
- 3da3cf2f3f Merge pull request #48625 from Ch3LL/ssh_state
  - f590eb2b02 Update state.py
  - 9790ee3d0d Follow up to PR #48555
- 33812f78f1 Merge pull request #48673 from Ch3LL/mac_port
  - 87dd85a220 Use different pub and ret ports for testprogram integration tests
- 5f6a7c4d89 Merge pull request #48675 from Ch3LL/mac_disable
  - c78efab828 Fix mac service.disable tests
- 93d2f51d2b Merge pull request #48658 from wyardley/wyardley-npm-json-output-2017
  - 7ff3c9c5ff Improve handling of npm json output (#43138)
- 3e293b0513 Merge pull request #48678 from OrlandoArcapix/fix-npm-dryrun-test
  - 851a404f6b Fix for issue #48677 - return True when no changes are to be made with npm.bootstrap with test=true
- 07a1f6520f Merge pull request #48580 from rallytime/fix-46884
  - 736b382e91 Don't error on retcode 0 in libcrypto.OPENSSL_init_crypto call
- 9874429741 Merge pull request #48628 from terminalmage/testinfra
  - 5ace9f01ec Fix NameError in testinfra module
- 0d4c80205f Merge pull request #48651 from gtmanfred/pylint-2017.7
  - b6ee52f859 handle pyzmq for python3.4
- 160ae29cf2 Merge pull request #48647 from gtmanfred/pylint-2017.7
  - ff818c4ca2 disable checks on pylint
- 47b1032efa Merge pull request #48593 from pritambaral/fix/2017.7-importlib-pyc-loading-order
  - add7894dee loader: Fix suffix order when importlib is used
- 9da7b2ec8d Merge pull request #48630 from dubb-b/pipeline-updates
  - 8594a8dd05 Adding PY_COLORS=1 as PY_COLORS = 1 instead
  - 314b0e3599 Adding PY_COLORS=1 for python programs to use ANSI Colors

    - b705e8f7a5 Adding correct spacing to options section
    - 9d8a7e07db Adding options to Jenkins pipline builds
  - a8ae2adf64 Merge pull request #48633 from saltstack/revert-48610-2017.7
    - bab4a769d4 Revert "only run pylint on files that change"
  - 6e32bb7f74 Merge pull request #48614 from rallytime/bp-48562
    - cb654bbf2c Add timeouts to all s3 queries
  - 1b6e6388f8 Merge pull request #48588 from garethgreenaway/48415_event_send_multi_master
    - fab25af1a9 Adding some quick documention about why we are setting ret=True following the channel.send.
    - bf78f4b188 If the channel send is sucessful and does not raise an exception, we set ret to True, in case a previous exception from a previous channel send to another master has sent it to False.
    - 8d1551c5fb When using Salt multi-master, if we encouter a salt master that has not accepted the minion key yet we should not exit right away, rather continue on and try the next salt master available in the list.
  - 24ffda49ba Merge pull request #48610 from gtmanfred/2017.7
    - 779d1a3dcb only run pylint on files that change
  - 5391dd0a8d Merge pull request #48584 from Ch3LL/mac_user_grp
    - 585ee9db90 Fix grp import for mac in test_user integration test
  - 61572b6780 Merge pull request #48555 from Ch3LL/ssh_id
    - f69932f506 fix pylint
    - 70e36764ee Add more tests for salt-ssh state.sls_id
    - dab80e805c Fix state.sls_id to run on ssh minion and not master
  - aa6dcf39e8 Merge pull request #48583 from Ch3LL/mac_flaky_tests
    - 4ba2299a87 import flaky decorator on mac tests
    - 811220b41e Add flaky decorator to mac_system and mac_timezone tests
  - 6973152057 Merge pull request #48534 from xetix/fix-zypper-latest_version
    - 9985f0b4c1 Lint: remove extra blank line
    - 5fbead8a36 Merge branch '2017.7' into fix-zypper-latest_version
    - 56ac449271 Merge branch 'fix-zypper-latest_version' of https://github.com/xetix/salt into fix-zypper-latest_version
    - 44e87f5a0f Adding testcase for this fix.
    - b354c6863c Fix behaviour of function latest_version in zypper module when multiple packages are passed to function. Function now properly return dict with empty string as version if latest packages is already installed, and multiple packages are passed to function
    - db35d0c1e1 Adding testcase for this fix.
    - d2513757ed Fix behaviour of function latest_version in zypper module when multiple packages are passed to function. Function now properly return dict with empty string as version if latest packages is already installed, and multiple packages are passed to function

        - 10124034cb Merge pull request #48582 from dwoz/test_prepend_fix
            - f37571e0bd Merge branch '2017.7' into test_prepend_fix
        - 13f67335f3 Merge pull request #48564 from dwoz/test_prepend_fix
            - 136ddf5f54 Finally fix prepend for real
            - 66b25e65bf Fix failing prepend test
        - b8ce27729f Merge pull request #48558 from dwoz/test_file_fix
            - c858bf477f Remove which mock to get the test passing
        - a15c65202d Merge pull request #48552 from KaiSforza/jenkins-junit-2017
            - 72b1830974 Set up junit in jenkins
        - f73108026e Merge pull request #48550 from gtmanfred/2017.7
            - d5c603d9b6 add posargs to tox pylint
        - 113b0426f8 Merge pull request #48535 from asnell/asnell-patch-2
            - d328f6e43d Update manage_file helptext

- **PR** #48744: (rallytime) [2018.3] Update DOCBANNER with new SaltConf18 image @ *2018-07-24 17:15:40 UTC*
    - 6c7f1c549d Merge pull request #48744 from rallytime/saltconf-ad-2018.3
    - dbc1f8b772 Update DOCBANNER with new SaltConf18 image
- **PR** #48726: (dwoz) Skip test when syslog not available @ *2018-07-24 16:12:44 UTC*
    - 04ab17a4d6 Merge pull request #48726 from dwoz/syslogtest
    - 00f04cfd59 Skip test when syslog not available
- **PR** #48727: (dwoz) Windows compatible cp test @ *2018-07-24 14:41:34 UTC*
    - af837424aa Merge pull request #48727 from dwoz/file_cp_test_fix
    - 6e7824266b Windows compatible cp test
- **ISSUE** #40004: (te-af-github) svn changes report (refs: #47510)
- **ISSUE** #21025: (RobertFach) svn.export reports invalid change data (2014.7.x) (refs: #47510)
- **PR** #48710: (rallytime) Back-port #47510 to 2018.3 @ *2018-07-23 21:11:18 UTC*
    - **PR** #47510: (daa) Fix svn export invalid changes data (refs: #48710)
    - 04125b077d Merge pull request #48710 from rallytime/bp-47510
    - ebcd3b1c6b fixed test for svn.export
    - e79f4aa5cc put name under "new" key in changes dictionary to be in line with svn.latest
    - 163f99089c made svn.export changes data dictionary, fixes #21025, #40004
        - 3a941055c1 Use *salt.utils.platform.is_windows*
        - 33e48f7ad7 Finally fix prepend for real
- **PR** #48699: (terminalmage) Add missing release notes mention of file.blockreplace changes @ *2018-07-23 13:34:32 UTC*
    - e1dd10be70 Merge pull request #48699 from terminalmage/issue48695
    - 7ad832c901 Add missing release notes mention of file.blockreplace changes

- **PR** #48700: (dwoz) Call sync_modules to fix Windows test case @ *2018-07-22 22:23:48 UTC*
    - ab44a05fe4 Merge pull request #48700 from dwoz/ext_mod_test
    - d731e6bf8e Call sync_all to fix Windows test case
- **PR** #48653: (terminalmage) docker_container.running: Fix regression in test mode @ *2018-07-20 19:15:47 UTC*
    - 6a494205f2 Merge pull request #48653 from terminalmage/docker-unboundlocal
    - b0440871a0 Merge branch '2018.3' into docker-unboundlocal
    - 9515dd17ff docker_container.running: Fix regression in test mode
- **ISSUE** #47059: (OrlandoArcapix) Some states incorrectly return None instead of an empty dict when there are no changes (refs: #48685)
- **ISSUE** #46985: (OrlandoArcapix) grafana4_user.present and grafana4_org.present states not working in 2018.3.0 (refs: #48685)
- **PR** #48685: (bbinet) Use empty dict for 'changes' instead of None @ *2018-07-20 17:12:43 UTC*
    - **PR** #48671: (bbinet) Few fixes to the grafana module and states (refs: #48685)
    - 266c6f250e Merge pull request #48685 from bbinet/none-changes
    - 842eb3da24 Use empty dict for 'changes' instead of None
- **PR** #48670: (bbinet) Add the "traverse" jinja filter @ *2018-07-20 16:36:21 UTC*
    - 5a6f7a085c Merge pull request #48670 from bbinet/traverse
    - 32b6d22cf2 Fix RST syntax issue in doc
    - 37a41226d5 Add the "traverse" jinja filter
- **PR** #48594: (pritambaral) Backport 48418 @ *2018-07-20 14:57:19 UTC*
    - **PR** #48418: (pritambaral) Fix multi-sls *salt.state* orchestration in masterless systems (refs: #48594)
    - ff11763fd3 Merge pull request #48594 from pritambaral/bp-48418
    - adcee28b84 Fix multi-sls *salt.state* orchestration in masterless systems
- **ISSUE** #48184: (mmulqueen) Invalid DMI prevents salt-minion from running (refs: #48440)
- **PR** #48440: (mmulqueen) Make core.py tolerant of invalid chars in DMI data @ *2018-07-20 14:48:07 UTC*
    - **PR** #48216: (Ch3LL) Fix UnicodeDecodeError when reading file to determine virtual grain (refs: #48440)
    - 111908f519 Merge pull request #48440 from mmulqueen/patch-1
    - 5bbbdc74cd Make core.py tolerant of invalid chars in DMI data
- **PR** #48663: (Ch3LL) [2018.3] Fix service.disabled test for macosx @ *2018-07-20 14:32:22 UTC*
    - 4e77b242f2 Merge pull request #48663 from Ch3LL/srv_disable_mac_2018
    - bddfbd75d2 reverse assertion order for service disable test
    - 2bfaf7fe89 Fix service.disabled test for macosx
- **PR** #48304: (Ch3LL) Fix macosx grains when swapusage returns comma @ *2018-07-20 14:30:59 UTC*
    - dfc1582475 Merge pull request #48304 from Ch3LL/swap_mac
    - 5d6135e1e2 Fix macosx grains when swapusage returns comma
- **PR** #48671: (bbinet) Few fixes to the grafana module and states (refs: #48685) @ *2018-07-20 13:58:02 UTC*
    - 9d7eabd24e Merge pull request #48671 from bbinet/grafana-fixes

  - 3c5083f550 Add support for dry run (test=True) of grafana states
  - 7e7ae8ace7 influxdb: fix conflicts in function arg names with client_args
  - 73a24b774a Ignore readOnly option when diffing
  - 81f0cf8a3f Add missing profile argument to grafana4.get_user_data
- **PR** #48650: (gtmanfred) [2018.3] handle pyzmq for python3.4 @ *2018-07-18 17:34:37 UTC*
  - e828167114 Merge pull request #48650 from gtmanfred/pylint-2018.3
  - 3c1882077e handle pyzmq for python3.4
- **PR** #48648: (gtmanfred) [2018.3] disable checks on pylint @ *2018-07-18 16:21:37 UTC*
  - a6f8acf4dc Merge pull request #48648 from gtmanfred/pylint-2018.3
  - 7c34c2965e disable checks on pylint
- **PR** #48634: (dwoz) Sync modules for state tests @ *2018-07-17 20:33:06 UTC*
  - 01c9c59a97 Merge pull request #48634 from dwoz/state-module-test-fix
  - 9dfd2ae392 Sync modules for state tests
- **PR** #48585: (astorath) replaced meta tag @ *2018-07-16 14:11:02 UTC*
  - fb7bfc757c Merge pull request #48585 from astorath/fix-vault-meta
  - 42fd3d5f67 replaced meta tag
- **PR** #48579: (Ch3LL) Fix python2 syslog returner expecting string bytes not unicode @ *2018-07-13 19:50:16 UTC*
  - 4b16537a29 Merge pull request #48579 from Ch3LL/fix_syslog
  - d8288a0ffc add unicode type for syslog test for tag
  - 17e69382d5 Fix python2 syslog returner expecting string bytes not unicode
- **ISSUE** #48542: (calvinhp) Missing state git.cloned is listed in the docs as new for 2018.3.2 (refs: #48547)
- **PR** #48547: (gtmanfred) fix git.cloned doc versionadded @ *2018-07-13 18:46:25 UTC*
  - 3549ce408d Merge pull request #48547 from gtmanfred/git.cloned
  - b596a945ea Merge branch '2018.3' into git.cloned
  - e41f4922a0 fix git.cloned doc versionadded
- **ISSUE** #48110: (whytewolf) file.line on windows not treating unix line endings correctly. (refs: #48503)
- **PR** #48503: (rallytime) Back-port #46291 to 2018.3 @ *2018-07-13 17:14:47 UTC*
  - **PR** #48380: (twangboy) Detect and preserve line endings (refs: #48503)
  - **PR** #46291: (t0fik) Line mixed line ends (refs: #48503)
  - **PR** #45498: (t0fik) Added support for mixed line ending in file (refs: #46291)
  - ee257a1f91 Merge pull request #48503 from rallytime/bp-46291
  - 8bc71688ea Remove /sr.* pattern from test
  - 5e2e2a63fa Update file unit tests to handle "writelines" change
  - 5d38aa8b33 Update file.line to use *writelines* instead of joining
  - d0b6e82ecb Add 'name' as the passed in filepath for test_line_insert_ensure_before_first_line test
  - d8920cb61f Update *test_line_insert_ensure_before_first_line* to use new mock_open methodologies

  - 475f075d8e Handle list of lines instead of strings in file.line func
  - b9ddd53b04 Added comments
  - f3517a1852 List comprehension replaced
  - adfa3aed0d Empty match on delete or replace not causing IndexError exception
  - 5169b1f7fd Comprehensions converting to unicode replaced by salt.utils.data.decode_list
  - d3e8679e05 Removed regex compilation
  - f29815b49b Fixed file permissions
  - d2af81e9c7 Make integration green. Added test for mode ensure insert before first line
  - 9b7df671a5 file.line function refactored
  - 3af551ebe1 /sr.* pattern should raise exception
  - 935a9b9d56 test_line_insert_end fixed
  - 7d6e3ad2e2 Make tests green
  - 75a7e368a6 Setting end of line
  - 489e381100 Added _set_line_eol and _get_eol functions
  - aacbb8c0e2 line function refactored to work on list
  - de668166f9 _regex_to_static refactored to work on lists
  - 464eef6fe1 _get_line_indent renamed to _set_line_indent
- **ISSUE** #48507: (emersonveenstra) mysql_grants.present escape option incorrectly reports failure (refs: #48561)
- **PR** #48561: (garethgreenaway) [2018.3] fixes to grants in mysql module @ *2018-07-13 14:24:58 UTC*
  - 2d923f2943 Merge pull request #48561 from garethgreenaway/48507_mysql_grants_incorrect_failure
  - 14c59da72c Following a successful grant application in the MySQL module, the new grant and the desired grant are mismatched because the new grant from Mysql included hashmarks. This change adds the replace which is included for other items such as database name and username.
- **PR** #48529: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-07-12 17:59:35 UTC*
  - 5b5a930449 Merge pull request #48529 from rallytime/merge-2018.3
  - f9b8c4b257 Use () when using with_tempfile decorator in tests
  - d3190ca0c0 Update old utils paths to new utils paths
  - 29b05ffdd1 Merge branch '2017.7' into '2018.3'
    - 0c0c05c2bc Merge pull request #48527 from saltstack/revert-48525-2017.7
      - 18d06c1a93 Revert "add kazoo for testing zookeeper."
    - 05bad3e71c Merge pull request #48526 from twangboy/rollback_certifi-win32
      - 361b3cdc2f Rollback python-certifi-win32
    - db066effe4 Merge pull request #48521 from Martin819/2017.7
      - d5d01f6698 Fix for GlusterFS 4.0 and above
    - ed0bd2bbec Merge pull request #48525 from gtmanfred/2017.7
      - d7a6bff2b2 add kazoo for testing zookeeper.
    - ad5a959ab9 Merge pull request #48388 from garethgreenaway/48277_2017_7_file_roots_wrong

        - 6f11da35a7 Fixing a few things in the new test.
        - cf747b14ec Fixing lint issue
        - 16d36c79c8 When pillar items are compiled a new render is instantiated but the file_roots is the pillar_roots. This change forces the __opts__['file_roots'] to be set to what is set in actual_file_roots for all renderers once compile_pillar has finished. Adding a test when this situation is run via a orchestration runner.
    - 38df812257 Merge pull request #48512 from gtmanfred/npm
        - 9ba0f6b3a0 pin pm and grunt packages for npm tests
    - 04ba31147f Merge pull request #48513 from rallytime/merge-2017.7
        - a466a4880f Merge branch '2017.7.7' into '2017.7'
        - cfd21ed22c Merge pull request #48172 from rallytime/2017.7.7-release-notes
        - a0ad794132 Update release notes for 2017.7.7
        - 8af4452134 Merge pull request #48157 from gtmanfred/2017.7.7
        - d8209e8a40 always listen when gathering job info
        - b98c52ee51 Merge pull request #48140 from rallytime/man-pages-2017.7.7
        - 8893bf0d4c Update man pages for 2017.7.7
        - baa0363336 Merge pull request #48136 from gtmanfred/2017.7.7
        - fce1c31146 bootstrap kitchen branch tests with 2017.7.6
        - b0ba08f4d9 Merge pull request #48134 from rallytime/release-notes-2017.7.7
        - 217005b8f1 Add missing *v* for tag reference
        - d53569d1e3 Add release notes file for 2017.7.7
        - 084de927fe Merge pull request #48098 from rallytime/bp-48075-2017.7.7
        - e4e62e8b3a Ensure that the shared list of jids is passed when creating the Minion. Fixes an issue when minions are pointed at multiple syndics.
    - f0352ea95a Merge pull request #48514 from gtmanfred/2017.7
        - 739bf92599 upgrade dependencies for pytest tests
    - 5372b60137 Merge pull request #48491 from grokrecursion/group-fix-v2
        - 1556b37221 fixed pylint errors
        - f6d9177dab fixed indentation for pep8
        - b9556bf923 v2 try of gid converted to integer
    - 16d3daab3c Merge pull request #48375 from Ch3LL/file_copy
        - d16a790775 Add mode to _makedirs call in file.copy
        - 2cbcb4fd26 Add user and group to makedirs cmd in file.copy
    - d38951b1b3 Merge pull request #48490 from KaiSforza/cifixes
        - 9fe7199af5 Mark failed if anything is found
        - 6749a6bf19 Stop archiving the docs

- **PR** #48553: (KaiSforza) Set up junit in jenkins @ *2018-07-12 16:24:01 UTC*

  - a95b8efd8a Merge pull request #48553 from KaiSforza/jenkins-junit-2018
  - 412ffcc956 Set up junit in jenkins
- **PR** #48551: (gtmanfred) add posargs to tox pylint @ *2018-07-12 14:35:25 UTC*
  - c4aed02740 Merge pull request #48551 from gtmanfred/2018.3
  - a6a286af28 add posargs to tox pylint
- **PR** #48533: (terminalmage) Fix UnicodeDecodeError in sh beacon @ *2018-07-12 13:38:43 UTC*
  - 260da0d545 Merge pull request #48533 from terminalmage/sh-beacon
  - 7ac5ac06c9 Fix UnicodeDecodeError in sh beacon
- **ISSUE** #48519: (Lutseslav) Mysql module alter_db quotation missing (refs: #48520)
- **PR** #48520: (Lutseslav) Add quotes to schema name in ALTER DATABASE. @ *2018-07-11 19:15:08 UTC*
  - 8761ac2589 Merge pull request #48520 from Lutseslav/fix-alter-databases-with-special-symbols
  - d088b00ff2 Merge branch '2018.3' into fix-alter-databases-with-special-symbols
  - 59629e9757 Merge branch '2018.3' into fix-alter-databases-with-special-symbols
  - da8336712e Add quotes to schema name in ALTER DATABASE.
- **PR** #48528: (gtmanfred) add 2018.3 requirements for tox/pytest tests @ *2018-07-11 18:13:40 UTC*
  - ad4f7efafd Merge pull request #48528 from gtmanfred/2018.3
  - 221559358a add 2018.3 requirements for tox/pytest tests
- **ISSUE** #48336: (JuanManuelVizcainoAbad) file.directory (refs: #48399, #48398)
- **PR** #48508: (rallytime) [2018.3] Fix 2 bugs found in the file.check_perms function @ *2018-07-11 15:03:32 UTC*
  - **PR** #48399: (garethgreenaway) [2017.7] fixes to module/file.py (refs: #48508)
  - **PR** #48398: (garethgreenaway) [2018.3] fixes to module/file.py (refs: #48508)
  - 70e5fcb8a5 Merge pull request #48508 from rallytime/fix-file-bug
  - 0747eb5476 Move comment string join and test/changes check to bottom of file.check_perms
  - aefa1dbe21 Fix up bad merge - remove extra section of "mode" changes
- **ISSUE** #48277: (dvenckus) init.sls with included states fails with more than one, 'Template was specified incorrectly: False' (refs: #48388, #48389)
- **ISSUE** #46986: (github-abcde) opts file_roots gets overwritten with pillar_roots in orchestration run (refs: #48388, #48389)
- **PR** #48389: (garethgreenaway) [2018.3] reset file_roots for renderers after compile_pillar @ *2018-07-11 13:13:33 UTC*
  - 5a0b274bf1 Merge pull request #48389 from garethgreenaway/48277_2018_3_file_roots_wrong
  - 4b492fa633 Removing unused start_time variable from test_orchestration_with_pillar_dot_items test.
  - 30c5855d20 Fixing a few things in the new test.
  - 36b7253418 Fixing lint issue
  - ab6abfad31 When pillar items are compiled a new render is instantiated but the file_roots is the pillar_roots. This change forces the __opts__['file_roots'] to be set to what is set in actual_file_roots for all renderers once compile_pillar has finished. Adding a test when this situation is run via a orchestration runner.

- **ISSUE** #48342: (jeffclay) UnicodeDecodeError when using cache mysql (refs: #48495)
- **PR** #48495: (garethgreenaway) [2018.3] Fix to mysql cache module @ *2018-07-10 12:54:10 UTC*
  - f9fd1a889a Merge pull request #48495 from garethgreenaway/48342_mysql_cache_unicode
  - 0029f19033 Ensure the query_string is a bytestring, otherwise an attempt will be made to convert the msgpack data to unicode which will result in a UnicodeDecodeError error.
- **PR** #48487: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-07-09 18:38:00 UTC*
  - fc3eeef6b3 Merge pull request #48487 from rallytime/merge-2018.3
  - 54341d1a8d Update old utils paths to use new utils paths
  - 68aabff0cf Merge branch '2017.7' into '2018.3'
  - 22cd4206c4 Merge pull request #48472 from gtmanfred/2017.7
    - c7a3a7d8bd update jenkins tests to use tox for lint
    - 44aaac1d33 use tox to run pylint
  - 27ea8f35ea Merge pull request #48476 from twangboy/fix_dependencies
    - b0087d425c Add license info
    - 935f9b560c Fix dependencies
  - acf42864aa Merge pull request #48399 from garethgreenaway/48336_2017_7_ensure_chmod_setuid_with_chown
    - 8efd33320f Normalize the mode before we compare it.
    - f894f0ecb8 Setting the mode with setuid or setgid bits in addition to setting the owner and group will force the setuid & setgid bits to reset. This change ensures that we set the mode after setting the owner & group.
  - 6166ff6b78 Merge pull request #48471 from gtmanfred/2017.7
    - 392ab4e51f Add some configurations to tox
  - 6234d9b15d Merge pull request #48433 from discogestalt/fix-redis.hmset
    - 65817ac74d Use clean_kwargs method instead
    - f7fa7f57c6 Found another issue with redismod.hmset
    - 5624865f86 Fix issue with redismod.hmset method
  - aacee0fe2c Merge pull request #48428 from terminalmage/fix-jobs.lookup_jid
    - 6509aa9089 Fix outputter detection in jobs.lookup_jid runner
  - a7e52f0de2 Merge pull request #48429 from rallytime/bp-46824
    - 8b4486248d Added ignore_retcode to mock unit tests
    - f8beab71dd Regression to ignore retcodes on crontab calls
  - 4576ef20bc Merge pull request #48432 from dwoz/file-prepend-again
    - 349a2b279e Prepend test needs file.touch method
- **PR** #48481: (terminalmage) Improve the slack engine docs @ *2018-07-09 13:41:15 UTC*
  - 4d4d2eacce Merge pull request #48481 from terminalmage/slack-engine
  - 56e8a1eb93 Improve the slack engine docs
  - dde9c0640d Add information on creating bot users

  - 513c6af975 Fix inaccurate docs

- **PR** #48348: (dwoz) Fix multiple git module tests @ *2018-07-06 20:35:58 UTC*
  - 491b5b077c Merge pull request #48348 from dwoz/test_git
  - 5ca5e060b3 Merge branch '2018.3' into test_git
- **PR** #48454: (terminalmage) Improve error message when ext_pillar is incorrectly formatted @ *2018-07-06 18:03:44 UTC*
  - dd6a6a97c5 Merge pull request #48454 from terminalmage/issue48107
  - 269dbab7f6 Improve error message when ext_pillar is incorrectly formatted
- **ISSUE** #48336: (JuanManuelVizcainoAbad) file.directory (refs: #48399, #48398)
- **PR** #48398: (garethgreenaway) [2018.3] fixes to module/file.py (refs: #48508) @ *2018-07-06 18:02:43 UTC*
  - a9dc758e73 Merge pull request #48398 from garethgreenaway/48336_ensure_chmodd_setuid_with_chown
  - 2454652914 Normalize the mode before we compare it.
  - 08ad5665de Fixing test to ensure it works under py3
  - 2b25d8c95b Setting the mode with setuid or setgid bits in addition to setting the owner and group will force the setuid & setgid bits to reset. This change ensures that we set the mode after setting the owner & group.
- **PR** #48431: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-07-05 17:23:00 UTC*
  - 2baa7f189f Merge pull request #48431 from rallytime/merge-2018.3
  - 3e59dda0b1 Merge branch '2017.7' into '2018.3'
    - 422441505d Merge pull request #48422 from rallytime/fix-48230
      - a6abf85621 Allow file.managed to work with uppercase source_hash in test=true mode
      - 4c691ac57a Add regression test for Issue #48230
    - 83e387c951 Merge pull request #48410 from dwoz/service_dead_test_fix
      - 5c0923448d Fix service dead test on windows
    - d3ba345da5 Merge pull request #48413 from twangboy/fix_47811
      - c5aad04140 handle unc paths
    - 827ef2aac4 Merge pull request #48416 from dwoz/service_integration_fix
      - c43f150dcd Fix service integration test (py3)
    - 10fe7d2b9b Merge pull request #48385 from Ch3LL/1update_version_doc_2017.7
      - 3cf335b0b9 Update release versions for the 2017.7 branch
    - f8a9a037e6 Merge pull request #48405 from Ch3LL/rm_inprog_2017.7
      - 03ed5167fa [2017.7] Remove In Progress Warning on 2017.7.7
    - 8ac867c168 Merge pull request #48409 from dwoz/test_prepend_fix
      - 9c7085b70f Fix file state prepend test
    - 10e93bff7f Merge pull request #48396 from dwoz/symlink_test_fix
      - d3456d31e6 Fix file.symlink state test for windows
- **PR** #48310: (mtorromeo) Backport ini_manage fixes to 2018.3 @ *2018-07-05 16:27:19 UTC*

  - 432cbbb5ee Merge pull request #48310 from mtorromeo/ini-manage-backports
  - 88f80fdf77 Fixed pylint useless-super-delegation warning.
  - e20f4cacd4 Uniformed reports between ini.options_absent and ini.options_present
  - 2c3ae0b6c9 pylint cosmetics.
  - 0c0d10d18f Fixes dry run false positive when option value is a number.
  - 70144bccbb Fixes dry run in ini_manage + Fixes related bug - when working with options which are not in section. + Fixes related tests + Fixes pylint warnings
- **ISSUE** #48402: (elsmorian) Running SaltStack master with no init system repeatedly logs "could not determine init system " (refs: #48441)
- **PR** #48441: (terminalmage) Switch init system log message to debug @ *2018-07-05 14:43:25 UTC*
  - 84fd3d2784 Merge pull request #48441 from terminalmage/issue48402
  - d758995763 Switch init system log message to debug
- **PR** #48386: (Ch3LL) Update release versions for the 2018.3 branch @ *2018-07-03 13:37:32 UTC*
  - 871c9e3b98 Merge pull request #48386 from Ch3LL/1update_version_doc_2018.3
  - 84a1994110 Update release versions for the 2018.3 branch
- **PR** #48404: (Ch3LL) [2018.3] Remove In Progress Warning on 2017.7.7 @ *2018-07-03 13:33:18 UTC*
  - ddb83ad4ce Merge pull request #48404 from Ch3LL/rm_inprog_2018.3
  - ef288f2a74 [2018.3] Remove In Progress Warning on 2017.7.7
- **PR** #48339: (terminalmage) Backport tojson filter from #48309 to 2018.3 @ *2018-07-02 17:52:53 UTC*
  - **PR** #48309: (terminalmage) Drop support for Python 2 unicode string literals in YAML renderer (refs: #48339)
  - 5b38019ca0 Merge pull request #48339 from terminalmage/backport-tojson-filter
  - 4433bec891 Add release notes for tojson jinja filter
  - ac36998801 Tweak docs to reflect backported filter
  - c636b18cf0 Add unit test for tojson filter
  - e34e39f0e2 Document filter deprecation
  - b9a4f288b2 Deprecate json_encode_dict and json_encode_list jinja filters
  - 3896e3468c Use upstream tojson filter, if present
  - 1499c6abcf Implement tojson jinja filter for those using Jinja < 2.9
- **ISSUE** #48274: (ipmb) s3.get signature failure with + in the object name (refs: #48328)
- **PR** #48328: (garethgreenaway) [2018.3] Fixes to utils/s3.py @ *2018-07-02 13:22:31 UTC*
  - 222c1d65db Merge pull request #48328 from garethgreenaway/48274_s3_get_signature_fail
  - 22cafc8334 fixing lint issue
  - 131486203e Need to pass the URL path through quote to ensure any special characters are being quoted properly. if we include local_file for a PUT, read the file into data before passing it along to requests.requests.
- **PR** #48358: (Ch3LL) Fix corrupt public key with m2crypto python3 @ *2018-07-02 13:10:32 UTC*
  - d3c658bdcf Merge pull request #48358 from Ch3LL/m2crypto_fix

  - 8fdd34d430 m2crypto open file in rb mode for pub key
  - a964db4663 Fix corrupt public key with m2crypto python3
- **ISSUE** #48367: (asenci) salt.proxy.fx2: dictionary changed size during iteration (refs: #48368)
- **ISSUE** #46765: (roskens) pkg.mod_repo fails with a python error when removing a dictionary key (refs: #46776)
- **PR** #48384: (rallytime) Back-port #48368 to 2018.3 @ *2018-07-01 04:19:10 UTC*
  - **PR** #48368: (asenci) Fix issue #48367 using the same approach as PR #46776 (refs: #48384)
  - **PR** #46776: (gtmanfred) fix shrinking list in for loop bug (refs: #48368)
  - 2fd63d66dc Merge pull request #48384 from rallytime/bp-48368
  - 430c462f34 Fix issue #48367 using the same approach as PR #46776
- **PR** #48394: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-07-01 04:13:03 UTC*
  - 3b53e2d206 Merge pull request #48394 from rallytime/merge-2018.3
  - 047f5b5f39 Update old utils paths to new utils paths
  - c4fd2a0930 Merge branch '2017.7' into '2018.3'
  - 7e25f26837 Merge pull request #48362 from twangboy/fix_48276
    - 77629f640a Fix module unit tests
    - b5ca560b73 Fix lint error
    - 210e280ecf Use explicit parameter names
    - 776db98a5c Handle missing registry entries
  - 80222b6b7c Merge pull request #48383 from rallytime/bp-48379
    - 9fc7815594 Clean up the workspaces at the end for every job
  - aa68aa774a Merge pull request #48382 from rallytime/bp-48346
    - 6a70ba222e Use the right ssh key to destroy kitchen
  - 9519f640e7 Merge pull request #48381 from rallytime/bp-48330
    - 6857bedcc2 Add warnings plugin to replace violations
  - 14db0aa35d Merge pull request #48363 from dubb-b/2017.7
    - c0b962e28d Changing debug to info for logging
  - 5b29cd326f Merge pull request #48352 from Ch3LL/wheel_doc
    - 89ada68165 Add missing key to accept_dict function in wheel docs
  - 02cf19ee3b Merge pull request #48329 from rallytime/codeowners-roster-files
    - 5e56615e00 Trigger review requests for team-ssh for roster files
  - 53cf1794be Merge pull request #48349 from ralex/fix-manjaro-service-behaviour
    - 6cafce547e Disable the *service* module on Manjaro since it is using systemd
  - 89dfcf3a4e Merge pull request #48324 from Ch3LL/update_version_doc_2017.7
    - 47845ba810 Update release versions for the 2017.7 branch
  - ece4c30aac Merge pull request #48331 from rallytime/bp-48215
    - 9d6ba3e247 Set jobs to pending when they come in

    - 727d2b4ffe Use proper creds with githubNotify
    - d355861c5c Test using different credentials
  - ab05e00d8c Merge pull request #48332 from rallytime/bp-48321
    - d72af6ab4e Jenkins needs the / to get the directory
  - 21ed5b97ce Merge pull request #48295 from rallytime/bp-48193
    - 352fe33fd6 Separate set_result() to a new line apart from Future() call
    - c0180ff33d Properly configure syndic in test case
    - d0a98534a9 Properly wait on returns in saltnado
  - 9aa4687ab9 Merge pull request #48117 from twangboy/fix_48026
    - dd37f8fbc2 Disable services that are set to manual
  - caf630487c Merge pull request #48207 from rallytime/bp-48189
    - e9d09e0375 Use old is_windows utils path on 2017.7
    - b965d6c9b3 If pip binary is passed to bin_env, use that pip binary
  - 06a927b2aa Merge pull request #48293 from rallytime/bp-47453
    - e96ab6778e dont reset system locale when running rabbitmqctl commands
- **ISSUE** #45939: (andygabby) user.present with hash_password: True detects change on every state.apply/highstate (refs: #47147)
- **PR** #47147: (eliasp) states.user.present: Make usage of *hash_password* idempotent @ *2018-06-30 13:29:27 UTC*
  - 9b364e25cf Merge pull request #47147 from eliasp/2018.3-issue-45939-shadow-hash-salt
  - dd3be1d76e Add warning log message when using MD5 for hashing shadow passwords
  - 5451ab6b7a states.user.present: Make usage of *hash_password* idempotent
- **PR** #48297: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-06-30 12:57:35 UTC*
  - 89857ea8b9 Merge pull request #48297 from rallytime/merge-2018.3
  - 9498618418 Use *saltenv* opts instead of *environment*
  - 1c8bd35f28 Update old utils paths to use new utils paths
  - c7bb8a50b0 Merge branch '2017.7' into '2018.3'
    - 95ef006e00 Merge pull request #48219 from zer0def/lxc-bootstrap-fixes
      - c77058560b Merge branch '2017.7' into lxc-bootstrap-fixes
      - d63cf3f072 Fixes another case of legacy configuration key usage warning getting in the way of changing container's state.
    - 3327181507 Merge pull request #48234 from dwoz/thin_dir
      - 70c603451b Fix py2 thin dir issues
    - 83d7d286c4 Merge pull request #48080 from lusche/2017.7
      - 917dc985fc #47984 remove the line completly
      - ba12ee947b Merge branch '2017.7' of https://github.com/saltstack/salt into 2017.7
      - dfb13d5051 Bugfix #47984 messed up cert serial
    - bccb4dcd46 Merge pull request #48232 from gtmanfred/cloud

      - d108112e1a do not expand kwargs for cloud.action
    - 056f43f663 Merge pull request #48238 from mirceaulinic/fix-sdb-cache
      - 51e5fbfa1d SDB cache module: AttributeError: 'Cache' object has no attribute 'set'
    - 731ec0a11a Merge pull request #48037 from terminalmage/fix-custom-types-sync-docs
      - 052ae83c4b Update versionchanged
      - 8b1bd0eda2 Update test to reflect changed argument name
      - 5e75936198 Change 2018.3.2 to 2018.3.3
      - c53ad603fc Rename sync -> sync_mods per review suggestion
      - e4d67c5fd8 Update docs to include references to new "sync" argument
      - cb8e6f9fb8 Remove redundant mocking
      - bc3ad795e9 Add test for sync argument to state.sls
      - f81ccd1fdd Add sync option to state.apply/state.sls
      - 8289b07e24 Fix documentation on when custom types are synced
    - 0bae927048 Merge pull request #48249 from rallytime/2017.7.7-release-notes-update
      - 36032c8ee7 Update release notes for 2017.7.7
    - 8e06471817 Merge pull request #48242 from asnell/asnell-patch-1
      - f66bf60073 Add sample list data via command line pillar
- **PR** #48364: (dubb-b) Changing debug to info for logging @ *2018-06-28 20:48:24 UTC*
  - c5746deb5f Merge pull request #48364 from dubb-b/2018.3
  - d8260b6628 Changing debug to info for logging
- **PR** #48354: (rallytime) [2018.3] Merge forward from 2018.3.2 to 2018.3 @ *2018-06-28 20:26:31 UTC*
  - 0202493b60 Merge pull request #48354 from rallytime/merge-2018.3.2-to-2018.3
  - fface7cc74 Merge branch '2018.3.2' into '2018.3'
- **PR** #48347: (dwoz) Fixes fileserver maintenance process on Windows @ *2018-06-28 13:17:42 UTC*
  - bbabbbf76b Merge pull request #48347 from dwoz/sigfix
  - ccdff5029f Remove unneeded attribute
    - e8362adf71 fopen does not support encoding
    - 487161ba8f Fix file encoding on windows
    - fc760685ee Fix multiple git module tests
- **PR** #47975: (terminalmage) Add a new git.cloned state @ *2018-06-27 20:53:42 UTC*
  - 67303d7901 Merge pull request #47975 from terminalmage/issue47937
  - 34b24bb7fa Merge branch '2018.3' into issue47937
- **PR** #48323: (Ch3LL) Update release versions for the 2018.3 branch @ *2018-06-27 13:44:58 UTC*
  - 5b8d55428e Merge pull request #48323 from Ch3LL/update_version_doc_2018.3
  - b4548aca56 Update release versions for the 2018.3 branch
- **PR** #48326: (Ch3LL) Remove In Progress Warning for 2018.3.2 Release @ *2018-06-27 13:43:16 UTC*

  - 7cf403e313 Merge pull request #48326 from Ch3LL/rn_remove_in_prog
  - dfce1ad5ed Remove In Progress Warning for 2018.3.2 Release
- **PR** #48301: (terminalmage) Fix typos in new mock_open docs @ *2018-06-25 19:08:14 UTC*
  - 60ffad5126 Merge pull request #48301 from terminalmage/docs
  - 6c33345f05 Fix typos in new mock_open docs
- **PR** #48292: (rallytime) Back-port #48288 to 2018.3 @ *2018-06-25 19:07:11 UTC*
  - **PR** #48288: (rmarchei) fix lsattr.get in file.check_perms (refs: #48292)
  - 4760f87153 Merge pull request #48292 from rallytime/bp-48288
  - d6620573bb fix lsattr.get in file.check_perms
- **PR** #48296: (rallytime) Back-port #48258 to 2018.3 @ *2018-06-25 19:04:41 UTC*
  - **PR** #48258: (nullify005) preserve tuples for ldap modification operations (refs: #48296)
  - 27207fc510 Merge pull request #48296 from rallytime/bp-48258
  - 8b7f36f264 preserve tuples for ldap modification operations
- **ISSUE** #48204: (zerthimon) State mysql_user.present Exception with mysql 8.0.11 (refs: #48275, #48228)
- **PR** #48275: (garethgreenaway) [2018.3] Follow up fix to MySQL user for password management @ *2018-06-25 13:23:35 UTC*
  - d5265da945 Merge pull request #48275 from garethgreenaway/mysql_fix_followup
  - bb024b9115 Removing unnecessary code.
  - 8b542e1745 With MySQL versions 8.0.11 and beyond, since the PASSWORD function has been removed we need to hash the password string before comparing when checking if the user exists.
- **PR** #48286: (terminalmage) Don't load the grains again when printing them via salt-call -g @ *2018-06-25 13:19:18 UTC*
  - 1f306525a7 Merge pull request #48286 from terminalmage/caller-print_grains
  - caa62c051f Don't load the grains again when printing them via salt-call -g
- **PR** #48279: (terminalmage) Add documentation/further enhancements to rewritten mock_open @ *2018-06-25 13:12:30 UTC*
  - 27a0e95174 Merge pull request #48279 from terminalmage/mock_open-docs
  - 65c575ae9c Fix spelling error
  - 830a624ce8 Ignore IOError when tearing down filehandles not opened for reading
  - 3b6356f4b0 Raise TypeError when trying to read from filehandles not opened for reading
  - 09aae0c82b Add more examples and information on strict string types
  - 8c069d105d Add some mock_open docs
  - a13d1fe1a0 Update mock_open usage to reflect read_data type enforcement
  - 374a8ce31f Add more mock_open tests
  - 0f06adb008 Improve code-reuse in mock_open tests
  - 0e8c83bac6 2 MockFH enhancements:
- **ISSUE** #48141: (zerthimon) salt-api request causes exception (refs: #48236)

- **PR** #48236: (garethgreenaway) [2018.3] Fix to cmd_batch @ *2018-06-25 12:58:49 UTC*
  - 6ee856631b Merge pull request #48236 from garethgreenaway/48141_salt_api_local_batch
  - aea4f7ae14 Fixing test_local_batch
  - ce3e1eaf2c Removing sort.
  - 83accf3b20 Fixing test_local_batch
  - ff14b99752 Fixing a typo in the comment.
  - 85cef126e0 Adding note about why salt.utils.versions is being re-imported. Adding a test for local_batch.
  - 808805fd3d Fixing cmd_batch to work correctly when called via salt-api.
- **ISSUE** #48204: (zerthimon) State mysql_user.present Exception with mysql 8.0.11 (refs: #48275, #48228)
- **PR** #48228: (garethgreenaway) [2018.3] Fixes to mysql module @ *2018-06-22 15:44:00 UTC*
  - 09b036c26e Merge pull request #48228 from garethgreenaway/48204_mysql_user_password_exception
  - 1b2ffcef1d Updating the mysql module to not use the PASSWORD when MySQL is version 8.0.11 or higher, where the PASSWORD function has been removed.
- **PR** #48252: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-06-22 15:13:42 UTC*
  - 67cb5563ef Merge pull request #48252 from rallytime/merge-2018.3
  - 84ec655c24 Merge branch '2017.7' into '2018.3'
    - a172f9de84 Merge pull request #48226 from terminalmage/fix-alias-docstring
      - fb237272f5 Don't display "None" in SaltInvocationError when Salt installed using -OO
      - 90c90f5d5c Fix docstring construction in alias_function when Salt installed using -OO
    - c6a0207cae Merge pull request #48227 from Ch3LL/fix_win_tests
      - 236773e3e9 Skip new sha256 files on repo.saltstack.com/windows
    - 6929423528 Merge pull request #48131 from bowmanjd-lms/fix-apk-python3
      - f5d2835299 Merge branch '2017.7' into fix-apk-python3
      - 022f9cba50 Fix py3-incompatible dict.keys()[x] call in apk.py
    - 868c17377f Merge pull request #48185 from DSRCorporation/bugs/47901_future_done_exception
      - 5f63316311 Merge branch '2017.7' into bugs/47901_future_done_exception
      - 9f2dbf94cb Don't call .exception() on future unless it's done.
    - 80a3e37c2e Merge pull request #48192 from twangboy/fix_pip_version_2017.7
      - 59ae2cc5fc Use –disable-pip-version-check for build_env_#.ps1
    - 12e2b8882a Merge pull request #48201 from zer0def/lxc-bootstrap-fixes
      - 6f6d3d40d4 Take lxcpath into account when bootstrapping new containers. Version comparison fixes.
    - e079fce38d Merge pull request #48190 from terminalmage/fix-vfstab-test
      - 8d70d14362 Fix mount.vfstab unit test
    - c8b9cec986 Merge pull request #48115 from KaiSforza/pr-lint-2017
      - 494727ab39 Add docs, kitchen tests
      - 317023bb20 Move .jenkins to .ci for future

      - dedc313cee Make it parallel so we don't fail right away
      - cd13426726 Add a .jenkins file to run pylint
    - fc1752b721 Merge pull request #48174 from terminalmage/docker-update-mine
      - 9af09e0127 Mock config.get due to changes in _refresh_mine_cache
      - c5802ad465 Fix the version number in versionadded
      - d403ae58d7 Add unit test for docker.update_mine
      - 44c275698a Actually it's more than just add/remove that updates the mine
      - 0cb6996b07 Add release notes mention of docker.update_mine config option
      - de05097b20 Add docs for new config option
      - 04c55a9178 Add note in mine.get_docker docstring about new config item
      - 4e456255c0 Allow mine update to be disabled using new config option

- **ISSUE** #48029: (vinian) syndic failed to auth when restart salt-master on syndic master (refs: #48034)
- **PR** #48034: (vinian) restart salt-syndic when salt-master restart @ *2018-06-21 20:12:37 UTC*
  - 9f0bd90c5b Merge pull request #48034 from vinian/restart-salt-syndic-when-salt-master-retart
  - f7652d8c8b restart salt-syndic when salt-master restart
- **PR** #48250: (rallytime) Update release notes for 2018.3.2 @ *2018-06-21 18:30:21 UTC*
  - ab2ba942ad Merge pull request #48250 from rallytime/2018.3.2-release-notes-update
  - 19c104b6cd Update release notes for 2018.3.2
- **PR** #48166: (terminalmage) Add trace logging and optimize file handling in grains.core.os_data @ *2018-06-21 18:02:33 UTC*
  - 51928ff050 Merge pull request #48166 from terminalmage/salt-jenkins-1000
  - b73df0ab00 Change call_args to call.args to reflect changes to MockFH
  - 0c2cc07704 remove unused import
  - b8c0a55cf3 Add test for tell()
  - 1cbe89feee Implement tell() for filehandle iteration
  - efb8f49d42 Add tests for read_data being a list, and containing unicode or bytestrings
  - 38df912fa6 Operate on a copy of the read_data
  - 71eeae1240 Update mount module unit tests to reflect changes in mock_open
  - 16c414e120 Update nfs3 module tests to reflect changes in mock_open
  - b7eab25d6c Update grub_legacy module tests to reflect changes in mock_open
  - cc002b8fd7 Update new network unit test to reflect changes in mock_open
  - a8f11594f1 Update new core grains tests to reflect changes in mock_open
  - 84ce18d956 Update crypt unit tests to reflect changes in mock_open
  - 67c036dc2d Add MockCall to tests.support.mock to track calls
  - 2556a1e13d Remove unused import
  - fd9d700157 Add additional docs to MockOpen class

- 6acb4c83ec Update newly added test to reflect renaming of handles attribute
- 77e5288d42 Update fibre_channel grains tests to reflect changes in mock_open
- ed40371a06 Update timezone module tests to reflect changes in mock_open
- fc0aa9934f Track call_count in MockOpen
- 2598d2453a Update snapper module tests to reflect changes in mock_open
- 51b3faa7b4 Add support for passing multiple strings for a given match in read_data
- 55487c175c Fix mock_open call to use new multifile syntax
- 7f516ef73a Update puppet module tests to reflect changes in mock_open
- 33a97c4ecc Update btmp beacon tests to reflect changes in mock_open
- bc027cfa94 Update wtmp beacon tests to reflect changes in mock_open
- f5823252bb Track call args/kwargs in MockFH
- 335591ca90 Mock the seek function in MockFH
- 7eb4b1ae1c Update cp.push test to reflect changes to mock_open
- 2be19cfa89 Report correct location when reading using explicit size and EOF reached
- 5ec95ba5ca On second thought, actually implement tell()
- 19022eb9e5 Add tell mock to MockFH
- 4b5a393445 Update junos tests to reflect changes to mock_open
- 278a222b09 Update dnsutil tests to reflect changes to mock_open
- 4e67955572 Replace the rest of mock_open with a class
- 75307a47c5 Update linux_sysctl tests to reflect changes to mock_open
- 05c68fd5d9 Use explicit config file and fix remaining mac_sysctl tests
- 836fde9a30 Allow Python 2 to accept an exception as read_data
- 543385fd02 Add writelines_calls property
- 42fa842456 Make read funcs mocks so their calls can be tracked
- 675f03c58f Update mac_sysctl tests to reflect new mock_open behavior
- 3d2c41d395 Update file module tests to reflect new mock_open behavior
- b9200dbc3e add a dict containing the handles to the mock_open return object
- 852ba4b982 Add mocked close() function to MockFH
- f6b46bc608 Remove unused import
- 20f60a769b Add blank lines to appease linter
- 48d7cfa6d3 Add multifile tests for mock_open
- 1861e9b944 mock_open: rewrite multi-file support
- 5e6b539770 Use function for empty string
- cb2620ad2b Update core grains tests to reflect EAFP changes
- 5d09b178d7 Separate mocked file contents per filename, not glob
- 875102f538 Modify mock_open to support multiple file paths

  - 096ace74df Move lsb_release parsing into its own function
  - 6a0828beed Add unit tests for mock_open
  - 8ba6cadac7 More mock_open bugfixes
  - 329dea218e Add a bunch of logging for linux os_data core grains
  - 2c64b270df Add timestamp to the minion's log_fmt_console
- **PR** #48216: (Ch3LL) Fix UnicodeDecodeError when reading file to determine virtual grain (refs: #48440) @ *2018-06-20 16:53:41 UTC*
  - 6072d1ef9c Merge pull request #48216 from Ch3LL/grains_unicode
  - c4334f3f14 Fix UnicodeDecodeError when reading file to determine virtual grain
- **PR** #48212: (Ch3LL) Fix python3 ec2 salt-cloud TypeError when installing salt (refs: #49041) @ *2018-06-20 16:00:21 UTC*
  - 77a75ebdcf Merge pull request #48212 from Ch3LL/py3_ec2
  - b0d75f459a Fix python3 ec2 salt-cloud TypeError when installing salt
- **ISSUE** #47984: (jeduardo) x509 module/state writing wrong certificate serial number to CRL (refs: #47986, #48080)
- **PR** #48209: (rallytime) Back-port #47986 to 2018.3 @ *2018-06-20 13:44:17 UTC*
  - **PR** #48080: (lusche) Bugfix #47984 messed up cert serial (refs: #48209)
  - **PR** #47986: (jeduardo) Fix serial number writing into CRL files with the x509 module (refs: #48209)
  - de614d31f5 Merge pull request #48209 from rallytime/bp-47986
  - f4b3bd5d2c Changed salt.utils.fopen to salt.utils.files.fopen
  - 8686872909 Fixed typos, removed repeated unit tests, and applied code fixes suggested by linter.
  - 382df48a72 Removed useless new line
  - 5aa99d14c4 Added unit tests for CRL creation and certificate revocation with CRL
  - cc12844922 Fixed a problem where the OpenSSL bindings refuse to consume unicode strings.
  - 399cf08860 Stopped converting the certificate hexadecimal serial number to an integer in order to avoid breaking CRLs.
- **PR** #48210: (rallytime) [2018.3] Merge forward from 2018.3.1 to 2018.3 @ *2018-06-20 13:43:19 UTC*
  - 1f88df6e89 Merge pull request #48210 from rallytime/merge-2018.3
  - dae65da256 Merge branch '2018.3.1' into '2018.3'
- **PR** #48186: (rallytime) Add autodoc module for saltcheck.py @ *2018-06-19 19:03:55 UTC*
  - 5b4897f050 Merge pull request #48186 from rallytime/saltcheck-docs
  - 314fc2d889 Clean up some doc references
  - da7603d879 Add autodoc module for saltcheck.py
- **PR** #48191: (twangboy) Use the –disable-pip-version-check option @ *2018-06-19 18:13:44 UTC*
  - 7274a33c80 Merge pull request #48191 from twangboy/fix_pip_check
  - 5c00fce6bf Use the –disable-pip-version-check option
- **ISSUE** #48122: (pasmon) Salt minion 2017.7.6 disregards "bin_env" in pip state (refs: #48189)

- **PR** #48189: (gtmanfred) If pip binary is passed to bin_env, use that pip binary (refs: #48207) @ *2018-06-19 18:08:13 UTC*
  - 96f79b0674 Merge pull request #48189 from gtmanfred/pip
  - 96c59f3d93 If pip binary is passed to bin_env, use that pip binary
- **PR** #48165: (terminalmage) Fix regression with top_file_merging_strategy=same @ *2018-06-19 18:03:07 UTC*
  - 71e385501b Merge pull request #48165 from terminalmage/issue48144
  - 92ac2a2d6a Remove unused imports
  - 15a44d5dd4 Remove redundant top file merging tests
  - 12100d9bd3 Add more top file merging tests
  - 0ed686cb3f Add unit test for show_top with "same" merging strategy
  - 35e5492fb7 Revert 7058f10 / 3df6fa7
- **ISSUE** #46806: (ezh) Lack of debug messages on authentication fail. (refs: #46807, #48179)
- **PR** #48179: (ezh) 2018.3 auth @ *2018-06-19 14:16:40 UTC*
  - **PR** #46807: (ezh) Reduce initial authentication setup complexity (refs: #48179)
  - 2a8e1c6539 Merge pull request #48179 from ezh/2018.3-auth
  - 9ed2d2ec55 Fix integration.shell.test_key and integration.shell.test_runner
  - 26a6f79730 Fix integration tests test_list_acc_wrong_eauth, test_salt_run_with_wrong_eauth
  - b84c4321c4 Add more verbose debug messages for auth subsystem
- **PR** #48188: (gtmanfred) allow virtual aliases to be used for the driver name @ *2018-06-18 21:49:16 UTC*
  - 0231008cf8 Merge pull request #48188 from gtmanfred/2018.3
  - 45249d3e10 allow virtual aliases to be used for the driver name
- **PR** #48116: (KaiSforza) Add jenkinsfiles to define tests in 2018 @ *2018-06-18 21:27:10 UTC*
  - 0f95238dca Merge pull request #48116 from KaiSforza/pr-lint-2018
  - f158bed5bd Add docs, kitchen tests
  - 904a70c187 Move .jenkins to .ci for future
  - 365fa0e51f Make it parallel so we don't fail right away
  - e594979745 Add a .jenkins file to run pylint
- **ISSUE** #48146: (rmarchei) mysql_query.run: exception on 2018.3 (refs: #48164)
- **PR** #48164: (garethgreenaway) [2018.3] Fixes to mysql state @ *2018-06-18 13:10:20 UTC*
  - 60c9490bdc Merge pull request #48164 from garethgreenaway/48146_mysql_output_to_file
  - a040643a82 Accounting for certain situations when the query result is not a string, but actually a dictionary.
- **ISSUE** #48113: (gaetanquentin) state file.line has error and erase file content completly, while with mode test=true it is ok (refs: #48156)
- **PR** #48156: (garethgreenaway) [2018.3] Unicode fixes for file.line @ *2018-06-17 19:34:08 UTC*
  - 85ebcbd3f2 Merge pull request #48156 from garethgreenaway/48113_ensure_writing_strings_with_file_dot_line
  - 5a9ef0d1ae Unless we're using py2 and Windows, ensure we're writing out a string when using file.line.

- **PR** #48161: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-06-17 19:33:29 UTC*
    - 33400e72db Merge pull request #48161 from rallytime/merge-2018.3
    - 0733fa1b02 Update utils path for which function to new path
    - 398cc78224 Merge branch '2017.7' into '2018.3'
    - df2a156338 Merge pull request #48061 from garethgreenaway/port_47049_2017_7
        - 7c472fed51 Fixing failing test_set_hwclock_aix test.
        - ccb0acc958 Porting #47049 to 2017.7.
    - 5ec3cf2dd4 Merge pull request #48143 from Ch3LL/fix_long_job
        - 48b5d2e9d3 Add timeout argument to run_salt for ShellCase
    - e1e566d1f8 Merge pull request #48135 from rallytime/release-notes-2017.7
        - 7a97f157b3 Add missing *v* for tag reference
        - 2f2b69ed37 Add "in progress" notation to 2017.7.7 release notes
        - 06a1151a63 Add release notes file for 2017.7.7
        - 885b2862ce Move 2017.7.7 release notes to 2017.7.8
    - ac9dabbfaa Merge pull request #48105 from rallytime/merge-2017.7
        - cdb45874de Merge branch '2017.7.6' into '2017.7'
            - 16916d6bd2 Merge pull request #47880 from gtmanfred/2017.7.6
            - 987ae6358b quote python_version in requirements.txt
            - 27670168ea Merge pull request #47875 from rallytime/release-notes-dot-six
            - 58dee4c829 Remove sentence about the release being in progress
            - b3dcb7330e Remove "in progress" too
            - 99e1df7823 Update 2017.7.6 release notes: remove "unreleased" text
            - 1f0bada07c Merge pull request #47873 from gtmanfred/2017.7.6
            - a1c1b5b5cf allow tornado 5.0 to be installed only for python2
    - 2da56a69d9 Merge pull request #48101 from rallytime/update-doc-refs-2017.7
        - 7ba6f5fb36 Update 2017.7.7 reference to 2017.7.8
    - 5a0e3d46e7 Merge pull request #48091 from terminalmage/fix-file_roots-monkeypatching
        - 6fe711ad76 Reverse monkeypatching after test_symlink_list finishes
    - 053b019a8f Merge pull request #48088 from rallytime/update_version_doc_2017.7
        - 1b8d1c936b Remove "in progress" info for 2017.7.6 release notes
        - 9a0f4d190a Update release versions for the 2017.7 branch
    - d6d2c5ee18 Merge pull request #48075 from garethgreenaway/48038_jobs_are_not_deduping_minion_side
        - c537b3275b Ensure that the shared list of jids is passed when creating the Minion. Fixes an issue when minions are pointed at multiple syndics.
    - aa33cc0b0c Merge pull request #48071 from terminalmage/gitfs-docs
        - 24545204b3 Fix inaccurate gitfs_saltenv example in GitFS Walkthrough

  - fb4ceacb88 Merge pull request #48053 from rallytime/bp-48040
    - 265b22b194 states/github.py fix for incorrect positional argument
  - f37dcaac6d Merge pull request #48024 from gtmanfred/2017.7
    - 56b074ab27 allow specifying a different state to run on converge
    - cc9c4b4d5a add pytest coverage and xml junits
- **ISSUE** #48145: (max-arnold) Failed to load ext_pillar saltclass: can only assign an iterable (refs: #48155)
- **PR** #48155: (max-arnold) Do not fail on empty saltclass classes (fix for #48145) @ *2018-06-15 20:21:28 UTC*
  - 9aa9a83b10 Merge pull request #48155 from max-arnold/2018.3
  - 63ab02c648 Do not fail on empty saltclass classes declaration #48145
  - 3d4fcbe3aa Add test for issue #48145
- **PR** #48104: (twangboy) Fix *unit.modules.test_win_status @ 2018-06-15 17:05:34 UTC*
  - 152c09cc0c Merge pull request #48104 from twangboy/fix_test_win_status
  - 1ec3f436ee Fix test_error_logged_if_process_get_owner_fails
- **PR** #48147: (brejoc) Fix for sorting of multi-version packages @ *2018-06-15 17:03:18 UTC*
  - 8cf03110d4 Merge pull request #48147 from brejoc/2018.3-multi-version-fix
  - 90ed25447d Swtiching to salt.utils.versions like linter suggested
  - ed09574073 Fix for sorting of multi-version packages (bsc#1097174 and bsc#1097413)
- **PR** #47956: (dmurphy18) Improved support for mount on AIX @ *2018-06-15 13:40:01 UTC*
  - 2edc5f5442 Merge pull request #47956 from dmurphy18/aix_mount_support
  - f2a3e321db Improved support for mount on AIX
- **ISSUE** #48012: (Timandes) Found UnicodeDecodeError when trying to start Salt Master (refs: #48081)
- **PR** #48081: (terminalmage) Fix UnicodeDecodeError when parsing hosts file with non-ascii @ *2018-06-15 12:41:13 UTC*
  - c05c176782 Merge pull request #48081 from terminalmage/issue48012
  - 8d2fb0bf25 Fix cp.push test
  - e230a7223f Fix definition of test data for wtmp/btmp beacon tests
  - 82874a8c10 Import six like we do everywhere else
  - 54f9a2ab54 Remove unused import
  - 767cc7e87f fix tabs
  - db19636f56 Get rid of additional newline append
  - 7427e192ba Fix more crappy mocking
  - 4eaa5789ce Fix crappy mocking
  - 248467edac Add regression test for _generate_minion_id()
  - 5e62d6d45f Fix UnicodeDecodeError when parsing hosts file with non-ascii
  - e6a4744f85 Use errno instead of hard-coding error 2
  - ff63f36932 Fixes / enhancements for mock_open

- **PR** #48133: (rallytime) Updates the 2018.3.2 and 2018.3.3 release notes files @ *2018-06-14 21:21:15 UTC*
  - **PR** #48129: (rallytime) Add release notes for 2018.3.2 (refs: #48133)
  - 5d92e2763a Merge pull request #48133 from rallytime/release-notes-2018.3
  - afe1e91972 Add "in progress" notation to 2018.3.2 release notes
  - 3e4272ac09 Add release notes for 2018.3.2
  - 88c584cb0d Move 2018.3.2 release notes to 2018.3.3
- **PR** #48103: (terminalmage) Fix for gitfs base env being pinned to commit ID @ *2018-06-14 16:29:24 UTC*
  - 0e3f846836 Merge pull request #48103 from terminalmage/gitfs-base-pinned-to-commit
  - 31ab2fe8de Fix for gitfs base env being pinned to commit ID
- **PR** #48077: (twangboy) Fix issue with *salt.utils.parsers* on Windows @ *2018-06-14 16:22:57 UTC*
  - 4b6f1c7f75 Merge pull request #48077 from twangboy/fix_parsers
  - 0f7d7691a2 Call logger in each case
  - 0e99dd741c Fix logic for non-root
  - 2d2534a688 Fix parsers for Windows, fix tests
- **PR** #48096: (twangboy) Fix *unit.utils.test_which* on Windows @ *2018-06-14 13:15:14 UTC*
  - 80332b32a7 Merge pull request #48096 from twangboy/fix_test_which
  - d26fc56f13 Use os.sep for paths
- **PR** #48102: (rallytime) [2018.3] Update 2018.3.2 references to 2018.3.3 @ *2018-06-14 13:14:26 UTC*
  - 66b2200913 Merge pull request #48102 from rallytime/update-doc-refs-2018.3
  - 1c9bcce3d8 Update 2018.3.2 references to 2018.3.3
- **PR** #48109: (rallytime) Back-port #47851 to 2018.3 @ *2018-06-14 13:09:04 UTC*
  - **PR** #47851: (rares-pop) Fixup! add master.py:FileserverUpdate **kwargs (refs: #48109)
  - 2902ee0b14 Merge pull request #48109 from rallytime/bp-47851
  - e9dc30bf8e Fixup! add master.py:FileserverUpdate **kwargs
- **ISSUE** #47925: (JonGriggs) GitFS looking for files in the master branch only (refs: #47943)
- **PR** #47943: (terminalmage) Make sure we set the effective environment when lock_saltenv is True @ *2018-06-13 20:02:00 UTC*
  - 2529292568 Merge pull request #47943 from terminalmage/issue47925
  - 534e1a7100 Merge branch '2018.3' into issue47925
- **PR** #48089: (rallytime) Update release versions for the 2018.3 branch @ *2018-06-13 14:03:44 UTC*
  - 9e1d0040e4 Merge pull request #48089 from rallytime/update_version_doc_2018.3
  - fad6a0991e Remove "in progress" info for 2018.3.1 release notes
  - a3b3b0a0e1 Remove "in progress" info for 2017.7.6 release notes
  - f9be1b9125 Update release versions for the 2018.3 branch
- **PR** #48054: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-06-13 12:58:46 UTC*
  - b10b7355a0 Merge pull request #48054 from rallytime/merge-2018.3

  - 508d70fabf Update old utils paths to use new paths
  - 3d2ea16c3a Merge branch '2017.7' into '2018.3'
    - ffee3576ec Merge pull request #48044 from twangboy/wua_add_fail_code
      - d41d0c25eb Add faile code for WinHTTP send/receive error
    - 871a910fcb Merge pull request #48009 from dwoz/winswarmfix
      - 5027c7bb84 minionswarm runs on windows
    - 0dcaead36d Merge pull request #47968 from gtmanfred/2017.7
      - ae1b0d28bb pass LANG and HOME into tox commands
      - 8d15b93d00 remove tox virtualenvs from doc tests
      - e1872e2036 simplify tox.ini
      - 81bd01f5af add tests.txt for running tests with tox requirements
      - 556a2067fc fix masterapi test to use testing directory for configs
      - 5a41f484ef add EXPENSIVE_TESTS
      - 055cd5a6ba set DESTRUCTIVE_TESTS environment variable for pytest
    - 42dd6b83eb Merge pull request #47978 from twangboy/fix_test_pkg
      - 8bb36b3a4b Add try/finally, fix typo in 7zip def file
    - a67b21ef3d Merge pull request #47959 from twangboy/fix_test_service
      - 0cd47aa81e Remove unused import
      - d4b42540e3 Fix failing test when service doesn't exist
    - b81d482067 Merge pull request #47983 from garethgreenaway/backport_47843_2017_7
      - 7b7fb1f5d5 Adding missing test sls file.
      - 54e51bc627 Backporting #47843 to 2017.7, updating test for 2017.7.
    - d56ddad22c Merge pull request #47533 from twangboy/fix_47178
      - 24717cbc21 Change version added to 2017.7.7
      - 0592f1bac2 Fix issues with functions that user makedirs on Windows
    - 5c56b8c755 Merge pull request #47827 from twangboy/fix_47791
      - fbbd91f09e Add more descriptive debug message
      - 365f81651b Fix deprecated exception handling
      - 8dd6710b93 Use local instead of network
      - 980d99d74b Fix issue when archive is on mapped drive

- **PR** #48060: (gtmanfred) mark test as expensive @ *2018-06-13 12:58:21 UTC*
  - c83818e4f9 Merge pull request #48060 from gtmanfred/2018.3
  - 914935c4d4 Merge branch '2018.3' into 2018.3
- **PR** #48057: (terminalmage) Fix link without target in 2018.3.0 release notes @ *2018-06-11 18:50:17 UTC*
  - a4eb41623d Merge pull request #48057 from terminalmage/docs-2018.3
  - 20f71ff6f6 Fix link without target in 2018.3.0 release notes

  - d8c035e5e5 mark test as expensive

- **PR** #48042: (terminalmage) Switch to trace level logging for further test failure troubleshooting @ *2018-06-11 14:03:48 UTC*
  - 0f9a3122df Merge pull request #48042 from terminalmage/salt-jenkins-1000
  - b33a0b5eaa Switch to trace level logging for further test failure troubleshooting
- **PR** #48041: (terminalmage) salt.utils.hashutils: Only decode to utf-8 on Windows @ *2018-06-11 13:31:41 UTC*
  - 6c5389189a Merge pull request #48041 from terminalmage/hashutils-fix-windows
  - 58c7cd33d7 salt.utils.hashutils: Only decode to utf-8 on Windows
- **PR** #48014: (cro) Find job pause @ *2018-06-08 13:48:45 UTC*
  - dcae209951 Merge pull request #48014 from cro/find_job_pause
  - 440aa67c4f Lint: Add blank line
  - 7b0e99a511 One more case where returner doesn't respond
  - 5abeedf882 Catch two cases when a returner is not able to be contacted–these would throw a stacktrace.
- **PR** #47915: (garethgreenaway) [2018.3] state runner pause resume kill @ *2018-06-07 16:08:30 UTC*
  - ea85f882bc Merge pull request #47915 from garethgreenaway/state_runner_pause_resume_kill
  - 2ecbe9c034 fixing typo in alias_function call.
  - 4377e0cc08 Some fixes to the set_pause and rm_pause function in the state runner, renaming to in line with the functions in the state module. Including aliases to previous names for back-ward compatibility. Including a soft_kill function to kill running orchestration states. A new test to test soft_kill functionality.
- **PR** #48002: (rallytime) Back-port #47923 to 2018.3 @ *2018-06-07 15:25:39 UTC*
  - **PR** #47923: (isbm) Prevent crash if files in file.recurse (refs: #48002)
  - 5c11fcde60 Merge pull request #48002 from rallytime/bp-47923
  - 9465e5f1e9 Use to_unicode from stringutils avoid deprecation warning
  - 73d33cbfc3 Prevent crash if files in file.recurse
- **PR** #48003: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-06-07 15:25:19 UTC*
  - 3d0a087c71 Merge pull request #48003 from rallytime/merge-2018.3
  - a6533a9332 Merge branch '2017.7' into '2018.3'
    - 998a1c13d5 Merge pull request #47966 from mbunkus/fix-postfix-prefix-free-key-names-assumption
      - 0ae402d1b5 Fix postfix.set_main's assumption of prefix-free key names
    - df50ce7b1b Merge pull request #47824 from twangboy/fix_setup.py
      - 4538b3abb3 Remove 'len-as-condition' disablement
      - d4efcc3c8b Skip lint errors
      - 1b3977f8d4 Remove m2crypto download for Windows
    - 3e91a31134 Merge pull request #47922 from damon-atkins/2017.7_win_pkg_keys_fix
      - 7129203b1b Merge branch '2017.7' into 2017.7_win_pkg_keys_fix
    - 8008fca2f6 Merge pull request #47933 from terminalmage/add-pillar.get-docs

        - 4eb022b675 Clarify pillar.get docs
      - ab565bfaa3 Merge pull request #47944 from terminalmage/fix-git.detached-docs
        - 1d082b4389 Replace use of deprecated argument name in git.detached docstring
        - 6253793cb6 Merge branch '2017.7' into 2017.7_win_pkg_keys_fix
        - 4ce0fe6546 win_pkg under py3 keys returns a view instead of a list, wrap keys in list()

- **PR** #48022: (morganwillcock) win_wua state: fix function name in examples @ *2018-06-07 15:03:52 UTC*
  - 0a05212f60 Merge pull request #48022 from morganwillcock/patch-1
  - 443f4d6f21 win_wua state: fix function name in examples
- **ISSUE** #47546: (ender8282) archive.extracted fails when archive includes files with non-ascii names with 2018.3.0 (refs: #48015, #47572)
- **PR** #48015: (garethgreenaway) [2018.3] more unicode nonsense in archive module @ *2018-06-07 14:44:55 UTC*
  - 406efb161e Merge pull request #48015 from garethgreenaway/47546_more_unicode_nonsense
  - f457f9cb84 Adding a test to ensure archive.list returns the right results when a tar file contains a file with unicode in it's name.
  - 9af49bc595 Ensure member names are decoded before adding to various lists.
- **ISSUE** #47955: (frogunder) 2018.3.1 Creating Windows machine in Amazon using salt-cloud fails. (refs: #47989)
- **PR** #47989: (dwoz) Properly decode password from aws using m2crypto @ *2018-06-06 20:44:28 UTC*
  - 1ce7d6c38a Merge pull request #47989 from dwoz/awscloud
  - 23ab2727be Fix linter errors
  - 6c8daa2b85 Properly decode password from aws using m2crypto
- **PR** #47976: (terminalmage) Capture and log output of processes that timed out and were killed @ *2018-06-06 19:17:08 UTC*
  - 76068c83f5 Merge pull request #47976 from terminalmage/salt-jenkins-1000
  - 41e5a75027 Add catch_timeout to run_script
  - 551ada8e4d Capture and log output of processes that timed out and were killed
- **PR** #47990: (terminalmage) Docs fixes in 2018.3 release branch @ *2018-06-06 17:15:17 UTC*
  - f12a52736e Merge pull request #47990 from terminalmage/2018.3-docs
  - f89d2ccd30 Clean up Sphinx warnings
  - f5fc936865 Add clarifying comment so that we don't break this
  - 94849780ac fix docs build error due to IndexError
- **PR** #47982: (twangboy) Fix *unit.modules.test_win_dns_client* tests @ *2018-06-06 15:02:10 UTC*
  - 9d56af3cad Merge pull request #47982 from twangboy/fix_test_win_dns_client
  - 2dcdf6788c Force string to ModuleType
- **PR** #47985: (twangboy) Fix *unit.states.test_reg* test failure @ *2018-06-06 14:56:03 UTC*
  - 1b4c72345b Merge pull request #47985 from twangboy/fix_test_reg
  - 4ce551811f Remove the unnecessary *to_unicode* stuff
  - 863ded39fa Mock *reg.cast_vdata*

- **ISSUE** #47236: (MorphBonehunter) x509.private_key_managed broken after upgrade to 2018.3.0 (refs: #47957)
- **PR** #47957: (garethgreenaway) [2018.8] Ensure x509 passphrase is a string @ *2018-06-05 13:27:33 UTC*
  - 84100570f7 Merge pull request #47957 from garethgreenaway/47236_ensure_passphrase_is_a_string
  - 99bad3cca6 removing unnecessary change
  - e22df8473d Ensuring that when a passphrase is passed in, it is returned as a string from the passphrase callback.
    - d0cf06a24d Make sure we set the effective environment when lock_saltenv is True
      - 8da6ce45dc Finish the docstring for git.cloned state
      - 4a8260a5e5 Add tests for git.cloned state
      - 311179da05 Add git.cloned state
- **PR** #47926: (dmurphy18) Backport of PR 47808 to 2018.3, improved grains support for AIX @ *2018-06-04 19:31:24 UTC*
  - **PR** #47808: (dmurphy18) Updated AIX support for grains reported (refs: #47926)
  - c355da9e0d Merge pull request #47926 from dmurphy18/backport_pr47808
  - ac15d2093a Backport of PR 47808 to 2018.3, improved grains support for AIX
- **PR** #47927: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-06-04 18:55:12 UTC*
  - a46e42ab11 Merge pull request #47927 from rallytime/merge-2018.3
  - 5baab66f4b Lint: Whitespace fix
  - 7e1d278d19 Update old utils paths to use new utils paths
  - 3273bbdab7 Merge branch '2017.7' into '2018.3'
    - a2b588299b Merge pull request #47860 from terminalmage/state-writing-docs
      - 8cb33d3dec Quiet the linter
      - af51e16f23 Use more elegant RST syntax
      - 91cd57d1e0 Update mocking to reflect changes in service module
      - b9ace5a859 Fix crappy mocking
      - bc9e3acef3 Lint fixes
      - d37f7e4f04 Add 2017.7.6 release notes
      - 9dddeeefab Improve documentation on syncing states
      - f465fa3ca7 Add best practices docs for writing states
      - 93ee5ee2b0 Fix all Sphinx warnings
      - 64b9b4d0b8 Clarify that name would override the id declaration
    - d2f3377b5b Merge pull request #47877 from dwoz/sleep_audit
      - ebc7cde9cb Revert job chunk wait time change
    - 735e92093e Merge pull request #47917 from dwoz/winsuite
      - 118601ebd6 Fix windows tests suite breakage
    - 63efb76d51 Merge pull request #47822 from Ch3LL/win_user_test
      - 3a691b405f add user_home path for both windows and linux

      - 788abf771e Add user state integration tests to windows
    - b9da4f1221 Merge pull request #47876 from doesitblend/add-file-read-windows
      - 3f7e7ec327 Add file.read function to Windows module
    - cbe2ecfae8 Merge pull request #47882 from frogunder/45013
      - 38d114a2d2 add whoami test
- **PR** #47931: (rallytime) Back-port #47913 to 2018.3 @ *2018-06-04 14:39:16 UTC*
  - **PR** #47913: (rares-pop) Fixup! beacons/avahi_announce.py finding grains (refs: #47931)
  - 4db1f8c603 Merge pull request #47931 from rallytime/bp-47913
  - c51e732f4d Fixup! beacons/avahi_announce.py finding grains
- **PR** #47930: (rallytime) Back-port #47725 to 2018.3 @ *2018-06-04 14:38:58 UTC*
  - **PR** #47725: (bbczeuz) network_settings beacon: Update example to follow new cfg convention (refs: #47930)
  - d05a6c70d3 Merge pull request #47930 from rallytime/bp-47725
  - 5c666409f3 Spelling fix
  - cfe0a5bb42 network_settings: Fix doc using new list/dict syntax
- **PR** #47557: (L4rS6) fix broken rabbitmq list policies in rabbitmq version 3.7 @ *2018-06-04 13:25:07 UTC*
  - 484d83014b Merge pull request #47557 from L4rS6/fix-broken-rabbitmq-list-policies
  - 6d7ef27557 Merge branch '2018.3' into fix-broken-rabbitmq-list-policies
  - e043ea2833 refactored list_policies code and added more tests
  - eea81feb7b fix broken rabbitmq list policies in rabbitmq version 3.7
- **PR** #47949: (terminalmage) Continued troubleshooting on salt-jenkins 1000 @ *2018-06-04 13:17:14 UTC*
  - 866a2fbf5e Merge pull request #47949 from terminalmage/salt-jenkins-1000
  - 9c369c6aa8 Remove temp logging from test
  - af02d5ba7b Add result logging to run_script, make returns DRY
- **PR** #47941: (terminalmage) Update test logging for salt-jenkins @ *2018-06-03 02:40:56 UTC*
  - 0b85d1273f Merge pull request #47941 from terminalmage/salt-jenkins-1000
  - fb66368af9 Update test logging for salt-jenkins 1000
- **PR** #47924: (Ch3LL) Catch all exceptions in git import for salt.utils.gitfs @ *2018-06-01 21:06:58 UTC*
  - 58b5f2b4dd Merge pull request #47924 from Ch3LL/mac_gitpy
  - 75c51ad69b Catch all exceptions in git import for salt.utils.gitfs
- **ISSUE** #47269: (isdig) Reopen Windows Minion That Has IAM Role Cannot Access to S3 (refs: #47879)
- **PR** #47879: (dwoz) Fix ami role usage warts #47269 @ *2018-06-01 14:11:55 UTC*
  - b88ec5258d Merge pull request #47879 from dwoz/ami_role_fix
  - 3884c2cf5f Fix ami role usage warts #47269
- **PR** #47788: (twangboy) Use pytz to calculate timezones @ *2018-05-31 18:33:20 UTC*
  - b09596982b Merge pull request #47788 from twangboy/fix_47559

  - 0d229c6182 Merge branch '2018.3' into fix_47559
  - 9e612ec9e7 Fix markup in release notes
  - 60499d18f0 Skip test if pytz not present
  - 72cc361c7b Move pytz to 3rd party import, add to __virtual__
  - b8a6488688 Update __virtual__ function
  - 9923176b68 Use __utils__, fix unit tests
  - 73e033f555 Return offset in the same format as Unix
  - 5656183c5e Add timezone.list, add 2018.3.2 release notes
  - 986f6c9b2a Use pytz to calculate timezones
- **PR** #47899: (terminalmage) salt-jenkins issue 1000 @ *2018-05-31 18:24:49 UTC*
  - ae73cb3947 Merge pull request #47899 from terminalmage/salt-jenkins-1000
  - 25afc932f7 WIP salt-jenkins issue 1000
- **PR** #47753: (Ch3LL) Add stderr launchctl helper class and fix service mac tests @ *2018-05-31 18:16:57 UTC*
  - 8f78e3aef6 Merge pull request #47753 from Ch3LL/service_mac
  - 185c9e9ae2 only stop service if its running
  - e15e674955 Add stderr launchctl helper class and fix service mac tests
- **PR** #47908: (meaksh) Align SUSE salt-master.service 'LimitNOFILES' limit with upstream Salt @ *2018-05-31 18:13:27 UTC*
  - 828bf39695 Merge pull request #47908 from meaksh/2018.3-align-suse-salt-master-service
  - efe308013a Align SUSE salt-master.service 'LimitNOFILES' limit with upstream Salt
- **PR** #47868: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-05-30 21:20:38 UTC*
  - b4bac9b41b Merge pull request #47868 from rallytime/merge-2018.3
  - e5d386e91c Update solaris core grains test
  - 267f09c1a0 Lint: Remove unused import
  - 120ee16b70 Replace old utils paths with new utils paths
  - dc32b67c03 Merge branch '2017.7' into '2018.3'
    - 4b28e510f1 Merge pull request #47853 from dwoz/py3-cloud-test
      - c2f8aef7c5 Fix for py3 ec2 cloud tests
    - 215efa6b73 Merge pull request #47847 from garethgreenaway/47833_reactor_runner_documentation
      - 03676712de Adding some addition documentation to the reactor runner indicating that the reactor system must be active prior to using it.
    - 0bf651b470 Merge pull request #47846 from Ch3LL/p_refresh_test
      - 019edad8e4 Fix flaky refresh pillar integration test
    - 799fce979d Merge pull request #47552 from twangboy/fix_46981
      - cffc9b7806 Merge branch '2017.7' into fix_46981
      - 81308a4a44 Add release notes for 2017.7.7

    - 008af0ac6b Fix unit tests
    - 87097eefb6 Add comments about how get is returning data
    - f1f1bfc5c1 Show GPO settings, raise error if trying to set gpo managed settings
- ffc2ebc373 Merge pull request #47691 from lomeroe/solaris_product_grains
    - 6889befc6b Merge branch '2017.7' into solaris_product_grains
    - 49053bc106 lint fix
    - 6f185c9179 another lint fix
    - dbffba6876 fix tons of errors in my tests
    - 8c9355d34c Lint fix
    - b29ec75da7 Update regexes in core grains for detecting the 'product' grain on Solaris Sparc systems. Additionally, copy the 'product' grain to 'productname' to be consistent with other OSes.
- 96b7c0cc76 Merge pull request #47778 from Ch3LL/win_state_pkg
    - e1786de6b0 Merge branch '2017.7' into win_state_pkg
    - f8c467d3e6 Fix text editor error
    - 872e162137 Add test_pkg integration state tests to windows
- 6f934c2259 Merge pull request #47823 from Ch3LL/win_dns_test
    - 7848114d6a Add win_dns module integration tests for windows
- e63e04602d Merge pull request #47838 from rallytime/bp-47810
    - bd2b62fa66 better debug message
    - 83565c55df Address PR feedback
    - 50bce3a2f3 make decode_linode_plan_label a private function
    - 2ba4fc4cea fix raising when a 'GB' format invalid plan is supplied
    - 3afb50d5a2 slight cleanup
    - 319fbd3406 match quotation mark types properly
    - 95e0202223 more consistent use of parens in logged warning
    - f52926ca87 log a warning when the user supplied a label we could decode but was not in the proper format
    - 121303d827 reduce complexity of get_plan_id by moving decoding of the user-supplied label to its own function
    - cf534c7314 Fix Linode plan selection
- 9ef6e60617 Merge pull request #47805 from twangboy/add_sign.bat
    - 66d8b0331a Add sign.bat script for signing packages
- 93420950d3 Merge pull request #47500 from dwoz/winuser
    - 0ca5224cbf Merge branch '2017.7' into winuser
    - e5948902af Use salt utils method for this_user
    - a056a293f1 Centeralize test username lookup

      - 6a6ab69722 Get the current username on windows
    - 552526f309 Merge pull request #47802 from rallytime/bp-40892
      - bb357da084 add minion function to reload beacon #35960
    - c9cccf53b9 Merge pull request #47816 from Ch3LL/disable_windows
      - 2509d36888 Add windows to service disable ERROR check in tests
    - 2cb6634c6b Merge pull request #47773 from frogunder/45011
      - 7c9b0bda33 add win_servermanager.list_available test
    - e65dece202 Merge pull request #47807 from dwoz/winrmtests
      - 1f1cc1357a Increase instance size for cloud tests
      - be8dcd21f1 Try an even bigger timeout
- **PR** #47900: (rallytime) [2018.3] Merge forward from 2018.3.1 to 2018.3 @ *2018-05-30 18:09:41 UTC*
  - 21550dbd2d Merge pull request #47900 from rallytime/merge-2018.3.1-to-2018.3
  - 2801eecb26 Merge branch '2018.3.1' into '2018.3'
- **ISSUE** #30367: (paulfanelli) salt-run orchestration via state.sls always successful; it is ignoring state return value (refs: #47843)
- **PR** #47843: (garethgreenaway) [2018.3] orchestration results False when function is False (refs: #47983) @ *2018-05-30 17:21:54 UTC*
  - a0bf565741 Merge pull request #47843 from garethgreenaway/30367_orch_return_false_when_function_fails
  - 09242697b8 Merge branch '2018.3' into 30367_orch_return_false_when_function_fails
- **PR** #47866: (rallytime) Back-port #47814 to 2018.3 @ *2018-05-29 19:55:24 UTC*
  - **PR** #47814: (travispaul) Bugfixes and unit tests for pkgin module (refs: #47866)
  - da6f7a5538 Merge pull request #47866 from rallytime/bp-47814
  - d50c0ab96b Lint test_pkgin.py
  - 4dac0b4a31 pkgin latest_version bugfix
  - 4ae0313797 Bugfixes and unit tests for pkgin module
- **PR** #47865: (rallytime) Back-port #47600 to 2018.3 @ *2018-05-29 19:54:44 UTC*
  - **PR** #47600: (travispaul) Prevent crash on NetBSD and OpenBSD when no swap is configured. (refs: #47865)
  - d9e7f2b5b1 Merge pull request #47865 from rallytime/bp-47600
  - f9f464fa51 Prevent crash on NetBSD and OpenBSD when no swap is configured.
- **PR** #47221: (azelezni) Fix boto_ec2 unused parameter "network_interfaces" @ *2018-05-29 14:41:35 UTC*
  - 61561a8279 Merge pull request #47221 from azelezni/fix-boto_ec2-network_interfaces
  - 03ee0023eb Fix unused network_interfaces parameter
  - 1ece61ddfa Fix unused network_interfaces parameter
  - 3e074be9c3 Fixing lint
  - 02609b6e61 Adding state files for new test.

  - 377e34c689 Updating function in saltmod to ensure that the result is a failure if the function being run returns as False.

- **PR** #47850: (rallytime) Fix autodoc for new swarm module @ *2018-05-26 01:53:16 UTC*
  - c4d828c986 Merge pull request #47850 from rallytime/swarm-autodoc
  - 467c8fa45c Fix autodoc for new swarm module
- **PR** #47800: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-05-26 01:17:40 UTC*
  - 30b31c5db5 Merge pull request #47800 from rallytime/merge-2018.3
  - 9e3ce39e8c Typo: test_type –> text_type
  - 8877489386 Fix bad merge caught by test in zpool state
  - c328450be2 Update old utils paths to new paths
  - 4abfd26e86 Merge branch '2017.7' into '2018.3'
    - 46618d2ce7 Merge pull request #47765 from meaksh/2017.7-fix-zypper-error-reading-repo
      - 4475ba19b8 Prevent zypper from parsing repo configuration from not .repo files
    - 0e87559ee3 Merge pull request #47781 from rallytime/update-linode-sizes
      - a90c1b760e Update cloud test profile and docs to use new Linode size lables
    - 3ddc56cb9b Merge pull request #47748 from rallytime/merge-2017.7
      - b75f5ae38e Merge branch '2017.7.6' into '2017.7'
      - edf94c915e Merge pull request #47775 from gtmanfred/2017.7.6
      - 548f65d056 catch UnsupportedOperation with AssertionError
      - 17596f3025 Merge branch '2017.7.6' into '2017.7'
      - 8c38ecd75f Merge pull request #47769 from gtmanfred/2017.7.6
      - 3fdfc0fa82 skip test that breaks test suite
      - e458f4a031 Merge branch '2017.7.6' into '2017.7'
      - 0d5b473ce2 Merge pull request #47747 from Ch3LL/rn_2017.7.6
      - d4aa83b92d Add changelog to 2017.7.6 release notes
      - 8a5b34f7d9 Merge pull request #47702 from damon-atkins/2017.7.6_fix_pkg.latest_state
      - adcc094e08 Merge branch '2017.7.6' into 2017.7.6_fix_pkg.latest_state
      - d610c192d9 Merge pull request #47700 from yannj-fr/2017.7.6
      - 961c1ef61e fix roots modification time check
      - 2a73e905df Merge branch '2017.7.6' into 2017.7.6
      - 266749420f Merge pull request #47632 from gtmanfred/2017.7.6
      - 2c50c0d2f5 fix pylint
      - 4a29057b16 Fix last test for tornado
      - 550ef2e272 allow using tornado 5.0
      - 62e468448b handle new _create_stream in tornado 5.0
      - 2643c356af Merge pull request #47720 from rallytime/bp-47692-2017.7.6

    - 6e5cb36839 Default windows to m1.small for ec2-classic
    - 20d9785244 fix roots modification time check
    - aef37dd1ce fix roots modification time check
    - d51662e053 Ensure targeted_pkgs always contains value for non-windows.
    - 83b4224cf8 Adjusted based on feed back.
    - 12f983ce9f Whitespace lint issues
    - 075d3d3c49 pkg.install execution module on windows ensures the software package is installed when no version is specified, it does not upgrade the software to the latest. This is per the design. pkg.latest must provide the versions to install to pkg.install
- 0ddc864f90 Merge pull request #47779 from dwoz/winrmtests
    - d5eafe9d53 Use common timeout variable
    - 7e948eb540 Increase ec2 cloud timeouts to 1000
- 17c4c8443c Merge pull request #47430 from baniobloom/vpc_association_not_found
    - 0f1d007f91 added catch for VPCAssociationNotFound
- 9c5dde1449 Merge pull request #47525 from baniobloom/duplicate_sec_group_rule
    - dcc35df489 if we are trying to add the same rule then we are already in the desired state, return true
- 536ce2fa42 Merge pull request #47772 from gtmanfred/2017.7
    - 67756a50fd lock down dependencies for kitchen-salt
- 12b330f049 Merge pull request #47749 from terminalmage/issue47717
    - cd484bef69 Fix ”dnf list upgrades” parsing
- 5cd18b9e70 Merge pull request #47755 from dwoz/winrmtests
    - 57dd89e6c3 Default to ec2 classic compatible images
    - f89668920a Be explicit about winrm setting
- f45a96ba1d Merge pull request #47668 from Ch3LL/win_pkg_test
    - a8981024de Add pkg.latest_version windows test
- 7b12444dfa Merge pull request #47750 from dwoz/cloudtest
    - 63b722ba21 Allow ssh_interface to default to public_ips
- 27fbb42210 Merge pull request #47737 from dwoz/win_timeout
    - e88833a07d Cloud test fixup
- 1a93f060fb Merge pull request #47729 from UtahDave/fix_get_info_doc
    - 443a2d72a2 fix cli example to match function name
- 00a13761c7 Merge pull request #47682 from terminalmage/issue47182
    - d0243e8f23 Suppress spurious lint failure
    - 3b449f11fc Add regression test for excludes issue
    - 28a7d2b81c Skip __exclude__ in find_sls_ids
- 518f7bcc62 Merge pull request #47708 from darkpixel/47696-do-not-enumerate-none

  - 54e9bf9ec9 Merge branch '2017.7' into 47696-do-not-enumerate-none
- cad062eaa7 Merge pull request #47681 from rallytime/merge-2017.7
  - 6c06cb3ae3 Merge branch '2017.7.6' into '2017.7'
- fd2d45d9c2 Merge pull request #47683 from Ch3LL/mac_user_test
  - 5a1b25b9c7 Remove unnecessary setUp in states.test_user test for mac
- f323799c42 Merge pull request #47692 from dwoz/win_instance_type
  - 20b6070d54 Default windows to m1.small for ec2-classic
- 990ece5cd5 Merge pull request #47665 from gtmanfred/cloud
  - fd8a02decb use dictupdate to update the providers dictionary and merge lists
- a2ed8cbb7f Merge pull request #47151 from zer0def/configparser-defaultsect
  - a8f2ad977e Merge branch '2017.7' into configparser-defaultsect
  - cca2a806c9 Made interaction with [DEFAULT] section in ConfigParser as sane as upstream permits.
- e6fb8342ef Merge pull request #47673 from terminalmage/bp-45808
  - 454291ad62 Fix errors when attempting to cache files with long names or URLs
- b0446aab07 Merge pull request #47670 from gtmanfred/2017.7
  - dfd5a8715f add a pytest.ini and update a dependency for kitchen
- 0d06da6944 Merge pull request #47626 from gtmanfred/2017.7
  - 418364a533 make dev_python27.txt have the same message as dev_python34.txt
  - d53b44d10f remove lock file
  - 493ed7f93d use the toml dict format
  - e0f7cc1447 add proxy tests decorator to pytest conftest
  - 0a621dd0ec add Pipfile for managing dependencies in salt
  - f78b81db94 simplify dev and base.txt to single files
- a07a8906a0 Merge pull request #47578 from Ch3LL/mac_service_disabled
  - 6032a01f55 change disable check for upstart service on ubuntu14
  - 33b4cfbc5b Change service assertion check for systemd platforms
  - 14896f9743 change codeauthor and class name
  - 502c5bdff5 Ensure mac_service.disabled is correctly querying services

95d4488f01 Merge branch '2017.7.6' into 47696-do-not-enumerate-none

16c2153385 Merge pull request #47667 from Ch3LL/mac_user_enable

ba40d3d1a1 Update test_mac_user_enable_auto_login to test both py2 and py3

a4921e86c9 Merge pull request #47645 from Ch3LL/py3_rm_pip

225d90ad4c query the pip path for test test_issue_2087_missing_pip

e441733ac1 Merge pull request #47646 from rallytime/bp-47601-and-47643

    9e1d1a5ef8 Fix typo

    4e94609136 Remove unwanted file

    0109249c78 use ignore-undefined-variable

    37caecb7f4 Ignore pylint WindowsError

    c1135d90c7 Better doc string

    e53d6b9ed9 Skip tests when we can not use runas

    041e4c6ddb Wrap properties loop with a check for None. Closes GH-47696

- **PR** #47581: (twangboy) Add get_encoding salt util @ *2018-05-25 20:40:07 UTC*
  - da9eaa1825 Merge pull request #47581 from twangboy/fix_47274
  - ff6600f25e Merge branch '2018.3' into fix_47274
  - 6eff2f847b Add suggested changes
  - e27e9fd1e7 Fix tests on Py3
  - f398cbbdda Use os.linesep.join instead of textwrap.dedent
  - 771392e299 Fix unit tests, add newline='' to io.open
  - c0f735dde3 Remove comment
  - 6d877bb48b Add comment
  - 9f369d3f22 Remove to_encoding, create get_encoding
  - 68be0f9ed2 Add is_encoding salt util
- **ISSUE** #47694: (thuhak) utils/boto and utils/boto3 load error (refs: #47726)
- **PR** #47726: (gtmanfred) rename boto to botomod @ *2018-05-25 20:37:37 UTC*
  - b860d95b2c Merge pull request #47726 from gtmanfred/2018.3
  - 43997a466e fix tests
  - 59180e09a8 switch all salt.utils.boto* calls to __utils__ calls
  - 7e30f459c9 rename boto to botomod
- **ISSUE** #47701: (babs) elasticsearch.index_template_present state with check_definition: True fails. (refs: #47703)
- **PR** #47803: (rallytime) Back-port #47703 to 2018.3 @ *2018-05-23 20:12:30 UTC*
  - **PR** #47703: (babs) Fix elasticsearch.index_template_present with check_definition (refs: #47803)
  - 2a270162d0 Merge pull request #47803 from rallytime/bp-47703
  - 8fce0c562d Fix elasticsearch with check_definition
- **PR** #47804: (rallytime) Back-port #47761 to 2018.3 @ *2018-05-23 20:11:55 UTC*
  - **PR** #47761: (zerthimon) Fix userdata in openstack instances (refs: #47804)
  - 780043e6fd Merge pull request #47804 from rallytime/bp-47761
  - b7bc306333 Fix userdata in openstack instances
- **PR** #47776: (garethgreenaway) [2018.3] Fixes to failing _before_connect tests @ *2018-05-23 15:10:35 UTC*
  - 1efe891c42 Merge pull request #47776 from garethgreenaway/970_two_minion_unit_tests_failing

  - a5011b49ab Merge branch '2018.3' into 970_two_minion_unit_tests_failing
  - ef24f72169 Fixing unit.test_minion.MinionTestCase.test_beacons_before_connect and unit.test_minion.MinionTestCase.test_scheduler_before_connect.
- **PR** #47783: (rallytime) [2018.3] Merge forward from 2018.3.1 to 2018.3 @ *2018-05-23 14:51:56 UTC*
  - 4a6ca67883 Merge pull request #47783 from rallytime/merge-2018.3
  - 1adb8ea27f Merge branch '2018.3.1' into '2018.3'
- **PR** #47149: (meaksh) [2018.3] Strip trailing commas on Linux user's GECOS fields @ *2018-05-22 21:18:49 UTC*
  - f8a6a85d28 Merge pull request #47149 from meaksh/2018.3-remove-trailing-commas-on-linux-user-gecos-fields
  - 60ec3230db Merge branch '2018.3' into 2018.3-remove-trailing-commas-on-linux-user-gecos-fields
- **ISSUE** #46938: (racooper) Beacons documentation update for 2018.3 (refs: #47740)
- **PR** #47740: (garethgreenaway) [2018.3] Updating Beacon topic to include list based configuration for Beacons @ *2018-05-21 13:58:01 UTC*
  - 19f4137b63 Merge pull request #47740 from garethgreenaway/46938_beacons_documentation
  - 5d23ef4dd8 Updating Beacon topic to include list based configuration for Beacons
- **PR** #47710: (terminalmage) salt.loader: add error logging when whitelist lookup fails @ *2018-05-18 16:56:18 UTC*
  - 22807ac756 Merge pull request #47710 from terminalmage/fix-loader-whitelist
  - 6192391b7b Add additional missing modules to whitelist
  - 7192c38f66 Change key -> function to make log message more clear
  - 91f9fd38fd Fix loader whitelists in unit tests
  - d4f2662e5b Add error logging when whitelist lookup fails
- **PR** #47680: (rallytime) [2018.3] Merge forward from 2018.3.1 to 2018.3 @ *2018-05-17 21:30:27 UTC*
  - b91c0f5647 Merge pull request #47680 from rallytime/merge-2018.3
  - 9b1773a763 Merge branch '2018.3.1' into '2018.3'
- **PR** #47545: (dmurphy18) Fix non-root build and signing for rpm packages @ *2018-05-17 21:20:10 UTC*
  - dbf12f9be2 Merge pull request #47545 from dmurphy18/fix_nonroot_build
  - 220f887fa6 Fixed review comment, changed file.chown to file.makedirs_perms
  - b15a1652b5 Changed versionadded from 2018.3.1 to 2018.3.2, to reflect when change should be accepted
  - 90eb03e375 Additional error checking and minor cleanup
  - 40d77e03d2 Correct building rpms with non-root user
- **ISSUE** #47546: (ender8282) archive.extracted fails when archive includes files with non-ascii names with 2018.3.0 (refs: #48015, #47572)
- **PR** #47572: (garethgreenaway) [2018.3] archive module, filenames with Unicode characters @ *2018-05-17 20:41:11 UTC*
  - 69056e5b67 Merge pull request #47572 from garethgreenaway/47546_archive_non_ascii_filenames
  - 5e97b8b44a Updating with requested changes.
  - cc1aa75a2f only convert to bytes when using Python2

  - 0fe32f4066 Updating integration/modules/test_archive to include filenames with unicode characters.
  - 5a9cadd125 Accounting for when files in an archive contain non-ascii characters
- **PR** #47661: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-05-16 15:53:36 UTC*
  - cb04d9c37e Merge pull request #47661 from rallytime/merge-2018.3
  - dd3e6f3f0a Merge branch '2017.7' into '2018.3'
    - f63d801858 Merge pull request #47639 from rallytime/merge-2017.7
      - 1503f1020f Merge branch '2017.7.6' into '2017.7'
      - 6f178ca908 Merge pull request #47570 from gtmanfred/2017.7.6
      - 84aa034e03 Update dependency to msgpack
      - 98bd598701 Merge pull request #47523 from rallytime/man-pages
      - 48ecb78dec [2017.7.6] Update man pages
      - e608ea9617 Merge pull request #47517 from rallytime/bp-47505-2017.7.6
      - 0734578533 Raise proper invocation errors
    - db0eef3e8b Merge pull request #47643 from dwoz/runas_detect
      - 8c5e54b8b3 Fix typo
      - 69bc88f1c6 Remove unwanted file
    - 6dd3ab9f06 Merge pull request #47601 from dwoz/runas_detect
      - 3235ac08cb use ignore-undefined-variable
      - 58911d510a Ignore pylint WindowsError
      - 17987d3c5a Better doc string
      - a48ac26573 Skip tests when we can not use runas
    - f61ccae627 Merge pull request #47611 from Ch3LL/deb_doc
      - 6d2ddd050f One more grammar fixup
      - 2d5ff01261 Grammar fix
      - 138847c9c4 Update debian install docs to reflect new latest debian releases
    - 9e2fe284e4 Merge pull request #47598 from rallytime/bp-47568-2017.7
      - 09458c65cf Add exception logging on serialize/deserialize exceptions
      - c62c855f9c salt.serializers.yaml/yamlex: remove invalid multi_constructor
    - 8a7913ccc3 Merge pull request #47548 from Ch3LL/syslog
      - 31f13a4197 sysloghander: check for 3.5.4 python version
      - a020352a03 Catch Sysloghandler errors when log file does not exist
    - 82fb6ba366 Merge pull request #47508 from gtmanfred/2017.7
      - 1a87e7455f allow pulling the mysql_query.run_file to pull from the fileserver
    - be29b58a09 Merge pull request #47562 from rallytime/bp-47513
      - 357bc084b3 fix #46546

    - d3121fcfa4 Merge pull request #47471 from meaksh/2017.7-fix-inconsistent-scheduled-jid-with-returners
      - f079939500 Do not override jid on returners, only sending back to master
    - 662f6086db Merge pull request #47313 from Circuitsoft/2017.7
      - 237560b745 Merge branch '2017.7' into 2017.7
      - cb674fb1cb Enable opkg as pkgrepo handler on Poky
      - 7a58fd157e Enable opkg on non-NILinuxRT systems
      - 139360c55f Add Poky to OS Information Grains
      - d74057224d Un-normalize os_family in pkgrepo state
- **PR** #47672: (terminalmage) Backport #47487 to 2018.3 @ *2018-05-16 13:55:40 UTC*
  - **PR** #47487: (rosscdh) bad ref to error fix (refs: #47672)
  - d17a16b648 Merge pull request #47672 from terminalmage/bp-47487
  - 12abbfdcf7 bad ref to error fix
- **ISSUE** #47501: (mateiw) salt-ssh doesn't propagate pillar when overriding pillar in module.run + state.apply (refs: #47504)
- **PR** #47504: (mateiw) Fix pillar propagation in salt-ssh when overriding pillar in module.run + state.apply @ *2018-05-16 13:55:05 UTC*
  - 9d4f5203a7 Merge pull request #47504 from mateiw/2018.3-salt-ssh-pillar-propagation-issue-47501
  - 3ffa412ee5 Merge branch '2018.3' into 2018.3-salt-ssh-pillar-propagation-issue-47501
  - 2f1485e067 Option to merge current pillar with opts['pillar'] during pillar compile
- **PR** #47212: (weswhet) fix macOS running as user. @ *2018-05-16 13:54:23 UTC*
  - 4fe78bbb9b Merge pull request #47212 from weswhet/fix-macos-runas
  - 8ee11d74af Merge branch '2018.3' into fix-macos-runas
  - e78fa45927 adding in requested changes
  - 33c1492099 Merge branch '2018.3' into fix-macos-runas
  - 0ecabcae78 adding in a fix for running commands as a user on macos
- **ISSUE** #47587: (FedericoOlivieri) –out=table broken in 2018.3.0 'str' object but received a 'unicode' (refs: #47674)
- **PR** #47674: (terminalmage) Fix regression in table outputter due to unicode types @ *2018-05-16 13:41:04 UTC*
  - 0c7439a955 Merge pull request #47674 from terminalmage/issue47587
  - 43e3dcd398 Fix Python 3 incompatibility in table outputter
  - d729656703 Add unit tests for table outputter
  - 907d182dea Fix regression in table outputter due to unicode strings
- **PR** #47617: (aesposito91) Update napalm.py @ *2018-05-15 15:44:35 UTC*
  - d1a1a594e9 Merge pull request #47617 from aesposito91/2018.3
  - 9f5d201dd7 Update napalm.py
- **ISSUE** #46871: (guettli) Sentry Logging Handler (refs: #47649)

- **PR** #47649: (paulcollinsiii) Some additional details about Sentry for the docs @ *2018-05-15 15:41:36 UTC*
  - 049abe2850 Merge pull request #47649 from paulcollinsiii/sentry_docs
  - 2a8dfd65d7 Some additional details about Sentry for the docs
- **PR** #47642: (garethgreenaway) [2018.3] Updating templates used by salt-extend for unicode goodness @ *2018-05-15 15:38:26 UTC*
  - 8a21b864df Merge pull request #47642 from garethgreenaway/unicode_update_salt_extend_templates
  - 68d73f7afc Updating the templates that the salt-extend utility uses to include unicode_literals & print_function
    - 0232a6e1ad Add 'other' as valid kwargs for 'user.add' method
    - ff861d9089 Merge branch '2018.3' into 2018.3-remove-trailing-commas-on-linux-user-gecos-fields
- **PR** #47603: (terminalmage) Move/merge more test modules @ *2018-05-14 20:48:43 UTC*
  - 99f53c0a9f Merge pull request #47603 from terminalmage/more-test-renaming
  - d612bd27e4 Move/merge more test modules
- **PR** #47640: (rallytime) [2018.3] Merge forward from 2018.3.1 to 2018.3 @ *2018-05-14 17:59:06 UTC*
  - cc3daa88f3 Merge pull request #47640 from rallytime/merge-2018.3
  - 2827c56399 Merge branch '2018.3.1' into '2018.3'
    - f1680f1d9b Do make comparisons in a single line
    - 7dea455c12 Add unit test for new method 'user.chother'
    - 694882632c Add 'other' attribute to GECOS fields to avoid inconsistencies with chfn
    - 6c089c9de1 Fix unsupported chars checking on GECOS fields
    - d5c9ca0fbc Add unit tests for GECOS fields
    - ed940c4f58 Strip trailing commas on Linux user GECOS fields
- **PR** #47589: (Ch3LL) Remove duplicate file id in watch_any doc example @ *2018-05-10 22:06:00 UTC*
  - 4ec63dd041 Merge pull request #47589 from Ch3LL/watch_any_doc
  - 8cb97a48f5 Add additional state to watch_any doc example
  - f7223a3bbf Remove duplicate file id in watch_any doc example
- **PR** #47564: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-05-10 19:13:36 UTC*
  - 9836a7e62b Merge pull request #47564 from rallytime/merge-2018.3
  - 295e302fcf Lint: Fix syntax error from bad merge-conflict resolution
  - 25f03ae425 Merge branch '2017.7' into '2018.3'
  - 291d3d3bd8 Merge pull request #47540 from terminalmage/cmdmod-home-fix
    - e4b277f82e Fix corner case where runas user's HOME env value is incorrect
  - 49af577b3e Merge pull request #47547 from terminalmage/gitfs-tutorial
    - 806ffb298a Add masterless mode docs to gitfs tutorial
  - 476ba053d1 Merge pull request #47538 from terminalmage/pip-docs
    - 98e8ffeba4 Fix/clarify some of the pip module documentation
  - 7cc8d9d30a Merge pull request #47524 from dwoz/whitelistlines

        - 2605ff8712 Gracefully handle blank lines in whitelist.txt
    - f1a5b13072 Merge pull request #47527 from dwoz/test_file_name
        - 4aea7ca571 Update doc string
        - b3289e5a1c Fix test file name
    - e9c1723175 Merge pull request #47408 from Ch3LL/skip_diskusage
        - 068da8ad7d Merge branch '2017.7' into skip_diskusage
        - 1a7ffb4e0d return error if diskusage not available
        - 10bd63a976 Skip status.diskusage integration tests on macsox
    - 2f50ff7bc8 Merge pull request #47480 from baniobloom/ConflictingDomainExists
        - d22ed7dffa added handling for the aws error ConflictingDomainExists
    - 7f53be6e92 Merge pull request #47444 from terminalmage/render_state-spurious-error
        - a1e9fe00fd Skip trying to render a template for a nonexistant SLS file
    - 50b9c4d79d Merge pull request #47478 from terminalmage/rename-pip-state-test
        - 9f7a9ebebd Rename pip state test modules to match naming convention
    - e78ac0f9b7 Merge pull request #47505 from dwoz/win_dsc_fixes
        - d1fcb40d1c Raise proper invocation errors
    - 0d4d5047d8 Merge pull request #47499 from dwoz/win_run_timeout_again
        - c6697b9f16 Move kill process tree and re-use it
    - 6a4d0380b1 Merge pull request #47493 from dwoz/win_run_timeout
    - fec1233dc4 Add support for windows timeout to run_salt
- **PR** #47565: (rallytime) Back-port #47440 to 2018.3 @ *2018-05-10 18:15:50 UTC*
    - **PR** #47440: (kt97679) fix for the race condition, details are here: https://github.com/salt… (refs: #47565)
    - b22657ff9e Merge pull request #47565 from rallytime/bp-47440
    - 7c43417d46 addressed feedback
    - 1abe05207c fix for the race condition, details are here: https://github.com/saltstack/salt/issues/33223#issuecomment-386117236
- **PR** #47590: (Ch3LL) Fix firewalld prune_services deprecation warning @ *2018-05-10 17:01:48 UTC*
    - 17e0fdca6c Merge pull request #47590 from Ch3LL/firewall_warn
    - d29b8e0ae2 Fix firewalld prune_services deprecation warning
- **PR** #47568: (terminalmage) salt.serializers.yaml/yamlex: remove invalid multi_constructor (refs: #47598) @ *2018-05-10 16:21:09 UTC*
    - 2fcb108d71 Merge pull request #47568 from terminalmage/salt-jenkins-971
- **ISSUE** #38671: (MikeSpaceG) zpool.get error (refs: #47224)
- **PR** #47563: (rallytime) Back-port #47224 to 2018.3 @ *2018-05-10 16:18:07 UTC*
    - **PR** #47224: (sjorge) Zpool zol fixes (refs: #47563)
    - e3ee7053fb Merge pull request #47563 from rallytime/bp-47224
    - 1d191445a7 FIX #38671 - zpool.get should support older zfs version

  - 23705b12cb zpool.scub docs should mention pause is not always available
  - d73885aa5c Make sure the docs mention that createboot is only available on illumos
- **PR** #47569: (Ch3LL) Update salt.utils.path mock in virtual core test @ *2018-05-10 15:18:31 UTC*
  - 8e26624db3 Merge pull request #47569 from Ch3LL/core_virt
- **PR** #47576: (rallytime) [2018.3] Small Documentation Fixes @ *2018-05-10 14:08:11 UTC*
  - fc93715eea Merge pull request #47576 from rallytime/doc-updates
  - ca927fc14b Add extra lines that are needed for proper code-block formatting
  - b330d763a4 Remove infoblox state autodoc file

28.2.15 Salt 2018.3.4 Release Notes

Version 2018.3.4 is a bugfix release for *2018.3.0*.

State Changes

- The `host.present` state can now remove the specified hostname from IPs not specified in the state. This can be done by setting the newly-added `clean` argument to `True`.

Statistics

- Total Merges: **531**
- Total Issue References: **111**
- Total PR References: **412**
- Contributors: **76** (5uper5hoot, Ch3LL, ClaudiuPID, Giandom, KaiSforza, MTecknology, ManicoW, OrangeDog, ShantonRU, The-Loeki, Yxnt, aarnaud, amendlik, angeloudy, bartlaarhoven, bbh-kmd, bbinet, bdrung, bergmannf, bluesliverx, bornwitbugs, brejoc, cachedout, casselt, cro, cstarke, dgmorrisjr, dmurphy18, dubb-b, dwoz, frogunder, garethgreenaway, gtmanfred, isbm, jacobweinstock, jgleissner, jodok, jpsv, jyurdal, kiemlicz, kunal-bajpai, lexvella, lomeroe, m03, madrisan, mat813, mattp-, max-arnold, mchugh19, meaksh, michaelgibson, nhavens, pirogoeth, rallytime, rkrieger, rmarcinik, rongzeng54, rwaweber, s0undt3ch, sathieu, sheagcraig, silenius, sunyq, t0fik, terminalmage, terrible-broom, thebluesnevrdie, thetaurean, tlemarchand, tonybaloney, twangboy, waynew, weswhet, whytewolf, yosnoop, zwo-bot)

Changelog for v2018.3.3..v2018.3.4

Generated at: 2019-02-06 20:56:17 UTC

- **PR** #51525: (twangboy) Backport 51095 to 2018.3.4 @ *2019-02-06 20:20:37 UTC*
  - a37fb7d Merge pull request #51525 from twangboy/backport_51095_2018.3.4
  - b6acc92 Backport 51095
- **PR** #51520: (frogunder) Update man pages for 2018.3.4 @ *2019-02-06 18:42:13 UTC*
  - 602ddc8 Merge pull request #51520 from frogunder/man_pages
  - 0cbe3f8 Update man pages for 2018.3.4
- **ISSUE** #51428: (yosnoop) salt.utils.dns.lookup fails with dig method on EL7 (refs: #51429)

- **PR** #51429: (yosnoop) The +nocl is supported much wider than +noclass @ *2019-02-06 17:48:48 UTC*
  - 1f981c6 Merge pull request #51429 from yosnoop/replace-noclass-with-nocl-for-dig
  - 5231e73 Merge branch '2018.3.4' into replace-noclass-with-nocl-for-dig
  - 14e6fb7 In case the test machine can't resolve
  - 95e8a2f The +nocl is supported much wider than +noclass
  - a00d708 Test dig options for salt.utils.dns.lookup
- **ISSUE** #49226: (defanator) salt-cloud grains inheritance from provider to profile (2018 vs 2016) (refs: #51366)
- **PR** #51441: (Ch3LL) Backport #51366 into 2018.3.4 @ *2019-02-06 17:42:19 UTC*
  - **PR** #51366: (dwoz) Cloud provider and profile configs are merged [2018.3] (refs: #51441)
  - 209ef32 Merge pull request #51441 from Ch3LL/bp-51366
  - 5bc466e Fix python 3 path for shell test cases
  - 5351e5c Add testing path to run_script commands
  - 2d8a364 Fix linter issues
  - 9b8718a Cloud provider and profile configs are merged
- **PR** #51458: (twangboy) Add Py2 Warning for Py2 Installer @ *2019-02-06 17:33:02 UTC*
  - 90a8682 Merge pull request #51458 from twangboy/add_py2_warning
  - 05522c0 Merge branch '2018.3.4' into add_py2_warning
- **PR** #51460: (twangboy) Update installer mac 2018.3.4 @ *2019-02-06 17:31:18 UTC*
  - b492021 Merge pull request #51460 from twangboy/update_installer_mac_2018.3.4
  - 3c92688 Merge branch '2018.3.4' into update_installer_mac_2018.3.4
- **PR** #51471: (twangboy) Fix the failing dns test on Windows @ *2019-02-03 22:03:34 UTC*
  - 892d98c Merge pull request #51471 from twangboy/fix_test_win_dns_client
  - c9f6b01 Merge branch '2018.3.4' into fix_test_win_dns_client
- **PR** #51462: (ShantonRU) Fix insecure SQL queries in MySQL module @ *2019-02-01 21:19:20 UTC*
  - a05305d Merge pull request #51462 from ShantonRU/fix-insecure-sql-queries-in-mysql-module-1
  - d3ef2cb Merge branch '2018.3.4' into fix-insecure-sql-queries-in-mysql-module-1
  - f47dda6 Fix improper use of socket authentication in mysql.user_chpass
  - ed62a2f Fix insecure SQL queries in mysql.user_chpass
    - 33922c9 Fix the failing dns test on Windows
- **PR** #51445: (terminalmage) Remove custom grains module from 2018.3 branch @ *2019-02-01 16:48:13 UTC*
  - 48936c2 Merge pull request #51445 from terminalmage/fix-custom-grains-traceback
  - 191235d Remove custom grains module from 2018.3 branch
  - 72fd1a2 Roll back icon change
  - 7dd1704 Add welcome_py2.rtf
  - c93e9e3 Add Py2 Warning to installer package
  - b658900 Add Py2 Warning for Py2 Installer

- **PR** #51438: (twangboy) Add missing raise statement @ *2019-01-30 20:04:05 UTC*
  - a880e42 Merge pull request #51438 from twangboy/fix_raise_2018.3.4
  - 5830fad Add missing raise statement
- **PR** #51432: (Ch3LL) Backport #51324 into 2018.3.4 @ *2019-01-30 17:51:50 UTC*
  - **PR** #51324: (garethgreenaway) [2018.3] Fixes to x509 module and test (refs: #51432)
  - 28b898c Merge pull request #51432 from Ch3LL/bp-51324
  - bd41ff0 Fixing various bits in x509 module and unit test to ensure tests are passing on python2 & python3
- **PR** #51377: (twangboy) Fix issue when directory doesn't exist in LGPO module (2018.3.4) @ *2019-01-29 17:33:25 UTC*
  - 01717d2 Merge pull request #51377 from twangboy/fix_auditpol_2018.3.4
  - 19e5181 Fix issue when directory doesn't exist
- **PR** #51338: (Ch3LL) Backport #51207 into 2018.3.4 @ *2019-01-27 23:44:59 UTC*
  - **PR** #51207: (twangboy) Backport 50771 to 2018.3 (refs: #51338)
  - **PR** #50771: (terminalmage) Fix failing opensuse tests (refs: #51207)
  - 43a9d2f Merge pull request #51338 from Ch3LL/bp-51207
  - 345ae67 Backport PR# 50771 to 2018.3 branch
- **ISSUE** #51158: (AndreiPashkin) saltmod.state doesn't respect "roster" argument (refs: #51170)
- **PR** #51337: (Ch3LL) Backport #51170 into 2018.3.4 @ *2019-01-27 23:44:24 UTC*
  - **PR** #51170: (dwoz) Honor roster system for ssh orchestrations (refs: #51337)
  - 3dcd11d Merge pull request #51337 from Ch3LL/bp-51170
  - 4e85505 Merge branch '2018.3.4' into bp-51170
- **ISSUE** #50221: (tkaehn) file.managed: contents_pillar and pillar.file_tree with keep_newline is broken in 2018.3.3 (refs: #51252)
- **PR** #51340: (Ch3LL) Backport #51252 into 2018.3.4 @ *2019-01-27 23:42:49 UTC*
  - **PR** #51252: (dwoz) Peserve newlines from pillar data for file.managed [2018.3] (refs: #51340)
  - 7656aa5 Merge pull request #51340 from Ch3LL/bp-51252
  - e74a6c6 Fix py3 managed.file test failure
  - 6f2c7cf Remoe un-needed log statement
  - 824950c Fix 50221 regression test on Windows
  - 1840af9 peserve newlines from pillar data for file.managed
- **PR** #51290: (s0undt3ch) [2018.3.4] Fix minion start @ *2019-01-25 15:14:47 UTC*
  - e90ca1d Merge pull request #51290 from s0undt3ch/hotfix/fix-network
  - da3aace Merge branch '2018.3.4' into hotfix/fix-network
  - 04f16b8 Fix tests. We no longer get *ipaddress* instances back.
  - c3fa82a Fix minion start issue
    - c893dc4 Fix saltmod roster test
    - b61854f Roster defines the roster system not a file

        - 484bab4 Fix linter
        - 5e2db2f Honor roster file for ssh orchestrations
- **PR** #51317: (terminalmage) Fix regression in dynamic pillarenv @ *2019-01-25 09:29:44 UTC*
    - c8db7e3 Merge pull request #51317 from terminalmage/fix-pr-50417
    - 4be2891 Merge branch '2018.3.4' into fix-pr-50417
- **PR** #51311: (twangboy) Fix failing test *unit.states.test_file.TestFileState.test_managed* on Windows @ *2019-01-24 21:09:35 UTC*
    - e246fdb Merge pull request #51311 from twangboy/fix_test_file.test_managed
    - f4177e0 Remove mode parameter on Windows
    - 4c4ded3 Fix regression in dynamic pillarenv
- **ISSUE** #51195: (littleski) fstab improperly escaped when spaces are used in options (refs: #51226)
- **ISSUE** #27160: (martinadolfi) salt.states.mount persistence error using spaces in route (refs: #39593)
- **PR** #51292: (Ch3LL) Backport #51226 into 2018.3.4 @ *2019-01-24 11:11:34 UTC*
    - **PR** #51226: (garethgreenaway) [2018.3] Handle spaces in fstab opts, similar fix to #39593 (refs: #51292)
    - **PR** #39593: (zwo-bot) Fix issue #27160 - escape whitespaces in fstab entry in module mount when persist=True (refs: #51226)
    - 40b20126 Merge pull request #51292 from Ch3LL/bp-51226
    - aba114e Merge branch '2018.3.4' into bp-51226
- **PR** #51295: (Ch3LL) Backport #51279 into 2018.3.4 @ *2019-01-24 11:10:19 UTC*
    - **PR** #51279: (twangboy) Backport __context__ support from develop (refs: #51295)
    - fc0bec5 Merge pull request #51295 from Ch3LL/bp-51279
    - 5c4a7a4 Merge branch '2018.3.4' into bp-51279
- **PR** #51293: (Ch3LL) Backport #51233 into 2018.3.4 @ *2019-01-24 09:17:49 UTC*
    - **PR** #51233: (Ch3LL) Backport #51105 to 2018.3 (refs: #51293)
    - **PR** #51105: (garethgreenaway) [develop] Report permission changes in test mode (refs: #51233)
    - dacd1ea Merge pull request #51293 from Ch3LL/bp-51233-2
    - a155fd7 Accounting for when permissions are not returned from file.check_perms, eg. windows systems.
    - 53ba068 When test=True and there are permissions changes, report it.
    - d2eba77 Remove network.py patch
    - d431c71 Add __context__ support
        - de6c13d Handle spaces in fstab opts, similar fix to #39593
- **ISSUE** #51248: (johnnybubonic) "[WARNING] The file_roots parameter is not properly formatted, using defaults" (refs: #51278)
- **PR** #51278: (dwoz) Warn about proper validation option @ *2019-01-22 22:13:13 UTC*
    - 1abffb2 Merge pull request #51278 from dwoz/issue51248
    - c614198 Use pillar roots validation method
    - 0d2b0c6 Fix linter warnings

  - 5b98895 Fix wart in pillar_roots validation
- **PR** #51230: (Ch3LL) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2019-01-18 19:50:10 UTC*
  - e53e9b1 Merge pull request #51230 from Ch3LL/merge-2018.3
  - 0ef7675 Merge branch '2017.7' into '2018.3'
  - 4a12c5f Merge pull request #50852 from mirceaulinic/fix-capirca
    - fa91267 Merge branch '2017.7' into fix-capirca
    - 71913ff Merge branch '2017.7' into fix-capirca
    - 6fc3484 Lint?
    - 3d56750 Extra newline for lint
    - 5176cd8 Improve the Capirca and related modules for ACL config generation
  - 3f4759b Merge pull request #50923 from astronouth7303/patch-2
    - 4f631a6 Document {% include %}
  - 210e817 Merge pull request #51111 from s0undt3ch/2017.7
    - 08e0ae7 Update to salt-bootstrap v2019.01.08
- **ISSUE** #51211: (sathieu) file_roots envs with slash leads to "No such file or directory: '/var/cache/salt/master/file_lists/roots/.foo/bar.w' (refs: #51212)
- **ISSUE** #48132: (BenjaminSchiborr) Dynamic File Roots (refs: #51212)
- **PR** #51212: (sathieu) Allow slashes in file_roots envs. @ *2019-01-17 17:29:18 UTC*
  - 26008ef Merge pull request #51212 from sathieu/file_roots_slash
  - b769f18 Allow slash in file_roots envs
  - f9437e2 Test for file_roots envs with slash leads to "No such file or directory"
- **PR** #51209: (garethgreenaway) [2018.3] Back-port PR #51124 to 2018.3 @ *2019-01-17 13:28:32 UTC*
  - **PR** #51124: (Ch3LL) Handle encoding when opening file for both py2/py3 (refs: #51209)
  - 5916282 Merge pull request #51209 from garethgreenaway/bp-51124
  - 82acce4 Handle encoding when opening file for both py2/py3
- **PR** #51184: (twangboy) Fix win_lgpo_auditpol salt util on Windows @ *2019-01-16 18:08:01 UTC*
  - 094bd33 Merge pull request #51184 from twangboy/fix_auditpol
  - 2e68d35 directly call cmd.run_all
- **PR** #51191: (Ch3LL) Backport #50918 into 2018.3 @ *2019-01-16 18:02:19 UTC*
  - **PR** #50918: (twangboy) Fix *test_debian_ip* on Windows (refs: #51191)
  - ac71922 Merge pull request #51191 from Ch3LL/bp-50918
  - 7300d84 Skip debian tests on Windows
  - 4622433 Fix *test_debian_ip* on Windows
- **ISSUE** #49883: (kiemlicz) salt-cloud azurearm is not working (refs: #51144)
- **PR** #51144: (jgleissner) 2018.3 azurearm py3 @ *2019-01-15 16:46:35 UTC*
  - e7f53a3 Merge pull request #51144 from jgleissner/2018.3-azurearm-py3

  - 3178f1a msazure.py: remove unused imports
  - 99a4b9d azurearm: fix list_locations
  - aebe34b azurearm: fix show_instance function
  - d80669f msazure: fix object_to_dict for python 3
- **PR** #51171: (twangboy) Add random sleep time to avoid overloading Windows @ *2019-01-15 16:45:33 UTC*
  - 1206f92 Merge pull request #51171 from twangboy/fix_crypt
  - 6caa93b Add random sleep time to avoid overloading Windows
- **PR** #51174: (jpsv) Update OpsGenie to use their V2 API (2018.3) @ *2019-01-15 16:44:54 UTC*
  - a96deed Merge pull request #51174 from jpsv/2018.3
  - 055823e Update opsgenie.py
  - 501c8df Merge branch '2018.3' into 2018.3
  - 220aa8a Update opsgenie.py
  - 5dc21cc Update to from V1 API to V2
  - f93ce7c Update to V2 API
- **ISSUE** #42280: (michaelgibson) make ldap eauth 2 factor compatible (refs: #42426, #45347)
- **PR** #51010: (amendlik) Allow unauthenticated bind for listing LDAP groups @ *2019-01-15 16:43:42 UTC*
  - **PR** #45811: (gtmanfred) Fix ldap eauth (refs: #51010)
  - **PR** #45347: (amendlik) LDAP group membership (refs: #51010)
  - **PR** #42426: (michaelgibson) adding 2-factor auth capability to ldap eauth module - #42280 (refs: #45347)
  - 563d487 Merge pull request #51010 from amendlik/ldap-groups
  - cd336ed Merge branch '2018.3' into ldap-groups
  - 94f8ee4 Update unit tests for auth.ldap
  - 08e36d0 Allow unauthenticated bind for listing LDAP groups
- **PR** #51112: (s0undt3ch) [2018.3] Update to salt-bootstrap v2019.01.08 @ *2019-01-15 16:42:09 UTC*
  - 6dc3752 Merge pull request #51112 from s0undt3ch/2018.3
  - 4f1d846 Update to salt-bootstrap v2019.01.08
- **PR** #51097: (terminalmage) Fix improper exception raises (2018.3) @ *2019-01-15 16:38:25 UTC*
  - b19c990 Merge pull request #51097 from terminalmage/fix-keyerror-raise
  - a3c0b49 Fix poorly-written test
  - c0ff7b1 Fix more incorrect exception raises
  - f11dbcc Fix improper KeyError raise
- **PR** #51169: (twangboy) More descriptive error when missing GitPython or PyGit2 @ *2019-01-15 16:37:44 UTC*
  - 8920a86 Merge pull request #51169 from twangboy/fix_git_state
  - f2d1180 More descriptive error when missing GitPython or PyGit2
- **ISSUE** #51160: (amendlik) Minion crashes with long hostname under Python 3 (refs: #51173, #51161)
- **PR** #51173: (dwoz) Avoid exceptions by passing bytes to idna @ *2019-01-15 16:32:31 UTC*

  - **PR** #51161: (amendlik) Ignore exceptions when generating the minion ID (refs: #51173)
  - 58eede6 Merge pull request #51173 from dwoz/issue51160
  - 11cc7b3 Remove un-needed test
  - 0f87812 Merge develop, Revert exception handling
  - 7365a1a Pass bytes to idna
- **PR** #51147: (Ch3LL) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2019-01-15 02:05:51 UTC*
  - 06854cf Merge pull request #51147 from Ch3LL/merge-2018.3
  - d39d8b7 Merge branch '2018.3' into merge-2018.3
- **ISSUE** #51160: (amendlik) Minion crashes with long hostname under Python 3 (refs: #51173, #51161)
- **PR** #51161: (amendlik) Ignore exceptions when generating the minion ID (refs: #51173) @ *2019-01-14 22:47:54 UTC*
  - b79c09e Merge pull request #51161 from amendlik/long-hostname
  - 6fcc4ad Merge branch '2018.3' into long-hostname
- **ISSUE** #51067: (rkrieger) KeyError u'manufacturer' causes salt-minion crash on start with OpenBSD arm64 system (refs: #51159)
- **ISSUE** #3: (thatch45) libvirt module (refs: #51159)
- **PR** #51159: (rkrieger) Prevent crash by testing for 'manufacturer' in osdata on OpenBSD @ *2019-01-14 18:08:22 UTC*
  - 1424734 Merge pull request #51159 from rkrieger/fix-issue-51067
  - 1440ccf Prevent crash by testing for 'manufacturer' in osdata on OpenBSD
  - 1a4e307 Ignore exceptions when generating the minion ID
- **PR** #51146: (terminalmage) Fix invalid assignment in vsphere module @ *2019-01-12 21:44:33 UTC*
  - 5e37326 Merge pull request #51146 from terminalmage/fix-vsphere-tuple-issue
  - 5d60022 Fix invalid assignment in vsphere module
- **PR** #51154: (twangboy) Add support for Advanced Audit policies @ *2019-01-12 21:43:27 UTC*
  - fad8621 Merge pull request #51154 from twangboy/auditpol
  - 9064376 Add support for Advanced Audit policies
- **PR** #51151: (Ch3LL) Backport #51061 to 2018.3 @ *2019-01-12 03:18:44 UTC*
  - **PR** #51061: (garethgreenaway) [fluorine] fix to salt.utils.vt.Terminal (refs: #51151)
  - f7a100c Merge pull request #51151 from Ch3LL/bp_51061
  - ffdae27 When writing output to stdout we want to ensure that the data is a string not bytes. Under py2 the salt.utils.data.encode function results in a string but under py3 the result is a bytestring. Swapping out salt.utils.data.encode for salt.utils.stringutils.to_str.
- **PR** #51150: (Ch3LL) Back-port #49508 to 2018.3 @ *2019-01-12 03:18:19 UTC*
  - **PR** #49508: (bergmannf) Fix X509 remote signing on python3 (refs: #51150)
  - 3a50a57 Merge pull request #51150 from Ch3LL/bp-49508
  - 7ae53ad Do not silently ignore errors.
  - deb0134 Convert to string before sending via 'publish'.

  - 651c551 Use to_str salt.utils when writing to a file.
- **ISSUE** #50433: (zerthimon) Wildcard grant fails in mysql_grants.present in mysql 8.x (refs: #51152)
- **PR** #51152: (garethgreenaway) [2018.3] Fixes to mysql module to handle ALL PRIVILEGES grant @ *2019-01-12 03:17:15 UTC*
  - 8bb589d Merge pull request #51152 from garethgreenaway/50433_handle_grants_better
  - f7caa4d Fixing lint
  - e8c8c0f Adding some tests to ensure "ALL PRIVILEGES" is handled correctly in 8.0 and 5.6
  - b4bfd9f Add additional grants. Adding logic to handle when ALL or ALL PRIVILEGES is passed for the grant, including some logic to handle the fact that the grants are split when show grants is run for a particular user.
    - 74edfd9 Fix pylint
    - 81a3d47 Merge branch '2017.7' into '2018.3'
    - 8a76a19 Merge pull request #50858 from jasonarewhy/bugfix-2017.7/ubuntu-networking-searchdomain
      - 3395a3d Merge branch '2017.7' into bugfix-2017.7/ubuntu-networking-searchdomain
      - 69d25aa update debian_ip test
      - a6b896c get searchdomain in Ubuntu 12+
- **ISSUE** #50849: (5uper5hoot) MySQL version needs to be decoded (refs: #51133)
- **PR** #51133: (5uper5hoot) Fixes mysql server version comparison. @ *2019-01-10 16:45:38 UTC*
  - dda1ebb Merge pull request #51133 from 5uper5hoot/issue-#50849
  - ec14e13 Fixes mysql server version comparison.
- **PR** #50878: (max-arnold) Fix profile outputter docs @ *2019-01-08 18:32:40 UTC*
  - 9d4321d Merge pull request #50878 from max-arnold/output-docs
  - d7403da Fix profile outputter docs
- **PR** #51076: (OrangeDog) Typos in index_template_present documentation @ *2019-01-08 18:25:20 UTC*
  - ce87504 Merge pull request #51076 from OrangeDog/patch-1
  - 568e4fb Revert add -> check
  - 4d7ebe7 Typos in index_template_present documentation
- **PR** #51081: (garethgreenaway) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2019-01-07 20:34:16 UTC*
  - d08abf4 Merge pull request #51081 from garethgreenaway/merge-2018.3
  - 027804a Merge branch '2018.3' into merge-2018.3
- **ISSUE** #50966: (klafkoff) 2018.3.3 missing a patch to salt/modules/nxos.py (refs: #51080)
- **PR** #51080: (garethgreenaway) [2018.3] Fix to nxos module @ *2019-01-07 18:07:25 UTC*
  - 335088b Merge pull request #51080 from garethgreenaway/50966_nxos_ensure_kwargs_list
  - 230a66b Merge branch '2018.3' into 50966_nxos_ensure_kwargs_list
  - 629dc25 Ensure kwargs is a list before we attempt to loop through. Similar fix to https://github.com/rallytime/salt/commit/430c462f34eb4eedf2384e06fbee8dc19b8026f6 in the proxy module.
    - a20791b Merge branch '2018.3' into merge-2018.3

- **PR** #51079: (terminalmage) Fix shadow.set_date/shadow.set_expire on F28 @ *2019-01-07 16:28:36 UTC*
    - c47b646 Merge pull request #51079 from terminalmage/salt-jenkins-1238
    - ccd5c15 Fix shadow.set_date/shadow.set_expire on F28
- **ISSUE** #47043: (xBytez) lsattr slowing down archive.extracted (refs: #51044)
- **PR** #51044: (terminalmage) archive.extracted: Don't run lsattr when enforcing user/group ownership @ *2019-01-03 19:26:03 UTC*
    - aeeb047 Merge pull request #51044 from terminalmage/issue47043
    - 787b79a archive.extracted: Don't run lsattr when enforcing user/group ownership
- **ISSUE** #50964: (gtmanfred) Add a long description to setup.py for the new pypi.org (refs: #51031)
- **PR** #51034: (garethgreenaway) Back-port PR #51031 to 2018.3 @ *2019-01-03 15:42:22 UTC*
    - **PR** #51031: (waynew) Set long_description (refs: #51034)
    - 04fdae5 Merge pull request #51034 from garethgreenaway/bp-51031
    - cd310c2 Set long_description
- **ISSUE** #50634: (marek-obuchowicz) Debian pkg.managed gpg/tty issue (refs: #50972)
- **PR** #50972: (pirogoeth) Use gpg's –batch in apt-key adv during repo key add @ *2019-01-03 02:34:13 UTC*
    - 4685e0e Merge pull request #50972 from pirogoeth/sjohn/2018.3-apt-key-batch
    - cd3d755 Merge branch '2018.3' into sjohn/2018.3-apt-key-batch
- **PR** #51028: (whytewolf) AIX lsattr fix for file module. @ *2019-01-03 01:52:33 UTC*
    - 44bc862 Merge pull request #51028 from whytewolf/aix_lsattr_fix
    - 1b733c6 Set lsattr functions to return None if *salt.utils.platform.is_aix* returns true.
    - 7410008 Merge branch '2018.3' into sjohn/2018.3-apt-key-batch
- **PR** #51018: (garethgreenaway) [fluorine] Merge forward from 2017.7 to 2018.3 @ *2019-01-01 23:11:02 UTC*
    - b4774a8 Merge pull request #51018 from garethgreenaway/merge-2018.3
        - 773bf31 Use gpg's –batch in apt-key adv during repo key add
            - 9386aab Merge branch '2017.7' into merge-2018.3
            - 998feca Merge pull request #51030 from terminalmage/issue38502
            - 6dbd81f Add unit test to ensure we don't diff bare repos
    - 12f7583 Merge branch '2017.7' into merge-2018.3
    - 4529150 Merge pull request #50977 from rallytime/config-docs
        - 43c39e7 Add documentation to master config docs for job_cache_store_endtime
        - 887cd07 Add some salt-api options to the master config docs
        - 0d49746 Fix some formatting in master config docs
    - 409594b Merge pull request #50991 from twangboy/requests_2017.7
        - 32b9d2a Update requests for osx
        - 413dc72 Merge branch '2017.7' of https://github.com/saltstack/salt into requests_2017.7
    - e053074 Merge pull request #50983 from garethgreenaway/bp-50885

        - 09a8e6e fix a del in cloud vmware that crash when the key isn't there
        - 4196690 Update requests to 2.21.0
        - 23f78b0 Upgrade requests to 2.20.1
- **PR** #50990: (twangboy) Requests 2018.3 @ *2018-12-31 20:19:01 UTC*
    - c53723a Merge pull request #50990 from twangboy/requests_2018.3
    - e8b9233 Update requests on osx
    - b22856f Update requests to 2.21.0
    - 242389b Merge branch '2018.3' of https://github.com/saltstack/salt into requests_2018.3
    - 0d4b799 Upgrade requests to 2.20.1
- **PR** #51004: (twangboy) Add tests for the win_wusa state and module @ *2018-12-31 20:03:30 UTC*
    - cecd108 Merge pull request #51004 from twangboy/win_wusa
    - 0e92528 Merge branch '2018.3' into win_wusa
    - c0d2495 Fix lint, add test docs
    - 1750270 Add state tests for win_wusa
    - 6285b64 Update win_wusa and add tests
- **PR** #51014: (madrisan) ssh module: missing space in error message @ *2018-12-31 19:59:19 UTC*
    - 3e0515b Merge pull request #51014 from madrisan/patch-1
    - ab980fb ssh module: missing space in error message
- **PR** #50883: (terminalmage) Backport #49622 to 2018.3 branch @ *2018-12-31 19:58:11 UTC*
    - **PR** #49622: (angeloudy) fix command line options (refs: #50883)
    - 582d6b4 Merge pull request #50883 from terminalmage/bp-49622
    - c93dbfa Update archive.py
    - 925a9c9 Update archive.py
    - ed2bed3 fix command line options
- **PR** #50945: (garethgreenaway) [2018.3] Merge forward from 2017.7 to 2018.8 @ *2018-12-30 00:05:47 UTC*
    - fd9711b Merge pull request #50945 from garethgreenaway/merge-2017.7
    - 43b3fa4 Merge branch '2018.3' into merge-2017.7
- **ISSUE** #50854: (Ch3LL) [fluorine] windows stops when cannot connect to master (refs: #50855)
- **PR** #51006: (dwoz) Let minions try to connect to master indefinitly @ *2018-12-28 21:10:39 UTC*
    - **PR** #50855: (dwoz) Let minions try to connect to master indefinitly (refs: #51006)
    - b8615e1 Merge pull request #51006 from dwoz/backport_50855
    - 5563631 Let minions try to connect to master indefinitly
- **ISSUE** #50989: (mpx8701) SLS error (An exception occurred in this state: Traceback) (refs: #51005)
- **PR** #51005: (garethgreenaway) [2018.3] Fixes to junos state module @ *2018-12-28 20:51:31 UTC*
    - cfed697 Merge pull request #51005 from garethgreenaway/50989_fixing_junos_state_arguments

  - d4ac180 Fixing the arguments accepted by cli function in the junos state module which are passed along to the cli function in junos module, so they align.
  - 3cfca8e Merge branch '2018.3' into merge-2017.7
- **PR** #50982: (twangboy) Fix issue with LGPO when returning empty dict (2018.3) @ *2018-12-26 19:58:12 UTC*
  - daa591b Merge pull request #50982 from twangboy/fix_lgpo_2018.3
  - d942934 Get rid of requests import (CaseInsensitiveDict)
    - 6170fa5 Merge branch '2018.3' into merge-2017.7
- **ISSUE** #46886: (bbh-kmd) Unable to get salt-minion to call salt-call with alternate configuration (refs: #50959)
- **PR** #50984: (garethgreenaway) Back-port PR #50959 to 2018.3 @ *2018-12-26 19:29:54 UTC*
  - **PR** #50959: (bbh-kmd) #46886: Use correct path to config_dir when running salt from alternate placement. (refs: #50984)
  - 110bc66 Merge pull request #50984 from garethgreenaway/bp-50959
  - 20fa5c9 Update sudo.py
- **PR** #50958: (bdrung) Fix twilio version checking @ *2018-12-25 10:46:15 UTC*
  - 8fc89a0 Merge pull request #50958 from bdrung/fix-twilio
  - df8d7b9 Fix twilio version checking
    - d89c9b5 Merge branch '2018.3' into merge-2017.7
- **ISSUE** #50947: (cachedout) Log full of IPv6 warnings (refs: #50962)
- **PR** #50962: (terminalmage) Remove redundant overrides for ip_address/ip_interface @ *2018-12-21 22:04:34 UTC*
  - bb87822 Merge pull request #50962 from terminalmage/ipv6-warnings
  - 773b983 Merge branch '2018.3' into ipv6-warnings
  - f3b1254 Remove redundant overrides for ip_address/ip_interface
    - cea01e7 Updating valigdate_tgt to ensure tests pass.
    - cdc7825 yet another lint fix.
    - f44a87e One last lint fix.
    - 45f363b Fixing various lint errors.
    - 6a65bf9 Merge branch '2017.7' into merge-2017.7
    - 747dd69 Merge pull request #50876 from terminalmage/issue50153
      - 676a12a Fix compound matching in eauth
    - 07fca64 Merge pull request #50909 from damon-atkins/jenkins_pylint
      - da1fc32 change params.CHANGE_BRANCH to env.CHANGE_BRANCH
    - e7f0ab6 Merge pull request #50873 from Ch3LL/bp-50740
      - 5fd9ed9 Merge branch '2017.7' into bp-50740
    - 2f77b2d Merge pull request #50869 from garethgreenaway/50848_mysql_exception_test
      - 3767d4f Merge branch '2017.7' into 50848_mysql_exception_test
      - af3e145 Cleanup.

      - b85c5bf Adding a test to ensure exception handling is correct.
      - 4242e64 Merge branch '2017.7' into bp-50740
    - a5caab7 Merge pull request #50872 from Ch3LL/bp-50864
      - fc9c64b fix typo in file.managed documentation
      - fd73f86 Catch UnicodeDecodeErrors in id verification
- **PR** #50937: (Ch3LL) Fix external_pillars documentation @ *2018-12-20 19:15:20 UTC*
  - 49f2a35 Merge pull request #50937 from Ch3LL/pillar_doc
  - 1f01489 Fix external_pillars documentation
- **ISSUE** #50919: (dpkirchner) state.apply test documentation unclear (refs: #50940)
- **PR** #50940: (terminalmage) Fix confusing state module documentation @ *2018-12-20 18:29:26 UTC*
  - 4213527 Merge pull request #50940 from terminalmage/issue50919
  - 3318927 Fix confusing state module documentation
- **PR** #50929: (twangboy) Fix *unit.utils.test_lgpo_netsh* on Windows for Py3 (2018.3) @ *2018-12-20 17:50:37 UTC*
  - 421e4f4 Merge pull request #50929 from twangboy/fix_test_lgpo_netsh_2018.3
  - 01f9beb Fix failing lgpo netsh tests
- **PR** #50917: (dwoz) Cherry pick test fixes from fluorine @ *2018-12-19 17:56:19 UTC*
  - b08b7e1 Merge pull request #50917 from dwoz/test_fixes
  - 16d10fb Fix linter warnings
  - b22f827 Increase and standardize ShellCase timeouts
  - a5c8c18 Do not pass unicode in environment
- **PR** #50913: (bdrung) Fix documentation build @ *2018-12-19 16:03:55 UTC*
  - 59df6b4 Merge pull request #50913 from bdrung/fix-doc-build
  - ecdb452 doc: Indroduce MOCK_MODULES_MAPPING
  - cf64824 doc: Move mock_decorator_with_params up
  - 08ef669 doc: Address pylint issues in conf.py
  - 67a3e37 doc: Mark preamble variable as raw string
  - b1e6080 doc: Fix reference to salt executors
  - e48b324 doc: Replace xa0 (non-breaking space) by spaces
  - b973e7a doc: Fix iterating over the Mock object in Python 3
  - bb29948 doc: Do not mock non-existing __mro_entries__ attribute
  - e6cf3c4 Do not mock json when building the documentation
  - 4acaf45 Fix sphynx error about tornado.version_info
- **ISSUE** #50892: (cruscio) archive.extracted source_hash is case sensitive (refs: #50906)
- **PR** #50906: (garethgreenaway) [2018.3] Ensure source_hash returned from get_source_sum is lowercase. @ *2018-12-19 04:22:23 UTC*
  - f4e6b88 Merge pull request #50906 from garethgreenaway/50892_archive_extracted_source_hash_case_sensitive

  - bcf8f06 fixing lint
  - 543ec7b Ensure source_hash returned from get_source_sum is lowercase.
- **PR** #50867: (Ch3LL) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-12-14 21:32:23 UTC*
  - b95b44f Merge pull request #50867 from Ch3LL/merge-2018.3
  - 512293f Remove dunder_dictionaries.rst. Was added to modules/developing.rst
  - 89b4e09 Merge branch '2017.7' into '2018.3'
  - 858cfac Merge pull request #50863 from 5uper5hoot/fix-issue-50848
    - c90d201 Unpack exception args in python3 compatible way.
  - e28b2ba Merge pull request #50752 from kstreee/fix-memory-leak
    - 98cc63a Fixes a memory leak un-cleared stream connections.
  - 1f57602 Merge pull request #50837 from Ch3LL/pyren_doc
    - a873019 Clarify py render docs when data context is available
  - fdff675 Merge pull request #50844 from twangboy/fix_grains
    - 473f4d7 Fix some unrelated lint
    - fd1a072 Fix grains for future windows releases
  - 32dfaba Merge pull request #50816 from dubb-b/2017.7
    - dc7ff48 Pinning bootstrap to the tagged version of salt.
  - b1e5f3f Merge pull request #50709 from mattLLVW/service_unmask
    - b199255 Merge branch '2017.7' into service_unmask
    - 38addd1 salt style guideline
    - 823c4ad Implement unmask in states.services
    - 91a6bf5 Functional unmask, must add tests
  - e4e9563 Merge pull request #50633 from astronouth7303/loader-docs
    - 03e3168 whitelist_modules is linked via enable_whitelist_modules for some reason? idk
    - 5ca1f8e Typos
    - 7cb1a87 Add a quick index of loader-related settings
    - 318ca48 Make a note to not import modules
    - 6294658 Merge remote-tracking branch 'origin/2017.7' into loader-docs
    - 678958e Editing
    - 9d59629 Merge branch '2017.7' into loader-docs
    - 374d0e9 Finish the last sections
    - 98da110 Migrate Dynamic Module Distribution to Modular Systems.
    - 1bdd49d Add more links and write some system summaries
    - bd3796a Update TOC
    - 3a9ab1b Merge dunders into developing, expand to cover all dunders.
    - d06739a Add short writeups for a bunch of systems.

    - 705e7e4 Add a bunch of cross linking
    - 40f9b16 Add missing module type sections
    - 172d8c2 Refactor headings
    - 53154a0 Oops, missed that
    - 0dcd341 Possible horrific use of toctrees? maybe
    - bed3509 Refactor, add a TOC
    - f647c1d Fill in some missing modules
    - 9d4b596 Merge branch '2017.7' into loader-docs
    - 26468df Add missing blank line
    - 205fea9 Some more revising
    - d4a8f7e Remove some code from some links
    - 65e4199 I feel like it should be up there
    - d44f5ce Rephrase footnote
    - 125e46e Add section on how modules get loaded, including giant table of names
    - 5bfc6c6 Update reference to modules/index
    - 4b096c1 Move all the module-development documentation into a single folder
  - dcdb9b2 Merge pull request #50750 from bewing/pepa12
    - 9931a41 Fix last key detection in pepa key_value_to_tree
  - cb85d5d Merge pull request #50733 from marmarek/patch-3
    - 1011421 Report salt-call exit code from ssh_py_shim.py
  - b086ff3 Merge pull request #50756 from GwiYeong/local-client-fix
    - c488092 fix local client missing minion return
- **ISSUE** #50850: (akrus) key_cache: sched prints error messages in logs (refs: #50859)
- **PR** #50859: (garethgreenaway) [2018.3] Fixes for key_cache and python3 @ *2018-12-13 23:57:07 UTC*
  - 216fa14 Merge pull request #50859 from garethgreenaway/50850_python_key_cache_file_binary
  - ebc9a01 When reading and writing the key cache file, when using Python3, ensuring the file is read & written in binary mode.
- **PR** #50828: (twangboy) Fix issues with lgpo state and util @ *2018-12-13 16:37:33 UTC*
  - 6b45aad Merge pull request #50828 from twangboy/fix_lgpo
  - 05baa95 Merge branch '2018.3' into fix_lgpo
- **PR** #50843: (bdrung) Fix RemoveCapacityFromDiskgroupTestCase require pyvmomi @ *2018-12-12 20:36:32 UTC*
  - 6ea059d Merge pull request #50843 from bdrung/fix-tests
  - 0d00ae4 Fix RemoveCapacityFromDiskgroupTestCase require pyvmomi
- **ISSUE** #50829: (ghost) process.absent UnicodeDecodeError (refs: #50839)
- **PR** #50839: (terminalmage) Fix UnicodeDecodeError in ps module @ *2018-12-12 19:59:12 UTC*
  - 14d7d1d Merge pull request #50839 from terminalmage/issue50829

  - 3a3d9b7 Ensure that unicode types returned from compatibility funcs
  - 1a00abc Add unit test for _get_proc_cmdline
  - 9eaa2ed Remove extraneous comments
- **ISSUE** saltstack/salt#49393: (Poil) file.directory recurse not always changing perms (refs: #50653)
- **PR** #50653: (lexvella) Check file_mode in file.directory and _check_directory @ *2018-12-12 16:30:52 UTC*
  - 264a042 Merge pull request #50653 from lexvella/2018.3
  - 4ea9386 Merge branch '2018.3' into 2018.3
  - 42e4ab3 Merge branch '2018.3' into 2018.3
  - 79f6b42 Check file_mode in file.directory and _check_directory
    - 1233e2b Fix some lint
    - 56c0e55 Fix issues with lgpo state and util
- **ISSUE** #45026: (angeloudy) bytes error in mysql module (refs: #50823)
- **PR** #50823: (terminalmage) salt.modules.mysql: Decode fetch results before making string comparisons @ *2018-12-11 21:04:41 UTC*
  - a8ad17a Merge pull request #50823 from terminalmage/issue45026
  - ec8e116 Decode fetch results before making string comparisons
- **PR** #50817: (dubb-b) Adding git 'v' to the bootstrap options @ *2018-12-11 17:28:10 UTC*
  - e31e731 Merge pull request #50817 from dubb-b/2018.3
  - 48fc01f Adding git 'v' to the bootstrap options
- **PR** #50780: (twangboy) Add netsh mechanism in the LGPO module @ *2018-12-10 22:09:20 UTC*
  - 9b6b39c Merge pull request #50780 from twangboy/lgpo_netsh
  - 5eea728 Fix some lint
  - f077783 Add some tests
  - 2dde12a Fix one more lint item
  - c55b7b4 Fix some lint
  - 1247598 Add more policies
  - b2b9795 Fix overly long lines
  - f18d1aa Add additional policies that use netsh
  - cb50648 Add versionadded tags
  - 00cbf26 Add the netsh mechanism to the lgpo module
- **PR** #50783: (sathieu) gitfs/git_pillar: Fix UnicodeDecodeError while cleaning stale refs @ *2018-12-10 20:34:53 UTC*
  - 0351296 Merge pull request #50783 from sathieu/gitfs_stalerefs_unicode
  - ef160a6 gitfs/git_pillar: Fix UnicodeDecodeError while cleaning stale refs
- **ISSUE** #45867: (Nick2253) Icinga2 Module uses incorrect paths (refs: #50615)
- **PR** #50765: (ClaudiuPID) Fix icinga2 cert path @ *2018-12-10 18:10:24 UTC*
  - **PR** #50615: (ClaudiuPID) Fixes icinga2 certs path for newer versions 2.8+ (refs: #50765)

  - deadf9c Merge pull request #50765 from ClaudiuPID/fix-icinga2-cert-path
  - ce1842e Merge branch '2018.3' into fix-icinga2-cert-path
  - 31878d5 Pylint fixes
  - 47bfba1 Migrated to cmd.run_all and tested on working srv
  - a1dddce Fixes icinga2 certs path for newer versions 2.8+
- **PR** #50692: (terminalmage) Update a couple test modules to use with_tempdir/with_tempfile decorators @ *2018-12-10 17:24:02 UTC*
  - 9f8289b Merge pull request #50692 from terminalmage/update-tests
  - 36a4a1a Merge branch '2018.3' into update-tests
  - be14517 Make x509 tests use tempfiles instead of hard-coding them
  - a394cd7 Update file integration tests to use decorator for tempdirs
- **ISSUE** #32245: (tkwilliams) git_pillar should permit automatic mapping of environment to branch a-la gitfs (refs: #50768)
- **PR** #50768: (sathieu) git_pillar: Add support for all_saltenvs parameter @ *2018-12-10 16:39:37 UTC*
  - 3a71567 Merge pull request #50768 from sathieu/git_pillar_all_saltenvs
  - e971412 git_pillar: Add support for all_saltenvs parameter
- **ISSUE** #48801: (dstoliker) salt.states.user.present - mindays, warndays, and inactdays won't return to salt's default values (refs: #50796)
- **PR** #50796: (dwoz) Fix #48801 min, max, inact, and warn changes @ *2018-12-10 16:30:14 UTC*
  - 12a8447 Merge pull request #50796 from dwoz/issue48801
  - 6b95223 fix linter errors
  - 87a04df Fix #48801 min, max, inact, and warn changes
- **ISSUE** #46388: (dynek) IPv4 & v6 network configuration on Debian (refs: #49355)
- **ISSUE** #40262: (pjediny) Salt enforces configuration from /etc/network/interfaces.d/* to /etc/network/interface (refs: #49355)
- **ISSUE** #38672: (valentin2105) IPv6 Network states is incomplete (refs: #49355)
- **PR** #50804: (Ch3LL) Back-port #49355 to 2018.3 @ *2018-12-10 16:21:31 UTC*
  - **PR** #49355: (MTecknology) Improve management of interfaces in debian-based distributions (refs: #50804)
  - **PR** #46980: (MTecknology) Fix broken rendering debian_eth.jinja. (refs: #49355)
  - b266e8b Merge pull request #50804 from Ch3LL/bp-49355
  - d34eade Use a temp file instead of /etc/network/interfaces for unit tests.
  - 622c354 Support reading multiple addresses from interface files.
  - 15c34fe Support unicode in space-delimited list; fixes unit tests in py2.
  - 0b59561 Added documentation about debian interfaces.d/*. (Fixes: #40262)
  - 1b096fb Removed python lint.
  - 6e41225 Finished adding support for multiple IP addresses.
  - 4c7df5f Cleaned up documentation/examples in states.network:

  - 601f434 Added support for -ipaddrs and -ipv6ipaddrs to modules.debian_ip().
  - 638f899 Added support for loopback devices to modules.debian_ip(). (Fixes: #38672)
  - cf61c8d Added a bunch of unit tests for modules.debian_ip.build_interface().
  - b905cf6 Added support for dns_{nameservers,search} being a list or string.
- **PR** #50767: (dwoz) Make sure ldap passwords are honored @ *2018-12-05 22:59:17 UTC*
  - dc9414c Merge pull request #50767 from dwoz/ldap_no_pass
  - 67aa591 Fix linter errors
  - be3945a Remove crufty import
  - 91f50a4 Add more ldap auth tests for ZD-2993
  - 055a8d5 Make sure ldap passwords are honored
- **ISSUE** #50364: (sathieu) git_pillar mountpoint ignores the root parameter (refs: #50626)
- **PR** #50626: (sathieu) git_pillar: Allow root and mountpoint parameters together @ *2018-12-05 17:46:59 UTC*
  - 60aa0d5 Merge pull request #50626 from sathieu/git_pillar_mountpoint_and_root
  - 5423c27 git_pillar: Allow root and mountpoint parameters together
- **PR** #50745: (bbinet) Fix grafana dashboard updating when nothing has changed @ *2018-12-04 19:26:32 UTC*
  - c02387e Merge pull request #50745 from bbinet/fix-grafana-dashboard-update
  - 7019d47 Fix grafana dashboard updating when nothing has changed
- **PR** #50737: (garethgreenaway) [develop] Merge forward from 2017.7 to 2018.8 @ *2018-12-04 16:02:13 UTC*
  - ae81498 Merge pull request #50737 from garethgreenaway/merge-2017.7
  - fde8db5 Merge branch '2017.7' into merge-2017.7
  - 246c055 Merge pull request #50718 from dwoz/run_run_timeout
    - ecd8486 Honor run_run timeout for shell tests
  - 979b3aa Merge pull request #50702 from damon-atkins/jenkins_pylint
    - 5936066 ensure archiveArtifacts are always collected. Use CHANGE_BRANCH to detect merge-forward
  - 22acaaa Merge pull request #50632 from astronouth7303/loader-exception-handler
    - 0f72815 Merge branch '2017.7' into loader-exception-handler
  - d9e7be1 Merge pull request #50670 from damon-atkins/jenkins_pylint
    - b6d1605 ci/lint corrected filenames and test condition on full lint Also fix CODEOWNER team name
      - 194801a Merge branch 'loader-exception-handler' of github.com:astronouth7303/salt into loader-exception-handler
      - dadeb0d Merge branch '2017.7' into loader-exception-handler
      - 8697744 Use single quotes per style guide
      - 07868f1 Improve logging output
      - fb05bd5 Eat an exception from an entry point instead of letting it kill the system.
- **PR** #50685: (Ch3LL) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-11-30 18:46:07 UTC*

  - edaa9fe Merge pull request #50685 from Ch3LL/merge_2018.3
  - d46aff0 Merge branch 2017.7 into 2018.3
    - 6dc3afc Merge pull request #50672 from garethgreenaway/49691_file_replace_search_only_return_false
      - 7e7df06 lint
      - 6b73c76 When using file.replace, with the search_only option, if the pattern does not exist in the file then we should return False.
    - 1236b51 Merge pull request #50659 from garethgreenaway/49954_gem_installed_less_than_greater_than_support
      - 0ec8bcf When using the gem installed state, when passing a version that includes greater than or less than symbols, ensure that the installed versions meets that requirement.
    - 6317f3a Merge pull request #50583 from damon-atkins/jenkins_pylint
      - 2d1f51c Fix lint only changes, full lint on merge forwards - lint only changes previous diff picked up out of data files, when the branch was out of date. - full limit on merge forward to pick up changes in the lint checks between versions. - update CODEOWNERS for .ci/*
    - 15bf09a Merge pull request #50605 from Oloremo/fix-supervisord-dead-state-idempotency
      - ba3f281 Merge branch '2017.7' into fix-supervisord-dead-state-idempotency
      - cf9ab84 Merge branch '2017.7' into fix-supervisord-dead-state-idempotency
      - af386a2 Fixed the supervisord.dead state idempotency
- **ISSUE** #50016: (abednarik) State augeas.change do not update config file (refs: #50690)
- **PR** #50690: (garethgreenaway) [2018.3] fixes to augeas_cfg module @ *2018-11-30 18:44:42 UTC*
  - 7cb2405 Merge pull request #50690 from garethgreenaway/50016_ensure_path_is_str
  - cec7cb4 Older versions of python-augeas need the path passed to aug.match to be a string.
- **ISSUE** #49392: (Poil) hosts not replacing existing entry (refs: #50657)
- **PR** #50657: (terminalmage) Rewrite host.present state to allow for removal of entries not matching IP @ *2018-11-29 17:58:40 UTC*
  - a5c7639 Merge pull request #50657 from terminalmage/issue49392
  - 2ddb587 Add release notes for new "clean" option
  - e4946f9 Rename "remove" argument to "clean"
  - 7fd3bce Add remove parameter to host.present state
  - 2671a30 Add unit tests for host.present state
  - 30f1b85 Remove non-matching IPs from hosts file
  - ec297a0 Performance improvements in hosts module
  - d1c2038 Add generic ip address validation function
  - 9f54720 Add ignore ability to process_read_exception
- **ISSUE** #48759: (MyIgel) acme.cert: Fail to update certificate after adding aliases / missing –expand (refs: #50683)
- **PR** #50683: (garethgreenaway) [2018.3] Fixes to acme module when adding aliases to existing certificate @ *2018-11-29 17:46:54 UTC*
  - b604785 Merge pull request #50683 from garethgreenaway/48759_adding_aliases_include_expand

  - 3940a0f When adding alises to an existing Certbot certificate, if we see a message about expanding in the stderr returned from cmd.run_all we should rerun the cmd with –expand included.

- **ISSUE** #50406: (glkappe) salt-ssh can't use Mongo returner (refs: #50664)
- **PR** #50664: (garethgreenaway) [2018.3] Fix to logic for configuring returners @ *2018-11-28 05:24:30 UTC*
  - 15f9ae4 Merge pull request #50664 from garethgreenaway/50406_salt_ssh_returner_configuration
  - 3525411 When pulling values out of the available configuration for returners we should always default to using keys for those returners, eg. mongo.user for the username. Otherwise in certain situations, eg. when using salt-ssh we will end up with the wrong value for the user.
- **PR** #50652: (twangboy) Fix *unit.utils.test_mac_utils* on Windows @ *2018-11-27 20:45:41 UTC*
  - afbcf03 Merge pull request #50652 from twangboy/fix_test_mac_utils
  - c4cb729 Fix path issues with the tests
- **PR** #50640: (garethgreenaway) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-11-26 22:33:22 UTC*
  - 61a824f Merge pull request #50640 from garethgreenaway/merge-2017
  - 409c98f Merge branch '2018.3' into merge-2017
- **ISSUE** #50403: (prometheanfire) tornado 5.1.1 is not co-installable with salt on python2.7 (refs: #50639)
- **PR** #50639: (dwoz) Handle pipenv like version restrictions @ *2018-11-26 19:42:11 UTC*
  - 20a2de3 Merge pull request #50639 from dwoz/pipfix
  - 9c6f653 Handle pipenv like version restrictions
    - fced78b lint
    - 45a1aa3 Merge branch '2017.7' into merge-2017
    - b2cf134 Merge pull request #50463 from dwoz/zmqpubserv
      - a289e3c Merge branch '2017.7' into zmqpubserv
    - 6679242 Merge pull request #50590 from garethgreenaway/bp-50333
      - 1f5aa4b pkg.installed currently fails when sources is used along with hold: True. This was due to a change in #48426 that swapped out sending the pkgs variable for the desired variable instead. This caused problems with pkg.hold because desired and sources are always populated, and pkg.hold can only include one or the other. This change just includes desired in the call to pkg.hold since desired has the same value for sources.
      - 39e811b Add issue url to tests
      - c4d7a43 Add debug log when connecting
      - 4f81432 Set subscribe via sockopt for better compatibility
      - 6882209 More reliable pub server publishing
- **ISSUE** #49957: (whytewolf) salt-master scheduled items ignore maxrunning. (refs: #50130)
- **PR** #50130: (garethgreenaway) [2018.3] Fixes to schedule maxrunning on master @ *2018-11-26 18:35:35 UTC*
  - 5b7ab35 Merge pull request #50130 from garethgreenaway/49957_master_schedule_ignoring_maxrunning
  - 2f6dac9 Updating test_maxrunning_master to Mock get_running_jobs.
  - abda0a4 Reverting changes to doc/conf.py.
  - 6d072fc Some additional cleanup.

  - 4bd0406 Adding integration.scheduler.test_maxrunning to ignore list.
  - 45e160d Adding integration.scheduler.test_maxrunning to filename_map.yml
  - 2201a46 Fixing lint
  - 6c0ee32 removing unused cache_jobs function from utils/master.py
  - c2ed1d4 Adding some logging when the cache files are unable to be removed.
  - 44940d6 Ensure minion tests using minion options.
  - 722be43 Fixing lint
  - 65b4421 Adding some master specific functions to uitls/masters.py to determine if a Salt process is running. Updating utils/schedule.py to use the appropriate running function either from utils/master.py or utils/minion.py depending on where the scheduled job is running. Adding tests to test maxrunning in scheduled jobs for both the minion and master.
  - 0d65304 Swapping manual mocking to autodoc_mock_imports
- **ISSUE** #50542: (doug-stratoscale) mysql.user_exists doesn't work with Mysql 8.0.13 (refs: #50551)
- **PR** #50551: (garethgreenaway) [2018.3] Fixes to verify_login in mysql module @ *2018-11-26 17:44:25 UTC*
  - 26759c2 Merge pull request #50551 from garethgreenaway/50542_mysql_ensure_verify_login_uses_connection_host
  - 96742aed4 Merge branch '2018.3' into 50542_mysql_ensure_verify_login_uses_connection_host
  - eb510a6 Merge branch '2018.3' into 50542_mysql_ensure_verify_login_uses_connection_host
  - 6d2309d Merge branch '2018.3' into 50542_mysql_ensure_verify_login_uses_connection_host
  - db89b27 Merge branch '2018.3' into 50542_mysql_ensure_verify_login_uses_connection_host
  - 0284323 Ensure that verify_login is using the host from the connection_args and not the host associated with the user. Adding a test to ensure user_exists when the passed host is the MySQL wildcard %.
- **PR** #50619: (s0undt3ch) Don't squash tracebacks and Unicode fixes @ *2018-11-26 17:04:31 UTC*
  - 14c4fd5 Merge pull request #50619 from s0undt3ch/2018.3
  - a4771d2 UNICODE!
  - 72205a5 DO! NOT! SQUASH! TRACEBACKS!
- **ISSUE** #50562: (soostdijck) scheduled highstate ignores blackout (refs: #50586)
- **PR** #50586: (garethgreenaway) [2018.3] minion_blackout for scheduled jobs @ *2018-11-23 11:15:38 UTC*
  - 59809a1 Merge pull request #50586 from garethgreenaway/50562_minion_blackout_scheduler
  - 48673b7 Adding some defaults for minion_blackout code when pillar & grains are not available in opts. Removing some debugging code.
  - 9f422b2 Moving the minion_blackout code to the start of the try…except. Adding a test to ensure we get a log.exception when minion_blackout is True.
  - 2bf8fa5 Adding support for minion_blackout for jobs that are run from the Salt scheduler.
- **PR** #50576: (garethgreenaway) Back-port PR #50549 to 2018.3 @ *2018-11-21 14:39:55 UTC*
  - **PR** #50549: (cachedout) Add security credits to release notes (refs: #50576)
  - 5c28179 Merge pull request #50576 from garethgreenaway/bp-50549
  - 807068d Typo fix
  - 2ef8307 Typo fix

  - 7c8957a Typo fix
  - c1ca603 Merge branch '2018.3' into bp-50549
  - b163380 Add security credits to release notes
- **PR** #50599: (terminalmage) Merge Sphinx fixes from 2017.7 to 2018.3 @ *2018-11-21 13:31:18 UTC*
  - c9531d8 Merge pull request #50599 from terminalmage/merge-2018.3
  - ba82b07 Merge remote-tracking branch 'upstream/2017.7' into merge-2018.3
  - 1b8a8ac Merge pull request #50592 from terminalmage/docs-2017.7
    - db15776 Remove overline from section title
    - 8127dd5 Disable unused shorturls plugin
- **ISSUE** #50254: (jmcook1) file.line adding line every time (refs: #50366)
- **PR** #50366: (t0fik) 2018.3 fix issue50254 @ *2018-11-20 20:31:30 UTC*
  - 60b4622 Merge pull request #50366 from jdsieci/2018.3-fix-issue50254
  - a9b9fa2 Fixed pylint warnigs
  - 342786b Issue #50254 fixed
  - 1ab59e7 Added integration tests
  - 110f74f Added unit tests reproducing issue
- **ISSUE** #50558: (angeloudy) 'test=True' does not work with artifactory.download (refs: #50580)
- **ISSUE** #50155: (mikemartino) artifactory.downloaded state is not respecting test=True (refs: #50580)
- **PR** #50580: (garethgreenaway) [2018.3] Adding test=True to artifactory.download. @ *2018-11-20 16:37:14 UTC*
  - 735c9f4 Merge pull request #50580 from garethgreenaway/50155_artificatory_download_state_test_true
  - 7dd57d0 Lint.
  - 3661ee0 Adding test=True to artifactory.download.
- **ISSUE** #46292: (asymetrixs) win_wua should have possibility to specify source file (refs: #50397)
- **PR** #50577: (garethgreenaway) Back-port PR #50397 to 2018.3 @ *2018-11-19 21:28:42 UTC*
  - **PR** #50397: (tlemarchand) Add Win WUSA management, to install Windows Update files (.msu). (refs: #50577)
  - 015834d Merge pull request #50577 from garethgreenaway/bp-50397
  - dd2d872 Documentation
  - af6d21f Import unicode_literals from future and add versionadded tag
  - 09dc09d Disable powershell modules list Add list_kbs function
  - 05852d0 pchanges to changes named parameters
  - 16525e4 Add Win WUSA management, to install Windows Update files (.msu).
- **PR** #50570: (garethgreenaway) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-11-19 20:04:05 UTC*
  - 12e0d80 Merge pull request #50570 from garethgreenaway/merge-2018.3
  - 4c85119 merge-forward 2017.7 to 2018.3
  - 02238a7 Merge pull request #50520 from dwoz/issue_50459

    - 1aa2e34 Simplifie and portable homedir lookup
    - f14f747 Fix windows home dir lookup
    - d3620c3 Always set .gitconfig dir to user's home
  - edd9738 Merge pull request #50532 from dubb-b/2017.7
    - 009e52d Adding tag to ec2 cloud tests
  - b0446f5 Merge pull request #50500 from terminalmage/2017.7-docs
    - 870a364 Set docs jenkins job to treat warnings as errors
    - e03bdcd Update docs references for states with func_alias
    - 8a2c08f Disable custom sphinx automodule
- **ISSUE** #50266: (4815162342lost) grains.virtual show incorrect info (refs: #50519)
- **PR** #50519: (garethgreenaway) [2018.3] Fix to virtual core grain @ *2018-11-16 23:42:18 UTC*
  - 6fba83c Merge pull request #50519 from garethgreenaway/50266_core_virtual_grain_fixes
  - dcae2f0 Merge branch '2018.3' into 50266_core_virtual_grain_fixes
- **PR** #50427: (sathieu) Backports git_pillar tests improvements to 2018.3 @ *2018-11-16 16:37:29 UTC*
  - **PR** #50425: (sathieu) git_pillar: Add tests for mountpoint parameter (refs: #50427)
  - **PR** #50407: (sathieu) git_pillar: Add tests for root parameter (refs: #50427)
  - **PR** #50378: (sathieu) Don't fail on git_pillar tests when destructive tests are not enable (refs: #50427)
  - 84edd62 Merge pull request #50427 from sathieu/bp-git_pillar_tests
  - 12ea1cc git_pillar: Add tests for mountpoint parameter
  - fce467e git_pillar: Add tests for root parameter
  - ba150a5 Don't fail on git_pillar tests when destructive tests are not enable
- **PR** #50530: (terminalmage) Enable proxy/ssh daemons when filename mapping causes those tests to be run @ *2018-11-15 23:03:06 UTC*
  - 4651481 Merge pull request #50530 from terminalmage/salt-jenkins-1209
  - fbee11d Enable proxy/ssh daemons when filename mapping causes those tests to be run
  - 4f30611 lint
  - 4f9eb95 Fixing a typo in the _virtual function, should be checking for existing grains in osdata not grains. Updating the detection to look for /sys/bus/xen/drivers/xenconsole instead of specifically looking for any files under /sys/bus/xen/drivers. Some systems that are not running as Xen PV hosts include files under that location, particular Oracle Linux.
- **ISSUE** #50224: (mruepp) Augeas changes state with setm not working (refs: #50526)
- **PR** #50526: (terminalmage) Work around augeas' lack of unicode support @ *2018-11-15 17:33:26 UTC*
  - 6c07ba4 Merge pull request #50526 from terminalmage/issue50224
  - ee246ed Work around augeas' lack of unicode support
- **PR** #50503: (terminalmage) Set jenkins docs job to treat any warnings as errors (2018.3 branch) @ *2018-11-14 19:30:24 UTC*
  - d83da51d Merge pull request #50503 from terminalmage/2018.3-docs
  - a4fc108 Fix docs configuration for renamed logstash engine

  - 5b06464 Set docs jenkins job to treat warnings as errors
  - 9b6c669 Update docs references for states with func_alias
  - 62b5d40 Disable custom sphinx automodule
- **ISSUE** #48343: (bornwitbugs) CPU Beacon TypeError (refs: #50514)
- **PR** #50514: (bornwitbugs) Load beacon fix @ *2018-11-14 19:20:33 UTC*
  - fa2f4a5 Merge pull request #50514 from bornwitbugs/load_beacon_fix
  - e59ced6 Merge branch '2018.3' of github.com:saltstack/salt into load_beacon_fix
- **PR** #50509: (Ch3LL) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-11-14 17:52:06 UTC*
  - a50377b Merge pull request #50509 from Ch3LL/merge-2018.3
  - 178dab3 Merge branch 2017.7 into 2018.3
  - c00fd43 Merge pull request #50493 from rallytime/bp-50362
    - 60559cf Remove flaky test
    - a696a8f Fix test_matcher on Windows
  - 3cac1d0 Merge pull request #50494 from terminalmage/2017.7-docs
    - d561b26 Late import to fix docs build errors
  - 0829944 Merge pull request #50479 from Ch3LL/digital_ocean_location
    - 917781d Lint: Add blank line
    - 3583519 Update digital ocean location for cloud tests
  - accf681 Merge pull request #50480 from terminalmage/docs
    - c34690e Don't import obsolete Directive class from Sphinx
  - 2e9b3d2 Merge pull request #50478 from rallytime/update-bootstrap-docs
    - 66b078c Update the bootstrap script documentation to point to bootstrap repo
- **PR** #50502: (twangboy) Add definition for RestrictRemoteSAM @ *2018-11-14 14:30:38 UTC*
  - 55b4914 Merge pull request #50502 from twangboy/lgpo_2018.3
  - fe92a3a Add definition for RestrictRemoteSAM
- **ISSUE** #48734: (oddfellow) salt-run jobs.list_jobs has Target: unknown-target (refs: #50328)
- **ISSUE** #43453: (kunal-bajpai) Runner job *load* is never saved to job cache (refs: #43454)
- **PR** #50328: (rallytime) Fix issue with salt-run jobs.list_jobs where Target: unknown-target @ *2018-11-13 19:49:58 UTC*
  - **PR** #43454: (kunal-bajpai) Fix save_load which is never called for returner jobs (refs: #50328)
  - c1dde7e Merge pull request #50328 from rallytime/fix-48734
  - 2c51cf9 Make target test job more specific to reduce test flakiness
  - caace3f Rewrite job cache target test to be more robust
  - b29fcfa New test must be py3 compatible…oops. :)
  - c684c29 Fix issue #48734
  - 6859e0e Add regression test for issue #48734

- **PR** #50499: (dwoz) Fix 2018.3 build @ *2018-11-13 18:46:54 UTC*
  - 351faeb Merge pull request #50499 from dwoz/build_fix
  - 5754b4a Fix 2018.3 build
- **PR** #50496: (terminalmage) Fix 2018.3 docs build warnings @ *2018-11-13 17:21:09 UTC*
  - 6f4b23f Merge pull request #50496 from terminalmage/2018.3-docs
  - cb25641 Fix header underline length to avoid another sphinx warning
  - 9272f7b Fix more sphinx build errors
  - 18a8104 Late import to fix docs build errors
  - d457132 Fix sphinx build warnings in 2018.3 branch
- **ISSUE** #50227: (sheagcraig) mac_utils *_available_services* will fail with malformed launchd plists present (refs: #50228)
- **PR** #50492: (rallytime) Back-port #50228 and #50443 to 2018.3 @ *2018-11-13 16:59:15 UTC*
  - **PR** #50443: (weswhet) fix when checking for service Label on macOS (refs: #50492)
  - **PR** #50228: (sheagcraig) Fix 50227 malformed plists (refs: #50492, #50443)
  - 6b272c1 Merge pull request #50492 from rallytime/bp-50228-and-50443
  - 7879a2d Fix logging format
  - f578392 Lint Fix: add mock_open import
  - e5b999b add try/except when checking for service Label
  - 3c36bdd Fix linting errors.
  - fab57de Write tests to handle invalid launchd plist XML.
  - bb429b0 Condense service result assertions, add a binary plist test.
  - a809fef Factor out another helper function, implement broken symlink test.
  - 216f7ef Update testing for mac_utils services.
  - 89d8267 Fix trailing-slash bug in User LaunchAgent appending.
  - 0e23b6e Handle bad plists by ignoring them, with py2 and py3 specific approaches
  - fa44149 Avoid *AttributeError* when checking for exception types.
  - fd62b33 Add secondary exception handler for malformed plists.
  - ae6f49b Make test correctly generate a double exception.
- **PR** #50486: (twangboy) Back port lgpo from Fluorine @ *2018-11-13 09:40:17 UTC*
  - da980d9 Merge pull request #50486 from twangboy/lgpo_2018.3
  - bbbc07b Back port lgpo from Fluorine
- **ISSUE** #50218: (ata-sql) git.latest with unless command fails after upgrade to 2018.3.3 (refs: #50456)
- **PR** #50456: (terminalmage) Two bugfixes @ *2018-11-13 06:06:52 UTC*
  - 7dd3438 Merge pull request #50456 from terminalmage/issue50218
  - 6143408 Add unit test for TimedProc regression
  - e234abd Add support for list of commands in git states' onlyif/unless

  - cbf05c8 TimedProc: Don't assume args is a string
  - 87de250 Exit gracefully when mod_run_check raises exception
- **PR** #50455: (dubb-b) Adding the released version as the bootstrap minion. @ *2018-11-12 22:52:53 UTC*
  - d8514e6 Merge pull request #50455 from dubb-b/2018.3
  - fb61482 adding git
  - 181b8a4 removing stable
  - fc93ad6 Adding the released version as the bootstrap minion.
- **PR** #50483: (rallytime) Back-port #50272 to 2018.3 @ *2018-11-12 22:45:05 UTC*
  - **PR** #50272: (jyurdal) Fix out of order docker Env comparison (refs: #50483)
  - cfb3351 Merge pull request #50483 from rallytime/bp-50272
  - 0f02da7 Remove unnecessary u prefix from string literals
  - fc26ae8 Add unit test for comparing containers with the same env vars
  - cf7cb4b extra blank line after function definition(pylint)
  - 1de9ce3 Compare sorted docker environment
- **PR** #50481: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-11-12 22:44:15 UTC*
  - 360ae6a Merge pull request #50481 from rallytime/merge-2018.3
  - 1517bb3 Update old utils paths to use new paths
  - d8e7c47 Merge branch '2017.7' into '2018.3'
  - e5f97cd Merge pull request #50467 from dwoz/test_kitchen_up
    - 8c5e93e Use patched version of test-kitchen until next release
  - b16ff9c Merge pull request #50469 from dwoz/minion_timeout_test
    - 57f702d Fix linter warning
    - 686153a Fix up integration.minion.test_timeout
  - 45b438e Merge pull request #50470 from dwoz/set_time_flaky
    - 794ba17 Reset date and time after changing
  - c85561e Merge pull request #50442 from rallytime/replace-docbanner
    - 193325d Update docbanner hyperlink
  - 7200568 Merge pull request #50439 from rallytime/bootstrap-contributing-docs
    - d2158aa Add bootstrap section to contributing docs
  - cca7a36 Merge pull request #50441 from rallytime/replace-docbanner
    - 2af92bd Update SaltConf doc banner
  - 1e72d20 Merge pull request #50414 from flassman/backmerge-for-typeerror-windows-python3
    - 20fff15 use exc.winerror not exc[0] to avoid python3 TypeError
  - 93095cb Merge pull request #50415 from flassman/pass-cwd-to-runas_system
    - 0113137 cwd needs to be passed from runas() to runas_system()
- **ISSUE** #50461: (absolutejam) elementaryOS 'pkg' __virtual__ returned false (refs: #50468)

- **PR** #50468: (garethgreenaway) [2018.3] Adding elementary override to grains/core.py @ *2018-11-12 16:41:57 UTC*
  - 3aaad17 Merge pull request #50468 from garethgreenaway/50461_fix_elementary_os_family_grain
  - 77e8dcc On later versions of elementary, the os_family is being populated as elementary. In order for the aptpkg module to function, we need to override is to be Debian.
- **ISSUE** #50311: (marek-obuchowicz) pkg.installed state fails even though it suceeded (refs: #50590, #50333)
- **ISSUE** #46689: (mxork) pkg.installed: hold: True not applied to a package which is already installed. (refs: #48426)
- **PR** #50333: (garethgreenaway) [2018.3] Fix to pkg.installed for sources and hold: True (refs: #50590) @ *2018-11-08 18:02:33 UTC*
  - **PR** #48426: (garethgreenaway) [2017.7] fixes to states/pkg.py (refs: #50333)
  - d9179fb Merge pull request #50333 from garethgreenaway/50311_pkg_installed_fails_sources_hold
  - 7bc9b3d Merge branch '2018.3' into 50311_pkg_installed_fails_sources_hold
  - 6e96728 pkg.installed currently fails when sources is used along with hold: True. This was due to a change in #48426 that swapped out sending the pkgs variable for the desired variable instead. This caused problems with pkg.hold because desired and sources are always populated, and pkg.hold can only include one or the other. This change just includes desired in the call to pkg.hold since desired has the same value for sources.
- **PR** #50434: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-11-08 17:40:39 UTC*
  - 193c80a Merge pull request #50434 from rallytime/merge-2018.3
  - 83dfbb2 Merge branch '2017.7' into '2018.3'
    - ef4509e Merge pull request #50370 from dwoz/pubd_msg_logs
      - 58ced12 log jid to payload size
      - 3c537b5 Log message flow to publish daemon
- **PR** #50417: (meaksh) Get the right target when using "__env__" on git ext_pillar to avoid merging problems @ *2018-11-08 17:07:03 UTC*
  - 73ce80f Merge pull request #50417 from meaksh/2018.3-fix-git_pillar-__env__-solving-for-merging
  - 71af32d Remove unicode references
  - b7af5e5 Test git ext_pillar across multiple repos using __env__
  - 59894e2 Resolve target branch when using __env__
- **ISSUE** #50292: (dstoliker) sysctl in test=True mode results in [ERROR] Could not open sysctl file (refs: #50423)
- **PR** #50423: (garethgreenaway) [2018.3] Fixes to sysctl modules and state @ *2018-11-08 14:13:00 UTC*
  - e9464c9 Merge pull request #50423 from garethgreenaway/50292_no_error_log_test_true
  - 95a7f12 Adding file check and empty list return to freebsd_sysctl.py
  - 469c090 Adding a check to see if the config file exists, if not then return an empty list so we get the would be changes. Adding a test for this functionality.
- **ISSUE** #50420: (weswhet) mac_system virtual function fails to load on 2018.3.3 (refs: #50421)
- **PR** #50421: (weswhet) fixing an issue on mac_system where the virtual function wouldnt load @ *2018-11-07 21:57:44 UTC*
  - fccf875 Merge pull request #50421 from weswhet/fix-mac_system-loader

    - 042445d fixing an issue on mac_system where the virtual function wouldnt load
- **PR** #50409: (casselt) Fix test mode for mount state if device_name_regex is used. @ *2018-11-07 16:04:45 UTC*
    - d2fc910 Merge pull request #50409 from casselt/mount-device_name_regex-test-mode-fix
    - 56d5a32 Fix test mode for mount state if device_name_regex is used.
- **PR** #50383: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-11-06 15:51:53 UTC*
    - c2fda9d Merge pull request #50383 from rallytime/merge-2018.3
    - e993692 Merge branch '2017.7' into '2018.3'
        - cd7c95f Merge pull request #50128 from mgomersbach/fix-py3-reload-compat
            - e453e82 Satisfy pylint, use naming instead of W or E classification
            - d20581c Merge branch 'fix-py3-reload-compat' of github.com:mgomersbach/salt into fix-py3-reload-compat
            - 48c6a4f Merge branch '2017.7' into fix-py3-reload-compat
            - 5cd9741 Satisfy pylint
            - a2838db Satisfy pylint checks
            - 05c598c Fix import error by utilizing six and strict version spec
            - 95174e6 Merge branch '2017.7' into fix-py3-reload-compat
            - ab8b5b4 Update workaround to root of file
            - 37ab1f8 Update workaround to root of file
            - 253ab11 Workaround for py2 builtin, =<3.3 imp and >=3.4 libimport quirks, solves #50127
        - aa6b149 Merge pull request #50354 from rallytime/version_num_docs
            - 0621f7d Add a few more feature release codenames to the version_numbers doc
        - 6472100 Merge pull request #50359 from dwoz/clitest
            - 18b5d43 Increase cli batch test timeout for increased reliablity
        - 99c8f35 Merge pull request #50345 from dwoz/pub_d_logging
            - fd58523 Fix linter warnings
            - 5e1db3c Fix ZMQ PubServer daemon logging works on winndows
            - 8899832 Configure logging for ZMQ PubServer daemon
        - 44f7af4 Merge pull request #50350 from dubb-b/2017.7
            - 346a831 Updating the AMI's and sizes for the test run.
        - b947a18 Merge pull request #50274 from rallytime/bp-41472
            - 0a1e607 When the minion key is denied by the master then exit status 77 (permission denied)
        - 4725acd Merge pull request #50195 from damon-atkins/jenkins_pylint
            - eb13407 capture some extra info
        - ab8e845 Merge pull request #50329 from twangboy/fix_cloud_winrm
            - cf5fb04 Check WinRM port before initiating connection
        - e535b38 Merge pull request #50252 from dwoz/backport_fs_fix

        - 59ac284 Do not allow age to be a negative number

- **ISSUE** #49523: (MTecknology) [salt-cloud] Non-ascii chars cause an exception in salt-cloud (refs: #50146, #50236)
  - **PR** saltstack/salt#50174: (rallytime) Back-port #50146 to 2018.3 (refs: #50231)
- **PR** #50236: (rallytime) Fix salt-cloud UnicodeEncodeError when writing to stdout @ *2018-11-05 22:04:18 UTC*
  - **PR** #50235: (rallytime) Don't encode the return values in utils/vt.py (refs: #50236)
  - **PR** #50231: (rallytime) Revert "Back-port #50146 to 2018.3" (refs: #50236)
  - **PR** #50230: (rallytime) Add various ssh tests to the filemap for salt/utils/vt.py changes (refs: #50291, #50235)
  - **PR** #50174: (rallytime) Back-port #50146 to 2018.3 (refs: #50236)
  - **PR** #50146: (MTecknology) Improve handling of non-ascii characters in terminal output. (Fixes: #49523) (refs: #50235, #‘saltstack/salt‘#50174‘_‘_, #50174, #50231, #50236)
  - 46c7dc2d Merge pull request #50236 from rallytime/fix-cloud-vt
  - c1984c9 Fix salt-cloud UnicodeEncodeError when writing to stdout
- **ISSUE** #50062: (whytewolf) max_event_size does not always trim large events. (refs: #50382)
- **PR** #50382: (garethgreenaway) [2018.3] Fix to utils/dicttrim.py to honor max_event_size with nested dictionaries @ *2018-11-05 20:04:06 UTC*
  - ed4fa87 Merge pull request #50382 from garethgreenaway/50062_max_event_size_does_not_always_trim_large_events
  - 039f3d1 Increase limit for trimming.
  - 7f77bfe Adding test for utils/dicttrim.py
  - 9b55ac7 Adding _trim_dict_in_dict to utils/dictrim.py to be called from trim_dict when data contains a nested dictionary. This will ensure that values will still be trimmed correctly.
- **PR** #50387: (rallytime) Back-port #50339 to 2018.3 @ *2018-11-05 20:03:15 UTC*
  - **PR** #50339: (bartlaarhoven) The driver requires dependency, not the other way around. (refs: #50387)
  - 493d653 Merge pull request #50387 from rallytime/bp-50339
  - a3f264d Lint: Add extra blank line
  - bd18412 The driver requires dependency, not the other way around.
- **ISSUE** #41342: (githubcdr) beacons.state does not save as list (refs: #50347)
- **PR** #50347: (garethgreenaway) [2018.3] Fixes to beacon state module @ *2018-11-05 14:19:41 UTC*
  - 3022b7d Merge pull request #50347 from garethgreenaway/41342_beacon_state_module_fixes
  - 8885d1f Merge branch '2018.3' into 41342_beacon_state_module_fixes
  - c2354a9 Fixing beacon state test. Adding a reset function to beacon module to clear out beacon configuration. Useful for tests runs to ensure bits are left over between runs.
  - 2390f47 Adding SaltReturnAssertsMixin back in
  - 677d090 Remove zookeeper reference.
  - f3dee43 Updating the beacon state module to ensure that the format of the beacon data that is being sent along to the beacon execution module is in the right format.
- **ISSUE** #41583: (seanjnkns) mount.swap for swap file test=True results wrong (refs: #50358)

- **PR** #50358: (garethgreenaway) [2018.3] fixes to function mount.swap in mount state @ *2018-11-05 14:16:50 UTC*
  - 929525a Merge pull request #50358 from garethgreenaway/41583_mount_swap_test_true_wrong_results
  - b8ded8b Updating the swap function in the mount.swap function also check the device name when checking the fstab data. Updating tests to reflect new behavior.
    - d07c84d reordering ret dictionary in load test to ascending order
    - f704128 fixing type in load beacon module and adding option validators to config check function along with mock default values in test, fixes #48343
    - 4114a4f fixing type in load beacon module and adding option validators to config check function along with mock default values in test
- **ISSUE** #49225: (onnodb) [win_lgpo] lgpo_set always reports failure; Registry.pol corruption (refs: #50006)
- **ISSUE** #48782: (mike2523) [win_lgpo] Causes corrupt Registry.pol file (refs: #50006)
- **PR** #50006: (lomeroe) Correct more issues in lgpo @ *2018-11-02 16:32:29 UTC*
  - 3ffa392 Merge pull request #50006 from lomeroe/issue48782_2018.3
  - dc58252 update to use a single line if statement when dealing with prepended text
  - 0e011ad add runTest method to class for PY2
  - 377ec09 add a test for #47784
  - 0a14505 change backslashes in comment string to fix lint error
  - 02181d3 add fix/test for #50079
  - 05f2951 more lint fixes
  - 94b3335 lint fixes in test
  - 2a79626 update test to actually work
  - 42840ec add missing comma in function call
  - c283f50 Log a warning message instead of an exception when a SID cannot be converted to a username (for user rights assignments)
  - 271bd70 capture and print exception information
  - bc94b8e update method for creating size field of **delvals items for py3 compatibility
  - d9fe28a fix clobbering of admx_search_results which was keeping some policies from being properly detected
- **PR** #50340: (garethgreenaway) [2018.3] Backporting #48087 to 2018.3 @ *2018-11-01 18:15:37 UTC*
  - **PR** #48087: (mat813) Fix patching sysctl.conf on FreeBSD. (refs: #50340)
  - 12de4b9 Merge pull request #50340 from garethgreenaway/backport_48087_to_2018_3
  - 56c87ff Backporting #48087 to 2018.3
- **PR** #50275: (rallytime) Back-port #49752 to 2018.3 @ *2018-11-01 16:50:17 UTC*
  - **PR** #49752: (jodok) add support for clonenum parameter (refs: #50275)
  - 5e46434 Merge pull request #50275 from rallytime/bp-49752
  - e463d99 Merge branch '2018.3' into bp-49752
  - 1e10c6e add support for clonenum parameter

- **ISSUE** #47425: (raddessi) states/netyang.managed ignores models when passed as a kwarg (refs: #50301)
- **PR** #50316: (rallytime) Back-port #50301 to 2018.3 @ *2018-11-01 16:32:04 UTC*
  - **PR** #50301: (tonybaloney) Fixes issues raised in #47425 for states.netyang (refs: #50316)
  - 51e333a Merge pull request #50316 from rallytime/bp-50301
  - 367b689 Merge pull request #32 from tonybaloney/bp50301
    - 529687a update test assertions
  - ba526ad remove unused import
  - 220e57f create 2 simple unit tests for the netyang state module and fix issue in #47425
  - 20bf306 reproduce bug in 47425
- **PR** #50290: (twangboy) Fix *test_state* for Windows @ *2018-10-31 19:28:34 UTC*
  - af1075b Merge pull request #50290 from twangboy/fix_test_state
  - 69c02fc Fix some lint and some typos
  - d184013 Fix typo
  - 92b5ea0 Create salt util that copies file security info
- **ISSUE** #50050: (terminalmage) Backport runtests.py's –from-filenames functionality to 2018.3 branch (refs: #50308, #50291)
- **PR** #50291: (rallytime) Back-port #50230 to 2018.3 @ *2018-10-31 19:22:25 UTC*
  - **PR** #50230: (rallytime) Add various ssh tests to the filemap for salt/utils/vt.py changes (refs: #50291, #50235)
  - 045d81c Merge pull request #50291 from rallytime/bp-50230
  - 1aae681 Merge branch '2018.3' into bp-50230
  - 1ef712a Add various ssh tests to the filemap for salt/utils/vt.py changes
- **PR** #50315: (rallytime) [2018.3] Merge forward from 2018.3.3 to 2018.3 @ *2018-10-31 14:16:34 UTC*
  - 00437aa Merge pull request #50315 from rallytime/merge-2018.3
  - 422465e Merge branch '2018.3.3' into '2018.3'
  - 4d9e2a0 Merge pull request #50088 from vulnbe/fix-incorrect-id-handling-2018-3-3
  - 619f2f2 Fix incorrect id handling in file state
- **PR** #50309: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-10-30 16:07:25 UTC*
  - 84ee7a3 Merge pull request #50309 from rallytime/merge-2018.3
  - 381a59d Merge branch '2017.7' into '2018.3'
  - 11fa33e Merge pull request #50295 from dubb-b/remove-cloud-change
    - 80982ad Removing an un needed change and maybe fixing the lint error
  - e00c507 Merge pull request #50287 from whytewolf/boto_vpc
    - 5a39644 minor correction to boto_vpc.route_table_present example, subnet_names was on wrong indent level
  - 37ae06c Merge pull request #50280 from gtmanfred/2017.7
    - 5808074 fix InstallRequirement.from_line for pip 18.1

- **ISSUE** #50050: (terminalmage) Backport runtests.py's –from-filenames functionality to 2018.3 branch (refs: #50308, #50291)
- **PR** #50308: (rallytime) Fix matcher grain file @ *2018-10-30 15:27:47 UTC*
  - **PR** #50182: (terminalmage) Backport runtests.py modifications from develop to 2018.3 (refs: #50308)
  - b805bab Merge pull request #50308 from rallytime/test-fixes
  - f238522 Fix matcher grain file
- **PR** #50270: (terrible-broom) apt: speedup installing packages with '>=' or '<=' and reduced memory… @ *2018-10-30 13:24:36 UTC*
  - 57fe6cd Merge pull request #50270 from terrible-broom/feature/apt-cache-show-instead-of-dump
  - 6c1b05a Lint: add extra blank line
  - a4df706 fix review notes
  - d059edd apt: speedup installing packages with '>=' or '<=' and reduced memory consumption
- **PR** #50293: (gtmanfred) update jenkins version for 2018.3 @ *2018-10-29 23:39:34 UTC*
  - 6a65402 Merge pull request #50293 from saltstack/jenkins
  - fe6fbc4 update jenkins version for 2018.3
  - **PR** saltstack/salt-jenkins#1182: (gtmanfred) clean up salt.utils for using 2018.3.3 (refs: #50289)
- **PR** #50289: (gtmanfred) ignore empty line in localectl status output @ *2018-10-29 21:39:07 UTC*
  - 935f181 Merge pull request #50289 from gtmanfred/2018.3
  - fb43061 fix test
  - 5eb0e37 add extra space to localemod test
  - e8dc8ba if line in output of localectl is empty, ignore it
- **PR** #50276: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-10-29 19:05:15 UTC*
  - 10ba8b4 Merge pull request #50276 from rallytime/merge-2018.3
  - f05abb0 Merge branch '2017.7' into '2018.3'
    - 6ae8a33 Merge pull request #50281 from dubb-b/add_subnet
      - 4fdcf7c Lint: Add extra blank line
      - cebfd8a Fixing lint error
      - 0da6301 Merge branch '2017.7' of https://github.com/saltstack/salt into add_subnet
      - 2df6115 Adding subnetID so it knows the right VPC to build the node in.
  - a0cfbdc Merge branch '2017.7' into '2018.3'
    - b94a43b Merge pull request #50184 from ethanculler/fix-minion-return-exception-with-return
      - 283d41c Merge branch '2017.7' into fix-minion-return-exception-with-return
      - a4e54d7 Added handling for minion return exceptions containing the word "return"
    - 6b009b4 Merge pull request #50258 from dubb-b/fix_cloud_ec2
      - 6a1e9f0 Adding securitygroupname as the config instead of securitygroup
    - 3e1aac9 Merge pull request #50251 from gtmanfred/2017.7
      - 061e599 bump salt version to 2017.7.8

        - fe9bc81 Merge pull request #50233 from Ch3LL/bp-50232
            - f83460d change order of search engine ids
            - bf3f217 Update search engine IDs for releases
- **PR** #50282: (twangboy) Fix *unit.beacons.test_diskusage* on Windows @ *2018-10-29 18:37:13 UTC*
    - 0f1dc08 Merge pull request #50282 from twangboy/fix_test_diskusage
    - fd369dd Mock is_windows = False
- **PR** #50182: (terminalmage) Backport runtests.py modifications from develop to 2018.3 (refs: #50308) @ *2018-10-29 16:25:07 UTC*
    - f7c2f04 Merge pull request #50182 from terminalmage/issue50050
    - 113d76e Revert ”Move jinja templating tests alongside salt/utils/jinja.py’s tests”
    - 3057ca4 Lint fix: add extra blank line in core.py
    - 8968f9d Revert ”Rename test_map_conf test file to respect naming convention”
    - 85a9395 Remove saltclass pillar unit test for list expansion
    - 3d95673 Use get_config_dir() instead of self.config_dir
    - e8b3a62 Add init file back in for unit/client pkg
    - d527dd2 Rename states/test_libvirt.py into states/test_virt.py
    - ab4a670 Remove pgjsonb returner unit tests
    - 68ab3ee Remove ”test_custom” from grains section of filename_map
    - 6814698 Remove test_error from scheduler tests
    - c55dcf0 Add init file back in for integration/cloud/clouds pkg
    - f485b4a enable testing only filemaps
    - dfa05b9 Remove test files for beacons that don’t exist in 2018.3
    - 85612cf runtests.py: Fix IndexError in –from-filenames
    - 25097f9 Make sure from-filenames intersect with names-file
    - ed864ce runtests.py: Accept modified file list from a text file
    - 059754c Add some doc tests to the filemap that should run on every PR
    - b1ee2e5 Set a default value for the filename map path
    - 5ea1024 Add documentation explaining naming convention for test files
    - 8f790c0 Fix conflicting dict key in filename map
    - f2e5512 Evan Moar Lint
    - 65fef9a Moar Lint
    - 39c2455 Lint
    - dd20f12 Make sure we always run the module names test
    - 2ac939e Add test to check test module names and make sure they match the naming convention
    - 785d298 Add integration.netapi.rest_tornado.test_app to filename map
    - 662ca0d Rename files to match naming convention

  - 186d333 Add back changes lost in cherry-pick
  - a236531 Add back test accidentally deleted when cherry-picking
  - eaa4da3 Rename providers to clouds to respect naming convention
  - 85b96de Rename some files to avoid them being identified as test modules
  - bfb6015 Rename/consolidate salt.utils unit tests to conform to naming convention
  - 84749dd Consolitdate and move salt-ssh unit tests to proper location
  - 71825a5 Consolidate and move loader tests to test/unit/test_loader.py
  - 8326a09 Rename test_map_conf test file to respect naming convention
  - 6d9c2eb Move jinja templating tests alongside salt/utils/jinja.py's tests
  - 1ef43a3 Add additional paths to the filename map
  - 75e6731 Use the new helper to get the test mods
  - c3dc8ec Add function to get all of the test modules
  - c5593bc Rename custom_grains.py so it's not identified as a test module
  - 330c211 Consolidate saltclass pillar unit tests into single module
  - a642619 Rename test pgjsonb returner test file to match naming convention
  - 6433c48 Don't import old module name
  - a97e032 Move helper outside of for loop
  - 3f1b6fd Add more unit test mapping
  - ac3406d Fill out the filename map a bit
  - 07d6f2a Enhancements to –from-filenames, –filename-map options
  - b851a41 Don't run unit or integration tests unnecessarily
  - 569276b Add an initial filename_map
  - 44199b1 Add –from-filenames and –filename-map options to parser
- **PR** #50267: (m03) Fix missing aptly config paths @ *2018-10-29 13:27:23 UTC*
  - 1e8b092 Merge pull request #50267 from m03/fix-aptly-config-paths-20183
  - 319bb31 Fix missing aptly config paths
- **PR** #50232: (Ch3LL) Update search engine IDs for releases (refs: #50233) @ *2018-10-25 20:15:08 UTC*
  - ab0b022 Merge pull request #50232 from Ch3LL/search_latest
  - 3452d9b change order of search engine ids
  - c9546df Update search engine IDs for releases
- **ISSUE** #49523: (MTecknology) [salt-cloud] Non-ascii chars cause an exception in salt-cloud (refs: #50146, #50236)
  - **PR** saltstack/salt#50174: (rallytime) Back-port #50146 to 2018.3 (refs: #50231)
- **PR** #50231: (rallytime) Revert "Back-port #50146 to 2018.3" (refs: #50236) @ *2018-10-25 20:12:15 UTC*
  - **PR** #50146: (MTecknology) Improve handling of non-ascii characters in terminal output. (Fixes: #49523) (refs: #50235, #'saltstack/salt'#50174'_'_, #50174, #50231, #50236)
  - 1c8cdf9 Merge pull request #50231 from saltstack/revert-50174-bp-50146

  - c76ac1b Revert "Back-port #50146 to 2018.3"
- **ISSUE** #50132: (ManicoW) XCP-ng packages and services management (refs: #50140)
- **PR** #50172: (rallytime) Back-port #50140 to 2018.3 @ *2018-10-25 14:00:32 UTC*
  - **PR** #50140: (ManicoW) Adding XCP-ng support (refs: #50172)
  - 8dd4f43 Merge pull request #50172 from rallytime/bp-50140
  - 02e0ea0 lint fix
  - 819c46c fix lint errors and code factorization
  - 8aa3598 Fix lint error
  - 9092c4e Adding support for XCP-ng
  - c516bc5 Adding XCP-ng to os family map
- **PR** #50063: (twangboy) Use long when setting REG_QWORD @ *2018-10-25 13:45:03 UTC*
  - 9ad43f2 Merge pull request #50063 from twangboy/fix_50039
  - 5fdba7d Disable lint error
  - 84b8dd6 Add tests
  - 4644bb6 Use long when setting REG_QWORD
- **ISSUE** #49965: (nkrishnakishor) Beacon: Usage in windows (refs: #50188)
- **PR** #50188: (garethgreenaway) [2018.3] fixes to diskusage beacon @ *2018-10-25 13:41:19 UTC*
  - 01b083e Merge pull request #50188 from garethgreenaway/49965_beacon_diskusage_windows
  - 1f0ab27 Merge branch '2018.3' into 49965_beacon_diskusage_windows
  - 42b3203 Fixing lint
  - f65fa57 Fixing a bug in the diskusage beacon that prevented it from working on Windows. Adding a couple tests to test functionality on Windows.
- **PR** #50214: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-10-25 13:38:40 UTC*
  - 9ea51c0 Merge pull request #50214 from rallytime/merge-2018.3
  - 897145c Merge branch '2017.7' into '2018.3'
    - 017d394 Merge pull request #50202 from Ch3LL/_2017.7.8
      - d520f9a Add tests with filname in token data
      - d45f6c7 Add empty token salt-api integration tests
      - 652dbf6 Ensure that tokens are hex to avoid hanging/errors in cherrypy
    - f28a4fa Merge pull request #50211 from channias/fix-spm-modules-install
      - 9b203d5 Fix broken install of additionnal modules in SPM packages
    - 6b4e07b Merge pull request #50212 from dwoz/test_no_fail
      - 7f613ac Do not fail if process already ended
    - 85e22d2 Merge pull request #50207 from Ch3LL/2update_version_doc_2017.7
      - 198d25c Add 2017.7.8 previous release
      - d4e26ac Update release versions for the 2017.7 branch

        - d035695 Merge pull request #50205 from Ch3LL/rn_2017.7.8
            - 71ee68b [2017.7.8] remove in progress and add security details
        - ce341f2 Merge pull request #50180 from max-arnold/2017.7
            - 881a521 Revert curly braces example for nested dictionaries
        - 622bb51 Merge pull request #50026 from damon-atkins/jenkins_pylint
            - aebce88 Merge branch '2017.7' into jenkins_pylint
            - 5f708fa use awk to create the other files, so git does not need to search again
            - 6b96a24 Adjust jenkins linting process, only run stage if required, use git diff to find files, instead of find, report on status, changed and deleted files, lint only changed files.
        - aa66d1b Merge pull request #50158 from vutny/doc/2017.7/fix-http-tutorial
            - fe2d70d [DOC] Fix JSON payload example in HTTP tutorial
- **ISSUE** #50162: (rbthomp) Scheduled job runs in endless loop when using when: and splay: together. (refs: #50216)
- **PR** #50216: (garethgreenaway) [2018.3] Fixes to scheduler, list of whens plus splay @ *2018-10-25 13:31:26 UTC*
    - 51b3fa4 Merge pull request #50216 from garethgreenaway/50162_when_plus_splay_endless_loop
    - 685509f Fixing an issue when a combination of the when parameter as a list plus using the splay parameter would cause the schedule to continuously run jobs in an endless loop, regardless of if their scheduled time had been receached. Also fixing a related issue where scheduled jobs that rely on _next_fire_time were not being run as the corrected splayed time but rather running at the original scheduled time. Adding new tests.
- **PR** #50190: (dwoz) Fix test_managed_file_with_grains_data on Windows @ *2018-10-24 22:43:35 UTC*
    - 43303fb Merge pull request #50190 from dwoz/grains_template_test
    - b77db48 Fix test_managed_file_with_grains_data on Windows
- **PR** #50187: (garethgreenaway) [2018.3] Adding fixes to scheduler tests to 2018.3 @ *2018-10-24 19:45:32 UTC*
    - **PR** #49871: (garethgreenaway) [fluorine] Updating scheduler tests for OS X (refs: #50187)
    - 462314b Merge pull request #50187 from garethgreenaway/port_49871_to_2018_3
    - 25e7cb6 Merge branch '2018.3' into port_49871_to_2018_3
    - d232dfb Update test_eval.py
    - d8d8559 Fixing lint
    - 76e6232 Removing unnecessary debugging lines
    - d2d0622 Back porting #49871 to 2018.3. Adding some additional changes to ensure they are passing on OS X.
- **PR** #50201: (Ch3LL) [2018.3] Ensure that tokens are hex to avoid hanging/errors in cherrypy @ *2018-10-24 19:44:23 UTC*
    - 8ad9a1b Merge pull request #50201 from Ch3LL/_2018.3.3
- **PR** #50208: (Ch3LL) Update release versions for the 2018.3 branch @ *2018-10-24 15:52:19 UTC*
    - 88673ad Merge pull request #50208 from Ch3LL/2update_version_doc_2018.3
    - cdbf493 Add 2017.7.8 previous release
    - 6f775e7 Update release versions for the 2018.3 branch

- **PR** #50206: (Ch3LL) [2018.3.3] remove in progress and add security details @ *2018-10-24 15:50:50 UTC*
    - db8ef1e Merge pull request #50206 from Ch3LL/rn_2018.3.3
    - 896caa1 Add 2016.11.10 release notes to 2018.3 branch
    - 1e34158 [2018.3.3] remove in progress and add security details
- **PR** #50200: (rallytime) Back-port #50181 to 2018.3 @ *2018-10-24 15:49:59 UTC*
    - **PR** #50181: (cro) Testinfra has a python packaging issue that is making version 1.17.0 fail to install some places (refs: #50200)
    - 3d3db70 Merge pull request #50200 from rallytime/bp-50181
    - f3e5399 Ignore testinfra 1.17.0
- **ISSUE** #49523: (MTecknology) [salt-cloud] Non-ascii chars cause an exception in salt-cloud (refs: #50146, #50236)
- **PR** #50174: (rallytime) Back-port #50146 to 2018.3 (refs: #50236) @ *2018-10-23 18:16:45 UTC*
    - **PR** #50146: (MTecknology) Improve handling of non-ascii characters in terminal output. (Fixes: #49523) (refs: #50235, #‘saltstack/salt‘#50174‘_‘_, #50174, #50231, #50236)
    - 5f904dd Merge pull request #50174 from rallytime/bp-50146
    - 601a5b8 Improve handling of non-ascii characters in terminal output. (Fixes: #49523)
- **ISSUE** #50142: (m03) stringutils.to_none AttributeError: ’module’ object has no attribute ’tex_type’ (refs: #50143)
- **PR** #50173: (rallytime) Back-port #50143 to 2018.3 @ *2018-10-23 17:13:55 UTC*
    - **PR** #50143: (m03) Fix #50142, stringutils.to_none (refs: #50173)
    - **PR** #45429: (terminalmage) [PY3] Add unicode_literals to salt.utils modules (R-S) (refs: #50143)
    - 1cd0339 Merge pull request #50173 from rallytime/bp-50143
    - f5d6080 Add test_to_none
    - c0ce36b Fix #50142, stringutils.to_none
- **PR** #50171: (terminalmage) Fix invalid RST link in docker_network docs @ *2018-10-23 13:40:36 UTC*
    - 9331f1e Merge pull request #50171 from terminalmage/docs
    - 7ea1d6c Fix invalid RST link in docker_network docs
- **PR** #50161: (rmarcinik) Add a missing result for task.info @ *2018-10-23 13:31:43 UTC*
    - 16eece3 Merge pull request #50161 from rmarcinik/patch-1
    - 9848f06 Add a missing result for task.info
- **PR** #50136: (terminalmage) Add some debug logging before and after states gather pillar data @ *2018-10-22 14:40:38 UTC*
    - 28bc5e3 Merge pull request #50136 from terminalmage/debug-logging
    - 8592a0b Add some debug logging before and after states gather pillar data
- **ISSUE** #50129: (kmatsoukas) Call to network.get_route fails when interface name contains a dash (refs: #50147)
- **PR** #50147: (garethgreenaway) [2018.3] Fix to modules/network.py @ *2018-10-22 14:31:03 UTC*
    - 593c7ff Merge pull request #50147 from garethgreenaway/50129_network_get_route_fails_dash_in_name
    - 8fa87e3 Update test_network.py

  - 3117969 Fixing scenario where the interface for the default route has a dash in it, regular expression used in get_route currently doesn't not account for this.. Adding some additional tests for network.get_route.
- **PR** #50156: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-10-22 14:24:39 UTC*
  - 5c7a7d9 Merge pull request #50156 from rallytime/merge-2018.3
  - 87e20fd Merge branch '2017.7' into '2018.3'
  - 1a90c4f Merge pull request #50144 from dwoz/winrm_fs_release
    - 0951d85 Use newest winrm-fs release
  - e096560 Merge pull request #50113 from rallytime/bp-49989
    - 27c7ac9 Increase centos7-py2 kitchen pr timeout to 8 hours
- **PR** #50148: (MTecknology) Proxmox fixups @ *2018-10-22 11:55:10 UTC*
  - ce375f4 Merge pull request #50148 from MTecknology/proxmox-fixups
  - 4959947 Race condition still exists at one second timeout.
  - 3a62f74 Add support for setting root ssh pubkey to proxmox containers.
- **PR** #50085: (dwoz) Fix *test_win_pkg* yet again @ *2018-10-19 18:16:16 UTC*
  - 0ece06e Merge pull request #50085 from dwoz/win_pkg_redux
  - f22630e Merge branch '2018.3' into win_pkg_redux
  - 0b033a6 Do not allow age to be a negative number
  - a24d8b8 Debug file list cache
  - 9f5722a The test is not a flush problem
  - 071f8a0 Add cache file name to list cache log
  - 128c2c5 Flush and sync file contents
  - 7d361d8 Add debug log when file list is returned from cache
- **PR** #49980: (twangboy) Add support for AddPrinterDrivers @ *2018-10-19 13:19:41 UTC*
  - 0069247 Merge pull request #49980 from twangboy/lgpo_add_printer_drivers
  - 76c9301 Add support for AddPrinterDrivers
- **PR** #50114: (rallytime) Back-port #50061 to 2018.3 @ *2018-10-19 11:46:03 UTC*
  - **PR** #50061: (rongzeng54) pkg.install bugfix (refs: #50114)
  - de23c83 Merge pull request #50114 from rallytime/bp-50061
  - 1f09104 Add unit test for pkg.install with epoch
  - af253c6 pkg.install bugfix
- **ISSUE** #50074: (bigpick) Arista EOS Salt Minion - Debug Log Flooded with Repeated Message (refs: #50115)
- **PR** #50115: (garethgreenaway) [2018.3] Apply scheduler fixes and tests from #49104 to 2018.3 @ *2018-10-19 11:45:31 UTC*
  - **PR** #49104: (garethgreenaway) [fluorine] Fixes to scheduler for jobs with seconds, minutes, etc. (refs: #50115)
  - 9188bec Merge pull request #50115 from garethgreenaway/apply_49104_to_2018_3
  - adb1e79 Apply scheduler fixes and tests from #49104 to 2018.3

- **PR** #50119: (terminalmage) Squelch spurious errors logged in unit tests @ *2018-10-19 11:42:31 UTC*
  - e892042 Merge pull request #50119 from terminalmage/boto-tests
  - 91812ff Squelch spurious errors logged in unit tests
- **PR** #50120: (terminalmage) Fix bad merge conflict resolution @ *2018-10-19 11:38:35 UTC*
  - 35449b4 Merge pull request #50120 from terminalmage/salt-jenkins-1137
  - 1ddfd26 Fix bad merge conflict resolution
- **PR** #50013: (Giandom) Passed kwargs to db_exists in db_remove method @ *2018-10-18 11:28:30 UTC*
  - 5123488 Merge pull request #50013 from Giandom/2018.3
  - 9c91df3 Merge branch '2018.3' into 2018.3
  - 06bb245 Merge branch '2018.3' into 2018.3
  - 04a99e0 Passed kwargs to db_remove
- **PR** #50092: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-10-17 19:57:07 UTC*
  - aa94f60 Merge pull request #50092 from rallytime/merge-2018.3
  - 8609064 Update old utils paths to use new paths
  - a984ad8 Merge branch '2017.7' into '2018.3'
    - 7ec3840 Merge pull request #49794 from twangboy/fix_groupadd_test
      - 3d3b67d Gate for Windows
      - decfa52 Check value of result instead of entire dict
    - e562838 Merge pull request #50028 from basseed/fix-windows-virtual-grain
      - 8129ac6 removed whitespaces
      - 82fa866 Fix #50015, Windows grain defaults to physical as unix grain
- **ISSUE** #50073: (jorotenev) boto_cloudfront update distribution fails (refs: #50082)
- **PR** #50082: (garethgreenaway) [2018.3] Fixing a typo in the boto_cloudfront @ *2018-10-17 19:35:00 UTC*
  - d1f2c5d Merge pull request #50082 from garethgreenaway/50073_fixing_typo_boto_cloudfront
  - b5b90a1 Fixing a typo in the boto_cloudfront.
- **PR** #50090: (garethgreenaway) [2018.3] Fix to utils/vault.py @ *2018-10-17 13:57:30 UTC*
  - 05bd2a0 Merge pull request #50090 from garethgreenaway/fix_vault_sdb_runner_test
  - 444a4e5 Merge branch '2018.3' into fix_vault_sdb_runner_test
- **ISSUE** #49275: (dgengtek) execution module cp.cache_file fails with error 'invalid arguments to setopt' (refs: #50080)
- **PR** #50080: (terminalmage) Ensure all string kwargs passed to Tornado's httpclient are str types @ *2018-10-16 21:01:41 UTC*
  - 7092879 Merge pull request #50080 from terminalmage/issue49275
  - 4a6c327 Ensure all string kwargs passed to Tornado's httpclient are str types
- **ISSUE** #44979: (hunkeelin) Firewalld state cause horrid performance during saltrun. (refs: #49811)
- **PR** #49811: (nhavens) Optimize firewalld state @ *2018-10-16 11:26:46 UTC*
  - 45b6da3 Merge pull request #49811 from nhavens/optimize-firewalld-state

  - 7d6e628 Merge branch '2018.3' into optimize-firewalld-state
  - a718b0c Merge branch '2018.3' into optimize-firewalld-state
  - f7299b9 Merge branch '2018.3' into optimize-firewalld-state
  - f289618 Merge branch '2018.3' into optimize-firewalld-state
  - 41af513 Merge branch '2018.3' into optimize-firewalld-state
  - 7cebab4 Merge branch '2018.3' into optimize-firewalld-state
  - c38bb7d Merge branch '2018.3' into optimize-firewalld-state
  - abed058 Merge branch '2018.3' into optimize-firewalld-state
  - 0ec1f82 Merge branch '2018.3' into optimize-firewalld-state
  - db6b23b Merge branch '2018.3' into optimize-firewalld-state
  - 4ca7e12 Merge branch '2018.3' into optimize-firewalld-state
  - 4f7c914 fix invalid icmp type handling in firewalld state
  - 2d2e2eb resolve lint error in firewalld state
  - 081424d resolve indentation issues in updates to firewalld state
  - c92c4fe optimize firewalld.present rich rule handling
  - 25d1daf optimize firewalld.present source handling
  - 5f62c14 optimize firewalld.present interface handling
  - 69f2c6c optimize firewalld.present service handling
  - 71ec8e7 optimize firewalld.present port forward handling
  - f18e09e optimize firewalld.present open port handling
  - 35193dc clean up firewalld.present masquerade code (DRY)
  - 8c88784 optimize firewalld.present icmp block handling

- **PR** #50055: (rallytime) Back-port #50024 to 2018.3 @ *2018-10-16 09:14:06 UTC*
  - **PR** #50024: (rwaweber) docs: Correct napalm topic spelling mistake and smooth out sentence (refs: #50055)
  - b94e0a0 Merge pull request #50055 from rallytime/bp-50024
  - 1ea72a0 docs: Correct spelling mistake and smooth out sentence
- **PR** #50056: (rallytime) Back-port #50029 to 2018.3 @ *2018-10-16 09:12:59 UTC*
  - **PR** #50029: (thebluesnevrdie) Expose docs for Ansible modules (refs: #50056)
  - 14f3fac Merge pull request #50056 from rallytime/bp-50029
  - 2e46a78 Expose docs for Ansible modules
- **PR** #50064: (rallytime) Back-port #50022 to 2018.3 @ *2018-10-16 09:09:47 UTC*
  - **PR** #50022: (bluesliverx) Make test check_pillar more lenient (refs: #50064)
  - f3fe4b6 Merge pull request #50064 from rallytime/bp-50022
  - 40da3ce Make test check_pillar more lenient
- **ISSUE** #49988: (whytewolf) currently http error logging of urls contains full URL including passwords. (refs: #50066)

- **PR** #50066: (garethgreenaway) [2018.3] Redact any passwords from source paths @ *2018-10-16 09:08:25 UTC*
  - c2a46ce Merge pull request #50066 from garethgreenaway/49988_redact_passwords_in_source
  - 27ef1e0 Redact any passwords that are in http/https based source paths.
- **PR** #50068: (garethgreenaway) [2018.3] Updating salt.utils.mac_utils @ *2018-10-16 09:07:10 UTC*
  - ab2cb51 Merge pull request #50068 from garethgreenaway/1140_mac_system_tests_2018_3_failing
  - eedeacb Updating another reference to salt.utils.mac_utils to use __utils__
    - 6ef5ce4 Due to a previous PR the test_sdb_runner in sdb.test_vault was failing because of a exception that was being swallowed in the test run_run function. The cause was that when vault related functions were being run, if they were being run on the master then they were being forced to run through the _get_token_and_url_from_master() function, which is pull the id element out of the grains dictionary. When the call was taking place from a runner, the exception was popping up since there is no id when called from a runner. This fix checks to see if the id exists in the dictionary first, if it is there then _get_token_and_url_from_master() is called, otherwise _use_local_config is called.
- **PR** #49987: (terminalmage) Make Pillar no longer munge file_roots @ *2018-10-15 21:58:08 UTC*
  - 4c4bb5a Merge pull request #49987 from terminalmage/pillar_roots
  - f59506d Update tests to reflect behavior changes
  - 58f5fde Make Pillar no longer munge file_roots
- **PR** #50065: (rallytime) Back-port #50044 to 2018.3 @ *2018-10-15 21:32:37 UTC*
  - **PR** #50044: (mchugh19) add saltclass info from release notes to docs (refs: #50065)
  - 1e377ac Merge pull request #50065 from rallytime/bp-50044
  - 1eaaab7 Doc title formatting must match doc name length
  - 6ac26b6 add saltclass docs from release notes
- **PR** #50038: (s0undt3ch) Don't squash the traceback @ *2018-10-15 20:36:55 UTC*
  - 8a6b435 Merge pull request #50038 from s0undt3ch/2018.3
  - 852257a Don't squash the traceback
- **PR** #50054: (rallytime) Back-port #50004 to 2018.3 @ *2018-10-15 20:36:03 UTC*
  - **PR** #50004: (kiemlicz) https://github.com/saltstack/salt/issues/49883 (refs: #50054)
  - dbfd7ee Merge pull request #50054 from rallytime/bp-50004
  - 88e3202 https://github.com/saltstack/salt/issues/49883
- **PR** #50057: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-10-15 20:35:46 UTC*
  - a9e9974 Merge pull request #50057 from rallytime/merge-2018.3
  - 1b5b018 Merge branch '2017.7' into '2018.3'
    - 9eb17ea Merge pull request #50047 from rallytime/bp-49605
      - e3a0f7b Add gce fix to 2017.7.9 release notes
      - e6d86fc document GCE instance credentials configuration
      - c3a0d1b allow empty service_account_private_key in GCE driver
    - 6cdd390 Merge pull request #50017 from gtmanfred/iptables

      - e552e4f Use parse_known_args when checking iptables rules
    - a6a66b9 Merge pull request #50012 from dwoz/norm_log
      - f96b29d Use normal log formatting and add jid
    - 15f86b2 Merge pull request #50001 from kyentei/doc-fix-gpg
      - 3d6ee3e Fix gpg state Documentation
    - 7a235d7 Merge pull request #50000 from kyentei/doc-fix-lvm
      - c2f3d87 Fix LVM state documentation
    - 4edd094 Merge pull request #49999 from kyentei/doc-fix-proxy
      - a180ed7 Fix proxy documentation
    - 58d2302 Merge pull request #49998 from kyentei/doc-fix-process
      - edd4f40 Fix documentation on process.py
    - 411635b Merge pull request #49997 from kyentei/doc-fix-mysql
      - 01059cc Fix documentation for mysql processlist

- **PR** #50060: (rallytime) [2018.3] Fix some pylint issues that have popped up recently @ *2018-10-15 20:35:22 UTC*
  - f1bcb1f Merge pull request #50060 from rallytime/fix-lint
  - a8f4f39 Merge branch '2018.3' into fix-lint
- **ISSUE** #49927: (thetaurean) File Tree external pillar fails to render templates in python 3 (refs: #49943)
- **PR** #50052: (rallytime) Back-port #49943 to 2018.3 @ *2018-10-15 20:06:20 UTC*
  - **PR** #49943: (thetaurean) Coerce string when calling compile_template_str (refs: #50052)
  - b26b21c Merge pull request #50052 from rallytime/bp-49943
  - d336dec Handle non-ASCII chars during template data decode
  - 435afb7 Adjusted import to expose stringutils namespace
  - 958c630 Coerce string when calling compile_template_str
  - e73f13d [2018.3] Fix some pylint issues that have popped up recently
- **PR** #49809: (Ch3LL) Add additional asserts to help investigate test_win_pkg failure @ *2018-10-13 18:00:09 UTC*
  - fef7669 Merge pull request #49809 from Ch3LL/win_pkg_flaky
  - f795b60 Merge pull request #7 from dwoz/win_pkg_flaky
    - beaab2a Fix win_pkg test
  - 75809b4 Merge pull request #6 from dwoz/win_pkg_flaky
    - e5ea3f1 Try running fsync to avoid fs caching race condition
  - fd5e842 Merge pull request #5 from dwoz/win_pkg_flaky
    - c386826 Log results of cache_dir for now
  - 2cd4058 Merge pull request #4 from dwoz/win_pkg_flaky
    - 1005b91 Debug win_pkg
  - 9dce5cc Fix the assertion message order
  - 18e9a28 Add additional asserts to help investigate test_win_pkg failure

- **PR** #49983: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-10-11 13:27:34 UTC*
  - 8949873 Merge pull request #49983 from rallytime/merge-2018.3
  - f8e73e8 Merge branch '2017.7' into '2018.3'
    - a8a3811 Merge pull request #49972 from rallytime/release-notes-2017.7.9
      - 6c00003 Add 2017.7.9 release notes file
    - 8de58d6 Merge pull request #49974 from rallytime/bp-49940
      - a70358a Update mod_watch() description for all states.
      - bf3d064 Another attempt to clean up service.mod_watch description.
      - 165b799 Clean up documentation/language in service state. (Fixes #40819)
    - d68e5d3 Merge pull request #49978 from whytewolf/inotify_doc_change
      - b4d9523 using an exclude list that does not list every operating system leaves open the possability that it would work on those not listed
    - 89cbbb6 Merge pull request #49934 from rallytime/fix-49925
      - 216fd64 Remove event_publisher_pub_hwm and salt_event_pub_hwm from documentation
- **PR** #49979: (cachedout) Update license date and include NOTICE @ *2018-10-11 12:57:21 UTC*
  - c7d2482 Merge pull request #49979 from cachedout/license_update
  - b4f148e Break out support and contributing docs into separate files
  - 1dd0e34 Update license date and include NOTICE
- **PR** #49973: (rallytime) Add 2018.3.4 release notes file @ *2018-10-10 17:46:04 UTC*
  - ef61fc6 Merge pull request #49973 from rallytime/release-notes-2018.3.4
  - 32e7d90 Add 2018.3.4 release notes file
- **PR** #49975: (rallytime) Back-port #49958 to 2018.3 @ *2018-10-10 17:44:56 UTC*
  - **PR** #49958: (garethgreenaway) [fluorine] Fixes to mac_system.py (refs: #49975)
  - 7d8f10c Merge pull request #49975 from rallytime/bp-49958
  - a229a50 Swap the call for mac_utils.available_services to use __utils__ so mac_utils.available_services has access to __context__
- **PR** #49946: (isbm) osversion/osversion_info/osmajorrelease grains fixes @ *2018-10-10 14:01:05 UTC*
  - c912d3a Merge pull request #49946 from isbm/isbm-osversion_info-fixes
  - a17ba35 Fix proper part name in the string-bound CPE
  - 9d49607 Expand unit test to verify part name
  - a6b7d01 Remove linebreak
  - 5376886 Keep CPE_NAME only for opensuse series
  - 8af3904 Add part parsing
  - e202b41 Prevent possible crash if CPE_NAME is wrongly written in the distro
  - 712083d Add unit test for broken CPE_NAME
  - 1306ddb Add unit test for v2.3 of CPE format
  - 04b61d3 Add unit test for WFN format of CPE_NAME

  - 035690f Override VERSION_ID from os-release, if CPE_NAME is given
  - 08e235b Remove unnecessary linebreak
  - 853e163 Add CPE_NAME parsing
- **ISSUE** #48204: (zerthimon) State mysql_user.present Exception with mysql 8.0.11 (refs: #49918)
- **PR** #49918: (garethgreenaway) [2018.3] Fixes to MySQL module @ *2018-10-10 14:00:11 UTC*
  - 18fb5df Merge pull request #49918 from garethgreenaway/48204_mysql_user_password_exception_part_deux
  - 92c0680 Merge branch '2018.3' into 48204_mysql_user_password_exception_part_deux
  - 71e53a3 Cleanup of debugging.
  - dd96c13 Fixing lint.
  - ac7da89 Tweaking a couple tests to account for new scenarios with later versions.
  - 4320c43 Removing __mysql_hash_password, no longer needed.
  - 0475acf Removing unnecessary empty line.
  - 9265195 Updating the mysql update to handle changes in version 8.0.11 where PASSWORD was removed. Updating unit tests.
- **PR** #49961: (dmurphy18) Fix aix grains test for prtconf present @ *2018-10-10 13:38:02 UTC*
  - 1c2e5be Merge pull request #49961 from dmurphy18/fix_aix_grains
  - f38d543 Correct test for AIX grains prtconf
- **ISSUE** #49205: (ereslibre) Compound targeting includes unexpected targets in the result (refs: #49897, #49907)
- **PR** #49907: (brejoc) 2018.3 fix for list compound targeting @ *2018-10-10 13:33:40 UTC*
  - **PR** #49897: (brejoc) 2017.7 fix for list compound targeting (refs: #49907)
  - **PR** #49435: (brejoc) No longer passes missing in list compound engine (refs: #49897, #49907)
  - e427306 Merge pull request #49907 from brejoc/2018.3-fix-for-list-compound-targeting
  - 1bcadb9 Merge branch '2018.3' into 2018.3-fix-for-list-compound-targeting
  - 649c46c Adds check for engine before adding the additional engine argument
  - 7d0b20c Minor fix for def test_batch_run_grains_targeting
  - 0d64716 Ignore missing minions only when excluding them with 'not'
  - 6a5f3cd No longer passes missing in list compound engine
- **ISSUE** #48599: (angeloudy) module ldap3 'TypeError: ('expected a byte string in the list" (refs: #48666)
- **PR** #49947: (rallytime) Back-port #48666 to 2018.3 @ *2018-10-09 17:48:20 UTC*
  - **PR** #48666: (angeloudy) values need to be bytes when writing back to ldap (refs: #49947)
  - 143c723 Merge pull request #49947 from rallytime/bp-48666
  - 5166917 Update ldap3.py
  - 05e55bd values need to be bytes when writing to ldap
  - 0d749e6 values need to be bytes when writing back to ldap
- **PR** #49948: (rallytime) Skip pillar refresh test @ *2018-10-09 15:20:52 UTC*
  - dbf994e Merge pull request #49948 from rallytime/skip-flaky-test

  - df11c39 Skip pillar refresh test

- **PR** #49938: (s0undt3ch) Handle missing *pkg_resources* package @ *2018-10-09 13:41:09 UTC*
  - 614d172 Merge pull request #49938 from s0undt3ch/2018.3
  - b1c2928 Handle missing *pkg_resources* package
- **PR** #49942: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-10-09 13:38:04 UTC*
  - bce12c3 Merge pull request #49942 from rallytime/merge-2018.3
  - 91534ee Merge branch '2017.7' into '2018.3'
    - 327796e Merge pull request #49911 from dubb-b/slack-pr
      - 57e630c Adding failure messages for PR.
    - cf1df79 Merge pull request #49900 from dwoz/presence_wart
      - 7bd34f6 Connect before processing presence
    - 8c4707d Merge pull request #49899 from twangboy/fix_49892
      - a66765f Remove arrInstalled artifact
- **PR** #49930: (isbm) Get os_family for RPM distros from the RPM macros @ *2018-10-09 13:31:52 UTC*
  - a30c053 Merge pull request #49930 from isbm/isbm-osarch-bugfix
  - f8a75f1 Fix imports
  - 70d3de0 Strip and stringify the return for the osarch
  - a248371 Get os_family for RPM distros from the RPM macros.
- **PR** #49868: (dwoz) windows shell tests @ *2018-10-08 17:32:20 UTC*
  - 99c5dba Merge pull request #49868 from dwoz/shelltests
  - 9997f45 Merge branch '2018.3' into shelltests
- **PR** #49875: (dwoz) Add spm tests to Windows suite @ *2018-10-05 18:13:07 UTC*
  - eee82d3 Merge pull request #49875 from dwoz/win_spm_tests
  - e76a751 Use os.path.split for more consistancy
  - bc54d15 Fix wart in spm on windows
  - df5e271 Add spm tests for windows
- **PR** #49872: (twangboy) Fix issues with windows file permissions when using reset=True @ *2018-10-05 13:52:07 UTC*
  - c0e2404 Merge pull request #49872 from twangboy/fix_49861
  - 01bd847 Gate the win_function a little better
  - bcdb366 Skip on non-windows systems
  - 02f6335 Fix some lint
  - 8ecbe0e Add tests for check_perms
  - d6e7512 Fix issues with file permissions
- **ISSUE** #49582: (UtahDave) lgpo doesn't seem to allow for unsetting a gpo setting (refs: #49902)
- **ISSUE** #48661: (mike2523) win_lgpo: User Rights Assignment policies (refs: #49902)

- **PR** #49902: (lomeroe) lgpo better comments on user right assignments @ *2018-10-05 13:41:16 UTC*
  - d3a909f Merge pull request #49902 from lomeroe/issue48661_2018.3
  - 760d8f4 Merge branch '2018.3' into issue48661_2018.3
  - f72ca81 add comments on user right assignment policies when they are already set and note if cumulative_rights_assignment is True that the particular user is already granted the right
    - 75dd5ab Merge branch '2018.3' into shelltests
- **PR** #49896: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-10-04 21:00:24 UTC*
  - 4eb43af Merge pull request #49896 from rallytime/merge-2018.3
  - 41c8b31 Merge branch '2017.7' into '2018.3'
  - b959033 Merge pull request #49739 from twangboy/fix_49660
    - f3400ad Merge branch '2017.7' into fix_49660
    - 0a4f5fc Merge branch '2017.7' into fix_49660
    - 1f9bbc8 Populate changes dictionary
  - 71669bd Merge pull request #49885 from dwoz/gemfile
    - 791e3ff Use dwoz/winrm-fs for chunked downloads
    - f3999e1 Move vagrant to it's own group
  - 0662e37 Merge pull request #49870 from KaiSforza/ci_actually_fail
    - 9ef9206 Actually catch the exception when we fail
      - 19072f0 Use os.path.split for more consistancy
      - 6c22459 Merge remote-tracking branch 'origin/shelltests' into shelltests
      - 43b292f Merge branch '2018.3' into shelltests
      - 876c040 Fix wart in test_cp module
      - 9e6ff11 Add syndic tests to whitelist.txt for Windows
      - 9cab293 Fix wart in spm on windows
      - 58e9d1b Add spm tests to whitelist.txt for Windows
      - 7393610 Add more shell integration tests to whitelist
      - ce85d60 Add shell.test_key to whitelist and fix wart
      - 6fbadd7 Skip tests not valid on windows
      - 9debf29 Add shell cp tests and fix them
      - 8bfa634 Fix warts in integration.shell.test_matcher
      - 45d2142 Fix wart in test_cp module
      - 0e4fbc5 Add syndic tests to whitelist.txt for Windows
      - f72ad01 Fix wart in spm on windows
      - 4141cc8 Add spm tests to whitelist.txt for Windows
      - 02ad722 Add more shell integration tests to whitelist
      - 48275c0 Add shell.test_key to whitelist and fix wart

      - e5edff4 Skip tests not valid on windows
      - a389729 Add shell cp tests and fix them
- **ISSUE** #49744: (MTecknology) Group state cannot force empty group membership (refs: #49745)
- **PR** #49745: (MTecknology) Remove all users from group when members list is empty. @ *2018-10-03 13:48:40 UTC*
  - 85478cd Merge pull request #49745 from MTecknology/fix-49744
  - 61afb5d Merge branch '2018.3' into fix-49744
  - 269bde2 Merge branch '2018.3' into fix-49744
  - 0b5d916 Modify if structure, as requested.
  - f57ce67 Remove all users from group when members list is empty. (Fixes #49744)
- **PR** #49867: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-10-03 13:23:35 UTC*
  - 05cd56b Merge pull request #49867 from rallytime/merge-2018.3
  - b53c3e6 Update old utils paths to use new utils paths
  - a65c60d Merge branch '2017.7' into '2018.3'
    - 7fa2043 Merge pull request #49858 from Paulo-Nunes/fix-requisite-doc
      - fb6f818 Edit requisite documentation
    - 0587fbc Merge pull request #49828 from dwoz/testslogencoding
      - 2b323df Merge branch '2017.7' into testslogencoding
      - 7ce1b64 Use utf-8 encoding for salt-runtests.log
    - 15ccffd Merge pull request #49848 from terminalmage/fix-version-rst-2017.7
      - af2dddb Fix badly formatted versionadded directive
    - 297031b Merge pull request #49777 from terminalmage/issue49738
      - 8a9945e Update test docstring to reflect switch to using SHA1
      - 7ddfd72 Switch from base64 encoding to sha1 digest for unique cache filename
      - 0e26dc6 Explicitly import salt.utils.hashutils
      - dca90d7 Skip parallel states test on Windows until we can get it to work
      - 5a11067 Add additional info to the assert to aid in troubleshooting
      - cb19086 Remove/replace unnecessary setUp/tearDown
      - b3e9678 Fix parallel states with long ID dec or name
    - 98c9372 Merge pull request #49763 from twangboy/fix_49730
      - 99bc8f1 Merge branch '2017.7' into fix_49730
      - 7a9278b Fix broken tests, add new tests
      - 2cd1509 Merge branch '2017.7' into fix_49730
      - defba1c Merge branch '2017.7' into fix_49730
      - af1ed1c Add/fix tests
      - 5005a4d Only clear dns entries if you pass []

      - 048561e Add ability to clear DNS entries on static DNS

- **PR** #49876: (KaiSforza) Actually catch the exception when we fail @ *2018-10-03 12:26:58 UTC*
  - 42b540b Merge pull request #49876 from KaiSforza/ci_actually_fail_2018
  - c88a81d Actually catch the exception when we fail
- **PR** #49852: (terminalmage) Rename logstash engine filename to make it work with the loader @ *2018-10-02 17:42:10 UTC*
  - 1c8def9 Merge pull request #49852 from terminalmage/issue49627
  - 559cf1b Rename logstash engine filename to make it work with the loader
- **PR** #49850: (rallytime) Back-port #49798 to 2018.3 @ *2018-10-02 17:40:29 UTC*
  - **PR** #49798: (mattp-) saltnado: teach runner about full_return (refs: #49850)
  - 675df25 Merge pull request #49850 from rallytime/bp-49798
  - 4b6dcd7 saltnado: teach runner about full_return
- **ISSUE** #49204: (mchugh19) file.directory state fails if directory contains an invalid symlink (refs: #49209, #49827)
- **PR** #49846: (rallytime) Back-port #49650 and #49827 to 2018.3 @ *2018-10-01 20:05:24 UTC*
  - **PR** #49827: (dgmorrisjr) fixing mis-spelling of lattrs in file.py, referncing #49204 (refs: #49846)
  - **PR** #49650: (Yxnt) fix aliyun cloud typeerror (refs: #49846)
  - 93d064a Merge pull request #49846 from rallytime/bp-49650
  - fb7fed7 referncing #49204, fixing mis-spelling of lattrs on line 4514 per request from @gtmanfred
  - ec9fa92 use stringutils instead of hard code
  - a307ae0 fix aliyun cloud typeerror
- **ISSUE** #49520: (doesitblend) Multi-master DNS Issue - minion connect fail (refs: #49764)
- **PR** #49764: (garethgreenaway) [2018.3] Adding retry_dns_count to minion @ *2018-10-01 14:20:03 UTC*
  - ec97806 Merge pull request #49764 from garethgreenaway/49520_multimaster_dns_issue_fix
  - 30ccc5e Merge branch '2018.3' into 49520_multimaster_dns_issue_fix
  - 741928b Fixes per request from @rallytime.
  - 04c5983 Moving the check for an empty self.minions into a io_loop.call_later call.
  - 44ee2ec Updating default for retry_dns_count
  - 75f64a6 Removing unneccesary code.
  - e66dc18 Updating the resolve_dns function in minion.py to include a new minion configuration option which will control how many attempts will be made when the master hostname is unable to be resolved before giving up.
- **PR** #49847: (terminalmage) Fix badly formatted versionadded directive (2018.3 branch) @ *2018-10-01 14:04:27 UTC*
  - abdab68 Merge pull request #49847 from terminalmage/fix-version-rst-2018.3
  - a127548 Fix badly formatted versionadded directive
- **ISSUE** #49043: (awerner) Unicode broken with file.blockreplace on Python2 (refs: #49782)
- **PR** #49782: (dwoz) Fix issue 49043 @ *2018-10-01 13:43:02 UTC*

  - aca87ab Merge pull request #49782 from dwoz/issue-49043
  - b7d904f Merge branch '2018.3' into issue-49043
  - ed35633 Revert diff newline change
  - 2a901e9 Fix issue 49043
- **ISSUE** #49671: (mchugh19) Vault execution module broken in pillar lookups (refs: #49820)
- **PR** #49820: (garethgreenaway) [2018.3] Fixing vault when being used from Pillar @ *2018-10-01 13:40:25 UTC*
  - 1a41d1b Merge pull request #49820 from garethgreenaway/49671_fixing_vault_pillar
  - a57872f Merge branch '2018.3' into 49671_fixing_vault_pillar
  - 5f13219 Merge branch '2018.3' into 49671_fixing_vault_pillar
  - b0ba2ec Fixing the scenario when vault values are used in Pillar, but due to a previous change the minion was not being granted token based access.
- **PR** #49842: (rallytime) [2018.3] Merge forward from 2018.3.3 to 2018.3 @ *2018-10-01 13:21:42 UTC*
  - bf9c175 Merge pull request #49842 from rallytime/merge-2018.3
  - 81b0aa7 Add "in progress" notation back to the 2018.3.3 release notes
  - 0e76103 Merge branch '2018.3.3' into '2018.3'
- **PR** #49839: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-09-30 16:32:53 UTC*
  - e6382cb Merge pull request #49839 from rallytime/merge-2018.3
  - 179e4b1 Merge branch '2017.7' into '2018.3'
  - fd46d35 Merge pull request #49823 from dubb-b/timeout_addition
    - b25ac9f Adding some changes for the status
    - f6d9679 Updating syntax to be better and with new lines
    - 855ffe6 Adding timeout to all pipelines so that the build aborts
- **PR** #49815: (isbm) Bugfix/rework: IPv6 scope errors (bp) @ *2018-09-28 19:44:59 UTC*
  - fccc08c Merge pull request #49815 from isbm/isbm-ipv6-scope-errors-2018.3
  - c14f26f Reverse skipping tests: if no ipaddress
  - bb37ba0 Add IPv6 scoped test
  - facb34a Add additional check
  - 615b09a Lintfix: W0611
  - 3ce265e Remove unnecessary manipulation with IPv6 scope outside of the IPv6Address object instance
  - c314e1c Revert method remapping with pylint updates
  - 0a86112 Lintfix refactor: remove duplicate returns as not needed
  - 5932a0d Lintfix
  - f52ea0b Add Py3.4 old implementation's fix
  - 1c02a31 Fix wrong type swap
  - 54233ca Do not use introspection for method swap
  - 12038ea Simplify checking clause

  - 7f2208b Fix py2 case where the same class cannot initialise itself on Python2 via super.
  - cefb16b Add extra detection for hexadecimal packed bytes on Python2. This cannot be detected with type comparison, because bytes == str and at the same time bytes != str if compatibility is not around
  - 379ead4 Lintfix: mute not called constructors
  - 66ec29c Add scope on str
  - f5644bf Add logging to the ip_interface
  - 8bb5438 Add real exception message
  - 693e4d4 Move docstrings to their native places
  - 77f1c44 Remove duplicated code
  - 40d0e6b Use ternary operator instead
  - d05999e Remove multiple returns
  - 3fde850 Remove unnecessary operator
  - deb0b4e Remove duplicated code
  - 8e62633 Remove unnecessary variable for import detection
  - 7a6ea3b Remove multiple returns and add check for address syntax
  - b40e1e1 Add debugging to the ip_address method (py2 and py3)
  - cab5f62 Add logging
  - eeab9f2 Isolate Py2 and Py3 mode
  - 49b32b7 Check version via object
  - 1ae94f1 Override standard IPv6Address class
  - 5e970ea Fix unicode imports in compat
  - bbeec4d Fix ipaddress import
  - 561c99e Remove unused import
  - 0498e94 Fix ipaddress imports
  - b4f1a72 Add missing docstrings
- **PR** #49784: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-09-28 17:29:30 UTC*
  - 804d52c Merge pull request #49784 from rallytime/merge-2018.3
  - c6b9d18 Merge branch '2018.3' into merge-2018.3
  - 26461f9 Add helper import comments
  - 22f055c Merge branch '2017.7' into '2018.3'
    - d2a193b Merge pull request #49792 from KaiSforza/fixscriptedpipes
      - 5b16996 Fix the new pipelines
  - dd27d80 Merge branch '2017.7' into '2018.3'
    - 8a71a43 Merge pull request #49774 from KaiSforza/declarativeToScripted
      - cae88f2 Move all pipelines to be fully scripted
    - 8fe8aac Merge pull request #49736 from mattLLVW/fix-file-touch

      - 1da9f34 Fix test opts in append, prepend states/file.py

- **PR** #49790: (weswhet) fixing an issue where sentry logger would fail to get tags from grain… @ *2018-09-28 13:39:04 UTC*
  - d191b08 Merge pull request #49790 from weswhet/fixing-sentry-logger-2018.3
  - 201697d fixing an issue where sentry logger would fail to get tags from grains and some other lint fixes for this file
- **ISSUE** #49626: (PabloLemos) target grains ip_interface no return received (refs: #49791)
- **PR** #49791: (terminalmage) Fix 3 bugs in subdict matching @ *2018-09-28 13:38:34 UTC*
  - 4d9fc7c Merge pull request #49791 from terminalmage/issue49626
  - 6795472 Add clarifying comment
  - ac0f800 Fix 3 bugs in subdict matching
- **PR** #49806: (isbm) Bugfix: zypper ZYPPER_EXIT_NO_REPOS exit code @ *2018-09-28 13:20:36 UTC*
  - 52b0472 Merge pull request #49806 from isbm/isbm-zypper-errcode-6-bp
  - 756ef77 Update error list for zypper
  - 5b02548 Add error logging
- **ISSUE** saltstack/salt-jenkins#1121: (rallytime) [2018.3] 4 related tests failing on 2018.3 Python 3 (refs: #49795)
- **PR** #49795: (gtmanfred) fix test_managed_file_with_grains_data @ *2018-09-26 22:48:36 UTC*
  - fee6701 Merge pull request #49795 from gtmanfred/tests
  - cf5c179 fix test_managed_file_with_grains_data
- **PR** #49786: (rallytime) Back-port #49759 to 2018.3 @ *2018-09-26 18:22:01 UTC*
  - **PR** #49759: (mattp-) runners.state.orch: generate jid if missing (refs: #49786)
  - 04e5869 Merge pull request #49786 from rallytime/bp-49759
  - df3791c state.orch: generate jid if missing
- **PR** #49770: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-09-25 19:03:25 UTC*
  - 4138683 Merge pull request #49770 from rallytime/merge-2018.3
  - d9ed59b Merge branch '2017.7' into '2018.3'
  - f664515 Merge pull request #49760 from Ch3LL/raid_docs
    - 0eb90c1 Add mdadm as dependency in mdadm state docs
  - d5a75bf Merge pull request #49749 from erwindon/docufix
    - baee678 fixed link in documentation
  - 5ef2def Merge pull request #49703 from twangboy/fix_49675_2017.7
    - 9127222 Merge branch '2017.7' into fix_49675_2017.7
    - 4f54422 Fix reboot needed detection
  - b5726d2 Merge pull request #49718 from vutny/doc/2017.7/add-missing-mock-kwarg
    - 3e19176 [DOC] Add missing *mock* argument for *state.apply*
- **PR** #49757: (bbinet) Fix influxdb_user state when test=True @ *2018-09-24 19:14:22 UTC*
  - db6f68a Merge pull request #49757 from bbinet/fix-influxdb_user-test

  - 0d9ef4b Fix influxdb_user state when test=True
- **PR** #49679: (dwoz) Add more tests to whitelist @ *2018-09-22 17:46:50 UTC*
  - ec2386e Merge pull request #49679 from dwoz/whitelist
  - 0a3d95f Fix linter issues
  - 4c457c9 Close handles only if they exist
  - a44b37a Merge remote-tracking branch 'saltstack/2018.3' into whitelist
- **PR** #49722: (meaksh) Fixes some issues on the Zypper module @ *2018-09-21 13:38:03 UTC*
  - 6a3e127 Merge pull request #49722 from meaksh/2018.3-fix-zypper-issue-on-python3
  - bb9cb6e Fix wrong queryformat for zypper list_provides
  - d40c034 Fix index error when running on Python 3
- **PR** #49720: (cstarke) Seperate prlctl and prlsrvctl checks into each requiring function @ *2018-09-21 13:32:46 UTC*
  - e7bbb83 Merge pull request #49720 from cstarke/2018.3
  - b0be6aa Add documentation for prlctl and prlsrvctl binary requirement
  - 4f8476b Fix tests; add test for CommandExecutionError
  - 3c96dd2 Fix tests for parallels module
  - 9034c4e Add import for CommandExecutionError
  - 752b6f8 Seperate prlctl and prlsrvctl checks into each requiring function
- **PR** #49615: (terminalmage) ping_interval: use service.restart instead of signaling @ *2018-09-21 13:26:41 UTC*
  - f745e43 Merge pull request #49615 from terminalmage/ping-interval
  - b49b018 Move service name logic to a separate function
  - 2fddba3 ping_interval: use service.restart instead of signaling
- **PR** #49731: (terminalmage) Add warning about using jinja filters to dump dicts in Jinja @ *2018-09-21 13:16:04 UTC*
  - 0a530d2 Merge pull request #49731 from terminalmage/jinja-filter-warning
  - 8393560 Add warning about using jinja filters to dump dicts in Jinja
- **PR** #49696: (jgleissner) loosen azure sdk dependencies in azurearm cloud driver @ *2018-09-20 19:17:15 UTC*
  - 91acb82 Merge pull request #49696 from jgleissner/2018.3-azurearm-deps
  - 467e017 remove unused import from azurearm driver
  - bad28b6 loosen azure sdk dependencies in azurearm cloud driver
- **PR** #49684: (twangboy) Detect when system needs reboot after domain join @ *2018-09-20 16:17:50 UTC*
  - d434e36 Merge pull request #49684 from twangboy/fix_49675
  - 3600467 Fix some try/except blocks, mark a lazy test
  - 871c11a Fix other functions in the same manner
  - 1b5bc66 Merge branch '2018.3' into fix_49675
  - 95e6f54 Detect when system needs reboot after domain join
    - 9048970 Add some debug info to assert

        - 536fce6 Increase function timeout
        - ed4de63 Fix file serialize test
        - 5ac3738 Merge remote-tracking branch 'saltstack/2018.3' into whitelist
- **PR** #49710: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-09-19 18:05:50 UTC*
    - 936cae5 Merge pull request #49710 from rallytime/merge-2018.3
    - d19c67f Merge branch '2017.7' into '2018.3'
    - 6bd85b5 Merge pull request #49706 from garethgreenaway/1120_remove_mock_os_path_join
        - f4ccc75 Removing mocking of os.path.join from two tests, one of which was failing when used with coverage argument. No reason we should be mocking os.path.join.
    - c38dc61 Merge pull request #49692 from rallytime/bp-49688
        - fde7575 Order the runtests suites
            - 1f60753 Do not use close_fds on Windows OS
            - 4a54c7c Fix file serialize test
            - 0ec1e1e Increase timeout on flaky test
            - 6ae924b Skip test that is not applicable on windows
            - 5980200 Increase timeout on flaky test
            - 1757edd Increase ModuleTest.run_function timeout
            - 7c7353c Add more tests to whitelist
- **PR** #49708: (rallytime) Back-port #49555 to 2018.3 @ *2018-09-19 13:12:12 UTC*
    - **PR** #49555: (bergmannf) Change StringIO and BytesIO imports (refs: #49708)
    - b082f79 Merge pull request #49708 from rallytime/bp-49555
    - ed25416 Change StringIO import in python2 to import the class.
- **PR** #49707: (rallytime) Back-port #49702 to 2018.3 @ *2018-09-19 13:09:10 UTC*
    - **PR** #49702: (rallytime) Mark orchestration test as expensive (refs: #49707)
    - 0e97d9e Merge pull request #49707 from rallytime/bp-49702
    - 7ff24e5 Mark orchestration test as expensive
- **PR** #49693: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-09-18 20:05:28 UTC*
    - 0c4005d Merge pull request #49693 from rallytime/merge-2018.3
    - b39dacf Merge branch '2017.7' into '2018.3'
    - 37cbd27 Merge pull request #49685 from rallytime/pillar-headings
        - 02b2d54 Split intro docs for cmd_yamlex pillar into shorter lines.
        - c4cbe5b Fix heading line-lengths
    - 9bad295 Merge pull request #49689 from terminalmage/2017.7-docs
        - 7e5bab7 Fix docs build errors (2017.7 branch)
- **PR** #49674: (rallytime) [2018.3] Merge forward from 2018.3.3 to 2018.3 @ *2018-09-18 13:37:00 UTC*
    - a12176d Merge pull request #49674 from rallytime/merge-2018.3

  - 7bcb809 Merge pull request #31 from dwoz/merge-2018.3
    - cb35ebb Account for filename changes
  - f3b122f Merge branch '2018.3.3' into '2018.3'
- **PR** #49593: (dwoz) Add more tests to whitelist for Windows @ *2018-09-17 14:57:44 UTC*
  - e636dac Merge pull request #49593 from dwoz/add_whitelist
  - 2e1bf13 Merge branch '2018.3' into add_whitelist
  - 2340a58 Fix runner returns test on Windows
  - f908cde Merge branch '2018.3' into add_whitelist
  - be5fccf Fix linter - blank lines
  - e51b0f8 Fix binary file manage test
  - 9e3b44d Fix linter issues
  - ca1b05e Fix multiple issues in x509 module and state
  - e603e7a Fix encoding issues in file and x509 states
  - 7f93c7f Fix map file loading on windows
  - 2959da0 Add some runners test files to whitelist
  - bee2902 Make windows file remove return like others
  - 5770b05 Fix hitelist file order
  - 10ec8be Fix linter issues
  - ccbcb55 Add more tests to whitelist for Windows
- **PR** #49639: (isbm) Retire MD5 checksum for pkg mgmt plugins @ *2018-09-17 14:51:54 UTC*
  - 7823839 Merge pull request #49639 from isbm/isbm-yumnotify-md5
  - 11a94ca Remove an empty line
  - adb8324 Use SHA256 algorithm for zyppnotify plugin
  - 27001cf Use SHA256 algorithm for yumnotify plugin
- **PR** #49652: (rallytime) [2018.3] Merge forwrad from 2018.3.3 to 2018.3 @ *2018-09-14 21:20:39 UTC*
  - aecbe12 Merge pull request #49652 from rallytime/merge-2018.3
  - a8a3820 Merge branch '2018.3.3' into '2018.3'
- **PR** #49646: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-09-14 13:30:22 UTC*
  - 530e281 Merge pull request #49646 from rallytime/merge-2018.3
  - 186f57e Merge branch '2017.7' into '2018.3'
  - 3c8ec8a Merge pull request #49644 from rallytime/mark-flaky-test
    - be8a187 Mark grains test as flaky
  - 8ac66f1 Merge pull request #49643 from rallytime/bp-49642
    - accef8c Skip some unreliable tests
  - 7a75d5e Merge pull request #49597 from rallytime/bp-47232
    - c477b70 Fixed usage of ipaddress

  - 2f6cdd9 Merge pull request #49614 from rallytime/bp-49546
    - 0f3881a Skip flaky test on Python 3
- **PR** #49642: (rallytime) Skip some unreliable tests (refs: #49643) @ *2018-09-13 18:24:43 UTC*
  - bb1dcdf Merge pull request #49642 from rallytime/flaky-tests
  - 8e74d1c Skip some unreliable tests
- **PR** #49629: (rallytime) [2018.3] Merge forward from 2018.3.3 to 2018.3 @ *2018-09-13 14:00:08 UTC*
  - e319ecc Merge pull request #49629 from rallytime/merge-2018.3
  - b628ad7 Merge branch '2018.3.3' into '2018.3'
- **PR** #49632: (garethgreenaway) [2018.3] Fixing integration.states.test_file.FileTest.test_directory_max_depth @ *2018-09-13 13:51:56 UTC*
  - 93d43fa Merge pull request #49632 from garethgreenaway/fixing_failing_test_python_3_7
  - 588f744 Fixing failing test under python 3.7 causaed by changes to how os.makedirs sets initial permissions.
- **PR** #49580: (garethgreenaway) [2018.3] Fixing tests for Python 3.7 @ *2018-09-12 16:33:35 UTC*
  - 7042fc0 Merge pull request #49580 from garethgreenaway/fixing_failing_test_python_3_7
  - b6afdae Fixing lint.
  - d2fb68c Fixes various tests that were failing under python 3.7.
- **PR** #49606: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-09-12 14:26:33 UTC*
  - ab19082 Merge pull request #49606 from rallytime/merge-2018.3
  - c168268 Update old utils paths to use new paths
  - 19598d8 Merge branch '2017.7' into '2018.3'
    - aa560c5 Merge pull request #49595 from rallytime/merge-2017.7
      - 09247a3 Merge branch '2017.7.8' into '2017.7'
      - 4d865b4 Merge pull request #49560 from rallytime/pkgng-fix
      - f3e1534 [2017.7.8] Use correct path for is_true function
      - 92369f0 Merge pull request #49541 from twangboy/fix_osx_build_2
      - 750d3fa Remove trusted host switches
      - 397296f Merge pull request #49539 from twangboy/fix_osx_build_2
      - 6b42e73 Fix issues with build script on OSX
    - e8171de Merge pull request #49596 from terminalmage/kill-bare-excepts-with-fire
      - 729dc08 Kill bare excepts with fire
    - 63b6643 Merge pull request #49578 from twangboy/fix_49566
      - 6a5a69c Merge branch '2017.7' into fix_49566
    - f9a01b7 Merge pull request #49584 from terminalmage/squelch-varstack-error
      - c79a823 Squelch error logging when varstack is not installed
    - d4d2b36 Merge pull request #49579 from dwoz/flaky_test
      - dbe40df Increase timeout for flaky git test

        - 5cbe60b Merge pull request #49570 from rallytime/doc-fixes
            - de80bda Fix docstring + import order, move pylint disable to own line
            - 393bc0f Fix doc build errors in saltnado documentation
            - 6105ebb Use setup.py clean –all
- **PR** #49620: (dwoz) Adding passing windows nacl runner tests to whitelist @ *2018-09-12 14:19:11 UTC*
    - 0ce91d6 Merge pull request #49620 from dwoz/nacltest
    - 51500ac Adding passing windows nacl runner tests to whitelist
- **PR** #49532: (meaksh) Fix wrong 'recurse' behavior on for linux_acl.present/absent states @ *2018-09-12 14:16:15 UTC*
    - c1f97c4 Merge pull request #49532 from meaksh/2018.3-fix-linux_acl-recursive-problems
    - cf27069 Fix some pylint issues
    - a0ed59d Fix recursive cases on linux_acl.absent state
    - 4a01849 Add unit tests to cover recursive cases of linux_acl states
    - f739b6f Fix typo on variable name
    - dc6d1d7 Fix wrong recurse behavior on for linux_acl.present state
- **ISSUE** #49027: (lachlanmunro) x509.certificate_managed certificate is mangled on process under py3 m2crypto (refs: #49561)
- **ISSUE** #49008: (lachlanmunro) x509.certificate_managed state fails under py3 m2crypto with "int too large" (refs: #49561)
- **PR** #49561: (dwoz) x509 module and state bug fixes @ *2018-09-11 14:01:43 UTC*
    - 283e7d7 Merge pull request #49561 from dwoz/m2crypto_test
    - 5e591ea Merge branch '2018.3' into m2crypto_test
- **PR** #49592: (dwoz) Fix binary file manage test @ *2018-09-10 23:03:18 UTC*
    - 4478a44 Merge pull request #49592 from dwoz/bin_file_fix
    - 8556c3d Fix binary file manage test
- **PR** #49571: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-09-10 12:58:15 UTC*
    - 9004422 Merge pull request #49571 from rallytime/merge-2018.3
    - a00aba9 Merge branch '2017.7' into '2018.3'
        - 5d267a2 Merge pull request #49547 from rallytime/bp-49518
            - 4bb09f7 Skip service module integration tests on CentOS 7
        - 4948071 Merge pull request #49549 from rallytime/codeowners-duplicate
            - 4b257d2 Remove duplicate comment in CODEOWNERS file
            - faafe74 Merge branch '2018.3' into m2crypto_test
- **PR** #49577: (dwoz) Multiple unittest fixes for 2018.3 Python 3 Windows @ *2018-09-08 05:53:38 UTC*
    - 4669a9a Merge pull request #49577 from dwoz/unit_test_fixes
    - aa07b3f Multiple unittest fixes for 2018.3 Python 3 Windows
- **PR** #49533: (silenius) [cherry-pick] Optimize postgres groups management @ *2018-09-07 18:58:57 UTC*

  - 8b971e5 Merge pull request #49533 from silenius/2018.3.3
  - 7eb8d6c Postgress module unit test fix.
  - b56aedc Optimize postgres groups management
- **PR** #49543: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-09-07 18:19:32 UTC*
  - 1a41b47 Merge pull request #49543 from rallytime/merge-2018.3
  - cf508a4 Merge branch '2017.7' into '2018.3'
    - 75a3b52 Merge pull request #49527 from terminalmage/remove-special-envs-handling
      - 1478a76 Remove special handling for file_envs fileserver func
    - 13f1d26 Merge pull request #49512 from rallytime/merge-2017.7
      - de7ef3d Add "in progess" notation back to 2017.7.8 release notes
      - edad2dc Merge branch '2017.7.8' into '2017.7'
      - d24c96f Merge pull request #49505 from Ch3LL/rn_7.8_2
      - d40b644 Update 2017.7.8 ChangeLog with new fixes
      - 0d7b173 Merge pull request #49498 from rallytime/pin-cherrypy-2017.7.8
      - 81a6155 Pin CherryPy version to < 18.0.0 in requirements files for PY2
      - 52ab2c0 Merge pull request #49466 from rallytime/bp-49461
      - 5842800 Revert "Allow for not being prompted to supply a password to deploy keys to a…"
      - 1bbe7df Merge pull request #49284 from twangboy/fix_installer_osx
      - a112eaa Fix path to libsodium tarball
      - e0e8779 Merge pull request #49272 from twangboy/fix_installer_more
      - a8f054b Add 64bit binaries for KB2999226 to the x86 installer
      - 0eb6ddf Merge pull request #49218 from twangboy/fix_installer
      - e8a1d2f Add more descriptive error when KB not found
      - 71737ea Suppress all ui on vcredist installation
      - 7dae9bb Fix Windows and OSX installers
      - e484f26 Merge pull request #49116 from twangboy/fix_installer
      - 1227095 Add nonfatal switch to VCRedist
      - 0b9f2f8 Merge pull request #49113 from Ch3LL/rn_7.8
      - f6b70bb Add changelog to 2017.7.8 release notes
    - 208cfe6 Merge pull request #49515 from rallytime/update-codeowners
      - 1faab8b Add some files to the CODEOWNERS files for team-core to own
    - 34e5174 Merge pull request #49327 from twangboy/fix_win_service
      - 74e944b Mark test_service_status_running as flaky
      - c69560c Merge branch '2017.7' into fix_win_service
      - 53e2e05 Merge branch '2017.7' into fix_win_service
      - cf3d287 Merge branch '2017.7' into fix_win_service

        - 1641851 Fix the name of the _cmd_quote test
        - 97567af Skip tests on Linux systems
        - 0ca9cd1 Fix some lint
        - 3800966 Add tests
        - b0d646d Add more details to the functioning of _cmd_quote
        - a9856e2 Fix issues with win_service

- **ISSUE** #49452: (gtmanfred) [python3.7] remove usage of collections.abc stuff from the collections shim (refs: #49487)
- **PR** #49487: (garethgreenaway) [2018.3] Swapping out collections imports @ *2018-09-07 15:36:04 UTC*
    - baafd6e Merge pull request #49487 from garethgreenaway/49452_collections_abc
    - 2fb3ef7 Merge branch '2018.3' into 49452_collections_abc
    - ebfd7f9 Merge branch '2018.3' into 49452_collections_abc
    - b27e86b Swapping out the version check for a try…except on the import from collections.abc with a fallback to importing from collections.
    - d118a9f Updating various imports that have been moved from collections to collections.abc, and will be deprecated in collections beginning in 3.8.
        - abc9c1a Fix linter issues
        - 8e42384 Fix multiple issues in x509 module and state
        - 2a52158 Fix encoding issues in file and x509 states
        - 49a6da7 Fix map file loading on windows
- **PR** #49528: (dwoz) Fix merge wart @ *2018-09-06 23:12:39 UTC*
    - 06935e9 Merge pull request #49528 from dwoz/mergetests
    - 6a4f823 Fix merge wart
- **PR** #49548: (garethgreenaway) [2018.3] Disabling State boto tests for Python 3.7+ @ *2018-09-06 21:50:10 UTC*
    - 3298e70 Merge pull request #49548 from garethgreenaway/1075_disable_boto_tests_part_deux
    - 0f5191e Disable various boto tests when run under python 3.7 because of //github.com/spulec/moto/issues/1706. which was causing the test suite to hang on unit tests. This PR is disabling the tests in the test_boto_vpc state tests.
- **PR** #49521: (terminalmage) Fix _get_hash in splay executor @ *2018-09-06 14:06:54 UTC*
    - c346825 Merge pull request #49521 from terminalmage/fix-splay-executor
    - bbb8fe8 Fix _get_hash in splay executor
- **PR** #49526: (terminalmage) Prevent lookup error when trying to lookup fileserver function from LazyDict @ *2018-09-06 14:01:30 UTC*
    - 3f7c89c Merge pull request #49526 from terminalmage/fix-backends-error-logging
    - 1f0288e Prevent lookup error when trying to lookup fileserver function from LazyDict
- **PR** #49524: (garethgreenaway) [2018.3] Disable boto tests under 3.7 @ *2018-09-06 13:35:18 UTC*
    - bf939f9 Merge pull request #49524 from garethgreenaway/1075_disable_boto_tests

  - 5ec5578 Disable various boto tests when run under python 3.7 because of //github.com/spulec/moto/issues/1706. which was causing the test suite to hang on unit tests.

- **PR** #49511: (rallytime) [2018.3] Merge forward from 2018.3.3 to 2018.3 @ *2018-09-05 19:52:25 UTC*
  - f73f2e5 Merge pull request #49511 from rallytime/merge-2018.3
  - 3ffc6c2 Merge branch '2018.3.3' into '2018.3'
- **PR** #49500: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-09-04 19:02:31 UTC*
  - 81a49a8 Merge pull request #49500 from rallytime/merge-2018.3
  - 2317b5e Merge branch '2017.7' into '2018.3'
    - 1cdaea2 Merge pull request #49475 from dwoz/flaky_tests
      - f214929 Fix a flaky test and mark another as flaky
    - 4ac4305 Merge pull request #49457 from rallytime/labels-docs
      - d1e51db Remove references to unused milestones in docs
    - 41dcc3f Merge pull request #49461 from saltstack/revert-47100-ssh
      - 0d7180a Revert "Allow for not being prompted to supply a password to deploy keys to a…"
- **PR** #49497: (rallytime) Back-port #49473 to 2018.3 @ *2018-09-04 18:44:25 UTC*
  - **PR** #49473: (rallytime) Use correct and explicit paths for salt utils libs (refs: #49497)
  - 16ca360 Merge pull request #49497 from rallytime/bp-49473
  - 1aa3935 Lint: Add range import from six
  - d06f6a5 Use correct and explicit paths for salt utils libs
- **PR** #49474: (dwoz) Work around git-python resource leaks @ *2018-09-02 19:42:32 UTC*
  - 93df5c4 Merge pull request #49474 from dwoz/gitfs_unit
  - 56068e9 Merge branch '2018.3' into gitfs_unit
  - e697ddc Work around git-python resource leaks
- **PR** #49471: (Ch3LL) Remove string conversion on frame object in sigusr1 handler @ *2018-09-02 18:05:13 UTC*
  - 43ead5d Merge pull request #49471 from Ch3LL/debug_sigusr1
  - 1631b2f Remove string conversion on frame object in sigusr1 handler
- **PR** #49454: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-08-31 21:36:34 UTC*
  - 764b816 Merge pull request #49454 from rallytime/merge-2018.3
  - a97a7b0 Fix lint and test failures caused by bad merge
  - f7226f4 Merge branch '2017.7' into '2018.3'
    - 7a166bc Merge pull request #49376 from twangboy/fix_48608
      - bf32e0c Merge branch 'fix_48608' of https://github.com/twangboy/salt into fix_48608
      - 9766f0d Merge branch '2017.7' into fix_48608
      - fb97b00 Fix broken path to libsodium tarball
      - 535d83e Fix test for list_users
      - 40d3f2e Use dscl to get list of users

    - e426459 Use a set to avoid duplicates
- 03d9750 Merge pull request #49434 from dwoz/cleanup_exception_handling
    - 9daa992 Fix group remove test logic
    - d74fab7 Clean up exception handling on py3
- 7099164 Merge pull request #49389 from gtmanfred/2017.7
    - 6c01662 remove cmd key from load
    - bdf3df3 add cli example
    - 251f321 pass load on
    - be7c041 add test
    - 0ace5c1 make file envs compatible with transport format
- 24faa5e Merge pull request #49408 from terminalmage/issue49269
    - d02ec34 Allow our custom yaml dumper to NamespacedDictWrapper objects
- 5746fc8 Merge pull request #49402 from rallytime/bp-49321
    - 7dec9fc Add flaky decorator to serializer test
- 5c723b0 Merge pull request #49375 from terminalmage/revert-49185
    - aaea274 Merge branch '2017.7' into revert-49185
    - 7372e9d Add minion documentation for enable_gpu_grains
    - ea1b53c Fix incorrect master docs for enable_gpu_grains
    - 638210a Add validation type and default minion value for enable_gpu_grains
    - 49ed156 Add enable_gpu_grains to the minion config stub
    - d1b7fb4 Revert "Update documentation to correctly state enable_gpu_grains default"
- 11d87e4 Merge pull request #49387 from twangboy/fix_win_repo
    - 2bf7eda Import GLOBAL_ONLY
- 651631d Merge pull request #49390 from dwoz/archive_module
    - b332beb Fix archive tests for py3
- c591d2b Merge pull request #49385 from dwoz/sonofa
    - f9d7dbd The autoruns module has not been renamed
- 5d2c9b6 Merge pull request #49371 from rallytime/bp-49064
    - bc52f7c Stop running lint on all files when no changes
- f1c904e Merge pull request #49380 from twangboy/doc_48758
    - 6cefbdf Add docs from a comment on issue 48758
- b22a43d Merge pull request #49324 from dwoz/windows_pr_builds
    - 7abd9cd Merge branch '2017.7' into windows_pr_builds
- b2e4121 Merge pull request #49379 from dwoz/whitelistagain
    - 56ea4ee Account for more tests that are not in 2017.7
- 7fa9120 Merge pull request #49366 from rallytime/bp-49232

        - 74b05ef fix HTTP method for acl_info
        - b9fa7db Merge branch '2017.7' into windows_pr_builds
        - 77f5fd3 Merge branch '2017.7' into windows_pr_builds
        - 7cdba27 Add ci scripts for windows PR builds
- **PR** #49025: (twangboy) Fix several issues with LGPO @ *2018-08-31 18:25:34 UTC*
    - 4ea22e5 Merge pull request #49025 from twangboy/fix_48661
    - c27fd18 Merge branch '2018.3' into fix_48661
    - 222c503 Fix several issues with LGPO
- **ISSUE** #49281: (aarnaud) etcd_cache : Not working for mine (refs: #49283)
- **PR** #49283: (aarnaud) Fix #49281 etcd_cache with mine cache @ *2018-08-31 18:24:23 UTC*
    - dd9ec94 Merge pull request #49283 from aarnaud/fix-etcd-cache
    - 13e1a17 Merge branch '2018.3' into fix-etcd-cache
    - 6c0fb9c Fix #49281 etcd_cache with mine cache
- **PR** #49459: (dwoz) Batch test fix cherry-pick @ *2018-08-30 23:02:12 UTC*
    - 64227f9 Merge pull request #49459 from dwoz/batch_fix
    - fc39dcf Simplify cli test fixes
- **PR** #49453: (dwoz) Fix up py3 git config tests @ *2018-08-30 22:00:19 UTC*
    - 75285a3 Merge pull request #49453 from dwoz/conffix
    - ac3c379 Fix up py3 git config tests
- **PR** #49411: (terminalmage) Allow our custom yaml dumper to NamespacedDictWrapper objects (2018.3 branch) @ *2018-08-30 13:05:37 UTC*
    - a962efc Merge pull request #49411 from terminalmage/issue49269-2018.3
    - a4cb35b Remove support for overriding the class name
    - 82a8b73 Allow our custom yaml dumper to NamespacedDictWrapper objects
- **PR** #49407: (rallytime) Mark pkg module tests and flaky @ *2018-08-30 13:01:31 UTC*
    - 8022a3b Merge pull request #49407 from rallytime/pkg-flaky
    - 1609e11 Mark pkg module tests and flaky
- **PR** #49406: (rallytime) [2018.3] Merge forward from 2018.3.3 to 2018.3 @ *2018-08-30 13:00:34 UTC*
    - 7f8c38a Merge pull request #49406 from rallytime/merge-2018.3
    - 4939ee3 Merge branch '2018.3.3' into '2018.3'
- **PR** #49401: (rallytime) Mark output file permissions test as flaky @ *2018-08-29 18:13:12 UTC*
    - 68d630f Merge pull request #49401 from rallytime/flaky-output-test
    - e13216b Mark output file permissions test as flaky
- **PR** #49400: (rallytime) Mark pillar refresh test as flaky @ *2018-08-29 18:12:51 UTC*
    - 23610ed Merge pull request #49400 from rallytime/flaky-pillar-test
    - fb62af6 Mark pillar refresh test as flaky

- **ISSUE** #49150: (sunyq) hosts.rm_host changes /etc/hosts file with odd characters (refs: #49394)
- **PR** #49394: (sunyq) fix issue'#49150'_ hosts.rm_host changes /etc/hosts with odd characters @ *2018-08-29 13:33:49 UTC*
  - 502e91c Merge pull request #49394 from sunyq/2018.3
  - 76ba5b6 fix issue'#49150'_ hosts.rm_host changes /etc/hosts with odd characters
- **PR** #49386: (dwoz) Cherry-pick whitelist fix. @ *2018-08-29 01:11:05 UTC*
  - 2cf2e62 Merge pull request #49386 from dwoz/whitelistfix
  - 2b08cf1 Account for more tests that are not in 2017.7
- **PR** #49372: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-08-28 19:03:53 UTC*
  - ec04282 Merge pull request #49372 from rallytime/merge-2018.3
  - ab6cc1c Update old utils paths to use new paths
  - ac406c4 Merge branch '2017.7' into '2018.3'
  - dfa3861 Merge pull request #49354 from dwoz/fix_whitelist
    - aeb0fa3 custom_grains tests do not exist
  - 14b8094 Merge pull request #49346 from Ch3LL/bp-49345
    - ce2733e upgrade including linux kernels
  - d8f2945 Merge pull request #49344 from cachedout/test_matcher_flaky
    - a34ef2e Matcher test marked flaky
  - f08e720 Merge pull request #49353 from dwoz/boto_fix
    - 376019f Do not raise exception if passed bytes
  - 4690ad0 Merge pull request #49339 from dwoz/more_tests
    - d50324b Merge branch '2017.7' into more_tests
  - e371ad9 Merge pull request #49341 from terminalmage/issue48717
    - 1971d09 Update dynamic git_pillar docs to enclose env name in quotes
  - 00b58ee Merge pull request #49337 from dwoz/win_service_test_fix
    - 1c1727e Fix broken service stop test
  - 81428ed Merge pull request #49291 from dwoz/dedent
    - 1991198 Fix review nits
    - 6abc2da Re-factor dedent to fix warts
    - 75b2e3d Use salt.utils.to_* functions
    - fa78a7b Add dedent that sets line endings
      - cd6706e Fix merge wart
      - b9c344b Fix module.archive tests for win py3
      - ac60fb8 Comment strange code
      - 5fefd6f Simplify cli test fixes
      - 1765d53 Fix archive tests on Windows platform

        - 06f271f Add archive module tests to whitelist.txt
        - 9cb8fe2 Add more tests to whitelist for windows
        - b8af238 Fix indent wart
        - bc971e4 add doc tests
        - 8899a5e Add more client tests to whitelist
        - 14cd7b48 Add cli tests to whitelist
        - 3e55d10 Platform module doesn't exist in 2017.7
        - 2733c02 Use double quotes on windows commands
        - 0394ece The set command does not work like inline env vars
  - 1fd5cf1 Merge pull request #49320 from rallytime/fix-48694
    - 0964b5e Update documentation for #48694 and add deprecation warning
  - c323096 Merge pull request #49314 from rallytime/bp-49277
    - 6d691b2 Prepend current directory when path is just filename
  - c7bae5e Merge pull request #49290 from rallytime/bp-44504
    - 6224f7b calling range is going up to the upper limit but not including it
  - 7a44e59 Merge pull request #49289 from rallytime/bp-49170
    - 4c29c17 forcing mine update in the proxy minion as well
    - b1d581a force mine update on minion start
  - 221ea22 Merge pull request #49278 from rallytime/bp-49253
    - b331b5c skip ID 7 for vmware hard drives
- **PR** #49356: (dwoz) Fix tests that use timed_subprocess for py3 @ *2018-08-28 13:30:08 UTC*
  - 1faf6a7 Merge pull request #49356 from dwoz/cmdmod
  - 6ffeaae Fix tests that use timed_subprocess for py3
- **ISSUE** #48299: (dosercz) git.latest reports deleted tags in remote repo as new_tags in changes (refs: #49305)
- **PR** #49336: (terminalmage) Fix half-baked comment @ *2018-08-27 00:07:55 UTC*
  - **PR** #49305: (terminalmage) Allow git.latest to remove local tags which have been removed remotely (refs: #49336)
  - 2d8055e Merge pull request #49336 from terminalmage/issue48299
  - 175e161 Fix half-baked comment
- **PR** #49331: (dwoz) Use salt.utils to ensure string type @ *2018-08-26 01:42:08 UTC*
  - a2e7033 Merge pull request #49331 from dwoz/strfix
  - 7513474 Use salt.utils to ensure string type
- **PR** #49312: (Ch3LL) Fix keyerror in manage.bootstrap @ *2018-08-25 19:55:44 UTC*
  - d70eda8 Merge pull request #49312 from Ch3LL/ssh_list_hosts
  - da71c97 Fix keyerror in manage.bootstrap
- **PR** #49316: (rallytime) [2018.3] Merge forward from 2018.3.3 to 2018.3 @ *2018-08-25 19:53:04 UTC*

  - 6b2eba7 Merge pull request #49316 from rallytime/merge-2018.3
  - 39c442c Merge branch '2018.3.3' into '2018.3'
- **ISSUE** #48880: (damntoken) Can't run cmd.run with UTF-8 chars as arguments / parameters. With custom module. (refs: #49322)
- **PR** #49322: (dwoz) Encode shell commands explicitly. @ *2018-08-25 04:43:26 UTC*
  - 5766e98 Merge pull request #49322 from dwoz/state_module_test_fix
  - 1f7d50d Encode shell commands explicitly.
- **PR** #49299: (dwoz) Work around cmd.run unicode issues in test for now @ *2018-08-25 04:43:03 UTC*
  - b47da0a Merge pull request #49299 from dwoz/test_fix
  - f4bd644 Fix string formatting wart in file state tests
  - e84e608 Fix wart in file state test
  - 71d4465 Work around cmd.run unicode issues in test for now
- **ISSUE** #48299: (dosercz) git.latest reports deleted tags in remote repo as new_tags in changes (refs: #49305)
- **PR** #49305: (terminalmage) Allow git.latest to remove local tags which have been removed remotely (refs: #49336) @ *2018-08-24 17:29:03 UTC*
  - b65890c Merge pull request #49305 from terminalmage/issue48299
  - 4a093d9 lint
  - ed7b994 Gate tag manipulation behind a sync_tags argument
  - 29de855 Add test for a removed tag
  - eb9a459 Properly handle tags deleted from remote repository
  - 916c15a Add git.tag function
- **ISSUE** #49009: (msciciel) file_ignore_regex / file_ignore_glob not working properly (refs: #49308)
- **PR** #49308: (terminalmage) Don't include ignored paths in mtime map @ *2018-08-24 17:28:41 UTC*
  - 459354a Merge pull request #49308 from terminalmage/issue49009
  - 2badd7f Don't include ignored paths in mtime map
- **PR** #49282: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 @ *2018-08-24 16:45:50 UTC*
  - 914bb09 Merge pull request #49282 from rallytime/merge-2018.3
  - 21a51c9 Add 'minion_data_cache: True' to mocked opts for minions unit tests
  - 196ce3f Call the *with_tempfile* decorator correctly for 2018.3
  - 37f24fd Fix unit test for file state
  - 6f9a158 Update old utils paths with new paths
  - c3ecefc Merge branch '2017.7' into '2018.3'
    - 879c3ad Merge pull request #49262 from bloomberg/short-circuit
      - a3594db extend #488588 to cover SyncAuth class
    - 1900aff Merge pull request #49144 from twangboy/fix_48163
      - ff5ec86 Work with seconds
      - 7264008 Fix some lint (remove whitespace)`

  - 4f9973d Improve timeout in delete
  - a6ecb75 Remove testing debug stuff
  - 7dd7953 Fix start/stop functions
  - fade0a6 Merge pull request #10 from damon-atkins/patch-1
  - 57c4b9f 2017_win_service_damon
  - d44eaee Add timeout support to the state
  - d579b3e Add timeout parameter
- 395bae6 Merge pull request #49259 from gtmanfred/flaky-2017.7
  - 93a576e flaky tests are flaky yo
  - f518bd3 mark orchestration state tests as flaky
- 62cc0df Merge pull request #49231 from minusf/mount-check-name
  - eb5cab3 fix some underhanging indent while here…
  - ec2a091 check for mandatory parameters to avoid false positives
- d55d484 Merge pull request #49242 from dwoz/blockreplace_better_fix
  - 72c3727 Use six to make sure content is unicode
  - 1bf0b18 Better blockfix replace
  - dd4fcd3 Revert "Multiple block replace test fixes"
  - d335842 Use os.linesep.join instead of textwrap.dedent
- 85f6d36 Merge pull request #49236 from terminalmage/issue32737
  - 233bbae Allow compound matching in eauth config expressions
- 5ba7f60 Merge pull request #49187 from erwindon/py27fixnumberformat
  - 12261a5 trailing whitespace removal
  - 2149e22 Fix for #45620: "Salt CLI is rounding floats to 2 decimal places" (actually: Salt CLI is using only 12 digits for precision)
- f8c55b8 Merge pull request #49184 from ralish/external_nodes_docs
  - 4fe38dc Merge branch '2017.7' into external_nodes_docs
  - 2e29b29 Remove obsolete documentation on external_nodes setting
- 65205a4 Merge pull request #49185 from ralish/enable_gpus_grains_doc
  - 2fe675c Update documentation to correctly state enable_gpu_grains default
- 5aa282e Merge pull request #48032 from zer0def/parted-naming
  - 563ad25 Merge branch '2017.7' into parted-naming
  - 07f8631 Fixed partition names with spaces effectively containing only the first word.
- 0157eac Merge pull request #49164 from terminalmage/issue49154
  - c2aba7a Merge branch '2017.7' into issue49154
  - b4544d7 Add keep_source integration tests
  - 82638c6 Fix bug in keep_source for non-templated salt:// file sources

        - b510441 Merge pull request #49179 from dwoz/test_unit_states_file
            - 913ea5e Fix directory unit test
            - 3363238 Account for normalized dirs in unit tests
        - 7486fd5 Merge pull request #49162 from erwindon/wheel_error_error
            - ca5df04 Merge branch '2017.7' into wheel_error_error
            - 4335c5c Must have 2 lines between imports and code
            - ec1f013 Improved solution as per @dwoz's suggestion
            - 20f134e Fixed unknown 'exceptions' under Python3 (#49152)
        - 7043286 Merge pull request #49143 from Ch3LL/bp-49142
            - d2e73cc Remove -Z script_arg for cloud tests
        - 8ab55f5 Merge pull request #49167 from dwoz/filefix
            - b5ba073 Simplify dict keys lookup
            - 6844251 Fix remaining file state integration tests (py3)
        - 90bd560 Merge pull request #49163 from dwoz/tmp_dir
            - 3384864 add Exception type
            - 9da79dd0 Allow test suite to finish if tmp dir removal fails
        - aa01a67 Merge pull request #49136 from Ch3LL/bootstrap_2017.7
            - de40dfb [2017.7] Update bootstrap script to latest release (2018.08.15)
        - 29fccbf Merge pull request #49118 from dwoz/test_file_fixes
            - c6b781e Multiple fixes for integration.states.test_file
        - 133e400 Merge pull request #49103 from dwoz/cmd_test_fix
        - 566a4ea Install the launcher so we can execute py files

- **ISSUE** saltstack/salt-jenkins#1075: (Ch3LL) [2018.3.3] arch python3 tests do not finish (refs: #49303)
- **PR** #49303: (gtmanfred) use os._exit instead of sys.exit when daemonizing @ *2018-08-24 14:15:54 UTC*
    - 4c81c2e Merge pull request #49303 from gtmanfred/forking
    - e06ce49 use os._exit instead of sys.exit when daemonizing
- **PR** #49276: (terminalmage) Fix paths to moved functions @ *2018-08-23 16:38:55 UTC*
    - 158c1ca Merge pull request #49276 from terminalmage/fix-moved-funcs
    - 322a988 Fix paths to moved functions
- **PR** #49270: (dwoz) Add async helper to test_sock_path_len @ *2018-08-23 16:17:54 UTC*
    - ef39f85 Merge pull request #49270 from dwoz/test_event_fix
    - 51bf79a Add async helper to test_sock_path_len
- **PR** #49199: (jacobweinstock) Fix typeerror unicode @ *2018-08-23 14:38:17 UTC*
    - 5da743a Merge pull request #49199 from jacobweinstock/fix-typeerror-unicode
    - 25608ef Merge branch '2018.3' into fix-typeerror-unicode
    - 11ad11c add unicode_literals back to handle logging and other things.

  - 6944baf add salt.utils.stringutils.to_bytes. add comment on unicode_literals.
- **ISSUE** #49174: (rbthomp) Beacon diskusage reporting incorrect diskusage value. (refs: #49264)
- **PR** #49264: (gtmanfred) fix diskusage beacon @ *2018-08-23 14:36:07 UTC*
  - e28f8f1 Merge pull request #49264 from gtmanfred/diskusage
  - 15857c2c fix diskusage
- **PR** #49253: (gtmanfred) skip ID 7 for vmware hard drives (refs: #49278) @ *2018-08-23 14:35:31 UTC*
  - f6b2b93 Merge pull request #49253 from gtmanfred/vmware
  - 6e776ca skip ID 7 for vmware hard drives
- **ISSUE** #49159: (wenxin-wang) incorrect default value of minion's 'ipv6' option (refs: #49263)
- **PR** #49263: (terminalmage) Revert default value for ipv6 to None @ *2018-08-23 13:02:13 UTC*
  - 2a2bd48 Merge pull request #49263 from terminalmage/issue49159
  - bf5d907 Revert default value for ipv6 to None
- **PR** #49244: (dwoz) Test fixes: unit.fileserver.test_gitfs @ *2018-08-22 20:19:42 UTC*
  - ec32428 Merge pull request #49244 from dwoz/gitfs_fixes
  - b533fa1 Simplify by using to_unicode helper
  - 5a0cda5 Older GitPython version do not have a close method
  - 7452715 Fix up fileserver.test_gitfs tests on windows
  - 800aa25 Remove unicode filenames on windows python 2
- **PR** #49265: (terminalmage) Fix failing unit file module unit tests @ *2018-08-22 19:35:07 UTC*
  - 2d7f846 Merge pull request #49265 from terminalmage/fix-failing-file-tests
  - 6b51f71 Fix failing unit file module unit tests
- **PR** #49240: (dwoz) file state test fixes @ *2018-08-22 18:16:42 UTC*
  - 4634752 Merge pull request #49240 from dwoz/state_tests
  - 1679a19 Work around listdir encoding issues on py2 windows
  - 07ed841 file state test fixes
- **PR** #49260: (gtmanfred) [2018.3] flaky tests @ *2018-08-22 16:40:39 UTC*
  - 55ded9f Merge pull request #49260 from gtmanfred/flaky-2018.3
  - 41973ff flaky tests are flaky yo
  - c927f1b mark orchestration state tests as flaky
- **PR** #49245: (dwoz) Skip grep unit tests on windows @ *2018-08-22 15:18:08 UTC*
  - 6d023ca Merge pull request #49245 from dwoz/skip_grep_windows
  - e305891 Skip grep unit tests on windows
- **PR** #49243: (dwoz) Revert newline translation change @ *2018-08-22 15:12:26 UTC*
  - 75beede Merge pull request #49243 from dwoz/fileserver_tests
  - 86e3c46 Revert newline translation change

- **PR** #48545: (bbinet) Fix unreachable ssh-id-wrapper template when root_dir is relative @ *2018-08-21 15:12:01 UTC*
    - 60605f0 Merge pull request #48545 from bbinet/fix-unreacheable-ssh-id-wrapper
    - 284dcf7 Fix unreachable ssh-id-wrapper template when root_dir is relative
- **ISSUE** #49204: (mchugh19) file.directory state fails if directory contains an invalid symlink (refs: #49209, #49827)
- **PR** #49209: (gtmanfred) don't blow up check_perms if lsattr file doesn't exist @ *2018-08-20 19:02:42 UTC*
    - a4c51a9 Merge pull request #49209 from gtmanfred/lsattr
    - ab918ca don't blow up check_perms if lsattr file doesn't exist
- **ISSUE** #49060: (The-Loeki) On Py3, redis job cache throws TypeError bytes / str (refs: #49190)
- **PR** #49190: (The-Loeki) decode_reponses=True on all Redis API interfaces @ *2018-08-20 18:57:26 UTC*
    - d29af00 Merge pull request #49190 from The-Loeki/redis-fix
    - d126318 Merge branch '2018.3' into redis-fix
    - 7ff72eb redis cache no utf8 of course
    - bdaef39 decode_reponses=True on all Redis API interfaces
- **PR** #49197: (dwoz) File state line ending fixes @ *2018-08-20 18:45:04 UTC*
    - 644ba3d Merge pull request #49197 from dwoz/test_fix
    - a7cfe35 File state line ending fixes
- **PR** #49192: (dwoz) Test fixes flaky test and unicode environment key/value @ *2018-08-19 10:48:03 UTC*
    - aaa9e37 Merge pull request #49192 from dwoz/git_test_fix_ng
    - 8ea89e9 Test fixes
- **PR** #49191: (The-Loeki) Make salt.utils.vault._get_connection publicly available @ *2018-08-19 10:39:35 UTC*
    - 63cc737 Merge pull request #49191 from The-Loeki/vault_publ_conn
    - cab4192 publicize salt.utils.get_vault_connection()
- **PR** #49189: (dwoz) Merge pull request #49171 from Ch3LL/bp-49103 @ *2018-08-18 20:51:59 UTC*
    - **PR** #49171: (Ch3LL) [2018.3.3] cherry pick #49103 (refs: #49189)
    - **PR** #49103: (dwoz) Install the launcher so we can execute py files (refs: #49189, #49171)
    - 1d843d4 Merge pull request #49189 from dwoz/cmd_test_fix
    - 8fd0c38 Merge pull request #49171 from Ch3LL/bp-49103
- **PR** #49188: (cachedout) Fix bug in test_pkg @ *2018-08-18 17:36:22 UTC*
    - 5f905d7 Merge pull request #49188 from cachedout/fix_test_pkg
    - 2cd6519 Fix bug in test_pkg
- **PR** #49056: (Ch3LL) Add orch runner with mine.get integration tests @ *2018-08-18 12:54:48 UTC*
    - 6fb8d4a Merge pull request #49056 from Ch3LL/test_mine
    - 3a8fed5 Handle race condition when running mine.update in orch mine test
    - b4e421c Add orch runner with mine.get integration tests
- **PR** #49126: (The-Loeki) TLS cert_info fixes @ *2018-08-18 12:19:33 UTC*

  - **PR** #49123: (The-Loeki) Fix return of tls.cert_info extensions (refs: #49126)
  - 0d0e11d Merge pull request #49126 from The-Loeki/tls-impr
  - 22240c0 Merge branch '2018.3' into tls-impr
  - 3ce2c3e tls.cert_info: strip newlines/spaces from extensions
  - d522d01 tls alt_names fix
  - d435dbc Fix bytes-return of tls.cert_info extensions
  - 9c7d3ea TLS cert_info: Accept IP Addresses as valid alt_names
  - 7d6ecaf tls.check_info: Allow PEM as string input
- **PR** #49186: (dwoz) Fix typo in #49180 @ *2018-08-18 12:07:00 UTC*
  - **PR** #49180: (dwoz) Cherry-pick test fixes (refs: #49186)
  - 7dd70eb Merge pull request #49186 from dwoz/test_fixes
  - 95efc51 Fix typo
- **PR** #49180: (dwoz) Cherry-pick test fixes (refs: #49186) @ *2018-08-18 04:08:17 UTC*
  - 6d57ac1 Merge pull request #49180 from dwoz/test_fixes
  - 3c49544 Account for file renames
  - acc144d Fix directory unit test
  - a7ee07a Fix is_windows checks
  - 74b197f Account for normalized dirs in unit tests
  - a18d937 Merge pull request #49167 from dwoz/filefix
- **PR** #49091: (terminalmage) Rename/re-organize test files in 2018.3 branch @ *2018-08-17 12:26:44 UTC*
  - **PR** #47337: (terminalmage) Modify runtests.py to run a subset of tests based on filenames (refs: #49091)
  - 8bda1e5 Merge pull request #49091 from terminalmage/rename-test-files
  - 3d8523b Fix import for moved module
  - 613f061 Lint
  - 81f272f Rename outputter tests to match naming convention
  - fba0336 Rename venafi test module to bring it closer to the naming convention
  - 9c14500 Rename exec module integration test files to respect naming convention
  - 91fa09e Rename providers to clouds to respect naming convention
  - d56dcdd Rename "providers" dir to "clouds" to respect naming convention
  - f1b7d54 Rename virtualenv state test module to respect naming convention
  - c1f94db Split ssh state tests into separate modules to respect naming convention
  - 9c8ba4a Split mysql database tests into two modules to respect naming convention
  - d51748e Rename some files to avoid them being identified as test modules
  - 1fccbf7 Rename acl unit test module to confirm to naming convention
  - 160223d Consolidate/move fileclient/fileserver tests to conform to naming convention
  - 1aacf77 Rename/consolidate salt.utils unit tests to conform to naming convention

  - 5a0990b Rename pillar unit test modules to reflect naming convention
  - 516594b Consolidate netapi tests and move them to reflect naming convention
  - 8f5fef5 Consolitdate and move salt-ssh unit tests to proper location
  - 9801ce3 Consolidate and move loader tests to test/unit/test_loader.py
  - f72bbc5 Rename beacon unit test modules to conform with naming convention
  - fa76cb6 Move inspectlib tests to conform with naming convention
  - 81e3823 Move shared kernelpkg test cases to tests/support/
  - e0d95b6 Consolidate misplaced tests into tests/unit/test_config.py
  - 3a1ec12 Rename slack engine test file to respect naming convention
  - c9c9d20 Rename test_map_conf test file to respect naming convention
  - 5c936c0 Rename the aptpkg state module to respect naming convention
  - ac50eb8 Remove redundant "test" from filename
  - b03a563 Move jinja templating tests alongside salt/utils/jinja.py's tests
  - b49d8df Fix bad copypasta in docstring
  - 069102b Rename sentry returner test to match naming convention
  - 2398083 Use tests.support.paths instead of deprecated integration module
  - ffb842d Move salt.utils.context tests to their proper location
  - d69069b Move context cache tests to their proper location

- **PR** #49062: (weswhet) fix memoize on available macOS services @ *2018-08-15 15:34:25 UTC*
  - c8510a6 Merge pull request #49062 from weswhet/fix-mac-available-services
  - f1b40c4 decorator link fix, updating context names, as well as updating macutils tests for latest changes
  - 1e4497d fixing an issue with memoize on macOS services, switching to useing __context__ instead
- **ISSUE** #49100: (pruiz) salt.utils.yamldumper.dump/safe_dump incorrectly return 'str' instead of unicode on python2.7 (centos) (refs: #49125)
- **PR** #49125: (terminalmage) Ensure that we don't feed jinja2.Markup() a str type @ *2018-08-15 04:02:30 UTC*
  - 21435be Merge pull request #49125 from terminalmage/issue49100
  - ec258e2 Add integration tests for yaml and json filters
  - f5f16cf Ensure that we don't feed jinja2.Markup() a str type

28.2.16 Salt 2018.3.5 Release Notes

Version 2018.3.5 is a CVE-fix release for *2018.3.0*.

Security Fix

CVE-2019-17361

With the Salt NetAPI enabled in addition to having a SSH roster defined, unauthenticated access is possible when specifying the client as SSH. Additionally, when the raw_shell option is specified any arbitrary command may be run on the Salt master when specifying SSH options.

28.2.17 Salt 2017.7.0 Release Notes - Codename Nitrogen

28.2.18 Python 3

The 2017.7 Salt Release adds initial Python 3 support.

The default Python version of Salt will remain Python 2, although Python 3 packages will be supplied for users who want to help test this new feature.

28.2.19 Python 2.6 Deprecation

Salt will no longer support Python 2.6. We will provide python2.7 packages on our repo for RedHat and CentOS 6 to ensure users can still run Salt on these platforms.

As this will impact the installation of additional dependencies for salt modules please use pip packages if there is not a package available in a repository. You will need to install the python27-pip package to get access to the correct pip27 executable: `yum install python27-pip`

28.2.20 Known Issues

The following salt-cloud drivers have known issues running with Python 3. These drivers will not work with Python 3, and Python 2.7 should be used instead:

- Joyent
- When running under Python 3, users who require Unicode support should ensure that a locale is set on their machines. Users using the *C* locale are advised to switch to a UTF-aware locale to ensure proper functionality with Salt with Python 3.

Remember to update the Salt Master first

Salt's policy has always been that when upgrading, the minion should never be on a newer version than the master. Specifically with this update, because of changes in the fileclient, the 2017.7 minion requires a 2017.7 master.

Backwards compatibility is still maintained, so older minions can still be used.

More information can be found in the *Salt FAQ*

States Added for Management of systemd Unit Masking

The *`service.masked`* and *`service.umasked`* states have been added to allow Salt to manage masking of systemd units.

Additionally, the following functions in the `systemd` execution module have changed to accommodate the fact that indefinite and runtime masks can co-exist for the same unit:

- `service.masked` - The return from this function has changed from previous releases. Before, `False` would be returned if the unit was not masked, and the output of `systemctl is-enabled <unit name>` would be returned if the unit was masked. However, since indefinite and runtime masks can exist for the same unit at the same time, this function has been altered to accept a `runtime` argument. If `True`, the minion will be checked for a runtime mask assigned to the named unit. If `False`, then the minion will be checked for an indefinite mask. If one is found, `True` will be returned. If not, then `False` will be returned.

- `service.masked` - This function used to just run `systemctl is-enabled <unit name>` and based on the return from this function the corresponding mask type would be removed. However, if both runtime and indefinite masks are set for the same unit, then `systemctl is-enabled <unit name>` would show just the indefinite mask. The indefinite mask would be removed, but the runtime mask would remain. The function has been modified to accept a `runtime` argument, and will attempt to remove a runtime mask if that argument is set to `True`. If set to `False`, it will attempt to remove an indefinite mask.

These new `runtime` arguments default to `False`.

Pillar Encryption

Beginning in 2016.3.0 the CLI pillar data passed to several functions could conditionally be passed through a renderer to be decrypted. This functionality has now been extended to pillar SLS files as well. See *here* for detailed documentation on this feature.

Grains Changes

- The `osmajorrelease` grain has been changed from a string to an integer. State files, especially those using a templating language like Jinja, may need to be adjusted to account for this change.
- Add ability to specify disk backing mode in the VMWare salt cloud profile.

State Module Changes

- The *`service.running`* and *`service.dead`* states now support a `no_block` argument which, when set to `True` on systemd minions, will start/stop the service using the `--no-block` flag in the `systemctl` command. On non-systemd minions, a warning will be issued.
- The *`module.run`* state has dropped its previous syntax with `m_` prefix for reserved keywords. Additionally, it allows running several functions in a batch.

  Note: It is necessary to explicitly turn on the new behavior (see below)

```
# Before
run_something:
  module.run:
    - name: mymodule.something
    - m_name: 'some name'
    - kwargs: {
      first_arg: 'one',
      second_arg: 'two',
      do_stuff: 'True'
    }

# After
run_something:
  module.run:
    - mymodule.something:
      - name: some name
      - first_arg: one
      - second_arg: two
      - do_stuff: True
```

Since a lot of users are already using *module.run* states, this new behavior must currently be explicitly turned on, to allow users to take their time updating their SLS files. However, please keep in mind that the new syntax will take effect in the next feature release of Salt (Oxygen) and the old usage will no longer be supported at that time.

Another feature of the new *module.run* is that it allows calling many functions in a single batch, such as:

```
run_something:
  module.run:
    - mymodule.function_without_parameters:
    - mymodule.another_function:
      - myparam
      - my_other_param
```

In a rare case that you have a function that needs to be called several times but with the different parameters, an additional feature of "tagging" is to the rescue. In order to tag a function, use a colon delimiter. For example:

```
run_something:
  module.run:
    - mymodule.same_function:1:
    - mymodule.same_function:2:
      - myparam
      - my_other_param
    - mymodule.same_function:3:
      - foo: bar
```

The example above will run *mymodule.same_function* three times with the different parameters.

To enable the new behavior for *module.run*, add the following to the minion config file:

```
use_superseded:
  - module.run
```

- The default for the `fingerprint_hash_type` option used in the `present` function in the `ssh` state changed from `md5` to `sha256`.

Execution Module Changes

- Several functions in the `systemd` execution module have gained a `no_block` argument, which when set to `True` will use `--no-block` in the `systemctl` command.
- In the `solarisips` `pkg` module, the default value for the `refresh` argument to the `list_upgrades` function has been changed from `False` to `True`. This makes the function more consistent with all of the other `pkg` modules (The other `pkg.list_upgrades` functions all defaulted to `True`).
- The functions which handle masking in the `systemd` module have changed. These changes are described above alongside the information on the new states which have been added to manage masking of systemd units.
- The *pkg.list_repo_pkgs* function for yum/dnf-based distros has had its default output format changed. In prior releases, results would be organized by repository. Now, the default for each package will be a simple list of versions. To get the old behavior, pass `byrepo=True` to the function.
- A `pkg.list_repo_pkgs` function has been added for both *Debian/Ubuntu* and `Arch Linux`-based distros.
- The *system* module changed its return format from "HH:MM AM/PM" to "HH:MM:SS AM/PM" for *get_system_time*.

- The default for the `fingerprint_hash_type` option used in the *ssh* execution module changed from `md5` to `sha256`.

Proxy Module Changes

The *proxy_merge_grains_in_module* configuration variable introduced in 2016.3, has been changed, defaulting to `True`.

The connection with the remote device is kept alive by default, when the module implements the `alive` function and *proxy_keep_alive* is set to `True`. The polling interval is set using the *proxy_keep_alive_interval* option which defaults to 1 minute.

The developers are also able to use the *proxy_always_alive*, when designing a proxy module flexible enough to open the connection with the remote device only when required.

Wildcard Versions in `pkg.installed` States

- The *pkg.installed* state now supports wildcards in package versions, for the following platforms:
  - SUSE/openSUSE Leap/Thumbleweed
  - Debian/Ubuntu
  - RHEL/CentOS
  - Arch Linux

  This support also extends to any derivatives of these distros, which use the *aptpkg*, *yumpkg*, or `pacman` providers for the `pkg` virtual module.

  Using wildcards can be useful for packages where the release name is built into the version in some way, such as for RHEL/CentOS which typically has version numbers like `1.2.34-5.el7`. An example of the usage for this would be:

```
mypkg:
  pkg.installed:
    - version: '1.2.34*'
```

Master Configuration Additions

- *syndic_forward_all_events* - Option on multi-syndic or single when connected to multiple masters to be able to send events to all connected masters.
- *eauth_acl_module* - In case external auth is enabled master can get authenticate and get the authorization list from different auth modules.
- *keep_acl_in_token* - Option that allows master to build ACL once for each user being authenticated and keep it in the token.

Minion Configuration Additions

- *pillarenv_from_saltenv* - When set to `True` (default is `False`), the *pillarenv* option will take the same value as the effective saltenv when running states. This would allow a user to run `salt '*' state.apply mysls saltenv=dev`, and the SLS for both the state and pillar data would be sourced from the `dev` environment, essentially the equivalent of running `salt '*' state.apply mysls saltenv=dev` `pillarenv=dev`. Note that if *pillarenv* is set in the minion config file, or if `pillarenv` is provided on the CLI, it will override this option.

salt-api Changes

The `rest_cherrypy` netapi module has received a few minor improvements:

- A CORS bugfix.
- A new `/token` convenience endpoint to generate Salt eauth tokens.
- A proof-of-concept JavaScript single-page application intended to demonstrate how to use the Server-Sent Events stream in an application. It is available in a default install by visiting the `/app` URL in a browser.

Python API Changes

`expr_form` Deprecation

The *LocalClient*'s `expr_form` argument has been deprecated and renamed to `tgt_type`. This change was made due to numerous reports of confusion among community members, since the targeting method is published to minions as `tgt_type`, and appears as `tgt_type` in the job cache as well.

While `expr_form` will continue to be supported until the **2019.2.0** release cycle (two major releases after this one), those who are using the *LocalClient* (either directly, or implictly via a *netapi module*) are encouraged to update their code to use `tgt_type`.

`full_return` Argument in `LocalClient` and `RunnerClient`

An `full_return` argument has been added to the `cmd` and `cmd_sync` methods in `LocalClient` and `RunnerClient` which causes the return data structure to include job meta data such as `retcode`.

This is useful at the Python API:

```
>>> import salt.client
>>> client = salt.client.LocalClient()
>>> client.cmd('*', 'cmd.run', ['return 1'], full_return=True)
{'jerry': {'jid': '20170520151213898053', 'ret': '', 'retcode': 1}}
```

As well as from salt-api:

```
% curl -b /tmp/cookies.txt -sS http://localhost:8000 \
    -H 'Content-type: application/json' \
    -d '[{
        "client": "local",
        "tgt": "*",
        "fun": "cmd.run",
        "arg": ["return 1"],
        "full_return": true
    }]'

{"return": [{"jerry": {"jid": "20170520151531477653", "retcode": 1, "ret": ""}}]}
```

Jinja

Filters

New filters in 2017.7.0:

- *to_bool*
- *exactly_n_true*
- *exactly_one_true*
- *quote*
- *regex_search*
- *regex_match*
- *uuid*
- *is_list*
- *is_iter*
- *min*
- *max*
- *avg*
- *union*
- *intersect*
- *difference*
- *symmetric_difference*
- *is_sorted*
- *compare_lists*
- *compare_dicts*
- *is_hex*
- *contains_whitespace*
- *substring_in_list*
- *check_whitelist_blacklist*
- *date_format*
- `str_to_num`
- *to_bytes*
- *json_decode_list*
- *json_decode_dict*
- `rand_str`
- *md5*
- *sha256*
- *sha512*

- *base64_encode*
- *base64_decode*
- *hmac*
- *http_query*
- *is_ip*
- *is_ipv4*
- *is_ipv6*
- *ipaddr*
- *ipv4*
- *ipv6*
- *network_hosts*
- *network_size*
- *gen_mac*
- *mac_str_to_bytes*
- *dns_check*
- *is_text_file*
- *is_binary_file*
- *is_empty_file*
- *file_hashsum*
- *list_files*
- *path_join*
- *which*

Logs

Another new feature - although not limited to Jinja only - is being able to log debug messages directly from the template:

```
{%- do salt.log.error('logging from jinja') -%}
```

See the *logs* paragraph.

Network Automation

NAPALM

Introduced in 2016.11, the modules for cross-vendor network automation have been improved, enhanced and widenened in scope:

- Manage network devices like servers: the NAPALM modules have been transformed so they can run in both proxy and regular minions. That means, if the operating system allows, the salt-minion package can be installed directly on the network gear. Examples of such devices (also covered by NAPALM) include: Arista, Cumulus, Cisco IOS-XR or Cisco Nexus.
- Not always alive: in certain less dynamic environments, maintaining the remote connection permanently open with the network device is not always beneficial. In those particular cases, the user can select to initialize the connection only when needed, by specifying the field `always_alive: false` in the *`proxy configuration`* or using the *`proxy_always_alive`* option.
- Proxy keepalive: due to external factors, the connection with the remote device can be dropped, e.g.: packet loss, idle time (no commands issued within a couple of minutes or seconds), or simply the device decides to kill the process. In 2017.7.0 we have introduced the functionality to re-establish the connection. One can disable this feature through the *`proxy_keep_alive`* option and adjust the polling frequency speciying a custom value for *`proxy_keep_alive_interval`*, in minutes.

New modules:

- *`Netconfig state module`* - Manage the configuration of network devices using arbitrary templates and the Salt-specific advanced templating methodologies.
- `Network ACL execution module` - Generate and load ACL (firewall) configuration on network devices.
- *`Network ACL state`* - Manage the firewall configuration. It only requires writing the pillar structure correctly!
- *`NAPALM YANG execution module`* - Parse, generate and load native device configuration in a standard way, using the OpenConfig/IETF models. This module contains also helpers for the states.
- `NAPALM YANG state module` - Manage the network device configuration according to the YANG models (OpenConfig or IETF).
- *`NET finder`* - Runner to find details easily and fast. It's smart enough to know what you are looking for. It will search in the details of the network interfaces, IP addresses, MAC address tables, ARP tables and LLDP neighbors.
- *`BGP finder`* - Runner to search BGP neighbors details.
- *`NAPALM syslog`* - Engine to import events from the napalm-logs library into the Salt event bus. The events are based on the syslog messages from the network devices and structured following the OpenConfig/IETF YANG models.
- `NAPALM Helpers` - Generic helpers for NAPALM-related operations. For example, the `Compliance report` function can be used inside the state modules to compare the expected and the existing configuration.

New functions:

- *`Configuration getter`* - Return the whole configuration of the network device.
- *`Optics getter`* - Fetches the power usage on the various transceivers installed on the network device (in dBm).

New grains: *`Host`*, *`Host DNS`*, *`Username`* and *`Optional args`*.

Custom Refspecs in GitFS / git_pillar / winrepo

It is now possible to specify the refspecs to use when fetching from remote repositories for GitFS, git_pillar, and winrepo. More information on how this feature works can be found *here* in the GitFS Walkthrough. The git_pillar and winrepo versions of this feature work the same as their GitFS counterpart.

git_pillar "mountpoints" Feature Added

See *here* for detailed documentation.

Big Improvements to Docker Support

The old `docker` state and execution modules have been moved to salt-contrib. The `dockerng` execution module has been renamed to `docker` and now serves as Salt's official Docker execution module.

The old `dockerng` state module has been split into 4 state modules:

- *`docker_container`* - States to manage Docker containers
- *`docker_image`* - States to manage Docker images
- *`docker_volume`* - States to manage Docker volumes
- *`docker_network`* - States to manage Docker networks

The reason for this change was to make states and requisites more clear. For example, imagine this SLS:

```
myuser/appimage:
  docker.image_present:
    - sls: docker.images.appimage

myapp:
  docker.running:
    - image: myuser/appimage
    - require:
      - docker: myuser/appimage
```

The new syntax would be:

```
myuser/appimage:
  docker_image.present:
    - sls: docker.images.appimage

myapp:
  docker_container.running:
    - image: myuser/appimage
    - require:
      - docker_image: myuser/appimage
```

This is similar to how Salt handles MySQL, MongoDB, Zabbix, and other cases where the same execution module is used to manage several different kinds of objects (users, databases, roles, etc.).

Note: With the Moby announcement coming at this year's DockerCon, Salt's *`docker`* execution module (as well as the state modules) work interchangeably when **docker** is replaced with **moby** (e.g. *`moby_container.running`*, *`moby_image.present`*, *`moby.inspect_container`*, etc.)

The old syntax will continue to work until the **2019.2.0** release of Salt. The old `dockerng` naming will also continue to work until that release, so no immediate changes need to be made to your SLS files (unless you were still using the old docker states that have been moved to salt-contrib).

The *`docker_container.running`* state has undergone a significant change in how it determines whether or not a container needs to be replaced. Rather than comparing individual arguments to their corresponding values in the named container, a temporary container is created (but not started) using the passed arguments. The two containers are then compared to each other to determine whether or not there are changes, and if so, the old container is stopped and destroyed, and the temporary container is renamed and started.

Salt still needs to translate arguments into the format which docker-py expects, but if it does not properly do so, the *skip_translate* argument can be used to skip input translation on an argument-by-argument basis, and you can then format your SLS file to pass the data in the format that the docker-py expects. This allows you to work around any changes in Docker's API or issues with the input translation, and continue to manage your Docker containers using Salt. Read the documentation for *skip_translate* for more information.

Note: When running the `docker_container.running` state for the first time after upgrading to 2017.7.0, your container(s) may be replaced. The changes may show diffs for certain parameters which say that the old value was an empty string, and the new value is `None`. This is due to the fact that in prior releases Salt was passing empty strings for these values when creating the container if they were undefined in the SLS file, where now Salt simply does not pass any arguments not explicitly defined in the SLS file. Subsequent runs of the state should not replace the container if the configuration remains unchanged.

New SSH Cache Roster

The `SSH cache Roster` has been rewritten from scratch to increase its usefulness. The new roster supports all minion matchers, so it is now possible to target minions identically through *salt* and *salt-ssh*.

Using the new `roster_order` configuration syntax it's now possible to compose a roster out of any combination of grains, pillar and mine data and even Salt SDB URLs. The new release is also fully IPv4 and IPv6 enabled and even has support for CIDR ranges.

Salt-SSH Default Options

Defaults for rosters can now be set, so that they don't have to be set on every entry in a roster or specified from the commandline.

The new option is `roster_defaults` and is specified in the master config file:

```
roster_defaults:
  user: daniel
  sudo: True
  priv: /root/.ssh/id_rsa
  tty: True
```

Blacklist or Whitelist Extmod Sync

The modules that are synced to minions can now be limited.

The following configuration options have been added for the master:

- `extmod_whitelist`
- `extmod_blacklist`

and for the minion:

- `extmod_whitelist`
- `extmod_blacklist`

Additional Features

- The *mine.update* function has a new optional argument `mine_functions` that can be used to refresh mine functions at a more specific interval than scheduled using the `mine_interval` option. However, this argument can be used by explicit schedule. For example, if we need the mines for `net.lldp` to be refreshed every 12 hours:

```
schedule:
  lldp_mine_update:
    function: mine.update
    kwargs:
      mine_functions:
        net.lldp: []
    hours: 12
```

- The `salt` runner has a new function: *salt.execute*. It is mainly a shortcut to facilitate the execution of various functions from other runners, e.g.:

```
ret1 = __salt__['salt.execute']('*', 'mod.fun')
```

New Modules

Beacons

- *salt.beacons.log*

Cache

- *salt.cache.redis_cache*

Engines

- *salt.engines.stalekey*
- *salt.engines.junos_syslog*
- *salt.engines.napalm_syslog*

Execution modules

- `salt.modules.apk`
- *salt.modules.at_solaris*
- *salt.modules.boto_kinesis*
- *salt.modules.boto3_elasticache*
- *salt.modules.boto3_route53*
- *salt.modules.capirca_acl*
- *salt.modules.freebsd_update*
- *salt.modules.grafana4*

- *salt.modules.heat*
- *salt.modules.icinga2*
- *salt.modules.kubernetesmod*
- *salt.modules.logmod*
- *salt.modules.mattermost*
- `salt.modules.namecheap_dns`
- *salt.modules.namecheap_domains*
- `salt.modules.namecheap_ns`
- *salt.modules.namecheap_users*
- *salt.modules.namecheap_ssl*
- `salt.modules.napalm`
- `salt.modules.napalm_acl`
- *salt.modules.napalm_yang_mod*
- *salt.modules.pdbedit*
- *salt.modules.solrcloud*
- *salt.modules.statuspage*
- *salt.modules.zonecfg*
- *salt.modules.zoneadm*

Grains

- *salt.grains.metadata*
- `salt.grains.mdata`

Outputters

- *salt.output.table_out*

Pillar

- *salt.pillar.postgres*
- *salt.pillar.vmware_pillar*

Returners

- *salt.returners.mattermost_returner*
- *salt.returners.highstate_return*

Roster

- *salt.roster.cache*

Runners

- *salt.runners.bgp*
- *salt.runners.mattermost*
- *salt.runners.net*

SDB

- *salt.sdb.yaml*
- *salt.sdb.tism*
- *salt.sdb.cache*

States

- *salt.states.boto_kinesis*
- salt.states.boto_efs
- *salt.states.boto3_elasticache*
- *salt.states.boto3_route53*
- *salt.states.docker_container*
- *salt.states.docker_image*
- *salt.states.docker_network*
- *salt.states.docker_volume*
- *salt.states.elasticsearch*
- *salt.states.grafana4_dashboard*
- *salt.states.grafana4_datasource*
- *salt.states.grafana4_org*
- *salt.states.grafana4_user*
- *salt.states.heat*
- *salt.states.icinga2*
- *salt.states.influxdb_continuous_query*
- *salt.states.influxdb_retention_policy*
- *salt.states.kubernetes*
- *salt.states.logadm*
- *salt.states.logrotate*
- *salt.states.msteams*

- *`salt.states.netacl`*
- *`salt.states.netconfig`*
- `salt.states.netyang`
- `salt.states.nix`
- *`salt.states.pdbedit`*
- *`salt.states.solrcloud`*
- *`salt.states.statuspage`*
- *`salt.states.vault`*
- *`salt.states.win_wua`*
- *`salt.states.zone`*

Deprecations

General Deprecations

- Removed support for aliasing `cmd.run` to `cmd.shell`.
- Removed support for Dulwich from *GitFS*.
- Beacon configurations should be lists instead of dictionaries.
- The `PidfileMixin` has been removed. Please use `DaemonMixIn` instead.
- The `use_pending` argument was removed from the `salt.utils.event.get_event` function.
- The `pending_tags` argument was removed from the `salt.utils.event.get_event` function.

Configuration Option Deprecations

- The `client_acl` configuration option has been removed. Please use `publisher_acl` instead.
- The `client_acl_blacklist` configuration option has been removed. Please use `publisher_acl_blacklist` instead.
- The `win_gitrepos` configuration option has been removed. Please use the `winrepo_remotes` option instead.
- The `win_repo` configuration option has been removed. Please use `winrepo_dir` instead.
- The `win_repo_mastercachefile` configuration option has been removed. Please use the `winrepo_cachefile` option instead.

Module Deprecations

The `git` execution module had the following changes:

- The `fmt` argument was removed from the `archive` function. Please use `format` instead.
- The `repository` argument was removed from the `clone` function. Please use `url` instead.
- The `is_global` argument was removed from the `config_set` function. Please use `global` instead.
- The `branch` argument was removed from the `merge` function. Please use `rev` instead.

- The `branch` argument was removed from the `push` function. Please use `rev` instead.

The `glusterfs` execution module had the following functions removed:

- `create`: Please use `create_volume` instead.
- `delete`: Please use `delete_volume` instead.
- `list_peers`: Please use `peer_status` instead.

The `htpasswd` execution module had the following function removed:

- `useradd_all`: Please use `useradd` instead.

The `img` execution module has been removed. All of its associated functions were marked for removal in the 2017.7.0 release. The functions removed in this module are mapped as follows:

- `mount_image`/`mnt_image`: Please use `mount.mount` instead.
- `umount_image`: Please use `mount.umount` instead.
- `bootstrap`: Please use `genesis.bootstrap` instead.

The `smartos_virt` execution module had the following functions removed:

- `create`: Please use `start` instead.
- `destroy` Please use `stop` instead.
- `list_vms`: Please use `list_domains` instead.

The `virt` execution module had the following functions removed:

- `create`: Please use `start` instead.
- `destroy` Please use `stop` instead.
- `list_vms`: Please use `list_domains` instead.

The `virtualenv_mod` execution module had the following changes:

- The `package_or_requirement` argument was removed from both the `get_resource_path` and the `get_resource_content` functions. Please use `package` instead.
- The `resource_name` argument was removed from both the `get_resource_path` and `get_resource_content` functions. Please use `resource` instead.

The `win_repo` execution module had the following changes:

- The `win_repo_source_dir` option was removed from the `win_repo` module. Please use `winrepo_source_dir` instead.

The `xapi` execution module had the following functions removed:

- `create`: Please use `start` instead.
- `destroy`: Please use `stop` instead.
- `list_vms`: Please use `list_domains` instead.

The `zypper` execution module had the following function removed:

- `info`: Please use `info_available` instead.

Pillar Deprecations

- Support for the `raw_data` argument for the file_tree ext_pillar has been removed. Please use `keep_newline` instead.
- SQLite3 database connection configuration previously had keys under pillar. This legacy compatibility has been removed.

Proxy Minion Deprecations

- The `proxy_merge_grains_in_module` default has been switched from `False` to `True`.

Salt-API Deprecations

- The `SaltAPI.run()` function has been removed. Please use the `SaltAPI.start()` function instead.

Salt-Cloud Deprecations

- Support for using the keyword `provider` in salt-cloud provider config files has been removed. Please use `driver` instead. The `provider` keyword should now only be used in cloud profile config files.

Salt-SSH Deprecations

- The `wipe_ssh` option for `salt-ssh` has been removed. Please use the `ssh_wipe` option instead.

State Deprecations

The `apache_conf` state had the following functions removed:

- `disable`: Please use `disabled` instead.
- `enable`: Please use `enabled` instead.

The `apache_module` state had the following functions removed:

- `disable`: Please use `disabled` instead.
- `enable`: Please use `enabled` instead.

The `apache_site` state had the following functions removed:

- `disable`: Please use `disabled` instead.
- `enable`: Please use `enabled` instead.

The `chocolatey` state had the following functions removed:

- `install`: Please use `installed` instead.
- `uninstall`: Please use `uninstalled` instead.

The `git` state had the following changes:

- The `config` function was removed. Please use `config_set` instead.
- The `is_global` option was removed from the `config_set` function. Please use `global` instead.

- The `always_fetch` option was removed from the `latest` function, as it no longer has any effect. Please see the *2015.8.0* release notes for more information.
- The `force` option was removed from the `latest` function. Please use `force_clone` instead.
- The `remote_name` option was removed from the `latest` function. Please use `remote` instead.

The `glusterfs` state had the following function removed:

- `created`: Please use `volume_present` instead.

The `openvswitch_port` state had the following change:

- The `type` option was removed from the `present` function. Please use `tunnel_type` instead.

Build Notes

Windows Installer Packages

Windows Installer packages have been patched with the following PR: 42347

28.2.21 Salt 2017.7.1 Release Notes

Version 2017.7.1 is a bugfix release for *2017.7.0*.

Statistics

- Total Merges: **16**
- Total Issue References: **12**
- Total PR References: **31**
- Contributors: **11** (Ch3LL, TiteiKo, garethgreenaway, gtmanfred, llua, rallytime, seedickcode, skizunov, terminalmage, twangboy, whiteinge)

Security Fix

CVE-2017-12791 Maliciously crafted minion IDs can cause unwanted directory traversals on the Salt-master

Correct a flaw in minion id validation which could allow certain minions to authenticate to a master despite not having the correct credentials. To exploit the vulnerability, an attacker must create a salt-minion with an ID containing characters that will cause a directory traversal. Credit for discovering the security flaw goes to: Vernhk@qq.com

Changelog for v2017.7.0..v2017.7.1

Generated at: 2018-05-26 20:28:44 UTC

- **ISSUE** saltstack/salt-jenkins#460: (Ch3LL) decorator tests failing on python3 (refs: #42548)
- **PR** #42595: (gtmanfred) make sure to pass arg as well @ *2017-07-28 16:21:58 UTC*
  - **PR** #42548: (gtmanfred) pass in empty kwarg for reactor (refs: #42595)
  - a50fe5433a Merge pull request #42595 from gtmanfred/2017.7.1
  - 8f73804b24 make sure to pass arg as well

- **PR** #42597: (rallytime) Back-port #42590 to 2017.7.1 @ *2017-07-28 00:20:01 UTC*
  - **PR** #42590: (TiteiKo) Fix missing tornado import (refs: #42597)
  - 3b583330de Merge pull request #42597 from rallytime/bp-42590
  - 8818b06f22 Fix missing tornado import
- **ISSUE** #42404: (gabekahen) [2017.7] file.managed with cmd_check "No such file or directory" (refs: #42411)
- **ISSUE** #33708: (pepinje) visudo check command leaves cache file in /tmp (refs: #38063, #42411)
- **PR** #42598: (rallytime) Back-port #42411 to 2017.7.1 @ *2017-07-28 00:19:13 UTC*
  - **PR** #42411: (seedickcode) Fix file.managed check_cmd file not found - Issue #42404 (refs: #42598)
  - **PR** #38063: (llua) tmp file clean up in file.manage - fix for #33708 (refs: #42411)
  - 76f1e53e10 Merge pull request #42598 from rallytime/bp-42411
  - 190cdb8693 Fix file.managed check_cmd file not found - Issue #42404
- **PR** #42564: (rallytime) Back-port #42555 to 2017.7.1 @ *2017-07-26 17:32:02 UTC*
  - **PR** #42555: (Ch3LL) add changelog to 2017.7.1 release notes (refs: #42564)
  - 5c7def9a43 Merge pull request #42564 from rallytime/bp-42555
  - 7bcaa5a4cc small markup fix for title
  - d066b599ca add changelog to 2017.7.1 release notes
- **ISSUE** saltstack/salt-jenkins#460: (Ch3LL) decorator tests failing on python3 (refs: #42548)
- **PR** #42548: (gtmanfred) pass in empty kwarg for reactor (refs: #42595) @ *2017-07-26 00:41:20 UTC*
  - 711b742c54 Merge pull request #42548 from gtmanfred/2017.7.1
  - 0257c1dc32 pass in empty kwarg for reactor
  - b948e980d2 update chunk, not kwarg in chunk
- **ISSUE** #42519: (xuhcc) Error when installing package from file under Arch Linux (refs: #42522)
- **PR** #42522: (gtmanfred) pacman wildcard is only for repository installs @ *2017-07-24 20:51:05 UTC*
  - 50c1635dcc Merge pull request #42522 from gtmanfred/2017.7.1
  - 7787fb9e1b pacman wildcard is only for repository installs
- **PR** #42508: (rallytime) Back-port #42474 to 2017.7.1 @ *2017-07-24 20:49:51 UTC*
  - **PR** #42474: (whiteinge) Cmd arg kwarg parsing test (refs: #42508)
  - **PR** #39646: (terminalmage) Handle deprecation of passing string args to load_args_and_kwargs (refs: #42474)
  - 05c07ac049 Merge pull request #42508 from rallytime/bp-42474
  - 76fb074433 Add a test.arg variant that cleans the pub kwargs by default
  - 624f63648e Lint fixes
  - d246a5fc61 Add back support for string kwargs
  - 854e098aa0 Add LocalClient.cmd test for arg/kwarg parsing
- **ISSUE** #42427: (grichmond-salt) Issue Passing Variables created from load_json as Inline Pillar Between States (refs: #42435)
- **PR** #42472: (rallytime) Back-port #42435 to 2017.7.1 @ *2017-07-24 15:11:13 UTC*

  - **PR** #42435: (terminalmage) Modify our custom YAML loader to treat unicode literals as unicode strings (refs: #42472)
  - 95fe2558e4 Merge pull request #42472 from rallytime/bp-42435
  - 5c47af5b98 Modify our custom YAML loader to treat unicode literals as unicode strings
- **ISSUE** #42374: (tyhunt99) [2017.7.0] salt-run mange.versions throws exception if minion is offline or unresponsive (refs: #42436)
- **PR** #42473: (rallytime) Back-port #42436 to 2017.7.1 @ *2017-07-24 15:10:29 UTC*
  - **PR** #42436: (garethgreenaway) Fixes to versions function in manage runner (refs: #42473)
  - 5b99d45f54 Merge pull request #42473 from rallytime/bp-42436
  - 82ed919803 Updating the versions function inside the manage runner to account for when a minion is offline and we are unable to determine it's version.
- **ISSUE** #42381: (zebooka) Git.detached broken in 2017.7.0 (refs: #42399)
- **ISSUE** #38878: (tomlaredo) [Naming consistency] git.latest "rev" option VS git.detached "ref" option (refs: #38898)
- **PR** #42471: (rallytime) Back-port #42399 to 2017.7.1 @ *2017-07-24 15:09:50 UTC*
  - **PR** #42399: (rallytime) Update old "ref" references to "rev" in git.detached state (refs: #42471)
  - **PR** #38898: (terminalmage) git.detached: rename ref to rev for consistency (refs: #42399)
  - 3d1a2d3f9f Merge pull request #42471 from rallytime/bp-42399
  - b9a4669e5a Update old "ref" references to "rev" in git.detached state
- **ISSUE** #42400: (Enquier) Conflict in execution of passing pillar data to orch/reactor event executions 2017.7.0 (refs: #42031)
- **PR** #42470: (rallytime) Back-port #42031 to 2017.7.1 @ *2017-07-24 15:09:30 UTC*
  - **PR** #42031: (skizunov) Fix: Reactor emits critical error (refs: #42470)
  - 09766bccbc Merge pull request #42470 from rallytime/bp-42031
  - 0a0c6287a4 Fix: Reactor emits critical error
- **ISSUE** #41949: (jrporcaro) Event returner doesn't work with Windows Master (refs: #42027)
- **PR** #42469: (rallytime) Back-port #42027 to 2017.7.1 @ *2017-07-21 22:41:02 UTC*
  - **PR** #42027: (gtmanfred) import salt.minion for EventReturn for Windows (refs: #42469)
  - d7b172a15b Merge pull request #42469 from rallytime/bp-42027
  - ed612b4ee7 import salt.minion for EventReturn for Windows
- **PR** #42466: (rallytime) Back-port #42452 to 2017.7.1 @ *2017-07-21 19:41:24 UTC*
  - **PR** #42452: (Ch3LL) update windows urls to new py2/py3 naming scheme (refs: #42466)
  - 8777b1a825 Merge pull request #42466 from rallytime/bp-42452
  - c10196f68c update windows urls to new py2/py3 naming scheme
- **PR** #42439: (rallytime) Back-port #42409 to 2017.7.1 @ *2017-07-21 17:38:10 UTC*
  - **PR** #42409: (twangboy) Add Scripts to build Py3 on Mac (refs: #42439)
  - fceaaf41d0 Merge pull request #42439 from rallytime/bp-42409
  - 8176964b41 Remove build and dist, sign pkgs

  - 2c14d92a07 Fix hard coded pip path
  - 82fdd7c2e1 Add support for Py3
  - 2478447246 Update Python and other reqs
- **ISSUE** #42403: (astronouth7303) [2017.7] Pillar empty when state is applied from orchestrate (refs: #42433)
- **PR** #42441: (rallytime) Back-port #42433 to 2017.7.1 @ *2017-07-21 17:37:01 UTC*
  - **PR** #42433: (terminalmage) Only force saltenv/pillarenv to be a string when not None (refs: #42441)
  - 660400560b Merge pull request #42441 from rallytime/bp-42433
  - 17f347123a Only force saltenv/pillarenv to be a string when not None

28.2.22 Salt 2017.7.2 Release Notes

Version 2017.7.2 is a bugfix release for *2017.7.0*.

Statistics

- Total Merges: **329**
- Total Issue References: **73**
- Total PR References: **236**
- Contributors: **47** (Ch3LL, CorvinM, DmitryKuzmenko, Giandom, Mapel88, Mareo, SuperPommeDeTerre, The-Loeki, abulford, amendlik, blarghmatey, brejoc, cachedout, carsonoid, cro, damon-atkins, darcoli, dmurphy18, frankiexyz, garethgreenaway, gtmanfred, hibbert, isbm, ixs, jettero, jmarinaro, justincbeard, kkoppel, llua, lomeroe, m03, mcalmer, mirceaulinic, morganwillcock, nhavens, pabloh007, rallytime, seedickcode, shengis, skizunov, terminalmage, the-glu, thusoy, twangboy, vitaliyf, vutny, whiteinge)

Security Fix

CVE-2017-14695 Directory traversal vulnerability in minion id validation in SaltStack. Allows remote minions with incorrect credentials to authenticate to a master via a crafted minion ID. Credit for discovering the security flaw goes to: Julian Brost (julian@0x4a42.net)

CVE-2017-14696 Remote Denial of Service with a specially crafted authentication request. Credit for discovering the security flaw goes to: Julian Brost (julian@0x4a42.net)

Changelog for v2017.7.1..v2017.7.2

Generated at: 2018-05-26 21:06:12 UTC

- **PR** #43868: (rallytime) Back-port #43847 to 2017.7.2 @ *2017-10-03 12:00:52 UTC*
  - **PR** #43847: (cachedout) Fix to module.run (refs: #43868)
  - dd0b3388cf Merge pull request #43868 from rallytime/bp-43847
  - e21d8e9583 Use six.iterkeys() instead of dict.keys()
  - c297ae5557 Improve failures for module.run states
  - 782e67c199 Lint
  - a6c2d78518 Fix typo found by @s0undt3ch

  - 0cac15e502 Fix to module.run [WIP]
- **PR** #43871: (rallytime) Add updated release notes to 2017.7.2 branch @ *2017-10-03 11:59:29 UTC*
  - 47af4ae38a Merge pull request #43871 from rallytime/update-release-notes
  - 2337904656 Add updated release notes to 2017.7.2 branch
- **PR** #43756: (gtmanfred) split build and install for pkg osx @ *2017-09-26 20:51:28 UTC*
  - 88414d5f73 Merge pull request #43756 from gtmanfred/2017.7.2
  - f7df41fa94 split build and install for pkg osx
- **ISSUE** #43077: (Manoj2087) Issue with deleting key via wheel (refs: #43330)
- **PR** #43585: (rallytime) Back-port #43330 to 2017.7.2 @ *2017-09-19 17:33:34 UTC*
  - **PR** #43330: (terminalmage) Fix reactor regression + unify reactor config schema (refs: #43585)
  - 89f629233f Merge pull request #43585 from rallytime/bp-43330
  - c4f693bae8 Merge branch '2017.7.2' into bp-43330
- **ISSUE** #43447: (UtahDave) When using Syndic with Multi Master the top level master doesn't reliably get returns from lower minion. (refs: #43526)
- **PR** #43586: (rallytime) Back-port #43526 to 2017.7.2 @ *2017-09-19 15:36:27 UTC*
  - **PR** #43526: (DmitryKuzmenko) Forward events to all masters syndic connected to (refs: #43586)
  - abb7fe4422 Merge pull request #43586 from rallytime/bp-43526
  - e076e9b634 Forward events to all masters syndic connected to.
  - 7abd07fa07 Simplify client logic
  - b5f10696c2 Improve the reactor documentation
  - 7a2f12b96a Include a better example for reactor in master conf file
  - 531cac610e Rewrite the reactor unit tests
  - 2a35ab7f39 Unify reactor configuration, fix caller reactors
  - 4afb179bad Un-deprecate passing kwargs outside of 'kwarg' param
- **PR** #43551: (twangboy) Fix preinstall script on OSX for 2017.7.2 @ *2017-09-18 18:35:35 UTC*
  - 3d3b09302d Merge pull request #43551 from twangboy/osx_fix_preinstall_2017.7.2
  - c3d9fb63f0 Merge branch '2017.7.2' into osx_fix_preinstall_2017.7.2
- **PR** #43509: (rallytime) Back-port #43333 to 2017.7.2 @ *2017-09-15 21:21:40 UTC*
  - **PR** #43333: (damon-atkins) Docs are wrong cache_dir (bool) and cache_file (str) cannot be passed as params + 1 bug (refs: #43509)
  - 24691da888 Merge pull request #43509 from rallytime/bp-43333-2017.7.2
  - b3dbafb035 Update doco
  - 5cdcdbf428 Update win_pkg.py
  - c3e16661c3 Docs are wrong cache_dir (bool) and cache_file (str) cannot be passed on the cli (#2)
  - f33395f1ee Fix logic in */etc/paths.d/salt* detection
- **PR** #43440: (rallytime) Back-port #43421 to 2017.7.2 @ *2017-09-11 20:59:53 UTC*

  - **PR** #43421: (gtmanfred) Revert "Reduce fileclient.get_file latency by merging _file_find and … (refs: #43440)
  - 8964cacbf8 Merge pull request #43440 from rallytime/bp-43421
  - ea6e661755 Revert "Reduce fileclient.get_file latency by merging _file_find and _file_hash"
- **PR** #43377: (rallytime) Back-port #43193 to 2017.7.2 @ *2017-09-11 15:32:23 UTC*
  - **PR** #43193: (jettero) Prevent spurious "Template does not exist" error (refs: #43377)
  - **PR** #39516: (jettero) Prevent spurious "Template does not exist" error (refs: #43193)
  - 7fda186b18 Merge pull request #43377 from rallytime/bp-43193
  - 842b07fd25 Prevent spurious "Template does not exist" error
- **ISSUE** #42459: (iavael) Broken ldap groups retrieval in salt.auth.ldap after upgrade to 2017.7 (refs: #43283)
- **PR** #43315: (rallytime) Back-port #43283 to 2017.7.2 @ *2017-09-05 20:04:25 UTC*
  - **PR** #43283: (DmitryKuzmenko) Fix ldap token groups auth. (refs: #43315)
  - 85dba1e898 Merge pull request #43315 from rallytime/bp-43283
  - f29f5b0cce Fix for tests: don't require 'groups' in the eauth token.
  - 56938d5bf2 Fix ldap token groups auth.
- **ISSUE** #43259: (mahesh21) NameError: global name '__opts__' is not defined (refs: #43266)
- **PR** #43266: (gtmanfred) switch virtualbox cloud driver to use __utils__ @ *2017-08-30 18:36:20 UTC*
  - 26ff8088cb Merge pull request #43266 from gtmanfred/virtualbox
  - 382bf92de7 switch virtualbox cloud driver to use __utils__
- **ISSUE** #42936: (Mapel88) bug in win_iis module & state - container_setting (refs: #43073)
- **PR** #43073: (Mapel88) Fix bug #42936 - win_iis module container settings @ *2017-08-30 18:34:37 UTC*
  - ee209b144c Merge pull request #43073 from Mapel88/patch-2
  - b1a3d15b28 Remove trailing whitespace for linter
  - 25c8190e48 Fix pylint errors
  - 1eba8c4b8e Fix pylint errors
  - 290d7b54af Fix plint errors
  - f4f32421ab Fix plint errors
  - ec20e9a19a Fix bug #43110 - win_iis module
  - 009ef6686b Fix dictionary keys from string to int
  - dc793f9a05 Fix bug #42936 - win_iis state
  - 13404a47b5 Fix bug #42936 - win_iis module
- **PR** #43254: (twangboy) Fix *unit.modules.test_inspect_collector* on Windows @ *2017-08-30 15:46:07 UTC*
  - ec1bedc646 Merge pull request #43254 from twangboy/win_fix_test_inspect_collector
  - b401340e6c Fix *unit.modules.test_inspect_collector* on Windows
- **ISSUE** #43241: (mirceaulinic) Error whilst collecting napalm grains (refs: #43255)
- **PR** #43255: (gtmanfred) always return a dict object @ *2017-08-30 14:47:15 UTC*

  - 1fc7307735 Merge pull request #43255 from gtmanfred/2017.7
  - 83b0bab34b opt_args needs to be a dict
- **PR** #43229: (twangboy) Bring changes from #43228 to 2017.7 @ *2017-08-30 14:26:55 UTC*
  - **PR** #43228: (twangboy) Win fix pkg.install (refs: #43229)
  - fa904ee225 Merge pull request #43229 from twangboy/win_fix_pkg.install-2017.7
  - e007a1c26e Fix regex, add .
  - 23ec47c74c Add _ to regex search
  - b1788b1e5f Bring changes from #43228 to 2017.7
- **PR** #43251: (twangboy) Skips *unit.modules.test_groupadd* on Windows @ *2017-08-30 13:56:36 UTC*
  - 25666f88f7 Merge pull request #43251 from twangboy/win_skip_test_groupadd
  - 5185071d5a Skips *unit.modules.test_groupadd* on Windows
- **PR** #43256: (twangboy) Skip mac tests for user and group @ *2017-08-30 13:18:13 UTC*
  - a8e09629b2 Merge pull request #43256 from twangboy/win_skip_mac_tests
  - cec627a60b Skip mac tests for user and group
- **ISSUE** #42279: (dafyddj) win_lgpo matches multiple policies due to startswith() (refs: #43116, #43156, #43166, #43226)
- **PR** #43226: (lomeroe) Fixes for issues in PR #43166 @ *2017-08-29 19:05:39 UTC*
  - **PR** #43166: (lomeroe) Backport #43116 to 2017.7 (refs: #43226)
  - **PR** #43156: (lomeroe) Backport #43116 to 2017.7 (refs: #43166)
  - **PR** #43116: (lomeroe) Fix 42279 in develop (refs: #43166, #43156)
  - **PR** #39773: (twangboy) Make win_file use the win_dacl salt util (refs: #43226)
  - ac2189c870 Merge pull request #43226 from lomeroe/fix_43166
  - 0c424dc4a3 Merge branch '2017.7' into fix_43166
  - 324cfd8d1e correcting bad format statement in search for policy to be disabled (fix for #43166) verify that file exists before attempting to remove (fix for commits from #39773)
- **PR** #43227: (twangboy) Fix *unit.fileserver.test_gitfs* for Windows @ *2017-08-29 19:03:36 UTC*
  - 6199fb46dc Merge pull request #43227 from twangboy/win_fix_unit_test_gitfs
  - c956d24283 Fix is_windows detection when USERNAME missing
  - 869e8cc603 Fix *unit.fileserver.test_gitfs* for Windows
- **PR** #43217: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2017-08-28 16:36:28 UTC*
  - 6adc03e4b4 Merge pull request #43217 from rallytime/merge-2017.7
  - 3911df2f4b Merge branch '2016.11' into '2017.7'
  - 5308c27f9f Merge pull request #43202 from garethgreenaway/42642_2016_11_augeas_module_revert_fix
    - ef7e93eb3f Reverting this change due to it breaking other uses.
  - f16b7246e4 Merge pull request #43103 from aogier/43101-genesis-bootstrap
    - db94f3bb1c better formatting
    - e5cc667762 tests: fix a leftover and simplify some parts

    - 13e5997457 lint
    - 216ced69e5 allow comma-separated pkgs lists, quote args, test deb behaviour
    - d8612ae006 fix debootstrap and enhance packages selection/deletion via cmdline
  - 4863771428 Merge pull request #42663 from StreetHawkInc/fix_git_tag_check
    - 2b5af5b59d Remove refs/tags prefix from remote tags
    - 3f2e96e561 Convert set to list for serializer
    - 2728e5d977 Only include new tags in changes
    - 4b1df2f223 Exclude annotated tags from checks
    - 389c037285 Check remote tags before deciding to do a fetch #42329
- **PR** #43201: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2017-08-25 22:56:46 UTC*
  - a563a9422a Merge pull request #43201 from rallytime/merge-2017.7
  - d40eba6b37 Merge branch '2016.11' into '2017.7'
    - 4193e7f0a2 Merge pull request #43199 from corywright/disk-format-alias
      - f00d3a9ddc Add *disk.format* alias for *disk.format_*
    - 5471f9fe0c Merge pull request #43196 from gtmanfred/2016.11
      - ccd2241777 Pin request install to version
    - ace2715c60 Merge pull request #43178 from terminalmage/issue43143
      - 2640833400 git.detached: Fix traceback when rev is a SHA and is not present locally
    - 12e9507b9e Merge pull request #43179 from terminalmage/old-deprecation
      - 3adf8ad04b Fix missed deprecation
    - b595440d90 Merge pull request #43171 from terminalmage/salt-utils-warning
      - 7b5943a31a Add warning about adding new functions to salt/utils/__init__.py
    - 4f273cac4f Merge pull request #43173 from Ch3LL/add_branch_docs
      - 1b24244bd3 Add New Release Branch Strategy to Contribution Docs
- **PR** #42997: (twangboy) Fix *unit.test_test_module_names* for Windows @ *2017-08-25 21:19:11 UTC*
  - ce04ab4286 Merge pull request #42997 from twangboy/win_fix_test_module_names
  - 2722e9521d Use os.path.join to create paths
- **ISSUE** #26995: (jbouse) Issue with artifactory.downloaded and snapshot artifacts (refs: #43006)
- **PR** #43006: (SuperPommeDeTerre) Try to fix #26995 @ *2017-08-25 21:16:07 UTC*
  - c0279e491e Merge pull request #43006 from SuperPommeDeTerre/SuperPommeDeTerre-patch-#26995
  - 30dd6f5d12 Merge remote-tracking branch 'upstream/2017.7' into SuperPommeDeTerre-patch-#26995
  - f42ae9b8cd Merge branch 'SuperPommeDeTerre-patch-#26995' of https://github.com/SuperPommeDeTerre/salt into SuperPommeDeTerre-patch-#26995
    - 50ee3d5682 Merge remote-tracking branch 'remotes/origin/2017.7' into SuperPommeDeTerre-patch-#26995
    - 0b666e100b Fix typo.
    - 1b8729b3e7 Fix for #26995

  - e314102978 Fix typo.
  - db11e1985b Fix for #26995
- **ISSUE** #43162: (MorphBonehunter) docker_container.running interference with restart_policy (refs: #43184)
- **PR** #43184: (terminalmage) docker.compare_container: Perform boolean comparison when one side's value is null/None @ *2017-08-25 18:42:11 UTC*
  - b6c5314fe9 Merge pull request #43184 from terminalmage/issue43162
  - 081f42ad71 docker.compare_container: Perform boolean comparison when one side's value is null/None
- **PR** #43165: (mirceaulinic) Improve napalm state output in debug mode @ *2017-08-24 23:05:37 UTC*
  - 688125bb4f Merge pull request #43165 from cloudflare/fix-napalm-ret
  - c10717dc89 Lint and fix
  - 1cd33cbaa9 Simplify the loaded_ret logic
  - 0bbea6b04c Document the new compliance_report arg
  - 3a906109bd Include compliance reports
  - 3634055e34 Improve napalm state output in debug mode
- **PR** #43155: (terminalmage) Resolve image ID during container comparison @ *2017-08-24 22:09:47 UTC*
  - a6a327b1e5 Merge pull request #43155 from terminalmage/issue43001
  - 0186835ebf Fix docstring in test
  - a0bb654e46 Fixing lint issues
  - d5b2a0be68 Resolve image ID during container comparison
- **PR** #43170: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2017-08-24 19:22:26 UTC*
  - c071fd44c8 Merge pull request #43170 from rallytime/merge-2017.7
  - 3daad5a3a2 Merge branch '2016.11' into '2017.7'
    - 669b376abf Merge pull request #43151 from ushmodin/2016.11
      - c5841e2ade state.sls hangs on file.recurse with clean: True on windows
- **ISSUE** #43040: (darcoli) gitFS ext_pillar with branch name __env__ results in empty pillars (refs: #43041)
- **PR** #43168: (rallytime) Back-port #43041 to 2017.7 @ *2017-08-24 19:07:23 UTC*
  - **PR** #43041: (darcoli) Do not try to match pillarenv with __env__ (refs: #43168)
  - 034c325a09 Merge pull request #43168 from rallytime/bp-43041
  - d010b74b87 Do not try to match pillarenv with __env__
- **PR** #43172: (rallytime) Move new utils/__init__.py funcs to utils.files.py @ *2017-08-24 19:05:30 UTC*
  - **PR** #43056: (damon-atkins) safe_filename_leaf(file_basename) and safe_filepath(file_path_name) (refs: #43172)
  - d48938e6b4 Merge pull request #43172 from rallytime/move-utils-funcs
  - 5385c7901e Move new utils/__init__.py funcs to utils.files.py
- **ISSUE** #43043: (pabloh007) docker.save and docker.load problem (refs: #43061)
- **PR** #43061: (pabloh007) Have docker.save use the image name when valid if not use image id, i… @ *2017-08-24 16:32:02 UTC*

  - e60f586442 Merge pull request #43061 from pabloh007/fix-save-image-name-id
  - 0ffc57d1df Have docker.save use the image name when valid if not use image id, issue when loading and image is savid with id issue #43043
- **ISSUE** #42279: (dafyddj) win_lgpo matches multiple policies due to startswith() (refs: #43116, #43156, #43166, #43226)
- **PR** #43166: (lomeroe) Backport #43116 to 2017.7 (refs: #43226) @ *2017-08-24 15:01:23 UTC*
  - **PR** #43156: (lomeroe) Backport #43116 to 2017.7 (refs: #43166)
  - **PR** #43116: (lomeroe) Fix 42279 in develop (refs: #43166, #43156)
  - 9da57543f8 Merge pull request #43166 from lomeroe/bp-43116-2017.7
  - af181b3257 correct fopen calls from salt.utils for 2017.7
  - f74480f11e lint fix
  - ecd446fd55 track xml namespace to ensure policies w/duplicate IDs or Names do not conflict
  - 9f3047c420 add additional checks for ADM policies that have the same ADMX policy ID (#42279)
- **PR** #43056: (damon-atkins) safe_filename_leaf(file_basename) and safe_filepath(file_path_name) (refs: #43172) @ *2017-08-23 17:35:02 UTC*
  - 44b3caead1 Merge pull request #43056 from damon-atkins/2017.7
  - 08ded1546e more lint
  - 6e9c0957fb fix typo
  - ee41171c9f lint fixes
  - 8c864f02c7 fix missing imports
  - 964cebd954 safe_filename_leaf(file_basename) and safe_filepath(file_path_name)
- **PR** #43146: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2017-08-23 16:56:10 UTC*
  - 6ca9131a23 Merge pull request #43146 from rallytime/merge-2017.7
  - bcbe180fbc Merge branch '2016.11' into '2017.7'
    - ae9d2b7985 Merge pull request #42986 from renner/systemd-notify
      - 79c53f3f81 Fallback to systemd_notify_call() in case of socket.error
      - f1765472dd Notify systemd synchronously (via NOTIFY_SOCKET)
    - b420fbe618 Merge pull request #43037 from mcarlton00/fix-bhyve-grains
      - 73315f0cf0 Issue #43036 Bhyve virtual grain in Linux VMs
    - 0a86f2d884 Merge pull request #43100 from vutny/doc-add-missing-utils-ext
      - af743ff6c3 [DOCS] Add missing *utils* sub-dir listed for *extension_modules*
- **PR** #43123: (twangboy) Fix *unit.utils.test_which* for Windows @ *2017-08-23 16:01:39 UTC*
  - 03f652159f Merge pull request #43123 from twangboy/win_fix_test_which
  - ed97cff5f6 Fix *unit.utils.test_which* for Windows
- **ISSUE** #42505: (ikogan) selinux.fcontext_policy_present exception looking for selinux.filetype_id_to_string (refs: #43068)
- **PR** #43142: (rallytime) Back-port #43068 to 2017.7 @ *2017-08-23 15:56:48 UTC*

  - **PR** #43068: (ixs) Mark selinux._filetype_id_to_string as public function (refs: #43142)
  - 5a4fc07863 Merge pull request #43142 from rallytime/bp-43068
  - efc1c8c506 Mark selinux._filetype_id_to_string as public function
- **PR** #43038: (twangboy) Fix *unit.utils.test_url* for Windows @ *2017-08-23 13:35:25 UTC*
  - 0467a0e3bf Merge pull request #43038 from twangboy/win_unit_utils_test_url
  - 7f5ee55f57 Fix *unit.utils.test_url* for Windows
- **PR** #43097: (twangboy) Fix *group.present* for Windows @ *2017-08-23 13:19:56 UTC*
  - e9ccaa61d2 Merge pull request #43097 from twangboy/win_fix_group
  - 43b0360763 Fix lint
  - 9ffe315d7d Add kwargs
  - 4f4e34c79f Fix group state for Windows
- **PR** #43115: (rallytime) Back-port #42067 to 2017.7 @ *2017-08-22 20:09:52 UTC*
  - **PR** #42067: (vitaliyf) Removed several uses of name.split('.')[0] in SoftLayer driver. (refs: #43115)
  - 8140855627 Merge pull request #43115 from rallytime/bp-42067
  - 8a6ad0a9cf Fixed typo.
  - 9a5ae2bba1 Removed several uses of name.split('.')[0] in SoftLayer driver.
- **PR** #42962: (twangboy) Fix *unit.test_doc test* for Windows @ *2017-08-22 18:06:23 UTC*
  - 1e1a81036c Merge pull request #42962 from twangboy/win_unit_test_doc
  - 201ceae4c4 Fix lint, remove debug statement
  - 37029c1a16 Fix unit.test_doc test
- **PR** #42995: (twangboy) Fix malformed requisite for Windows @ *2017-08-22 16:50:01 UTC*
  - d347d1cf8f Merge pull request #42995 from twangboy/win_fix_invalid_requisite
  - 93390de88b Fix malformed requisite for Windows
- **PR** #43108: (rallytime) Back-port #42988 to 2017.7 @ *2017-08-22 16:49:27 UTC*
  - **PR** #42988: (thusoy) Fix broken negation in iptables (refs: #43108)
  - 1c7992a832 Merge pull request #43108 from rallytime/bp-42988
  - 1a987cb948 Fix broken negation in iptables
- **PR** #43107: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2017-08-22 16:11:25 UTC*
  - c6993f4a84 Merge pull request #43107 from rallytime/merge-2017.7
  - 328dd6aa23 Merge branch '2016.11' into '2017.7'
  - e2bf2f448e Merge pull request #42985 from DSRCorporation/bugs/15171_recursion_limit
    - 651b1bab09 Properly handle *prereq* having lost requisites.
  - e51333306c Merge pull request #43092 from mitodl/2016.11
    - d4b113acdf Fixed issue with silently passing all tests in Testinfra module
  - 77a443ce8e Merge pull request #43060 from twangboy/osx_update_pkg_scripts
    - ef8a14cdf9 Remove /opt/salt instead of /opt/salt/bin

    - 2dd62aa1da Add more information to the description
    - f44f5b70dc Only stop services if they are running
    - 3b62bf953c Remove salt from the path
    - ebdca3a0f5 Update pkg-scripts
  - 1b1b6da803 Merge pull request #43064 from terminalmage/issue42869
    - 093c0c2f77 Fix race condition in git.latest
  - 96e8e836d1 Merge pull request #43054 from lorengordon/fix/yumpkg/config-parser
    - 3b2cb81a72 fix typo in salt.modules.yumpkg
    - 38add0e4a2 break if leading comments are all fetched
    - d7f65dc7a7 fix configparser import & log if error was raised
    - ca1b1bb633 use configparser to parse yum repo file
- **PR** #42996: (twangboy) Fix *unit.test_stateconf* for Windows @ *2017-08-21 22:43:58 UTC*
  - f9b4976c02 Merge pull request #42996 from twangboy/win_fix_test_stateconf
  - 92dc3c0ece Use os.sep for path
- **PR** #43024: (twangboy) Fix *unit.utils.test_find* for Windows @ *2017-08-21 22:38:10 UTC*
  - 19fc644c9b Merge pull request #43024 from twangboy/win_unit_utils_test_find
  - fbe54c9a33 Remove unused import six (lint)
  - b04d1a2f18 Fix *unit.utils.test_find* for Windows
- **PR** #43088: (gtmanfred) allow docker util to be reloaded with reload_modules @ *2017-08-21 22:14:37 UTC*
  - 1a531169fc Merge pull request #43088 from gtmanfred/2017.7
  - 373a9a0be4 allow docker util to be reloaded with reload_modules
- **PR** #43091: (blarghmatey) Fixed issue with silently passing all tests in Testinfra module @ *2017-08-21 22:06:22 UTC*
  - 83e528f0b3 Merge pull request #43091 from mitodl/2017.7
  - b502560e61 Fixed issue with silently passing all tests in Testinfra module
- **PR** #41994: (twangboy) Fix *unit.modules.test_cmdmod* on Windows @ *2017-08-21 21:53:01 UTC*
  - 5482524270 Merge pull request #41994 from twangboy/win_unit_test_cmdmod
  - a5f7288ad9 Skip test that uses pwd, not available on Windows
- **ISSUE** #42873: (TheVakman) osquery Data Empty Upon Return / Reporting Not Installed (refs: #42933)
- **PR** #42933: (garethgreenaway) Fixes to osquery module @ *2017-08-21 20:48:31 UTC*
  - b33c4abc15 Merge pull request #42933 from garethgreenaway/42873_2017_7_osquery_fix
  - 8915e62bd9 Removing an import that is not needed.
  - 74bc377eb4 Updating the other function that uses cmd.run_all
  - e6a4619ec1 Better approach without using python_shell=True.
  - 5ac41f496d When running osquery commands through cmd.run we should pass python_shell=True to ensure everything is formatted right. #42873
- **PR** #43093: (gtmanfred) Fix ec2 list_nodes_full to work on 2017.7 @ *2017-08-21 20:21:21 UTC*

  - 53c2115769 Merge pull request #43093 from gtmanfred/ec2
  - c7cffb5a04 This block isn't necessary
  - b7283bcc6f _vm_provider_driver isn't needed anymore
- **ISSUE** #43085: (brejoc) Patch for Kubernetes module missing from 2017.7 and 2017.7.1 (refs: #43087)
- **PR** #43087: (rallytime) Back-port #42174 to 2017.7 @ *2017-08-21 18:40:18 UTC*
  - **PR** #42174: (mcalmer) kubernetes: provide client certificate authentication (refs: #43087)
  - 32f9ade4db Merge pull request #43087 from rallytime/bp-42174
  - cf6563645b add support for certificate authentication to kubernetes module
- **PR** #43029: (terminalmage) Normalize the salt caching API @ *2017-08-21 16:54:58 UTC*
  - 882fcd846f Merge pull request #43029 from terminalmage/fix-func-alias
  - f8f74a310c Update localfs cache tests to reflect changes to func naming
  - c4ae79b229 Rename other refs to cache.ls with cache.list
  - ee59d127e8 Normalize the salt caching API
- **ISSUE** #42843: (brejoc) Kubernetes module won't work with Kubernetes Python client > 1.0.2 (refs: #42845)
- **PR** #43039: (gtmanfred) catch ImportError for kubernetes.client import @ *2017-08-21 14:32:38 UTC*
  - **PR** #42845: (brejoc) API changes for Kubernetes version 2.0.0 (refs: #43039)
  - dbee735f6e Merge pull request #43039 from gtmanfred/kube
  - 7e269cb368 catch ImportError for kubernetes.client import
- **PR** #43058: (rallytime) Update release version number for jenkins.run function @ *2017-08-21 14:13:34 UTC*
  - c56a8499b3 Merge pull request #43058 from rallytime/fix-release-num
  - d7eef70df0 Update release version number for jenkins.run function
- **PR** #43051: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2017-08-18 17:05:57 UTC*
  - 7b0c94768a Merge pull request #43051 from rallytime/merge-2017.7
  - 153a463b86 Lint: Add missing blank line
  - 84829a6f8c Merge branch '2016.11' into '2017.7'
  - 43aa46f512 Merge pull request #43048 from rallytime/bp-43031
    - 35e45049e2 use a ruby gem that doesn't have dependencies
  - ad89ff3104 Merge pull request #43023 from terminalmage/fix-jenkins-xml-caching
    - 33fd8ff939 Update jenkins.py
    - fc306fc8c3 Add missing colon in *if* statement
    - 822eabcc81 Catch exceptions raised when making changes to jenkins
    - 91b583b493 Improve and correct exception raising
    - f096917a0e Raise an exception if we fail to cache the config xml
  - 2957467ed7 Merge pull request #43026 from rallytime/bp-43020
    - 0eb15a1f67 test with gem that appears to be abandoned
  - 4150b094fe Merge pull request #43033 from rallytime/bp-42760

    - 3e3f7f5d8e Catch TypeError thrown by m2crypto when parsing missing subjects in certificate files.
  - b124d3667e Merge pull request #43032 from rallytime/bp-42547
    - ea4d7f4176 Updated testinfra modules to work with more recent versions
  - a88386ad44 Merge pull request #43027 from pabloh007/fix-docker-save-push-2016-11
    - d0fd949f85 Fixes ignore push flag for docker.push module issue #42992
  - 51d16840bb Merge pull request #42890 from DSRCorporation/bugs/42627_salt-cp
    - cfddbf1c75 Apply code review: update the doc
    - afedd3b654 Typos and version fixes in the doc.
    - 9fedf6012e Fixed 'test_valid_docs' test.
    - 999388680c Make chunked mode in salt-cp optional (disabled by default).
  - b3c253cdfa Merge pull request #43009 from rallytime/merge-2016.11
    - 566ba4fe76 Merge branch '2016.3' into '2016.11'
      - 13b8637d53 Merge pull request #42942 from Ch3LL/2016.3.6_follow_up
      - f281e1795f move additional minion config options to 2016.3.8 release notes
      - 168604ba6b remove merge conflict
      - 8a07d95212 update release notes with cve number
      - 149633fdca Add release notes for 2016.3.7 release
      - 7a4cddcd95 Add clean_id function to salt.utils.verify.py
      - bbb1b29ccb Merge pull request #42954 from Ch3LL/latest_2016.3
      - b551e66744 [2016.3] Bump latest and previous versions
      - 5d5edc54b7 Merge pull request #42949 from Ch3LL/2016.3.7_docs
      - d75d3741f8 Add Security Notice to 2016.3.7 Release Notes
  - 37c63e7cf2 Merge pull request #43021 from terminalmage/fix-network-test
    - 4089b7b1bc Use socket.AF_INET6 to get the correct value instead of doing an OS check
  - 8f6423247c Merge pull request #43019 from rallytime/bootstrap_2017.08.17
    - 2f762b3a17 Update bootstrap script to latest stable: v2017.08.17
  - ff1caeee68 Merge pull request #43014 from Ch3LL/fix_network_mac
    - b8eee4401e Change AF_INET6 family for mac in test_host_to_ips
- **PR** #43035: (rallytime) [2017.7] Merge forward from 2017.7.1 to 2017.7 @ *2017-08-18 12:58:17 UTC*
  - **PR** #42948: (Ch3LL) [2017.7.1] Add clean_id function to salt.utils.verify.py (refs: #43035)
  - **PR** #42945: (Ch3LL) [2017.7] Add clean_id function to salt.utils.verify.py (refs: #43035)
  - d15b0ca937 Merge pull request #43035 from rallytime/merge-2017.7
  - 756128a896 Merge branch '2017.7.1' into '2017.7'
    - ab1b099730 Merge pull request #42948 from Ch3LL/2017.7.0_follow_up
- **ISSUE** #42989: (blbradley) GitFS GitPython performance regression in 2017.7.1 (refs: #43002)
- **PR** #43034: (rallytime) Back-port #43002 to 2017.7 @ *2017-08-17 23:18:16 UTC*

  - **PR** #43002: (the-glu) Try to fix #42989 (refs: #43034)
  - bcbb973a71 Merge pull request #43034 from rallytime/bp-43002
  - 350c0767dc Try to fix #42989 by doing sslVerify and refspecs for origin remote only if there is no remotes
- **ISSUE** #42375: (dragonpaw) salt.modules.*.__virtualname__ doens't work as documented. (refs: #42523, #42958)
- **PR** #42958: (gtmanfred) runit module should also be loaded as runit @ *2017-08-17 22:30:23 UTC*
  - 9182f55bbb Merge pull request #42958 from gtmanfred/2017.7
  - fd6874668b runit module should also be loaded as runit
- **PR** #43031: (gtmanfred) use a ruby gem that doesn't have dependencies (refs: #43048) @ *2017-08-17 22:26:25 UTC*
  - 5985cc4e8e Merge pull request #43031 from gtmanfred/test_gem
  - ba80a7d4b5 use a ruby gem that doesn't have dependencies
- **PR** #43030: (rallytime) Small cleanup to dockermod.save @ *2017-08-17 22:26:00 UTC*
  - 246176b1a6 Merge pull request #43030 from rallytime/dockermod-minor-change
  - d6a5e85632 Small cleanup to dockermod.save
- **ISSUE** #42992: (pabloh007) docker.save flag push does is ignored (refs: #42993, #43027)
- **PR** #42993: (pabloh007) Fixes ignored push flag for docker.push module issue #42992 @ *2017-08-17 18:50:37 UTC*
  - 160001120b Merge pull request #42993 from pabloh007/fix-docker-save-push
  - fe7554cfeb Fixes ignored push flag for docker.push module issue #42992
- **ISSUE** #42941: (danlsgiga) pkg.installed fails on installing from HTTPS rpm source (refs: #42967)
- **PR** #42967: (terminalmage) Fix bug in on_header callback when no Content-Type is found in headers @ *2017-08-17 18:48:52 UTC*
  - 9009a971b1 Merge pull request #42967 from terminalmage/issue42941
  - b838460816 Fix bug in on_header callback when no Content-Type is found in headers
- **ISSUE** #43008: (evelineraine) states.service.running always succeeds when watched state has changes (refs: #43016)
- **PR** #43016: (gtmanfred) service should return false on exception @ *2017-08-17 18:08:05 UTC*
  - 58f070d7a7 Merge pull request #43016 from gtmanfred/service
  - 21c264fe55 service should return false on exception
- **PR** #43020: (gtmanfred) test with gem that appears to be abandoned (refs: #43026) @ *2017-08-17 16:40:41 UTC*
  - 973d288eca Merge pull request #43020 from gtmanfred/test_gem
  - 0a1f40a664 test with gem that appears to be abandoned
- **PR** #42999: (garethgreenaway) Fixes to slack engine @ *2017-08-17 15:46:24 UTC*
  - 9cd0607fd4 Merge pull request #42999 from garethgreenaway/slack_engine_allow_editing_messages
  - 0ece2a8f0c Fixing a bug that prevented editing Slack messages and having the commands resent to the Slack engine.
- **PR** #43010: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2017-08-17 15:10:29 UTC*

  - 31627a9163 Merge pull request #43010 from rallytime/merge-2017.7
  - 8a0f948e4a Merge branch '2016.11' into '2017.7'
  - 1ee9499d28 Merge pull request #42968 from vutny/doc-salt-cloud-ref
    - 44ed53b1df [DOCS] Fix link to Salt Cloud Feature Matrix
  - 923f9741fe Merge pull request #42291 from vutny/fix-38839
    - 5f8f98a01f Fix #38839: remove *state* from Reactor runner kwags
  - c20bc7d515 Merge pull request #42940 from gtmanfred/2016.11
    - 253e216a8d fix IP address spelling
    - bd63074e7a create new ip address before checking list of allocated ips
  - d6496eca72 Merge pull request #42959 from rallytime/bp-42883
    - c6b9ca4b9e Lint fix: add missing space
    - 5597b1a30e Skip 2 failing tests in Python 3 due to upstream bugs
    - a0b19bdc27 Update account id value in boto_secgroup module unit test
    - 60b406e088 @mock_elb needs to be changed to @mock_elb_deprecated as well
    - 6ae1111295 Replace @mock_ec2 calls with @mock_ec2_deprecated calls
  - 6366e05d0d Merge pull request #42944 from Ch3LL/2016.11.6_follow_up
    - 7e0a20afca Add release notes for 2016.11.7 release
    - 63823f8c3e Add clean_id function to salt.utils.verify.py
  - 49d339c976 Merge pull request #42952 from Ch3LL/latest_2016.11
    - 74e7055d54 [2016.11] Bump latest and previous versions
  - b0d2e05a79 Merge pull request #42950 from Ch3LL/2016.11.7_docs
    - a6f902db40 Add Security Notice to 2016.11.77 Release Notes
  - c0ff69f88c Merge pull request #42836 from lyft/backport-utils.versions-to-2016.11
    - 86ce7004a2 Backport salt.utils.versions from develop to 2016.11
  - 64a79dd5ac Merge pull request #42919 from rallytime/bp-42871
    - 4e46c968e6 Update joyent.rst
  - bea8ec1098 Merge pull request #42918 from rallytime/bp-42848
    - cdb48126f7 Make lint happier.
    - 62eca9b00b Execute fire_master asynchronously in the main minion thread.
  - 52bce329cb Merge pull request #42861 from twangboy/win_pkg_install_salt
    - 0d3789f0c6 Fix pkg.install salt-minion using salt-call
  - b9f4f87aa5 Merge pull request #42798 from s-sebastian/2016.11
    - 1cc86592ed Update return data before calling returners
- **ISSUE** #42842: (Giandom) retreive kwargs passed with slack engine (refs: #42884)
- **PR** #42884: (Giandom) Convert to dict type the pillar string value passed from slack @ *2017-08-16 22:30:43 UTC*
  - 82be9dceb6 Merge pull request #42884 from Giandom/2017.7.1-fix-slack-engine-pillar-args

  - 80fd733c99 Update slack.py
- **PR** #42963: (twangboy) Fix *unit.test_fileclient* for Windows @ *2017-08-16 14:18:18 UTC*
  - 42bd553b98 Merge pull request #42963 from twangboy/win_unit_test_fileclient
  - e9febe4893 Fix unit.test_fileclient
- **PR** #42964: (twangboy) Fix *salt.utils.recursive_copy* for Windows @ *2017-08-16 14:17:27 UTC*
  - 7dddeeea8d Merge pull request #42964 from twangboy/win_fix_recursive_copy
  - 121cd4ef81 Fix *salt.utils.recursive_copy* for Windows
- **ISSUE** #42943: (mirceaulinic) *extension_modules* defaulting to */var/cache/minion* although running under proxy minion (refs: #42946)
- **PR** #42946: (mirceaulinic) extension_modules should default to $CACHE_DIR/proxy/extmods @ *2017-08-15 21:26:36 UTC*
  - 6da4d1d95e Merge pull request #42946 from cloudflare/px_extmods_42943
  - 73f9135340 extension_modules should default to /proxy/extmods
- **PR** #42945: (Ch3LL) [2017.7] Add clean_id function to salt.utils.verify.py (refs: #43035) @ *2017-08-15 18:04:20 UTC*
  - 95645d49f9 Merge pull request #42945 from Ch3LL/2017.7.0_follow_up
  - dcd92042e3 remove extra doc
  - 693a504ef0 update release notes with cve number
- **ISSUE** #42427: (grichmond-salt) Issue Passing Variables created from load_json as Inline Pillar Between States (refs: #42435)
- **PR** #42812: (terminalmage) Update custom YAML loader tests to properly test unicode literals @ *2017-08-15 17:50:22 UTC*
  - **PR** #42435: (terminalmage) Modify our custom YAML loader to treat unicode literals as unicode strings (refs: #42812)
  - 47ff9d5627 Merge pull request #42812 from terminalmage/yaml-loader-tests
  - 9d8486a894 Add test for custom YAML loader with unicode literal strings
  - a0118bcece Remove bytestrings and use textwrap.dedent for readability
- **PR** #42953: (Ch3LL) [2017.7] Bump latest and previous versions @ *2017-08-15 17:23:28 UTC*
  - 5d0c2198ac Merge pull request #42953 from Ch3LL/latest_2017.7
  - cbecf65823 [2017.7] Bump latest and previous versions
- **PR** #42951: (Ch3LL) Add Security Notice to 2017.7.1 Release Notes @ *2017-08-15 16:49:56 UTC*
  - 730e71db17 Merge pull request #42951 from Ch3LL/2017.7.1_docs
  - 1d8f827c58 Add Security Notice to 2017.7.1 Release Notes
- **PR** #42868: (carsonoid) Stub out required functions in redis_cache @ *2017-08-15 14:33:54 UTC*
  - c1c8cb9bfa Merge pull request #42868 from carsonoid/redisjobcachefix
  - 885bee2a7d Stub out required functions for redis cache
- **PR** #42810: (amendlik) Ignore error values when listing Windows SNMP community strings @ *2017-08-15 03:55:15 UTC*

  - e192d6e0af Merge pull request #42810 from amendlik/win-snmp-community
  - dc20e4651b Ignore error values when listing Windows SNMP community strings
- **PR** #42920: (cachedout) pid_race @ *2017-08-15 03:49:10 UTC*
  - a1817f1de3 Merge pull request #42920 from cachedout/pid_race
  - 5e930b8cbd If we catch the pid file in a transistory state, return None
- **PR** #42925: (terminalmage) Add debug logging to troubleshoot test failures @ *2017-08-15 03:47:51 UTC*
  - 11a33fe692 Merge pull request #42925 from terminalmage/f26-debug-logging
  - 8165f46165 Add debug logging to troubleshoot test failures
- **PR** #42913: (twangboy) Change service shutdown timeouts for salt-minion service (Windows) @ *2017-08-14 20:55:24 UTC*
  - a537197030 Merge pull request #42913 from twangboy/win_change_timeout
  - ffb23fbe47 Remove the line that wipes out the cache
  - a3becf8342 Change service shutdown timeouts
- **PR** #42800: (skizunov) Fix exception when master_type=disable @ *2017-08-14 20:53:38 UTC*
  - ca0555f616 Merge pull request #42800 from skizunov/develop6
  - fa5822009f Fix exception when master_type=disable
- **PR** #42679: (mirceaulinic) Add multiprocessing option for NAPALM proxy @ *2017-08-14 20:45:06 UTC*
  - 3af264b664 Merge pull request #42679 from cloudflare/napalm-multiprocessing
  - 9c4566db0c multiprocessing option tagged for 2017.7.2
  - 37bca1b902 Add multiprocessing option for NAPALM proxy
  - a2565ba8e5 Add new napalm option: multiprocessing
- **ISSUE** #42611: (nhavens) selinux.boolean state does not return changes (refs: #42612)
- **PR** #42657: (nhavens) back-port #42612 to 2017.7 @ *2017-08-14 19:42:26 UTC*
  - **PR** #42612: (nhavens) fix for issue #42611 (refs: #42657)
  - 4fcdab3ae9 Merge pull request #42657 from nhavens/2017.7
  - d73c4b55b7 back-port #42612 to 2017.7
- **PR** #42709: (whiteinge) Add token_expire_user_override link to auth runner docstring @ *2017-08-14 19:03:06 UTC*
  - d2b6ce327a Merge pull request #42709 from whiteinge/doc-token_expire_user_override
  - c7ea631558 Add more docs on the token_expire param
  - 4a9f6ba44f Add token_expire_user_override link to auth runner docstring
- **ISSUE** #42803: (gmcwhistler) master_type: str, not working as expected, parent salt-minion process dies. (refs: #42848)
- **ISSUE** #42753: (grichmond-salt) SaltReqTimeout Error on Some Minions when One Master in a Multi-Master Configuration is Unavailable (refs: #42848)
- **PR** #42848: (DmitryKuzmenko) Execute fire_master asynchronously in the main minion thread. (refs: #42918) @ *2017-08-14 18:28:38 UTC*
  - c6a7bf02e9 Merge pull request #42848 from DSRCorporation/bugs/42753_mmaster_timeout

  - 7f5412c19e Make lint happier.
  - ff66b7aaf0 Execute fire_master asynchronously in the main minion thread.
- **PR** #42911: (gtmanfred) cloud driver isn't a provider @ *2017-08-14 17:47:16 UTC*
  - 6a3279ea50 Merge pull request #42911 from gtmanfred/2017.7
  - 99046b441f cloud driver isn't a provider
- **PR** #42860: (skizunov) hash_and_stat_file should return a 2-tuple @ *2017-08-14 15:44:54 UTC*
  - 4456f7383d Merge pull request #42860 from skizunov/develop7
  - 5f85a03636 hash_and_stat_file should return a 2-tuple
- **PR** #42889: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2017-08-14 14:16:20 UTC*
  - c6ca7d639f Merge pull request #42889 from rallytime/merge-2017.7
  - fb7117f2ac Use salt.utils.versions.LooseVersion instead of distutils
  - 29ff19c587 Merge branch '2016.11' into '2017.7'
    - c15d0034fe Merge pull request #41977 from redmatter/fix-dockerng-network-ignores-test
      - 1cc2aa503a Fix dockerng.network_* ignoring of tests=True
    - 3b9c3c5671 Merge pull request #42886 from sarcasticadmin/adding_docs_salt_outputs
      - 744bf954ff Adding missing output flags to salt cli
    - e5b98c8a88 Merge pull request #42882 from gtmanfred/2016.11
      - da3402a53d make sure cmd is not run when npm isn't installed
    - 5962c9588b Merge pull request #42788 from amendlik/saltify-timeout
      - 928b523797 Remove waits and retries from Saltify deployment
    - 227ecddd13 Merge pull request #42877 from terminalmage/add-cron-state-virtual
      - f1de196740 Add virtual func for cron state module
    - ab9f6cef33 Merge pull request #42859 from terminalmage/gitpython-git-cli-note
      - 35e05c9515 Add note about git CLI requirement for GitPython to GitFS tutorial
    - 682b4a8d14 Merge pull request #42856 from gtmanfred/2016.11
      - b458b89fb8 skip cache_clean test if npm version is >= 5.0.0
    - 01ea854029 Merge pull request #42864 from whiteinge/syndic-log-root_dir
      - 4b1f55da9c Make syndic_log_file respect root_dir setting
- **PR** #42898: (mirceaulinic) Minor eos doc correction @ *2017-08-14 13:42:21 UTC*
  - 4b6fe2ee59 Merge pull request #42898 from mirceaulinic/patch-11
  - 93be79a135 Index eos under the installation instructions list
  - f903e7bc39 Minor eos doc correction
- **PR** #42883: (rallytime) Fix failing boto tests (refs: #42959) @ *2017-08-11 20:29:12 UTC*
  - 1764878754 Merge pull request #42883 from rallytime/fix-boto-tests
  - 6a7bf99848 Lint fix: add missing space
  - 43643227c6 Skip 2 failing tests in Python 3 due to upstream bugs

  - 7f46603e9c Update account id value in boto_secgroup module unit test
  - 7c1d493fdd @mock_elb needs to be changed to @mock_elb_deprecated as well
  - 3055e17ed5 Replace @mock_ec2 calls with @mock_ec2_deprecated calls
- **PR** #42885: (terminalmage) Move weird tearDown test to an actual tearDown @ *2017-08-11 19:14:42 UTC*
  - b21778efac Merge pull request #42885 from terminalmage/fix-f26-tests
  - 462d653082 Move weird tearDown test to an actual tearDown
- **ISSUE** #42870: (boltronics) webutil.useradd marked as deprecated:: 2016.3.0 by mistake? (refs: #42887)
- **PR** #42887: (rallytime) Remove extraneous ”deprecated” notation @ *2017-08-11 18:34:25 UTC*
  - 9868ab6f3b Merge pull request #42887 from rallytime/fix-42870
  - 71e7581a2d Remove extraneous ”deprecated” notation
- **PR** #42881: (gtmanfred) fix vmware for python 3.4.2 in salt.utils.vmware @ *2017-08-11 17:52:29 UTC*
  - da71f2a11b Merge pull request #42881 from gtmanfred/vmware
  - 05ecc6ac8d fix vmware for python 3.4.2 in salt.utils.vmware
- **ISSUE** #42843: (brejoc) Kubernetes module won’t work with Kubernetes Python client > 1.0.2 (refs: #42845)
- **PR** #42845: (brejoc) API changes for Kubernetes version 2.0.0 (refs: #43039) @ *2017-08-11 14:04:30 UTC*
  - c7750d5717 Merge pull request #42845 from brejoc/updates-for-kubernetes-2.0.0
  - 81674aa88a Version info in :optdepends: not needed anymore
  - 71995505bc Not depending on specific K8s version anymore
  - d8f7d7a7c0 API changes for Kubernetes version 2.0.0
- **PR** #42678: (frankiexyz) Add eos.rst in the installation guide @ *2017-08-11 13:58:37 UTC*
  - 459fdedc67 Merge pull request #42678 from frankiexyz/2017.7
  - 1598571f52 Add eos.rst in the installation guide
- **ISSUE** #42646: (gmacon) SPM fails to install multiple packages (refs: #42778)
- **PR** #42778: (gtmanfred) make sure to use the correct out_file @ *2017-08-11 13:44:48 UTC*
  - 4ce96eb1a1 Merge pull request #42778 from gtmanfred/spm
  - 7ef691e8da make sure to use the correct out_file
- **ISSUE** saltstack/salt-jenkins#480: (rallytime) [2017.7] PY3 Debian 8 has several vmware unit tests failing (refs: #42857)
- **PR** #42857: (gtmanfred) use older name if _create_unverified_context is unvailable @ *2017-08-11 13:37:59 UTC*
  - 3d05d89e09 Merge pull request #42857 from gtmanfred/vmware
  - c1f673eca4 use older name if _create_unverified_context is unvailable
- **PR** #42866: (twangboy) Change to GitPython version 2.1.1 @ *2017-08-11 13:23:52 UTC*
  - 7e8cfff21c Merge pull request #42866 from twangboy/osx_downgrade_gitpython
  - 28053a84a6 Change GitPython version to 2.1.1
- **PR** #42855: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2017-08-10 21:40:39 UTC*
  - 3ce18637be Merge pull request #42855 from rallytime/merge-2017.7

  - 08bbcf5790 Merge branch '2016.11' into '2017.7'
  - 2dde1f77e9 Merge pull request #42851 from terminalmage/bp-42651
    - a3da86eea8 fix syntax
    - 6ecdbcec1d make sure names are correct
    - f83b553d6e add py3 for versionlock
    - 21934f61bb python2- prefix for fedora 26 packages
  - c746f79a3a Merge pull request #42806 from rallytime/fix-42683
    - 8c8640d6b8 Update doc references in glusterfs.volume_present
  - 27a8a2695a Merge pull request #42829 from twangboy/win_pkg_fix_install
    - 83b9b230cd Add winrepo to docs about supporting versions in pkgs
    - 81fefa6e67 Add ability to pass version in pkgs list
  - 3c3ac6aeb2 Merge pull request #42838 from twangboy/win_doc_pki
    - f0a1d06b46 Standardize PKI Client
    - 7de687aa57 Document requirements for win_pki
  - b3e2ae3c58 Merge pull request #42805 from rallytime/bp-42552
    - 5a91c1f2d1 update consul module following this documentation https://www.consul.io/api/acl.html
  - d2ee7934ed Merge pull request #42804 from rallytime/bp-42784
    - dbd29e4aaa only read file if it is not a string
  - 4cbf8057b3 Merge pull request #42826 from terminalmage/fix-spelling
    - 00f93142e4 Fix misspelling of "versions"
  - de997edd90 Merge pull request #42786 from Ch3LL/fix_typo
    - 90a2fb66a2 Fix typo for template_dict in http docs
  - bf6153ebe5 Merge pull request #42795 from lomeroe/bp-42744_201611
    - 695f8c1ae4 fix #42600 in develop
  - 61fad97286 Merge pull request #42748 from whiteinge/save-before-output
    - de60b77c82 Workaround Orchestrate problem that highstate outputter mutates data
  - a4e3e7e786 Merge pull request #42764 from amendlik/cloud-win-loop
    - f3dcfca4e0 Fix infinite loops on failed Windows deployments
  - da85326ad4 Merge pull request #42694 from gtmanfred/2016.11
    - 1a0457af51 allow adding extra remotes to a repository

- **ISSUE** #42774: (rossengeorgiev) pkg.installed succeeds, but fails when you specify package version (refs: #42808)
- **PR** #42808: (terminalmage) Fix regression in yum/dnf version specification @ *2017-08-10 15:59:22 UTC*
  - f954f4f33a Merge pull request #42808 from terminalmage/issue42774
  - c69f17dd18 Add integration test for #42774
  - 78d826dd14 Fix regression in yum/dnf version specification

- **ISSUE** #42639: (amnonbc) k8s module needs a way to manage configmaps (refs: #42807)
- **PR** #42807: (rallytime) Update modules –> states in kubernetes doc module @ *2017-08-10 14:10:40 UTC*
  - d9b0f44885 Merge pull request #42807 from rallytime/fix-42639
  - 152eb88d9f Update modules –> states in kubernetes doc module
- **ISSUE** #42818: (Mapel88) Bug in win_iis module - "create_cert_binding" (refs: #42841)
- **PR** #42841: (Mapel88) Fix bug #42818 in win_iis module @ *2017-08-10 13:44:21 UTC*
  - b8c7bda68d Merge pull request #42841 from Mapel88/patch-1
  - 497241fbcb Fix bug #42818 in win_iis module
- **ISSUE** #42697: (Ch3LL) [Python3] NameError when running salt-run manage.versions (refs: #42782)
- **PR** #42782: (rallytime) Add a cmp compatibility function utility @ *2017-08-09 22:37:29 UTC*
  - 135f9522d0 Merge pull request #42782 from rallytime/fix-42697
  - d707f94863 Update all other calls to "cmp" function
  - 5605104285 Add a cmp compatibility function utility
- **PR** #42784: (gtmanfred) only read file if ret is not a string in http.query (refs: #42804) @ *2017-08-08 17:20:13 UTC*
  - ac752223ad Merge pull request #42784 from gtmanfred/http
  - d397c90e92 only read file if it is not a string
- **ISSUE** #42600: (twangboy) Unable to set 'Not Configured' using win_lgpo execution module (refs: #42744, #42794, #42795)
- **PR** #42794: (lomeroe) Backport #42744 to 2017.7 @ *2017-08-08 17:16:31 UTC*
  - **PR** #42744: (lomeroe) fix #42600 in develop (refs: #42794, #42795)
  - 44995b1abf Merge pull request #42794 from lomeroe/bp-42744
  - 0acffc6df5 fix #42600 in develop
- **ISSUE** #42707: (cro) Service module and state fails on FreeBSD (refs: #42708)
- **PR** #42708: (cro) Do not change the arguments of the function when memoizing @ *2017-08-08 13:47:01 UTC*
  - dcf474c47c Merge pull request #42708 from cro/dont_change_args_during_memoize
  - a260e913b5 Do not change the arguments of the function when memoizing
- **PR** #42783: (rallytime) Sort lists before comparing them in python 3 unit test @ *2017-08-08 13:25:15 UTC*
  - **PR** #42206: (rallytime) [PY3] Fix test that is flaky in Python 3 (refs: #42783)
  - ddb671b8fe Merge pull request #42783 from rallytime/fix-flaky-py3-test
  - 998834fbac Sort lists before compairing them in python 3 unit test
- **PR** #42721: (hibbert) Allow no ip sg @ *2017-08-07 22:07:18 UTC*
  - d69822fe93 Merge pull request #42721 from hibbert/allow_no_ip_sg
  - f58256802a allow_no_ip_sg: Allow user to not supply ipaddress or securitygroups when running boto_efs.create_mount_target
- **ISSUE** #42538: (marnovdm) docker_container.running issue since 2017.7.0: passing domainname gives Error 500: json: cannot unmarshal array into Go value of type string (refs: #42769)

- **PR** #42769: (terminalmage) Fix domainname parameter input translation @ *2017-08-07 20:46:07 UTC*
    - bf7938fbe0 Merge pull request #42769 from terminalmage/issue42538
    - 665de2d1f9 Fix domainname parameter input translation
- **PR** #42388: (The-Loeki) pillar.items pillar_env & pillar_override are never used @ *2017-08-07 17:51:48 UTC*
    - 7bf2cdb363 Merge pull request #42388 from The-Loeki/patch-1
    - 664f4b577b pillar.items pillar_env & pillar_override are never used
- **PR** #42770: (rallytime) [2017.7] Merge forward from 2017.7.1 to 2017.7 @ *2017-08-07 16:21:45 UTC*
    - 9a8c9ebffc Merge pull request #42770 from rallytime/merge-2017.7.1-into-2017.7
    - 6d17c9d227 Merge branch '2017.7.1' into '2017.7'
- **PR** #42768: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2017-08-07 16:21:17 UTC*
    - c765e528d0 Merge pull request #42768 from rallytime/merge-2017.7
    - 0f75482c37 Merge branch '2016.11' into '2017.7'
        - 7b2119feee Merge pull request #42669 from garethgreenaway/42642_2016_11_augeas_module_fix
            - 24413084e2 Updating the call to shlex_split to pass the posix=False argument so that quotes are preserved.
        - 30725769ed Merge pull request #42629 from xiaoanyunfei/tornadoapi
            - 1e13383b95 tornado api
        - f0f00fcee1 Merge pull request #42655 from whiteinge/rest_cherrypy-reenable-stats
            - deb6316d67 Fix lint errors
            - 6bd91c8b03 Reenable cpstats for rest_cherrypy
        - 21cf15f9c3 Merge pull request #42693 from gilbsgilbs/fix-rabbitmq-tags
            - 78fccdc7e2 Cast to list in case tags is a tuple.
            - 287b57b5c5 Fix RabbitMQ tags not properly set.
        - f2b0c9b4fa Merge pull request #42574 from sbojarski/boto-cfn-error-reporting
            - 5c945f10c2 Fix debug message in "boto_cfn._validate" function.
            - 181a1beecc Fixed error reporting in "boto_cfn.present" function.
        - bc1effc4f2 Merge pull request #42623 from terminalmage/fix-unicode-constructor
            - fcf45889dd Fix unicode constructor in custom YAML loader
- **PR** #42651: (gtmanfred) python2- prefix for fedora 26 packages (refs: #42851) @ *2017-08-07 14:35:04 UTC*
    - 3f5827f61e Merge pull request #42651 from gtmanfred/2017.7
    - 8784899942 fix syntax
    - 178cc1bd81 make sure names are correct
    - f179b97b52 add py3 for versionlock
    - 1958d18634 python2- prefix for fedora 26 packages
- **ISSUE** #42688: (hibbert) salt.modules.boto_efs module Invalid type for parameter Tags - type: <type 'dict'>, valid types: <type 'list'>, <type 'tuple'> (refs: #42689)
- **PR** #42689: (hibbert) boto_efs_fix_tags: Fix #42688 invalid type for parameter tags @ *2017-08-06 17:47:07 UTC*

  - 791248e398 Merge pull request #42689 from hibbert/boto_efs_fix_tags
  - 157fb28851 boto_efs_fix_tags: Fix #42688 invalid type for parameter tags
- **ISSUE** #42705: (hbruch) salt.states.docker_container.running replaces container on subsequent runs if oom_kill_disable unsupported (refs: #42745)
- **PR** #42745: (terminalmage) docker.compare_container: treat null oom_kill_disable as False @ *2017-08-05 15:28:20 UTC*
  - 1b3407649b Merge pull request #42745 from terminalmage/issue42705
  - 710bdf6115 docker.compare_container: treat null oom_kill_disable as False
- **ISSUE** #42649: (tehsu) local_batch no longer working in 2017.7.0, 500 error (refs: #42704)
- **PR** #42704: (whiteinge) Add import to work around likely multiprocessing scoping bug @ *2017-08-04 23:03:13 UTC*
  - 5d5b22021b Merge pull request #42704 from whiteinge/expr_form-warn-scope-bug
  - 03b675a618 Add import to work around likely multiprocessing scoping bug
- **ISSUE** #42741: (kkoppel) docker_container.running keeps re-creating containers with links to other containers (refs: #42743)
- **PR** #42743: (kkoppel) Fix docker.compare_container for containers with links @ *2017-08-04 16:00:33 UTC*
  - 888e954e73 Merge pull request #42743 from kkoppel/fix-issue-42741
  - de6d3cc0cf Update dockermod.py
  - 58b997c67f Added a helper function that removes container names from container HostConfig:Links values to enable compare_container() to make the correct decision about differences in links.
- **ISSUE** #42668: (UtahDave) Minions under syndics don't respond to MoM (refs: #42710)
- **ISSUE** #42545: (paul-mulvihill) Salt-api failing to return results for minions connected via syndics. (refs: #42710)
- **PR** #42710: (gtmanfred) use subtraction instead of or @ *2017-08-04 15:14:14 UTC*
  - 03a7f9bbee Merge pull request #42710 from gtmanfred/syndic
  - 683561a711 use subtraction instead of or
- **PR** #42670: (gtmanfred) render kubernetes docs @ *2017-08-03 20:30:56 UTC*
  - 005182b6a1 Merge pull request #42670 from gtmanfred/kube
  - bca17902f5 add version added info
  - 4bbfc751ae render kubernetes docs
- **PR** #42712: (twangboy) Remove master config file from minion-only installer @ *2017-08-03 20:25:02 UTC*
  - df354ddabf Merge pull request #42712 from twangboy/win_build_pkg
  - 8604312a7b Remove master conf in minion install
- **PR** #42714: (cachedout) Set fact gathering style to 'old' for test_junos @ *2017-08-03 13:39:40 UTC*
  - bb1dfd4a42 Merge pull request #42714 from cachedout/workaround_jnpr_test_bug
  - 834d6c605e Set fact gathering style to 'old' for test_junos
- **PR** #42481: (twangboy) Fix *unit.test_crypt* for Windows @ *2017-08-01 18:10:50 UTC*
  - 4c1d931654 Merge pull request #42481 from twangboy/win_unit_test_crypt

  - 102509029e Remove chown mock, fix path seps
- **PR** #42654: (morganwillcock) Disable ZFS in the core grain for NetBSD @ *2017-08-01 17:52:36 UTC*
  - 8bcefb5e67 Merge pull request #42654 from morganwillcock/zfsgrain
  - 49023deb94 Disable ZFS grain on NetBSD
- **ISSUE** #42421: (bfilipek) archive.extracted on Windows failed when dir not exist (refs: #42453)
- **PR** #42453: (gtmanfred) don't pass user to makedirs on windows @ *2017-07-31 19:57:57 UTC*
  - 5baf2650fc Merge pull request #42453 from gtmanfred/makedirs
  - 559d432930 fix tests
  - afa7a13ce3 use logic from file.directory for makedirs
- **PR** #42603: (twangboy) Add runas_passwd as a global for states @ *2017-07-31 19:49:49 UTC*
  - fb81e78f71 Merge pull request #42603 from twangboy/win_fix_runas
  - 0c9e40012b Remove deprecation, add logic to state.py
  - 464ec34713 Fix another instance of runas_passwd
  - 18d6ce4d55 Add global vars to cmd.call
  - 6c71ab6f80 Remove runas and runas_password after state run
  - 4ea264e3db Change to runas_password in docs
  - 61aba35718 Deprecate password, make runas_password a named arg
  - 41f0f75a06 Add new var to list, change to runas_password
  - b9c91eba60 Add runas_passwd as a global for states
- **PR** #42541: (Mareo) Avoid confusing warning when using file.line @ *2017-07-31 19:41:58 UTC*
  - 75ba23c253 Merge pull request #42541 from epita/fix-file-line-warning
  - 2fd172e07b Avoid confusing warning when using file.line
- **PR** #42625: (twangboy) Fix the list function in the win_wua execution module @ *2017-07-31 19:27:16 UTC*
  - 3d328eba80 Merge pull request #42625 from twangboy/fix_win_wua
  - 1340c15ce7 Add general usage instructions
  - 19f34bda55 Fix docs, formatting
  - b17495c9c8 Fix problem with list when install=True
- **ISSUE** #42514: (rickh563) *module.run* does not work as expected in 2017.7.0 (refs: #42602)
- **PR** #42602: (garethgreenaway) Use superseded and deprecated configuration from pillar @ *2017-07-31 18:53:06 UTC*
  - 25094ad9b1 Merge pull request #42602 from garethgreenaway/42514_2017_7_superseded_deprecated_from_pillar
  - 2e132daa73 Slight update to formatting
  - 74bae13939 Small update to something I missed in the first commit. Updating tests to also test for pillar values.
  - 928a4808dd Updating the superseded and deprecated decorators to work when specified as pillar values.
- **PR** #42621: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2017-07-28 19:45:51 UTC*
  - b7cd30d3ee Merge pull request #42621 from rallytime/merge-2017.7

  - 58dcb58a47 Merge branch '2016.11' into '2017.7'
    - cbf752cd73 Merge pull request #42515 from gtmanfred/backslash
      - cc4e45656d Allow not interpreting backslashes in the repl
    - 549495831f Merge pull request #42586 from gdubroeucq/2016.11
      - 9c0b5cc1d6 Remove extra newline
      - d2ef4483e4 yumpkg.py: clean
      - a96f7c09e0 yumpkg.py: add option to the command "check-update"
    - 6b45debf28 Merge pull request #41988 from redmatter/fix-dockerng-network-matching
      - 9eea796da8 Add regression tests for #41982
      - 3369f0072f Fix broken unit test test_network_absent
      - 0ef6cf634c Add trace logging of dockerng.networks result
      - 515c612808 Fix dockerng.network_* name matching
- **ISSUE** #34245: (Talkless) ini.options_present always report state change (refs: #41690)
- **PR** #42618: (rallytime) Back-port #41690 to 2017.7 @ *2017-07-28 19:27:11 UTC*
  - **PR** #41690: (m03) Fix issue #34245 with ini.options_present reporting changes (refs: #42618)
  - d48749b476 Merge pull request #42618 from rallytime/bp-41690
  - 22c6a7c7ff Improve output precision
  - ee4ea6b860 Fix #34245 ini.options_present reporting changes
- **ISSUE** #42588: (ixs) salt-ssh fails when using scan roster and detected minions are uncached (refs: #42589)
- **PR** #42619: (rallytime) Back-port #42589 to 2017.7 @ *2017-07-28 19:26:36 UTC*
  - **PR** #42589: (ixs) Fix ssh-salt calls with scan roster for uncached clients (refs: #42619)
  - e671242a4f Merge pull request #42619 from rallytime/bp-42589
  - cd5eb93903 Fix ssh-salt calls with scan roster for uncached clients
- **ISSUE** #41982: (abulford) dockerng.network_* matches too easily (refs: #42006, #41988)
- **PR** #42006: (abulford) Fix dockerng.network_* name matching @ *2017-07-28 15:52:52 UTC*
  - **PR** #41988: (abulford) Fix dockerng.network_* name matching (refs: #42006)
  - 7d385f8bdc Merge pull request #42006 from redmatter/fix-dockerng-network-matching-2017.7
  - f83960c02a Lint: Remove extra line at end of file.
  - c7d364ec56 Add regression tests for #41982
  - d31f2913bd Fix broken unit test test_network_absent
  - d42f781c64 Add trace logging of docker.networks result
  - 8c00c63b55 Fix dockerng.network_* name matching
- **ISSUE** #12587: (Katafalkas) salt-cloud custom functions/actions (refs: #42616)
- **PR** #42616: (amendlik) Sync cloud modules @ *2017-07-28 15:40:36 UTC*
  - ee8aee1496 Merge pull request #42616 from amendlik/sync-clouds
  - ab21bd9b5b Sync cloud modules when saltutil.sync_all is run

- **PR** #42601: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2017-07-27 22:32:07 UTC*
  - e2dd443002 Merge pull request #42601 from rallytime/merge-2017.7
  - 36a1bcf8c5 Merge branch '2016.11' into '2017.7'
    - 4b16109122 Merge pull request #42339 from isbm/isbm-jobs-scheduled-in-a-future-bsc1036125
      - bbba84ce2d Bugfix: Jobs scheduled to run at a future time stay pending for Salt minions (bsc#1036125)
    - 6c5a7c604a Merge pull request #42077 from vutny/fix-jobs-scheduled-with-whens
      - b1960cea44 Fix scheduled job run on Master if *when* parameter is a list
    - f9cb536589 Merge pull request #42414 from vutny/unify-hash-params-format
      - d1f2a93368 DOCS: unify hash sum with hash type format
    - 535c922511 Merge pull request #42523 from rallytime/fix-42375
      - 685c2cced6 Add information about returning a tuple with an error message
      - fa466519c4 Add a mention of the True/False returns with __virtual__()
    - 0df0e7e749 Merge pull request #42527 from twangboy/win_wua
      - 0373791f2a Correct capatlization
      - af3bcc927b Document changes to Windows Update in 10/2016
    - 69b06586da Merge pull request #42551 from binocvlar/fix-lack-of-align-check-output
      - c4fabaa192 Remove '-s' (–script) argument to parted within align_check function
    - 9e0b4e9faf Merge pull request #42573 from rallytime/bp-42433
      - 0293429e24 Only force saltenv/pillarenv to be a string when not None
    - e931ed2517 Merge pull request #42571 from twangboy/win_add_pythonpath
      - d55a44dd1a Avoid loading user site packages
      - 9af1eb2741 Ignore any PYTHON* environment vars already on the system
      - 4e2fb03a95 Add pythonpath to batch files and service
    - de2f397041 Merge pull request #42387 from DSRCorporation/bugs/42371_KeyError_WeakValueDict
      - e721c7eee2 Don't use *key in weakvaluedict* because it could lie.
    - 641a9d7efd Merge pull request #41968 from root360-AndreasUlm/fix-rabbitmqctl-output-handler
      - 76fd941d91 added tests for rabbitmq 3.6.10 output handler
      - 3602af1e1b Fix rabbitmqctl output handler for 3.6.10
    - 66fede378a Merge pull request #42479 from gtmanfred/interface
      - c32c1b2803 fix pylint
      - 99ec634c6b validate ssh_interface for ec2
    - a925c7029a Merge pull request #42516 from rallytime/fix-42405
      - e3a6717efa Add info about top file to pillar walk-through example to include edit.vim
- **ISSUE** #42148: (sjorge) [2017.7.0rc1] use_superseded and module.run changes from release notes do nothing? (refs: #42270)

- **PR** #42290: (isbm) Backport of #42270 @ *2017-07-27 22:30:05 UTC*
  - **PR** #42270: (The-Loeki) State module.run/wait misses args when looking for kwargs (refs: #42290)
  - 22eea389fa Merge pull request #42290 from isbm/isbm-module_run_parambug_42270_217
  - e38d432f90 Fix docs
  - 1e8a56eda5 Describe function tagging
  - 1d7233224b Describe function batching
  - 1391a05d5e Bugfix: syntax error in the example
  - 8c71257a4b Call unnamed parameters properly
  - 94c97a8f25 Update and correct the error message
  - ea8351362c Bugfix: args gets ignored alongside named parameters
  - 74689e3462 Add ability to use tagged functions in the same set
- **PR** #42251: (twangboy) Fix *unit.modules.test_win_ip* for Windows @ *2017-07-27 19:22:03 UTC*
  - 4c20f1cfbb Merge pull request #42251 from twangboy/unit_win_test_win_ip
  - 97261bfe69 Fix win_inet_pton check for malformatted ip addresses
- **PR** #42255: (twangboy) Fix *unit.modules.test_win_system* for Windows @ *2017-07-27 19:12:42 UTC*
  - 2985e4c0e6 Merge pull request #42255 from twangboy/win_unit_test_win_system
  - acc0345bc8 Fix unit tests
- **PR** #42528: (twangboy) Namespace *cmp_to_key* in the pkg state for Windows @ *2017-07-27 18:30:23 UTC*
  - a573386260 Merge pull request #42528 from twangboy/win_fix_pkg_state
  - a040443fa1 Move functools import inside pylint escapes
  - 118d5134e2 Remove namespaced function *cmp_to_key*
  - a02c91adda Namespace *cmp_to_key* in the pkg state for Windows
- **ISSUE** #42521: (rickh563) chocolatey.installed broken on 2017.7.0 (refs: #42534)
- **PR** #42534: (jmarinaro) Fixes AttributeError thrown by chocolatey state @ *2017-07-27 17:59:50 UTC*
  - 62ae12bcd9 Merge pull request #42534 from jmarinaro/2017.7
  - b242d2d6b5 Fixes AttributeError thrown by chocolatey state Fixes #42521
- **ISSUE** #40354: (exc414) CentOS 6.8 Init Script - Sed unterminated address regex (refs: #42557)
- **ISSUE** #37312: (gtmanfred) CLI flags should take overload settings in the config files (refs: #42557)
- **PR** #42557: (justincbeard) Fixing output so –force-color and –no-color override master and min… @ *2017-07-27 17:07:33 UTC*
  - 52605c249d Merge pull request #42557 from justincbeard/bugfix_37312
  - ee3bc6eb10 Fixing output so –force-color and –no-color override master and minion config color value
- **PR** #42567: (skizunov) Fix disable_<tag-name> config option @ *2017-07-27 17:05:00 UTC*
  - ab33517efb Merge pull request #42567 from skizunov/develop3
  - 0f0b7e3e0a Fix disable_<tag-name> config option
- **PR** #42577: (twangboy) Compile scripts with -E -s params for Salt on Mac @ *2017-07-26 22:44:37 UTC*

  - 30bb941179 Merge pull request #42577 from twangboy/mac_scripts
  - 69d5973651 Compile scripts with -E -s params for python
- **PR** #42524: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2017-07-26 22:41:06 UTC*
  - 60cd078164 Merge pull request #42524 from rallytime/merge-2017.7
  - 14d8d795f6 Merge branch '2016.11' into '2017.7'
    - 1bd5bbccc2 Merge pull request #42509 from clem-compilatio/fix-42417
      - 72924b06b8 Fix _assign_floating_ips in openstack.py
    - 4bf35a74de Merge pull request #42464 from garethgreenaway/2016_11_remove_tmp_identity_file
      - ff24102d51 Uncomment the line that removes the temporary identity file.
    - e2120dbd0e Merge pull request #42443 from garethgreenaway/42357_pass_args_kwargs_correctly
      - 635810b3e3 Updating the slack engine in 2016.11 to pass the args and kwrags correctly to LocalClient
    - 8262cc9054 Merge pull request #42200 from shengis/sqlite3_fix_row_absent_2016.11
      - 407b8f4bb3 Fix #42198 If where_args is not set, not using it in the delete request.
    - d9df97e5a3 Merge pull request #42424 from goten4/2016.11
      - 1c0574d05e Fix error message when tornado or pycurl is not installed
- **PR** #42575: (rallytime) [2017.7] Merge forward from 2017.7.1 to 2017.7 @ *2017-07-26 22:39:10 UTC*
  - 2acde837df Merge pull request #42575 from rallytime/merge-2017.7.1-into-2017.7
  - 63bb0fb2c4 pass in empty kwarg for reactor
  - 2868061ee4 update chunk, not kwarg in chunk
  - 46715e9d94 Merge branch '2017.7.1' into '2017.7'
- **PR** #42555: (Ch3LL) add changelog to 2017.7.1 release notes @ *2017-07-26 14:57:43 UTC*
  - 1d93e92194 Merge pull request #42555 from Ch3LL/7.1_add_changelog
  - fb69e71093 add changelog to 2017.7.1 release notes
- **PR** #42266: (twangboy) Fix *unit.states.test_file* for Windows @ *2017-07-25 20:26:32 UTC*
  - 07c2793e86 Merge pull request #42266 from twangboy/win_unit_states_test_file
  - 669aaee10d Mock file exists properly
  - a4231c9827 Fix ret mock for linux
  - 0c484f8979 Fix unit tests on Windows
- **PR** #42484: (shengis) Fix a potential Exception with an explicit error message @ *2017-07-25 18:34:12 UTC*
  - df417eae17 Merge pull request #42484 from shengis/fix-explicit-error-msg-x509-sign-remote
  - 0b548c72e1 Fix a potential Exception with an explicit error message
- **ISSUE** saltstack/salt-jenkins#396: (Ch3LL) Python3 Fix Test: JoyentTest.test_instance (refs: #42529)
- **ISSUE** #41720: (rallytime) [Py3] Some salt-cloud drivers do not work using Python 3 (refs: #42529)
- **PR** #42529: (gtmanfred) Fix joyent for python3 @ *2017-07-25 16:37:48 UTC*
  - 0f25ec76f9 Merge pull request #42529 from gtmanfred/2017.7

  - b7ebb4d81a these drivers do not actually have an issue.
  - e90ca7a114 use salt encoding for joyent on 2017.7
- **PR** #42465: (garethgreenaway) [2017.7] Small fix to modules/git.py @ *2017-07-24 17:24:55 UTC*
  - 488457c5a0 Merge pull request #42465 from garethgreenaway/2017_7_remove_tmp_identity_file
  - 1920dc6079 Uncomment the line that removes the temporary identity file.
- **ISSUE** #23516: (dkiser) BUG: cron job scheduler sporadically works (refs: #42077)
- **PR** #42107: (vutny) [2017.7] Fix scheduled jobs if *when* parameter is a list @ *2017-07-24 17:04:12 UTC*
  - **PR** #42077: (vutny) Fix scheduled job run on Master if *when* parameter is a list (refs: #42107)
  - **PR** #41973: (vutny) Fix Master/Minion scheduled jobs based on Cron expressions (refs: #42077)
  - 4f044999fa Merge pull request #42107 from vutny/2017.7-fix-jobs-scheduled-with-whens
  - 905be493d4 [2017.7] Fix scheduled jobs if *when* parameter is a list
- **PR** #42506: (terminalmage) Add PER_REMOTE_ONLY to init_remotes call in git_pillar runner @ *2017-07-24 16:59:21 UTC*
  - 6eaa0763e1 Merge pull request #42506 from terminalmage/fix-git-pillar-runner
  - 6352f447ce Add PER_REMOTE_ONLY to init_remotes call in git_pillar runner
- **PR** #42502: (shengis) Fix azurerm query to show IPs @ *2017-07-24 15:54:45 UTC*
  - b88e645f10 Merge pull request #42502 from shengis/fix_azurerm_request_ips
  - 92f1890701 Fix azurerm query to show IPs
- **PR** #42180: (twangboy) Fix *unit.modules.test_timezone* for Windows @ *2017-07-24 14:46:16 UTC*
  - c793d83d26 Merge pull request #42180 from twangboy/win_unit_test_timezone
  - 832a3d86dd Skip tests that use os.symlink on Windows
- **PR** #42474: (whiteinge) Cmd arg kwarg parsing test @ *2017-07-24 14:13:30 UTC*
  - **PR** #39646: (terminalmage) Handle deprecation of passing string args to load_args_and_kwargs (refs: #42474)
  - 083ff00410 Merge pull request #42474 from whiteinge/cmd-arg-kwarg-parsing-test
  - 0cc0c0967a Lint fixes
  - 66093738c8 Add back support for string kwargs
  - 622ff5be40 Add LocalClient.cmd test for arg/kwarg parsing
  - 9f4eb80d90 Add a test.arg variant that cleans the pub kwargs by default
- **PR** #42425: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2017-07-21 22:43:41 UTC*
  - c91a5e539e Merge pull request #42425 from rallytime/merge-2017.7
  - ea457aa0a5 Remove ALIASES block from template util
  - c673b64583 Merge branch '2016.11' into '2017.7'
    - 42bb1a64ca Merge pull request #42350 from twangboy/win_fix_ver_grains_2016.11
      - 8c048403d7 Detect Server OS with a desktop release name
    - 0a72e56f6b Merge pull request #42356 from meaksh/2016.11-AliasesLoader-wrapper-fix
      - 915d94219e Allow to check whether a function is available on the AliasesLoader wrapper

        - 10eb7b7a79 Merge pull request #42368 from twangboy/win_fix_build_2016.11
            - a7c910c31e Remove build and dist directories before install
        - 016189f62f Merge pull request #42370 from rallytime/merge-2016.11
            - 0aa5dde1de Merge branch '2016.3' into '2016.11'
            - e9b0f20f8a Merge pull request #42359 from Ch3LL/doc-update-2016.3
            - dc85b5edbe [2016.3] Update version numbers in doc config for 2017.7.0 release
        - f06a6f1796 Merge pull request #42360 from Ch3LL/doc-update-2016.11
            - b90b7a7506 [2016.11] Update version numbers in doc config for 2017.7.0 release
        - e0595b0a0f Merge pull request #42319 from rallytime/config-docs
            - b40f980632 Add more documentation for config options that are missing from master/minion docs
        - 78940400e3 Merge pull request #42352 from CorvinM/issue42333
            - 526b6ee14d Multiple documentation fixes

- **ISSUE** #42357: (Giandom) Salt pillarenv problem with slack engine (refs: #42443, #42444)
- **PR** #42444: (garethgreenaway) [2017.7] Fix to slack engine @ *2017-07-21 22:03:48 UTC*
    - 10e4d9234b Merge pull request #42444 from garethgreenaway/42357_2017_7_pass_args_kwargs_correctly
    - f411cfc2a9 Updating the slack engine in 2017.7 to pass the args and kwrags correctly to LocalClient
- **PR** #42461: (rallytime) Bump warning version from Oxygen to Fluorine in roster cache @ *2017-07-21 21:33:25 UTC*
    - 723be49fac Merge pull request #42461 from rallytime/bump-roster-cache-deprecations
    - c0df0137f5 Bump warning version from Oxygen to Fluorine in roster cache
- **ISSUE** #42374: (tyhunt99) [2017.7.0] salt-run mange.versions throws exception if minion is offline or unresponsive (refs: #42436)
- **PR** #42436: (garethgreenaway) Fixes to versions function in manage runner @ *2017-07-21 19:41:07 UTC*
    - 09521602c1 Merge pull request #42436 from garethgreenaway/42374_manage_runner_minion_offline
    - 0fd39498c0 Updating the versions function inside the manage runner to account for when a minion is offline and we are unable to determine it's version.
- **ISSUE** #42427: (grichmond-salt) Issue Passing Variables created from load_json as Inline Pillar Between States (refs: #42435)
- **PR** #42435: (terminalmage) Modify our custom YAML loader to treat unicode literals as unicode strings (refs: #42812) @ *2017-07-21 19:40:34 UTC*
    - 54193ea543 Merge pull request #42435 from terminalmage/issue42427
    - 31273c7ec1 Modify our custom YAML loader to treat unicode literals as unicode strings
- **ISSUE** #42381: (zebooka) Git.detached broken in 2017.7.0 (refs: #42399)
- **ISSUE** #38878: (tomlaredo) [Naming consistency] git.latest "rev" option VS git.detached "ref" option (refs: #38898)
- **PR** #42399: (rallytime) Update old "ref" references to "rev" in git.detached state @ *2017-07-21 19:38:59 UTC*
    - **PR** #38898: (terminalmage) git.detached: rename ref to rev for consistency (refs: #42399)

  - 0b3179135c Merge pull request #42399 from rallytime/fix-42381
  - d9d94fe02f Update old "ref" references to "rev" in git.detached state
- **ISSUE** #42400: (Enquier) Conflict in execution of passing pillar data to orch/reactor event executions 2017.7.0 (refs: #42031)
- **PR** #42031: (skizunov) Fix: Reactor emits critical error @ *2017-07-21 19:38:34 UTC*
  - bd4adb483d Merge pull request #42031 from skizunov/develop3
  - 540977b4b1 Fix: Reactor emits critical error
- **ISSUE** #41949: (jrporcaro) Event returner doesn't work with Windows Master (refs: #42027)
- **PR** #42027: (gtmanfred) import salt.minion for EventReturn for Windows @ *2017-07-21 19:37:03 UTC*
  - 3abf7ad7d7 Merge pull request #42027 from gtmanfred/2017.7
  - fd4458b6c7 import salt.minion for EventReturn for Windows
- **PR** #42454: (terminalmage) Document future renaming of new rand_str jinja filter @ *2017-07-21 18:47:51 UTC*
  - 994d3dc74a Merge pull request #42454 from terminalmage/jinja-docs-2017.7
  - 98b661406e Document future renaming of new rand_str jinja filter
- **PR** #42452: (Ch3LL) update windows urls to new py2/py3 naming scheme @ *2017-07-21 17:20:47 UTC*
  - 4480075129 Merge pull request #42452 from Ch3LL/fix_url_windows
  - 3f4a918f73 update windows urls to new py2/py3 naming scheme
- **ISSUE** #42404: (gabekahen) [2017.7] file.managed with cmd_check "No such file or directory" (refs: #42411)
- **ISSUE** #33708: (pepinje) visudo check command leaves cache file in /tmp (refs: #42411, #38063)
- **PR** #42411: (seedickcode) Fix file.managed check_cmd file not found - Issue #42404 @ *2017-07-20 21:59:17 UTC*
  - **PR** #38063: (llua) tmp file clean up in file.manage - fix for #33708 (refs: #42411)
  - 33e90be1fe Merge pull request #42411 from seedickcode/check_cmd_fix
  - 4ae3911f01 Fix file.managed check_cmd file not found - Issue #42404
- **PR** #42409: (twangboy) Add Scripts to build Py3 on Mac @ *2017-07-20 21:36:34 UTC*
  - edde31376a Merge pull request #42409 from twangboy/mac_py3_scripts
  - ac0e04af72 Remove build and dist, sign pkgs
  - 9d66e273c4 Fix hard coded pip path
  - 7b8d6cbbd2 Add support for Py3
  - aa4eed93c8 Update Python and other reqs
- **ISSUE** #42403: (astronouth7303) [2017.7] Pillar empty when state is applied from orchestrate (refs: #42433)
- **PR** #42433: (terminalmage) Only force saltenv/pillarenv to be a string when not None (refs: #42573) @ *2017-07-20 21:32:24 UTC*
  - 82982f940d Merge pull request #42433 from terminalmage/issue42403
- **PR** #42408: (CorvinM) Fix documentation misformat in salt.states.file.replace @ *2017-07-20 00:45:43 UTC*
  - a71938cefe Merge pull request #42408 from CorvinM/file-replace-doc-fix
  - 246a2b3e74 Fix documentation misformat in salt.states.file.replace

- **PR** #42347: (twangboy) Fixes problem with Version and OS Release related grains on certain versions of Python @ *2017-07-19 17:05:43 UTC*
  - d385dfd19d Merge pull request #42347 from twangboy/win_fix_ver_grains
  - ef1f663fc9 Detect server OS with a desktop release name
- **PR** #42366: (twangboy) Remove build and dist directories before install @ *2017-07-19 16:37:41 UTC*
  - eb9e4206c9 Merge pull request #42366 from twangboy/win_fix_build
  - 0946002713 Add blank line after delete
  - f7c0bb4f46 Remove build and dist directories before install
- **PR** #42373: (Ch3LL) Add initial 2017.7.1 Release Notes File @ *2017-07-19 16:28:46 UTC*
  - af7820f25d Merge pull request #42373 from Ch3LL/add_2017.7.1
  - ce1c1b6d28 Add initial 2017.7.1 Release Notes File
- **PR** #42150: (twangboy) Fix *unit.modules.test_pip* for Windows @ *2017-07-19 16:01:17 UTC*
  - 59e012b485 Merge pull request #42150 from twangboy/win_unit_test_pip
  - 4ee24202fc Fix unit tests for test_pip
- **PR** #42154: (twangboy) Fix *unit.modules.test_reg_win* for Windows @ *2017-07-19 16:00:38 UTC*
  - ade25c6b34 Merge pull request #42154 from twangboy/win_unit_test_reg
  - 00d9a52802 Fix problem with handling REG_QWORD in list values
- **PR** #42182: (twangboy) Fix *unit.modules.test_useradd* for Windows @ *2017-07-19 15:55:33 UTC*
  - 07593675e2 Merge pull request #42182 from twangboy/win_unit_test_useradd
  - 8260a71c07 Disable tests that require pwd in Windows
- **PR** #42364: (twangboy) Windows Package notes for 2017.7.0 @ *2017-07-18 19:24:45 UTC*
  - a175c40c1d Merge pull request #42364 from twangboy/release_notes_2017.7.0
  - 96517d1355 Add note about patched windows packages
- **PR** #42361: (Ch3LL) [2017.7] Update version numbers in doc config for 2017.7.0 release @ *2017-07-18 19:23:22 UTC*
  - 4dfe50e558 Merge pull request #42361 from Ch3LL/doc-update-2017.7
  - dc5bb301f7 [2017.7] Update version numbers in doc config for 2017.7.0 release
- **PR** #42363: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2017-07-18 18:40:48 UTC*
  - 587138d771 Merge pull request #42363 from rallytime/merge-2017.7
  - 7aa31ff030 Merge branch '2016.11' into '2017.7'
    - b256001760 Merge pull request #42353 from terminalmage/fix-git-test
      - 14cf6ce322 is_windows is a function, not a propery/attribute
    - 866a1febb4 Merge pull request #42264 from rallytime/fix-41116
      - bd638880e3 Add mono-spacing to salt-minion reference for consistency
      - 30d62f43da Update minion restart section in FAQ doc for windows
    - 9a707088ad Merge pull request #42275 from terminalmage/issue42194
      - 663874908a pkg.installed: pack name/version into pkgs argument

- e588f235e0 Merge pull request #42269 from rallytime/fix-41721
  - f2250d474a Add a note about using different styles of quotes.
  - 38d9b3d553 Add some clarity to "multiple quotes" section of yaml docs
- 5aaa214a75 Merge pull request #42282 from rallytime/fix-42152
  - f032223843 Handle libcloud objects that throw RepresenterErrors with –out=yaml
- fb5697a4bc Merge pull request #42308 from lubyou/42295-fix-file-absent-windows
  - 026ccf401a Force file removal on Windows. Fixes #42295
- da2a8a518f Merge pull request #42314 from rallytime/fix-42267
  - c406046940 Add clarification to salt ssh docs about key auto-generation.
- acadd54013 Merge pull request #41945 from garethgreenaway/41936_allow_identity_files_with_user
  - 44841e5626 Moving the call to cp.get_file inside the with block to ensure the umask is preserved when we grab the file.
  - f9ba60eed8 Merge pull request #1 from terminalmage/pr-41945
  - 1b6026177c Restrict set_umask to mkstemp call only
  - 68549f3496 Fixing umask to we can set files as executable.
  - 4949bf3ff3 Updating to swap on the new salt.utils.files.set_umask context_manager
  - 8faa9f6d92 Updating PR with requested changes.
  - 494765e939 Updating the git module to allow an identity file to be used when passing the user parameter
- f90e04a2bc Merge pull request #42289 from CorvinM/bp-41543
  - 357dc22f05 Fix user creation with empty password
- a91a3f81b1 Merge pull request #42123 from vutny/fix-master-utils-import
  - 6bb8b8f98c Add missing doc for `utils_dirs` Minion config option
  - f1bc58f6d5 Utils: add example of module import
- e2aa5114e4 Merge pull request #42261 from rallytime/minor-doc-fix
  - 8c76bbb53d Some minor doc fixes for dnsutil module so they'll render correctly
- 3e9dfbc9cc Merge pull request #42262 from rallytime/bp-42224
  - c31ded341c Remove duplicate instruction in Openstack Rackspace config example
- 7780579c36 Merge pull request #42181 from garethgreenaway/42137_backport_fix_from_2017_7
  - a34970b45b Back porting the fix for 2017.7 that ensures the order of the names parameter.
- 72537868a6 Merge pull request #42253 from gtmanfred/2016.11
  - 53e25760be Only use unassociated ips when unable to allocate
- b2a4698b5d Merge pull request #42252 from UtahDave/2016.11local
  - e6a9563d47 simple doc updates
- 781fe13be7 Merge pull request #42235 from astronouth7303/patch-1-2016.3
  - 4cb51bd03a Make note of dig partial requirement.

    - 08e7d8351a Abolish references to *dig* in examples.
- 83cbd76f16 Merge pull request #42215 from twangboy/win_iis_docs
    - c07e22041a Add missing config to example
- 274946ab00 Merge pull request #42211 from terminalmage/issue40928
    - 22a18fa2ed Only pass a saltenv in orchestration if one was explicitly passed (2016.11)
- 89261cf06c Merge pull request #42173 from rallytime/bp-37424
    - 01addb6053 Avoid Early Convert ret['comment'] to String
- 3b17fb7f83 Merge pull request #42175 from rallytime/bp-39366
    - 53f7b987e8 Pass sig to service.status in after_toggle
- ea16f47f0a Merge pull request #42172 from rallytime/merge-2016.11
    - b1fa332a11 Merge branch '2016.3' into '2016.11'
    - 8fa1fa5bb1 Merge pull request #42155 from phsteve/doc-fix-puppet
    - fb2cb78a31 Fix docs for puppet.plugin_sync so code-block renders properly and sync is spelled consistently
- 6307b9873f Merge pull request #42176 from rallytime/bp-42109
    - 686926daf7 Update aws.rst - add Debian default username
- 28c4e4c3b7 Merge pull request #42095 from terminalmage/docker-login-debugging
    - bd27870a71 Add debug logging to dockerng.login
- 2b754bc5af Merge pull request #42119 from terminalmage/issue42116
    - 9a268949e3 Add integration test for 42116
    - 1bb42bb609 Fix regression when CLI pillar override is used with salt-call
- 8c0a83cbb5 Merge pull request #42121 from terminalmage/issue42114
    - d14291267f Fix pillar.get when saltenv is passed
- 687992c240 Merge pull request #42094 from terminalmage/quiet-exception
    - 47d61f4edf Prevent command from showing in exception when output_loglevel=quiet
- dad255160c Merge pull request #42163 from vutny/fix-42115
    - b27b1e340a Fix #42115: parse libcloud "rc" version correctly
- 2a8ae2b3b6 Merge pull request #42164 from Ch3LL/fix_kerb_doc
    - 7c0fb248ec Fix kerberos create_keytab doc
- 678d4d4098 Merge pull request #42141 from rallytime/bp-42098
    - bd80243233 Change repo_ng to repo-ng
- c8afd7a3c9 Merge pull request #42140 from rallytime/bp-42097
    - 9c4e132540 Import datetime
    - 1435bf177e require large timediff for ipv6 warning
- c239664c8b Merge pull request #42142 from Ch3LL/change_builds
    - e1694af39c Update builds available for rc1

- **PR** #42340: (isbm) Bugfix: Jobs scheduled to run at a future time stay pending for Salt … @ *2017-07-18 18:13:36 UTC*
  - 55b7a5cb4a Merge pull request #42340 from isbm/isbm-jobs-scheduled-in-a-future-2017.7-bsc1036125
  - 774d204d65 Bugfix: Jobs scheduled to run at a future time stay pending for Salt minions (bsc#1036125)
- **PR** #42327: (mirceaulinic) Default skip_verify to False @ *2017-07-18 18:04:36 UTC*
  - e72616c5f1 Merge pull request #42327 from mirceaulinic/patch-10
  - c830573a2c Trailing whitespaces
  - c83e6fc696 Default skip_verify to False
- **ISSUE** #42151: (sjorge) Doc errors in jinja doc for develop branch (refs: #42179)
- **PR** #42179: (rallytime) Fix some documentation issues found in jinja filters doc topic @ *2017-07-18 18:01:57 UTC*
  - ba799b2831 Merge pull request #42179 from rallytime/fix-42151
  - 798d29276e Add note about "to_bytes" jinja filter issues when using yaml_jinja renderer
  - 1bbff572ab Fix some documentation issues found in jinja filters doc topic
- **ISSUE** #42076: (abulford) dockerng.volume_present test looks as though it would cause a change (refs: #42087, #42086)
- **PR** #42087: (abulford) Make result=true if Docker volume already exists @ *2017-07-17 18:41:47 UTC*
  - **PR** #42086: (abulford) Make result=true if Docker volume already exists (refs: #42087)
  - 8dbb93851d Merge pull request #42087 from redmatter/fix-dockerng-volume-present-result-2017.7
  - 2e1dc95500 Make result=true if Docker volume already exists
- **ISSUE** #42166: (sjorge) [2017.7.0rc1] jinja filter network_hosts fails on large IPv6 networks (refs: #42186)
- **PR** #42186: (rallytime) Use long_range function for IPv6Network hosts() function @ *2017-07-17 18:39:35 UTC*
  - c84d6db548 Merge pull request #42186 from rallytime/fix-42166
  - b8bcc0d599 Add note to various network_hosts docs about long_run for IPv6 networks
  - 11862743c2 Use long_range function for IPv6Network hosts() function
- **PR** #42210: (terminalmage) Only pass a saltenv in orchestration if one was explicitly passed (2017.7) @ *2017-07-17 18:22:39 UTC*
  - e7b79e0fd2 Merge pull request #42210 from terminalmage/issue40928-2017.7
  - 771ade5d73 Only pass a saltenv in orchestration if one was explicitly passed (2017.7)
- **PR** #42236: (mirceaulinic) New option for napalm proxy/minion: provider @ *2017-07-17 18:19:56 UTC*
  - 0e49021b0e Merge pull request #42236 from cloudflare/napalm-provider
  - 1ac69bd737 Document the provider option and rearrange the doc
  - 4bf4b14161 New option for napalm proxy/minion: provider
- **PR** #42257: (twangboy) Fix *unit.pillar.test_git* for Windows @ *2017-07-17 17:51:42 UTC*
  - 3ec5bb1c2f Merge pull request #42257 from twangboy/win_unit_pillar_test_git
  - 45be32666a Add error-handling function to shutil.rmtree
- **PR** #42258: (twangboy) Fix *unit.states.test_environ* for Windows @ *2017-07-17 17:50:38 UTC*

  - 36395625c2 Merge pull request #42258 from twangboy/win_unit_states_tests_environ
  - 55b278c478 Mock the reg.read_value function
- **PR** #42265: (rallytime) Gate boto_elb tests if proper version of moto isn't installed @ *2017-07-17 17:47:52 UTC*
  - 894bdd2b19 Merge pull request #42265 from rallytime/gate-moto-version
  - 78cdee51d5 Gate boto_elb tests if proper version of moto isn't installed
- **PR** #42277: (twangboy) Fix *unit.states.test_winrepo* for Windows @ *2017-07-17 17:37:07 UTC*
  - baf04f2a2d Merge pull request #42277 from twangboy/win_unit_states_test_winrepo
  - ed89cd0b93 Use os.sep for path seps
- **PR** #42309: (terminalmage) Change "TBD" in versionadded to "2017.7.0" @ *2017-07-17 17:11:45 UTC*
  - be6b211683 Merge pull request #42309 from terminalmage/fix-versionadded
  - 603f5b7de6 Change "TBD" in versionadded to "2017.7.0"
- **PR** #42206: (rallytime) [PY3] Fix test that is flaky in Python 3 (refs: #42783) @ *2017-07-17 17:09:53 UTC*
  - acd29f9b38 Merge pull request #42206 from rallytime/fix-flaky-test
  - 2be4865f48 [PY3] Fix test that is flaky in Python 3
- **PR** #42126: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2017-07-17 17:07:19 UTC*
  - 8f1cb287cf Merge pull request #42126 from rallytime/merge-2017.7
- **PR** #42078: (damon-atkins) pkg.install and pkg.remove fix version number input. @ *2017-07-05 06:04:57 UTC*
  - 4780d7830a Merge pull request #42078 from damon-atkins/fix_convert_flt_str_version_on_cmd_line
  - 09d37dd892 Fix comment typo
  - 7167549425 Handle version=None when converted to a string it becomes 'None' parm should default to empty string rather than None, it would fix better with existing code.
  - 4fb2bb1856 Fix typo
  - cf55c3361c pkg.install and pkg.remove on the command line take number version numbers, store them within a float. However version is a string, to support versions numbers like 1.3.4
- **PR** #42105: (Ch3LL) Update releasecanddiate doc with new 2017.7.0rc1 Release @ *2017-07-04 03:14:42 UTC*
  - 46d575acbc Merge pull request #42105 from Ch3LL/update_rc
  - d4e7b91608 Update releasecanddiate doc with new 2017.7.0rc1 Release
- **ISSUE** #41885: (astronouth7303) Recommended pip installation outdated? (refs: #42099)
- **PR** #42099: (rallytime) Remove references in docs to pip install salt-cloud @ *2017-07-03 22:13:44 UTC*
  - d38548bbbd Merge pull request #42099 from rallytime/fix-41885
  - c2822e05ad Remove references in docs to pip install salt-cloud
- **ISSUE** #42076: (abulford) dockerng.volume_present test looks as though it would cause a change (refs: #42087, #42086)
- **PR** #42086: (abulford) Make result=true if Docker volume already exists (refs: #42087) @ *2017-07-03 15:48:33 UTC*
  - 81d606a8cb Merge pull request #42086 from redmatter/fix-dockerng-volume-present-result
  - 8d549685a7 Make result=true if Docker volume already exists

- **ISSUE** #25842: (shikhartanwar) Running salt-minion as non-root user to execute sudo commands always returns an error (refs: #42021)
- **PR** #42021: (gtmanfred) Set concurrent to True when running states with sudo @ *2017-06-30 21:02:15 UTC*
  - 7160697123 Merge pull request #42021 from gtmanfred/2016.11
  - 26beb18aa5 Set concurrent to True when running states with sudo
- **PR** #42029: (terminalmage) Mock socket.getaddrinfo in unit.utils.network_test.NetworkTestCase.test_host_to_ips @ *2017-06-30 20:58:56 UTC*
  - b784fbbdf8 Merge pull request #42029 from terminalmage/host_to_ips
  - 26f848e111 Mock socket.getaddrinfo in unit.utils.network_test.NetworkTestCase.test_host_to_ips
- **PR** #42055: (dmurphy18) Upgrade support for gnupg v2.1 and higher @ *2017-06-30 20:54:02 UTC*
  - e067020b9b Merge pull request #42055 from dmurphy18/handle_gnupgv21
  - e20cea6350 Upgrade support for gnupg v2.1 and higher
- **PR** #42048: (Ch3LL) Add initial 2016.11.7 Release Notes @ *2017-06-30 16:00:05 UTC*
  - 74ba2abc48 Merge pull request #42048 from Ch3LL/add_11.7
  - 1de5e008a0 Add initial 2016.11.7 Release Notes

28.2.23 Salt 2017.7.3 Release Notes

Version 2017.7.3 is a bugfix release for *2017.7.0*.

Statistics

- Total Merges: **501**
- Total Issue References: **94**
- Total PR References: **423**
- Contributors: **86** (3add3287, BenoitKnecht, Ch3LL, CorvinM, Da-Juan, DmitryKuzmenko, Giandom, The-Loeki, UtahDave, adelcast, amendlik, angeloudy, anlutro, arthurlogilab, basepi, benediktwerner, brejoc, cachedout, campbellmc, chnrxn, clan, corywright, damon-atkins, dincamihai, dmurphy18, eliasp, eradman, forksaber, frogunder, gaborn57, garethgreenaway, golmaal, gracinet, gtmanfred, haam3r, isbm, jettero, jf, jubrad, keesbos, kris-anderson, lomeroe, mateiw, mattLLVW, mephi42, mirceaulinic, mkurtak, morganwillcock, msummers42, mtorromeo, multani, mvivaldi, mz-bmcqueen, nasenbaer13, nicholasmhughes, oarmstrong, pkruk, pratik705, psagers, rallytime, rbjorklin, rcallphin, renner, rhoths, richardsimko, rklaren, roaldnefs, s0undt3ch, samodid, skizunov, skjaro, steverweber, sumeetisp, t0fik, techhat, terminalmage, timfreund, timka, tkwilliams, twangboy, unthought, vernondcole, vutny, wedge-jarrad, whytewolf, xuhcc)

Windows Changes

`pkg` Execution Module'

Significate changes (PR #43708 & #45390, damon-atkins) have been made to the pkg execution module. Users should test this release against their existing package sls definition files.

- `pkg.list_available` no longer defaults to refreshing the winrepo meta database.

- *`pkg.install`* without a `version` parameter no longer upgrades software if the software is already installed. Use `pkg.install version=latest` (or simply use a *`pkg.latest`* state to get the old behavior.
- *`pkg.list_pkgs`* now returns multiple versions if software installed more than once.
- *`pkg.list_pkgs`* now returns `Not Found` when the version is not found instead of `(value not set)` which matches the contents of the sls definitions.
- *`pkg.remove`* will wait up to 3 seconds (normally about a second) to detect changes in the registry after removing software, improving reporting of version changes.
- *`pkg.remove`* can remove `latest` software, if `latest` is defined in sls definition.
- Documentation was update for the execution module to match the style in new versions, some corrections as well.
- All install/remove commands are prefix with cmd.exe shell and cmdmod is called with a command line string instead of a list. Some sls files in saltstack/salt-winrepo-ng expected the commands to be prefixed with cmd.exe (i.e. the use of &).
- Some execution module functions results, now behave more like their Unix/Linux versions.

`cmd` Execution Module

Due to a difference in how Python's `subprocess.Popen()` spawns processes on Windows, passing the command as a list of arguments can result in problems. This is because Windows' *CreateProcess* requires the command to be passed as a single string. Therefore, `subprocess` will attempt to re-assemble the list of arguments into as string. Some escaped characters and quotes can cause the resulting string to be incorrectly-assembled, resulting in a failure to execute the command.

Salt now deals with these cases by joining the list of arguments correctly and ensuring that the command is passed to `subprocess.Popen()` as a string.

Changelog for v2017.7.2..v2017.7.3

Generated at: 2018-05-26 21:36:50 UTC

- **ISSUE** #45743: (frogunder) Multi-master PKI not working on Py3 (refs: #45755)
- **PR** #45755: (terminalmage) salt.crypt: Ensure message is encoded before signing @ *2018-01-29 19:04:50 UTC*
  - 1439da8d76 Merge pull request #45755 from terminalmage/issue45743
  - 8af1251c59 salt.crypt: Ensure message is encoded before signing
- **PR** #45700: (Ch3LL) Add PRs to 2017.7.3 Release Notes @ *2018-01-25 20:56:45 UTC*
  - fe194d755f Merge pull request #45700 from Ch3LL/7.3_rn
  - 84c8216901 Add PRs to 2017.7.3 Release Notes
- **PR** #45681: (damon-atkins) 2017.7.3 Release notes for Windows @ *2018-01-25 15:13:18 UTC*
  - ce41f6a6ee Merge pull request #45681 from damon-atkins/2017.7.3_win_release_notes
  - 1d21f86228 Update 2017.7.3.rst
- **PR** #45672: (rallytime) Back-port #45667 to 2017.7.3 @ *2018-01-25 14:04:54 UTC*
  - **PR** #45667: (gtmanfred) default to upgrading when refreshing on archlinux (refs: #45672)
  - 2f303439b7 Merge pull request #45672 from rallytime/bp-45667

  - 74bbaeb7ce we should default to upgrading when refreshing on archlinux
- **PR** #45669: (rallytime) Update man pages for 2017.7.3 release @ *2018-01-24 21:04:59 UTC*
  - 23ff1264e0 Merge pull request #45669 from rallytime/man-pages-2017.7.3
  - d31b41adeb Update man pages for 2017.7.3 release
- **PR** #45666: (terminalmage) Fix failing pkg integration tests for releases with no '.' @ *2018-01-24 17:19:10 UTC*
  - 9a17405ba6 Merge pull request #45666 from terminalmage/salt-jenkins-793
  - 4a6ab729dd Fix failing pkg integration tests for releases with no '.'
- **PR** #45664: (rallytime) Back-port #45452 to 2017.7.3 @ *2018-01-24 15:33:13 UTC*
  - **PR** #45452: (adelcast) opkg.py: make owner fuction return value, instead of iterator (refs: #45664)
  - 0717f7a578 Merge pull request #45664 from rallytime/bp-45452
  - 369720677b opkg.py: make owner function return value, instead of iterator
- **PR** #45649: (rallytime) Back-port #45634 to 2017.7.3 @ *2018-01-24 14:59:43 UTC*
  - **PR** #45634: (Ch3LL) Add different service name for Mac 10.13 test (refs: #45649)
  - 7934372b7b Merge pull request #45649 from rallytime/bp-45634
  - 1c78fc23ea Add different service name for Mac 10.13 test
- **PR** #45654: (twangboy) Merge forward #45638 @ *2018-01-24 14:59:14 UTC*
  - **PR** #45638: (twangboy) Win fix shell info (refs: #45654)
  - 770f0c4664 Merge pull request #45654 from twangboy/win_fix_shell_info_2017.7.3
  - 5bb01aeb8c Merge forward #45638
- **PR** #45653: (rallytime) Back-port #45611 to 2017.7.3 @ *2018-01-24 05:20:11 UTC*
  - **PR** #45611: (terminalmage) Fix unnecessary/incorrect usage of six.binary_type (refs: #45653)
  - 6fc293da46 Merge pull request #45653 from rallytime/bp-45611
  - 0a6b06d8ea Fix unnecessary/incorrect usage of six.binary_type
- **PR** #45642: (rallytime) Back-port #45636 to 2017.7.3 @ *2018-01-23 22:00:30 UTC*
  - **PR** #45636: (Ch3LL) Fix mac service and pkg tests for 10.13 (refs: #45642)
  - 0a07e0d259 Merge pull request #45642 from rallytime/bp-45636
  - df0ad54c9a remove unnecessary variable for test
  - acb14fd43d fix pylint
  - a9b12cd1ea Fix mac service and pkg tests for 10.13
- **PR** #45645: (rallytime) Back-port #45606 to 2017.7.3 @ *2018-01-23 21:54:45 UTC*
  - **PR** #45606: (terminalmage) Fix bug affecting salt-ssh when root_dir differs from the default (refs: #45645)
  - f37a5b6d8d Merge pull request #45645 from rallytime/bp-45606
  - d52d96f30a Fix bug affecting salt-ssh when root_dir differs from the default
- **PR** #45641: (rallytime) Back-port #45508 to 2017.7.3 @ *2018-01-23 21:18:39 UTC*
  - **PR** #45508: (frogunder) fix test_archive test for mac on 2017.7 branch (refs: #45641)
  - e659793c09 Merge pull request #45641 from rallytime/bp-45508

  - e6917a291e fix test_archive test for mac on 2017.7 branch
- **PR** #45604: (rallytime) Back-port #45582 to 2017.7.3 @ *2018-01-22 16:54:15 UTC*
  - **PR** #45582: (terminalmage) Two salt-ssh fixes (refs: #45604)
  - ced3269ae8 Merge pull request #45604 from rallytime/bp-45582
  - bc8a450cc7 Remove state.py utils file from thin list
  - 629e6c9674 Further fixes to for salt-ssh test under heavy load
  - 0dff596b59 Add salt/utils/state.py to thin tarball
  - a61afda100 Pass on OSError if thin tarball already removed
- **PR** #45591: (gtmanfred) mark minion_blackout tests as flaky @ *2018-01-22 00:14:31 UTC*
  - 4672baa6c8 Merge pull request #45591 from gtmanfred/2017.7.3
  - f7fd35fc4a test updating the minion blackout timeout to 10 seconds
- **PR** #45585: (rallytime) Back-port #45579 to 2017.7.3 @ *2018-01-22 00:13:59 UTC*
  - **PR** #45579: (terminalmage) Test suite stability fixes (refs: #45585)
  - 2a992f9017 Merge pull request #45585 from rallytime/bp-45579
  - 0292c8345b Lint fix: use six's map
  - 108d8cbeef Use correct utils path for 2017.7
  - a38f4cb6d6 Restrict pyzmq optimizations to pyzmq >= 14.3.0
  - 58ad558346 Fix event unpack
- **PR** #45573: (gtmanfred) update 2017.7.3 tests @ *2018-01-20 20:05:13 UTC*
  - 19cd97ed3b Merge pull request #45573 from gtmanfred/2017.7.3
  - bd3cb47fa7 fix mock for opensuse
  - 808e26e69a test simple website
- **PR** #45570: (gtmanfred) Fix tests for 2017.7.3 @ *2018-01-20 15:01:21 UTC*
  - e72d81ef22 Merge pull request #45570 from gtmanfred/2017.7.3
  - 1f71f301ba specify checking man page path
  - 2ddbcb45c1 fix pkg_resources for usage with testing pip
  - 0ba39a7108 switch systemd-journald for sshd for arch service test
- **PR** #45538: (gtmanfred) Backport test fixes to 2017.7.3 @ *2018-01-19 14:39:44 UTC*
  - 7bc60c56d4 Merge pull request #45538 from gtmanfred/2017.7.3
  - 801e0639b6 Merge branch '2017.7.3' into 2017.7.3
- **PR** #45533: (rallytime) Back-port #45529 to 2017.7.3 @ *2018-01-18 22:52:29 UTC*
  - **PR** #45529: (Ch3LL) Fix UnboundLocalError for pacman pkg installs (refs: #45533)
  - 8ad65e3359 Merge pull request #45533 from rallytime/bp-45529
  - 6d56c64d88 Fix UnboundLocalError for pacman pkg installs
    - 8d907ee1a0 fix moto version
    - 1241ab5fc6 fix test boto imports

    - f4b6367cf9 fix fedora pkg test

- **ISSUE** #45394: (dmurphy18) git.latest fails when "depth" is used with a non-default branch (refs: #45399)
- **PR** #45442: (rallytime) Back-port #45399 to 2017.7.3 @ *2018-01-17 17:20:48 UTC*
  - **PR** #45399: (terminalmage) Fix git.latest failure when rev is not the default branch (refs: #45442)
  - 7379f9e3e5 Merge pull request #45442 from rallytime/bp-45399
  - 590a6db626 Lint: use support TMP path instead of integration TMP path
  - c081b2c62c Fix git.latest failure when rev is not the default branch
- **PR** #45468: (twangboy) Fix some issues with reg.py @ *2018-01-16 22:23:47 UTC*
  - ee5090f69b Merge pull request #45468 from twangboy/win_reg
  - a0d21c6354 Fix some issues with reg.py
- **ISSUE** #44913: (ari) FreeBSD packaging install performance regression (refs: #45174)
- **PR** #45434: (rallytime) Back-port #45174 to 2017.7.3 @ *2018-01-14 12:43:16 UTC*
  - **PR** #45174: (eradman) Do not force pkg reinstall on FreeBSD (refs: #45434)
  - ef7a896eb6 Merge pull request #45434 from rallytime/bp-45174
  - b310ff7ab8 Do not force pkg reinstall on FreeBSD
- **PR** #45395: (rallytime) Back-port #45380 to 2017.7.3 @ *2018-01-12 18:49:20 UTC*
  - **PR** #45380: (twangboy) Backport changes from #45308 (refs: #45395)
  - **PR** #45308: (twangboy) Fix *integration.modules.test_state* for Windows (refs: #45380)
  - c3fdd1dcc4 Merge pull request #45395 from rallytime/bp-45380
  - 0356b3d56f Backport changes from #45308
- **ISSUE** #44107: (anlutro) salt-ssh 2017.7 doesn't work with Python 3, missing backports_abc (refs: #45294)
- **PR** #45294: (gtmanfred) include backports_abc @ *2018-01-11 18:18:16 UTC*
  - f7da716d32 Merge pull request #45294 from gtmanfred/2017.7
  - 3633ceeaa7 Merge branch '2017.7' into 2017.7
  - 29806e4496 ignore salt.ext in pylint
  - 8b597a4890 include backports_abc
- **ISSUE** #43130: (boltronics) module.run documentation issues (refs: #45381)
- **PR** #45381: (gtmanfred) fix module.run docs @ *2018-01-11 18:02:38 UTC*
  - f77a3e9cd4 Merge pull request #45381 from gtmanfred/module.run
  - 230e899192 fix module.run docs
- **ISSUE** #43995: (dragonpaw) Using zmq built with –enable-draft breaks Salt (refs: #45368)
- **PR** #45368: (DmitryKuzmenko) Fixes to work with pyzmq with –enable-drafts @ *2018-01-11 17:53:16 UTC*
  - 8efd29f4d9 Merge pull request #45368 from DSRCorporation/bugs/zmq_draft
  - 7622e355cf Minor: removed a stale comment.
  - 00f31bf9b5 Fixes to work with pyzmq with –enable-drafts
- **PR** #45371: (rallytime) Back-port #45158 to 2017.7 @ *2018-01-11 17:51:38 UTC*

  - **PR** #45158: (terminalmage) Fix integration.modules.test_state.StateModuleTest.test_exclude (refs: #45371)
  - 22c3efda06 Merge pull request #45371 from rallytime/bp-45158
  - 3565bc2bf2 Don't use include-test SLS in orch tests
  - 8bc17e0d7a Fix integration.modules.test_state.StateModuleTest.test_exclude
- **PR** #45387: (renner) Set SHELL environment variable @ *2018-01-11 16:23:21 UTC*
  - **PR** #40630: (mateiw) develop: SUSE specific changes to salt-api.service (refs: #45387)
  - **PR** #40620: (mateiw) SUSE specific changes to salt-api.service (refs: #40630, #45387)
  - 3a0e2de995 Merge pull request #45387 from renner/patch-2
  - 530ddd2d29 Set SHELL environment variable
- **PR** #45388: (terminalmage) Fix loader error in 2017.7 tests @ *2018-01-11 16:13:53 UTC*
  - dcf98a2260 Merge pull request #45388 from terminalmage/fix-test-loader-error
  - 5473c085d9 Fix loader error in 2017.7 tests
- **PR** #45382: (terminalmage) Skip flaky test on 2017.7 branch @ *2018-01-11 14:23:05 UTC*
  - d15f9e1020 Merge pull request #45382 from terminalmage/salt-jenkins-686
  - ff3039db6c Skip flaky test on 2017.7 branch
- **PR** #45369: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2018-01-10 22:14:05 UTC*
  - dbe21b2c0d Merge pull request #45369 from rallytime/merge-2017.7
  - f65e091df8 Merge branch '2016.11' into '2017.7'
    - 0959ae4ea3 Merge pull request #45327 from lomeroe/bp-44861_2016.11
      - 784139f734 Check for values other than 0 or 1
    - a6db5f95f0 Merge pull request #45268 from damon-atkins/2016.11_win_pkg_pkg_install_latest
      - 325a9f0f66 Update 2016.11.9.rst
      - 4da9200b9c Update 2016.11.9.rst
      - 126aee36ac Update 2016.11.9.rst
      - 1c01967943 Update 2016.11.9.rst
      - a0d89882b8 Fix pkg.install packagename version=latest i.e. if on an old version upgrade to the latest
- **PR** #45379: (rhoths) Minor spelling/grammar fixes in the highstate returner documentation @ *2018-01-10 20:09:52 UTC*
  - 55979b3a48 Merge pull request #45379 from rhoths/rhoths-doc-highstate-1
  - afbbd492cd Minor spelling/grammar fixes in highstate returner
- **PR** #45358: (UtahDave) gate the minion data cache refresh events. @ *2018-01-10 17:21:05 UTC*
  - **PR** #45299: (garethgreenaway) [2017.7] config gate auth_events (refs: #45358)
  - 541e59fa75 Merge pull request #45358 from UtahDave/gate_data_cache_refresh
  - 379b6cd23e should be *self*, not *salt*
  - a82e158f2d gate the minion data cache refresh events.

- **PR** #45297: (Ch3LL) Allow macosx service state tests to check for pid return @ *2018-01-09 20:47:24 UTC*
  - fb87010461 Merge pull request #45297 from Ch3LL/mac_service_state
  - 4e569b5802 Allow macosx service state tests to check for pid return
- **PR** #45351: (dmurphy18) Update debbuild to explicitly include source code for Debian, Ubuntu @ *2018-01-09 17:21:51 UTC*
  - beedf6e815 Merge pull request #45351 from dmurphy18/upd_debbuild
  - 478dc70092 Update debbuild flags
- **PR** #45299: (garethgreenaway) [2017.7] config gate auth_events (refs: #45358) @ *2018-01-09 15:00:30 UTC*
  - 66da9b47bc Merge pull request #45299 from garethgreenaway/config_gate_auth_events
  - 9a15ec3430 Updating versionadded string. Fixing typo.
  - edfc3dc078 Adding in documention for *auth_events* configuration option
  - 3ee4eabffd Fixing small typo
  - 6a28bddcc9 Adding some code to config gate if auth_events are sent
- **PR** #44856: (Ch3LL) Add state.running ssh integration test @ *2018-01-08 21:40:50 UTC*
  - 8d04c2b3d4 Merge pull request #44856 from Ch3LL/running_test
  - 9a35a73711 add time limit to while loop
  - aeb5f4e248 Add state.running ssh integration test
- **ISSUE** saltstack/salt-jenkins#675: (rallytime) [2017.7] unit.states.test_file.TestFileState.test_directory is failing on Fedora 27 and CentOS 6 (refs: #45295)
- **PR** #45295: (gtmanfred) test directory that doesn't exist @ *2018-01-08 20:59:53 UTC*
  - d0e5e70277 Merge pull request #45295 from gtmanfred/test_directory
  - e6178fe6d4 Merge branch '2017.7' into test_directory
  - 24114e91c1 test was different slightly on 2017.7
  - d20fc93625 test directory that doesn't exist
- **ISSUE** saltstack/salt-jenkins#678: (rallytime) [2017.7] Proxy Minion Tests for Py3 are failing (refs: #45302)
- **PR** #45302: (gtmanfred) fix proxy tests for py3 on 2017.7 @ *2018-01-08 17:41:58 UTC*
  - f49b204b75 Merge pull request #45302 from gtmanfred/proxyp3
  - b295ec0429 make dummy proxy module py3 compatible
  - 8736e21f65 fix starting proxy minion on py3
  - e2824a7253 fix py3 tests
- **PR** #45279: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2018-01-08 17:26:49 UTC*
  - eea7158e82 Merge pull request #45279 from rallytime/merge-2017.7
  - 8025b14584 Merge branch '2016.11' into '2017.7'
    - 1c5e905b61 Merge pull request #45256 from rallytime/bp-45034
      - 68f971b38f Apply test fixes from #45034 to parsers_test.py
      - 9454236694 Fix for pidfile removal logging
- **PR** #44853: (gtmanfred) remove not from vault utils @ *2018-01-05 17:43:18 UTC*

  - dab4a8cff3 Merge pull request #44853 from gtmanfred/vault
  - bfee1cead6 set role for loading minion config
  - c5af2e5048 if utils is not loaded, load it
  - 6a5e0f9ac1 remove not from vault utils
- **PR** #45277: (rallytime) Back-port #45025 to 2017.7 @ *2018-01-05 15:35:53 UTC*
  - **PR** #45025: (steverweber) Fix pillar include merge order (refs: #45277)
  - f09d0e5fdb Merge pull request #45277 from rallytime/bp-45025
  - 942c14bb29 pillar body overrides includes
  - 1152202fdc fix pillar includes from merging over the current sls defines
- **PR** #45276: (rallytime) Back-port #45260 to 2017.7 @ *2018-01-05 14:45:40 UTC*
  - **PR** #45260: (gtmanfred) Make some kitchen-salt tests blue (refs: #45276)
  - fc84f1104f Merge pull request #45276 from rallytime/bp-45260
  - 9ab1af738f switch kitchen-salt to use rsync transport to preserve symlinks
  - cf98ed472e fix up symlinks
- **ISSUE** #43340: (syphernl) Upgrading Salt via Salt results in dying minions and broken dpkg (refs: #45255)
- **PR** #45255: (rallytime) Back-port #44427 to 2017.7 @ *2018-01-04 21:46:17 UTC*
  - **PR** #44427: (samodid) use KillMode=process for salt-minion.service (refs: #45255)
  - ff9880c498 Merge pull request #45255 from rallytime/bp-44427
  - 6ceafbbf3a use KillMode=process for salt-minion.service
- **ISSUE** #23454: (HontoNoRoger) SLS rendering error with Salt-SSH (pydsl) (refs: #45251)
- **PR** #45251: (forksaber) Fix #23454 : make pydsl work with salt-ssh @ *2018-01-04 21:33:09 UTC*
  - e715eb603f Merge pull request #45251 from forksaber/salt-ssh-pydsl
  - b3660d5190 [#23454] make pydsl work with salt-ssh
- **PR** #45254: (Ch3LL) Add darwin value for ssh grain items tests on MacOSX @ *2018-01-04 21:31:35 UTC*
  - 2934b60d53 Merge pull request #45254 from Ch3LL/fix_mac_grain_ssh
  - b4b59b89cd remove platform from salt.utils call for 2017.7
  - 85e853a63d Add darwin value for ssh grain items tests on MacOSX
- **PR** #45135: (twangboy) Fix win_dacl problems with SIDs @ *2018-01-04 21:01:48 UTC*
  - af2d880303 Merge pull request #45135 from twangboy/win_fix_dacl
  - b31e08946a Merge branch '2017.7' into win_fix_dacl
  - 35a417f510 Fix win_dacl problems with SIDs
- **ISSUE** #43806: (Ch3LL) Add spm man Test to Auto Test Suite (refs: #44930)
- **PR** #44930: (frogunder) man_spm_test @ *2018-01-04 20:58:02 UTC*
  - d0a3770035 Merge pull request #44930 from frogunder/man_spm
  - 48e6953e1f fix_string_error
  - c9fa4ed2a7 man_spm_test

- **PR** #45259: (Ch3LL) Fix MacOSX Service Status Check and integration test @ *2018-01-04 14:25:01 UTC*
  - 543eebf411 Merge pull request #45259 from Ch3LL/fix-mac-service-test
  - 74e6ed60ea Fix MacOSX Service Status Check and integration test
- **PR** #45263: (sumeetisp) Updating python version for 2017.7 @ *2018-01-04 14:16:26 UTC*
  - bbbd1872a7 Merge pull request #45263 from sumeetisp/2017.7
  - e3a5ee3a08 Merge branch '2017.7' into 2017.7
  - 71aea9a3bc Merge pull request #1 from sumeetisp/sumeetisp-python-version
    - 1b4806e2b9 Updating python version
- **PR** #45244: (twangboy) Fix search/replace in Py3 @ *2018-01-04 14:02:22 UTC*
  - d46e1197be Merge pull request #45244 from twangboy/win_fix_portable.py
  - e3a8279c01 Get path to python binary based on executable
  - 03aec37040 Fix search/replace in Py3
- **PR** #45233: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2018-01-03 15:34:00 UTC*
  - eba360870a Merge pull request #45233 from rallytime/merge-2017.7
  - a3d251b2cd Merge branch '2016.11' into '2017.7'
    - b75f50afe3 Merge pull request #45235 from rallytime/bp-45209
      - 2d0a9bbf7e enable UsePAM for ssh tests
  - 5d9a1e91e9 Merge branch '2016.11' into '2017.7'
    - 3ab962b01a Merge pull request #44965 from gtmanfred/2016.11
      - a5d8a6340e check if VALUE is a string_type
    - 40fb30f63f Merge pull request #45232 from rasathus/2016.11
      - 7a2bd8f49b Merge branch '2016.11' into 2016.11
      - de53c45c29 Backport #27160 to 2016.11
- **PR** #45175: (amendlik) Pkg uptodate @ *2018-01-02 17:38:36 UTC*
  - 693cc807e8 Merge pull request #45175 from amendlik/pkg-uptodate
  - 4f514a29a7 Merge branch '2017.7' into pkg-uptodate
- **PR** #45226: (gtmanfred) Update kitchen to use runtests verifier on 2017.7 @ *2017-12-31 18:13:28 UTC*
  - 1b3f3ba1be Merge pull request #45226 from gtmanfred/2017.7
  - 4f3b089e0e fix copying back
  - f56f062a6a download xml for junit
  - 7cc342a5d6 use new runtests verifier
- **PR** #45221: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2017-12-30 18:08:29 UTC*
  - 7d3a6cbc65 Merge pull request #45221 from rallytime/merge-2017.7
  - 508599e159 Merge branch '2016.11' into '2017.7'
  - 707ef55175 Merge pull request #45161 from lomeroe/bp-44944_2016.11
    - 0a4c6b5a83 remove references to six.unichr

    - f3196d795d lint fixes for static regexes
    - 11b637d108 lint fixes
    - c14d6282ad do not decode registry.pol file wholesale, but instead decode individual elements of the file
  - 6f52034e08 Merge pull request #45199 from gtmanfred/status
    - fb07f9ea7d status.pid returns pid ids not process names
- **ISSUE** #45176: (thuhak) osquery execution module does't work with attrs parameter (refs: #45204)
- **PR** #45204: (garethgreenaway) [2017.7] Fixes to osquery module & addition of unit tests @ *2017-12-30 13:25:38 UTC*
  - abed378981 Merge pull request #45204 from garethgreenaway/45176_fixes_to_osquery_module
  - dc933e9e24 Fixing typo
  - d834bd1b6f Fixing some minor lint issues.
  - 4738205154 Fixing a bug when attributes are passed to various osquery module functions.
    - 66884334d9 Update states.pkg for Python3 compatibility
    - 2a7d76ad6e Fail pkg.uptodate if expected packages are not upgraded
    - 29ef67bac2 Test pkg.uptodate with failed upgrades
    - 23ab93353b Produce changes dict for pkg.uptodate dry-run mode
    - 7c67ec39d9 Add tests for pkg.uptodate state
- **PR** #45203: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2017-12-29 01:11:03 UTC*
  - 5991d8ca15 Merge pull request #45203 from rallytime/merge-2017.7
  - 430c913c8c Merge branch '2016.11' into '2017.7'
    - d3381e27d0 Merge pull request #45118 from garethgreenaway/44728_nodegroups_seq
      - 0ff811de70 Swapping import to be the old path for 2016.11
      - b3e2f388f5 Fix to allow nodegroups to include sequences
    - f969aca3a3 Merge pull request #45127 from twangboy/win_fix_pkg
      - 14639739f2 Fix issue with 1641 return code
    - dc357b39f0 Merge pull request #45137 from twangboy/win_fix_reg_tests
      - b6f4ef8d73 Catch correct error type in list_keys and list_values
    - 0aa1662731 Merge pull request #45130 from rallytime/api-groups
      - 2dcc8df845 Resolve groups for salt api
    - 7dc3cc4641 Merge pull request #45114 from twangboy/win_fix_pam
      - cf5eae1f77 Move pam library load to try/except block
- **PR** #45201: (rallytime) [2017.7] Check for running on python3 before decoding bytes @ *2017-12-28 22:59:14 UTC*
  - **PR** #45090: (angeloudy) fix TypeError in python 3 (refs: #45201)
  - 882267314f Merge pull request #45201 from rallytime/fix-jinja-template-test-failure
  - b4af3bdff8 Check for running on python3 before decoding bytes

- **PR** #45200: (rallytime) [2017.7] Fix docstring integration test failure @ *2017-12-28 22:58:34 UTC*
    - **PR** #44552: (Da-Juan) pip_state: Check if available upgrades fulfill version requirements. (refs: #45200)
    - 2e18398f12 Merge pull request #45200 from rallytime/fix-docstring-test-failure
    - a26d4795bd [2017.7] Fix docstring integration test failure
- **PR** #45186: (rallytime) Back-port #44922 to 2017.7 @ *2017-12-28 19:02:51 UTC*
    - **PR** #44922: (dincamihai) Fix salt-master for old psutil (refs: #45186)
    - 67d97303b5 Merge pull request #45186 from rallytime/bp-44922
    - 6970fe8103 Fix salt-master for old psutil
- **PR** #44624: (eliasp) Fix Traceback when using the *service.enabled* state on non-booted systems @ *2017-12-28 10:58:43 UTC*
    - 30d7f7257a Merge pull request #44624 from eliasp/fix-upstart-utmp-exception
    - 43d44e051a Do not blindly assume presence of either */var/run/utmp* or */run/utmp*, none of both might be available (e.g. on non-booted systems).
- **PR** #45183: (twangboy) Add libnacl dependency @ *2017-12-27 22:08:32 UTC*
    - 3832e7b227 Merge pull request #45183 from twangboy/win_add_libnacl_2017.7
    - b46845888d Add libnacl dependency
- **ISSUE** #44928: (rcallphin) Duplicating master token when no match for Minion policy (Vault Module) (refs: #44966)
- **PR** #44966: (rcallphin) Fix bug with vault runner creating token on empty policy @ *2017-12-22 20:30:37 UTC*
    - fbbf33574e Merge pull request #44966 from rcallphin/fix-bug-vault-empty-policy
    - 7f327ab760 Lint: Remove extra whitespace
    - 04ab6a5e9d Merge branch '2017.7' into fix-bug-vault-empty-policy
    - 5be463bb46 Merge branch '2017.7' into fix-bug-vault-empty-policy
    - 48d9cc3674 Fix bug with vault runner creating token on empty policy
- **PR** #44552: (Da-Juan) pip_state: Check if available upgrades fulfill version requirements. (refs: #45200) @ *2017-12-22 19:25:17 UTC*
    - 487207f61d Merge pull request #44552 from Da-Juan/avoid_unneeded_pip_install
    - 49a6a8f02e Merge branch '2017.7' into avoid_unneeded_pip_install
    - 3a8e62493d pip_state: Check if available upgrades fulfill version requirements
    - 62252d74d9 pip_state: Compare versions using pkg_resources.parse_version
    - 5219ab974c Add list_all_versions function to pip module
- **PR** #45090: (angeloudy) fix TypeError in python 3 (refs: #45201) @ *2017-12-22 18:11:13 UTC*
    - 5ae26f0c09 Merge pull request #45090 from angeloudy/2017.7
    - cf411f8984 Merge branch '2017.7' into 2017.7
    - 177fd18671 fix TypeError in python 3
- **ISSUE** #44315: (whytewolf) cmd.* cwd does not escape spaces. 2017.7.2 (refs: #45134)
- **PR** #45134: (garethgreenaway) [2017.7] fix to cmd.script for cwd with space @ *2017-12-22 15:31:24 UTC*

  - a1946730a9 Merge pull request #45134 from garethgreenaway/44315_cmd_script_cwd_with_space
  - 48eafe3206 Adding some tests to tests cmd.script with cwd
  - 8dfcf71b08 Adding _cmd_quote to handle cases when the current working directory for cmd.script might have a space in it.
- **PR** #44964: (Giandom) added-highstate-output-to-slack-engine @ *2017-12-21 21:32:01 UTC*
  - f41adfc913 Merge pull request #44964 from Giandom/2017.7-added-highstate-output-to-slack-engine
  - 4526c158f1 added-highstate-output-to-slack-engine
  - 573a0a4143 added-highstate-output-to-slack-engine
  - 9a6e03ce6e added-highstate-output-to-slack-engine
- **PR** #45124: (gtmanfred) enable using kitchen-salt with ec2 on 2017.7 @ *2017-12-21 19:11:27 UTC*
  - b49ee97938 Merge pull request #45124 from gtmanfred/2017.7
  - d0586013eb fix pylint
  - 59e2e56d13 chmod the xml files before trying to copy
  - a5c1410e23 catch IOError when copying xml files back
  - 23bd38ad66 enable using kitchen-salt on ec2
- **PR** #45087: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2017-12-20 22:24:51 UTC*
  - 42e894570d Merge pull request #45087 from rallytime/merge-2017.7
  - fe81e2d39a Merge branch '2016.11' into '2017.7'
    - 7e128e8f15 Merge pull request #45100 from rallytime/bp-45070
      - 0bdb46dab9 add clouds modules to index
    - bdf93f339d Merge pull request #45098 from rallytime/bp-45092
      - 80b6bd6813 Fix integration.states.test_pip.PipStateTest.test_pip_installed_weird_install
  - 4f21a2bbfd Merge branch '2016.11' into '2017.7'
    - 324b7d4058 Merge pull request #44078 from rossengeorgiev/fix-41044
      - a81a6fe23c fix #41044; allow for date param to be 0
    - 48a59761df Merge pull request #44970 from rallytime/update-bootstrap-script
      - b2c8057427 Update bootstrap script to latest release: 2017.12.13
    - 637fdaed58 Merge pull request #45069 from rallytime/bp-45040
      - aa438e1605 Installation Fails on headless machines.
    - 4d6d640381 Merge pull request #44969 from rallytime/bp-41305
      - 5c4bee43dc correct accept_vpc_peering_connection
    - 10de468f13 Merge pull request #45031 from terminalmage/fix-mysql-returner
      - f3bd12c27c Fix invalid exception class in mysql returner
    - 9a7406207f Merge pull request #44972 from terminalmage/bp-44958
      - a416bf0112 No need to manually do connect_pub, use listen=True in run_job
      - 3ec004bd2e Fix a race condition in manage runner

        - 1032ca3290 Merge pull request #44385 from gtmanfred/schedule
            - 9e15c38da2 add comma
            - 855d933cb7 schedule should be a dict
- **PR** #45112: (Ch3LL) Fix spm big file build test to check /tmp @ *2017-12-20 22:09:21 UTC*
    - 9550e742ac Merge pull request #45112 from Ch3LL/fix-arch
    - 1bd7110a14 Fix spm big file build test to check /tmp
- **ISSUE** #44303: (mwerickso) boto3_route53 module times out on retries (refs: #44976)
- **PR** #45068: (rallytime) Back-port #44976 to 2017.7 @ *2017-12-20 16:31:22 UTC*
    - **PR** #44976: (tkwilliams) Fix bad variable name in boto3_route53 module - resolves #44303 (refs: #45068)
    - 71f9c7ee49 Merge pull request #45068 from rallytime/bp-44976
    - 0ca0f37805 44303 - resolves #44303
- **ISSUE** #44961: (golmaal) The archive tar function fails to untar file when dest argument is passed (refs: #44983)
- **PR** #45099: (rallytime) Back-port #44983 to 2017.7 @ *2017-12-20 14:41:22 UTC*
    - **PR** #44983: (golmaal) Ref:44961 - Modified archive.tar to add dest at the end of the tar cmd (refs: #45099)
    - 54a33c0e1d Merge pull request #45099 from rallytime/bp-44983
    - 23361de8a2 Ref:44961 - Modified archive.tar to add dest argument at the end of the tar cmd.
- **ISSUE** #43533: (Ch3LL) Add status.pid Test to Auto Test Suite (refs: #44650)
- **PR** #44650: (frogunder) add status.pid test @ *2017-12-19 16:21:09 UTC*
    - e0d7b330fa Merge pull request #44650 from frogunder/status
    - 904c0da893 Merge branch '2017.7' into status
    - 619bd2be1e fix lint error
    - d406cb07a3 add status.pid test
- **ISSUE** #44516: (doesitblend) Windows PY3 Minion Returns UTF16 UnicodeError (refs: #45161, #44944)
- **PR** #44944: (lomeroe) win_lgpo registry.pol encoding updates (refs: #45161) @ *2017-12-19 14:42:49 UTC*
    - 422d8b8f1b Merge pull request #44944 from lomeroe/update_regpol_encoding
    - 07d04c7bc7 lint fixes for static regexes
    - d17c46ce41 lint fixes
    - ab8e431729 do not decode registry.pol file wholesale, but instead decode individual elements of the file
- **PR** #44938: (The-Loeki) Libcloud dns fixes @ *2017-12-18 15:47:18 UTC*
    - d9a4b9681e Merge pull request #44938 from The-Loeki/libcloud_dns_fixes
    - 276e8828ae libcloud_dns: pylint fix
    - c994423286 Merge branch '2017.7' into libcloud_dns_fixes
- **PR** #44951: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2017-12-16 13:16:24 UTC*
    - 5137be01ec Merge pull request #44951 from rallytime/merge-2017.7
    - a0d2dd2069 Lint fix
    - 9db4179462 Merge branch '2016.11' into '2017.7'

        - 68d901b12c Merge pull request #44770 from cruscio/2016.11
            - e2682bf441 Fix minion ping_interval documentation
        - d4ab55ec47 Merge pull request #44335 from gtmanfred/2016.11
            - 3f1268d67f fix patching for python 2.6
            - 1d0bd5bb32 Merge branch '2016.11' into 2016.11
            - f02b02032d Merge pull request #4 from terminalmage/pr-44335
            - b4eb1527a6 Add test for PR 44335
            - a30af3252e add docker-ce to docker subtype grains check
- **PR** #44995: (twangboy) Fix *unit.modules.test_file* for Windows @ *2017-12-15 17:05:49 UTC*
    - 698b04779e Merge pull request #44995 from twangboy/win_fix_atomicfile
    - 8316481944 Comment the salt import
    - fe34f0c877 Set owner properly on Windows
- **ISSUE** #44934: (vernondcole) http.wait_for_successful_query does not pause for documented intervals (refs: #44968)
- **PR** #44968: (gtmanfred) fix http wait for state @ *2017-12-14 20:06:01 UTC*
    - 2e1a57b9bc Merge pull request #44968 from gtmanfred/http
    - ca6936f6eb fix http wait for state
        - c72db283d5 libcloud_dns: Further fixes to state output, pylint fixes
        - e9bbc23b11 Merge branch '2017.7' into libcloud_dns_fixes
- **ISSUE** #44811: (xuhcc) rbenv.installed fails when rbenv installed globally (refs: #44900)
- **PR** #44900: (xuhcc) Fix TypeError during rbenv ruby installation when rbenv is not found @ *2017-12-14 17:37:14 UTC*
    - c4f0894689 Merge pull request #44900 from xuhcc/rbenv-ret-fix
    - fdd8310c31 Merge branch '2017.7' into rbenv-ret-fix
    - bfd0972d25 Fix TypeError during rbenv ruby installation when rbenv is not found
- **PR** #44974: (twangboy) Skip test_log_created on Windows @ *2017-12-14 13:59:25 UTC*
    - f0c2cf3cec Merge pull request #44974 from twangboy/win_skip_test_parsers
    - 40665d7b08 Skip test_log_created on Windows
- **ISSUE** #44820: (msteed) Custom returner breaks manage runner (refs: #44958)
- **PR** #44958: (terminalmage) Fix a race condition in manage runner (refs: #44972) @ *2017-12-13 15:20:36 UTC*
    - dad2d723ca Merge pull request #44958 from terminalmage/issue44820
    - ef749abfc6 No need to manually do connect_pub, use listen=True in run_job
    - 2ac70cfab5 Fix a race condition in manage runner
- **PR** #44956: (terminalmage) Avoid traceback when bogus value in pidfile @ *2017-12-13 14:30:12 UTC*
    - db58345abb Merge pull request #44956 from terminalmage/fix-get_pidfile
    - d66f3a98d7 Avoid traceback when bogus value in pidfile
- **ISSUE** #44932: (knine) ACLs Not Completely Verified (refs: #44945)

- **PR** #44945: (gtmanfred) Fix handling of effective acls @ *2017-12-12 21:49:34 UTC*
  - e8e3b3c8ff Merge pull request #44945 from gtmanfred/2017.7
  - 66bb755751 add test for effective acls
  - 0ff52a93dd use last entry in acl
- **PR** #44942: (rallytime) Update README with SaltConf18 info @ *2017-12-12 21:47:23 UTC*
  - 47dc7b7afb Merge pull request #44942 from rallytime/readme-saltconf-update
  - d1317c44e2 Update README with SaltConf18 info
- **ISSUE** #44665: (mvivaldi) Documentation of salt renders jinja (refs: #44943, #44895)
- **PR** #44943: (mvivaldi) Fix for the jinja documentation @ *2017-12-12 20:20:41 UTC*
  - 7572982419 Merge pull request #44943 from mvivaldi/filters-doc
  - d23ac4eabc Fix for the jinja documentation
- **ISSUE** #43417: (damon-atkins) win_pkg: pkg.install and pkg.remove general issues (refs: #44832, #43708)
- **PR** #44832: (damon-atkins) win_pkg: Merge full copy of 2016.11 with many fixes and improvements to 2017.7 @ *2017-12-12 18:30:06 UTC*
  - 465cacad83 Merge pull request #44832 from damon-atkins/2017.7_replace_with_newer_2016.11_win_pkg
  - a4f0b41ba2 Should be a smaller change set since recent update from 2016.11
  - 695334b201 Merge branch '2017.7_replace_with_newer_2016.11_win_pkg' of github.com:damon-atkins/salt into 2017.7_replace_with_newer_2016.11_win_pkg
    - 843e204582 Merge branch '2017.7' into 2017.7_replace_with_newer_2016.11_win_pkg
  - 4b60b1ec84 Merge remote branch 'refs/remotes/upstream/2017.7' into 2017.7_replace_with_newer_2016.11_win_pkg
  - b46f818a57 Raise a PR to fix 2016 issues commited here, fixed issues with merge.
  - 32ef1e12ae Merge branch '2017.7' into 2017.7_replace_with_newer_2016.11_win_pkg
  - 494835c3f2 I backported develop and applied a long list of fixes to 2016.11 this brings these fixes into 2017.7 - Software was not always being removed, general if & was in the string or msi was downloaded to uninstall the software - pkg.list_upgrades failed. Added support for 'latest' and 'Not Found' for version_cmp() to fix this. - output fixes - pkg.list_available no longer forces a pkg.refresh_db this is no longer required, as by default it will update if older than 6 hours - cmd /s /c is prefixed for all commands i.e. installs and removes. - cmd are now strings, instead of a list when using cmd.run. As windows only supports strings. And the " were being broken
- **PR** #44754: (twangboy) Fix inet_pton for Windows on Py3 @ *2017-12-12 14:04:20 UTC*
  - a811a92b17 Merge pull request #44754 from twangboy/win_fix_inet_pton
  - 25a20109fe Merge branch '2017.7' into win_fix_inet_pton
  - 849b99eb34 Merge branch '2017.7' into win_fix_inet_pton
  - df1e6a202b Use salt.ext.six
  - 5ac8112585 Use six to ensure unicode value
  - 9b5d8c421b Handle unicode values
- **PR** #44931: (pkruk) add missing parenthis to keep integration with python3 @ *2017-12-12 13:49:39 UTC*
  - 53b34e24cd Merge pull request #44931 from pkruk/fix-missing-parenthis

  - b1ed739b44 Merge branch '2017.7' into fix-missing-parenthis
  - 4f1b1f12d2 Merge branch 'fix-missing-parenthis' of https://github.com/pkruk/salt into fix-missing-parenthis
    - 3475d3fa01 add missing parenthis to keep integration with python3
  - adf38cacfb add missing parenthis to keep integration with python3
    - ad55e33f57 libcloud_dns: fix state output
    - a68d594e3a libcloud_dns: copy args before deleting from them
- **PR** #44891: (twangboy) Fix issue with unsafe path in Windows jenkins tests @ *2017-12-11 21:10:43 UTC*
  - ba6146250a Merge pull request #44891 from twangboy/win_fix_verify
  - 7232579167 Allow test suite file_roots as a safe path
- **PR** #44921: (Ch3LL) Add test to ensure log files are created @ *2017-12-11 18:24:16 UTC*
  - 85160fd297 Merge pull request #44921 from Ch3LL/log_test
  - 3bb58fb577 skip salt-key log creation test
  - 6a379195bc Add test to ensure log files are created
- **PR** #44787: (rallytime) GroupAdd test: Add destructive test decorator to entire class @ *2017-12-11 18:14:18 UTC*
  - 54d29a61cb Merge pull request #44787 from rallytime/groupadd-destructive-clean
  - 817ac002b0 Add destructive test decorator to test class
- **ISSUE** #44665: (mvivaldi) Documentation of salt renders jinja (refs: #44943, #44895)
- **PR** #44895: (mvivaldi) Jinja Filters doc @ *2017-12-11 15:32:07 UTC*
  - 0292e3612a Merge pull request #44895 from mvivaldi/filters-doc
  - 62409d608a Added Escape Filters and Set Theory Filters in jinja documentation
- **PR** #44879: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2017-12-10 16:53:44 UTC*
  - **PR** #44855: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 (refs: #44879)
  - df28f312ac Merge pull request #44879 from rallytime/merge-2017.7
  - 23c5a4ca3e Merge branch '2016.11' into '2017.7'
    - bb1f8dceaf Merge pull request #44579 from roaldnefs/fix-cron-identifier
      - df73a4c051 Merge branch '2016.11' into fix-cron-identifier
    - af0131fa1f Merge pull request #44852 from damon-atkins/2016.11_win_pkg_typo_n_fix
      - 0e7c19084f Lint: Remove extra whitespace
      - 7c7e21f94d Fix spelling typo, and fix backwards campatible minion option for repo location
    - 88c0d66b4e Merge pull request #44794 from terminalmage/issue44365
      - 3b8b6f25e6 Remove debugging line
      - 153bf45b03 Fix regression in file.managed when source_hash used with local file
    - c8bb9dfbbb Merge pull request #44738 from rallytime/bump-oxygen-warnings
      - ead3c569e1 Bump deprecation warnings from Oxygen to Fluorine
    - 88e3aab00d Merge pull request #44741 from gtmanfred/rhip

      - 439dc8dce6 if gateway is not specified use iface
      - 3ec4329307 Merge branch '2016.11' into fix-cron-identifier
      - 99fa05a456 Fix for bug in cron state
      - 97328faeac Fix for bug in cron module

- **PR** #44880: (UtahDave) Determine windows hardware arch correctly @ *2017-12-08 22:24:09 UTC*
  - 8e14bc3941 Merge pull request #44880 from UtahDave/2017.7local
  - 6e3c7ac1ac Merge branch '2017.7' into 2017.7local
- **PR** #44861: (twangboy) Fix win_lgpo for unknown values (refs: #45327) @ *2017-12-08 18:52:05 UTC*
  - dc51174670 Merge pull request #44861 from twangboy/win_fix_lgpo_invalid_value
  - 89f65e19ff Check for values other than 0 or 1
- **PR** #44621: (isbm) Bugfix: errors in external pillar causes crash, instead of report of them @ *2017-12-08 18:46:56 UTC*
  - f5a143f8c5 Merge pull request #44621 from isbm/isbm-bsc1068446-2017.7
  - 0d2675c4fe Use variable, instead of direct value
  - 1ddc47da0a Add unit test for _get_pillar_errors when external pillar is clean and internal contains errors
  - 68480d5dc9 Add unit test for _get_pillar_errors when both external and internal pillars contains errors
  - 218a59e93b Add unit test for _get_pillar_errors when external pillar has errors and internal is clean
  - 3ce19356c2 Add unit test for _get_pillar_errors when external and internal pillars are clean
  - 67034139d9 Fix unit test: wrong error types in side effect
  - d9359bca13 Bugfix: unit test mistakenly expects pillar errors as a string, while it is a list
  - 8c2bdc696b Bugfix: do not pull '_errors' from unchecked objects
  - d5e30999c7 Remove unused variable (no exception, within the try/finally block)
  - aad668d559 Fix and clarify docstring.
  - c2c47e4e71 Rename function from ambiguous name
  - 265de8e61c Bugfix the logic according to the exact described purpose of the function.
    - dae9c6aa5c Determine windows hardware arch correctly
- **PR** #43379: (twangboy) Fix file.managed on Windows with test=True @ *2017-12-07 21:10:43 UTC*
  - abe089ad54 Merge pull request #43379 from twangboy/win_fix_file.managed
  - edcd581ca5 Merge branch '2017.7' into win_fix_file.managed
  - a27bb6993a Fix py3 error
  - 0ff9fa498a Fix test_directory
  - 187bc1e61e Add back the try/finally blocks
  - d7241d004f Fix 2 more tests
  - d5dd42aebe Fix integration tests for Windows
  - d56bc9aae9 Fix typo
  - af5565859e Use file functions for symlink and remove

  - 72ac59c991 Fix some more integration tests for Linux
  - 3f0499cbc4 Fix some integration tests
  - a24b964ea5 Fix unit test to handle new Exception
  - e3c3845f73 Raise CommandExecutionError when file doesn't exist
  - 4602f499a2 Remove loader module mixin, add linux paths
  - 99b27c037f Add tests to avoid future regression
  - 5c215ed8c2 Fix documentation formatting
  - 6a4e77e4b9 Return empty or unmodified dict on file not found
- **ISSUE** #44565: (arthurlogilab) NameError: global name '__jid_event__' is not defined when running a runner in the scheduler (refs: #44570)
- **PR** #44570: (gtmanfred) Include client mixin globals in scheduler for runner modules @ *2017-12-07 20:23:33 UTC*
  - cf4cbcd340 Merge pull request #44570 from gtmanfred/2017.7
  - 7b17f9f63c Merge branch '2017.7' into 2017.7
- **PR** #44494: (skizunov) Fix broken *beacons_before_connect* feature @ *2017-12-07 18:24:49 UTC*
  - **PR** #38289: (skizunov) Add config options for running beacons/scheduler before connect (refs: #44494)
  - febb913743 Merge pull request #44494 from skizunov/develop2
  - 7adcfbf8ec Merge branch '2017.7' into develop2
- **ISSUE** #44298: (skjaro) ipset state check problem (refs: #44356)
- **ISSUE** #39552: (Xiami2012) ipset.check new implementation by @lingonl has countless critical bugs (refs: #44356)
- **PR** #44512: (rallytime) Back-port #44356 to 2017.7 @ *2017-12-07 14:44:50 UTC*
  - **PR** #44356: (skjaro) Fix ipset state with multiple entries and subtypes separated with comma (refs: #44512)
  - 284a817565 Merge pull request #44512 from rallytime/bp-44356
  - 6f92c71834 Merge branch '2017.7' into bp-44356
  - 9a325146df Fix lint violation
  - 5aac729855 Fix check multiple entries with subtypes separated with comma
- **PR** #44748: (twangboy) Fix auto login support for OSX @ *2017-12-07 14:22:23 UTC*
  - 74ee7ce541 Merge pull request #44748 from twangboy/osx_fix_auto_login
  - 068e463870 Fix lint, add integration tests
  - 3df886df75 Fix lint, add gtmanfreds change
  - 16cb24614f Add kcpassword functionality
- **PR** #44842: (twangboy) Win fix lgpo unicode on Py3 issue @ *2017-12-07 14:21:14 UTC*
  - b60cca174c Merge pull request #44842 from twangboy/win_fix_lgpo
  - efe77999d1 Gate log.debug statement behind successful pop
  - 1c0ec79cd1 Fix py3 issue

- **PR** #44843: (twangboy) Fix 2 typos in lgpo module @ *2017-12-06 17:56:44 UTC*
  - bb58e2fec0 Merge pull request #44843 from twangboy/win_fix_lgpo_typo
  - c8f93e6dd7 Fix 2 types, shorten line lengths for spellchecking
- **PR** #44827: (mz-bmcqueen) add more clone options to virtualbox and add better dhcp handling @ *2017-12-06 15:02:23 UTC*
  - d6c37ea19c Merge pull request #44827 from mz-bmcqueen/2017.7
  - 4ead3014b7 Merge branch '2017.7' into 2017.7
  - b7ce154014 Merge branch '2017.7' of https://github.com/mz-bmcqueen/salt into 2017.7
    - 2f80f431b3 Merge branch '2017.7' into 2017.7
  - c2018c9021 fix pylint complaints
  - c38ff74261 add more clone options to virtualbox and add better dhcp handling
- **PR** #44824: (Ch3LL) Add spm -y and -f arg integration tests @ *2017-12-05 21:49:32 UTC*
  - 019169ed61 Merge pull request #44824 from Ch3LL/spm_args
  - d8f81d2e4d fix pylint
  - 61ac5cf157 Add spm -y and -f arg integration tests
- **PR** #44742: (Ch3LL) Add salt-cloud action rename integration test @ *2017-12-05 17:44:50 UTC*
  - 59b930668c Merge pull request #44742 from Ch3LL/cloud_action_test
  - 951d09ca2f remove unnecessary try/except block
  - c329ced7ee Add salt-cloud action rename integration test
- **ISSUE** #42676: (mind-code) Changes in Pillar defined Beacons only apply after Minion restart (refs: #44771)
- **PR** #44771: (garethgreenaway) [2017.7] Back porting #44071 @ *2017-12-05 17:16:06 UTC*
  - **PR** #44071: (garethgreenaway) [develop] Various fixes to beacons (refs: #44771)
  - 10442d9211 Merge pull request #44771 from garethgreenaway/42676_backport_44071
  - ec2a8b2032 Merge branch '2017.7' into 42676_backport_44071
  - 180971203e Updating minion to respond to list_available events for beacons
  - db6fcefe62 Adding list_available which is used by the add function to verify that a becaon exists.
  - e9e0318bc6 Backporting fixes related to having beacons in pillar from #44071
- **PR** #44784: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2017-12-05 17:13:49 UTC*
  - **PR** #44732: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 (refs: #44784)
  - 23d151b40a Merge pull request #44784 from rallytime/merge-2017.7-1
  - 3d9eafc4bd Lint: Remove extra empty lines at end of files
  - 239f3511bf Merge branch '2016.11' into '2017.7'
    - 97e0cf569c Merge pull request #44699 from jfindlay/attr_file
      - 9e5a40ea7c Merge branch '2016.11' into attr_file
      - 5c34607f6c utils/files remove temp file upon move failure
    - 7434e0afdf Merge pull request #44714 from rallytime/fix-44556

    - 1bbe1abeb2 Allow –static option to display state runs with highstate output
- 998d714ee7 Merge pull request #44517 from whytewolf/publish_port_doc_missing
    - 4b5855283a missed one place where i didnt chanbge master_port from my copy to publish_port
    - e4610baea5 update doc to have publish port
- 6169b52749 Merge pull request #41279 from Ch3LL/add_grain_tests
    - 1b64f15692 Merge branch '2016.11' into add_grain_tests
- dc6de050a9 Merge pull request #44563 from creideiki/pgjsonb-timestamps-44544
    - 231e412ca4 Merge branch '2016.11' into pgjsonb-timestamps-44544
- 4369df020b Merge pull request #44602 from rallytime/fix-44601
    - ff303fd060 Handle timeout_monitor/TimeoutError issues for new versions of CherryPy
- 4a4756fc37 Merge pull request #44604 from lorengordon/doc-exclude
    - c4a6c40eb3 Documents the exclude argument in state execution module
    - 15c445e6b9 Send Unix timestamps to database in pgjsonb
    - 095f1b7d7a Merge branch '2016.11' into add_grain_tests
- 91d46d4cfc Merge pull request #44434 from whytewolf/1837
    - d148e39dda change from md to rst for code reference
    - 955e305bda fix bad english, as requested by cachedout
    - 7256fcc1c9 update note to take into account grains_cache
    - 7a2981585e Merge branch '2016.11' into 1837
    - aca0405b26 add a note that describes grain rebuilding on restart and refresh
    - 9ea4db4224 mock socket.getaddrinfo
    - 78a07e30f4 add more fqdn tests and remove some of the mocking
    - 5dbf4144ce add ipv6 in opts
    - eabc1b4f9c Add fqdn and dns core grain tests
- a3bd99317f Merge pull request #44321 from gvengel/fix-file-line-diff-output
    - 69a50204a6 Add newline for lint.
    - ef7b6bbb81 Fixed issue with file.line on Windows running Python 2.
    - 8f89c99fa5 Fix FileModuleTest setUp and tearDown to work on Windows.
    - 3ac5391f5f Namespace missing functions for file.line on Windows.
    - b2b8f075b9 Fixed test to work on Windows.
    - 5a5a2dd026 Added integration test for issue #41474
    - 24d7315f1a Fix file.line diff formatting.
- 9ca563718d Merge pull request #43708 from damon-atkins/2016.11_43417_Backport_and_Fixes
    - 04d03ea6b8 Updated comment
    - 1dd565e585 Merge remote branch 'upstream/2016.11' into 2016.11_43417_Backport_and_Fixes

            - dd48ba2616 Merge remote branch 'upstream/2016.11' into 2016.11_43417_Backport_and_Fixes
            - a0d08598bf dco fix
            - 9467899fc6 Merge remote branch 'upstream/2016.11' into 2016.11_43417_Backport_and_Fixes
            - 6dc180fd0e doco fixes
            - 2496a42ea4 lint fix
            - 2c937fbe19 Merge remote branch 'upstream/2016.11' into 2016.11_43417_Backport_and_Fixes
            - c9c8c48a4d all remove/install commands are passed to cmd.exe /s /c and commands are passed as strings to cmdmod
            - 350244bd93 typo in comments and doc strings.
            - ec31f5a9bd 2017.11/develop version() was ignoring saltenv setting.
            - b314549a32 Backport of devlop to 2016.11 with additional bug fixes
        - 68ea22188e Merge pull request #44477 from rallytime/bp-44424
            - 4a9f8dcc96 Fix #44423: Handle index=None and index=0 distinctly
        - 2c89050a24 Merge pull request #44483 from terminalmage/issue44034
            - a9db8becea salt-call: account for instances where __pillar__ is empty
        - b5c2028680 Merge pull request #44489 from whytewolf/1956_log-granular-levels
            - 9cdeb4e903 update log-granular-levels to describe what they are filtering on
        - ea07f9c54c Merge pull request #44193 from twangboy/win_fix_reg
            - 44d6d9f46d Remove unused import (lint)
            - f7502436bd Fix various issues
            - 221e6e3b91 make salt.utils.to_unicode return none when passed none
            - ce41acc788 Fix many issues with reg.py
            - 4a19df1f7f Use six.text_type instead of str
            - 1b12acd303 Check type before casting
            - 03fa37b445 Cast vdata to it's proper type
        - ed8da2450b Merge pull request #43863 from nicholasmhughes/fix-atomicfile-permission-copy
            - ea852ec5d3 remove index use with stat module attributes
            - dbeeb0e917 fixes #38452 atomicfile only copies mode and not user/group perms

- **PR** #44788: (kris-anderson) Example yaml of influxdb_user state @ *2017-12-04 14:28:45 UTC*
    - 4643a112e7 Merge pull request #44788 from kris-anderson/example-yaml-of-influxdb-user-state
    - afd23d058c converted yaml example to use 2 spaces
    - 29e410c1ea added a code-block example of how the yaml should be formatted
- **ISSUE** #42713: (boltronics) 2017.7.0 master upgrade breaks mine data on non-glob matching on minions (refs: #44735)
- **PR** #44735: (gracinet) Backported issue #42713 to 2017.7 @ *2017-12-04 01:43:23 UTC*

  - 4ebac09f60 Merge pull request #44735 from gracinet/42713_backport_2017.7
  - 6806d83314 Merge branch '2017.7' into 42713_backport_2017.7
  - fb586c6dce Backported issue #42713 to 2017.7
- **PR** #44766: (twangboy) Fix *unit.utils.test_process* for Windows @ *2017-12-02 13:15:53 UTC*
  - 06ce7b7328 Merge pull request #44766 from twangboy/win_fix_test_process
  - a5737e8fc3 Fix lint errors
  - be96de09cc Fix pickling error by decorating
- **ISSUE** #44083: (ari) timezone.system fails when /etc/localtime is missing on FreeBSD (refs: #44605)
- **PR** #44716: (rallytime) Back-port #44605 to 2017.7 @ *2017-12-01 23:12:24 UTC*
  - **PR** #44605: (campbellmc) Add handling for FreeBSD in timezone.zone_compare (refs: #44716)
  - f8b8a8966d Merge pull request #44716 from rallytime/bp-44605
  - 9d43221422 Correct indentation
  - d6e28ebed1 Add handling for FreeBSD in method zone_compare to avoid exception when /etc/localtime file does is absent. This is valid configuration on FreeBSD and represents UTC.
- **ISSUE** #41869: (mirceaulinic) Thorium: unable to execute runners (refs: #44781)
- **PR** #44781: (mirceaulinic) Correct the thorium runner @ *2017-12-01 22:55:52 UTC*
  - 8ed6287762 Merge pull request #44781 from cloudflare/thorium-fix-41869
  - 83c73a69cb Instance the Runner class instead of the RunnerClient as we're running on the Master
  - b72b7c5402 Correct the thorium runner
- **PR** #44466: (twangboy) Fix *unit.modules.test_disk* for Windows @ *2017-12-01 22:31:42 UTC*
  - 52596be102 Merge pull request #44466 from twangboy/win_fix_test_disk
  - 5615862f23 Fix some lint
  - 627d5ab0c9 Mock *salt.utils.which*
  - e5a96fe00f Skip test_fstype on Windows
- **ISSUE** #42763: (xuhcc) acme.cert state falsely reports about renewed certificate (refs: #44667)
- **PR** #44719: (rallytime) Back-port #44667 to 2017.7 @ *2017-12-01 15:20:49 UTC*
  - **PR** #44667: (oarmstrong) Fix acme.cert to run certbot non-interactively (refs: #44719)
  - b9ad4bba2d Merge pull request #44719 from rallytime/bp-44667
  - 3d85a260c4 Fix acme.cert to run certbot non-interactively
- **ISSUE** #44744: (brmzkw) roster_defaults breaks salt-ssh globbing (refs: #44747)
- **PR** #44747: (gtmanfred) use a copy so roster_defaults doesn't mangle @ *2017-12-01 15:13:48 UTC*
  - d23192c492 Merge pull request #44747 from gtmanfred/roster_defaults
  - 911411ed8f add unit test
  - eefcfc719c use a copy so roster_defaults doesn't mangle
- **ISSUE** #44694: (thuhak) state module at.absent does't work (refs: #44717)
- **PR** #44717: (garethgreenaway) [2017.7] Fixes to at module @ *2017-12-01 14:37:05 UTC*

  - 20f20ad9e1 Merge pull request #44717 from garethgreenaway/44694_at_absent_failing_to_find_jobs
  - 1f2b3c5f46 Merge branch '2017.7' into 44694_at_absent_failing_to_find_jobs
  - 3bb385b44e removing debugging logging
  - 7f0ff5a8b0 When passing IDs on the command line convert them all the strings for later comparision.
  - 99e436add4 When looking for job ids to remove based on the tag_name the comparision was comparing an INT to a STR, so the correct job id was not being returned.
- **ISSUE** #44136: (dupsatou) KeyError: 'runas' after updating to latest salt in yum repo. (refs: #44695)
- **PR** #44695: (gtmanfred) pop None for runas and runas_password @ *2017-12-01 14:35:01 UTC*
  - 6e61aa787f Merge pull request #44695 from gtmanfred/pop
  - 0efb90b6f7 Merge branch '2017.7' into pop
- **PR** #44725: (whytewolf) document note suggesting systemd-run –scope with cmd.run_bg @ *2017-11-30 19:18:06 UTC*
  - 20391c54c0 Merge pull request #44725 from whytewolf/1919_cmd.run_no_daemons
  - 4b11f8d66d add quick documentation suggesting systemd-run –scope if using cmd.run_bg with systemd
- **ISSUE** #42300: (mirceaulinic) Grains state doesn't work (fine) with proxy minions (refs: #44760)
- **ISSUE** #42074: (mirceaulinic) How to configure static grains for proxy minions (refs: #44549)
- **PR** #44760: (mirceaulinic) Fix the grains.setvals execution function when working with proxy minions @ *2017-11-30 18:27:02 UTC*
  - **PR** #44549: (mirceaulinic) Allow proxy minions to load static grains (refs: #44760)
  - 85451ae977 Merge pull request #44760 from cloudflare/px-grains-set-42300
  - 655139d01c Different path to the static grains file when running under a proxy minion
  - 3eec8dbc63 Dummy proxy: catch EOFError instead of IOError
- **ISSUE** #44583: (creideiki) Using splay in cron schedule throws exception "unsupported operand type(s) for +: 'NoneType' and 'int'" (refs: #44640)
- **PR** #44640: (vutny) Fix #44583: splay with cron-like scheduled jobs @ *2017-11-30 15:30:41 UTC*
  - 06fb80b69c Merge pull request #44640 from vutny/fix-cron-schedule-splay
  - d1f247e49e Add basic unit tests for schedule util eval func
  - 6ff8e75ac6 Fix #44583: splay with cron-like scheduled jobs
- **PR** #44712: (Ch3LL) Add pillar ssh integration tests @ *2017-11-30 15:29:33 UTC*
  - e5a1401b82 Merge pull request #44712 from Ch3LL/ssh_pillar_items
  - 97ec0e6ea0 Merge branch '2017.7' into ssh_pillar_items
  - c7f5af1274 Add pillar ssh integration tests
- **PR** #44763: (mirceaulinic) Just a small improvement to the Thorium documentation @ *2017-11-30 14:38:03 UTC*
  - 2e1c946990 Merge pull request #44763 from cloudflare/thorium-doc
  - f8d69dd0ba Add thorium_roots configuration example
  - 4610fb4e62 thorium_roots not thorium_roots_dir
- **PR** #44531: (mirceaulinic) Add deprecation notes for the NAPALM native templates @ *2017-11-30 14:18:56 UTC*

  - 8ba2df1ea0 Merge pull request #44531 from cloudflare/deprecate-napalm-tpl
  - b462776d8b Add deprecation notes for the NAPALM native templates
- **PR** #44737: (twangboy) Skip *unit.transport.test_ipc* for Windows @ *2017-11-29 19:18:21 UTC*
  - 7bde48282e Merge pull request #44737 from twangboy/win_skip_test_ipc
  - 4e0359b603 Skip IPC transport tests in Windows, not supported
- **PR** #44629: (Ch3LL) Add masterless state.highstate integration test @ *2017-11-29 19:05:23 UTC*
  - c5206113ce Merge pull request #44629 from Ch3LL/high_masterless
  - 9b7421b261 Change check to the state id
  - 9cc853e3d5 Add masterless state.highstate integration test
- **PR** #44613: (Ch3LL) Add pillar.items test for masterless @ *2017-11-29 14:43:11 UTC*
  - 2dc3e5c42a Merge pull request #44613 from Ch3LL/pillar_masterless
  - 2c2e1e2332 Merge branch '2017.7' into pillar_masterless
  - 69134e83ca Change order of local kwarg in run_call method
  - b3b5ecc6ff Add pillar.items test for masterless
- **PR** #44659: (Ch3LL) Add state.sls_id to ssh wrapper and tests @ *2017-11-29 14:41:47 UTC*
  - cc05481026 Merge pull request #44659 from Ch3LL/ssh_sls_id
  - 04b5a3dd4e Add state.sls_id to ssh wrapper and tests
- **PR** #44698: (Ch3LL) Add salt-ssh mine.get integration test @ *2017-11-28 22:15:29 UTC*
  - 642eed11e1 Merge pull request #44698 from Ch3LL/mine_ssh
  - f6a72acfe3 Merge branch '2017.7' into mine_ssh
  - 9e67babf85 Add teardown to remove ssh dir
  - f90b4f7653 Add salt-ssh mine.get integration test
- **PR** #44697: (Ch3LL) Sort the show_top results for test_state_show_top test @ *2017-11-28 20:35:41 UTC*
  - 5d82df5667 Merge pull request #44697 from Ch3LL/show_top_test
  - 974db59dc1 convert the assert to a union set instead
  - add43c4cfe Sort the show_top results for test_state_show_top test
- **PR** #44608: (Ch3LL) Add jinja to ssh sls test file @ *2017-11-27 22:00:28 UTC*
  - f2f6817e86 Merge pull request #44608 from Ch3LL/ssh_jinja
  - df669b551d Merge branch '2017.7' into ssh_jinja
  - ca97517795 Add jinja to ssh sls test file
- **ISSUE** #33957: (ghost) grains.setval doesn't setval if set in /etc/salt/minion (refs: #44663)
- **PR** #44663: (whytewolf) Update notes around grains topic, and salt.modules.grains and salt.state.grains @ *2017-11-27 21:33:38 UTC*
  - 04b97bcfad Merge pull request #44663 from whytewolf/ZD1777_ensure_understanding_of_minion_config_over_grains_fil
  - c9122e4b85 fixed pylint error, and updated description on at the top the the module and state.
  - 7fb208b5ad Update note in topics/grains to reflect that not all grains are ignored. only those set in the minion config

- **PR** #44332: (mirceaulinic) Improve the net.load_config execution function @ *2017-11-27 21:22:18 UTC*
  - 364deee6ac Merge pull request #44332 from cloudflare/improve-net-load
  - cd0bac87e6 Merge branch '2017.7' into improve-net-load
  - 6d861f9a74 Disable pylint warning
  - 3a0945ce3d Merge pull request #11 from tonybaloney/gh_44332_clone
    - 88ef9f18fc ignore lint error on import
    - 25427d845e convert key iterator to list as python 3 wont index an iterator
  - bce50154e5 Merge branch '2017.7' into improve-net-load
  - ba4a62769c Fix trailing spaces
  - 0a47a7acbf Merge pull request #10 from tonybaloney/gh_44332_clone
    - ba0280e727 linting updates
    - 78b90f3d0c add remaining tests
    - 386c4e5791 add tests for all the getters
  - f3d2d1aaaa Merge pull request #9 from tonybaloney/gh_44332_clone
    - c63222358b update tests with correct assertions and mock methods on device instance
    - b69c559c52 fix kwargs typo
  - edea76d3f3 Improve the net.load_config function
- **PR** #44664: (mvivaldi) Patch 1 @ *2017-11-27 21:17:20 UTC*
  - b6a1ed06b8 Merge pull request #44664 from mvivaldi/patch-1
  - 4551999ec7 Update jinja.py
  - ae13d57307 Update file.py
- **ISSUE** #42074: (mirceaulinic) How to configure static grains for proxy minions (refs: #44549)
- **PR** #44549: (mirceaulinic) Allow proxy minions to load static grains (refs: #44760) @ *2017-11-27 20:57:09 UTC*
  - 9ea4ee1479 Merge pull request #44549 from cloudflare/fix-proxy-grains
  - 7b03574ab6 Merge branch '2017.7' into fix-proxy-grains
  - 0320174ea4 Add doc note regarding static grains on proxy minions
  - 509d1af832 Allow proxy minions to load static grains
- **PR** #44572: (Ch3LL) Add watch_in integration test @ *2017-11-27 20:52:31 UTC*
  - 5ec7ea0bb5 Merge pull request #44572 from Ch3LL/watchin_test
  - 0a54584ddb Merge branch '2017.7' into watchin_test
  - 898c28e6d9 Merge branch '2017.7' into watchin_test
  - 3df70f3fed remove iter for watch_in failure test
  - ac437ddf90 add order check and remove iter
  - 5f2b4f434e Add watch_in integration test
    - c6733ac1ee pop None
- **PR** #44616: (Ch3LL) Add Non Base Environement salt:// source integration test @ *2017-11-22 16:13:54 UTC*

  - d6ccf4bb30 Merge pull request #44616 from Ch3LL/nonbase_test
  - 80b71652e3 Merge branch '2017.7' into nonbase_test
  - c9ba33432e Add Non Base Environement salt:// source integration test
- **PR** #44617: (Ch3LL) Add ssh thin_dir integration test @ *2017-11-22 16:12:51 UTC*
  - 3ace504c8c Merge pull request #44617 from Ch3LL/thindir_ssh
  - 071a1bd65b Merge branch '2017.7' into thindir_ssh
- **PR** #44625: (Ch3LL) Add salt-key -d integration test @ *2017-11-22 03:15:23 UTC*
  - 2cd618f99b Merge pull request #44625 from Ch3LL/delete_key_test
  - 443dc1e16b Merge branch '2017.7' into delete_key_test
- **ISSUE** #44601: (rallytime) CherryPy 12.0 removed support for "engine.timeout_monitor.on" config option (refs: #44602)
- **PR** #44614: (rallytime) [2017.7] Move PR #44602 forward to 2017.7 @ *2017-11-21 21:21:06 UTC*
  - **PR** #44602: (rallytime) Handle timeout_monitor attribute error for new versions of CherryPy (refs: #44614)
  - 4f30e845ee Merge pull request #44614 from rallytime/44602-2017.7
  - 628f015c1b Move TimoutError check lower down in exception list
  - d26d9ff5e4 Handle timeout_monitor/TimeoutError issues for new versions of CherryPy
  - 359a59dd64 Add salt-key -d integration test
  - 74ededafa7 Add ssh thin_dir integration test
    - 4d0806e28c Merge branch '2017.7' into develop2
    - 4d0d023115 Fix broken *beacons_before_connect* feature
      - 98536110d9 Merge branch '2017.7' into 2017.7
- **PR** #44571: (rallytime) Back-port #43822 to 2017.7 @ *2017-11-20 19:01:26 UTC*
  - **PR** #43822: (chnrxn) check_result: Correctly check the __extend__ state. (refs: #44571)
  - 136b9e3bc4 Merge pull request #44571 from rallytime/bp-43822
  - f81bb61f2d check_result: Correctly check the __extend__ state.
- **PR** #44588: (rallytime) Add documentation about logging before modules are loaded @ *2017-11-20 18:43:18 UTC*
  - **PR** #44576: (rallytime) Remove logging from top of napalm util file (refs: #44588)
  - **PR** #44439: (mirceaulinic) Adapt napalm modules to the new library structure (refs: #44576)
  - bea7f65291 Merge pull request #44588 from rallytime/logging-in-virtual-funcs
  - 90d1cb221d Add documentation about logging before modules are loaded
- **PR** #44513: (rallytime) Back-port #44472 to 2017.7 @ *2017-11-20 16:09:02 UTC*
  - **PR** #44472: (mephi42) nova: fix endpoint URL determination in _v3_setup() (refs: #44513)
  - a8044b73c3 Merge pull request #44513 from rallytime/bp-44472
  - 6e00e415d3 nova: fix endpoint URL determination in _v3_setup()
- **PR** #44596: (roaldnefs) Fixed Mattermost module documentation @ *2017-11-19 23:30:53 UTC*

  - f55b9daa63 Merge pull request #44596 from roaldnefs/fix-mattermost-doc
  - 549f4806ce Fixed documentation in Mattermost module
- **PR** #44528: (tkwilliams) INFRA-5978 - fix for https://github.com/saltstack/salt/issues/44290 @ *2017-11-17 17:35:44 UTC*
  - f84a2b5ab1 Merge pull request #44528 from bodhi-space/infra5978
  - ba1d57f5eb Merge branch '2017.7' into infra5978
  - 021692b6c9 INFRA-5978 - pylint / whitespace fix
  - c2210aaf7c INFRA-5978 - fix for https://github.com/saltstack/salt/issues/44290
- **PR** #44537: (Ch3LL) Add multiple salt-ssh state integration tests @ *2017-11-17 17:17:48 UTC*
  - 7f2dd0382c Merge pull request #44537 from Ch3LL/ssh_highlow
  - b98df6de24 Add known_hosts_file to salt-ssh opts_pkg in wfuncs
  - 913eedc699 Add multiple salt-ssh state integration tests
- **PR** #44576: (rallytime) Remove logging from top of napalm util file (refs: #44588) @ *2017-11-17 14:55:13 UTC*
  - **PR** #44439: (mirceaulinic) Adapt napalm modules to the new library structure (refs: #44576)
  - 1975fb41bc Merge pull request #44576 from rallytime/remove-napalm-logging
  - eb91af999e Remove logging from top of napalm util file
- **PR** #44575: (Ch3LL) Add service.running integration state test @ *2017-11-16 22:27:57 UTC*
  - c2c3048f46 Merge pull request #44575 from Ch3LL/ser_run_test
  - 7536150567 Add service.running integration state test
- **PR** #44518: (twangboy) Pass root_dir to the win_verify_env function @ *2017-11-16 20:57:49 UTC*
  - 24b1d7af31 Merge pull request #44518 from twangboy/win_fix_verify_env
  - 47114fdb30 Pass root_dirs to the win_verify_env function
    - 3385f7faf3 fix pylint
    - a2af3cb857 Include client mixin globals in scheduler for runner modules
- **PR** #44551: (mirceaulinic) Removes proxy minions false alarms and security risks @ *2017-11-16 15:09:14 UTC*
  - 1643bb7fd4 Merge pull request #44551 from cloudflare/annoying-tmpnam
  - ce1882943d Use salt.utils.files.mkstemp() instead
  - 6689bd3b2d Dont use dangerous os.tmpnam
  - 2d6176b0bc Fx2 proxy minion: clean return, like all the other modules
- **ISSUE** #30454: (favoretti) Using yaml serializer inside jinja template results in unicode being prepended by '!!python/unicode' (refs: #30481, #42064, #38554)
- **PR** #44541: (terminalmage) Fix test to reflect changes in YAML dumper @ *2017-11-15 13:23:58 UTC*
  - **PR** #42064: (The-Loeki) utils.jinja: use utils.yamldumper for safe yaml dumping (refs: #44541)
  - **PR** #38554: (multani) Fix YAML deserialization of unicode (refs: #42064)
  - **PR** #30481: (basepi) Add yaml_safe jinja filter (refs: #38554)
  - 60083ac27b Merge pull request #44541 from terminalmage/fix-yaml-test
  - 5b8f54084b Merge branch '2017.7' into fix-yaml-test

- **PR** #44538: (gtmanfred) Fix up some test kitchen stuff @ *2017-11-14 20:36:56 UTC*
    - 5c123eb551 Merge pull request #44538 from gtmanfred/kitchen
    - 3e04d2d44c use kitchen-sync for copying files
    - 9bc70fd31b back up to 2017.7.1 for kitchen tests
    - 3b93ea058b ubuntu 14 and centos 6 should not have py3 tests
    - 958e1aeb8d Fix test to reflect changes in YAML dumper
- **ISSUE** #30454: (favoretti) Using yaml serializer inside jinja template results in unicode being prepended by '!!python/unicode' (refs: #30481, #42064, #38554)
- **PR** #42064: (The-Loeki) utils.jinja: use utils.yamldumper for safe yaml dumping (refs: #44541) @ *2017-11-13 19:45:14 UTC*
    - **PR** #38554: (multani) Fix YAML deserialization of unicode (refs: #42064)
    - **PR** #30481: (basepi) Add yaml_safe jinja filter (refs: #38554)
    - 27a7b607b1 Merge pull request #42064 from The-Loeki/jinja_unicode
    - b1cf43c02d Merge branch '2017.7' into jinja_unicode
    - 8c2ac58523 Merge branch '2017.7' into jinja_unicode
    - 57dc6226a2 Merge branch '2017.7' into jinja_unicode
    - 0a8346b585 Merge branch '2017.7' into jinja_unicode
    - 393fe061b2 jinja utils: yaml import still necessary
    - 3c9130f9f0 utils.jinja: use utils.yamldumper for safe yaml dumping
- **PR** #43692: (mirceaulinic) Addressing a bug in the network find runner @ *2017-11-13 19:42:24 UTC*
    - b1f14c7518 Merge pull request #43692 from cloudflare/fix-net-runner
    - 02ffb4f38e Merge branch '2017.7' into fix-net-runner
    - 4b2f791bd2 Check if addr is short IPv6
    - 765504c137 Add all the possible keys to the result
- **ISSUE** #42393: (The-Loeki) pillarenv ignored with Salt Master pillar_cache: True (refs: #43689)
- **ISSUE** #36153: (krcroft) Pillarenv doesn't allow using separate pillar environments (refs: #43689)
- **PR** #43689: (The-Loeki) make cached pillars use pillarenv rather than saltenv @ *2017-11-13 19:30:00 UTC*
    - 1e94a5bd5f Merge pull request #43689 from The-Loeki/cached_pilarenv
    - 395c0c424d Merge branch '2017.7' into cached_pilarenv
    - 60e001733b make cached pillars use pillarenv rather than saltenv
- **PR** #43837: (twangboy) Fix *unit.states.test_archive* for Windows @ *2017-11-13 19:12:19 UTC*
    - f9b273a894 Merge pull request #43837 from twangboy/win_unit_test_archive
    - 5505a8819a Merge branch '2017.7' into win_unit_test_archive
    - b1dfe9c3c8 Format patching with statements for easier reading
    - ba2f2eb788 Add Erik's changes
    - 4ef1e3eb97 Fix *unit.states.test_archive* for Windows
- **PR** #44507: (Ch3LL) Increase sleep timeout for pillar refresh test @ *2017-11-13 18:29:06 UTC*

  - caa81728a0 Merge pull request #44507 from Ch3LL/pillar_time
  - ffa4bddcad Increase sleep timeout for pillar refresh test
- **PR** #44302: (morganwillcock) Fix traceback and incorrect message when resolving an unresolvable SID @ *2017-11-13 18:19:01 UTC*
  - cffea5ac71 Merge pull request #44302 from morganwillcock/badsid
  - f3af106e33 Merge branch 'badsid' of https://github.com/morganwillcock/salt into badsid
    - 95733fbb3b Merge branch '2017.7' into badsid
    - facc2cd16e Merge branch '2017.7' into badsid
  - c7cf5f6f70 Format pywintypes.error
  - 9572aabb67 Fix traceback and incorrect message when resolving an unresolvable SID
- **PR** #44439: (mirceaulinic) Adapt napalm modules to the new library structure (refs: #44576) @ *2017-11-13 17:43:24 UTC*
  - 32fc952000 Merge pull request #44439 from cloudflare/fix-napalm
  - f45378af04 Lint: remove extra spaces
  - c6a38258a3 Add napalm>2.0.0 note and update URLs
  - 52f73835b8 Adapt napalm modules to the new library structure
- **PR** #44457: (twangboy) Remove wmi monkeypatching @ *2017-11-13 17:38:52 UTC*
  - ebbe5949ea Merge pull request #44457 from twangboy/win_remove_wmi_monkeypatching
  - 6c872e95e6 Add back the setup_loader_modules function
  - 20273e3697 No need for setup_loader_modules since we're actually importing wmi
  - 8c107873cd Remove wmi monkeypatching
- **PR** #44490: (Ch3LL) Enable test_deploy ssh test @ *2017-11-13 17:12:48 UTC*
  - 1da1a97d7d Merge pull request #44490 from Ch3LL/ssh_ping
  - e952cd6712 Enable test_deploy ssh test
- **PR** #44491: (Ch3LL) Add salt-ssh raw integration tests @ *2017-11-13 15:47:12 UTC*
  - 18624d6798 Merge pull request #44491 from Ch3LL/ssh_raw
  - 3dc8673417 change class name to raw
  - 308596ac8d Add salt-ssh raw integration tests
- **PR** #44492: (twangboy) Fix *unit.utils.test_cloud* for Windows @ *2017-11-13 15:44:31 UTC*
  - aa17bfa8e7 Merge pull request #44492 from twangboy/win_skip_mode_check
  - 2f30ad93b1 Skips mode check in Windows
- **PR** #44484: (Ch3LL) Add orchestration tests when target exists or not @ *2017-11-10 19:24:22 UTC*
  - 5b95495e75 Merge pull request #44484 from Ch3LL/orch_test
  - f3ec6df76e Add orchestration tests when target exists or not
- **PR** #44480: (Ch3LL) Add integration pillar command line test @ *2017-11-10 19:14:31 UTC*
  - 62c42ca6fb Merge pull request #44480 from Ch3LL/override_pillar
  - 12fed1b4d8 Add integration pillar command line test

- **PR** #44317: (Ch3LL) Add state tests and state request system to salt-ssh @ *2017-11-10 18:28:43 UTC*
  - cc08ad2edc Merge pull request #44317 from Ch3LL/ssh_test
  - 46bce3bd5e add additional parser argument for ssh integration tests
  - e9231430b5 remove logic similar to cloud/proxy tests
  - c731eb8ea6 add ssh dir to test runner when –ssh-tests set
  - 8089a885c2 add wipe function to other run_ssh method
  - 200b12ae6a change versionadded salt version
  - e3ebb5e9b3 fix comment and variables
  - faef0886a7 Add state tests and state request system to salt-ssh
- **PR** #44478: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2017-11-10 18:00:56 UTC*
  - 6669035a30 Merge pull request #44478 from rallytime/merge-2017.7
  - 9fcc2a70b5 Merge branch '2016.11' into '2017.7'
    - a66cd67d15 Merge pull request #44260 from seanjnkns/issue-39901
      - ed8cccf457 #39901: Fix pylint
      - 43c81dfdee #39901: Add unit tests
      - 613d500876 Merge branch '2016.11' into issue-39901
      - b97e8046ca Utilize salt.utils.validate.net.* and _raise_error_iface
      - 6818f3631d Fixes #39901 for RH/CentOS 7
- **PR** #44444: (twangboy) LGPO: Issue with Maximum Password Age @ *2017-11-10 17:26:53 UTC*
  - 60719d0683 Merge pull request #44444 from twangboy/win_lgpo_non_zero
  - de6b394445 Remove unneeded functions
  - ee0914f7e9 Fix some lint, remove unnecessary function
  - d52a7c12db Fix typo in PasswordComplexity policy
  - 44f8f43812 Fix problem where 0 isn't 0
- **PR** #44467: (twangboy) Fix *unit.test_doc* for Windows @ *2017-11-10 15:21:58 UTC*
  - 4f3a79df07 Merge pull request #44467 from twangboy/win_fix_test_doc
  - 0a9e862bf4 Use regex to split
- **PR** #44443: (Ch3LL) Add salt-ssh grains.items test @ *2017-11-09 00:42:11 UTC*
  - ff4f13877f Merge pull request #44443 from Ch3LL/ssh_grains
  - 5d1a9af4b5 Add salt-ssh grains.items test
- **PR** #44429: (Ch3LL) Fix orch doc from pillat.get to pillar.get @ *2017-11-07 23:06:38 UTC*
  - dcdf2d4c90 Merge pull request #44429 from Ch3LL/orch_doc
  - 38ca5520f0 Fix orch doc from pillat.get to pillar.get
- **ISSUE** #42568: (clallen) Orchestration runner doesn't populate __pillar__ based on pillarenv (refs: #43817)
- **PR** #43817: (The-Loeki) Orchestrate runner forces pillarenv and saltenv to None @ *2017-11-07 06:00:16 UTC*
  - 62c4addef8 Merge pull request #43817 from The-Loeki/orch-pillarenv

  - 3fd652623c orchestrate runner: retain default envs
- **PR** #44408: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2017-11-06 15:53:00 UTC*
  - 9e4708b7b9 Merge pull request #44408 from rallytime/merge-2017.7
  - edbbd5fc2b Merge branch '2016.11' into '2017.7'
  - 5e289f42ba Merge pull request #44383 from gtmanfred/2016kitchen
    - b65f4ea4ea switch salt-jenkins over to saltstack
  - cab54e34b5 Merge pull request #44173 from twangboy/win_system_docs
    - 8e111b413d Fix some of the wording and grammer errors
    - a12bc5ae41 Use google style docstrings
  - 7aaea1d179 Merge pull request #44304 from jfindlay/cron_id
    - cc038c5bec states.cron identifier defaults to name
  - e4dbbde734 Merge pull request #44322 from rossengeorgiev/saltssh-docs-update
    - b18f2e5a6d fix program name and description for --static
    - 5b10918f02 updated CLI docs for salt-ssh
- **PR** #44358: (The-Loeki) Kubernetes client certificate file usage fix @ *2017-11-03 21:51:27 UTC*
  - b11da0d2da Merge pull request #44358 from The-Loeki/kube-client-cert-file
  - 35a8b0bb38 Kubernetes client certificate file usage fix
- **PR** #44347: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2017-11-03 21:48:21 UTC*
  - 1974e52c06 Merge pull request #44347 from rallytime/merge-2017.7
  - 9bad04b94b Merge branch '2016.11' into '2017.7'
    - 4e6f09e3eb Merge pull request #44345 from gtmanfred/2016kitchen
      - 79b8b2d0bf remove binding
    - 209847c8c2 Merge pull request #44342 from gtmanfred/2016kitchen
      - c50508f0b7 render template files platforms.yml and driver.yml
    - 1be65224cb Merge pull request #44339 from corywright/issue-44336-fix-archive-tar-docs-2016-11
      - 9c1c35a59f Remove leading dash (-) from options in archive.tar documentation
    - bebc33daf5 Merge pull request #44295 from HeinleinSupport/issue44272
      - f972715a45 fixes issue #44272
    - e7ca9f8407 Merge pull request #44286 from gtmanfred/2016.11
      - 193e715e37 use our git repo for kitchen-salt
- **PR** #44364: (Ch3LL) Include disk size check for test_spm_build_big_file test @ *2017-11-01 13:57:24 UTC*
  - aea9f4a115 Merge pull request #44364 from Ch3LL/fix_size_test
  - 952c6bfea4 Include file size check for test_spm_build_big_file test
- **ISSUE** #44239: (boltronics) --progress fails when hosts routed via syndic (refs: #44273)
- **PR** #44273: (DmitryKuzmenko) Workaround progressbar failure if minion is behind syndic. @ *2017-10-31 17:07:17 UTC*

  - 609de9367a Merge pull request #44273 from DSRCorporation/bugs/44239_syndic_progress
  - e1a7605623 Workaround progressbar failure if minion is behind syndic.
- **PR** #44350: (gtmanfred) update salt-jenkins repo to 2017.7 @ *2017-10-30 21:31:30 UTC*
  - eef6dbfa58 Merge pull request #44350 from gtmanfred/2017.7
  - cf71e3d9f2 update salt-jenkins repo to 2017.7
- **PR** #44346: (gtmanfred) remove binding from erb template rendering (2017.7) @ *2017-10-30 20:57:19 UTC*
  - d586b3bf97 Merge pull request #44346 from gtmanfred/2017.7
  - bf577c3d8b remove binding
- **PR** #44343: (gtmanfred) render template files platforms.yml and driver.yml (2017.7) @ *2017-10-30 20:04:22 UTC*
  - 547aac6658 Merge pull request #44343 from gtmanfred/2017.7
  - ec24fbc0c2 render template files platforms.yml and driver.yml
- **ISSUE** #44336: (corywright) Docs for archive.tar should not use leading dash for tar options (refs: #44339, #44338)
- **PR** #44338: (corywright) Remove leading dash from options in archive.tar docs (2017.7 and develop) @ *2017-10-30 18:59:33 UTC*
  - 6e2a74c18b Merge pull request #44338 from corywright/issue-44336-fix-archive-tar-docs-2017-7-and-newer
  - 49b0abc284 Remove leading dash (-) from options in archive.tar documentation
- **PR** #44265: (Ch3LL) Add service.status integration test @ *2017-10-30 15:00:12 UTC*
  - 71923bed97 Merge pull request #44265 from Ch3LL/service_test
  - 716aabc0bf Merge branch '2017.7' into service_test
  - dd5c823210 remove skipIf import
  - ff92f31cbe remove skipif for docker
  - c13f37eee4 change service name depending on os
  - 980c43ebc9 change skip message check to docker
  - 3955537609 change skip if check to docker
  - aa8875a0e2 change service name to docker
  - 654071028b change service to crond
  - 7911b4b3eb Add service.status integration test
- **PR** #44294: (nasenbaer13) Boto asg fixes, Backport of #43858 @ *2017-10-30 14:48:52 UTC*
  - **PR** #43858: (nasenbaer13) Boto_ASG fixes for scaling policy rate limiting and tag conversion (refs: #44294)
  - 8ae9769bfb Merge pull request #44294 from eyj/boto_asg
  - f5ad6aeb70 Debug log added when throttled by API
  - c05d9aeced Encode tags as utf-8, retry policy readout
- **PR** #44312: (rallytime) Back-port #44287 to 2017.7 @ *2017-10-30 14:25:56 UTC*
  - **PR** #44287: (jf) Fix utils.files.guess_archive_type to recognize the "tbz" extension as well (refs: #44312)

  - 68a9bebf90 Merge pull request #44312 from rallytime/bp-44287
  - 4d02e61f97 Merge branch '2017.7' into bp-44287
  - ba0eaae95e Fix utils.files.guess_archive_type to recognize the "tbz" extension as well (also tidy up list of extensions)
- **ISSUE** #44258: (oarmstrong) docker_container.running recreates containers with multiple links (refs: #44262)
- **PR** #44311: (rallytime) Back-port #44262 to 2017.7 @ *2017-10-30 14:25:35 UTC*
  - **PR** #44262: (oarmstrong) docker_container.running sort list of links (refs: #44311)
  - b8854e27c0 Merge pull request #44311 from rallytime/bp-44262
  - 72d617cfbe Merge branch '2017.7' into bp-44262
  - ae34a15503 docker_container.running sort list of links
- **PR** #44314: (gtmanfred) update .kitchen.yml to run py3 tests too @ *2017-10-30 14:23:15 UTC*
  - 48df79ef77 Merge pull request #44314 from gtmanfred/2017.7
  - 54265769c4 Merge branch '2017.7' into 2017.7
- **PR** #44316: (rallytime) Fix lint failure on 2017.7 branch @ *2017-10-27 18:36:08 UTC*
  - dbc5e224e9 Merge pull request #44316 from rallytime/fix-lint
  - 6d2490f6a0 Fix lint failure on 2017.7 branch
  - 39262b625e update .kitchen.yml to run py3 tests too
- **PR** #44279: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2017-10-27 16:17:19 UTC*
  - b2b0c770a4 Merge pull request #44279 from rallytime/merge-2017.7
  - 8237f45a46 Add print_function to __future__ import list
  - 055b0701de Lint fix from sloppy merge conflict resolution
  - 1c3cb5c6a4 Merge branch '2016.11' into '2017.7'
    - 8a1ea165af Merge pull request #44259 from gtmanfred/2016.11
      - 56a3ad8f68 fix pylint comments
      - 4add666db1 add comment to Gemfile and move copyartifacts
      - b4c8f7eb57 fix pylint
      - 392fd4f837 try newest salttesting
      - 79251287d0 add logging
      - 38963d5a82 use transport if not set in state_file
      - 10e309a64f which vagrant should go to stderr
      - 9307564de0 fix output columns
      - 2da22f87e1 test opennebula
      - 9f38f16905 add opennebula to Gemfile
      - 7465f9b27a add script for copying back artifacts
      - 255118cfd7 run tests with kitchen
    - 9d6bc8509b Merge pull request #44268 from twangboy/win_fix_lgpo_typo

        - a6a4c10a77 Fix typo
    - 0beb65a283 Merge pull request #44269 from terminalmage/fix-log-message
        - bc9cd65496 Fix log message in salt.utils.gitfs
    - 304dd2529d Merge pull request #44160 from gtmanfred/directory
        - a7d3d668f4 missed removing changes in the next test
        - ac0b5ec440 fix test
        - d3d00c3e62 add changes to test return
    - e10395483d Merge pull request #44205 from rallytime/bp-44177
        - b9940f8521 Fixing default redis.host in documentation
- **PR** #44291: (Ch3LL) add saltutil.refresh_pillar test @ *2017-10-27 15:19:43 UTC*
    - bd5b9dd0aa Merge pull request #44291 from Ch3LL/pillar_test
    - 34e2955445 add saltutil.refresh_pillar test
- **PR** #44267: (twangboy) Fix type and Py3 issues in LGPO module @ *2017-10-27 14:27:50 UTC*
    - ba17a1c4d0 Merge pull request #44267 from twangboy/win_fix_lgpo
    - 5d22d34cac Use unicode_literals
    - 40636397d8 Fix set for Py3
    - 8f8c706426 Fix typo
- **PR** #44285: (Ch3LL) add spm integration tests for remove and build @ *2017-10-26 21:20:10 UTC*
    - e16707c403 Merge pull request #44285 from Ch3LL/all_spm
    - 1f77f3e6a3 add skipif logic for fallocate cmd
    - 03b5c4bc6d add spm integration tests for remove and build
- **PR** #44301: (twangboy) Fix test_pydsl on Windows @ *2017-10-26 21:14:21 UTC*
    - 6392896a22 Merge pull request #44301 from twangboy/win_fix_test_pydsl
    - 6db23757bc Fix test_pydsl on Windows
- **PR** #44293: (UtahDave) Fix documentation grammar and spelling errors @ *2017-10-26 13:05:31 UTC*
    - 8787d02688 Merge pull request #44293 from UtahDave/fix_unittest_docs
    - c919648ab4 Fix documentation grammar and spelling errors
- **PR** #44248: (Ch3LL) SPM tests: use _spm_build_files method during test_build setup @ *2017-10-25 19:45:03 UTC*
    - 6e33743c1a Merge pull request #44248 from Ch3LL/spm_create_repo
    - 0a387c2ecd fix pylint
    - f383f05a93 Add SPM create_repo integration test
- **PR** #44253: (Ch3LL) Add multiple spm integration tests @ *2017-10-25 13:36:03 UTC*
    - bd75be24ca Merge pull request #44253 from Ch3LL/spm_install
    - 9e2e785034 add spm tests to test runner
    - 4729ccd32b Add multiple spm integration tests

- **PR** #44254: (twangboy) Fix *unit.modules.test_win_groupadd* for Windows @ *2017-10-25 13:33:40 UTC*
    - 75ee1ebc50 Merge pull request #44254 from twangboy/win_fix_test_win_groupadd
    - 609361bf48 Fix some lint errors
    - 1f44d8d5e6 Document helper functions
    - b0caec320e Move _get_all_groups up to the top
    - 7a3ff9387d Mock the rest of the tests
    - 5ce14df82c Change how members are retrieved in win_groupadd
    - 6ab82394be Set up mocking
- **PR** #44266: (Ch3LL) Add state, grains and service proxy tests @ *2017-10-25 13:08:50 UTC*
    - 4c23fa63bb Merge pull request #44266 from Ch3LL/proxy_tests
    - e5701b472d Add state, grains and service proxy tests
- **ISSUE** #43187: (mirceaulinic) How to point from an execution module that a certain function failed (refs: #44244)
- **PR** #44244: (mirceaulinic) Add explicit non-zero retcode to napalm config functions @ *2017-10-24 09:23:40 UTC*
    - c849f350ba Merge pull request #44244 from cloudflare/add-retcode
    - a1f27c9f00 Add explicit non-zero retcode to napalm config functions
- **ISSUE** #44227: (rklaren) salt-cloud leaves a broken vm around when the salt bootstrap fails (refs: #44228)
- **PR** #44228: (rklaren) Fixes #44227, make salt-cloud/libvirt cleanup after errors more robust @ *2017-10-23 17:09:35 UTC*
    - 195b225540 Merge pull request #44228 from rklaren/fix-salt-cloud-libvirt-cleanup-after-errors
    - 7917d1e61e Incorporate review comments.
    - 3a10b6aef1 Fixes #44227, make salt-cloud/libvirt cleanup after errors more robust
- **ISSUE** #19532: (stolendog) salt-ssh running git clone with not root user (refs: #43769)
- **ISSUE** #10582: (mtorromeo) Git ssh helper may be unable run (refs: #43769)
- **PR** #44008: (mtorromeo) Backport #43769 to 2017.7 @ *2017-10-23 14:19:57 UTC*
    - **PR** #43769: (mtorromeo) Copy git ssh-id-wrapper to /tmp only if necessary (Fixes #10582, #19532) (refs: #44008)
    - 01e7bab990 Merge pull request #44008 from mtorromeo/git-noexec-fix
    - a7a841d9d2 Merge branch '2017.7' into git-noexec-fix
    - d177240cfc Merge branch '2017.7' into git-noexec-fix
    - a63e6ca963 Copy git ssh-id-wrapper to /tmp only if necessary (Fixes #10582, Fixes #19532)
- **PR** #44202: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2017-10-23 14:18:30 UTC*
    - 85c0ef493f Merge pull request #44202 from rallytime/merge-2017.7
    - 99ff7a5c12 Merge branch '2016.11' into '2017.7'
        - 09ddfd0c08 Merge pull request #44167 from garethgreenaway/44140_debian_ip_fixes
            - 5f7555846f When looping through the various pre, post, up and down commands put them into the interface dict using the right internet family variable.

        - 9f9e936b52 Merge pull request #43830 from rallytime/bp-43644
            - 12845ae802 Several fixes for RDS DB parameter group management
        - 07db6a3d8b Merge pull request #43994 from oeuftete/fix-manage-runner-presence
            - f3980d7d83 Fix manage.present to show lost minions
        - a07537e258 Merge pull request #44188 from terminalmage/issue44150
            - 0692f442db yumpkg: Check pkgname instead of name to see if it is a kernel pkg
        - 715edc0cea Merge pull request #44158 from rallytime/bp-44089
            - 534faf0b7a Catch on empty Virtualbox network addr #43427

- **PR** #44208: (twangboy) Fix some lint in PR: 44080 @ *2017-10-20 16:42:02 UTC*
    - d7dc2bd0e8 Merge pull request #44208 from twangboy/win_fix_group.present
    - 61e2e9ccda Fix some lint
- **PR** #43843: (twangboy) Fix *unit.states.test_mount* for Windows @ *2017-10-20 14:27:25 UTC*
    - c6d27ada51 Merge pull request #43843 from twangboy/win_unit_test_mount
    - a862e0bf2d Remove unneeded import
    - d78f27466d Fix *unit.states.test_mount* for Windows
- **PR** #44111: (anlutro) Try to correctly parse debian codename from /etc/os-release @ *2017-10-19 22:23:26 UTC*
    - 372820ea38 Merge pull request #44111 from alprs/fix-deb8-py3-oscodename
    - 1e1e5a3ff6 try to correctly parse debian codename from /etc/os-release
- **PR** #44187: (twangboy) Fix pickling errors on Windows @ *2017-10-19 20:36:51 UTC*
    - 75136152c1 Merge pull request #44187 from twangboy/win_fix_unit_test_daemons.py
    - 64d2e4f732 Fix pickling errors on Windows
- **ISSUE** #44181: (jonans) Scheduler with multiple when values doesn't run (refs: #44186)
- **PR** #44186: (garethgreenaway) [2017.7] scheduler fixes @ *2017-10-19 20:36:04 UTC*
    - 7a89cd8697 Merge pull request #44186 from garethgreenaway/44181_scheduler_multiple_whens
    - 7eef3b3571 Adding a copy.deepcopy to the for loop that looks for old jobs to avoid stale jobs ending up in the list.
- **PR** #43896: (twangboy) Fix win_lgpo execution module @ *2017-10-19 20:13:18 UTC*
    - 1d16ae8ba7 Merge pull request #43896 from twangboy/win_fix_lgpo_scom
    - 648d1b8d99 Catch CommandExecutionError
    - 0040082d0a Fix pylint error
    - 91258cd6a8 Fix typo
    - 261dba347d Put the file.remove in a try/except/else block
    - 020c2a2b85 Fix syntax error
    - d5bec99126 Fix some lint
    - b96186d60d Fix INSTALL_LANGUAGE
    - 5471bd521f Fix problem with file handle

  - 5ec58c6200 Use System Install Language as default fallback
  - f9ad446019 Fix win_lgpo execution module
- **PR** #44080: (twangboy) Fix a regression in *group.present* in Windows @ *2017-10-19 20:10:44 UTC*
  - 98356b86af Merge pull request #44080 from twangboy/win_fix_group.present
  - 29bc80ff87 Improve get_sam_name
  - ef759a3875 Fix example in function docs for get_sam_name
  - 43740c5fed Document 15 character limit
  - 83f36cc2ef Account for 15 character limit in hostname
  - aa278966de Remove *args, pass gid as a keyword
  - 5230ecd7e1 Accept *args
- **PR** #44171: (Ch3LL) Add SPM Build Integration Tests @ *2017-10-19 19:49:14 UTC*
  - 5ef124bf2d Merge pull request #44171 from Ch3LL/spm_int
  - cd79e9444e remove unneded kwarg
  - 1541376c4f Add spm build test
- **PR** #44157: (benediktwerner) Added 'versionadded' tags to sensehat modules @ *2017-10-19 14:13:31 UTC*
  - 34a843252d Merge pull request #44157 from benediktwerner/2017.7
  - bd825b51cc Changed sensehat versionadded from 2017.7 to 2017.7.0
  - f1d3c5bbcf Added 'versionadded' tags to sensehat modules
- **PR** #44164: (terminalmage) Fix examples in docker_container.{stopped,absent} docstrings @ *2017-10-19 14:12:37 UTC*
  - 1427c72e1e Merge pull request #44164 from terminalmage/fix-docker-docstring
  - 7b46489e33 Fix examples in docker_container.{stopped,absent} docstrings
- **PR** #44168: (twangboy) Fix *unit.test_auth* for Windows @ *2017-10-19 14:12:22 UTC*
  - 77969c4161 Merge pull request #44168 from twangboy/win_skip_pam_eath
  - bb1d2eb85b Skip tests that are failing on PAM eauth
- **PR** #44151: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2017-10-18 16:52:30 UTC*
  - 88a776d9d2 Merge pull request #44151 from rallytime/merge-2017.7
  - 6aa8f03a4a Merge branch '2016.11' into '2017.7'
    - 0cd493b691 Merge pull request #44131 from rallytime/bp-44029
      - bebf301976 fixed test addressing issue #43307, disk.format_ to disk.format
      - b4ba7ae2fc addresses issue #43307, disk.format_ to disk.format
    - 3a68e356f8 Merge pull request #44093 from gtmanfred/fix-44087
      - 5455c5053b fix pylint
      - f749cafa25 don't filter if return is not a dict
    - c785d7a847 Merge pull request #44122 from cachedout/gpg_pr_template
      - e41e3d76be Typo fix

            - 37c7980880 Add note about GPG signing to PR template
        - bf90ea1f51 Merge pull request #44124 from rallytime/merge-2016.11
            - 59861291c8 Merge branch '2016.11.8' into '2016.11'
            - 57623e2abe Merge pull request #44028 from rallytime/bp-44011
            - 89e084bda3 Do not allow IDs with null bytes in decoded payloads
            - 206ae23f15 Don't allow path separators in minion ID
        - 13f3ffa83a Merge pull request #44097 from gtmanfred/openneb
            - c29655b2c2 Merge branch '2016.11' into openneb
            - bd2490b149 OpenNebula does not require the template_id to be specified
        - ac3e4df964 Merge pull request #44110 from roaldnefs/fix-doc-local-returner
            - efd58f7594 Merge branch '2016.11' into fix-doc-local-returner
            - 881f1822f2 Format fix code example local returner doc

- **ISSUE** #43918: (mwerickso) subset argument does not work with saltmod.state (refs: #43933)
- **PR** #43933: (gtmanfred) if expect_minions is passed use that instead @ *2017-10-18 16:43:39 UTC*
    - 0b47eb7242 Merge pull request #43933 from gtmanfred/2017.7
    - 272dcc6ba5 add inline comment about popping expect_minions
    - b615ce1762 if expect_minions is passed use that instead
- **PR** #44081: (skizunov) Windows: Fix usage of pkgrepo state @ *2017-10-18 16:16:46 UTC*
    - 36da1a7fac Merge pull request #44081 from skizunov/develop3
    - 351d16840b Move strip_uri to salt/utils/pkg/deb.py
    - f54c7a6f01 Windows: Fix usage of pkgrepo state
- **PR** #43913: (twangboy) Fix *unit.templates.test_jinja* for Windows @ *2017-10-17 21:09:05 UTC*
    - afcaa0c591 Merge pull request #43913 from twangboy/win_fix_test_jinja
    - a4e2d8059d Fix *unit.templates.test_jinja* for Windows
- **PR** #43917: (twangboy) Fix *unit.test_pillar* for Windows @ *2017-10-17 21:06:46 UTC*
    - fc5754c6a1 Merge pull request #43917 from twangboy/win_unit_test_pillar
    - 00dbba5712 Fix *unit.test_pillar* for Windows
- **PR** #44133: (cachedout) Fix typos in parallel states docs @ *2017-10-17 15:24:19 UTC*
    - 6252f82f58 Merge pull request #44133 from cachedout/fix_paralell_docs
    - 8d1c1e21f0 Fix typos in paralell states docs
- **PR** #44135: (timfreund) Insert missing verb in gitfs walkthrough @ *2017-10-17 14:32:13 UTC*
    - 0d3f5db867 Merge pull request #44135 from timfreund/insert_missing_verb
    - 9557504b75 Insert missing verb in gitfs walkthrough
- **PR** #44055: (nasenbaer13) Activate jid_queue also for SingleMinions to workaround (Backport) @ *2017-10-16 20:14:52 UTC*
    - **PR** #43860: (nasenbaer13) Activate jid_queue also for SingleMinions (occurs on reconnect) (refs: #44055)

  - a9700f6061 Merge pull request #44055 from eyj/jid_queue
  - 4bdd5bbf6b Merge branch '2017.7' into jid_queue
  - facef2227d Merge branch '2017.7' into jid_queue
  - 2fedcec6bb Merge branch '2017.7' into jid_queue
  - 255aa94c64 Activate jid_queue also for SingleMinions to workaround 0mq reconnection issues
- **PR** #44125: (rallytime) [2017.7] Merge forward from 2017.7.2 to 2017.7 @ *2017-10-16 20:02:25 UTC*
  - 2fba45cd3f Merge pull request #44125 from rallytime/merge-2017.7
  - c4ae4a6b50 Merge branch '2017.7.2' into '2017.7'
    - 5d719a2219 Merge pull request #44027 from rallytime/bp-44012
    - f7824e41f3 Don't allow path separators in minion ID
    - 44060dc9c1 Do not allow IDs with null bytes in decoded payloads
- **ISSUE** #43307: (marek-knappe) Filesystem creation is failing on newly created LV (refs: #44029)
- **PR** #44029: (msummers42) addresses issue #43307, disk.format_ to disk.format (refs: #44131) @ *2017-10-16 19:59:20 UTC*
  - 68974aa74d Merge pull request #44029 from msummers42/2017.7
  - 16e1c1dfc8 fixed test addressing issue #43307, disk.format_ to disk.format
  - 3d597db51c Merge branch '2017.7' into 2017.7
  - 18fb0be96a addresses issue #43307, disk.format_ to disk.format
- **PR** #44079: (skizunov) opkg: Fix usage with pkgrepo.managed @ *2017-10-16 19:58:13 UTC*
  - d0bbe65ffa Merge pull request #44079 from skizunov/develop2
  - 0614d1af30 Merge branch '2017.7' into develop2
  - b6b12fe495 opkg: Fix usage with pkgrepo.managed
- **PR** #44090: (pratik705) Fix create_attach_volumes salt-cloud action for gcp @ *2017-10-16 19:04:22 UTC*
  - 22a8253595 Merge pull request #44090 from pratik705/fix-create_attach_volumes_salt-cloud_action-GCP
  - 3eefd334c5 Fixed "create_attach_volumes" salt-cloud action for GCP
- **PR** #44121: (benediktwerner) Fixed code snippet in unit testing documentation @ *2017-10-16 18:28:36 UTC*
  - 888e5f51a2 Merge pull request #44121 from benediktwerner/2017.7
  - 1319c822bd Fixed code snippet in unit testing doc
- **PR** #44098: (twangboy) Return multiprocessing queue in LogSetupMock class @ *2017-10-16 18:14:30 UTC*
  - 9fe94d7843 Merge pull request #44098 from twangboy/win_mock_test_parsers
  - cc43ca27af Return multiprocessing queue in LogSetupMock class
- **PR** #44118: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2017-10-16 17:01:38 UTC*
  - 0ee04eaf1d Merge pull request #44118 from rallytime/merge-2017.7
  - bbec47afbc Merge branch '2016.11' into '2017.7'
    - c960ca32c2 Merge pull request #44092 from techhat/awsunicode
      - bbd9db4d00 One more encoding

      - 0e8b325667 Apparently __salt_system_encoding__ is a thing
      - 1e7211838d Use system encoding
      - 1af21bbe5e Made sure that unicoded data is sent to sha256()
    - d89c317d96 Merge pull request #44021 from whiteinge/cpstats-attribute-error
      - bf14e5f578 Also catch cpstats AttributeError for bad CherryPy release ~5.6.0
    - bbdabe242a Merge pull request #44025 from dayid/lover_typo
      - 385980c21a Merge branch '2016.11' of https://github.com/saltstack/salt into lover_typo
      - 266dc00a23 Typo correction of lover to lower
    - d8f3891a5e Merge pull request #44030 from rallytime/merge-2016.11
      - 53eaf0d75c Merge branch '2016.3' into '2016.11'
      - 64fd839377 Merge pull request #44010 from Ch3LL/2016.3.7_follow_up
      - 9a00302cd8 fix 2016.3.7 release notes merge conflict
      - 63da1214db Do not allow IDs with null bytes in decoded payloads
      - ee792581fc Don't allow path separators in minion ID
      - 8aab65c718 fix 2016.3.7 release notes merge conflict
      - bd73dcb02c Merge pull request #43977 from Ch3LL/3.8_sec
      - 5fb3f5f6b1 Add Security Notes to 2016.3.8 Release Notes

- **PR** #44099: (twangboy) Skip Master, Minion, and Syndic parser tests @ *2017-10-16 16:07:00 UTC*
  - 28fa097b9b Merge pull request #44099 from twangboy/win_skip_test_parsers
  - caf086c05a Skip Master, Minion, and Syndic parser tests
- **PR** #44106: (roaldnefs) Fix mattermost returner documentation @ *2017-10-16 13:12:23 UTC*
  - dbf112ead7 Merge pull request #44106 from roaldnefs/fix-doc-mattermost_returner
  - b3761a0401 Fix doc indentation in mattermost_returner
- **PR** #44054: (nasenbaer13) Backport of missing delete_on_termination @ *2017-10-13 15:45:25 UTC*
  - **PR** #43859: (nasenbaer13) Add missing delete_on_termination passthrough. Adapt docs. (refs: #44054)
  - fd2c51b76c Merge pull request #44054 from eyj/boto_lc
  - 34d4629a64 Merge branch '2017.7' into boto_lc
  - 9efd63526a Adapted documentation of delete_on_termination parameter
  - eb2bfd047b Add missing delete_on_termination passthrough. Adapt docs.
- **PR** #44076: (Ch3LL) Add spm shell tests @ *2017-10-13 14:32:19 UTC*
  - b61ed96268 Merge pull request #44076 from Ch3LL/spm_test
  - d2e91c33bd Add spm shell tests
- **PR** #44051: (twangboy) Fix some documentation formatting issues in the win_dacl state @ *2017-10-12 15:40:17 UTC*
  - e38f313ac0 Merge pull request #44051 from twangboy/win_fix_docs_dacl
  - 377d6b6171 Fix some docs in the win_dacl state module

- **PR** #44066: (Ch3LL) Add Known CherryPy Issue to 2017.7.2 Release Notes @ *2017-10-12 15:18:25 UTC*
    - a85837d72b Merge pull request #44066 from Ch3LL/cherry_release
    - 8e597fcce9 Add Known CherryPy Issue to 2017.7.2 Release Notes
- **ISSUE** #43643: (doublez13) salt-ssh: multiple targets fails after upgrade to 2017.7 (refs: #43889)
- **ISSUE** #43449: (ecgg) salt-ssh -L with hosts down or unreachable returns wrong results (refs: #43889)
- **PR** #43889: (CorvinM) Fix issue with using roster_defaults with flat or cloud rosters. @ *2017-10-11 23:22:11 UTC*
    - fcab77ac7b Merge pull request #43889 from CorvinM/issue43449
    - fefd28d896 Add futureproofing to roster_defaults to support roster dictionary options
    - aebe76b6f8 Fix issue with using roster_defaults with flat or cloud rosters. fixes #43449 fixes #43643
- **PR** #44031: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2017-10-11 22:03:31 UTC*
    - 3ad1c6d1d9 Merge pull request #44031 from rallytime/merge-2017.7
    - 1d4a6c3949 Lint: Fixup undefined variable errors
    - 788ad0609a Merge branch '2016.11' into '2017.7'
        - 0dbf41e79e Merge pull request #44011 from Ch3LL/2016.11.7_follow_up
            - c0149101c0 Do not allow IDs with null bytes in decoded payloads
            - 19481423dd Don't allow path separators in minion ID
        - d61300df20 Merge pull request #44023 from Ch3LL/11.9rn
            - 7f9015eb41 Add 2016.11.9 Release Note File
        - 9ff53bf63a Merge pull request #44019 from benediktwerner/2016.11
            - bc53598027 Fixed spelling mistake in salt_bootstrap tutorial
            - 6c30344824 Added missing tutorial docs to the tutorial index
        - 364523f5f8 Merge pull request #43955 from meaksh/2016.11-fix-2291
            - a81b78381b Merge branch '2016.11' into 2016.11-fix-2291
            - 44bc91bb98 Enable '–with-salt-version' parameter for setup.py script
        - fec714b91d Merge pull request #43962 from bobrik/kmod-built-in
            - 95ab901553 Report built-in modiles in kmod.available, fixes #43945
        - e434c39c4e Merge pull request #43960 from cro/ldap_nopw_bind2
            - 962a20cf4b Require that bindpw be non-empty if auth.ldap.anonymous=False
            - 9df3d91d8f Release notes blurb for change to bindpw requirements
        - e9dfda2177 Merge pull request #43991 from Ch3LL/3.8_sec_2
            - 1977df8462 Add Security Notes to 2016.3.8 Release Notes
        - 2346d2691e Merge pull request #43968 from rossengeorgiev/fix-zenoss-prod_state
            - e6d31c1ea6 fix zenoss state module not respecting test=true
        - 8d56a5ac45 Merge pull request #43776 from Ch3LL/2016.11.8_docs
            - f72bc00000 [2016.11] Bump latest and previous versions

    - 21bf71c3f5 Merge pull request #43976 from Ch3LL/11.8_sec
      - f0c3184288 Add Security Notes to 2016.11.8 Release Notes
    - 1d5397ab5b Merge pull request #43973 from terminalmage/fix-grains.has_value
      - bf45ae6e6a Fix grains.has_value when value is False
    - 9ac3f2ea7b Merge pull request #43888 from rallytime/bp-43841
      - 87d676f08a add -n with netstat so we don't resolve
    - f880ac4c08 Merge pull request #43916 from dereckson/fix-typo-cloud-scaleway
      - 15b8b8a9f4 Fix typo in salt-cloud scaleway documentation
- **PR** #44045: (isbm) Bugfix: always return a string "list" on unknown job target type. @ *2017-10-11 21:58:12 UTC*
  - 5db1e8c6ca Merge pull request #44045 from isbm/isbm-tgttype-fix-2017-port
  - 471ff35c2f Bugfix: always return a string "list" on unknown job target type.
- **ISSUE** #43949: (arthurlogilab) [logger] [sentry] KeyError: 'SENTRY_PROJECT' (refs: #43950)
- **PR** #44026: (rallytime) Back-port #43950 to 2017.7 @ *2017-10-11 15:27:49 UTC*
  - **PR** #43950: (arthurlogilab) [log/sentry] avoid KeyError: 'SENTRY_PROJECT' (refs: #44026)
  - 6c8f7fd5ec Merge pull request #44026 from rallytime/bp-43950
  - a37e0bad62 [log/sentry] avoid KeyError: 'SENTRY_PROJECT'
- **PR** #44012: (Ch3LL) Security Fixes for 2017.7.2 (refs: #44027) @ *2017-10-10 20:04:08 UTC*
  - 369ee8a132 Merge pull request #44012 from Ch3LL/2017.7.1_follow_up
  - 92e05cf1c0 Don't allow path separators in minion ID
  - 70133aa305 Do not allow IDs with null bytes in decoded payloads
- **PR** #44024: (Ch3LL) Add 2017.7.3 Release Note File @ *2017-10-10 20:03:12 UTC*
  - 4fe029a0ab Merge pull request #44024 from Ch3LL/7.3rn
  - 027f509368 Add 2017.7.3 Release Note File
- **ISSUE** #43997: (unthought) gce cloud provider breaks for make_master: True (refs: #43998)
- **PR** #43998: (unthought) Fix gce make_master @ *2017-10-10 20:01:25 UTC*
  - e484d16817 Merge pull request #43998 from unthought/fix-gce-make_master
  - 6e9f0fa24e Fix GCE provider: #create returns bootstrap result
- **ISSUE** #44013: (DenisBY) pkgrepo.managed broken in 2017.7.2 (refs: #44016)
- **PR** #44016: (terminalmage) Fix on_header callback when not redirecting and no Content-Type present @ *2017-10-10 19:59:24 UTC*
  - 82b92d54b3 Merge pull request #44016 from terminalmage/issue44013
  - d594b95f92 No need to set a specific encoding if one hasn't been provided via the headers
  - 425ede4b84 Fix on_header callback when not redirecting and no Content-Type present
- **PR** #43952: (t0fik) add requisites to stateconf (backport #43920) @ *2017-10-10 13:03:31 UTC*
  - **PR** #43920: (t0fik) Added missing requisites to stateconf renderer (refs: #43952)
  - bd879eb66e Merge pull request #43952 from jdsieci/2017.7_add_requisites_to_stateconf

  - 9994c64670 Merge branch '2017.7' into 2017.7_add_requisites_to_stateconf
- **PR** #43777: (Ch3LL) [2017.7] Bump latest and previous versions @ *2017-10-09 17:21:57 UTC*
  - a4358dfa36 Merge pull request #43777 from Ch3LL/2017.7.2_docs
  - 410c624f7a [2017.7] Bump latest and previous versions
- **PR** #43978: (Ch3LL) Add Security Notes to 2017.7.2 Release Notes @ *2017-10-09 17:20:04 UTC*
  - 2a064c1a72 Merge pull request #43978 from Ch3LL/7.2_sec
  - 57fd6f7bcb Add Security Notes to 2017.7.2 Release Notes
- **PR** #43932: (techhat) Don't try to modify dict while looping through it @ *2017-10-06 21:20:54 UTC*
  - d9530e3c52 Merge pull request #43932 from techhat/moddict
  - 4a77560646 Don't try to modify dict while looping through it
- **PR** #43956: (terminalmage) Fix fileclient's get_url when redirecting to a redirect @ *2017-10-06 21:19:41 UTC*
  - 39893a1dab Merge pull request #43956 from terminalmage/fix-get_url-redirects
  - 9a4f6a260f Fix fileclient's get_url when redirecting to a redirect
- **PR** #43943: (twangboy) Fix *unit.utils.test_utils* for Windows @ *2017-10-06 19:35:24 UTC*
  - 1baf286719 Merge pull request #43943 from twangboy/win_unit_test_utils
  - 254dac7723 Fix *unit.utils.test_utils* for Windows
    - 89200ff28e rebase from 2017.7.2
- **PR** #43939: (terminalmage) Fix typo in log message @ *2017-10-05 23:20:04 UTC*
  - a8f1750323 Merge pull request #43939 from terminalmage/fix-typo
  - 29d8cf4f26 Fix typo in log message
- **ISSUE** #43909: (frogunder) state.highstate not working on py3 setup (refs: #43910)
- **ISSUE** #43605: (cruscio) Module.Run: Passed invalid arguments to state.apply: can't serialize dict_keys(['task.create_task']) (refs: #43910)
- **PR** #43910: (terminalmage) Don't put unserializable dict.keys() into state return @ *2017-10-05 20:33:47 UTC*
  - 1a718eb1ed Merge pull request #43910 from terminalmage/issue43605
  - 042e092ac8 Don't put unserializable dict.keys() into state return
- **ISSUE** #41894: (DR3EVR8u8c) Salt-cloud can't resize root volume with public ami images (refs: #43907)
- **ISSUE** #39257: (aig787) Using del_root_vol_on_destroy option in salt-cloud gives IndexError (refs: #43907)
- **PR** #43927: (rallytime) Back-port #43907 to 2017.7 @ *2017-10-05 20:10:16 UTC*
  - **PR** #43907: (richardsimko) Make sure EBS volume exists before querying (refs: #43927)
  - **PR** #33115: (rbjorklin) Fix override of ec2 volumetype (refs: #43907)
  - a7a59868c8 Merge pull request #43927 from rallytime/bp-43907
  - f62e8ca87f Make sure volume exists before querying
- **PR** #43934: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2017-10-05 20:07:36 UTC*
  - 4fcd4709ea Merge pull request #43934 from rallytime/merge-2017.7
  - eaca3291e2 Merge branch '2016.11' into '2017.7'

    - 2ab7549d48 Merge pull request #43884 from UtahDave/2016.11local
      - e3b2857285 Merge branch '2016.11' into 2016.11local
    - 4b882d4272 Merge pull request #43869 from terminalmage/issue43522
      - fe28b0d4fb Only join cmd if it's not a string
      - 8c671fd0c1 Update SaltConf banner per Rhett's request
    - a2161efda3 Merge pull request #43707 from terminalmage/issue43373
      - 3ebde1895f Merge branch '2016.11' into issue43373
      - e580ed4caa Merge branch '2016.11' into issue43373
      - 5b3be6e8af Fix failing unit test
      - f73764481b Add missing support for use/use_in requisites to state.sls_id
- **ISSUE** #43658: (kvnaveen) KeyError: 'as_dict' [DEBUG] LazyLoaded nested.output (refs: #43886)
- **PR** #43886: (techhat) Fix object_to_dict in azure @ *2017-10-05 19:33:56 UTC*
  - 7d174172a0 Merge pull request #43886 from techhat/azuredict
  - 223a1eea83 Fix object_to_dict in azure
- **PR** #43899: (gtmanfred) enable tox for tests @ *2017-10-04 15:08:16 UTC*
  - 7038248820 Merge pull request #43899 from gtmanfred/2017.7
  - 51eca1a6bd enable tox for tests
- **PR** #43828: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2017-10-04 13:10:13 UTC*
  - a5abe33e1c Merge pull request #43828 from rallytime/merge-2017.7
  - 2ff02e4320 Merge branch '2016.11' into '2017.7'
    - 85b3aa332a Merge pull request #43807 from terminalmage/issue43522
      - d8708bf698 cmdmod: Don't list-ify string commands on Windows
    - ea8d273c2b Merge pull request #43768 from vutny/fix-pylint-deprecation-warnings
      - f8b3fa9da1 Merge branch '2016.11' into fix-pylint-deprecation-warnings
      - 651ed16ad3 Fix Pylint deprecated option warnings
- **PR** #43854: (keesbos) Map __env__ in git_pillar before sanity checks @ *2017-10-02 20:44:53 UTC*
  - **PR** #43656: (keesbos) Git pillar fixes (refs: #43854)
  - 36b0b1174b Merge pull request #43854 from keesbos/2017.7
  - fba9c9a935 Map __env__ in git_pillar before sanity checks
- **PR** #43847: (cachedout) Fix to module.run @ *2017-10-02 19:25:03 UTC*
  - c81e8457b8 Merge pull request #43847 from cachedout/module_run_compare
  - b11f8c8f29 Merge pull request #17 from terminalmage/pr-43847
    - 93eaba7c54 Use six.iterkeys() instead of dict.keys()
    - 5d56a03a67 Improve failures for module.run states
  - 71780beb5a Merge branch '2017.7' into module_run_compare

- **ISSUE** #43819: (mephi42) archive.extracted shows the http password in the comment field on failure (refs: #43844)
- **PR** #43844: (garethgreenaway) [2017.7] Changes to states/file.py and states/archived.py @ *2017-10-01 09:08:48 UTC*
  - dd01e0ce67 Merge pull request #43844 from garethgreenaway/43819_redact_url_additions
  - c58c72aff9 When using URLs in archive.extracted, on failure the username & password is in the exception. Calling salt.utils.url.redact_http_basic_auth to ensure the credentials are redacted.
    - f0b985cbbe Merge branch 'module_run_compare' of ssh://github.com/cachedout/salt into module_run_compare
      - aefc773c2f Merge branch '2017.7' into module_run_compare
- **PR** #43840: (twangboy) Fix *unit.states.test_augeas* for Windows @ *2017-09-29 21:53:21 UTC*
  - 1f52546eab Merge pull request #43840 from twangboy/win_fix_test_augeas
  - fd1d6c31de Fix *unit.states.test_augeas* for Windows
- **ISSUE** #43553: (dafyddj) Vagrant setup (Windows guest) broken on upgrade to 2017.7 (refs: #43801)
- **PR** #43801: (terminalmage) Properly handle UNC paths in salt.utils.path.readlink() @ *2017-09-29 09:58:02 UTC*
  - c6fd2cd452 Merge pull request #43801 from terminalmage/issue43553
  - 66e6e89dc7 Properly handle UNC paths in salt.utils.path.readlink()
- **PR** #43800: (Ch3LL) Add note to nitrogen release notes about pip for cent6 @ *2017-09-28 17:36:49 UTC*
  - 7304907db6 Merge pull request #43800 from Ch3LL/update_7.0
  - 50779c3b1c Add note to nitrogen release notes about pip for cent6
- **PR** #43779: (twangboy) Fix *unit.modules.test_state* for Windows @ *2017-09-28 14:27:03 UTC*
  - 6f687fdcff Merge pull request #43779 from twangboy/win_fix_test_state
  - a64fe75816 Use os agnostic paths
- **PR** #43782: (twangboy) Fix *unit.modules.test_virt* for Windows @ *2017-09-28 14:25:16 UTC*
  - db0f569f7a Merge pull request #43782 from twangboy/win_fix_test_virt
  - 7192332758 Fix *unit.modules.test_virt* for Windows
- **PR** #43723: (nicholasmhughes) Fix ini_manage error and change handling @ *2017-09-28 09:52:09 UTC*
  - dd4fc52f1e Merge pull request #43723 from nicholasmhughes/ini_manage-error-handling
  - d68c5c4be0 prevent exception when test=True
  - cfe37916c3 handling changes per section
  - 1c484f6ad5 prevent exception when test=True
- **PR** #43781: (twangboy) Fix *unit.modules.test_status* for Windows @ *2017-09-28 09:06:19 UTC*
  - 5e29507c21 Merge pull request #43781 from twangboy/win_fix_test_status
  - 16ae8253c1 Mock which, use os.linesep for cmd.run return
- **PR** #43785: (twangboy) Fix *unit.modules.test_znc* for Windows @ *2017-09-28 08:56:11 UTC*
  - 05c78ae649 Merge pull request #43785 from twangboy/win_fix_test_znc
  - 7d90721f6b Merge branch '2017.7' into win_fix_test_znc

  - 228e74c8e3 Fix *unit.modules.test_znc* for Windows
- **PR** #43786: (twangboy) Fix *unit.modules.test_zypper* for Windows @ *2017-09-28 08:51:59 UTC*
  - 10ddb8491c Merge pull request #43786 from twangboy/win_fix_test_zypper
  - 1c05e37a66 Merge branch '2017.7' into win_fix_test_zypper
  - aafec7ab0e Fix *unit.modules.test_zypper* for Windows
- **PR** #43773: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2017-09-28 08:48:39 UTC*
  - 9615ca32d5 Merge pull request #43773 from rallytime/merge-2017.7
  - f7035ed7da Merge branch '2017.7' into merge-2017.7
  - dfef4a722c Merge branch '2016.11' into '2017.7'
    - 1a8cc60bb4 Merge pull request #43772 from gtmanfred/2016.11
      - 0194c60960 dont print Minion not responding with quiet
    - 9dee896fb9 Merge pull request #43747 from rallytime/gpg-verification
      - 7a70de19f4 Merge branch '2016.11' into gpg-verification
      - 23bb4a5dde Add GPG Verification section to Contributing Docs
- **PR** #43784: (twangboy) Fix *unit.modules.test_win_service* @ *2017-09-28 03:14:39 UTC*
  - 9a9cc69d55 Merge pull request #43784 from twangboy/win_fix_test_win_service
  - 058e50e530 Fix *unit.modules.test_win_service*
- **PR** #43774: (The-Loeki) typo fix aka what is a 'masterarpi' @ *2017-09-27 18:52:19 UTC*
  - 1254da1df5 Merge pull request #43774 from The-Loeki/patch-1
  - 84bbe85e60 typo fix aka what is a 'masterarpi'
- **PR** #43732: (twangboy) Skip *unit.stats.test_mac_packages* on Windows @ *2017-09-27 14:48:08 UTC*
  - 3f888753d4 Merge pull request #43732 from twangboy/win_skip_mac_pkg_tests
  - 1c01e06097 Only skip test on Windows
  - ec99a3ce3c Fix lint error
  - 61f8a2f7ff Skip mac specific tests
- **PR** #43761: (Ch3LL) Release Notes for 2017.7.2 @ *2017-09-27 14:34:52 UTC*
  - fb86935d99 Merge pull request #43761 from Ch3LL/release_2017.7.2
  - caf5795856 add mac patch notes
  - 3d5fce0955 Add 2017.7.2 Release Notes
- **PR** #43767: (twangboy) Skip *unit.modules.test_snapper* on Windows @ *2017-09-27 14:10:27 UTC*
  - 5ea603cf16 Merge pull request #43767 from twangboy/win_skip_test_snapper
  - b41b9c8378 Skip snapper tests on Windows
- **PR** #43759: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2017-09-27 13:30:38 UTC*
  - 77c2c7cbf7 Merge pull request #43759 from rallytime/merge-2017.7
  - 120f49f2c4 Merge branch '2016.11' into '2017.7'
    - 1cc3ad1c8d Merge pull request #43733 from terminalmage/issue43729

        - 6e5c99bda0 Allow docker_events engine to work with newer docker-py
      - 5d38be4ff7 Merge pull request #43458 from terminalmage/issue42082
        - 5f90812b12 Fix missing PER_REMOTE_ONLY in cache.clear_git_lock runner
      - 023a563657 Merge pull request #43727 from rallytime/fix-43650
        - babad12d83 Revise "Contributing" docs: merge-forwards/release branches explained!

- **ISSUE** #43737: (syedaali) salt.loaded.int.module.boto_kinesis.__virtual__() is wrongly returning *None*. It should either return *True*, *False* or a new name. If you're the developer of the module 'boto_kinesis', please fix this. (refs: #43748)
- **PR** #43748: (rallytime) Add message to boto_kinesis modules if boto libs are missing @ *2017-09-27 13:19:33 UTC*
  - 5c203df056 Merge pull request #43748 from rallytime/fix-43737
  - 5a2593dbd3 Add message to boto_kinesis modules if boto libs are missing
- **PR** #43731: (twangboy) Fix *unit.beacons.test_status* for Windows @ *2017-09-26 16:25:12 UTC*
  - 2581098595 Merge pull request #43731 from twangboy/win_unit_beacons_test_status
  - dc1b36b7e2 Change expected return for Windows
- **PR** #43724: (brejoc) Improved delete_deployment test for kubernetes module @ *2017-09-26 16:19:31 UTC*
  - 10f3d47498 Merge pull request #43724 from brejoc/2017.7.kubernetes_delete_test
  - 85b0a8c401 Improved delete_deployment test for kubernetes module
- **PR** #43734: (twangboy) Fix *unit.modules.test_poudriere* for Windows @ *2017-09-26 14:13:47 UTC*
  - 13cc27bdab Merge pull request #43734 from twangboy/win_unit_test_poudriere
  - 922e60fa67 Add os agnostic paths
- **PR** #43742: (terminalmage) Fix incorrect value in docstring @ *2017-09-26 13:55:00 UTC*
  - 41aeee7ac8 Merge pull request #43742 from terminalmage/fix-docstring
  - 553335b1c9 Fix incorrect value in docstring
- **PR** #41998: (twangboy) Fix *unit.modules.test_environ* for Windows @ *2017-09-26 12:25:48 UTC*
  - d78b9a3294 Merge pull request #41998 from twangboy/win_unit_test_environ
  - d73ef44cf6 Mock with uppercase KEY
  - 048e16883f Use uppercase KEY
- **PR** #42036: (twangboy) Fix *unit.modules.test_file* for Windows @ *2017-09-26 12:23:10 UTC*
  - 7fbbea3806 Merge pull request #42036 from twangboy/win_unit_test_file
  - 056f3bb4c0 Use with to open temp file
  - 352fe69e35 Clarify the purpose of the for loop
  - b55172d5dc Split by Windows and Linux style line endings
  - e20aa5c39b Fix line, use os.sep instead of os.linesep
  - d5f27901e3 Fix additional bytestring issue
  - 716e99c453 Fix py3 bytestring problems
  - 543610570c Fix bytestring issues, fix errored tests

  - 9fe83a34a5 Remove old variable declaration
  - c5cf5e92c1 Fix many tests
- **PR** #43557: (clan) disable modify yaml constructor @ *2017-09-25 14:03:47 UTC*
  - a81d4b8d8d Merge pull request #43557 from clan/yaml
  - 485471c8a7 Merge branch '2017.7' into yaml
  - da15658304 remove modify yaml constructor
- **PR** #43566: (damon-atkins) 2017.7 update salt.utils.files.safe_filepath func @ *2017-09-25 13:58:29 UTC*
  - b5beec16e8 Merge pull request #43566 from damon-atkins/2017.7_update_safe_filename_func
  - c7a652784a remove blank line at end of file
  - e97651d49b Merge branch '2017.7' into 2017.7_update_safe_filename_func
  - 3b4c1bbf7f Merge branch '2017.7' into 2017.7_update_safe_filename_func
  - 4c88c80ef9 Merge branch '2017.7' into 2017.7_update_safe_filename_func
  - 4171d11838 utils.files.safe_filepath add support to override the os default directory separator
- **ISSUE** #43711: (wedge-jarrad) fcontext_get_policy emits command error if policy doesn't exist (refs: #43712)
- **PR** #43712: (wedge-jarrad) Ignore retcode on call to grep in selinux.py module @ *2017-09-25 13:56:17 UTC*
  - 3bb337cf6a Merge pull request #43712 from wedge-jarrad/fix-43711
  - 96c1ef48e6 Ignore retcode on call to grep in selinux.py module
- **ISSUE** #43659: (gaborn57) unable to retrieve pillar data in postgres db (refs: #43716)
- **PR** #43716: (gaborn57) Corrected custom port handling @ *2017-09-25 13:44:58 UTC*
  - 5b7411e335 Merge pull request #43716 from gaborn57/2017.7
  - 78137c0860 Corrected custom port handling
- **PR** #43700: (rklaren) Ensure salt-cloud with libvirt provider does not write low level errors to stderr @ *2017-09-25 01:47:25 UTC*
  - **PR** #43684: (rklaren) salt-cloud libvirt updates (refs: #43700)
  - 6bbd50c453 Merge pull request #43700 from rklaren/fix-libvirt-stderr-spam
  - 88530c4cb6 Lint fixes
  - 235bec492e salt-cloud + libvirt: Mention Fedora 26 support
  - 9aecf5f847 Remove stderr spam when using salt-cloud with libvirt
- **PR** #43702: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2017-09-25 01:26:20 UTC*
  - 437ac03801 Merge pull request #43702 from rallytime/merge-2017.7
  - 132b1b343b Merge branch '2017.7' into merge-2017.7
- **ISSUE** #38971: (morganwillcock) archive.extracted: lots of unnecessary file transferring, copying, and hashing (refs: #43681, #43518)
- **PR** #43681: (terminalmage) Backport the non-fileclient changes from PR 43518 to 2017.7 @ *2017-09-22 19:27:25 UTC*
  - **PR** #43518: (terminalmage) Reduce unnecessary file downloading in archive/file states (refs: #43681)
  - 47cd8723c6 Merge pull request #43681 from terminalmage/issue38971-2017.7

  - 91edf865e2 Merge branch '2017.7' into issue38971-2017.7
  - 84f34c93be Backport the non-fileclient changes from PR 43518 to 2017.7
- **ISSUE** #43396: (mkurtak) yumpkg pkg.installed slowed down due to wildcard namig support (refs: #43687)
- **PR** #43687: (mkurtak) yumpkg.py: install calls list_repo_pkgs only if wildcard is used in pkg name @ *2017-09-22 19:23:18 UTC*
  - 0a1c5185f5 Merge pull request #43687 from mkurtak/fix-43396
  - b1e64b11fb yumpkg.py: install calls list_repo_pkgs only if wildcard in pkg name is used
- **ISSUE** #43124: (UtahDave) publisher_acl with regex on username not working and has no documentation (refs: #43467)
- **PR** #43467: (DmitryKuzmenko) Bugs/43124 users regex @ *2017-09-22 19:21:09 UTC*
  - 3a79549af4 Merge pull request #43467 from DSRCorporation/bugs/43124_users_regex
  - 14bf2dd8ff Support regex in publisher_acl.
  - 9fe32f8b6e Regex support for user names in external_auth config.
- **ISSUE** #43381: (V3XATI0N) Sharing minion data cache causes false errors in returns (refs: #43670)
- **PR** #43670: (DmitryKuzmenko) Fix for *list* and *contains* redis cache logic. @ *2017-09-22 17:56:58 UTC*
  - 0e86266b93 Merge pull request #43670 from DSRCorporation/bugs/43381_redis_cache_fix
  - 1c979d5809 Update redis cache *contains* logic to use more efficient *sismember*.
  - 039d236948 Fixed *list* and *contains* redis cache logic.
    - 6e5cf65d65 Merge branch '2016.11' into '2017.7'
    - f46c858f25 Merge pull request #43648 from rallytime/handle-boto-vpc-errors
      - 54842b5012 Handle VPC/Subnet ID not found errors in boto_vpc module
- **PR** #43697: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2017-09-22 17:31:09 UTC*
  - aa47da35dd Merge pull request #43697 from rallytime/merge-2017.7
  - cbae45bec4 Lint: Remove extra line at end of file
  - fca4e5563a Merge branch '2016.11' into '2017.7'
  - 9dba34aa06 Merge pull request #43575 from akissa/fix-csr-not-recreated-if-key-changes
    - b1b4dafd39 Fix CSR not recreated if key changes
  - 1d4fa48209 Merge pull request #43672 from rallytime/bp-43415
    - 3fb42bc238 Fix env_order in state.py
  - ff832ee607 Merge pull request #43673 from rallytime/bp-43652
    - d91c47c6f0 Salt Repo has Deb 9 and 8
  - 365cb9fba8 Merge pull request #43677 from terminalmage/runners-docs-2016.11
    - 2fd88e94fa Fix RST headers for runners (2016.11 branch)
  - be38239e5d Merge pull request #43534 from twangboy/win_fix_pkg.install_2016.11
    - 1546c1ca04 Add posix=False to call to salt.utils.shlex_split
  - 0d3fd3d374 Merge pull request #43661 from moio/2016.11-multiprocessing-doc-fix
    - 625eabb83f multiprocessing minion option: documentation fixes

  - 6b4516c025 Merge pull request #43646 from brejoc/2016.11.4-pidfile-tests
    - 96f39a420b Fixed linting
    - 08fba98735 Fixed several issues with the test
    - 3a089e450f Added tests for pid-file deletion in DaemonMixIn
  - cfb1625741 Merge pull request #43591 from rallytime/merge-2016.11
    - 57b9d642c2 Merge branch '2016.11.8' into '2016.11'
      - e83421694f Merge pull request #43550 from twangboy/osx_fix_preinstall_2016.11.8
      - 1b0a4d39d2 Fix logic in */etc/paths.d/salt* detection
      - a648f75949 Merge pull request #43508 from rallytime/bp-43333
      - d4981a2717 Update doco
      - a7c8b9e048 Update win_pkg.py
      - 1d6dc6fb72 Docs are wrong cache_dir (bool) and cache_file (str) cannot be passed on the cli (#2)
      - e7009877bc Merge pull request #43434 from rallytime/2016.11.8-release-notes
      - 68f529ee5e Add 2016.11.8 release notes
  - 8671b91f62 Merge pull request #43572 from vutny/fix-salt-cloud-list-min-instance-set
    - 21966e7ce8 cloud.action: list_nodes_min returns all instances
- **PR** #43314: (twangboy) Fix *unit.utils.test_verify* for Windows @ *2017-09-21 22:26:13 UTC*
  - e6dc4d64df Merge pull request #43314 from twangboy/win_fix_unit.utils.test_verify
  - 9ada7f626c Merge branch '2017.7' into win_fix_unit.utils.test_verify
  - c0dc3f73ef Use sys.platform instead of salt.utils to detect Windows
  - e496d28cbf Fix *unit.utils.test_verify* for Windows
- **ISSUE** #43599: (vernondcole) Incorrect default for salt.cache.Cache() if opts does not define "cache" (refs: #43680)
- **PR** #43680: (vernondcole) correct default value for salt.cache.Cache @ *2017-09-21 20:09:36 UTC*
  - ec34df2c27 Merge pull request #43680 from vernondcole/fix-salt.cache.Cache-default
  - 292f8c79b8 correct default value for salt.cache.Cache
- **PR** #43530: (twangboy) Fixes removal of double-quotes by shlex_split in winrepo @ *2017-09-21 18:04:48 UTC*
  - 99d9d784b1 Merge pull request #43530 from twangboy/win_fix_pkg.install
  - 7f59119f95 Merge branch '2017.7' into win_fix_pkg.install
  - f146399f7a Use posix=False for shlex.split
- **PR** #43671: (rallytime) [2017.7] Merge forward from 2017.7.2 to 2017.7 @ *2017-09-21 16:39:49 UTC*
  - 12b5e62d81 Merge pull request #43671 from rallytime/merge-2017.7
  - a401166bd5 Merge branch '2017.7.2' into '2017.7'
- **PR** #43676: (terminalmage) Fix RST headers for runners (2017.7 branch) @ *2017-09-21 16:36:21 UTC*
  - e3a2fbc2a3 Merge pull request #43676 from terminalmage/runners-docs-2017.7
  - 9b74634b23 Fix badly-formatted RST in mattermost runner docstring

  - c0a79c70a4 Fix RST headers for runners (2017.7 branch)
- **PR** #43235: (brejoc) Improve delete_deployment handling @ *2017-09-20 21:33:33 UTC*
  - d02953ce6a Merge pull request #43235 from brejoc/improve-async-operation-handling-in-kubernetes-module
  - 4e8da3045f Fixed logic for windows fallback
  - 3b1cb884b9 Merge branch '2017.7' into improve-async-operation-handling-in-kubernetes-module
  - d1b5ec098c Merge branch '2017.7' into improve-async-operation-handling-in-kubernetes-module
  - 35cf69bc50 Moved exception Salt core
  - 7431ec64e3 Removed unused sys import
  - 0c71da95f6 Using salt method to identify MS Windows, single instead of double quotes
  - 20619b24c4 Fixed test for delete_deployment
  - 91076bbafa Merge branch '2017.7' into improve-async-operation-handling-in-kubernetes-module
  - 7b600e2832 Added pylint-disable statements and import for salt.ext.six.moves.range
  - 99fe138325 Code styling and added log message for timeout
  - dcd8d4f639 Merge branch '2017.7' into improve-async-operation-handling-in-kubernetes-module
  - 702a058c38 Fixed linting
  - 3fe623778e Added Windows fallback
  - 52b1cb8147 Compatibility with Python3.6
  - 767af9bb4f Added timeout for checking the deployment
  - 32d7d34fe5 First simple draft for the deletion verification
- **PR** #43554: (twangboy) Win fix chocolatey @ *2017-09-20 16:06:18 UTC*
  - 73cb0c27b5 Merge pull request #43554 from twangboy/win_fix_chocolatey
  - e04acb6216 Merge branch '2017.7' into win_fix_chocolatey
  - 56be5c35eb Improve logic for handling chocolatey states
  - bcbf7b4e68 Add logic for test=True
- **ISSUE** #43598: (davidvon) Passed invalid arguments to mysql.file_query: unsupported operand type(s) for +=: 'int' and 'tuple' (refs: #43625)
- **PR** #43625: (gtmanfred) results and columns are lists for mysql returns @ *2017-09-20 15:42:59 UTC*
  - ed7eeaaafb Merge pull request #43625 from gtmanfred/2017.7
  - f84b50a06b results and columns are lists for mysql returns
- **ISSUE** #43560: (smitelli) salt.states.linux_acl requires setfacl/getacl binaries but this is not obvious (refs: #43587, #43580)
- **PR** #43587: (rallytime) Add reason to linux_acl state loading failure @ *2017-09-19 16:26:51 UTC*
  - **PR** #43580: (garethgreenaway) Updating ACL module and state module documentation (refs: #43587)
  - 1bda4832ef Merge pull request #43587 from rallytime/fix-virtual
  - e5297e3869 Add reason to linux_acl state loading failure
- **PR** #43584: (cachedout) Enhance engines docs @ *2017-09-18 20:40:57 UTC*

  - 2e19533e3c Merge pull request #43584 from cachedout/engines_doc_clarification
  - 634536b0ff Merge branch '2017.7' into engines_doc_clarification
  - 1a619708c1 Enhance engines docs
- **PR** #43519: (terminalmage) Fix incorrect handling of pkg virtual and os_family grain @ *2017-09-18 20:35:01 UTC*
  - 50b134ef4c Merge pull request #43519 from terminalmage/fix-aptpkg
  - 0e3c447567 Fix incorrect handling of pkg virtual and os_family grain
- **PR** #43520: (clan) _search_name is " if acl type is other @ *2017-09-18 20:33:51 UTC*
  - dd953f36ae Merge pull request #43520 from clan/acl
  - 54216177c1 _search_name is " if acl type is other
- **PR** #43561: (wedge-jarrad) Clean up doc formatting in selinux state & module @ *2017-09-18 20:28:47 UTC*
  - ad9663a7fc Merge pull request #43561 from wedge-jarrad/selinux-doc-cleanup
  - 1bd263cd51 Clean up doc formatting in selinux state & module
- **ISSUE** #43560: (smitelli) salt.states.linux_acl requires setfacl/getacl binaries but this is not obvious (refs: #43587, #43580)
- **PR** #43580: (garethgreenaway) Updating ACL module and state module documentation (refs: #43587) @ *2017-09-18 20:11:26 UTC*
  - cc3d9c1a01 Merge pull request #43580 from garethgreenaway/43560_update_linux_acl_documentation
  - e63fae4c91 Merge branch '2017.7' into 43560_update_linux_acl_documentation
- **PR** #43523: (skizunov) Add back lost logic for multifunc_ordered @ *2017-09-18 17:46:16 UTC*
  - **PR** #38168: (skizunov) Add support for a multi-func job using same func more than once (refs: #43523)
  - bf7b23316f Merge pull request #43523 from skizunov/develop2
  - fb579321a9 Add back lost logic for multifunc_ordered
    - 117a0ddbbc Updating the documentation to call out the requirement for the getfacl and setfacl binaries
      - 49f25b9f19 Lint
      - 31d17c0124 Fix typo found by @s0undt3ch
      - 5dba74d2cb Fix to module.run [WIP]
- **ISSUE** #43447: (UtahDave) When using Syndic with Multi Master the top level master doesn't reliably get returns from lower minion. (refs: #43526)
- **PR** #43526: (DmitryKuzmenko) Forward events to all masters syndic connected to @ *2017-09-18 16:54:46 UTC*
  - e29efecf4f Merge pull request #43526 from DSRCorporation/bugs/43447_syndic_events_forwarding
  - 64d6109654 Merge branch '2017.7' into bugs/43447_syndic_events_forwarding
  - 3b2a529385 Merge branch '2017.7' into bugs/43447_syndic_events_forwarding
  - 0e4a744d95 Forward events to all masters syndic connected to.
- **ISSUE** #43077: (Manoj2087) Issue with deleting key via wheel (refs: #43330)
- **PR** #43330: (terminalmage) Fix reactor regression + unify reactor config schema @ *2017-09-18 16:46:11 UTC*
  - 56b671e087 Merge pull request #43330 from terminalmage/issue43077

  - a7b4e1f782 Simplify client logic
  - b85c8510c7 Improve the reactor documentation
  - 20f6f3cc39 Include a better example for reactor in master conf file
  - 4243a2211d Rewrite the reactor unit tests
  - 9db3f5ae6d Unify reactor configuration, fix caller reactors
  - 34b6c3b65f Un-deprecate passing kwargs outside of 'kwarg' param
- **ISSUE** #33793: (mstarostik) states.ssh_auth adds bogus newline before newly added keys (refs: #43483)
- **PR** #43505: (rallytime) Back-port #43483 to 2017.7 @ *2017-09-15 21:22:12 UTC*
  - **PR** #43483: (3add3287) Handle bogus newline before newly added keys (refs: #43505)
  - 078d5d17de Merge pull request #43505 from rallytime/bp-43483
  - c68dd5b8a4 Lint: fix spacing
  - 406f61ac9a Fix indentation from tabs to spaces
  - 923ec62771 Copy paste typo
  - 6f6619242f Fix checking for newline on end of file by properly checking the last byte of the file if the file is non empty.
- **ISSUE** #43464: (psagers) acme.cert state: IOError on failure to create a new certificate (refs: #43465)
- **PR** #43491: (rallytime) Back-port #43465 to 2017.7 @ *2017-09-15 18:24:47 UTC*
  - **PR** #43465: (psagers) acme.cert: avoid IOError on failure. (refs: #43491)
  - a6df3f2acc Merge pull request #43491 from rallytime/bp-43465
  - 3118faca0a acme.cert: avoid IOError on failure.
- **PR** #43492: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2017-09-15 18:23:49 UTC*
  - 3620c15c9a Merge pull request #43492 from rallytime/merge-2017.7
  - 4251ce5a27 Merge branch '2016.11' into '2017.7'
    - f2b86fa2db Merge pull request #43461 from twangboy/win_norestart
      - 2d269d1a76 Change all comment markers to '#'
      - d80aea16cb Handle ErrorCodes returned by VCRedist installer
      - fb31e9a530 Add /norestart switch to vcredist install
    - 90e8ca9c36 Merge pull request #43366 from brejoc/2016.11.pidfile-fix
      - 6e3eb76c79 Removed unused format argument
      - daf4948b3d Catching error when PIDfile cannot be deleted
    - a6c458607a Merge pull request #43442 from garethgreenaway/43386_2016_11_schedule_kwargs_pub
      - e637ecbe86 Merge branch '2016.11' into 43386_2016_11_schedule_kwargs_pub
      - 6114df8dc3 Adding a small check to ensure we do not continue to populate kwargs with __pub_ items from the kwargs item.
    - 3c429299f9 Merge pull request #43456 from rallytime/43445_follow_up
      - 35c1d8898d Add Neon to version list

    - 6db7a721c0 Merge pull request #43441 from meaksh/2016.11-salt-bash-completion-fix
        - be4f26ab21 Use $HOME to get the user home directory instead using '~' char
    - 05fff44a50 Merge pull request #43445 from rallytime/bump-deprecation-warning
        - c91cd1c6d9 Bump deprecation warning for boto_vpc.describe_route_table
    - c57dc5f0e3 Merge pull request #43432 from rallytime/bp-43419
        - c471a29527 make cache dirs when spm starts

- **ISSUE** #43479: (haam3r) Mattermost runner failing to retrieve config values due to unavailable config runner (refs: #43513)
- **PR** #43513: (haam3r) Issue #43479 No runners.config in 2017.7 branch @ *2017-09-15 14:58:27 UTC*
  - 8a90c7059b Merge pull request #43513 from haam3r/2017.7
  - 58f7d051c9 Issue #43479 No runners.config in 2017.7 branch
- **ISSUE** #42926: (nixjdm) network.system not setting hostname in hosts file, preventing sudo. (refs: #43431)
- **PR** #43431: (mattLLVW) Fix /etc/hosts not being modified when hostname is changed @ *2017-09-13 18:35:55 UTC*
  - c3d9e2d9b2 Merge pull request #43431 from mattLLVW/fix-hosts-deb
  - c6320b1dff Merge branch '2017.7' into fix-hosts-deb
  - a3b2e19149 Fix /etc/hosts not being modified when hostname is changed
- **PR** #43403: (twangboy) Proper timestamp conversion in *redis.lastsave* @ *2017-09-12 21:18:06 UTC*
  - a09f289fbb Merge pull request #43403 from twangboy/win_fix_redismod
  - f6da23e1aa Properly handle timestamp conversion
- **PR** #43463: (twangboy) Add */norestart* switch to vcredist installer @ *2017-09-12 20:29:27 UTC*
  - 0eaa5acb72 Merge pull request #43463 from twangboy/win_norestart_2017.7
  - 6984b8fd60 Add /norestart to vcredist installer
- **ISSUE** #43386: (rajvidhimar) Scheduler's job_kwargs not working as expected. (refs: #43443, #43442)
- **PR** #43443: (garethgreenaway) [2017.7] Fixes to scheduler __pub values in kwargs @ *2017-09-12 18:14:46 UTC*
  - 2fc237a806 Merge pull request #43443 from garethgreenaway/43386_2017_7_schedule_kwargs_pub
  - a29a9855a6 Fixing typo.
  - 2681b7d3fa Merge branch '2017.7' into 43386_2017_7_schedule_kwargs_pub
- **ISSUE** #39775: (mirceaulinic) Proxy *mine_interval* config ignored (refs: #41547)
- **PR** #41547: (mirceaulinic) Override proxy minion opts with pillar data @ *2017-09-11 21:47:51 UTC*
  - 5378ac7756 Merge pull request #41547 from cloudflare/px_merge_pillar_opts
  - aad39ba665 Document the new opts
  - cdc0d9674a Allow disabling the mines details merge
  - 732b63b0b9 Merge mine details whenever possible
  - 96b31d5643 Override proxy opts with pillar data when required
  - fd499887f9 Define new proxy merge pillar in opts… opts
  - abab6fd91c Override minion opts with pillar data

- **PR** #41943: (twangboy) Fix *unit.returners.test_local_cache* for Windows @ *2017-09-11 21:34:03 UTC*
  - 08d102c869 Merge pull request #41943 from twangboy/win_unit_test_local_cache
  - 3777b34572 Merge branch '2017.7' into win_unit_test_local_cache
  - 35b79ecde6 Remove *cur* variable, use time.time() in comparison
  - 9b61533b09 Get more accurate currnet time in local_cache
  - 844e3f65bc Fix unit tests for Windows
- **PR** #43424: (twangboy) Fix *unit.modules.test_hosts* for Windows @ *2017-09-11 21:28:41 UTC*
  - 50ab79f0cb Merge pull request #43424 from twangboy/win_unit_test_hosts
  - 90dcf8287c Fix *unit.modules.test_hosts* for Windows
- **PR** #42652: (skizunov) Fix loader.py's raw_mod() to look in all module dirs @ *2017-09-11 19:43:48 UTC*
  - 0f0ed5a093 Merge pull request #42652 from skizunov/develop3
  - d82e406f15 Fix loader.py's raw_mod() to look in all module dirs
- **PR** #43438: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2017-09-11 18:33:39 UTC*
  - ca091bc8a4 Merge pull request #43438 from rallytime/merge-2017.7
  - ef7b4242c3 Merge branch '2016.11' into '2017.7'
    - 57cccd75d0 Merge pull request #43390 from aogier/43387-genesis-qemu
      - 496f14a7e7 forgot to mock the proper one
      - 51c7a1ba00 only check if static_qemu is_executable()
      - 70642e495d better qemu_static parameter mangle in deboostrap management, tests
    - 6106aec696 Merge pull request #43356 from gtmanfred/2016.11
      - 3f19b247f3 Add handler.messages back in for test comparison
      - 9911b04208 fix test
      - 3c6ae99a77 never-download got readded
    - e638fac54e Merge pull request #43325 from doesitblend/salt-mine-doc-fix
      - 1e94d0ac3a Lint: Remove trailing whitespace
      - 51af8f8757 Fix mine_interval phrasing in default file
      - ba0cdd4536 Fix phrasing for mine_interval description
      - 9ff03c2d43 Update Salt Mine documentation to show that the mine_interval option is configured in minutes.
    - fc587f784a Merge pull request #43105 from aogier/43086-no-member
      - 5111cf8bad Merge branch '2016.11' into 43086-no-member
    - d97a680372 Merge pull request #43333 from damon-atkins/2016.11
      - 92de2bb498 Update doco
      - fc9c61d12e Update win_pkg.py
      - c91fc14704 Merge branch '2016.11' into 2016.11
      - cb3af2bbbd Docs are wrong cache_dir (bool) and cache_file (str) cannot be passed on the cli (#2)

      - · 42a118ff56 fixed cmd composition and unified his making across module
      - · 3fd59ed369 Adding a small check to ensure we do not continue to populate kwargs with __pub_ items from the kwargs item.

- **PR** #43320: (twangboy) Fix *unit.modules.test_alternatives* for Windows @ *2017-09-11 17:28:00 UTC*
  - a9592dd3e2 Merge pull request #43320 from twangboy/win_fix_alternatives
  - a909813fa5 Remove unused import (lint)
  - 3ef8d714cb Fix unit tests to mock salt.utils.path.readlink
  - c0d81aa1ce Use salt.utils.path.readlink
  - 7c4460164b Fix alternatives for Windows
- **PR** #43363: (twangboy) Fix *unit.modules.test_ini_manage* for Windows @ *2017-09-11 17:10:31 UTC*
  - 9b89e49846 Merge pull request #43363 from twangboy/scratch_ini_tests
  - a94319a082 Make sure formatting of TEST_FILE_CONTENT matches original
  - 6263bc8983 Remove print statement
  - 79cd3831ae Fix empty value preserved test
  - 85997391f1 Is this handled the same on Linux and Windows
- **PR** #43421: (gtmanfred) Revert "Reduce fileclient.get_file latency by merging _file_find and … @ *2017-09-11 17:07:18 UTC*
  - 673ce387c1 Merge pull request #43421 from gtmanfred/compat
  - f85bf8c18f Revert "Reduce fileclient.get_file latency by merging _file_find and _file_hash"
- **ISSUE** #42165: (arount) top_file_merging_strategy: merge does not works (refs: #43415)
- **PR** #43415: (mattLLVW) Fix env_order in state.py (refs: #43672) @ *2017-09-11 15:18:08 UTC*
  - 47d982fd37 Merge pull request #43415 from mattLLVW/fix-env-order
  - f6313a1b2c Merge branch '2017.7' into fix-env-order
  - e93a962980 Fix env_order in state.py
- **PR** #43422: (twangboy) Fix *unit.cloud.clouds.test_ec2* for Windows @ *2017-09-11 15:17:20 UTC*
  - e89e23a32e Merge pull request #43422 from twangboy/win_unit_cloud_ec2
  - 1379627334 Fix *unit.cloud.clouds.test_ec2* for Windows
- **PR** #43423: (twangboy) Fix *unit.modules.test_gem* for Windows @ *2017-09-11 15:15:28 UTC*
  - 54f833ac59 Merge pull request #43423 from twangboy/win_unit_test_gem
  - b2cea18d13 Fix *unit.modules.test_gem* for Windows
- **PR** #43419: (gtmanfred) make cache dirs when spm starts (refs: #43432) @ *2017-09-11 13:42:50 UTC*
  - b3116109e5 Merge pull request #43419 from gtmanfred/2017.7
  - 58378866e5 make cache dirs when spm starts
- **PR** #43371: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2017-09-08 15:39:12 UTC*
  - 9b27473763 Merge pull request #43371 from rallytime/merge-2017.7
  - 7b07b58396 Merge branch '2016.11' into '2017.7'
    - 0c986f5eba Merge pull request #43361 from rallytime/bp-43329

      - b09e5b4379 Fix #43295, better handling of consul initialization issues
    - 22287439e6 Merge pull request #42903 from junovitch/issue-35840-fix-preserve-minion-cache-2016.11
      - c9d4fdbd45 Merge branch '2016.11' into issue-35840-fix-preserve-minion-cache-2016.11
      - 93a68e32a5 Merge branch '2016.11' into issue-35840-fix-preserve-minion-cache-2016.11
      - 079f097985 Fix 'preserve_minion_cache: True' functionality (fixes #35840)
    - 4860e10757 Merge pull request #43360 from terminalmage/sj-496
      - 433bca14b1 Fix KeyError in yumpkg configparser code on Python 3
      - f6c16935d8 Move –showduplicates before repository-packages
    - 4ba2dbe41e Merge pull request #43244 from rallytime/release-branch-clarifications
      - 0d5a46dbaa Update release branch section with a few more details
    - 1a012eb3d7 Merge pull request #43359 from gtmanfred/ipaddr
      - 23d9abb560 ipaddr_start ipaddr_end for el7
    - 8f88111be8 Merge pull request #43247 from rallytime/mentionbot-backports
      - 2b85757d73 Always notify tkwilliams when changes occur on boto files
      - 40b5a29f90 Add basepi to userBlacklist for mention bot
      - bad8f56969 Always notify ryan-lane when changes occur on boto files

- **PR** #43398: (twangboy) Fix *unit.modules.test_mount* for Windows @ *2017-09-08 13:39:29 UTC*
  - 97f05ff603 Merge pull request #43398 from twangboy/win_fix_test_mount
  - 4a8d7e522c Fix tests, Use full path to salt.utils.which
- **PR** #43399: (twangboy) Fix *unit.modules.test_pam* for Windows @ *2017-09-08 13:37:50 UTC*
  - 6a4cc5c1b0 Merge pull request #43399 from twangboy/win_fix_test_pam
  - 6257aa964a Fix *unit.modules.test_pam* for Windows
- **PR** #43400: (twangboy) Fix *unit.modules.test_parted* for Windows @ *2017-09-08 13:37:00 UTC*
  - 2b5cfae3f8 Merge pull request #43400 from twangboy/win_unit_test_parted
  - 8e3e897ee2 Fix *unit.modules.test_parted* for Windows
- **PR** #43401: (twangboy) Fix *unit.modules.test_pw_group* for Windows @ *2017-09-08 13:35:45 UTC*
  - 332deeb013 Merge pull request #43401 from twangboy/win_unit_test_pw_group
  - 78e39a1b9d Fix *unit.modules.test_pw_group* for Windows
- **PR** #43402: (twangboy) Fix *unit.modules.test_qemu_nbd* for Windows @ *2017-09-08 13:34:58 UTC*
  - c0f54bfef1 Merge pull request #43402 from twangboy/win_unit_test_qemu_nbd
  - 531ce8022b Fix *unit.modules.test_qemu_nbd* for Windows
- **PR** #43404: (twangboy) Fix *unit.modules.test_seed* for Windows @ *2017-09-08 13:32:41 UTC*
  - be88fbb45f Merge pull request #43404 from twangboy/win_unit_test_seed
  - 6ceb895a84 Use os.path.join for paths
- **PR** #43301: (twangboy) Fix *unit.test_spm* for Windows @ *2017-09-08 13:24:35 UTC*

  - 612c6a8756 Merge pull request #43301 from twangboy/win_fix_unit_test_spm
  - 8608a6b303 Merge branch '2017.7' into win_fix_unit_test_spm
  - b8da04c04d Add Mike's changes
  - f36efbd6a7 Fix *unit.test_spm* for Windows
- **PR** #43372: (skizunov) Fix system.set_system_time when no hw clock is present @ *2017-09-07 17:45:33 UTC*
  - f959113694 Merge pull request #43372 from skizunov/develop5
  - 281e471853 Fix system.set_system_time when no hw clock is present
- **PR** #43193: (jettero) Prevent spurious "Template does not exist" error @ *2017-09-06 20:16:58 UTC*
  - **PR** #39516: (jettero) Prevent spurious "Template does not exist" error (refs: #43193)
  - 6d13535ed0 Merge pull request #43193 from jettero/template-dne-again
  - cde8aed2cf Merge branch '2017.7' into template-dne-again
- **ISSUE** #42706: (blarghmatey) Parallel Cache Failure (refs: #43018, #43159)
- **PR** #43159: (jubrad) Bp 43018 @ *2017-09-05 22:29:16 UTC*
  - **PR** #43056: (damon-atkins) safe_filename_leaf(file_basename) and safe_filepath(file_path_name) (refs: #43159, #43172)
  - **PR** #43018: (jubrad) Update state.py (refs: #43159, #43727)
  - 015cbc57d9 Merge pull request #43159 from jubrad/bp-43018
  - 25419a56db Merge branch '2017.7' into bp-43018
  - 971b4c0890 Merge branch '2017.7' into bp-43018
  - 4f8e6c65e5 access safe_filename_leaf through utils.files, changed in #43172
  - 42064883ea state.py remove unused urllib import
  - 4957268b37 update state.py to use safe_filename_leaf
  - b8ead879ed Fixing lint issues
  - 446457d017 Swapping *from* for *import*
  - fb80e17400 state.py: fix import and utf8 encode before quote
  - 1dcf167bb7 Update state.py
- **PR** #43232: (terminalmage) Improve inheritance in salt.utils.gitfs @ *2017-09-05 20:37:06 UTC*
  - 6e1b541b46 Merge pull request #43232 from terminalmage/gitfs-inheritance
  - 53bd3a3e23 Improve inheritance in salt.utils.gitfs
- **PR** #43238: (s0undt3ch) Include the line number by default on the log file format @ *2017-09-05 20:31:54 UTC*
  - 086b220091 Merge pull request #43238 from s0undt3ch/2017.7
  - 630a1db3ab Include the line number by default on the log file format
- **PR** #43294: (twangboy) Win build scripts @ *2017-09-05 20:12:54 UTC*
  - 09dc58cde5 Merge pull request #43294 from twangboy/win_build_scripts
  - 9979ccb613 Remove Py2 and Py3 in the same run
  - a5d9f85db6 Modifications to build scripts

- **PR** #43322: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2017-09-05 18:21:26 UTC*
  - 21ab306ef4 Merge pull request #43322 from rallytime/merge-2017.7
  - b1062f8c15 Merge branch '2016.11' into '2017.7'
    - 02867fdcd2 Merge pull request #43277 from rallytime/owners-file
      - 2b4da0f0e7 Add CODEOWNERS file
    - 1c1c484479 Merge pull request #43312 from lordcirth/fix-cron-docs
      - ec94a13750 cron docs: Remind user to use quotes for special strings
    - 0d1ed4b750 Merge pull request #43290 from lordcirth/fix-file-path-docs
      - 14a4591854 file.py docs: correct group and mode
      - d4214ca283 file.py docs: specify absolute paths
    - 26ff89539e Merge pull request #43274 from terminalmage/fix-int-types
      - d533877743 Use six.integer_types instead of int
    - cf21f91fb2 Merge pull request #43271 from twangboy/win_fix_pkg.install
      - 91b062f564 Fix formatting issue, spaces surrounding +
- **PR** #43324: (twangboy) Fix *unit.modules.test_chef* for Windows @ *2017-09-05 16:40:11 UTC*
  - 62429c547d Merge pull request #43324 from twangboy/fix_unit.modules.test_chef
  - 5bd5ea042a Fix *unit.modules.test_chef* for Windows
- **PR** #43268: (rallytime) Back-port #43237 to 2017.7 @ *2017-09-01 18:17:13 UTC*
  - **PR** #43237: (timka) .utils.aws.get_location() expects a dict (refs: #43268)
  - 367668a0a3 Merge pull request #43268 from rallytime/bp-43237
  - 047ad07da4 .utils.aws.get_location() expects a dict
- **PR** #43270: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2017-09-01 18:09:46 UTC*
  - 02504dd363 Merge pull request #43270 from rallytime/merge-2017.7
  - f8b025f6dc Merge branch '2016.11' into '2017.7'
  - 3a0b02f3ae Merge pull request #43228 from twangboy/win_fix_pkg.install
    - 13dfabb1ce Fix regex statement, add .
    - 31ff69f0ad Add underscore to regex search
    - 3cf2b6575c Fix spelling
    - ed030a35a5 Use regex to detect salt-minion install
    - e5daff495a Fix pkg.install
  - b4c689dff5 Merge pull request #43191 from viktorkrivak/fix-apache-config-multi-entity
    - c15bcbe1cc Merge remote-tracking branch 'upstream/2016.11' into fix-apache-config-multi-entity
    - 4164047951 Fix apache.config with multiple statement At this moment when you post more than one statement in config only last is used. Also file is rewrited multiple times until last statement is written. Example: salt '*' apache.config /etc/httpd/conf.d/ports.conf config="[{'Listen': '8080'}, {'Proxy': "Something"}]" Ends only with Proxy Something and ignore Listen 8080, This patch fix this issue.

  - b90e59ede9 Merge pull request #43154 from lomeroe/bp-43116-2016.11
  - 8f593b0b02 verify that files exist before trying to remove them, win_file.remove raises an exception if the file does not exist
  - 33a30bac06 correcting bad format statement in search for policy to be disabled
  - acc3d7ac82 correct fopen calls from salt.utils for 2016.11's utils function
  - 2da1cdd109 lint fix
  - 61bd12c0de track xml namespace to ensure policies w/duplicate IDs or Names do not conflict
  - f232bed9f9 add additional checks for ADM policies that have the same ADMX policy ID (#42279)
- **ISSUE** #42459: (iavael) Broken ldap groups retrieval in salt.auth.ldap after upgrade to 2017.7 (refs: #43283)
- **PR** #43283: (DmitryKuzmenko) Fix ldap token groups auth. @ *2017-09-01 17:49:46 UTC*
  - ece0e393ef Merge pull request #43283 from DSRCorporation/bugs/42459_broken_ldap_groups
  - 3ad6911210 Fix for tests: don't require 'groups' in the eauth token.
  - 1f104cf85b Fix ldap token groups auth.
- **PR** #43149: (BenoitKnecht) Fix iptables.get_rules when rules contain –nfmask or –ctmask @ *2017-09-01 15:57:05 UTC*
  - 4f023c4cb6 Merge pull request #43149 from BenoitKnecht/2017.7.1
  - 3c1ddc9bde modules: iptables: correctly parse *–nfmask*/*–ctmask*
- **ISSUE** #43258: (nomeelnoj) metadata_server_grains problems (refs: #43265)
- **PR** #43265: (gtmanfred) make sure meta-data grains work on ec2 @ *2017-09-01 15:31:12 UTC*
  - cf2b75bb86 Merge pull request #43265 from gtmanfred/2017.7
  - 04dd8ebedb make sure meta-data grains work on ec2
- **PR** #43299: (twangboy) Fix *unit.netapi.rest_cherrypy.test_tools* for Windows @ *2017-09-01 15:13:43 UTC*
  - 618b221895 Merge pull request #43299 from twangboy/win_fix_netapi_cherrypy
  - fd74acb603 Merge branch '2017.7' into win_fix_netapi_cherrypy
- **PR** #43300: (twangboy) Fix *unit.netapi.rest_tornado.test_handlers* for Windows @ *2017-09-01 13:10:11 UTC*
  - aee654da92 Merge pull request #43300 from twangboy/win_fix_netapi_rest_tornado
  - c93d2ed386 Use os.sep instead of '/'
  - 3fbf24b91a Use os.sep instead of '/'
- **ISSUE** #43259: (mahesh21) NameError: global name '__opts__' is not defined (refs: #43266)
- **PR** #43278: (gtmanfred) bootstrap can come from dunders @ *2017-08-31 13:31:20 UTC*
  - **PR** #43266: (gtmanfred) switch virtualbox cloud driver to use __utils__ (refs: #43278)
  - aed2975979 Merge pull request #43278 from gtmanfred/virtualbox
  - c4ae2de30f bootstrap can come from dunders
- **PR** #42975: (brejoc) Added unit tests for Kubernetes module @ *2017-08-30 20:30:16 UTC*
  - 479e0e06ac Merge pull request #42975 from brejoc/tests-for-kubernetes-module
  - fdad9177b5 Merge branch '2017.7' into tests-for-kubernetes-module
  - c227cb25ad Skipping test on ImportError

  - bd76a870ce Dunder vars are now defined via setup_loader_modules
  - 3c99e61637 Renamed test to match new convention
  - caf78d206d Fixed imports for pytest
  - c8e98c8d8a Added unit tests for Kubernetes module
- **ISSUE** #42935: (BenjaminSchubert) docker_image.present always ends up failing even on correct result. (refs: #43176)
- **PR** #43176: (terminalmage) docker_image states: Handle Hub images prefixed with "docker.io/" @ *2017-08-30 20:08:13 UTC*
  - ca7df1d4cf Merge pull request #43176 from terminalmage/issue42935
  - df18a89836 Lint: Remove unused import
  - 7279f98e92 docker_image states: Handle Hub images prefixed with "docker.io/"
  - f7c945f6e4 Prevent spurious "Template does not exist" error

28.2.24 Salt 2017.7.4 Release Notes

Version 2017.7.4 is a bugfix release for *2017.7.0*.

Statistics

- Total Merges: **8**
- Total Issue References: **4**
- Total PR References: **11**
- Contributors: **6** (Ch3LL, garethgreenaway, gtmanfred, marccardinal, rallytime, terminalmage)

Changelog for v2017.7.3..v2017.7.4

Generated at: 2018-05-26 21:48:28 UTC

- **PR** #46074: (Ch3LL) Update 2017.7.4 Release Notes with new fixes @ *2018-02-16 16:47:56 UTC*
  - b5b083fd26 Merge pull request #46074 from Ch3LL/update-7.4
  - 8d0eeeb059 Update 2017.7.4 Release Notes with new fixes
- **ISSUE** #45790: (bdarnell) Test with Tornado 5.0b1 (refs: #46066)
- **PR** #46066: (rallytime) Pin tornado version in requirements file @ *2018-02-16 16:40:05 UTC*
  - 32f3d00e44 Merge pull request #46066 from rallytime/pin-tornado
  - 6dc1a3b9dc Pin tornado version in requirements file
- **PR** #46036: (terminalmage) git.latest: Fix regression with identity file usage @ *2018-02-16 13:57:23 UTC*
  - 85761ee650 Merge pull request #46036 from terminalmage/issue43769
  - e2140d9a84 Mock the ssh.key_is_encrypted utils func
  - 169924b3fe Move ssh.key_is_encrypted to a utils module temporarily
  - 54f4d78f7a Only keep ssh.py in the Windows installer

  - 5f04531e1b Keep ssh state and execution modules in the installer
  - f2b69f703d git.latest: Fix regression with identity file usage
- **PR** #46009: (Ch3LL) Add 2017.7.4 Release Notes with PRs @ *2018-02-13 16:40:30 UTC*
  - 6d534c6e7e Merge pull request #46009 from Ch3LL/rn_7.4
  - ac0baf4b34 Add 2017.7.4 Release Notes with PRs
- **ISSUE** #45976: (ghost) 6a5e0f9 introduces regression that breaks Vault module for salt masterless (refs: #45981)
- **PR** #45981: (gtmanfred) use local config for vault when masterless @ *2018-02-13 15:22:01 UTC*
  - ca76a0b328 Merge pull request #45981 from gtmanfred/2017.7.3
  - 0d448457dc apparently local is not set by default
  - 2a92f4bc16 use local config for vault when masterless
- **ISSUE** #45915: (MatthiasKuehneEllerhold) 2017.7.3: Salt-SSH & Vault Pillar: Permission denied "minion.pem" (refs: #45928)
- **PR** #45953: (rallytime) Back-port #45928 to 2017.7.3 @ *2018-02-09 22:29:10 UTC*
  - **PR** #45928: (garethgreenaway) [2017.7] Fixing vault when used with pillar over salt-ssh (refs: #45953)
  - 6530649dbc Merge pull request #45953 from rallytime/bp-45928-2017.7.3
  - 85363189d1 Fixing vault when used with pillar over salt-ssh
- **ISSUE** #45893: (CrackerJackMack) archive.extracted ValueError "No path specified" in 2017.7.3 (refs: #45902)
- **PR** #45934: (rallytime) Back-port #45902 to 2017.7.3 @ *2018-02-09 16:31:08 UTC*
  - **PR** #45902: (terminalmage) Check the effective saltenv for cached archive (refs: #45934)
  - fb378cebb0 Merge pull request #45934 from rallytime/bp-45902
  - bb83e8b345 Add regression test for issue 45893
  - cdda66d759 Remove duplicated section in docstring and fix example
  - 4b6351cda6 Check the effective saltenv for cached archive
- **PR** #45935: (rallytime) Back-port #45742 to 2017.7.3 @ *2018-02-09 14:02:26 UTC*
  - **PR** #45742: (marccardinal) list.copy() is not compatible with python 2.7 (refs: #45935)
  - 0d74151c71 Merge pull request #45935 from rallytime/bp-45742
  - 6a0b5f7af3 Removed the chained copy
  - ad1150fad4 list.copy() is not compatible with python 2.7

28.2.25 Salt 2017.7.5 Release Notes

Version 2017.7.5 is a bugfix release for *2017.7.0*.

Statistics

- Total Merges: **213**
- Total Issue References: **58**
- Total PR References: **202**

- Contributors: **52** (Ch3LL, DmitryKuzmenko, GwiYeong, L4rS6, SteffenKockel, The-Loeki, amendlik, andreaspe, angeloudy, aphor, bdrung, cebe, ciiqr, damon-atkins, danlsgiga, ddoh94, dmurphy18, dwoz, eliasp, frogunder, garethgreenaway, gclinch, gtmanfred, jfindlay, kstreee, marccardinal, mcalmer, mchugh19, meaksh, michelsen, nullify005, oarmstrong, oeuftete, philpep, racker-markh, rallytime, redbaron4, roaldnefs, rongshengfang, rongzeng54, rrroo, samilaine, samodid, skizunov, terminalmage, tintoy, twangboy, viktordaniel, vutny, while0pass, whytewolf, zer0def)

Changes to `file.blockreplace` State

The `append_newline` argument was added to this state. Additionally, to improve idempotence, if the string represented by `marker_end` is found in the middle of the line, the content preceding the marker will be removed when the block is replaced. This allows one to remove `append_newline: False` from the SLS and have the block properly replaced if the end of the content block is immediately followed by the `marker_end` (i.e. no newline before the marker).

Note: This will require changes to your SLS if your `marker_end` does not include the very beginning of the content you want to keep.

See the *`file.blockreplace`* state documentation for further information.

Changelog for v2017.7.4..v2017.7.5

Generated at: 2018-05-26 21:50:00 UTC

- **PR** #46612: (Ch3LL) Add changelog to 2017.7.5 release notes @ *2018-03-19 20:47:38 UTC*
  - 19bb725698 Merge pull request #46612 from Ch3LL/7.5_rn
  - 6076bfa2ee Add changelog to 2017.7.5 release
- **PR** #46572: (dmurphy18) Addition of -sa flag to allow for revision numbers other than -0 or -1 @ *2018-03-19 20:07:26 UTC*
  - 31c78aef11 Merge pull request #46572 from dmurphy18/update_xxxbuild
  - c87511570d Merge branch '2017.7.5' into update_xxxbuild
- **ISSUE** saltstack/salt-jenkins#884: (gtmanfred) [2017.7.5][Fedora 27][py2/py3] integration.states.test_npm.NpmStateTest.test_npm_install_url_referenced_package (refs: #46577)
- **PR** #46577: (gtmanfred) Fix npm issue @ *2018-03-19 11:51:04 UTC*
  - cdd768fa4d Merge pull request #46577 from gtmanfred/2017.7.5
  - 78cbf7b5cd Fix npm issue
  - c76f7eb028 enable debug logging on the minionlog
- **PR** #46551: (terminalmage) Fix failing pkg integration test on OpenSUSE @ *2018-03-19 11:50:12 UTC*
  - e6682c660c Merge pull request #46551 from terminalmage/salt-jenkins-885
  - 703b5e7e65 Change versionadded to show that 2018.3.0 will not have this function
  - 010d260d06 Rewrite failing Suse pkg integration test
  - f3f5dec239 zypper.py: fix version argument being ignored
  - 214f2d6ad3 Add pkg.list_repo_pkgs to zypper.py
    - 0a541613f2 Additon of -sa flag to allow for revision numbers other than -0 or -1

- **ISSUE** saltstack/salt-jenkins#886: (gtmanfred) [2017.7.5][Fedora 26/Ubuntu 17.10][py3] integration.states.test_pip.PipStateTest.test_46127_pip_env_vars (refs: #46563)
- **PR** #46563: (gtmanfred) virtualenv version too old for python3.6 @ *2018-03-15 20:17:16 UTC*
  - bd62699ccb Merge pull request #46563 from gtmanfred/2017.7.5
  - 8d5ab72983 virtualenv version too old for python3.6
- **PR** #46561: (gtmanfred) disable verbose @ *2018-03-15 16:36:41 UTC*
  - 2916708124 Merge pull request #46561 from gtmanfred/2017.7.5
  - 2c39ac6dfb disable verbose
- **PR** #46537: (rallytime) Back-port #46529 to 2017.7.5 @ *2018-03-14 14:47:28 UTC*
  - **PR** #46529: (gtmanfred) retry if there is a segfault (refs: #46537)
  - ee3bff6e32 Merge pull request #46537 from rallytime/bp-46529
  - 289c7a228f retry if there is a segfault
- **PR** #46519: (rallytime) Update man pages for 2017.7.5 @ *2018-03-13 20:00:51 UTC*
  - 1271536a89 Merge pull request #46519 from rallytime/man-pages-2017.7.5
  - 782a5584f5 Update man pages for 2017.7.5
- **ISSUE** #46207: (seanjnkns) Issue #44034 still unresolved (refs: #46493)
- **ISSUE** #44034: (seanjnkns) salt-call pillar overrides broken in 2016.11.8 and 2017.7.2 (refs: #44483)
- **PR** #46493: (terminalmage) salt-call: don't re-use initial pillar if CLI overrides passed @ *2018-03-12 20:41:52 UTC*
  - **PR** #44483: (terminalmage) salt-call: account for instances where __pillar__ is empty (refs: #46493)
  - 0e90c8ca6f Merge pull request #46493 from terminalmage/issue46207
  - f06ff68f10 salt-call: don't re-use initial pillar if CLI overrides passed
- **PR** #46450: (gtmanfred) load grains for salt.cmd runner @ *2018-03-12 18:52:22 UTC*
  - b11a8fc8e0 Merge pull request #46450 from gtmanfred/salt_runner
  - 7974ff7264 load grains for salt.cmd runner
- **ISSUE** #30115: (gtmanfred) [BUG] listen does not appear to respect the special names directive (refs: #46337)
- **PR** #46337: (gtmanfred) Fix using names with listen and listen_in @ *2018-03-12 18:50:00 UTC*
  - 22d753364b Merge pull request #46337 from gtmanfred/2017.7
  - d6d9e36359 add tests for names and listen/listen_in
  - 3f8e0db572 let listen_in work with names
  - 7161f4d4df fix listen to be able to handle names
- **PR** #46413: (meaksh) Explore 'module.run' state module output in depth to catch "result" properly @ *2018-03-12 18:49:07 UTC*
  - b7191b8782 Merge pull request #46413 from meaksh/2017.7-explore-result-in-depth
  - 885751634e Add new unit test to check state.apply within module.run
  - 9f19ad5264 Rename and fix recursive method
  - 1476ace558 Fix Python3 and pylint issue

  - 726ca3044d Explore 'module.run' response to catch the 'result' in depth
- **PR** #46496: (gtmanfred) more test kitchen clean up @ *2018-03-12 18:28:34 UTC*
  - 02a79a2014 Merge pull request #46496 from gtmanfred/kitchen
  - da002f78d0 include virtualenv path for py3 windows
  - fe2efe03ea remove duplicate setup
- **ISSUE** #46329: (bdrung) test_create_deployments fails with python-kubernetes 4.0.0 (refs: #46330)
- **PR** #46330: (bdrung) Fix ValueError for template in AppsV1beta1DeploymentSpec @ *2018-03-12 16:56:18 UTC*
  - 5c4c182d75 Merge pull request #46330 from bdrung/fix_kubernetes_test_create_deployments
  - 5008c53c44 Fix ValueError for template in AppsV1beta1DeploymentSpec
- **ISSUE** #46479: (rongshengfang) boto_ec2.instance_present throwing KeyError exception when associating EIP to an existing instance (refs: #46482)
- **PR** #46482: (rongshengfang) Fix KeyError in salt/states/boto_ec2.py @ *2018-03-12 15:13:13 UTC*
  - c7e05d3ff4 Merge pull request #46482 from rongshengfang/fix-keyerror-in-instance_present
  - ed8c83e89a Fix KeyError in salt/states/boto_ec2.py when an EIP is being associated to an existing instance with the instance_present state.
- **PR** #46463: (terminalmage) Update requirements files to depend on mock>=2.0.0 @ *2018-03-09 19:24:41 UTC*
  - 573d51afec Merge pull request #46463 from terminalmage/mock-2.0
  - b958b4699c Update requirements files to depend on mock>=2.0.0
- **ISSUE** #46299: (gclinch) debconf module fails on Python 3 (refs: #46300)
- **PR** #46422: (rallytime) Back-port #46300 to 2017.7 @ *2018-03-09 19:19:25 UTC*
  - **PR** #46300: (gclinch) Python 3 support for debconfmod (fixes #46299) (refs: #46422)
  - a154d35fc7 Merge pull request #46422 from rallytime/bp-46300
  - 829dfde8e8 Change stringutils path to old utils path for 2017.7
  - 91db2e0782 Python 3 support
- **PR** #46320: (mcalmer) add warning about future config option change @ *2018-03-09 17:48:29 UTC*
  - 2afaca17a1 Merge pull request #46320 from mcalmer/warn-kubernetes
  - c493ced415 add warning about future config option change
- **PR** #46449: (bdrung) Make documentation theme configurable @ *2018-03-09 17:47:15 UTC*
  - c7f95581e3 Merge pull request #46449 from bdrung/make-doc-theme-configurable
  - 4a5da2d144 Make documentation theme configurable
- **PR** #46162: (rallytime) Add team-suse to CODEOWNERS file for zypper files @ *2018-03-09 17:46:13 UTC*
  - 10ce0e9e20 Merge pull request #46162 from rallytime/team-suse-zypper-owner
  - 13a295a3b7 Add *pkg* and *snapper* to team-suse
  - 35c7b7b0d3 Add btrfs, xfs, yumpkg, and kubernetes file to team-suse
  - 485d777ac0 Add team-suse to CODEOWNERS file for zypper files
- **PR** #46434: (gtmanfred) split return key value correctly @ *2018-03-09 17:45:21 UTC*
  - cac096b311 Merge pull request #46434 from gtmanfred/highstate_return

  - d18f1a55a7 fix pylint
  - 9e2c3f7991 split return key value correctly
- **ISSUE** #44452: (konstest) salt-cloud can't create snapshots, because there is a bug in the Unicode name of the virtual machine (refs: #46455)
- **PR** #46455: (whytewolf) .format remove fix for #44452 @ *2018-03-09 17:37:19 UTC*
  - 7dd71101ce Merge pull request #46455 from whytewolf/Issue_44452_unicode_cloud
  - 5fe474b1a8 .format remove fix for #44452
- **PR** #46428: (twangboy) Fix issue with dev env install on Windows @ *2018-03-09 14:52:46 UTC*
  - 4c8d9026d3 Merge pull request #46428 from twangboy/win_fix_reqs
  - e7ab97cc17 Remove six as a hard dep for Salt
  - cc67e5c2ef Set six to 1.11.0
- **PR** #46454: (gtmanfred) fix windows for kitchen @ *2018-03-08 21:19:31 UTC*
  - e834d9a63b Merge pull request #46454 from gtmanfred/kitchen
  - b8ab8434a5 fix windows for kitchen
- **ISSUE** #46451: (gmacon) SPM fails to start with customized cache location (refs: #46452)
- **PR** #46452: (gtmanfred) make spm cache_dir instead of all cachedirs @ *2018-03-08 21:12:20 UTC*
  - 2886dca88f Merge pull request #46452 from gtmanfred/spm_cache_dir
  - 169cf7a4e2 make spm cache_dir instead of all cachedirs
- **PR** #46446: (bdrung) Fix various typos @ *2018-03-08 21:11:47 UTC*
  - a188984cd9 Merge pull request #46446 from bdrung/fix-typos
  - 7e6e80be87 heat: Fix spelling mistake of environment
  - a3c54b50f6 Fix various spelling mistakes
- **ISSUE** #20581: (notpeter) Many environments: one pillar_root (all your envs are belong to base) (refs: #46309)
- **PR** #46309: (bdrung) Support dynamic pillar_root environment @ *2018-03-08 19:15:35 UTC*
  - e35fc5263c Merge pull request #46309 from bdrung/dynamic-pillarenv
  - 584b451fd1 Support dynamic pillar_root environment
- **ISSUE** #44032: (PhilippeAB) blockreplace marker_end isn't applied with newline (refs: #46430)
- **PR** #46430: (terminalmage) Improve reliability/idempotence of file.blockreplace state @ *2018-03-08 15:41:38 UTC*
  - 35fe9827fe Merge pull request #46430 from terminalmage/issue44032
  - f9f187e915 Improve reliability/idempotence of file.blockreplace state
- **PR** #46429: (twangboy) Fix problem with __virtual__ in win_snmp @ *2018-03-07 23:26:46 UTC*
  - 2bad0a21c0 Merge pull request #46429 from twangboy/win_fix_snmp
  - 8995a9b8de Fix problem with __virtual__ in win_snmp
- **PR** #46100: (jfindlay) Handle IPv6 scope parameter in resolv.conf @ *2018-03-07 19:51:20 UTC*
  - 93a572f229 Merge pull request #46100 from jfindlay/resolv_scope
  - d5561bedaf tests.unit.grains.core add scoped IPv6 nameserver

  - 4e2e62d508 salt.utils.dns parse scope param for ipv6 servers
- **PR** #46420: (bdrung) Fix SSH client exception if SSH is not found @ *2018-03-07 17:49:00 UTC*
  - 5acc1d5c54 Merge pull request #46420 from bdrung/2017.7
  - e48c13d9e0 Fix SSH client exception if SSH is not found
- **PR** #46379: (angeloudy) TypeError: a bytes-like object is required, not 'str' @ *2018-03-07 15:00:47 UTC*
  - ca6a76e317 Merge pull request #46379 from angeloudy/2017.7
  - 3acb59c74c Merge branch '2017.7' into 2017.7
  - d971e0c08b Fix indent
  - 269514683f Update http.py
  - 908c040ac3 Update http.py
  - 51ba3c135b Update http.py
  - 14aba24111 fix bytes-object required error in python 3
- **PR** #46404: (gtmanfred) get 2017.7 ready to switch over to the new jenkins @ *2018-03-07 14:29:30 UTC*
  - 73f9233557 Merge pull request #46404 from gtmanfred/kitchen
  - c56baa95a8 clone .git for the version tests
  - 3620611b5b fix unhold package for debian
  - 5219f7d2ba fix minion log path
- **ISSUE** #46192: (asymetrixs) salt-log-setup: AttributeError 'NoneType' object has no attribute 'flush' (refs: #46310)
- **PR** #46310: (twangboy) Update the Windows installer build scripts @ *2018-03-06 20:21:58 UTC*
  - ca28cfd4e4 Merge pull request #46310 from twangboy/win_update_installer_build
  - bcf8b19566 Update the installer build
- **PR** #46316: (twangboy) Fix issues with the DSC module @ *2018-03-06 20:16:18 UTC*
  - decccbeca3 Merge pull request #46316 from twangboy/win_fix_dsc
  - 2042d33d59 Fix issues with the DSC module
- **PR** #46394: (Ch3LL) Add mac py2 and py3 packages to mac installation docs @ *2018-03-06 16:45:30 UTC*
  - 95586678c3 Merge pull request #46394 from Ch3LL/mac_doc
  - 158add6661 change oxdownload to oxdownload-{python_version}
  - 21aa848c89 Add mac py2 and py3 packages to mac installation docs
- **ISSUE** #44831: (kivoli) cmd.wait deprecated but cannot replicate conditional execution with onchanges (refs: #46338)
- **PR** #46338: (rallytime) Remove cmd.wait deprecation reference in docs @ *2018-03-05 21:48:52 UTC*
  - 07b5d09ac1 Merge pull request #46338 from rallytime/fix-44831
  - 90771da999 Remove cmd.wait deprecation reference in docs
- **ISSUE** #42438: (ajoaugustine) Failed to send message: hipchat-message (refs: #46333)
- **PR** #46333: (danlsgiga) Fixes color parameter mismatch and handles 204 responses correctly @ *2018-03-05 19:42:26 UTC*

  - 3849e7a085 Merge pull request #46333 from danlsgiga/issue-42438
  - 3b13f37b44 Revert changes in the code and change docs instead
  - 38114a65d8 Fixes color parameter mismatch and handles 204 responses correctly
- **ISSUE** #44935: (grinapo) module.file.replace string seems to be mutated into arrays (refs: #46322)
- **PR** #46322: (terminalmage) yamlify_arg: don't treat leading dashes as lists @ *2018-03-05 15:40:17 UTC*
  - a8f2f1b063 Merge pull request #46322 from terminalmage/issue44935
  - 85ac6a9893 yamlify_arg: don't treat leading dashes as lists
- **PR** #46327: (samilaine) Modify the way a FQDN is handled in the vmware cloud provider. @ *2018-03-05 15:35:37 UTC*
  - da5c282cb2 Merge pull request #46327 from samilaine/fix-vmware-cloud-fqdn
  - 4b8dfb326f Modify the way a FQDN is handled in the vmware cloud provider.
- **PR** #46318: (terminalmage) Skip type-checking for several gitfs/git_pillar/winrepo params @ *2018-03-05 15:04:27 UTC*
  - 78c45d3786 Merge pull request #46318 from terminalmage/squelch-warnings
  - 5889b36646 Skip type-checking for several gitfs/git_pillar/winrepo params
- **ISSUE** #45535: (whytewolf) module_dirs left out salt-ssh, leaving custom ext_pillars and modules out of salt-ssh (refs: #46312)
- **PR** #46312: (gtmanfred) add module_dirs to salt ssh thin tarball @ *2018-03-05 15:00:48 UTC*
  - bb0d6fc263 Merge pull request #46312 from gtmanfred/2017.7
  - 749ae580ed add module_dirs to salt ssh thin tarball
- **ISSUE** #46127: (redbaron4) pip.installed does not pass env_vars when calling freeze to check if package is already installed (refs: #46242)
- **PR** #46242: (redbaron4) Pass env_vars to pip.freeze @ *2018-03-05 14:53:13 UTC*
  - 88b5f7383d Merge pull request #46242 from redbaron4/fix-46127
  - 06dba51617 Make changes from review
  - 727ebe1056 Merge branch '2017.7' into fix-46127
  - 08d1ee8baf Fix Python3 test errors
  - aa9d709015 Pass env_vars to pip.freeze
- **PR** #46265: (Ch3LL) Add username/password to profitbricks conf for cloud tests @ *2018-03-02 21:40:22 UTC*
  - a0716643e4 Merge pull request #46265 from Ch3LL/profit_cloud
  - d4893eab4c Add username/password to profitbricks conf for cloud tests
- **PR** #46306: (rallytime) Back-port #46256 to 2017.7 @ *2018-03-02 21:37:26 UTC*
  - **PR** #46256: (rallytime) Don't install msgpack 0.5.5 (refs: #46306)
  - ed7bffa7e0 Merge pull request #46306 from rallytime/bp-46256
  - 6439bce4a8 Don't install msgpack 0.5.5
- **PR** #46208: (terminalmage) Blacklist os.umask @ *2018-03-02 18:46:07 UTC*
  - 8c2c4e3316 Merge pull request #46208 from terminalmage/audit-umask-usage

  - 9c92aadce8 Disable blacklisted-function check for legitimate uses
  - 58a11aaa26 Disable pylint check in salt-ssh shim
  - ecadf67659 Blacklist os.umask
  - 31b1d98fcb Replace direct use of os.umask with use of existing context manager
  - 82ce546e18 Prevent failed os.makedirs from leaving modified umask in place
- **PR** #46293: (eliasp) Fix Python3 comparison *TypeError* in *salt.modules.upstart @ 2018-03-02 16:36:10 UTC*
  - **PR** #44624: (eliasp) Fix Traceback when using the *service.enabled* state on non-booted systems (refs: #46293)
  - 978e869490 Merge pull request #46293 from eliasp/2017.7-44624-py3-compat
  - 2e08b0d9c8 Fix Python3 comparison *TypeError* in *salt.modules.upstart*
- **ISSUE** #46128: (Boulet-) Mountpoint in git_pillar (refs: #46264)
- **PR** #46264: (terminalmage) Fix incorrect merge conflict resolution @ *2018-03-02 14:21:13 UTC*
  - bee4a66d0c Merge pull request #46264 from terminalmage/issue46128
  - 68000b7211 Fix incorrect merge conflict resolution
- **PR** #46296: (vutny) [DOC] Add missing params to *pillar.get* docstring @ *2018-03-02 14:19:41 UTC*
  - 1e0b3aa348 Merge pull request #46296 from vutny/doc-pillar-get
  - 1faa8331e1 [DOC] Add missing params to *pillar.get* docstring
- **PR** #45874: (GwiYeong) fix for local client timeout bug @ *2018-03-01 19:39:35 UTC*
  - c490a50452 Merge pull request #45874 from GwiYeong/2017.7-local-client-hotfix
  - 949aefc82b Merge branch '2017.7' into 2017.7-local-client-hotfix
  - 45d663f435 fix for local client timeout bug
- **PR** #46261: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2018-03-01 17:55:23 UTC*
  - 8e8a3a2897 Merge pull request #46261 from rallytime/merge-2017.7
  - 8256ae5ee5 Merge branch '2016.11' into '2017.7'
    - 140ef4d6b9 Merge pull request #46253 from rallytime/doc-banners
      - 07ed8c7db3 Update docbanner for SaltConf18
    - 9fe86ee520 Merge pull request #46179 from wedge-jarrad/cifs-remount-fix
      - 9ca25c4313 Add credentials and secretfile to mount.mounted mount_invisible_keys
- **ISSUE** #44046: (t2b) docker_container.running states fail if the argument ulimits is set and a watch requisite is triggered (refs: #46276)
- **PR** #46276: (terminalmage) salt.utils.docker.translate_input: operate on deepcopy of kwargs @ *2018-03-01 15:37:44 UTC*
  - 88a3166589 Merge pull request #46276 from terminalmage/issue44046
  - a14d4daf8c salt.utils.docker.translate_input: operate on deepcopy of kwargs
- **ISSUE** #46182: (oeuftete) docker_container.running is sensitive to HostConfig Ulimits ordering (refs: #46183)
- **PR** #46183: (oeuftete) Fix docker_container.running HostConfig Ulimits comparison @ *2018-02-28 22:22:11 UTC*

  - da60399b8f Merge pull request #46183 from oeuftete/fix-docker-container-running-host-config-ulimits
  - 5b09644429 Sort lists from Ulimits before comparing
  - 0b80f02226 Update old dockerng doc ref
- **ISSUE** #46259: (terminalmage) git_pillar_branch overrides branch defined in git_pillar configuration (refs: #46260)
- **ISSUE** #46258: (terminalmage) git_pillar_base doesn't work for values when PyYAML loads them as int/float (refs: #46260)
- **PR** #46260: (terminalmage) Normalize global git_pillar/winrepo config items @ *2018-02-28 22:05:26 UTC*
  - 509429f08c Merge pull request #46260 from terminalmage/git_pillar
  - b1ce2501fd Normalize global git_pillar/winrepo config items
- **PR** #46101: (jfindlay) In OpenRC exec module, make sure to ignore retcode on status @ *2018-02-28 20:01:37 UTC*
  - a97a3e6fb0 Merge pull request #46101 from jfindlay/openrc_ret
  - 2eef3c65a6 tests.unit.modules.gentoo_service add retcode arg
  - 81ec66fd8b modules.gentoo_service handle stopped retcode
- **PR** #46254: (rallytime) Update enterprise banner @ *2018-02-28 19:54:03 UTC*
  - 1a17593c05 Merge pull request #46254 from rallytime/enterprise-banner
  - f5fae3dedf Update enterprise banner
- **PR** #46250: (terminalmage) Add documentation to the fileserver runner @ *2018-02-28 18:53:49 UTC*
  - 8c50ff32bd Merge pull request #46250 from terminalmage/runner-docs
  - 91b4895087 Add documentation to the fileserver runner
- **ISSUE** #46215: (racker-markh) salt-cloud will only intermittently build rackspace cloud instances with purely private networks (refs: #46243)
- **PR** #46243: (racker-markh) Don't ignore 'private_ips' unnecessarily @ *2018-02-28 15:28:29 UTC*
  - 53067cca43 Merge pull request #46243 from racker-markh/fix-openstack-private-network-issue
  - 50c1e140f0 Don't check deny private_ips already in the original list of private_ips
- **ISSUE** #46109: (rombert) archive.extracted takes a long time (> 4 minutes) even though directory exists (refs: #46239)
- **PR** #46239: (terminalmage) archive.extracted: don't check source file when if_missing path exists @ *2018-02-28 15:01:36 UTC*
  - 15405c8760 Merge pull request #46239 from terminalmage/issue46109
  - 586d8b0dcf archive.extracted: don't check source file when if_missing path exists
- **PR** #46221: (terminalmage) Fix hanging tests in integration suite @ *2018-02-27 21:32:25 UTC*
  - 633e1208e4 Merge pull request #46221 from terminalmage/salt-jenkins-854
  - 0eb012659c Fix hanging tests in integration suite
- **PR** #46214: (vutny) [DOC] Replace *note* rST block for GitHub @ *2018-02-27 17:42:37 UTC*
  - 7917277345 Merge pull request #46214 from vutny/formulas-readme-formatting
  - d702846961 [DOC] Replace *note* rST block for GitHub

- **PR** #46203: (Ch3LL) Add 2017.7.5 Release Notes File @ *2018-02-26 21:17:48 UTC*
  - a2e099b744 Merge pull request #46203 from Ch3LL/7.5_release
  - 6ddf3246ce Add 2017.7.5 Release Notes File
- **PR** #46201: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2018-02-26 18:56:47 UTC*
  - 973b227818 Merge pull request #46201 from rallytime/merge-2017.7
  - 9ac2101baa Merge branch '2016.11' into '2017.7'
  - a4c5417d23 Merge pull request #46132 from rallytime/2016.11_update_version_doc
    - d2196b6df3 Update release versions for the 2016.11 branch
- **ISSUE** #34423: (bdrung) oscodename wrong on Debian 8 (jessie) (refs: #46139)
- **PR** #46139: (bdrung) Add os grains test cases for Debian/Ubuntu and fix oscodename on Ubuntu @ *2018-02-26 16:44:04 UTC*
  - 89cf2e5061 Merge pull request #46139 from bdrung/os-grains
  - 0b445f2a37 tests: Add unit tests for _parse_os_release()
  - f6069b77ed Fix osfinger grain on Debian
  - 8dde55a761 tests: Add os_grains test cases for Debian
  - ff02ab9937 tests: Add Ubuntu 17.10 (artful) os_grains test case
  - 77d5356aba Fix incorrect oscodename grain on Ubuntu
  - 7e62dc9fd2 tests: Support reading os-release files from disk
  - a92ec0db1b Make _parse_os_release() always callable
  - eee1fe5b38 tests: Dissolve _run_ubuntu_os_grains_tests
  - 1d6ef731fe tests: Deduplicate _run_os_grains_tests()
- **PR** #46133: (rallytime) Update release versions for the 2017.7 branch @ *2018-02-26 16:42:43 UTC*
  - c8c71e75ca Merge pull request #46133 from rallytime/2017.7_update_version_doc
  - 0ed338e643 Update release versions for the 2017.7 branch
- **ISSUE** #46124: (moremo) GitFS saltenv ref won't pick up multiple of the same ref (refs: #46185)
- **PR** #46185: (terminalmage) gitfs: Fix detection of base env when its ref is also mapped to a different env @ *2018-02-26 14:52:16 UTC*
  - 390d592aa6 Merge pull request #46185 from terminalmage/issue46124
  - 3b58dd0da0 gitfs: Fix detection of base env when its ref is also mapped to a different env
- **PR** #46148: (rallytime) [2017.7] Merge forward from 2017.7.3 to 2017.7 @ *2018-02-23 19:21:38 UTC*
  - 705caa8cca Merge pull request #46148 from rallytime/merge-2017.7
  - 25deebf7a6 Merge branch '2017.7.3' into '2017.7'
- **PR** #46137: (damon-atkins) [2017.7] update ec2 pillar arguments with better names @ *2018-02-23 13:32:04 UTC*
  - **PR** #45878: (damon-atkins) ec2_pillar update to fix finding instance-id (refs: #46137)
  - 10a47dcbc4 Merge pull request #46137 from damon-atkins/2017.7_fix_ec2_pillar2
  - 99e7f6a7d3 update ec2 pillar arguments with better names

- **ISSUE** #46004: (github-abcde) opts file_roots gets overwritten with pillar_roots in orchestration run (refs: #46145)
- **PR** #46145: (terminalmage) 3 small fixes for runners/orchestration @ *2018-02-22 22:11:11 UTC*
  - d74cb14557 Merge pull request #46145 from terminalmage/issue46004
  - 467ff841cd pillarenv argument should default to None and not the value from opts
  - 2a185855ea Better solution for fixing the opts munging in pillar.show_pillar runner
  - e2c4702e0c Update tests to reflect changes to the SaltCacheLoader
  - f9301fcc34 Document behavior when orchestration runnner invoked with non-orch states
  - 9644579cd0 Instantiate the SaltCacheLoader's fileclient in the __init__
  - f9a6c86e21 salt.runners.pillar.show_pillar: don't modify master opts
  - e0940a9fc4 Properly detect use of the state.orch alias and add orch jid to kwargs
- **PR** #46135: (rallytime) Back-port #46088 to 2017.7 @ *2018-02-22 15:11:14 UTC*
  - **PR** #46088: (rongzeng54) fix kernel subpackages install bug (refs: #46135)
  - 0398ce0482 Merge pull request #46135 from rallytime/bp-46088
  - 57a60f62a3 fix kernel subpackages install bug
- **ISSUE** #45837: (johje349) Salt Cloud does not recognise all Digitalocean sizes (refs: #46115)
- **PR** #46136: (rallytime) Back-port #46115 to 2017.7 @ *2018-02-21 19:17:23 UTC*
  - **PR** #46115: (samodid) update digitalocean salt-cloud driver (refs: #46136)
  - 1fcbbd1e02 Merge pull request #46136 from rallytime/bp-46115
  - 0a481d707f update digitalocean salt-cloud driver
- **PR** #45911: (twangboy) LGPO Module: Convert reg values to unicode for debug @ *2018-02-21 19:02:17 UTC*
  - 11e5e8eb86 Merge pull request #45911 from twangboy/win_fix_lgpo_unicode
  - bcde5cc625 Update log statement
  - e9fa53d3b7 Change the Invalid Data Message
  - c818d4b791 Convert reg values to unicode for debug
- **ISSUE** #46085: (zmedico) 2017.7.3 salt master with "open_mode: True" becomes unresponsive if minion submits empty public key (refs: #46123)
- **PR** #46123: (gtmanfred) If no pubkey is passed in openmode fail @ *2018-02-21 19:01:47 UTC*
  - 524a6a72a0 Merge pull request #46123 from gtmanfred/2017.7
  - 8d36730ef7 If no pubkey is passed in openmode fail
- **PR** #46131: (vutny) [DOC] Fix code-blocks for reStructuredText @ *2018-02-21 15:47:05 UTC*
  - e48fa58012 Merge pull request #46131 from vutny/doc-formula-formatting
  - d8fb051e44 [DOC] Fix code-blocks for reStructuredText
- **ISSUE** #42763: (xuhcc) acme.cert state falsely reports about renewed certificate (refs: #44603)
- **ISSUE** #40208: (bewing) Inconsistent state return when test=True (refs: #44603)
- **PR** #46118: (rallytime) Back-port #44603 to 2017.7 @ *2018-02-21 15:21:42 UTC*
  - **PR** #44603: (oarmstrong) Fix acme state to correctly return on test (refs: #46118)

  - 6cea44ee95 Merge pull request #46118 from rallytime/bp-44603
  - 2a2c23c66b Fix acme state to correctly return on test
- **PR** #46121: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2018-02-21 10:07:18 UTC*
  - 16c382b55b Merge pull request #46121 from rallytime/merge-2017.7
  - 4c2f504a85 Merge branch '2016.11' into '2017.7'
    - e197a0fbc5 Merge pull request #46076 from rallytime/bp-46066
      - b94d73c53e Pin tornado version in requirements file
    - c72c1bde5f Merge pull request #46093 from wedge-jarrad/contributing-doc-typo
      - 5a0fe104f7 Fix contributing doc typo
    - 3cb83ea87e Merge pull request #45992 from bgridley/fix-routes-present-state
      - 679787699c Add vpc_peering_connection_id to describe_route_tables route_keys
    - 8a60635da0 Merge pull request #46000 from terminalmage/issue45910
      - 8cf13325ee salt.states.reg.present: Prevent traceback when reg data is binary
    - 1f44e285dc Merge pull request #46011 from terminalmage/fix-solaris-runas
      - 8ee0a3a28b Move Solaris USER workaround up a bit
      - 13cdb52690 cmdmod.py: runas workaround for platforms that don't set a USER env var
    - 30fb8f7be0 Merge pull request #45467 from twangboy/win_exclude_hidden
      - ea41215646 Make the regex pattern less greedy
      - 6d223cffa7 Add tip about passing bogus saltenv
      - 1282ae3a93 Skip hidden first
      - 437a457911 Skip hidden dirs in genrepo
      - 87dc554dc3 Add final updates to docs
      - 3646d5c897 Fix some docs formatting, add some warnings
      - 35c81faf5a Log the source_dir when caching the files
      - 91c3da8dfd Improve docs for pkg.refresh_db
      - 4803d92707 Add some documentation
      - 08b82e0875 Fix lint error, use raw
      - 2f712691cf Exclude hidden directories in pkg.refresh_db
- **ISSUE** #46106: (amendlik) yumpkg.refresh_db hangs (refs: #46107)
- **PR** #46107: (amendlik) Add –assumeyes on YUM/DNF commands @ *2018-02-20 22:52:06 UTC*
  - b92346645b Merge pull request #46107 from amendlik/yumpkg-assumeyes
  - 8d9a432fb2 Add –assumeyes to yum/dnf commands in yumpkg.refresh_db
- **PR** #46094: (kstreee) Fix memory leak @ *2018-02-20 21:36:02 UTC*
  - 14fe423e0c Merge pull request #46094 from kstreee/fix-memory-leak
  - 48080a1bae Fixes memory leak, saltclients should be cleaned after used.
  - aba00805f4 Adds set_close_callback function to removes stream instance after closed from a set streams.

- **ISSUE** #13: (thatch45) Expand the stats module (refs: #46097)
- **PR** #46097: (vutny) [DOC] Put https link to the formulas doc page @ *2018-02-20 17:07:39 UTC*
  - 320c2037e1 Merge pull request #46097 from vutny/fix-https-link
  - 2062fd0e5c [DOC] Put https link to the formulas doc page
- **PR** #46103: (bdrung) Fix skipping Kubernetes tests if client is not installed @ *2018-02-20 16:33:42 UTC*
  - 0eb137fb4e Merge pull request #46103 from bdrung/2017.7
  - dd3f936557 Fix skipping Kubernetes tests if client is not installed
- **PR** #46070: (Ch3LL) add required arg to dns_check jinja doc example @ *2018-02-16 20:00:44 UTC*
  - c3a938e994 Merge pull request #46070 from Ch3LL/fix-doc-dns
  - 2a5d855d97 add required arg to dns_check jinja doc example
- **PR** #46067: (rallytime) Back-port #45994 to 2017.7 @ *2018-02-16 19:55:27 UTC*
  - **PR** #45994: (nullify005) Fix hosted zone Comment updates & quote TXT entries correctly (refs: #46067)
  - 01042e9d77 Merge pull request #46067 from rallytime/bp-45994
  - a07bb48726 Correct formatting for lint
  - e8678f633d Fix Comment being None not ” and inject quotes into the TXT ChangeRecords
- **ISSUE** #42932: (bobrik) cmd.run with bg: true doesn’t fail properly (refs: #45932)
- **PR** #45932: (The-Loeki) Fix cmd run_all bg error @ *2018-02-16 14:53:15 UTC*
  - **PR** #39980: (vutny) [2016.3] Allow to use *bg* kwarg for *cmd.run* state function (refs: #45932)
  - 5e0e2a30e2 Merge pull request #45932 from The-Loeki/fix_cmd_run_all_bg
  - f83da27ca5 Merge branch ’2017.7’ into fix_cmd_run_all_bg
  - 771758fbca Merge branch ’2017.7’ into fix_cmd_run_all_bg
  - c54fcf7a2d cmd: move separate DRY logging blocks into _run, prevent logging on bg=True, don’t use_vt on bg
  - ebb1f81a9b cmd run: when running in bg, force ignore_retcode=True
- **PR** #46062: (vutny) Fix typo in postgres_user.present state function @ *2018-02-16 14:44:29 UTC*
  - 45ace39961 Merge pull request #46062 from vutny/pg-user-state-fix-typo
  - a5fbe4e95e Fix typo in postgres_user.present state function
- **PR** #45763: (twangboy) Fix rehash function in win_path.py @ *2018-02-15 20:05:16 UTC*
  - edcb64de76 Merge pull request #45763 from twangboy/win_fix_path_rehash
  - b9a2bc7b29 Fix hyperlinks
  - 29912adc15 Move the test_rehash test to test_win_functions
  - adc594c183 Remove duplicate link
  - e84628c1eb Add some comments to the code
  - d50d5f582f Add additional info to docs for *broadcast_setting_change*
  - 3a54e09cd9 Rename setting to message
  - a3f9e99bc0 Change to a generic function to broadcast change

  - 79299361c3 Create refresh_environment salt util
  - 967b83940c Fix rehash function
- **PR** #46042: (jfindlay) Revise file_tree pillar module documentation @ *2018-02-15 19:29:52 UTC*
  - **PR** #46027: (jfindlay) Revise file_tree pillar module documentation (refs: #46042)
  - a46fbc546c Merge pull request #46042 from jfindlay/file_tree_doc
  - 0ba4954a4b salt.pillar.file_tree revise module documentation
  - 3c6a5bf967 salt.pillar.file_tree provide better debug info
  - bb1cdc451e salt.pillar.file_tree no stack trace when nodegroups undefined
- **PR** #46013: (rallytime) Back-port #45598 to 2017.7 @ *2018-02-15 16:11:05 UTC*
  - **PR** #45598: (nullify005) Patch around ResourceRecords needing to be present for AliasTarget (refs: #46013)
  - de86126dd8 Merge pull request #46013 from rallytime/bp-45598
  - 2ea3fef543 No lazy logging
  - f427b0febc Change formatting style of logging lines per review
  - ebb244396b Patch around ResourceRecords needing to be present for AliasTarget entries to work
- **ISSUE** #45825: (philpep) selinux.fcontext_policy_present doesn't work on Centos 6 with filetype = all files (refs: #45826)
- **PR** #46016: (rallytime) Back-port #45826 to 2017.7 @ *2018-02-14 18:16:24 UTC*
  - **PR** #45826: (philpep) Fix selinux.fcontext_policy_present for Centos 6 (refs: #46016)
  - 07e5735471 Merge pull request #46016 from rallytime/bp-45826
  - 1916e5c4a4 Fix selinux.fcontext_policy_present for Centos 6
- **ISSUE** #45784: (oarmstrong) SELinux module fcontext_get_policy fails with long regex (refs: #45785)
- **PR** #46015: (rallytime) Back-port #45785 to 2017.7 @ *2018-02-14 18:16:09 UTC*
  - **PR** #45785: (oarmstrong) m/selinux.fcontext_get_policy allow long filespecs (refs: #46015)
  - a1f4092811 Merge pull request #46015 from rallytime/bp-45785
  - ef6ffb1492 Resolve linting errors
  - 8047066c46 Remove unused import
  - 8f7c45935a Add tests for salt.modules.selinux.fcontext_get_policy
  - bafb7b4e6e Ensure parsed fields are stripped
  - a830a6e819 m/selinux.fcontext_get_policy allow long filespecs
- **PR** #46012: (rallytime) Back-port #45462 to 2017.7 @ *2018-02-14 18:14:56 UTC*
  - **PR** #45462: (aphor) emit port cli version, variants as separate args (refs: #46012)
  - 96097c037e Merge pull request #46012 from rallytime/bp-45462
  - 9f76836a6c emit port cli version, variants as separate args
- **PR** #45991: (terminalmage) yumpkg: Fix a couple issues with _get_extra_opts @ *2018-02-14 16:48:28 UTC*
  - 1279924f5f Merge pull request #45991 from terminalmage/fix-duplicate-extra-opts
  - 916766f651 yumpkg: Fix a couple issues with _get_extra_opts

- **PR** #46017: (rallytime) [2017.7] Merge forward from 2017.7.3 to 2017.7 @ *2018-02-13 21:43:15 UTC*
  - 8b9adc258e Merge pull request #46017 from rallytime/merge-2017.7
  - a06645ce71 Merge branch '2017.7.3' into '2017.7'
- **ISSUE** #45796: (L4rS6) aliases module doesn't follow symlinks (refs: #45797)
- **PR** #45988: (rallytime) Back-port #45797 to 2017.7 @ *2018-02-13 17:49:02 UTC*
  - **PR** #45797: (L4rS6) follow symlinks in aliases module (close #45796) (refs: #45988)
  - d20ff89414 Merge pull request #45988 from rallytime/bp-45797
  - 953a400d79 follow symlinks
- **PR** #45711: (bdrung) Fix Unicode tests when run with LC_ALL=POSIX @ *2018-02-13 17:42:07 UTC*
  - b18087cee0 Merge pull request #45711 from bdrung/fix-unicode-tests
  - b6181b5ed6 Fix Unicode tests when run with LC_ALL=POSIX
- **PR** #45878: (damon-atkins) ec2_pillar update to fix finding instance-id (refs: #46137) @ *2018-02-13 17:34:14 UTC*
  - 5271fb1d40 Merge pull request #45878 from damon-atkins/2017.7_fix_ec2_pillar
  - 0e74025714 Merge branch '2017.7' into 2017.7_fix_ec2_pillar
  - b4d0b23891 py3 fix
  - 75d9e20d8a Add ignoring 'terminated', 'stopped' instances, to improve changes of a single match
  - 0093472a37 added tag_key_list and tag_key_sep to create ec2_tags_list
  - afb3968aa7 ec2_pillar could not find instance-id, resolved. add support to use any tag to compare minion id against.
- **PR** #45942: (terminalmage) Fix incorrect translation of docker port_bindings -> ports (2017.7 branch) @ *2018-02-13 16:10:03 UTC*
  - cf367dbd04 Merge pull request #45942 from terminalmage/issue45679-2017.7
  - 89cbd72a0d Don't try to sort ports when translating docker input
  - 9cd47b39dd Fix incorrect translation of docker port_bindings -> ports
- **PR** #45959: (rallytime) A couple of grammar updates for the state compiler docs @ *2018-02-12 22:17:49 UTC*
  - dae41de7a8 Merge pull request #45959 from rallytime/state-doc-update
  - 6f781cb95d A couple of grammar updates for the state compiler docs
- **ISSUE** saltstack/salt#45884: (tintoy) "TypeError: can't serialize <NodeImage" when calling salt-cloud with the dimensiondata driver (refs: #45908)
- **ISSUE** #45884: (tintoy) "TypeError: can't serialize <NodeImage" when calling salt-cloud with the dimensiondata driver (refs: #45908)
- **PR** #45908: (tintoy) Fix for #45884 ("TypeError: can't serialize <NodeImage" when calling salt-cloud with the dimensiondata driver) @ *2018-02-12 22:05:29 UTC*
  - 007214f7bf Merge pull request #45908 from DimensionDataResearch/fix/issue/45884
  - 1a75786b5a Fix linter warnings.
  - 82ec0b589c Revert to using salt.utils.cloud.is_public_ip.
  - 9b6b01873b Fix violations reported by flake8.

  - a2bc155c73 Use __utils__['cloud.'] instead of salt.cloud.utils.
  - 98907a32cb Ensure 'auth' parameter is correctly passed to dimensiondata driver.
  - de26b03e2c Fix copy/paste bug in dimensiondata provider integration test.
  - 6b1b6be427 Add integration tests for dimensiondata cloud provider.
  - f6ea9fed7d Ensure that event data provided by the dimensiondata driver is serialisable.
- **PR** #45985: (garethgreenaway) [2017.7] Backport #45894 - Missing *format* in the call to write. @ *2018-02-12 20:22:31 UTC*
  - **PR** #45894: (while0pass) Fix inconcistencies in param description (refs: #45985)
  - efcbfa868c Merge pull request #45985 from garethgreenaway/2017_7_fixing_mac_tests_again
  - 7b8dc14433 Missing *format* in the call to write.
- **PR** #45958: (garethgreenaway) Backporting #45935 to 2017.7 @ *2018-02-12 16:25:07 UTC*
  - **PR** #45936: (garethgreenaway) [oxygen] Fix to log/handlers/__init__.py (refs: #45958)
  - **PR** #45935: (rallytime) Back-port #45742 to 2017.7.3 (refs: #45958)
  - **PR** #45742: (marccardinal) list.copy() is not compatible with python 2.7 (refs: #45935)
  - bf03abd07c Merge pull request #45958 from garethgreenaway/backport-fixing_mactests_queue_full
  - 25dffaae91 Backporting #45935
- **PR** #45949: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2018-02-09 22:32:09 UTC*
  - bab365d6c6 Merge pull request #45949 from rallytime/merge-2017.7
  - f51687e903 Merge branch '2016.11' into '2017.7'
  - 7779fea7ba Merge pull request #45940 from dmurphy18/fix_aix_cmdmod
    - dd2788419b Fix use of 'su' for AIX to use '-'
- **ISSUE** #45915: (MatthiasKuehneEllerhold) 2017.7.3: Salt-SSH & Vault Pillar: Permission denied "minion.pem" (refs: #45928)
- **PR** #45928: (garethgreenaway) [2017.7] Fixing vault when used with pillar over salt-ssh @ *2018-02-09 16:32:35 UTC*
  - 7fd00ec752 Merge pull request #45928 from garethgreenaway/45915_fixing_vault_pillar_for_salt_ssh
  - 259e60e5d4 Fixing vault when used with pillar over salt-ssh
- **PR** #45925: (terminalmage) Fix spelling error in docstring @ *2018-02-08 21:52:35 UTC*
  - 9d14ad9ccf Merge pull request #45925 from terminalmage/fix-spelling
  - 7a143fe454 Fix spelling error in docstring
- **PR** #45920: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2018-02-08 15:43:49 UTC*
  - 0cbe93cd69 Merge pull request #45920 from rallytime/merge-2017.7
  - e4e4744218 Merge branch '2016.11' into '2017.7'
  - 27ff82f996 Merge pull request #45864 from rallytime/release-note-fix
    - 104a24f244 Remove extraneous] in release notes for 2016.11.9
  - 5fa010de2b Merge pull request #45787 from rallytime/2016.11.9_docs
    - a38d4d44fa [2016.11] Bump latest and previous versions

- **ISSUE** #45805: (bgridley) Execution module victorops throws an error "RuntimeError: dictionary changed size during iteration" (refs: #45814)
- **PR** #45814: (gtmanfred) fix cookies dict size changing in http.query @ *2018-02-08 15:35:30 UTC*
  - 643a8a5278 Merge pull request #45814 from gtmanfred/2017.7
  - d8eec9aa97 fix cookies dict size changing in http.query
- **PR** #45877: (rallytime) Add release notes file for 2017.7.4 release @ *2018-02-08 14:07:43 UTC*
  - 3a3f87c16d Merge pull request #45877 from rallytime/new-release-notes
  - f937e8ba81 Add release notes file for 2017.7.4 release
- **PR** #45904: (rallytime) Back-port #41017 to 2017.7 @ *2018-02-08 13:57:45 UTC*
  - **PR** #41017: (cebe) Fixed typo in pkg state documentation (refs: #45904)
  - 1c3cc00670 Merge pull request #45904 from rallytime/bp-41017
  - 80c56cdcea Fixed typo in pkg state documentation
- **PR** #45907: (terminalmage) Fix backport of grains fix @ *2018-02-08 13:57:26 UTC*
  - 317d35bd15 Merge pull request #45907 from terminalmage/fix-grains-backport
  - 6cf7e50cc4 Fix backport of grains fix
- **PR** #45906: (rallytime) Back-port #45548 to 2017.7 @ *2018-02-08 13:57:07 UTC*
  - **PR** #45548: (viktordaniel) Update x509.py - documentation fix (refs: #45906)
  - dade5f0cab Merge pull request #45906 from rallytime/bp-45548
  - 1befa7386c Update x509.py
- **ISSUE** #45893: (CrackerJackMack) archive.extracted ValueError "No path specified" in 2017.7.3 (refs: #45902)
- **PR** #45902: (terminalmage) Check the effective saltenv for cached archive @ *2018-02-08 13:42:00 UTC*
  - 82c473a1fe Merge pull request #45902 from terminalmage/issue45893
  - 9d200efc26 Add regression test for issue 45893
  - 1468f1d0ff Remove duplicated section in docstring and fix example
  - 6cc5cd9b8a Check the effective saltenv for cached archive
- **PR** #45862: (rallytime) Back-port #45830 to 2017.7 @ *2018-02-08 13:22:26 UTC*
  - **PR** #45830: (garethgreenaway) [oxygen] Catch exception when logging queue is full (refs: #45862)
  - fdedde3cfb Merge pull request #45862 from rallytime/bp-45830
  - 1024856f9a Wrapping the put_nowait in a try…except and catching the exception when the multiprocessing queue is full. This situation is happening when running the full testing suite on MacOS where the queue limit is 32767 vs on Linux where the queue limit is unlimited.
- **PR** #45779: (The-Loeki) SSH shell shim: Don't use $() for optimal support @ *2018-02-05 18:35:21 UTC*
  - 43a45b42c3 Merge pull request #45779 from The-Loeki/patch-3
  - 8575ae3d52 Merge branch '2017.7' into patch-3
  - 47cf00d88e SSH shell shim: Don't use $() for optimal support
- **PR** #45788: (rallytime) [2017.7] Bump latest and previous versions @ *2018-02-05 15:30:46 UTC*
  - cca997d0da Merge pull request #45788 from rallytime/2017.7.3_docs

  - d5faf6126b [2017.7] Bump latest and previous versions
- **PR** #45842: (rallytime) Back-port #45827 to 2017.7 @ *2018-02-05 15:04:11 UTC*
  - **PR** #45827: (terminalmage) Fix traceback in disks grains when /sys/block not available (refs: #45842)
  - 746206cebe Merge pull request #45842 from rallytime/bp-45827
  - c631598a87 Fix traceback in disks grains when /sys/block not available
- **ISSUE** #44978: (doesitblend) State duration not always calculated (refs: #45721)
- **PR** #45721: (garethgreenaway) [2017.7] Ensure duration and start time exist @ *2018-02-05 14:59:33 UTC*
  - 900aadcd67 Merge pull request #45721 from garethgreenaway/44978_show_duration_when_no_state_run
  - 359265869f Adding a couple tests to ensure that duration is included in state run results even when states do not run.
  - 912347abc3 Include the duration when a state does not run, for example when the *onchanges* requisite is not met.
- **PR** #45517: (kstreee) Fixes base dir making logic to ensure not raising the exception when base directory already exists. @ *2018-02-05 14:56:23 UTC*
  - 80a2d009b4 Merge pull request #45517 from kstreee/fix-mkdir
  - 24d41f2451 Fixes base dir making logic to ensure not raising the exception when base directory already exists.
- **PR** #45835: (kstreee) Adds a missing return statement. @ *2018-02-02 22:51:41 UTC*
  - 7a4b1b2e77 Merge pull request #45835 from kstreee/fix-missing-return-statement
  - 68c7f3dcba Adds a missing return statement.
- **PR** #45840: (rallytime) Back-port #45603 to 2017.7 @ *2018-02-02 20:17:32 UTC*
  - **PR** #45603: (andreaspe) Fix for duplicate entries with pkrepo.managed (refs: #45840)
  - 0a04f118c2 Merge pull request #45840 from rallytime/bp-45603
  - 9653363131 Fix for duplicate entries with pkrepo.managed
- **ISSUE** #44315: (whytewolf) cmd.* cwd does not escape spaces. 2017.7.2 (refs: #45134)
- **PR** #45716: (ciiqr) fixed quoting of script path in cmd.script @ *2018-02-02 14:36:49 UTC*
  - **PR** #45134: (garethgreenaway) [2017.7] fix to cmd.script for cwd with space (refs: #45716)
  - bd2178cd5f Merge pull request #45716 from ciiqr/fix_cmd_script_quoting
  - 217791079b some code cleanup (lint errors and escape_argument as _cmd_quote)
  - 1c29bc5a3d fixed quoting of script path in cmd.script
- **ISSUE** #45684: (bdrung) salt documentation fails to build with Python 3 version of sphinx (refs: #45719)
- **PR** #45719: (bdrung) Fix python3 sphinx build @ *2018-02-02 14:20:37 UTC*
  - 272f912c7c Merge pull request #45719 from bdrung/fix-python3-sphinx-build
  - 179e8fbe73 doc: Do not mock non-existing __qualname__ attribute
  - 971e59ebe2 Drop enforcing new-style object for SaltYamlSafeLoader
- **PR** #45764: (mchugh19) support amazon linux 2 for service module @ *2018-02-02 14:12:07 UTC*
  - **PR** #45758: (mchugh19) support amazon linux 2 for service module (refs: #45764)

  - fc04336c3b Merge pull request #45764 from mchugh19/2017.7
  - 0a7f1a4d75 English better
  - 37e067c7b5 support amazon linux 2 for service module
- **PR** #45756: (roaldnefs) Fix Grafana4 states documentation @ *2018-01-31 19:01:33 UTC*
  - f234bf52f4 Merge pull request #45756 from roaldnefs/fix-grafana4-documentation
  - 92979c0b57 Fix grafana4 states documentation
- **PR** #45801: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2018-01-31 18:55:52 UTC*
  - 685b683db5 Merge pull request #45801 from rallytime/merge-2017.7
  - 26e992e011 Merge branch '2016.11' into '2017.7'
    - 746386d04c Merge pull request #45794 from vutny/doc-file-state-examples
      - ddfeae6a29 [DOC] Fix code-block rST directive in file state module
    - abc9ece214 Merge pull request #45780 from vutny/doc-pkgrepo-zypper
      - f80c7d8d69 [DOC] Add missing gpgautoimport for pkgrepo.managed
- **PR** #45802: (rallytime) [2017.7] Merge forward from 2017.7.3 to 2017.7 @ *2018-01-31 18:55:35 UTC*
  - c7d319f3bc Merge pull request #45802 from rallytime/merge-2017.7-from-2017.7.3
  - eb48513ba0 Merge branch '2017.7.3' into '2017.7'
- **ISSUE** #45738: (UtahCampusD) minion cache overwritten for scheduled jobs (refs: #45761)
- **PR** #45761: (gtmanfred) generate a jid for cache_jobs on the minion @ *2018-01-31 18:01:53 UTC*
  - 96e9232cc2 Merge pull request #45761 from gtmanfred/2017.7
  - 280767ed57 generate a jid for cache_jobs on the minion
- **ISSUE** #45301: (twangboy) Ctl+C causes stack trace on Windows (refs: #45707)
- **PR** #45707: (skizunov) Fix exception when shutting down logging listener @ *2018-01-30 13:28:10 UTC*
  - 38ed46a61a Merge pull request #45707 from skizunov/develop2
  - e84801a381 Ensure we have at least one logging root handler
  - 3da9b8dd33 Fix exception when shutting down logging listener
- **PR** #45773: (terminalmage) Fix misspellings @ *2018-01-30 13:24:52 UTC*
  - 53008ffec7 Merge pull request #45773 from terminalmage/fix-misspelling
  - 0a45f998fe Fix misspellings
- **ISSUE** #45489: (ipmb) cache.grains runner returns all minions when match is not found (refs: #45588)
- **PR** #45751: (rallytime) Back-port #45588 to 2017.7 @ *2018-01-29 17:12:25 UTC*
  - **PR** #45588: (samodid) update MasterPillarUtil get_minion_grains method (refs: #45751)
  - 454ed23f62 Merge pull request #45751 from rallytime/bp-45588
  - aa149a0e7a fix typo
  - 3e794a043d fix copy-paste error in get_minion_grains method doc string
  - 1fb94a08e0 update MasterPillarUtil
- **PR** #45753: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2018-01-29 17:11:11 UTC*

  - 860e21955c Merge pull request #45753 from rallytime/merge-2017.7
  - cb50cce181 Merge branch '2016.11' into '2017.7'
  - d7e09e2649 Merge pull request #45749 from vutny/fix-typo
    - e80bfb20c6 The *zypper.mod_repo*: fix typo in the docstring
  - cb6ce378ea Merge pull request #45459 from vutny/salt-cloud-fix-loading-utf-cache
    - b370796e9d Salt Cloud: write/read cached data in UTF-8 explicitly
    - cd999201be [2016.11] Salt Cloud: fix loading UTF-8 cached data
- **ISSUE** #40173: (gtmanfred) Document the Open File limit issue better (refs: #45688)
- **PR** #45688: (bdrung) Raise LimitNOFILE to default max open files @ *2018-01-29 14:26:57 UTC*
  - 9fb4d4a528 Merge pull request #45688 from bdrung/raise-max-open-files
  - bbedeec756 Raise LimitNOFILE to default max open files
- **PR** #45686: (bdrung) Use dbus-run-session instead of dbus-launch @ *2018-01-29 14:24:11 UTC*
  - 79da49ec8b Merge pull request #45686 from bdrung/2017.7
  - f49d0a0eec Use dbus-run-session instead of dbus-launch
- **PR** #45740: (terminalmage) Fix incorrect attempt at version comparison. @ *2018-01-29 14:12:05 UTC*
  - 7fb666bcd2 Merge pull request #45740 from terminalmage/fix-incorrect-version-comparison
  - 1e0b38dcaa Fix incorrect attempt at version comparison.
- **PR** #45747: (SteffenKockel) Fix typos @ *2018-01-29 13:53:27 UTC*
  - fe636f53f8 Merge pull request #45747 from SteffenKockel/2017.7
  - 319b513183 Fix typos
- **PR** #45734: (terminalmage) Fix traceback in CLI output when value is trimmed @ *2018-01-28 13:35:56 UTC*
  - eb91ae8b6e Merge pull request #45734 from terminalmage/fix-trimmed-output
  - 966ad07452 Fix traceback in CLI output when value is trimmed
- **PR** #45712: (bdrung) Decode git call output in Python 3 @ *2018-01-28 02:03:21 UTC*
  - 7516bfbffe Merge pull request #45712 from bdrung/fix-version-decode
  - 217183405a Decode git call output in Python 3
- **ISSUE** #44449: (brianthelion) salt-ssh + salt-cloud: cloud roster not working and/or *update_cachedir* is broken (refs: #45720)
- **PR** #45720: (dwoz) Salt cloud adds newly created insances to cache @ *2018-01-26 22:45:43 UTC*
  - 91b848debb Merge pull request #45720 from dwoz/issue-44449-prod-fix
  - 4a4bd6119d Salt cloud adds newly created insances to cache
- **PR** #45724: (eliasp) Typo (*Hellium → Helium*) @ *2018-01-26 22:37:44 UTC*
  - 831698f066 Merge pull request #45724 from eliasp/2017.7-typo-from-hell
  - bec78276f3 Replace left-over mistyped codename reference (*Hellium → 2014.7.0*)
- **PR** #45722: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2018-01-26 22:15:40 UTC*
  - cdb21a0186 Merge pull request #45722 from rallytime/merge-2017.7

  - 8e3a2e25fe Merge branch '2016.11' into '2017.7'
  - e4047a1234 Merge pull request #45511 from twangboy/win_fix_git
    - 160dd7c6ce Pull the first item in the list
    - 52d6d78150 Only keep ssh.py in the Windows installer
    - 54eb0db2c4 Keep ssh state and execution modules in the installer
    - 0fa801a329 Add additional path to find ssh.exe
  - a550e8d25d Merge pull request #45694 from twangboy/win_reg_add_keys
    - 8f53cd2d68 Add new keys to subkey_slash_check
    - 62050c711c Add support for additional reg keys
  - 7ceebf62f0 Merge pull request #45577 from zer0def/fix-git-detached-31363
    - a924b971ef Applied PR #40524 to *git.detached* state module function. (refs #31363)
- **PR** #45718: (rallytime) [2017.7] Merge forward from 2017.7.3 to 2017.7 @ *2018-01-26 16:49:44 UTC*
  - 3a413e96c5 Merge pull request #45718 from rallytime/merge-2017.7
  - f10c7ee92d Merge branch '2017.7.3' into '2017.7'
- **PR** #45690: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2018-01-26 14:41:44 UTC*
  - d0955519cf Merge pull request #45690 from rallytime/merge-2017.7
  - d4dac9f7cc Merge branch '2016.11' into '2017.7'
    - 3a6837e232 Merge pull request #45675 from Ch3LL/rn_2016.11.9
      - 7b5bed36d9 Add new commits to 2016.11.9 release notes
    - 915e259bad Merge pull request #45663 from rallytime/bp-45452-2016.11
      - ae94fb61d9 opkg.py: make owner function return value, instead of iterator
    - ecd75c137f Merge pull request #45651 from rallytime/merge-2016.11
      - 1583e1edbe Merge branch '2016.11.9' into '2016.11'
      - 10812969f0 Merge pull request #45638 from twangboy/win_fix_shell_info
      - 872da3ffba Only convert text types in the list_values function
      - 0e41535cdb Fix reg.py to only convert text types to unicode
      - 3579534ea5 Fix issue with detecting powershell
      - 2d1dd1186e Merge pull request #45564 from Ch3LL/r-notes-2016
      - 28e4398150 Merge pull request #45563 from Ch3LL/man_2016
    - 22bcd3d110 Merge pull request #45600 from vutny/doc-fix-references
      - 35675fe6b3 [DOC] Fix references on Salt Formulas page
    - 0d622f92a9 Merge pull request #45542 from UtahDave/doc_mixed_transports
      - b5b5054ec2 capitalize masters and minions
      - f542bdf566 Add warning about using mixed transports
    - c70b9dc20b Merge pull request #45565 from Ch3LL/r-notes-2016
      - 325f4cbcda Add PR changes to 2016.11.9 Release Notes

        - d8526062c1 Merge pull request #45562 from Ch3LL/man_2016
            - 529bc0c680 update release number for salt-call man page 2016.11.9
            - 11b7222148 Update man pages for 2016.11.9
- **PR** #45710: (michelsen) Added source argument to function call @ *2018-01-26 14:30:48 UTC*
    - 9c92e93834 Merge pull request #45710 from michelsen/fix-chocolatey-state-bug
    - 8accc0ce5c Added source argument to function call
- **PR** #45667: (gtmanfred) default to upgrading when refreshing on archlinux @ *2018-01-25 14:05:24 UTC*
    - 693f72d5a7 Merge pull request #45667 from gtmanfred/syu
    - 44c601102a we should default to upgrading when refreshing on archlinux
- **PR** #45674: (rallytime) [2017.7] Merge forward from 2017.7.3 to 2017.7 @ *2018-01-24 22:46:31 UTC*
    - bec946b080 Merge pull request #45674 from rallytime/merge-2017.7
    - 9f78e53d4b Merge branch '2017.7.3' into '2017.7'
- **ISSUE** #45590: (viq) webhook engine does not work with tornado 4.5.2 (refs: #45589)
- **PR** #45589: (gtmanfred) change webhook headers to dict @ *2018-01-24 22:32:37 UTC*
    - 50de847191 Merge pull request #45589 from gtmanfred/2017.7
    - 395d6f5c91 change webhook headers to dict
- **ISSUE** #45072: (vernondcole) cannot build documentation on Ubuntu 17.10 (refs: #45662)
- **PR** #45662: (bdrung) Fix documentation generation @ *2018-01-24 17:14:22 UTC*
    - e21088c1a4 Merge pull request #45662 from bdrung/2017.7
    - 71076afbcc doc: Define fake version for msgpack and psutil
    - b6a5b745b1 doc: Mock keyring module import
- **PR** #45650: (rallytime) Back-port #45555 to 2017.7 @ *2018-01-24 14:47:54 UTC*
    - **PR** #45555: (ddoh94) update winrepo_source_dir document (refs: #45650)
    - da82f190d2 Merge pull request #45650 from rallytime/bp-45555
    - e474d0416b update winrepo_source_dir document
- **PR** #45611: (terminalmage) Fix unnecessary/incorrect usage of six.binary_type @ *2018-01-23 22:53:20 UTC*
    - 79ee24c0c7 Merge pull request #45611 from terminalmage/tests-log-level
    - 6aa865cf54 Fix unnecessary/incorrect usage of six.binary_type
- **PR** #45652: (rallytime) [2017.7] Merge forward from 2017.7.3 to 2017.7 @ *2018-01-23 22:45:22 UTC*
    - 634d8dbcc0 Merge pull request #45652 from rallytime/merge-2017.7
    - 4e907dc84b Merge branch '2017.7.3' into '2017.7'
- **ISSUE** #45627: (bdrung) Failing unit tests in Debian package build (refs: #45630)
- **PR** #45630: (bdrung) Fix tests @ *2018-01-23 21:56:46 UTC*
    - dbdef8230e Merge pull request #45630 from bdrung/2017.7
    - 76d44e9490 Fix skipping test when boto is not installed
    - 2b9b262357 Fix unit.modules.test_cmdmod.CMDMODTestCase.test_run

- **PR** #45619: (garethgreenaway) [2017.7] Fixing test_mac_user_enable_auto_login @ *2018-01-23 21:56:03 UTC*
  - e5c9cd91e8 Merge pull request #45619 from garethgreenaway/2017_7_test_mac_user_enable_auto_login
  - f5f03e1e6c Fixing integration.modules.test_mac_user.MacUserModuleTest.test_mac_user_disable_auto_login
- **PR** #45644: (twangboy) Add missing space to deprecation warning @ *2018-01-23 21:55:11 UTC*
  - 8a95fc4257 Merge pull request #45644 from twangboy/win_fix_dep_warns
  - de9bc384cc Add missing space to deprecation warning
- **PR** #45634: (Ch3LL) Add different service name for Mac 10.13 test @ *2018-01-23 21:51:56 UTC*
  - c290b6320b Merge pull request #45634 from Ch3LL/mac-service
  - 31b712e27d Add different service name for Mac 10.13 test
- **PR** #45606: (terminalmage) Fix bug affecting salt-ssh when root_dir differs from the default @ *2018-01-23 20:03:49 UTC*
  - c28151f5f7 Merge pull request #45606 from terminalmage/salt-ssh-root-dir-fix
  - 3f9309521b Fix bug affecting salt-ssh when root_dir differs from the default
- **PR** #45636: (Ch3LL) Fix mac service and pkg tests for 10.13 @ *2018-01-23 18:44:56 UTC*
  - 0931b6417d Merge pull request #45636 from Ch3LL/mac-tests
- **PR** #45609: (rallytime) [2017.7] Merge forward from 2017.7.3 to 2017.7 @ *2018-01-22 20:24:36 UTC*
  - 63a294f498 Merge pull request #45609 from rallytime/merge-2017.7
  - a5fc3b3363 Merge branch '2017.7.3' into '2017.7'
- **ISSUE** #45431: (zer0def) boto3_route53.hosted_zone_present state can fail due to related execution module function's typo (refs: #45576)
- **PR** #45576: (zer0def) Fixed boto3_route53 execution module function signature *diSassociate_vpc_from_hosted_zone* typo. (refs #45431) @ *2018-01-22 19:37:13 UTC*
  - 59329957ca Merge pull request #45576 from zer0def/boto3-route53-typo
  - 21e1e9e226 Fixed boto3_route53 execution module function signature *diSassociate_vpc_from_hosted_zone* typo. (refs #45431)
- **PR** #45552: (rallytime) [2017.7] Merge forward from 2017.7.3 to 2017.7 @ *2018-01-19 19:12:49 UTC*
  - 42b0d27f71 Merge pull request #45552 from rallytime/merge-2017.7-from-.3
  - dba7410b80 Merge branch '2017.7.3' into '2017.7'
- **PR** #45551: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2018-01-19 18:21:27 UTC*
  - 879cfcb889 Merge pull request #45551 from rallytime/merge-2017.7
  - e0ffa32b49 Merge branch '2016.11' into '2017.7'
  - 18e814a7bb Merge pull request #45540 from rallytime/merge-2016.11
    - 441f819b7b Merge branch '2016.11.9' into '2016.11'
    - 654df0f526 Merge pull request #45532 from gtmanfred/2016.11.9
      - 6c26025664 fix mock for opensuse
  - 4f3b9b57fa Merge pull request #45522 from rallytime/merge-2016.11
    - 36c038c92a Merge branch '2016.11.9' into '2016.11'

    - 571c33aa39 Merge pull request #45518 from gtmanfred/2016.11.9
      - 5455d2dee6 fix centos 6 pip test
      - 40255194b0 fix fedora pkg test
  - 0638638fb9 Merge pull request #45504 from rallytime/merge-2016.11
    - d72fc74e8c Merge branch '2016.11.9' into '2016.11'
      - 4e0a0eec1f Merge pull request #45443 from rallytime/bp-45399-2016.11.9
      - 919e92c911 Fix git.latest failure when rev is not the default branch
      - ebd4db66b8 Merge pull request #45493 from damon-atkins/2016.11_fix_sls_defintion_wrong_type
      - af108440df win_pkg lint space after ,
      - c6e922a236 win_pkg lint issues
      - f4627d7a80 fix quote i.e. change ‘ to ’
      - 6938a4c099 pkg.refresh_db report an issue if a sls pkg definition id not a dict instead of aborting.
  - 5a2a31bfff Merge pull request #45495 from vutny/doc-rhel-pygit2-compat
    - 0d79b9eaff [DOC] Suggest to upgrade pygit2 and deps
  - 9c4fb42e5f Merge pull request #45481 from twangboy/fix_aptpkg
    - fd67b086b4 Fix if statement in __init__()
- **ISSUE** #42626: (UtahDave) new args and kwargs options for publisher_acl are not documented at all and very little for external_auth (refs: #45389)
- **PR** #45389: (DmitryKuzmenko) Docs update for function args limit in pub acl. @ *2018-01-18 22:56:10 UTC*
  - 28554ca935 Merge pull request #45389 from DSRCorporation/bugs/42626_pub_acl_doc
  - f33ebcada0 Doc note about user names regex matching in pub acl and eauth.
  - e29c0ff19e Docs update for function args limit in pub acl.
- **PR** #45483: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2018-01-18 22:54:12 UTC*
  - b3dc758ab0 Merge pull request #45483 from rallytime/merge-2017.7
  - de6d85959a Lint fix
  - 9f547a31f0 Merge branch '2016.11' into '2017.7'
    - b756760415 Merge pull request #45482 from rallytime/merge-2016.11
      - 3b38c77159 Merge branch '2016.11.9' into '2016.11'
      - 7322efba92 Merge pull request #45446 from rallytime/bp-45390
    - 96ae237d37 Merge pull request #45448 from rallytime/merge-2016.11.9
      - 646379d981 Merge branch '2016.11' into '2016.11.9'
    - 1ed323a3ee Merge pull request #45437 from terminalmage/fix-docstring
      - c11b16dc29 Fix incorrect wording in jboss7 docstrings
    - 600fa3939f Merge pull request #45390 from damon-atkins/2016.11_win_pkg_remove_final_fixes
      - 69f045ea24 lint too-many-blank-lines

        - 10a7501ede Update release notes
        - 6f2affe01c fix pkg.remove, pkg.list_pkgs
      - 057df44a4a Merge pull request #45399 from terminalmage/fix-git.latest-depth
      - 0cbc6767bf Fix git.latest failure when rev is not the default branch
      - b0ece9f4d4 Merge pull request #45424 from twangboy/win_reg
      - 30f06205f7 Fix some issues with reg.py
- **PR** #45529: (Ch3LL) Fix UnboundLocalError for pacman pkg installs @ *2018-01-18 19:01:49 UTC*
  - 5e26282843 Merge pull request #45529 from Ch3LL/pacman-sources
  - e619d49ef3 Fix UnboundLocalError for pacman pkg installs
- **PR** #45508: (frogunder) fix test_archive test for mac on 2017.7 branch @ *2018-01-18 16:04:36 UTC*
  - 840c97417d Merge pull request #45508 from frogunder/fix_mac_archive_tests_2017.7_branch
  - ccf062d62e fix test_archive test for mac on 2017.7 branch
- **PR** #45444: (rallytime) Back-port #45343 to 2017.7 @ *2018-01-17 17:17:59 UTC*
  - **PR** #45343: (rrroo) Support expr_form for manage.up, manage.down (refs: #45444)
  - e1403b6813 Merge pull request #45444 from rallytime/bp-45343
  - c7d2081390 Support expr_form for manage.up, manage.down
- **PR** #45465: (terminalmage) Backport #45095 to 2017.7 branch @ *2018-01-17 15:13:05 UTC*
  - **PR** #45095: (terminalmage) PY3: Make loader ignore .pyc files not in __pycache__ (refs: #45465)
  - 4b2c88e2e6 Merge pull request #45465 from terminalmage/bp-45095
  - 2f63a6dbf4 Optimization: don't allocate a new list to concatenate
  - 5074741130 EAFP
  - 85dbdc6a39 PY3: Make loader ignore .pyc files not in __pycache__
- **PR** #45365: (meaksh) Return an error when "gid_from_name" is set but group does not exist @ *2018-01-16 18:31:50 UTC*
  - 5f58a87e84 Merge pull request #45365 from meaksh/2017.7-issue-45345
  - da23067f80 Refactor to prevent logical bug when gid is 0
  - 9fdaa0d5e9 Update documentation for 'gid_from_name' parameter
  - 52f9c06908 Fix integration tests for 'user.present' state.
  - e2c32dc6dc Make pylint happy
  - a18dbe0c11 Return error when gid_from_name and group does not exist.
  - ce7b1f4baf Ensure empty string gid is set to None
- **ISSUE** #43535: (Ch3LL) Add pkg.latest_version Test to Auto Test Suite (refs: #44822)
- **PR** #44822: (frogunder) add pkg_latest_version test @ *2018-01-16 14:16:54 UTC*
  - de080983e3 Merge pull request #44822 from frogunder/pkg_latestversion
  - 08644e02a0 skip if mac
  - dfb68f32d2 fix if statements and string

  - 3504083849 add pkg_latest_version test
- **ISSUE** saltstack/salt-jenkins#603: (rallytime) [oxygen] CentOS 7 is failing several boto tests with module import failures (refs: #45401)
- **PR** #45435: (rallytime) Back-port #45401 to 2017.7 @ *2018-01-14 12:43:48 UTC*
  - **PR** #45401: (gtmanfred) fix boto import failures (refs: #45435)
  - cb3e0cffb3 Merge pull request #45435 from rallytime/bp-45401
  - b9761971c2 fix moto version
  - 0cd95d1cc6 fix test boto imports
- **PR** #45380: (twangboy) Backport changes from #45308 @ *2018-01-11 19:45:21 UTC*
  - **PR** #45308: (twangboy) Fix *integration.modules.test_state* for Windows (refs: #45380)
  - 2340f0b487 Merge pull request #45380 from twangboy/backport_45308
  - 419be8a9b5 Backport changes from #45308

28.2.26 Salt 2017.7.6 Release Notes

Version 2017.7.6 is a bugfix release for *2017.7.0*.

Statistics

- Total Merges: **182**
- Total Issue References: **60**
- Total PR References: **217**
- Contributors: **47** (Ch3LL, DmitryKuzmenko, GwiYeong, Quarky9, RichardW42, UtahDave, amaclean199, arif-ali, baniobloom, bdrung, benediktwerner, bmiguel-teixeira, cachedout, dafenko, damon-atkins, dwoz, ezh, folti, fpicot, frogunder, garethgreenaway, gtmanfred, isbm, jeroennijhof, jfindlay, jfoboss, kstreee, lomeroe, mattp-, meaksh, mirceaulinic, myinitialsarepm, mzbroch, nages13, paclat, pcjeff, pruiz, psyer, rallytime, s0undt3ch, skizunov, smitty42, terminalmage, twangboy, vutny, yagnik, yannj-fr)

Tornado 5.0 Support for Python 2 Only

Tornado 5.0 moves to using asyncio for all python3 versions. Because of this and changes in asyncio between python 3.4 and 3.5 to only be able to use one ioloop, which requires some rearchitecting, support for tornado 5.0 and python3 versions of salt has been delayed to a later release.

For now, to use tornado 5.0, the python 2 version of salt must be used.

Tornado 5.0 Support for Python 2 Only

Tornado 5.0 moves to using asyncio for all python3 versions. Because of this and changes in asyncio between python 3.4 and 3.5 to only be able to use one ioloop, which requires some rearchitecting, support for tornado 5.0 and python3 versions of salt has been delayed to a later release.

For now, to use tornado 5.0, the python 2 version of salt must be used.

Option to Return to Previous Pillar Include Behavior

Prior to version 2017.7.3, keys from *pillar includes* would be merged on top of the pillar SLS. Since 2017.7.3, the includes are merged together and then the pillar SLS is merged on top of that.

The `pillar_includes_override_sls` option has been added allow users to switch back to the pre-2017.7.3 behavior.

Changelog for v2017.7.5..v2017.7.6

Generated at: 2018-05-29 14:05:53 UTC

- **PR** #47775: (gtmanfred) catch UnsupportedOperation with AssertionError @ *2018-05-22 19:04:13 UTC*
  - edf94c915e Merge pull request #47775 from gtmanfred/2017.7.6
  - 548f65d056 catch UnsupportedOperation with AssertionError
- **PR** #47769: (gtmanfred) skip test that breaks test suite @ *2018-05-22 15:12:54 UTC*
  - 8c38ecd75f Merge pull request #47769 from gtmanfred/2017.7.6
  - 3fdfc0fa82 skip test that breaks test suite
- **PR** #47747: (Ch3LL) Add changelog to 2017.7.6 release notes @ *2018-05-21 14:25:00 UTC*
  - 0d5b473ce2 Merge pull request #47747 from Ch3LL/rn_2017.7.6
  - d4aa83b92d Add changelog to 2017.7.6 release notes
- **ISSUE** #47484: (whytewolf) Windows: pkg.latest state not updating packages. (refs: #47702)
- **PR** #47702: (damon-atkins) State pkg.latest called win pkg.install with list of pkgs and the required versions @ *2018-05-19 11:21:23 UTC*
  - 8a5b34f7d9 Merge pull request #47702 from damon-atkins/2017.7.6_fix_pkg.latest_state
  - adcc094e08 Merge branch '2017.7.6' into 2017.7.6_fix_pkg.latest_state
- **PR** #47700: (yannj-fr) fix roots modification time check @ *2018-05-18 18:42:17 UTC*
  - d610c192d9 Merge pull request #47700 from yannj-fr/2017.7.6
  - 961c1ef61e fix roots modification time check
    - 2a73e905df Merge branch '2017.7.6' into 2017.7.6
- **PR** #47632: (gtmanfred) handle new _create_stream in tornado 5.0 @ *2018-05-18 14:25:17 UTC*
  - 266749420f Merge pull request #47632 from gtmanfred/2017.7.6
  - 2c50c0d2f5 fix pylint
  - 4a29057b16 Fix last test for tornado
  - 550ef2e272 allow using tornado 5.0
  - 62e468448b handle new _create_stream in tornado 5.0
- **PR** #47720: (rallytime) Back-port #47692 to 2017.7.6 @ *2018-05-18 13:23:03 UTC*
  - **PR** #47692: (dwoz) Default windows to m1.small for ec2-classic (refs: #47720)
  - 2643c356af Merge pull request #47720 from rallytime/bp-47692-2017.7.6
  - 6e5cb36839 Default windows to m1.small for ec2-classic
    - 20d9785244 fix roots modification time check

  - aef37dd1ce fix roots modification time check
  - d51662e053 Ensure targeted_pkgs always contains value for non-windows.
  - 83b4224cf8 Adjusted based on feed back.
  - 12f983ce9f Whitespace lint issues
  - 075d3d3c49 pkg.install execution module on windows ensures the software package is installed when no version is specified, it does not upgrade the software to the latest. This is per the design. pkg.latest must provide the versions to install to pkg.install
- **PR** #47667: (Ch3LL) Update test_mac_user_enable_auto_login to test both py2 and py3 @ *2018-05-16 15:54:49 UTC*
  - 16c2153385 Merge pull request #47667 from Ch3LL/mac_user_enable
  - ba40d3d1a1 Update test_mac_user_enable_auto_login to test both py2 and py3
- **PR** #47645: (Ch3LL) query the pip path for test test_issue_2087_missing_pip @ *2018-05-15 17:16:10 UTC*
  - a4921e86c9 Merge pull request #47645 from Ch3LL/py3_rm_pip
  - 225d90ad4c query the pip path for test test_issue_2087_missing_pip
- **PR** #47646: (rallytime) Back-port #47601 and #47643 to 2017.7.6 @ *2018-05-15 14:04:45 UTC*
  - **PR** #47643: (dwoz) Remove unwanted file (refs: #47646)
  - **PR** #47601: (dwoz) Skip tests when we can not use runas (refs: #47646)
  - e441733ac1 Merge pull request #47646 from rallytime/bp-47601-and-47643
  - 9e1d1a5ef8 Fix typo
  - 4e94609136 Remove unwanted file
  - 0109249c78 use ignore-undefined-variable
  - 37caecb7f4 Ignore pylint WindowsError
  - c1135d90c7 Better doc string
  - e53d6b9ed9 Skip tests when we can not use runas
- **PR** #47570: (gtmanfred) Update dependency to msgpack @ *2018-05-10 13:23:09 UTC*
  - 6f178ca908 Merge pull request #47570 from gtmanfred/2017.7.6
  - 84aa034e03 Update dependency to msgpack
- **PR** #47523: (rallytime) [2017.7.6] Update man pages @ *2018-05-08 13:31:19 UTC*
  - 98bd598701 Merge pull request #47523 from rallytime/man-pages
  - 48ecb78dec [2017.7.6] Update man pages
- **ISSUE** #47443: (skylerberg) Input validation does not raise SaltInvocationError in win_dsc.py (refs: #47505)
- **PR** #47517: (rallytime) Back-port #47505 to 2017.7.6 @ *2018-05-07 19:42:37 UTC*
  - **PR** #47505: (dwoz) Raise proper invocation errors (refs: #47517)
  - e608ea9617 Merge pull request #47517 from rallytime/bp-47505-2017.7.6
  - 0734578533 Raise proper invocation errors
- **PR** #47476: (gtmanfred) Specify the cache directory for newer virtualenv modules @ *2018-05-04 19:20:45 UTC*
  - 611ca1fc03 Merge pull request #47476 from gtmanfred/2017.7

  - 1f91a85587 specify cache dir for pip install
  - 99e150e09c check for kitchen-vagrant gem before loading windows tests
- **PR** #47412: (twangboy) Fix issue where the cwd was being removed @ *2018-05-04 17:28:11 UTC*
  - 7c3f2c56da Merge pull request #47412 from twangboy/fix_47125
  - c9bab0b8e3 Merge branch '2017.7' into fix_47125
  - 2600e404d5 Fix overly long line
  - 5c8db05769 Fix issue where the cwd was being removed
- **PR** #47467: (twangboy) Remove unused settings, update NSIS @ *2018-05-04 17:11:37 UTC*
  - 4846e957c4 Merge pull request #47467 from twangboy/cleanup_settings
  - 9d498293b1 Remove unused settings, update NSIS
- **PR** #47196: (twangboy) Fix issues with pip @ *2018-05-04 14:23:04 UTC*
  - da9871d36b Merge pull request #47196 from twangboy/fix_47024
  - 14ee5537b9 Add @with_tempdir helper
  - 6c3b5fa6fa Fix typo
  - f031710af2 Merge branch '2017.7' into fix_47024
  - 7c46d9d0d4 Fix integration.modules.test_pip
  - 22ac81df63 Fix integration.modules.test_pip
  - 57d98224d4 Merge pull request #9 from terminalmage/twangboy/fix_47024
    - 37a13d8004 Update pip unit tests to reflect changes
    - 7f86779be0 Lint fix
  - c48d8f4f61 DRY and other fixes in pip module
  - b1117896a0 Change from global variable to __context__``
  - 3e6e524eca Fix some tests``
  - c94f0f20e4 Fix lint error
  - fd47b21530 Fix merge conflict
- **PR** #47455: (Ch3LL) Add In Progress Warning for 2017.7.6 Release Notes @ *2018-05-04 13:44:54 UTC*
  - e8c4524bae Merge pull request #47455 from Ch3LL/unreleased_rn
  - b6d0cc2ab7 Add In Progress Warning for 2017.7.6 Release Notes
- **PR** #47459: (gtmanfred) update ubuntu-rolling to 18.04 @ *2018-05-03 20:39:20 UTC*
  - 2c7a4b6179 Merge pull request #47459 from gtmanfred/2017.7
  - d228e72477 update ubuntu-rolling to 18.04
- **PR** #47462: (terminalmage) Fix docs build on Sphinx 1.7+ @ *2018-05-03 20:06:57 UTC*
  - 64a64c0ed7 Merge pull request #47462 from terminalmage/docs
  - 6d7803ece0 Fix docs build on Sphinx 1.7+
- **ISSUE** #47436: (lomeroe) Some Administrative Template policies are not properly set by lgpo (refs: #47438)
- **ISSUE** #44516: (doesitblend) Windows PY3 Minion Returns UTF16 UnicodeError (refs: #44944)

- **PR** #47438: (lomeroe) lgpo fix for issue #47436 @ *2018-05-03 14:40:27 UTC*
  - **PR** #44944: (lomeroe) win_lgpo registry.pol encoding updates (refs: #46913, #47438)
  - 6cd0d31c03 Merge pull request #47438 from lomeroe/double_admx_test
  - 4902f1e2ba check if a policy has either an enabled value or enabled list entry or a disabled value or disabled list entry when determining the state of the policy
- **ISSUE** #45790: (bdarnell) Test with Tornado 5.0b1 (refs: #47106, #47433)
- **PR** #47433: (s0undt3ch) Add missing requirements files not commited in #47106 @ *2018-05-02 20:57:14 UTC*
  - **PR** #47106: (DmitryKuzmenko) Tornado50 compatibility fixes (refs: #47433)
  - ed69821d19 Merge pull request #47433 from s0undt3ch/2017.7
  - 5abadf25d6 Add missing requirements files not commited in #47106
- **ISSUE** #47424: (bcharron) "salt-cloud -m" fails with nova driver: "There was a query error: u'state'" (refs: #47429)
- **PR** #47429: (gtmanfred) server_list_min should use state, not status @ *2018-05-02 16:27:56 UTC*
  - 7ae3497b0c Merge pull request #47429 from gtmanfred/2017.7
  - 8ae32033cc server_list_min should use state, not status
- **PR** #47399: (isbm) zeromq 17 deprecation warning backport from 2018.3 + tornado 5 fixes @ *2018-05-02 15:11:16 UTC*
  - 2f5fc4ecc5 Merge pull request #47399 from isbm/isbm-zeromq17-deprecationwarning-2017.7.2-v2
  - a36e49fd27 fix pylint
  - 98b5629b36 Fix imports
  - d94c0f0152 Remove unnecessary variable
  - 8e377b5653 Lintfix: E0203 and attribute access
  - 2aab70b1b8 Install ZMQ handler if <15 version
  - 296c589f4b Use ZMQ switch utility in the integration tests
  - ab5fa34d7c Use ZMQ_VERSION_INFO constant everywhere
  - 43b5558b82 Add trace logging on ZMQ sockets communication
  - 164204a9fe Remove duplicate code for ZMQ monitor handling
  - 834b1e4ff0 Remove obsolete ZMQIOLoop direct instance
  - 1c90cbdb3c Remove an empty line
  - ef2e0acd66 Add logging on ZMQ socket exception
  - 38ceed371d Lintfix: ident
  - 1ece6a5f52 Lintfix: line too long
  - 4e650c0b44 Remove code duplicate by reusing utilities functions
  - 57da54b676 Fix imports
  - 948368e9a1 Add libzmq version info builder
  - 0b4a17b859 Update log exception message
  - 116e1809fc Put a message alongside the exception to the logs

  - 4bc43124b7 Remove unnecessary ZMQ import and check for its presence
  - 05f4d40269 Use utility for ZMQ import handling in SSH client
  - 457ef7d9a5 Use utility for ZMQ import handling in flo/zero
  - 08dee6f5bd Use utility for ZMQ import handling
  - e2a353cfb0 Remove unnecessary ZMQ extra-check for cache utils
  - c8f2cc271d Remove unnecessary ZMQ extra-check for master utils
  - 3940667bb9 Remove old ZMQ import handling
  - f34a53e029 Use ZMQ utility for version check
  - cbb26dcb28 Use ZMQ installer for master
  - 453e83210a Add ZMQ version build
  - af9601e21d Use ZMQ importer utility in async
  - d50b2b2023 Incorporate tornado-5 fixes
  - 1fd9af0655 Add ZMQ backward-compatibility tornado installer for older versions
  - ad4b40415c Add one place for handling various ZMQ versions and IOLoop classes
- **PR** #47343: (Ch3LL) Add additional service module integration tests and enable for windows @ *2018-05-02 13:39:46 UTC*
  - b14e974b5f Merge pull request #47343 from Ch3LL/win_srv_test
  - 2173b6f549 ensure we are enabling/disabling before test
  - d58be06751 Add additionatl service module integration tests and enable for windows
- **PR** #47375: (terminalmage) Warn on use of virtual packages in pkg.installed state @ *2018-05-01 21:12:18 UTC*
  - b0f3fb577f Merge pull request #47375 from terminalmage/issue47310
  - fa2bea52bb Remove extra blank line to appease linter
  - f8ab2be81c Add debug logging if we fail to detect virtual packages
  - 67c4fc56ac Warn on use of virtual packages in pkg.installed state
- **PR** #47415: (kstreee) Fixes a bug of rest_tornado's 'local' client, complement fix of #46326 @ *2018-05-01 21:11:25 UTC*
  - **PR** #47200: (kstreee) Resolve a conflict with syndic timeout and bug fixes of the local client in rest_tornado (refs: #47415)
  - **PR** #47123: (rallytime) [develop] Merge forward from 2018.3 to develop (refs: #47200)
  - **PR** #47110: (kstreee) Fixes misusing of the timeout option. (refs: #47200)
  - **PR** #46692: (mattp-) saltnado bugfixes for ldap & syndics (refs: #47123, #47200)
  - **PR** #46326: (kstreee) Fixes a timing bug of saltnado's client local. (refs: #47110, #47123, #47200, #47415)
  - **PR** #45874: (GwiYeong) fix for local client timeout bug (refs: #46326)
  - 56235032f4 Merge pull request #47415 from kstreee/fix-local-client-tgt-bug
  - b8d37e0a1e To add a test case for the syndic environment, copies the test case which was written by @mattp- that was already merged into develop branch, related pr is #46692.
  - 4627bad1fd Realizes 'tgt' field into actual minions using ckminions to subscribe results of the minions before publishing a payload.

- **PR** #47286: (baniobloom) fixed vpc_peering_connection_name option @ *2018-05-01 19:02:10 UTC*
  - d65ceaee03 Merge pull request #47286 from baniobloom/vpc_peering_connection_name_fix
  - a968965087 Merge branch '2017.7' into vpc_peering_connection_name_fix
- **PR** #47270: (meaksh) Fix minion scheduler to return 'retcode' from executed functions @ *2018-04-30 18:21:55 UTC*
  - 8a5d4437bb Merge pull request #47270 from meaksh/2017.7-fix-retcode-on-schedule-utils
  - d299cf3385 Merge branch '2017.7' into 2017.7-fix-retcode-on-schedule-utils
  - b6da600fff Initialize __context__ retcode for functions handled via schedule util module
- **ISSUE** #47264: (jf) doc: https://docs.saltstack.com/en/latest/ref/modules/all/salt.modules.grains.html#salt.modules.grains.delval s/of pass/or pass/ (refs: #47371)
- **PR** #47371: (rallytime) Fix "of pass" typo in grains.delval docs: change to "or pass" @ *2018-04-30 18:18:46 UTC*
  - 5b51075384 Merge pull request #47371 from rallytime/fix-47264
  - a43485b49c Fix "of pass" typo in grains.delval docs: change to "or pass"
- **PR** #47389: (dwoz) Older GitPython versions will not have close @ *2018-04-29 16:42:06 UTC*
  - a86e53be66 Merge pull request #47389 from dwoz/moregittestfix
  - 67745c1362 Older GitPython versions will not have close
- **PR** #47388: (dwoz) Fix missing import @ *2018-04-28 18:33:14 UTC*
  - a5367eaf63 Merge pull request #47388 from dwoz/test_pip_fix
  - eb26321e8b Fix missing import
- **PR** #47380: (gtmanfred) add io_loop handling to runtests engine @ *2018-04-28 17:25:28 UTC*
  - 9b59b991c2 Merge pull request #47380 from gtmanfred/2017.7
  - 93d1445ec1 add io_loop handling to runtests engine
- **PR** #47384: (dwoz) Fix py2 version of pip test @ *2018-04-28 15:13:28 UTC*
  - 37822c0cbb Merge pull request #47384 from dwoz/test_pip_fix
  - a37a9da1fb Fix py2 version of pip test
- **PR** #47382: (dwoz) Close the repo and fix multiple tests @ *2018-04-28 15:09:17 UTC*
  - eefd96732e Merge pull request #47382 from dwoz/gitfs_tests
  - 1570708fac Close the repo and fix multiple tests
- **PR** #47369: (terminalmage) Return an empty dict if no search_order in ldap ext_pillar config file @ *2018-04-27 20:58:52 UTC*
  - 57c75ff660 Merge pull request #47369 from terminalmage/ldap_pillar
  - 085883ae2d Return an empty dict if no search_order in ldap ext_pillar config file
- **PR** #47363: (DmitryKuzmenko) Tornado5.0: Future.exc_info is dropped @ *2018-04-27 18:30:18 UTC*
  - bcc66dd9bf Merge pull request #47363 from DSRCorporation/bugs/replace_exc_info_with_exception
  - 3f7b93a23c Tornado5.0: Future.exc_info is dropped
- **PR** #47334: (terminalmage) pillar_ldap: Fix cryptic errors when config file fails to load @ *2018-04-27 17:53:51 UTC*

  - bcef34f7e1 Merge pull request #47334 from terminalmage/ldap_pillar
  - 0175a8687c pillar_ldap: Fix cryptic errors when config file fails to load
  - 65c3ba7ff1 Remove useless documentation
  - 5d67cb27de Remove unncessary commented line
- **PR** #47347: (dwoz) Proper fix for mysql tests @ *2018-04-27 17:27:53 UTC*
  - 31db8ca7ad Merge pull request #47347 from dwoz/test_mysql_fix_again
  - add78fb618 Fix linter warnings
  - 2644cc7553 Fix linter nits
  - 799c601184 Proper fix for mysql tests
- **PR** #47359: (gtmanfred) add mention of the formulas channel to the formulas docs @ *2018-04-27 16:56:13 UTC*
  - e573236848 Merge pull request #47359 from gtmanfred/2017.7
  - 6214ed8133 add mention of the formulas channel to the formulas docs
- **PR** #47317: (dwoz) Do not join a thread that is stopped @ *2018-04-27 13:15:09 UTC*
  - **PR** #47279: (dwoz) Gracefully shutdown worker threads (refs: #47317)
  - 629503b2a8 Merge pull request #47317 from dwoz/threadshutdown
  - 6db2a0e4d3 Log exceptions at exception level
  - d4ae787595 Do not join a thread that is stopped
- **PR** #47304: (cachedout) Pass timeout to salt CLI for tests @ *2018-04-27 13:11:58 UTC*
  - aacd5cefe3 Merge pull request #47304 from cachedout/test_cli_timeout_arg
  - 85025af83c Pass timeout to salt CLI for tests
- **PR** #47311: (Ch3LL) Add firewall execution modules tests for windows @ *2018-04-27 13:10:54 UTC*
  - 55534fb659 Merge pull request #47311 from Ch3LL/firewall_windows
  - 4e16c18c16 Add firewall module windows tests to whitelist
  - 4b2fc4ec66 Add windows firewall execution modules integration tests
- **PR** #47348: (dwoz) Ignore gitfs tests when symlinks not enabled @ *2018-04-27 13:08:27 UTC*
  - 1667375a80 Merge pull request #47348 from dwoz/no_symlinks
  - 94a70e847a Ignore gitfs tests when symlinks not enabled
- **PR** #47342: (dwoz) Fix mysql test cases @ *2018-04-27 00:50:53 UTC*
  - dac04261b5 Merge pull request #47342 from dwoz/test_mysql_fix
  - 7496f4c5a8 Fix mysql test cases
- **PR** #47341: (dwoz) Fix python 3 support for inet_pton function @ *2018-04-26 23:35:45 UTC*
  - 34e78ef564 Merge pull request #47341 from dwoz/inet_pton_fix
  - 85451f48d4 Fix python 3 support for inet_pton function
- **PR** #47339: (dwoz) Use salt.utils.fopen for line ending consistancy @ *2018-04-26 22:39:56 UTC*
  - e4779f3246 Merge pull request #47339 from dwoz/ssh_key_test_fix
  - e37a93a1ca Remove redundent close call

  - b2ae5889b7 Close the temporary file handle
  - 9f7f83a975 Use salt.utils.fopen for line ending consistancy
- **PR** #47335: (dwoz) Remove un-needed string-escape @ *2018-04-26 21:49:27 UTC*
  - b221860151 Merge pull request #47335 from dwoz/pip_test_fix
  - dcb6a22c00 Remove un-needed string-escape
- **PR** #47331: (dwoz) Do not encode usernames @ *2018-04-26 19:57:28 UTC*
  - 1c527bfd3a Merge pull request #47331 from dwoz/py3_wingroup_fix
  - cc154ef857 Do not encode usernames
- **PR** #47329: (cachedout) Credit Frank Spierings @ *2018-04-26 16:37:59 UTC*
  - 708078b152 Merge pull request #47329 from cachedout/frank_credit
  - 33c0644ac4 Credit Frank Spierings
- **PR** #47281: (Ch3LL) Add win_system integration module tests @ *2018-04-26 16:07:41 UTC*
  - a545e55543 Merge pull request #47281 from Ch3LL/system_test
  - c9181a75a6 Add destructivetest decorator on tests
  - 0d0c8987fc Add win_system integration module tests
- **PR** #47283: (Ch3LL) Add windows ntp integration module tests @ *2018-04-26 16:04:44 UTC*
  - b64d930df0 Merge pull request #47283 from Ch3LL/ntp_test
  - ced7f86546 Add windows ntp integration module tests
- **PR** #47314: (Ch3LL) Skip netstat test on macosx as its not supported @ *2018-04-26 16:00:37 UTC*
  - 910aff910f Merge pull request #47314 from Ch3LL/net_mac_test
  - 67beb1451c Skip netstat test on macosx as its not supported
- **PR** #47307: (rallytime) Back-port #47257 to 2017.7 @ *2018-04-26 15:16:23 UTC*
  - **PR** #47257: (jeroennijhof) Role is not a list but a dictionary (refs: #47307)
  - 0549ef7c16 Merge pull request #47307 from rallytime/bp-47257
  - 6c5b2f92bc Role is not a list but a dictionary
- **PR** #47312: (rallytime) Update bootstrap script to latest release: 2018.04.25 @ *2018-04-26 15:15:13 UTC*
  - d6ff4689f6 Merge pull request #47312 from rallytime/update-bootstrap-release
  - 765cce06a2 Update bootstrap script to latest release: 2018.04.25
- **PR** #47279: (dwoz) Gracefully shutdown worker threads (refs: #47317) @ *2018-04-25 21:15:43 UTC*
  - e0765f5719 Merge pull request #47279 from dwoz/py3_build_fix
  - 21dc1bab91 Pep-8 line endings
  - 717abedaf7 Fix comman wart
  - 4100dcd64c Close might get called more than once
  - dbe671f943 Stop socket before queue on delete
  - 9587f5c69e Silence pylint import-error for six.moves
  - 4b0c7d3b34 Fix typo

  - 05adf7c2b1 Use six.moves for queue import
  - fe340778fa Gracefully shutdown worker threads
- **PR** #47113: (jfindlay) Support proto for IPSec policy extension in iptables state @ *2018-04-25 18:01:19 UTC*
  - 44f19b2f94 Merge pull request #47113 from jfindlay/iptables_state
  - 8bd08012ee modules,states.iptables support proto for policy ext
- **PR** #47302: (Ch3LL) Remove unnecessary code from core grains and add test @ *2018-04-25 17:58:48 UTC*
  - b7a6206330 Merge pull request #47302 from Ch3LL/dead_code
  - daa68b4877 Add virtual grains test for core grains
  - a59dd2785d Remove dead code in core grains file for virt-what
- **PR** #47303: (baniobloom) Added clarity on oldest supported main release branch @ *2018-04-25 17:52:39 UTC*
  - e29362acfc Merge pull request #47303 from baniobloom/bug_fix_doc
  - b97c9df5f3 added clarity on how to figure out what is the oldest supported main release branch
- **ISSUE** #45790: (bdarnell) Test with Tornado 5.0b1 (refs: #47106, #47433)
- **PR** #47106: (DmitryKuzmenko) Tornado50 compatibility fixes (refs: #47433) @ *2018-04-25 15:32:37 UTC*
  - 0d9d55e013 Merge pull request #47106 from DSRCorporation/bugs/tornado50
  - 39e403b18d Merge branch '2017.7' into bugs/tornado50
  - 6706b3a2d1 Run off of a temporary config
  - d6873800d5 Allow running pytest>=3.5.0
  - 2da3983740 Tornado 5.0 compatibility fixes
- **ISSUE** #47258: (drewmat) service state no longer working after kernel upgrade (refs: #47271)
- **PR** #47271: (gtmanfred) load rh_service for amazon linux not booted with systemd @ *2018-04-25 14:47:06 UTC*
  - 2e014f4746 Merge pull request #47271 from gtmanfred/amazon
  - 8a53908908 Do not load rh_service module when booted with systemd
  - e4d1d5bf11 Revert "support amazon linux 2 for service module"
- **ISSUE** #44847: (malbertus) netconfig.managed state.apply unexpected behaviour of test & debug variables (refs: #47246)
- **PR** #47246: (mirceaulinic) Attempting to fix #44847: allow a different way to get the test and debug flags into the netconfig state @ *2018-04-25 14:44:02 UTC*
  - 599b0ed1e9 Merge pull request #47246 from cloudflare/fix-44847-2017.7
  - ad80028104 This way, we can pass flags such as `debug` into the state, but also `test`.
- **PR** #47220: (benediktwerner) Fix pip.installed when no changes occurred with pip >= 1.0.0 @ *2018-04-25 14:23:50 UTC*
  - **PR** #47207: (benediktwerner) Fix pip_state with pip3 if no changes occourred (refs: #47220)
  - **PR** #47102: (gtmanfred) dont allow using no_use_wheel for pip 10.0.0 or newer (refs: #47220)
  - 4e2e1f0719 Merge pull request #47220 from benediktwerner/fix-pip-2017.7
  - 0197c3e973 Fix pip test
  - 34bf66c09f Fix pip.installed with pip>=10.0.0

- **PR** #47272: (rallytime) Add windows tests and reg module/state to CODEOWNERS file for team-windows @ *2018-04-25 13:25:29 UTC*
  - 92e606251f Merge pull request #47272 from rallytime/reg-windows-codeowners
  - 9445af0185 Add windows tests and reg module/state to CODEOWNERS file for team-windows
    - 8de3d41adb fixed vpc_peering_connection_name option
- **PR** #47252: (rallytime) Fix the matching patterns in the CODEOWNERS file to use fnmatch patterns @ *2018-04-24 14:10:42 UTC*
  - 9dca5c0221 Merge pull request #47252 from rallytime/codeowners-fixes
  - 204b6af92b Fix the matching patterns in the CODEOWNERS file to use fnmatch patterns
- **ISSUE** #47173: (fpicot) pkg.installed ignores normalize parameter (refs: #47177)
- **PR** #47177: (fpicot) fix normalize parameter in pkg.installed @ *2018-04-24 13:37:54 UTC*
  - 3de1bb49c8 Merge pull request #47177 from fpicot/fix_47173_pkg_normalize
  - 149f846f34 fix normalize parameter in pkg.installed
- **PR** #47251: (Ch3LL) Update Docs to remove unnecessary + sign @ *2018-04-23 19:37:04 UTC*
  - 10e30515dc Merge pull request #47251 from Ch3LL/pub_fix_rn
  - fa4c2e6575 Update Docs to remove unnecessary + sign
- **PR** #47249: (Ch3LL) Add CVE number to 2016.3.6 Release @ *2018-04-23 19:05:42 UTC*
  - bb7850a431 Merge pull request #47249 from Ch3LL/pub_fix_rn
  - 24dea24b7e Add CVE number to 2016.3.6 Release
- **ISSUE** #47225: (pruiz) zfs.filesystem_present takes forever on a dataset with lots (10k+) of snapshots (refs: #47226, #47227)
- **PR** #47227: (pruiz) Fix issue #47225: avoid zfs.filesystem_present slowdown when dataset has lots of snapshots (2017.7 branch) @ *2018-04-23 14:05:58 UTC*
  - **PR** #47226: (pruiz) Fix issue #47225: avoid zfs.filesystem_present slowdown when dataset has lots of snapshots (refs: #47227)
  - 56933eb0b2 Merge pull request #47227 from pruiz/pruiz/zfs-dataset-present-slow-2017.7
  - fded61f19b Fix issue #47225: avoid zfs.filesystem_present slowdown when dataset has lots of snapshots
- **PR** #47167: (smitty42) Adding updates for python3 compatibility and new virtualbox SDK versi… @ *2018-04-23 13:20:42 UTC*
  - 9825065048 Merge pull request #47167 from smitty42/vbox-skd-fix
  - 5de53139cd Merge branch '2017.7' into vbox-skd-fix
- **PR** #47213: (dwoz) Fix coverage on py3 windows builds @ *2018-04-20 22:09:57 UTC*
  - 976f031170 Merge pull request #47213 from dwoz/py3win
  - ad9c7f63f0 Fix coverate on py3 windows builds
  - 91252bac95 Adding updates for python3 compatibility and new virtualbox SDK version support.
- **PR** #47197: (dwoz) Move process target to top level module namespace @ *2018-04-20 15:41:06 UTC*
  - cebcd6d069 Merge pull request #47197 from dwoz/testfix
  - 25803c9176 Move process target to top level module namespace

- **PR** #47193: (Ch3LL) Add network module integration tests @ *2018-04-20 13:37:19 UTC*
    - d4269c2b70 Merge pull request #47193 from Ch3LL/network_test
    - bbf9987c19 Add network module integration tests
- **PR** #47189: (Ch3LL) Add autoruns.list integration test for Windows @ *2018-04-19 21:16:34 UTC*
    - c777248a78 Merge pull request #47189 from Ch3LL/autoruns
    - 6a88bedb7a Add autoruns to windows whitelist
    - e9e4d4af70 Add autoruns.list integration test for Windows
- **PR** #47184: (Ch3LL) Add status module integration modules tests for Windows @ *2018-04-19 19:38:56 UTC*
    - 65f344e371 Merge pull request #47184 from Ch3LL/status_test
    - 25a84428b8 Add status module integration modules tests for Windows
- **PR** #47163: (rallytime) Updage jenkins module autodocs to use jenkinsmod name instead @ *2018-04-19 19:35:00 UTC*
    - **PR** #46801: (yagnik) rename jenkins to jenkinsmod (refs: #46900, #47163)
    - 965600ad6c Merge pull request #47163 from rallytime/jenkins-autodoc
    - 0039395017 Updage jenkins module autodocs to use jenkinsmod name instead
- **PR** #47185: (twangboy) Add additional integration tests to whitelist @ *2018-04-19 18:20:25 UTC*
    - 0a43dde5fc Merge pull request #47185 from twangboy/add_tests
    - 345daa0423 Add additional integration tests to whitelist
- **PR** #47172: (dwoz) Allow non admin name based runs on windows @ *2018-04-19 17:26:42 UTC*
    - 1a600bb9a4 Merge pull request #47172 from dwoz/cover_without_admin
    - cadd759727 Use warnings to warn user
    - 144c68e214 Allow non admin name based runs on windows
- **PR** #47110: (kstreee) Fixes misusing of the timeout option. (refs: #47200) @ *2018-04-18 17:16:20 UTC*
    - **PR** #46326: (kstreee) Fixes a timing bug of saltnado's client local. (refs: #47110, #47123, #47200, #47415)
    - **PR** #45874: (GwiYeong) fix for local client timeout bug (refs: #46326)
    - d5997d2301 Merge pull request #47110 from kstreee/fix-misusing-of-timeout
    - 0624aee0ed Fixes misusing of the timeout option.
- **ISSUE** #40948: (ScoreUnder) salt-call falsely reports a master as down if it does not have PKI directories created (refs: #40961)
- **PR** #40961: (terminalmage) Make error more explicit when PKI dir not present for salt-call @ *2018-04-18 16:08:17 UTC*
    - 87ca2b4003 Merge pull request #40961 from terminalmage/issue40948
    - 6ba66cca41 Fix incorrect logic in exception check
    - fed5041c5f Make error more specific to aid in troubleshooting
    - 8c67ab53b4 Fix path in log message
    - 3198ca8b19 Make error more explicit when PKI dir not present for salt-call
- **PR** #47134: (Ch3LL) Add user integration tests for windows OS @ *2018-04-18 14:29:40 UTC*

  - f5e63584d4 Merge pull request #47134 from Ch3LL/user_win_test
  - e7c9bc4038 Add user integration tests for windows OS
- **PR** #47131: (gtmanfred) add __cli opts variable for master processes @ *2018-04-17 21:33:57 UTC*
  - da2f6a3fac Merge pull request #47131 from gtmanfred/cli
  - 1b1c29bf62 add __cli for master processes
- **ISSUE** #47116: (pcjeff) pip 10.0.0 can not import pip.req (refs: #47121)
- **PR** #47129: (rallytime) Back-port #47121 to 2017.7 @ *2018-04-17 20:45:11 UTC*
  - **PR** #47121: (pcjeff) fix pip import error in pip 10.0.0 (refs: #47129)
  - 9b8e6ffb8c Merge pull request #47129 from rallytime/bp-47121
  - 11da526b21 add ImportError
  - bd0c23396c fix pip.req import error in pip 10.0.0
- **PR** #47102: (gtmanfred) dont allow using no_use_wheel for pip 10.0.0 or newer (refs: #47220) @ *2018-04-17 20:44:58 UTC*
  - eb5ac51a48 Merge pull request #47102 from gtmanfred/2017.7
  - 3dc93b310b fix tests
  - 8497e08f8e fix pip module for 10.0.0
  - 4c07a3d1e9 fix other tests
  - b71e3d8a04 dont allow using no_use_wheel for pip 10.0.0 or newer
- **PR** #47037: (twangboy) Fix build_env scripts @ *2018-04-17 18:54:17 UTC*
  - c1dc42e67e Merge pull request #47037 from twangboy/fix_dev_scripts
  - 990a24d7ed Fix build_env scripts
- **PR** #47108: (dwoz) Fix unit.utils.test_event.TestAsyncEventPublisher.test_event_subscription @ *2018-04-17 00:25:07 UTC*
  - 6a4c0b8a1a Merge pull request #47108 from dwoz/async_test_fix
  - 3d85e30ce5 AsyncTestCase is required for AsyncEventPublisher
- **PR** #47068: (cachedout) Catch an operation on a closed socket in a test @ *2018-04-16 19:56:03 UTC*
  - 03892eaf0b Merge pull request #47068 from cachedout/catch_value_error_socket_test
  - 7db5625632 Catch an operation on a closed socket in a test
- **PR** #47065: (dwoz) Jinja test fix @ *2018-04-16 16:16:42 UTC*
  - 1ea2885ec2 Merge pull request #47065 from dwoz/jinja_test_fix
  - 673cd31c65 Merge branch '2017.7' into jinja_test_fix
- **PR** #47077: (dwoz) Fix failing state test by normalizing line endings @ *2018-04-16 15:48:39 UTC*
  - 5293b5b5ca Merge pull request #47077 from dwoz/test_state_fix
  - 444da3f893 Fix py3 wart (chr vs bytesstring)
  - e8acca01c2 Fix failing state test by normalizing line endings
- **ISSUE** #46538: (HenriWahl) salt-cloud gives "FutureWarning: The behavior of this method will change in future versions." (refs: #47067)

- **PR** #47067: (gtmanfred) use the recommended opennebula lookup method @ *2018-04-16 15:48:15 UTC*
  - ca967de5da Merge pull request #47067 from gtmanfred/2017.7
  - f913a7859c use the recommended opennebula lookup method
- **PR** #47064: (dwoz) Fix fileserver roots tests @ *2018-04-14 21:30:23 UTC*
  - 7fddad6cd9 Merge pull request #47064 from dwoz/roots_tests_fix
  - 25fd7c0694 fix py3 wart, encode os.linesep
  - d79f1a1961 Fix fileserver roots tests
- **PR** #47069: (cachedout) Pass the timeout variable to the CLI when calling salt in tests @ *2018-04-14 15:20:25 UTC*
  - 977c6939c4 Merge pull request #47069 from cachedout/match_timeout_arg
  - b8990f5258 Pass the timeout variable to the CLI when calling salt in tests
- **PR** #47074: (dwoz) Kitchn should ignore artifacts directory @ *2018-04-14 13:06:19 UTC*
  - 2c4c19c622 Merge pull request #47074 from dwoz/ignore_artifacts
  - c3941efad0 Kitchn should ignore artifacts directory
- **ISSUE** #47000: (mvintila) Client API: full_return paramenter missing from cmd_subset function (refs: #47055)
- **PR** #47055: (mattp-) #47000 - add proper handling of full_return in cmd_subset @ *2018-04-13 20:17:10 UTC*
  - c484c0bd71 Merge pull request #47055 from bloomberg/GH-47000
  - 8af3f5b874 GH-47000: add proper handling of full_return in cmd_subset
- **PR** #47039: (twangboy) Fix winrm powershell script @ *2018-04-13 18:09:56 UTC*
  - f3496030cc Merge pull request #47039 from twangboy/win_fix_winrm_script
  - 6635b9003f Fix winrm powershell script
    - 46fa2c04de Fix py3 os.linesep wart
    - 3c565d7e54 Use salt.utils.fopen
    - aa965310f1 Clean up cruft
    - efc9866580 Jinja test fixes
- **PR** #46326: (kstreee) Fixes a timing bug of saltnado's client local. (refs: #47110, #47123, #47200, #47415) @ *2018-04-13 13:59:28 UTC*
  - **PR** #45874: (GwiYeong) fix for local client timeout bug (refs: #46326)
  - 1700a10ebe Merge pull request #46326 from kstreee/fix-client-local
  - 0f358a9c9e Fixes a timing bug of saltnado's client local.
- **ISSUE** #46877: (trudesea) Unable to apply GPO (Windows 2016) (refs: #46913)
- **ISSUE** #44516: (doesitblend) Windows PY3 Minion Returns UTF16 UnicodeError (refs: #44944)
- **PR** #46913: (lomeroe) 2017.7 Fix #46877 – win_lgpo start/shutdown script reading @ *2018-04-12 15:10:50 UTC*
  - **PR** #44944: (lomeroe) win_lgpo registry.pol encoding updates (refs: #46913, #47438)
  - c3c00316c5 Merge pull request #46913 from lomeroe/2017_7-fix46877
  - 369a0645ed move exception for clarity

  - 32ce5bfda5 Use configparser serializer object to read psscript.ini and script.ini startup/shutdown script files.
- **PR** #47025: (terminalmage) Fix server_id grain in PY3 on Windows @ *2018-04-12 15:08:00 UTC*
  - 9e37cfc9d6 Merge pull request #47025 from terminalmage/fix-server_id-windows
  - cb0cf89ed3 Fix server_id grain in PY3 on Windows
- **PR** #47027: (rallytime) Back-port #44508 to 2017.7 @ *2018-04-12 15:05:51 UTC*
  - **PR** #44508: (mzbroch) Capirca integration (refs: #47027)
  - 2e193cfb45 Merge pull request #47027 from rallytime/bp-44508
  - 8e72f362f4 Add priority field to support the latest capirca.
  - 112f92baab Add priority field to support the latest capirca.
- **PR** #47020: (rallytime) Back-port #46970 to 2017.7 @ *2018-04-11 21:48:25 UTC*
  - **PR** #46970: (garethgreenaway) [2017.7] fix to pkgrepo comments test (refs: #47020)
  - 385fe2bc1e Merge pull request #47020 from rallytime/bp-46970
  - 9373dff52b Update test_pkgrepo.py
  - 13cf9eb5b1 Removing debugging.
  - a61a8593e5 Removing suse from pkgrepo comments tests. the pkgrepo functions in SUSE pkg module do not support comments.
- **ISSUE** #46504: (jfoboss) ntp.managed fails on non-english systems (refs: #46539)
- **PR** #46539: (jfoboss) #46504 Fix @ *2018-04-11 14:13:24 UTC*
  - 8f994e7cf9 Merge pull request #46539 from jfoboss/patch-1
  - 6890122e41 Merge pull request #1 from twangboy/pull_46539
    - 19c3fadbe5 Fix unit test for win_ntp
  - 826a8d3099 Fixing #46504
- **PR** #46999: (gtmanfred) switch pip test package @ *2018-04-10 21:18:33 UTC*
  - 74d70e95a5 Merge pull request #46999 from gtmanfred/2017.7
  - 791af8f6ce switch pip test package
- **PR** #46023: (mattp-) add parallel support for orchestrations @ *2018-04-10 19:26:04 UTC*
  - 8adaf7f526 Merge pull request #46023 from bloomberg/parallel-orch
  - 0ac0b3ca29 Merge branch '2017.7' into parallel-orch
- **ISSUE** #46581: (qcpeter) puppet.fact tries to parse output to stderr (refs: #46613)
- **PR** #46613: (myinitialsarepm) Fix puppet.fact and puppet.facts to use stdout. @ *2018-04-10 15:18:07 UTC*
  - 39d65a39cf Merge pull request #46613 from myinitialsarepm/fix_puppet.fact_and_puppet.facts
  - 44ecd13abc Update tests to use cmd.run_all
  - 7d7d40f541 Merge branch '2017.7' into fix_puppet.fact_and_puppet.facts
  - 0ce1520bd0 Merge branch '2017.7' into fix_puppet.fact_and_puppet.facts
  - 69e1f6f681 Fix puppet.fact and puppet.facts to use stdout.
- **PR** #46991: (gtmanfred) use saltstack salt-jenkins @ *2018-04-10 14:19:00 UTC*

  - ba5421d988 Merge pull request #46991 from gtmanfred/windows
  - 98588c1dc5 use saltstack salt-jenkins
- **PR** #46975: (gtmanfred) Make windows work for test runs in jenkinsci @ *2018-04-10 13:41:18 UTC*
  - 00c4067585 Merge pull request #46975 from gtmanfred/windows
  - 1f69c0d7f8 make sure windows outputs xml junit files
  - 4a2ec1bbb3 support new versions of winrm-fs
  - b9efec8526 remove libnacl on windows
  - 2edd5eaf9e fix path
  - b03e272e44 windows work
- **PR** #46945: (vutny) [DOC] Fix Jinja block in FAQ page @ *2018-04-09 13:05:28 UTC*
  - 3cf2353e41 Merge pull request #46945 from vutny/doc-faq-fix-jinja
  - bfdf54e61d [DOC] Fix Jinja block in FAQ page
- **PR** #46925: (terminalmage) Remove reference to directory support in file.patch state @ *2018-04-06 13:54:47 UTC*
  - fc2f728665 Merge pull request #46925 from terminalmage/fix-file.patch-docstring
  - 97695657f0 Remove reference to directory support in file.patch state
- **PR** #46900: (rallytime) Back-port #46801 to 2017.7 @ *2018-04-06 13:47:44 UTC*
  - **PR** #46801: (yagnik) rename jenkins to jenkinsmod (refs: #46900, #47163)
  - eef6c518e1 Merge pull request #46900 from rallytime/bp-46801
  - 6a41e8b457 rename jenkins to jenkinsmod
- **PR** #46899: (rallytime) Back-port #45116 to 2017.7 @ *2018-04-06 13:47:17 UTC*
  - **PR** #45116: (arif-ali) fix adding parameters to http.query from sdb yaml (refs: #46899)
  - 71839b0303 Merge pull request #46899 from rallytime/bp-45116
  - b92f908da4 fix adding parameters to http.query from sdb yaml
    - 3d5e69600b address lint issues raised by @isbm
    - a9866c7a03 fix parallel mode py3 compatibility
    - 6d7730864a removing prereq from test orch
    - 6c8a25778f add integration test to runners/test_state to exercise parallel
    - 2c86f16b39 cherry-pick cdata KeyError prevention from #39832
    - 26a96e8933 record start/stop duration for parallel processes separately
    - e4844bdf2b revisit previous join() behavior in check_requisites
    - f00a359cdf join() parallel process instead of a recursive sleep
    - 6e7007a4dc add parallel support for orchestrations
- **ISSUE** #43529: (Ch3LL) Add publisher_acl Test to Auto Test Suite (refs: #44926)
- **PR** #44926: (frogunder) whitelist_acl_test @ *2018-04-05 15:09:26 UTC*
  - d0f5b43753 Merge pull request #44926 from frogunder/whitelisted_acl

  - 18e460fc30 Merge branch '2017.7' into whitelisted_acl
  - 1ad4d7d988 fix assert errors
  - e6a56016df update test
  - 19a2244cb7 whitelist_acl_test
- **ISSUE** #46456: (vitaliyf) "ValueError" when running orch with "subset" (refs: #46464)
- **PR** #46464: (gtmanfred) fix salt subset in orchestrator @ *2018-04-05 14:52:01 UTC*
  - 7d822f9cec Merge pull request #46464 from gtmanfred/orchestration
  - 637cdc6b7b fix pylint
  - 0151013ddb document *cli* option for cmd_subset
  - 4a3ed6607d add test for subset in orchestration
  - 3112359dd6 fix salt subset in orchestrator
- **ISSUE** #46523: (dwoz) Add a test to the cloud suite for Windows minion on EC2 (refs: #46879)
- **PR** #46879: (dwoz) Fix multiple typos causing tests to fail @ *2018-04-05 13:59:28 UTC*
  - 805ed1c964 Merge pull request #46879 from dwoz/cloudtestfix
  - dc54fc53c3 Fix multiple typos causing tests to fail
- **PR** #46647: (twangboy) Fix the tear down function in integration.modules.test_grains @ *2018-04-04 21:14:06 UTC*
  - f70f6de282 Merge pull request #46647 from twangboy/win_fix_test_grains
  - c179388b0e Fix the tear down function in integration.modules.test_grains.GrainsAppendTestCase
- **ISSUE** #46754: (nages13) grain item virtual_subtype shows 'Xen PV DomU' on Docker containers (refs: #46756)
- **ISSUE** #43405: (kfix) LXD-created LXC container is detected as a Xen domU (refs: #46756)
- **PR** #46756: (nages13) fix grains['virtual_subtype'] to show Docker on xen kernels @ *2018-04-04 20:53:49 UTC*
  - 91c078ce12 Merge pull request #46756 from nages13/bugfix-grain-virtual_subtype
  - 781f5030a4 Merge branch 'bugfix-grain-virtual_subtype' of https://github.com/nages13/salt into bugfix-grain-virtual_subtype
    - cd1ac4b7f9 Merge branch '2017.7' into bugfix-grain-virtual_subtype
    - 0ace76c0e7 Merge branch '2017.7' into bugfix-grain-virtual_subtype
    - 9eb6f5c0d0 Merge branch '2017.7' into bugfix-grain-virtual_subtype
    - 73d6d9d365 Merge branch '2017.7' into bugfix-grain-virtual_subtype
    - a4a17eba6a Merge branch '2017.7' into bugfix-grain-virtual_subtype
    - bf5034dbdb Merge branch '2017.7' into bugfix-grain-virtual_subtype
    - 8d12770951 Merge branch '2017.7' into bugfix-grain-virtual_subtype
  - 7e704c0e81 Moved down container check code below hypervisors to validate containers type running in virtual environment. Fixes #46754 & #43405
  - 710f74c4a6 fix grains['virtual_subtype'] to show Docker on xen kernels
- **ISSUE** #46762: (ScoreUnder) prereq stack overflow (refs: #46788, #46799)
- **PR** #46799: (garethgreenaway) [2017.7] Adding test for PR #46788 @ *2018-04-04 20:41:23 UTC*

  - **PR** #46788: (garethgreenaway) [2017.7] Ensure failed tags are added to self.pre (refs: #46799)
  - 058bbed221 Merge pull request #46799 from garethgreenaway/46762_prereq_shenanigans_tests
  - 13875e78cf Fixing documention string for test.
  - 3d288c44d4 Fixing test documentation
  - 6cff02ef6a Adding tests for #46788
- **PR** #46867: (terminalmage) Backport string arg normalization to 2017.7 branch @ *2018-04-04 18:06:57 UTC*
  - d9770bf3f8 Merge pull request #46867 from terminalmage/unicode-logging-normalization
  - 7652688e83 Backport string arg normalization to 2017.7 branch
- **PR** #46770: (twangboy) Change the order of SID Lookup @ *2018-04-04 17:33:10 UTC*
  - 9eb98b1f6e Merge pull request #46770 from twangboy/fix_46433
  - 89af0a6222 Merge branch '2017.7' into fix_46433
  - 67b4697578 Remove unused import (ling)
  - 9302fa5ab0 Clean up code comments
  - b383b9b330 Change the order of SID Lookup
- **ISSUE** #46826: (robgott) grain modules using tuples affect targeting (refs: #46839)
- **PR** #46839: (gtmanfred) match tuple for targets as well @ *2018-04-04 14:07:12 UTC*
  - 9c776cffb7 Merge pull request #46839 from gtmanfred/tupletarget
  - 3b7208ce27 match tuple for targets as well
- **ISSUE** #40245: (czhong111) salt-api automatically restart caused by "opening too many files" (refs: #46817)
- **ISSUE** #36374: (szjur) Descriptor leaks in multithreaded environment (refs: #46817)
- **ISSUE** #20639: (GrizzlyV) salt.client.LocalClient leaks connections to local salt master (refs: #46817)
- **PR** #46845: (rallytime) Back-port #46817 to 2017.7 @ *2018-04-03 19:52:29 UTC*
  - **PR** #46817: (mattp-) address filehandle/event leak in async run_job invocations (refs: #46845)
  - **PR** #32145: (paclat) fixes 29817 (refs: #46817)
  - 7db251dc11 Merge pull request #46845 from rallytime/bp-46817
  - 36a0f6d8ca address filehandle/event leak in async run_job invocations
- **PR** #46847: (dwoz) strdup from libc is not available on windows @ *2018-04-03 19:51:33 UTC*
  - e3d17ab7bc Merge pull request #46847 from dwoz/missing-strdup
  - 55845f4846 strdup from libc is not available on windows
- **ISSUE** #46765: (roskens) pkg.mod_repo fails with a python error when removing a dictionary key (refs: #46776)
- **PR** #46776: (gtmanfred) fix shrinking list in for loop bug @ *2018-04-03 17:32:16 UTC*
  - f2dd79f9c4 Merge pull request #46776 from gtmanfred/2017.7
  - edc1059ee0 fix shrinking list in for loop bug
- **PR** #46838: (gtmanfred) use http registry for npm @ *2018-04-03 17:02:32 UTC*
  - 1941426218 Merge pull request #46838 from gtmanfred/npm
  - bff61dd291 use http registry for npm

- **ISSUE** #42312: (frogunder) salt-call –local sys.doc none gives error/traceback in raspberry pi (refs: #46823)
- **PR** #46823: (rallytime) Improve __virtual__ checks in sensehat module @ *2018-04-03 16:56:08 UTC*
  - e544254e7b Merge pull request #46823 from rallytime/fix-42312
  - dafa820f93 Improve __virtual__ checks in sensehat module
- **PR** #46641: (skizunov) Make LazyLoader thread safe @ *2018-04-03 16:09:17 UTC*
  - 37f6d2de35 Merge pull request #46641 from skizunov/develop3
  - c624aa4827 Make LazyLoader thread safe
- **PR** #46837: (rallytime) [2017.7] Merge forward from 2016.11 to 2017.7 @ *2018-04-03 14:54:10 UTC*
  - 989508b100 Merge pull request #46837 from rallytime/merge-2017.7
  - 8522c1d634 Merge branch '2016.11' into '2017.7'
  - 3e844ed1df Merge pull request #46739 from rallytime/2016.11_update_version_doc
  - 4d9fc5cc0f Update release versions for the 2016.11 branch
- **PR** #46740: (rallytime) Update release versions for the 2017.7 branch @ *2018-04-03 14:36:07 UTC*
  - 307e7f35f9 Merge pull request #46740 from rallytime/2017.7_update_version_doc
  - 7edf98d224 Update 2018.3.0 information and move branch from "latest" to "previous"
  - 5336e866ac Update release versions for the 2017.7 branch
- **PR** #46783: (twangboy) Fix network.managed test=True on Windows @ *2018-04-03 12:54:56 UTC*
  - ebf5dd276f Merge pull request #46783 from twangboy/fix_46680
  - da5ce25ef3 Fix unit tests on Linux
  - b7f4f377cd Add space I removed
  - f1c68a09b5 Fix network.managed test=True on Windows
- **PR** #46821: (rallytime) Fix the new test failures from the mantest changes @ *2018-04-03 12:40:51 UTC*
  - **PR** #46778: (terminalmage) Replace flaky SPM man test (refs: #46821)
  - f652f25cc1 Merge pull request #46821 from rallytime/fix-mantest-failures
  - 209a8029c3 Fix the new test failures from the mantest changes
- **ISSUE** #46627: (vangourd) Win_LGPO fails on writing Administrative Template for Remote Assistance (refs: #46800)
- **PR** #46800: (lomeroe) fix win_lgpo to correctly create valuenames of list item types @ *2018-04-03 12:38:45 UTC*
  - c460f62081 Merge pull request #46800 from lomeroe/2017_7-46627
  - 2bee383e9d correct create list item value names if the valuePrefix attribute does not exist on the list item, the value is the value name, other wise, the valuename a number with the valuePrefix prepended to it
- **ISSUE** #46347: (twangboy) Buid 449: unit.modules.test_inspect_collector (refs: #46675)
- **PR** #46675: (dwoz) Skip test when git symlinks are not configured @ *2018-04-03 12:19:19 UTC*
  - df26f2641e Merge pull request #46675 from dwoz/inspectlib-tests
  - d39f4852d8 Handle non-zero status exception
  - 83c005802b Handle cases where git can not be found
  - 628b87d5c4 Skip test when git symlinks are not configured

- **ISSUE** #46808: (ezh) Sharedsecret authentication is broken (refs: #46809)
- **PR** #46815: (terminalmage) Backport #46809 to 2017.7 @ *2018-04-02 20:05:15 UTC*
  - **PR** #46809: (ezh) Fix sharedsecret authentication (refs: #46815)
  - 4083e7c460 Merge pull request #46815 from terminalmage/bp-46809
  - 71d5601507 Fix sharedsecret authentication
- **PR** #46769: (dwoz) Adding windows minion tests for salt cloud @ *2018-04-02 18:51:49 UTC*
  - 3bac9717f4 Merge pull request #46769 from dwoz/wincloudtest
  - eabc234e5d Fix config override name
  - 5c22a0f88d Use aboslute imports
  - 810042710d Set default cloud test timeout back to 500 seconds
  - 5ac89ad307 Use winrm_verify_ssl option causing tests to pass
  - 71858a709c allow not verifying ssl winrm saltcloud
  - ba5f11476c Adding windows minion tests for salt cloud
- **PR** #46786: (twangboy) Return int(-1) when pidfile contains invalid data @ *2018-04-02 18:42:12 UTC*
  - f1be939763 Merge pull request #46786 from twangboy/fix_46757
  - b0053250ff Remove int(), just return -1
  - 7d56126d74 Fixes some lint
  - 49b3e937da Return int(-1) when pidfile contains invalid data
- **PR** #46814: (terminalmage) Backport #46772 to 2017.7 @ *2018-04-02 18:39:37 UTC*
  - **PR** #46772: (bmiguel-teixeira) fix container removal if auto_remove was enabled (refs: #46814)
  - 89bf24b15c Merge pull request #46814 from terminalmage/bp-46772
  - a9f26f2ab8 avoid breaking if AutoRemove is not found
  - 97779c965d fix container removal if auto_remove was enabled
- **PR** #46813: (terminalmage) Get rid of confusing debug logging @ *2018-04-02 18:19:27 UTC*
  - 5ea4ffbdb6 Merge pull request #46813 from terminalmage/event-debug-log
  - 5d6de3a2eb Get rid of confusing debug logging
- **PR** #46766: (twangboy) Change the way we're cleaning up after some tests @ *2018-03-30 18:01:03 UTC*
  - e533b7182d Merge pull request #46766 from twangboy/win_fix_test_git
  - 5afc66452c Remove unused/redundant imports
  - 88fd72c52c Use with_tempfile decorator where possible
- **PR** #46778: (terminalmage) Replace flaky SPM man test (refs: #46821) @ *2018-03-30 17:55:14 UTC*
  - 69d450db84 Merge pull request #46778 from terminalmage/salt-jenkins-906
  - bbfd35d3ea Replace flaky SPM man test
- **ISSUE** #46762: (ScoreUnder) prereq stack overflow (refs: #46788, #46799)
- **PR** #46788: (garethgreenaway) [2017.7] Ensure failed tags are added to self.pre (refs: #46799) @ *2018-03-30 17:11:38 UTC*

  - c935ffb740 Merge pull request #46788 from garethgreenaway/46762_prereq_shenanigans
  - fa7aed6424 Ensure failed tags are added to self.pre.
- **ISSUE** #46354: (twangboy) Build 449: unit.test_state (refs: #46655)
- **ISSUE** #46350: (twangboy) Build 449: unit.test_pyobjects.RendererTests (refs: #46655)
- **ISSUE** #46349: (twangboy) Build 449: unit.test_pydsl (refs: #46655)
- **ISSUE** #46345: (twangboy) Build 449: unit.test_pyobjects.MapTests (Manual Pass) (refs: #46655)
- **PR** #46655: (dwoz) Fixing cleanUp method to restore environment @ *2018-03-29 18:31:48 UTC*
  - 395b7f8fdc Merge pull request #46655 from dwoz/pyobjects-46350
  - 5aabd442f2 Fix up import and docstring syntax
  - 62d64c9230 Fix missing import
  - 18b1730320 Skip test that requires pywin32 on *nix platforms
  - 45dce1a485 Add reg module to globals
  - 09f9322981 Fix pep8 wart
  - 73d06f664b Fix linter error
  - 009a8f56ea Fix up environ state tests for Windows
  - b4be10b8fc Fixing cleanUp method to restore environment
- **ISSUE** #36802: (rmarcinik) using clean=True parameter in file.recurse causes python process to spin out of control (refs: #46632)
- **PR** #46632: (dwoz) Fix file.recurse w/ clean=True #36802 @ *2018-03-29 18:30:42 UTC*
  - af45c49c42 Merge pull request #46632 from dwoz/file-recurse-36802
  - 44db77ae79 Fix lint errors and typo
  - cb5619537f Only change what is essential for test fix
  - eb822f5a12 Fix file.recurse w/ clean=True #36802
- **ISSUE** #46660: (mruepp) top file merging same does produce conflicting ids with gitfs (refs: #46751)
- **PR** #46751: (folti) top file merging strategy 'same' works again @ *2018-03-28 21:12:27 UTC*
  - 6e9f504ed1 Merge pull request #46751 from folti/2017.7
  - 7058f10381 same top merging strategy works again
- **PR** #46691: (Ch3LL) Add groupadd module integration tests for Windows @ *2018-03-28 18:01:46 UTC*
  - d3623e0815 Merge pull request #46691 from Ch3LL/win_group_test
  - 7cda825e90 Add groupadd module integration tests for Windows
- **ISSUE** #46352: (twangboy) Build 449: unit.test_client (refs: #46696)
- **PR** #46696: (dwoz) Windows *unit.test_client* fixes @ *2018-03-28 17:55:47 UTC*
  - 14ab50d3f4 Merge pull request #46696 from dwoz/win_test_client
  - ec4634fc06 Better explanation in doc strings
  - d9ae2abb34 Fix splling in docstring
  - b40efc5db8 Windows test client fixes

- **ISSUE** #45956: (frogunder) CTRL-C gives traceback on py3 setup (refs: #46032)
- **PR** #46732: (rallytime) Back-port #46032 to 2017.7 @ *2018-03-28 13:43:17 UTC*
  - **PR** #46032: (DmitryKuzmenko) Workaroung python bug in traceback.format_exc() (refs: #46732)
  - 1222bdbc00 Merge pull request #46732 from rallytime/bp-46032
  - bf0b962dc0 Workaroung python bug in traceback.format_exc()
- **ISSUE** #28142: (zmalone) Deprecate or update the copr repo (refs: #46749)
- **PR** #46749: (vutny) [DOC] Remove mentions of COPR repo from RHEL installation page @ *2018-03-28 13:20:50 UTC*
  - 50fe1e9480 Merge pull request #46749 from vutny/doc-deprecate-copr
  - a1cc55da3d [DOC] Remove mentions of COPR repo from RHEL installation page
- **PR** #46734: (terminalmage) Make busybox image builder work with newer busybox releases @ *2018-03-27 21:14:28 UTC*
  - bd1e8bcc7d Merge pull request #46734 from terminalmage/busybox
  - 6502b6b4ff Make busybox image builder work with newer busybox releases
- **ISSUE** saltstack/salt-jenkins#902: (rallytime) [2017.7/.5] Test failures for NPM on CentOS 6/7, Ubuntu 14, and OpenSUSE (refs: #46742)
- **PR** #46742: (gtmanfred) only use npm test work around on newer versions @ *2018-03-27 21:13:28 UTC*
  - c09c6f819c Merge pull request #46742 from gtmanfred/2017.7
  - fd0e649d1e only use npm test work around on newer versions
- **PR** #46743: (Ch3LL) Workaround getpwnam in auth test for MacOSX @ *2018-03-27 21:10:47 UTC*
  - 3b6d5eca88 Merge pull request #46743 from Ch3LL/mac_auth
  - 4f1c42c0e3 Workaround getpwnam in auth test for MacOSX
- **ISSUE** #26920: (david-fairbanks42) MySQL grant with underscore and wildcard (refs: #46171)
- **PR** #46171: (amaclean199) Fix mysql grant comparisons @ *2018-03-27 17:54:48 UTC*
  - b548a3e742 Merge pull request #46171 from amaclean199/fix_mysql_grants_comparison
  - 97db3d9766 Merge branch '2017.7' into fix_mysql_grants_comparison
  - 0565b3980e Merge branch '2017.7' into fix_mysql_grants_comparison
  - 8af407173d Merge branch '2017.7' into fix_mysql_grants_comparison
  - 00d13f05c4 Fix mysql grant comparisons by stripping both of escape characters and quotes. Fixes #26920
- **ISSUE** #5721: (ozgurakan) salt-minion can't restart itself (refs: #46709)
- **PR** #46709: (vutny) [DOC] Update FAQ about Salt self-restarting @ *2018-03-27 14:34:58 UTC*
  - 554400e067 Merge pull request #46709 from vutny/doc-faq-minion-master-restart
  - d0929280fc [DOC] Update FAQ about Salt self-restarting
- **PR** #46503: (psyer) Fixes stdout user environment corruption @ *2018-03-27 14:20:15 UTC*
  - 3f21e9cc65 Merge pull request #46503 from psyer/fix-cmd-run-env-corrupt
  - e8582e80f2 Python 3-compatibility fix to unit test
  - 27f651906d Merge pull request #1 from terminalmage/fix-cmd-run-env-corrupt

    - 172d3b2e04 Allow cases where no marker was found to proceed without raising exception
    - 35ad828ab8 Simplify the marker parsing logic
  - a09f20ab45 fix repr for the linter
  - 4ee723ac0f Rework how errors are output
  - dc283940e0 Merge branch '2017.7' into fix-cmd-run-env-corrupt
  - a91926561f Fix linting problems
  - e8d3d017f9 fix bytes or str in find command
  - 0877cfc38f Merge branch '2017.7' into fix-cmd-run-env-corrupt
  - 86176d1252 Merge branch '2017.7' into fix-cmd-run-env-corrupt
  - 3a7cc44ade Add python3 support for byte encoded markers
  - 09048139c7 Do not show whole env in error
  - ed94700255 fix missing raise statement
  - 15868bc88c Fixes stdout user environment corruption
- **PR** #46432: (twangboy) Default to UTF-8 for templated files @ *2018-03-26 19:02:14 UTC*
  - ac2a6616a7 Merge pull request #46432 from twangboy/win_locales_utf8
  - affa35c30d Revert passing encoding
  - a0ab27ef15 Merge remote-tracking branch 'dw/win_locales_utf8' into win_locales_utf8
    - 9f95c50061 Use default SLS encoding, fall back to system encoding
    - 6548d550d0 Use salt.utils.to_unicode
    - 8c0164fb63 Add ability to specify encoding in sdecode
    - 2e7985a81c Default to utf-8 on Windows
  - 8017860dcc Use salt.utils.to_unicode
  - c10ed26eab Add ability to specify encoding in sdecode
  - 8d7e2d0058 Default to utf-8 on Windows
- **PR** #46669: (terminalmage) Add option to return to pre-2017.7.3 pillar include merge order @ *2018-03-26 19:00:28 UTC*
  - fadc5e4ba4 Merge pull request #46669 from terminalmage/pillar-merge-order
  - b4a1d34b47 Add option to return to pre-2017.7.3 pillar include merge order
- **PR** #46711: (terminalmage) Add performance reminder for wildcard versions @ *2018-03-26 18:07:31 UTC*
  - b90f0d1364 Merge pull request #46711 from terminalmage/wildcard-versions-info
  - fc7d16f1af Add performance reminder for wildcard versions
- **ISSUE** #46353: (twangboy) Build 449: unit.returners.test_smtp_return (refs: #46693)
- **PR** #46693: (dwoz) File and Pillar roots are dictionaries @ *2018-03-26 15:15:38 UTC*
  - 6c80d90bb6 Merge pull request #46693 from dwoz/test_smtp_return
  - 5bf850c67f File and Pillar roots are dictionaries
- **ISSUE** #36153: (krcroft) Pillarenv doesn't allow using separate pillar environments (refs: #46543)

- **PR** #46543: (dafenko) Fix missing saltenv and pillarenv in pillar.item @ *2018-03-26 15:05:13 UTC*
  - 9a6bc1418c Merge pull request #46543 from dafenko/fix-add-saltenv-pillarenv-to-pillar-item
  - 6d5b2068aa Merge branch '2017.7' into fix-add-saltenv-pillarenv-to-pillar-item
  - 5219377313 Merge branch '2017.7' into fix-add-saltenv-pillarenv-to-pillar-item
  - b7d39caa86 Merge branch '2017.7' into fix-add-saltenv-pillarenv-to-pillar-item
  - 25f1074a85 Add docstring for added parameters
  - 973bc13955 Merge branch '2017.7' into fix-add-saltenv-pillarenv-to-pillar-item
  - 164314a859 Merge branch '2017.7' into fix-add-saltenv-pillarenv-to-pillar-item
  - 267ae9f633 Fix missing saltenv and pillarenv in pillar.item
- **PR** #46679: (vutny) [DOC] Correct examples in *pkg* state module @ *2018-03-26 14:40:07 UTC*
  - f776040e25 Merge pull request #46679 from vutny/doc-state-pkg
  - 4a730383bf [DOC] Correct examples in *pkg* state module
- **PR** #46646: (twangboy) Fix *unit.returners.test_local_cache* for Windows @ *2018-03-26 14:16:23 UTC*
  - 47409eaa6e Merge pull request #46646 from twangboy/win_fix_test_local_cache
  - 8d93156604 Fix *unit.returners.test_local_cache* for Windows
- **ISSUE** #46595: (aboe76) saltstack server_id changes with each run on python3 (refs: #46649)
- **PR** #46649: (terminalmage) Make server_id consistent on Python 3 @ *2018-03-26 13:58:59 UTC*
  - 0c2dce0416 Merge pull request #46649 from terminalmage/issue46595
  - e82a1aa1ec Make server_id consistent on Python 3
- **PR** #46588: (UtahDave) Don't crash when saltwinshell is missing @ *2018-03-21 20:26:31 UTC*
  - 4e7466a21c Merge pull request #46588 from UtahDave/no_crash_winshell
  - b7842a1777 Update error message.
  - 95dfdb91ca Don't stacktrace when salt-ssh w/o saltwinshell
- **ISSUE** #22063: (jeanpralo) Wildcard inside top.sls file for pillar (refs: #41423)
- **ISSUE** #20581: (notpeter) Many environments: one pillar_root (all your envs are belong to base) (refs: #46309)
- **PR** #46631: (rallytime) Fix pillar unit test failures: file_roots and pillar_roots environments should be lists @ *2018-03-21 19:22:49 UTC*
  - **PR** #46629: (terminalmage) Fix symlink loop when file_roots/pillar_roots is a string instead of a list (refs: #46631)
  - **PR** #46569: (rallytime) [2018.3] Merge forward from 2017.7 to 2018.3 (refs: #46631)
  - **PR** #46309: (bdrung) Support dynamic pillar_root environment (refs: #46631)
  - **PR** #41423: (RichardW42) pillar: target's state list support wildcard in top.sls (refs: #46631)
  - 33af3cfc7c Merge pull request #46631 from rallytime/update-pillar-unit-tests
  - 0f728186aa Fix pillar unit test failures: file_roots and pillar_roots environments should be lists
- **ISSUE** #26450: (typeshige) file.copy: source file is not present. (refs: #46640)
- **PR** #46640: (terminalmage) Clarify the docs for the file.copy state @ *2018-03-21 19:14:50 UTC*
  - d329e7af78 Merge pull request #46640 from terminalmage/file.copy-docs

  - 480c5f8faa Clarify the docs for the file.copy state
- **PR** #46642: (vutny) [DOC] Unify cloud modules index header @ *2018-03-21 19:13:28 UTC*
  - ff40590c06 Merge pull request #46642 from vutny/doc-cloud-index
  - 51e6aa54a1 [DOC] Unify cloud modules index header
- **PR** #46619: (rallytime) [2017.7] Merge forward from 2017.7.5 to 2017.7 @ *2018-03-20 19:03:30 UTC*
  - 83ed40c06a Merge pull request #46619 from rallytime/merge-2017.7
  - bcbddf5d07 Merge branch '2017.7.5' into '2017.7'
- **PR** #46584: (twangboy) Fix issue LGPO issue @ *2018-03-20 17:48:33 UTC*
  - df12135439 Merge pull request #46584 from twangboy/lgpo-46568
  - 661017104b Detect disabled reg_multi_sz elements properly
- **PR** #46624: (twangboy) Fix a few inconsitencies in the installer script @ *2018-03-20 17:47:44 UTC*
  - 2fd3aa487c Merge pull request #46624 from twangboy/win_fix_installer
  - fa0b0efe46 Fix some installer script inconsistencies
- **ISSUE** #46552: (JeffLee123) State with require requisite executes despite onfail requisite on another state. (refs: #46571)
- **PR** #46571: (garethgreenaway) [2017.7] fixes to state.py @ *2018-03-20 13:40:04 UTC*
  - f038e3c452 Merge pull request #46571 from garethgreenaway/46552_onfail_and_require
  - 152c43c843 Accounting for a case when multiple onfails are used along with requires. Previously if you have multiple states using 'onfail' and two of those states using a 'require' against the first one state, the last two will run even if the 'onfail' isn't met because the 'require' is met because the first state returns true even though it didn't excute. This change adds an additional hidden variable that is used when checking requisities to determine if the state actually ran.
- **ISSUE** #46512: (blarghmatey) git.pull failing when run from the salt scheduler (refs: #46520)
- **PR** #46520: (gtmanfred) pass utils to the scheduler for reloading in modules @ *2018-03-20 13:35:49 UTC*
  - 2677330e19 Merge pull request #46520 from gtmanfred/2017.7
  - caefedc095 make sure utils is empty for pickling for windows
  - 2883548e6b pass utils to the scheduler for reloading in modules
- **ISSUE** #44299: (nhavens) 2017.7.2 breaks pkgrepo.managed yum repo comments (refs: #46531)
- **PR** #46531: (terminalmage) Fix regression in yumpkg._parse_repo_file() @ *2018-03-20 13:34:59 UTC*
  - 7bc3c2e588 Merge pull request #46531 from terminalmage/issue44299
  - b70c3389da Fix case where no comments specified
  - ce391c53f4 Add regression test for #44299
  - c3e36a6c94 Fix regression in yumpkg._parse_repo_file()
  - f0c79e3da3 Slight modification to salt.utils.pkg.rpm.combine_comments()
- **ISSUE** #46521: (dwoz) *–name* argument not honored for cloud test suite (refs: #46567)
- **PR** #46567: (dwoz) Honor named tests when running integration suites @ *2018-03-20 13:24:42 UTC*
  - b80edb5d26 Merge pull request #46567 from dwoz/runtest-n-wart
  - 3b6901e19d Honor named tests when running integration suites

- **PR** #46580: (twangboy) Clarify some issues with msu files in win_dism.py @ *2018-03-16 18:57:55 UTC*
  - 1dcd22e767 Merge pull request #46580 from twangboy/win_update_docs_dism
  - d52b99d7a3 Clarify some issues with msu files in win_dism.py
- **ISSUE** #46073: (layer3switch) salt 2017.7.3 grains metadata collection in AWS EC2 cause failure and nested iteration (refs: #46541)
- **PR** #46541: (gtmanfred) handle user-data for metadata grains @ *2018-03-15 17:21:31 UTC*
  - 0a68c22332 Merge pull request #46541 from gtmanfred/metadata
  - 19bd1d9db5 handle user-data for metadata grains
- **ISSUE** #46427: (wasabi222) cumulus linux should use systemd as a default service pkg instead of debian_service (refs: #46547)
- **PR** #46547: (garethgreenaway) [2017.7] Disable service module for Cumulus @ *2018-03-15 16:15:00 UTC*
  - 048b2ba3f6 Merge pull request #46547 from garethgreenaway/46427_service_module_cumulus
  - edd0b11447 Merge branch '2017.7' into 46427_service_module_cumulus
  - ea3c16080e Disable the *service* module on Cumulus since it is using systemd.
- **PR** #46548: (Ch3LL) profitbrick test: check for foo,bar username,password set in profitbrick config @ *2018-03-15 14:25:27 UTC*
  - 98e3260b9a Merge pull request #46548 from Ch3LL/profit_test
  - db96c4e72e check for foo,bar username,password set in profitbrick config
- **PR** #46549: (Ch3LL) Fix dimensionsdata test random_name call @ *2018-03-15 14:23:41 UTC*
  - 79f2a76609 Merge pull request #46549 from Ch3LL/dimension_test
  - bb338c464c Fix dimensionsdata test random_name call
- **PR** #46529: (gtmanfred) retry if there is a segfault @ *2018-03-13 22:41:54 UTC*
  - 083846fe0e Merge pull request #46529 from gtmanfred/kitchen
  - 50d6e2c7be retry if there is a segfault
- **PR** #46511: (rallytime) Back-port #45769 to 2017.7 @ *2018-03-13 17:08:52 UTC*
  - **PR** #45769: (Quarky9) Surpress boto WARNING during SQS msg decode in sqs_engine (refs: #46511)
  - 5cc11129f1 Merge pull request #46511 from rallytime/bp-45769
  - a8ffceda53 Surpress boto WARNING during decode, reference: https://github.com/boto/boto/issues/2965

28.2.27 Salt 2017.7.7 Release Notes

Version 2017.7.7 is a bugfix release for *2017.7.0*.

The `2017.7.7` release contains only a small number of fixes, which are detailed below.

This release fixes two critical issues.

The first is Issue #48038, which is a critical bug that occurs in a multi-syndic setup where the same job is run multiple times on a minion.

The second issue is #48130. This bug appears in certain setups where the Master reports a Minion time-out, even though the job is still running on the Minion.

Both of these issues have been fixed with this release.

Statistics

- Total Merges: **5**
- Total Issue References: **2**
- Total PR References: **6**
- Contributors: **3** (garethgreenaway, gtmanfred, rallytime)

Changelog for v2017.7.6..v2017.7.7

Generated at: 2018-06-17 19:26:52 UTC

- **ISSUE** #48130: (rmarchei) Minion timeouts with 2018.3.1 (refs: #48157)
- **PR** #48157: (gtmanfred) always listen when gathering job info @ *2018-06-17 19:04:09 UTC*
  - 8af4452134 Merge pull request #48157 from gtmanfred/2017.7.7
  - d8209e8a40 always listen when gathering job info
- **PR** #48140: (rallytime) Update man pages for 2017.7.7 @ *2018-06-14 21:22:43 UTC*
  - b98c52ee51 Merge pull request #48140 from rallytime/man-pages-2017.7.7
  - 8893bf0d4c Update man pages for 2017.7.7
- **PR** #48136: (gtmanfred) [2017.7.7] bootstrap kitchen branch tests with 2017.7.6 @ *2018-06-14 21:20:16 UTC*
  - baa0363336 Merge pull request #48136 from gtmanfred/2017.7.7
  - fce1c31146 bootstrap kitchen branch tests with 2017.7.6
- **PR** #48134: (rallytime) Add release notes file for 2017.7.7 @ *2018-06-14 16:31:34 UTC*
  - b0ba08f4d9 Merge pull request #48134 from rallytime/release-notes-2017.7.7
  - 217005b8f1 Add missing *v* for tag reference
  - d53569d1e3 Add release notes file for 2017.7.7
- **ISSUE** #48038: (austinpapp) jobs are not dedup'ing minion side (refs: #48075)
- **PR** #48098: (rallytime) Back-port #48075 to 2017.7.7 @ *2018-06-14 12:53:42 UTC*
  - **PR** #48075: (garethgreenaway) [2017.7] Ensure that the shared list of jids is passed (refs: #48098)
  - 084de927fe Merge pull request #48098 from rallytime/bp-48075-2017.7.7
  - e4e62e8b3a Ensure that the shared list of jids is passed when creating the Minion. Fixes an issue when minions are pointed at multiple syndics.

28.2.28 Salt 2017.7.8 Release Notes

Version 2017.7.8 is a security and bugfix release for *2017.7.0*.

Statistics

- Total Merges: **237**
- Total Issue References: **48**
- Total PR References: **279**
- Contributors: **52** (AVeenstra, Ch3LL, Circuitsoft, DmitryKuzmenko, KaiSforza, Martin819, OrlandoArcapix, UtahDave, Vaelatern, abednarik, asnell, b1naryth1ef, baniobloom, basepi, bdrung, beornf, bmcorser, bowmanjd-lms, damon-atkins, darkpixel, discogestalt, doesitblend, dqminh, dubb-b, dwoz, frankiexyz, frogunder, fzipi, garethgreenaway, grokrecursion, gtmanfred, jacksontj, jagguli, lejambon, lomeroe, lordcirth, lusche, mbunkus, meaksh, mirceaulinic, nbraud, pritambaral, ralex, rallytime, rmcintosh, slaws, terminalmage, twangboy, twellspring, wyardley, xetix, zer0def)

Security Fix

CVE-2018-15751 Remote command execution and incorrect access control when using salt-api.

CVE-2018-15750 Directory traversal vulnerability when using salt-api. Allows an attacker to determine what files exist on a server when querying /run or /events.

Credit and thanks for discovery and responsible disclosure: nullbr4in, xcuter, koredge, loupos, blackcon, Naver Business Platform

New win_snmp behavior

- *`win_snmp.get_community_names`* now returns the SNMP settings actually in effect on the box. If settings are managed via GroupPolicy, those settings will be returned. Otherwise, normal settings are returned.
- *`win_snmp.set_community_names`* now raises an error when SNMP settings are being managed by GroupPolicy.

Option Added to Disable Docker Mine Updates

When a docker container is added, removed, started, stopped, etc., the results of a `docker.ps verbose=True all=True host=True` are sent to the *mine*, to be used by *`mine.get_docker`*.

A new config option (*`docker.update_mine`*) has been added. When set to `False`, Salt will not send this information to the mine. This is useful in cases where sensitive information is stored in the container's environment.

Changelog for v2017.7.7..v2017.7.8

Generated at: 2018-09-04 21:09:41 UTC

- **PR** #49498: (rallytime) Pin CherryPy version to < 18.0.0 in requirements files for PY2 @ *2018-09-04 17:55:02 UTC*
  - 0d7b173b24 Merge pull request #49498 from rallytime/pin-cherrypy-2017.7.8
  - 81a6155b6b Pin CherryPy version to < 18.0.0 in requirements files for PY2
- **ISSUE** #47087: (darkpixel) How do I stop the prompt "Permission denied for host <hostname>, do you want to deploy the salt-ssh key? (password required):"? (refs: #47100, #‘saltstack/salt‘#47100‘_‘_)
  - **PR** saltstack/salt#47100: (gtmanfred) Allow for not being prompted to supply a password to deploy keys to a… (refs: #49461)

- **PR** #49466: (rallytime) Back-port #49461 to 2017.7.8 @ *2018-08-31 15:44:11 UTC*
  - **PR** #49461: (gtmanfred) Revert "Allow for not being prompted to supply a password to deploy keys to a..." (refs: #49466)
  - 52ab2c0574 Merge pull request #49466 from rallytime/bp-49461
  - 58428003b1 Revert "Allow for not being prompted to supply a password to deploy keys to a..."
- **PR** #49284: (twangboy) Fix path to libsodium tarball @ *2018-08-23 17:19:24 UTC*
  - 1bbe7df6c4 Merge pull request #49284 from twangboy/fix_installer_osx
  - a112eaa597 Fix path to libsodium tarball
- **PR** #49272: (twangboy) Add 64 bit binaries for KB2999226 to the x86 installer @ *2018-08-23 14:27:09 UTC*
  - e0e8779985 Merge pull request #49272 from twangboy/fix_installer_more
  - a8f054b2d2 Add 64bit binaries for KB2999226 to the x86 installer
- **PR** #49218: (twangboy) Fix Windows and OSX installers @ *2018-08-21 01:10:30 UTC*
  - 0eb6ddf0e8 Merge pull request #49218 from twangboy/fix_installer
  - e8a1d2f772 Add more descriptive error when KB not found
  - 71737ea687 Suppress all ui on vcredist installation
  - 7dae9bb2a1 Fix Windows and OSX installers
- **PR** #49116: (twangboy) Fix windows installer script @ *2018-08-14 17:09:35 UTC*
  - e484f261ee Merge pull request #49116 from twangboy/fix_installer
  - 1227095c5d Add nonfatal switch to VCRedist
- **PR** #49113: (Ch3LL) Add changelog to 2017.7.8 release notes @ *2018-08-14 15:03:16 UTC*
  - 0b9f2f8884 Merge pull request #49113 from Ch3LL/rn_7.8
  - f6b70bb653 Add changelog to 2017.7.8 release notes
- **PR** #49051: (rallytime) Back-port #49046 to 2017.7.8 @ *2018-08-10 17:19:18 UTC*
  - **PR** #49046: (garethgreenaway) [2017.7] Another fix to tests/integration/modules/test_service.py (refs: #49051)
  - 01cf2c71ff Merge pull request #49051 from rallytime/bp-49046
  - fc0817cb35 The osfullname grain differs when using Python2 vs Python3, swapping this out for the "OS" grain which is consistent.
- **PR** #49044: (Ch3LL) increase timeout on vultr cloud integration test @ *2018-08-10 01:28:17 UTC*
  - 189e28691f Merge pull request #49044 from Ch3LL/vultr_test
  - 0dedfae4d1 increase timeout on vultr cloud integration test
- **PR** #49042: (rallytime) Mark a jinja template test as flaky @ *2018-08-09 21:05:54 UTC*
  - c606a32cf2 Merge pull request #49042 from rallytime/flaky-jinja-test
  - a43d9b4ba6 Mark a jinja template test as flaky
- **PR** #49041: (Ch3LL) [2017.7.8] backport PR #48212 @ *2018-08-09 21:05:27 UTC*
  - **PR** #48212: (Ch3LL) Fix python3 ec2 salt-cloud TypeError when installing salt (refs: #49041)
  - 6415b6f73b Merge pull request #49041 from Ch3LL/ec2_fix

  - cf7f2459b8 [2017.7.8] backport PR #48212
- **PR** #49030: (rallytime) Update netapi client tests @ *2018-08-09 17:11:14 UTC*
  - 1db036406b Merge pull request #49030 from rallytime/update-client-tests
  - f08ee6c6ae Update netapi client tests
- **PR** #49024: (rallytime) Fix test error in test_compiler jinja check @ *2018-08-09 12:43:23 UTC*
  - **PR** #49002: (rallytime) Skip test_jinja_deep_error on Debian 8 (refs: #49024)
  - f73ba21bc7 Merge pull request #49024 from rallytime/fix-deb-test
  - a9c16d9137 Fix test error in test_compiler jinja check
- **PR** #49020: (rallytime) Make grains integration test more robust @ *2018-08-08 20:59:18 UTC*
  - 6b6d68c615 Merge pull request #49020 from rallytime/fix-grains-test
  - f72a3ac6be Make grains integration test more robust
- **PR** #49002: (rallytime) Skip test_jinja_deep_error on Debian 8 (refs: #49024) @ *2018-08-08 19:18:39 UTC*
  - 92d6c25c7f Merge pull request #49002 from rallytime/skip-jinja-deep-error-test
  - 23b66ef8bb Skip test_jinja_deep_error on Debian 8
- **PR** #48999: (rallytime) Update expected return value in boto test @ *2018-08-08 14:42:15 UTC*
  - 41d9f11eb3 Merge pull request #48999 from rallytime/fix-boto-test
  - d0136b1be5 Update expected return value in boto test
- **PR** #48976: (rallytime) Skip unreliable tornado tests @ *2018-08-07 18:55:25 UTC*
  - a1e54634dc Merge pull request #48976 from rallytime/tornado
  - 0bd838ab6c Skip unreliable tornado tests
- **PR** #48979: (rallytime) Back-port #48959 to 2017.7.8 @ *2018-08-07 18:11:53 UTC*
  - **PR** #48959: (rallytime) Mark some more tests as flaky (refs: #48979)
  - 22713be9c1 Merge pull request #48979 from rallytime/bp-48959
  - aaf986d728 Mark one grains test as flaky & convert to pytest notation
  - e7e5abcf48 Mark 2 matcher tests as flaky
- **PR** #48970: (Ch3LL) [2017.7.8] Backport #48962 @ *2018-08-07 15:01:57 UTC*
  - **PR** #48962: (garethgreenaway) [2017.7] Fixing tests/integration/modules/test_service.py (refs: #48970)
  - 03aa0e49b0 Merge pull request #48970 from Ch3LL/back_48962
  - 3ce1b8a3c9 Update the elif block to only be true for versions below Debian 9.
- **PR** #48968: (rallytime) Update man pages for 2017.7.8 release @ *2018-08-07 14:29:08 UTC*
  - e6cea5e3c7 Merge pull request #48968 from rallytime/man-pages
  - 64fe3be41a Update man pages for 2017.7.8 release
- **PR** #48950: (KaiSforza) Added a quote to kitchen Jenkinsfiles @ *2018-08-06 14:29:08 UTC*
  - 848d583438 Merge pull request #48950 from KaiSforza/kitchenfix-2017.7
  - 5242cb143a Added a quote to kitchen Jenkinsfiles
- **PR** #48943: (rallytime) Mark some shell and runner integration tests as flaky @ *2018-08-06 13:08:14 UTC*

  - 928d688d65 Merge pull request #48943 from rallytime/flaky-tests
  - 668da57ab9 Mark some shell and runner integration tests as flaky
- **PR** #48940: (rallytime) Back-port #48852 to 2017.7 @ *2018-08-05 21:32:26 UTC*
  - **PR** #48852: (KaiSforza) Record all the artifacts from the build (refs: #48940)
  - cd42510d3a Merge pull request #48940 from rallytime/bp-48852
  - fa4ef92e79 Record all the artifacts from the build
- **PR** #48935: (garethgreenaway) [2017.7] Fix to test_pkg.test_pkg_015_installed_held @ *2018-08-05 19:47:39 UTC*
  - 43649a68be Merge pull request #48935 from garethgreenaway/1045_test_pkg_015_installed_held_centos
  - 0bb10107b6 Merge branch '2017.7' into 1045_test_pkg_015_installed_held_centos
  - 24d5e6a22f Fixing the test_pkg_015_installed_held test to be able to successfully run on CentOS
- **ISSUE** #47087: (darkpixel) How do I stop the prompt "Permission denied for host <hostname>, do you want to deploy the salt-ssh key? (password required):"? (refs: #47100)
- **PR** #47100: (gtmanfred) Allow for not being prompted to supply a password to deploy keys to a… @ *2018-08-05 19:12:58 UTC*
  - 2421e2a570 Merge pull request #47100 from gtmanfred/ssh
  - 5b443af7ae add key-deploy test
  - a131c9beeb Allow for not being prompted to supply a password to deploy keys to a minion with salt-ssh
- **ISSUE** #48865: (mts-avco) Windows packages are hidden in pkg.list_pkgs if they have no DisplayVersion (refs: #48891)
- **PR** #48891: (damon-atkins) win_pkg: Fix issue introduced in Jan 2018 DisplayVersion missing should result… @ *2018-08-05 18:53:47 UTC*
  - d541bd6446 Merge pull request #48891 from damon-atkins/2017.7_win_pkg.list_pkgs_not_found
  - a4af1dbfb1 Fix win_pkg issues introduced Jan 2018. If DisplayVersion does not exist it should return version as "Not Found"
- **ISSUE** #27056: (oogali) pkgng provider on FreeBSD does not do BATCH=yes (refs: #48730, #29909)
- **PR** #48896: (rallytime) Back-port #48730 to 2017.7 @ *2018-08-05 18:20:40 UTC*
  - **PR** #48730: (fzipi) Fix batch install on FreeBSD using pkgng (refs: #48896)
  - **PR** #29909: (abednarik) FreeBSD pkgng fix for non-interactive install. (refs: #48730)
  - **PR** #29907: (bmcorser) Presumably what was meant (refs: #48730)
  - 5f6a56f5dc Merge pull request #48896 from rallytime/bp-48730
  - 57aa204c9d Merge branch '2017.7' into bp-48730
  - 4995922584 Forgot variable in signature
  - 0503bc18b6 Fix batch install using pkgng
- **PR** #48933: (garethgreenaway) [2017.7] Fix to test_service_disable_doesnot_exist @ *2018-08-05 14:29:43 UTC*
  - 0c64bba865 Merge pull request #48933 from garethgreenaway/1022_debian_8_failing_service-test
  - 280d1d2ad2 Fixing failing test, integration.modules.test_service.ServiceModuleTest.test_service_disable_doesnot_exist, on Debian 8 and higher.
- **PR** #48922: (rallytime) Update backticks on job_cache docs @ *2018-08-03 21:05:46 UTC*

  - 0c3d2c6a09 Merge pull request #48922 from rallytime/cache-doc-error
  - 8ca89df7e8 Update backticks on job_cache docs
- **PR** #48866: (Ch3LL) Add cmd module integration tests for windows and fix space in path issue @ *2018-08-03 21:03:33 UTC*
  - 8d1fc4f8e5 Merge pull request #48866 from Ch3LL/cmd_win_tests
  - 905da13653 Merge branch '2017.7' into cmd_win_tests
- **PR** #48920: (rallytime) Back-port #48904 to 2017.7 @ *2018-08-03 15:17:07 UTC*
  - **PR** #48904: (KaiSforza) No rehashing in parallel (refs: #48920)
  - 57d58e7541 Merge pull request #48920 from rallytime/bp-48904-2017.7
  - a55f92954a No rehashing in parallel
- **PR** #48876: (Vaelatern) Make IP_LEARNING optional @ *2018-08-02 19:30:24 UTC*
  - e79ccb35b1 Merge pull request #48876 from Vaelatern/make-network-learning-optional
  - a4905b0e5d Make IP_LEARNING optional
- **PR** #48890: (Ch3LL) add sleep in test_cmd integration test @ *2018-08-02 19:15:08 UTC*
  - f8bfab4f9f Merge pull request #48890 from Ch3LL/cmd_test
  - acda4ed9ab add sleep in test_cmd integration test
- **PR** #48885: (rallytime) Mark shadow module integration test as flaky @ *2018-08-02 19:04:35 UTC*
  - 52722f6ded Merge pull request #48885 from rallytime/flaky-tests-shadow
  - da871a2d57 Mark shadow module integration test as flaky
- **PR** #48884: (rallytime) Separate compound matcher tests into individual tests @ *2018-08-02 19:04:03 UTC*
  - bf0895656e Merge pull request #48884 from rallytime/flaky-tests-matchers
  - 38d9eae537 Merge branch '2017.7' into flaky-tests-matchers
- **PR** #48868: (terminalmage) Fix race when SIGTERM/SIGINT received while lazyloading a module @ *2018-08-02 17:13:34 UTC*
  - a567666938 Merge pull request #48868 from terminalmage/fix-loader-race
  - 5f1169b9a2 Fix race when SIGTERM/SIGINT received while lazyloading a module
- **PR** #48883: (terminalmage) Fix failing git worktree tests @ *2018-08-02 16:51:40 UTC*
  - 0ca0b6f2f2 Merge pull request #48883 from terminalmage/salt-jenkins-1023
  - c61f75cb50 Fix failing git worktree tests
    - 3660dff13c Separate compound matcher tests into individual tests
- **PR** #48869: (Ch3LL) Catch socket.error exception in testprogram @ *2018-08-02 14:19:22 UTC*
  - a84f5155a1 Merge pull request #48869 from Ch3LL/mac_shell_tests
  - 3734b1ec89 Catch socket.error exception in testprogram
- **PR** #48867: (rallytime) Skip unreliable tornado test. @ *2018-08-01 21:07:07 UTC*
  - 7f56b8bf44 Merge pull request #48867 from rallytime/skip-tornado-test
  - 7320aa9104 Skip unreliable tornado test.
  - 3be11e06fe Add docs for new escape kwarg

  - 391bb8a411 use a specific path for just the cmd._run call
  - 62c66ba489 make sure we lower the check on shell
  - 9312a993a5 Add cmd module integration tests for windows and fix space in path issue
- **PR** #48853: (rallytime) Back-port #48850 to 2017.7 @ *2018-07-31 20:21:49 UTC*
  - **PR** #48850: (rallytime) Skip tests in integration.shell.test_master (refs: #48853)
  - 05f2d65de3 Merge pull request #48853 from rallytime/bp-48850
  - 3c33ee554a Skip tests in integration.shell.test_master
- **ISSUE** #46689: (mxork) pkg.installed: hold: True not applied to a package which is already installed. (refs: #48426)
- **PR** #48426: (garethgreenaway) [2017.7] fixes to states/pkg.py @ *2018-07-31 20:18:05 UTC*
  - 8a1285239a Merge pull request #48426 from garethgreenaway/46689_fixing_pkg_held_when_package_is_installed
  - 9b0f5dd212 Fixing identation, removing some unnecessary conditionals.
  - 727964ab55 One last cleanup.
  - 11cb86e6eb General cleanup in pkg state, reducing duplicate code. Fixing the requires_salt_modules decorator, sys.doc was returning too much information for the event to handle. This change specifically calls sys.doc with the module name.
  - 16fb6ae635 Make sure pkg.hold and pkg.unhold are available before running the test.
  - 998651102d Fixing a situation when a package is already installed via salt or manually and a state attempts to set that package to be held. Previously the holding/unholding logic was only being run against packages that were being installed. This change moves the holding logic outside and runs it against all desired packages. Adding a new test to test holding logic.
- **ISSUE** #47689: (OrlandoArcapix) Poor performance of pip.installed when given a list of packages (refs: #47734)
- **PR** #47734: (OrlandoArcapix) #47689 improve run-speed of pip package state @ *2018-07-31 19:15:35 UTC*
  - c8e69431ff Merge pull request #47734 from OrlandoArcapix/Issue47689-pip-state-performance
  - 662bd1f780 Merge branch '2017.7' into Issue47689-pip-state-performance
  - 66936b4f41 Changed string comparison in pip test to match new confirmation string - ref PR #47734.
  - bb5939d6ef Merge branch '2017.7' into Issue47689-pip-state-performance
  - d6a49ae41c Merge branch '2017.7' into Issue47689-pip-state-performance
  - d4083fc9d1 Merge branch 'Issue47689-pip-state-performance' of github.com:OrlandoArcapix/salt into Issue47689-pip-state-performance
    - 779b5fa785 Merge branch '2017.7' into Issue47689-pip-state-performance
  - f3653349ab Removed whitespaces at end of added comments lines
  - db11f2ff4b Merge branch '2017.7' into Issue47689-pip-state-performance
  - eac0178de2 Ref: #47689 - document additional kwarg passed to pip._check_if_installed function
  - 0d19803106 Merge branch '2017.7' into Issue47689-pip-state-performance
  - d3678bf2f3 #47689 fix lint errors
  - 4fec8f6bcc #47698 improve run-speed of pip package state checks by only loading the current package list once when checking multiple packages

- **PR** #48844: (AVeenstra) Fixed Python 3 incompatibility in methods in nilrt_ip and debian_ip. @ *2018-07-31 17:20:27 UTC*
  - 83a5b3cc47 Merge pull request #48844 from AVeenstra/fix-python3-incompatibility
  - f238779a62 Merge branch '2017.7' into fix-python3-incompatibility
  - 6b1805afc6 Fixed Python 3 incompatibility in methods in nilrt_ip and debian_ip.
- **ISSUE** #48637: (slaws) state.file.retention_schedule does not ignore relative directory when using getmtime() (refs: #48662)
- **PR** #48662: (slaws) Excluding relative dirs in state.file.retention_schedule @ *2018-07-31 16:15:09 UTC*
  - 136ff6735a Merge pull request #48662 from slaws/fix-retention-schedule-48637
  - 3c8f5f5b3d pylint fix
  - 5539eff39e Excluding relative dirs in state.file.retention_schedule
- **ISSUE** #48818: (guettli) Broken Link in Docs (refs: #48840)
- **PR** #48840: (gtmanfred) fix links in pkg doc. @ *2018-07-31 16:13:00 UTC*
  - 1fc04f281b Merge pull request #48840 from gtmanfred/docs
  - 8d2d268c4a fix links in pkg doc.
- **PR** #48834: (gtmanfred) read output of stringio if it is readable @ *2018-07-31 13:40:48 UTC*
  - 0a19f845ea Merge pull request #48834 from gtmanfred/slsutil
  - f9441d2bef read output of stringio if it is readable
- **PR** #48788: (Ch3LL) Add timzeone windows integration tests and fix get_zone @ *2018-07-27 20:14:30 UTC*
  - 2e00939a6e Merge pull request #48788 from Ch3LL/timezone_windows
  - de95a6a215 add unused import to timezone test file
  - 22e424859e add unused import to import
  - 0840fc3117 disable pylint import error
  - f09d1a2c7e Add timzeon windows integration tests and fix get_zone
- **PR** #48742: (frogunder) add windows integration tests for states.file @ *2018-07-27 13:57:27 UTC*
  - 8e61f3dce0 Merge pull request #48742 from frogunder/45014
  - f08058b043 Merge branch '45014' of https://github.com/frogunder/salt into 45014
    - 757fde70c6 Merge branch '2017.7' into 45014
  - 33c20c1ec0 fix tests
  - a7a914060d add windows integration tests for states.file
- **ISSUE** #48417: (samilaine) Contribution Documentation Refers to Incorrect Pylintrc (refs: #48795)
- **PR** #48795: (rallytime) Update linting docs to contain .testing.pylintrc use @ *2018-07-27 13:34:13 UTC*
  - d74f47b340 Merge pull request #48795 from rallytime/fix-48417
  - 5123b17ffd Update linting docs to contain .testing.pylintrc use
- **PR** #48789: (rallytime) Back-port #48783 to 2017.7 @ *2018-07-26 21:47:49 UTC*
  - **PR** #48783: (KaiSforza) Only run lint checks against changed files (refs: #48789)
  - **PR** #48610: (gtmanfred) only run pylint on files that change (refs: #48783)

  - bbea9ae936 Merge pull request #48789 from rallytime/bp-48783
  - 682a05bebe Threshold was wrong
  - d4ca0e3a97 test: except for OSError only
  - 4547231909 Only run lint checks against changed files
- **PR** #48731: (zer0def) Fixed *enable_vnc* runner arg being passed into *seed_cmd* module arg in *virt.init*. @ *2018-07-26 21:13:59 UTC*
  - 41464d4b39 Merge pull request #48731 from zer0def/virt-runner-init-args
  - a1fa081ad0 Documentation to missing parameters in virt.init runner.
  - 365ebdf539 Fixed *enable_vnc* runner arg being passed into *seed_cmd* module arg in *virt.init*.
- **PR** #48749: (Ch3LL) Update Saltstack Logo banner on docs.saltstack.com @ *2018-07-25 13:05:04 UTC*
  - 169afea16a Merge pull request #48749 from Ch3LL/logo_docs
  - 73b1fc29f0 Update Saltstack Logo banner on docs.saltstack.com
- **PR** #48743: (rallytime) Update DOCBANNER with new SaltConf18 image @ *2018-07-24 16:46:19 UTC*
  - 781c6a6c36 Merge pull request #48743 from rallytime/saltconf-ad-2017.7
  - 6df8fd8652 Update DOCBANNER with new SaltConf18 image
- **PR** #48720: (Ch3LL) Fix test_managed_source_hash_indifferent_case on macosx to correct tmp path @ *2018-07-23 22:32:56 UTC*
  - ce4e22224e Merge pull request #48720 from Ch3LL/mac_file_path
  - 01d25fae19 Fix test_managed_source_hash_indifferent_case on macosx to correct tmp path
- **PR** #48719: (Ch3LL) Enable service if disabled before running state service tests @ *2018-07-23 21:01:14 UTC*
  - 4b9f037d43 Merge pull request #48719 from Ch3LL/service_mac_state
  - f44a2fc349 Enable service if disabled before running state service tests
- **PR** #48715: (rallytime) [2017.7] Mark some tornado tests as flaky @ *2018-07-23 17:27:13 UTC*
  - b2431eab10 Merge pull request #48715 from rallytime/flaky-tests
  - 7332cce567 [2017.7] Mark some tornado tests as flaky
- **PR** #48672: (frogunder) add service enabled test @ *2018-07-23 15:38:49 UTC*
  - 801eae3b8d Merge pull request #48672 from frogunder/45012
  - 0747f2e58a add service enabled test
- **PR** #48691: (Ch3LL) Add windows pkg module integration tests @ *2018-07-22 20:01:30 UTC*
  - 689c231d2b Merge pull request #48691 from Ch3LL/win_repo_pkg_test
  - 4b7d6d80c5 Remove unnecessary jinja in curl.sls file
  - 2bedadfadb Add windows pkg module integration tests
- **ISSUE** #48627: (nbraud) acme module's group parameter is non-functional (refs: #48635)
- **ISSUE** #48626: (nbraud) acme module fails to set file permissions if the certificate is already present (refs: #48635)
- **PR** #48635: (nbraud) Bug fixes in the acme module & state @ *2018-07-22 19:53:49 UTC*
  - 83e4bba916 Merge pull request #48635 from nbraud/acme

  - 3673bae9de modules/acme: explicitely ignore the *perms* return value
  - 1800a231e8 Fixup some schema expectations
  - 8c718cb417 acme: Make the private key mode configurable
  - 917dea6761 modules/acme: Use file.check_perms ret-morphing powers
  - d2241ceb2d module/acme: Do not exit early when the certificate already exists
  - 98af0db826 modules/acme: Set the private key filemode to 0640
- **PR** #48345: (twangboy) Fix behavior of powercfg module and state @ *2018-07-20 17:37:13 UTC*
  - 85991680c8 Merge pull request #48345 from twangboy/fix_48169
  - ead19725b6 Merge branch '2017.7' into fix_48169
  - 653fbcb383 Fix some docs errata
  - 8898e5ff11 Add warn_until Fluorine
  - 707906ac15 Fix unit.state.test_powercfg
  - 32c5014eb6 Fix unit tests
  - 0d9c56e540 Add some more logging
  - b20453de9f Use minutes to set
  - 7dc7eb11c2 Fix documentation to denote seconds when setting
- **PR** #48656: (Ch3LL) Add windows ip module integration tests @ *2018-07-20 14:57:45 UTC*
  - 96447ce541 Merge pull request #48656 from Ch3LL/windows_ip_mod
  - 90c3f568b1 Add windows ip module integration tests
- **PR** #48638: (twangboy) Remove vcredist 2015 from Py3 installer @ *2018-07-20 14:56:40 UTC*
  - 0c5fff3dc8 Merge pull request #48638 from twangboy/fix_vcredist
  - e30d17099d Use goto instead of if statement
  - e1042fa084 Remove vcredist for Py3
- **PR** #48664: (Ch3LL) [2017.7] Fix service.disabled test for macosx @ *2018-07-20 14:32:56 UTC*
  - 1ebd96d909 Merge pull request #48664 from Ch3LL/srv_disable_mac
  - f1fbfad387 move the disable call up in try block
  - f60d21bda4 reverse assertion order for service disable test
  - 3727d1b3b9 switch try/except to match 2018.3
  - fb953c2369 [2017.7] Fix service.disabled test for macosx
- **PR** #48625: (Ch3LL) Follow up to PR #48555 @ *2018-07-20 14:27:55 UTC*
  - **PR** #48555: (Ch3LL) Fix state.sls_id not running on ssh minion (refs: #48625)
  - 3da3cf2f3f Merge pull request #48625 from Ch3LL/ssh_state
  - f590eb2b02 Update state.py
  - 9790ee3d0d Follow up to PR #48555
- **PR** #48673: (Ch3LL) Use different pub and ret ports for testprogram integration tests @ *2018-07-20 14:11:51 UTC*

  - 33812f78f1 Merge pull request #48673 from Ch3LL/mac_port
  - 87dd85a220 Use different pub and ret ports for testprogram integration tests
- **PR** #48675: (Ch3LL) Fix mac service.disable tests @ *2018-07-20 14:09:41 UTC*
  - 5f6a7c4d89 Merge pull request #48675 from Ch3LL/mac_disable
  - c78efab828 Fix mac service.disable tests
- **ISSUE** #43138: (F30) npm.installed mistakenly throws error for packages which are "installed via remote" (refs: #48492, #48658)
- **PR** #48658: (wyardley) Improve handling of json output (#43138) @ *2018-07-20 14:08:34 UTC*
  - **PR** #48492: (wyardley) Improve identifying json out in npm module (#43138) (refs: #48658)
  - 93d2f51d2b Merge pull request #48658 from wyardley/wyardley-npm-json-output-2017
  - 7ff3c9c5ff Improve handling of npm json output (#43138)
- **ISSUE** #48677: (OrlandoArcapix) npm.bootstrap does not return True (clean) with test=true and no changes (refs: #48678)
- **PR** #48678: (OrlandoArcapix) Fix for issue #48677 - return clean npm.bootstrap on no changes @ *2018-07-20 14:07:22 UTC*
  - 3e293b0513 Merge pull request #48678 from OrlandoArcapix/fix-npm-dryrun-test
  - 851a404f6b Fix for issue #48677 - return True when no changes are to be made with npm.bootstrap with test=true
- **ISSUE** #46884: (alexandergraul) salt.utils.rsax931._init_libcrypto() fails to initialize libopenssl1_1-1.1.0h (refs: #48580)
- **PR** #48580: (rallytime) Don't error on retcode 0 in libcrypto.OPENSSL_init_crypto call @ *2018-07-19 19:21:13 UTC*
  - **PR** #37772: (bdrung) Support initializing OpenSSL 1.1 (refs: #48580)
  - 07a1f6520f Merge pull request #48580 from rallytime/fix-46884
  - 736b382e91 Don't error on retcode 0 in libcrypto.OPENSSL_init_crypto call
- **PR** #48628: (terminalmage) Fix NameError in testinfra module @ *2018-07-18 21:34:03 UTC*
  - 9874429741 Merge pull request #48628 from terminalmage/testinfra
  - 5ace9f01ec Fix NameError in testinfra module
- **PR** #48651: (gtmanfred) [2017.7] handle pyzmq for python3.4 @ *2018-07-18 17:34:41 UTC*
  - 0d4c80205f Merge pull request #48651 from gtmanfred/pylint-2017.7
  - b6ee52f859 handle pyzmq for python3.4
- **PR** #48647: (gtmanfred) [2017.7] disable checks on pylint @ *2018-07-18 16:21:22 UTC*
  - 160ae29cf2 Merge pull request #48647 from gtmanfred/pylint-2017.7
  - ff818c4ca2 disable checks on pylint
- **ISSUE** #46924: (chschmitt) Stale custom roster from __pycache__ executed instead of changed .py file (refs: #48593)
- **PR** #48593: (pritambaral) Fix importlib pyc loading order @ *2018-07-17 21:06:28 UTC*
  - 47b1032efa Merge pull request #48593 from pritambaral/fix/2017.7-importlib-pyc-loading-order

  - add7894dee loader: Fix suffix order when importlib is used
- **PR** #48630: (dubb-b) Adding options to Jenkins pipline builds @ *2018-07-17 20:16:23 UTC*
  - 9da7b2ec8d Merge pull request #48630 from dubb-b/pipeline-updates
  - 8594a8dd05 Adding PY_COLORS=1 as PY_COLORS = 1 instead
  - 314b0e3599 Adding PY_COLORS=1 for python programs to use ANSI Colors
  - b705e8f7a5 Adding correct spacing to options section
  - 9d8a7e07db Adding options to Jenkins pipline builds
  - **PR** saltstack/salt#48610: (gtmanfred) only run pylint on files that change (refs: #48633)
- **PR** #48633: (gtmanfred) Revert "only run pylint on files that change" @ *2018-07-17 18:44:03 UTC*
  - a8ae2adf64 Merge pull request #48633 from saltstack/revert-48610-2017.7
  - bab4a769d4 Revert "only run pylint on files that change"
- **PR** #48614: (rallytime) Back-port #48562 to 2017.7 @ *2018-07-17 15:04:04 UTC*
  - **PR** #48562: (basepi) Add timeouts to all s3 queries (refs: #48614)
  - 6e32bb7f74 Merge pull request #48614 from rallytime/bp-48562
  - cb654bbf2c Add timeouts to all s3 queries
- **ISSUE** #48415: (doesitblend) Event.send short-circuiting in multi-master mode (refs: #48588)
- **PR** #48588: (garethgreenaway) [2017.7] event send multi master @ *2018-07-17 10:31:20 UTC*
  - 1b6e6388f8 Merge pull request #48588 from garethgreenaway/48415_event_send_multi_master
  - fab25af1a9 Adding some quick documention about why we are setting ret=True following the channel.send.
  - bf78f4b188 If the channel send is sucessful and does not raise an exception, we set ret to True, in case a previous exception from a previous channel send to another master has sent it to False.
  - 8d1551c5fb When using Salt multi-master, if we encouter a salt master that has not accepted the minion key yet we should not exit right away, rather continue on and try the next salt master available in the list.
- **PR** #48610: (gtmanfred) only run pylint on files that change (refs: #48783) @ *2018-07-17 01:36:50 UTC*
  - 24ffda49ba Merge pull request #48610 from gtmanfred/2017.7
  - 779d1a3dcb only run pylint on files that change
- **PR** #48584: (Ch3LL) Fix grp import for mac in test_user integration test @ *2018-07-16 19:28:13 UTC*
  - 5391dd0a8d Merge pull request #48584 from Ch3LL/mac_user_grp
  - 585ee9db90 Fix grp import for mac in test_user integration test
- **PR** #48555: (Ch3LL) Fix state.sls_id not running on ssh minion (refs: #48625) @ *2018-07-16 13:24:41 UTC*
  - 61572b6780 Merge pull request #48555 from Ch3LL/ssh_id
  - f69932f506 fix pylint
  - 70e36764ee Add more tests for salt-ssh state.sls_id
  - dab80e805c Fix state.sls_id to run on ssh minion and not master
- **PR** #48583: (Ch3LL) Add flaky decorator to mac_system and mac_timezone tests @ *2018-07-13 19:56:40 UTC*

  - aa6dcf39e8 Merge pull request #48583 from Ch3LL/mac_flaky_tests
  - 4ba2299a87 import flaky decorator on mac tests
  - 811220b41e Add flaky decorator to mac_system and mac_timezone tests
- **PR** #48534: (xetix) Fix behaviour of function latest_version in zypper module when multip… @ *2018-07-13 19:56:05 UTC*
  - 6973152057 Merge pull request #48534 from xetix/fix-zypper-latest_version
  - 9985f0b4c1 Lint: remove extra blank line
  - 5fbead8a36 Merge branch '2017.7' into fix-zypper-latest_version
  - 56ac449271 Merge branch 'fix-zypper-latest_version' of https://github.com/xetix/salt into fix-zypper-latest_version
    - 44e87f5a0f Adding testcase for this fix.
    - b354c6863c Fix behaviour of function latest_version in zypper module when multiple packages are passed to function. Function now properly return dict with empty string as version if latest packages is already installed, and multiple packages are passed to function
  - db35d0c1e1 Adding testcase for this fix.
  - d2513757ed Fix behaviour of function latest_version in zypper module when multiple packages are passed to function. Function now properly return dict with empty string as version if latest packages is already installed, and multiple packages are passed to function
- **PR** #48582: (dwoz) Finally fix prepend for real @ *2018-07-13 18:45:35 UTC*
  - 10124034cb Merge pull request #48582 from dwoz/test_prepend_fix
  - f37571e0bd Merge branch '2017.7' into test_prepend_fix
- **PR** #48564: (dwoz) Fix failing prepend test @ *2018-07-13 14:12:47 UTC*
  - 13f67335f3 Merge pull request #48564 from dwoz/test_prepend_fix
    - 136ddf5f54 Finally fix prepend for real
  - 66b25e65bf Fix failing prepend test
- **PR** #48558: (dwoz) Remove which mock to get the test passing @ *2018-07-12 19:04:12 UTC*
  - b8ce27729f Merge pull request #48558 from dwoz/test_file_fix
  - c858bf477f Remove which mock to get the test passing
- **PR** #48552: (KaiSforza) Set up junit in jenkins @ *2018-07-12 16:23:33 UTC*
  - a15c65202d Merge pull request #48552 from KaiSforza/jenkins-junit-2017
  - 72b1830974 Set up junit in jenkins
- **PR** #48550: (gtmanfred) add posargs to tox pylint @ *2018-07-12 14:34:25 UTC*
  - f73108026e Merge pull request #48550 from gtmanfred/2017.7
  - d5c603d9b6 add posargs to tox pylint
- **PR** #48535: (asnell) Update manage_file helptext @ *2018-07-12 13:34:47 UTC*
  - 113b0426f8 Merge pull request #48535 from asnell/asnell-patch-2
  - d328f6e43d Update manage_file helptext
  - **PR** saltstack/salt#48525: (gtmanfred) add kazoo for testing zookeeper. (refs: #48527)

- **PR** #48527: (gtmanfred) Revert "add kazoo for testing zookeeper." @ *2018-07-11 17:29:16 UTC*
  - 0c0c05c2bc Merge pull request #48527 from saltstack/revert-48525-2017.7
  - 18d06c1a93 Revert "add kazoo for testing zookeeper."
- **PR** #48526: (twangboy) Rollback python-certifi-win32 @ *2018-07-11 17:26:56 UTC*
  - 05bad3e71c Merge pull request #48526 from twangboy/rollback_certifi-win32
  - 361b3cdc2f Rollback python-certifi-win32
- **PR** #48521: (Martin819) Back-port #48222 to 2017.7 @ *2018-07-11 16:42:51 UTC*
  - **PR** #48222: (Martin819) Fix GlusterFS module for version 4.0 and above (refs: #48521)
  - db066effe4 Merge pull request #48521 from Martin819/2017.7
  - d5d01f6698 Fix for GlusterFS 4.0 and above
- **PR** #48525: (gtmanfred) add kazoo for testing zookeeper. @ *2018-07-11 16:09:40 UTC*
  - ed0bd2bbec Merge pull request #48525 from gtmanfred/2017.7
  - d7a6bff2b2 add kazoo for testing zookeeper.
- **ISSUE** #48277: (dvenckus) init.sls with included states fails with more than one, 'Template was specified incorrectly: False' (refs: #48388)
- **ISSUE** #46986: (github-abcde) opts file_roots gets overwritten with pillar_roots in orchestration run (refs: #48388)
- **PR** #48388: (garethgreenaway) [2017.7] reset file_roots for renderers after compile_pillar @ *2018-07-11 13:13:45 UTC*
  - ad5a959ab9 Merge pull request #48388 from garethgreenaway/48277_2017_7_file_roots_wrong
  - 6f11da35a7 Fixing a few things in the new test.
  - cf747b14ec Fixing lint issue
  - 16d36c79c8 When pillar items are compiled a new render is instantiated but the file_roots is the pillar_roots. This change forces the __opts__['file_roots'] to be set to what is set in actual_file_roots for all renderers once compile_pillar has finished. Adding a test when this situation is run via a orchestration runner.
- **PR** #48512: (gtmanfred) pin pm and grunt packages for npm tests @ *2018-07-11 13:06:06 UTC*
  - 38df812257 Merge pull request #48512 from gtmanfred/npm
  - 9ba0f6b3a0 pin pm and grunt packages for npm tests
- **PR** #48513: (rallytime) [2017.7] Merge forward from 2017.7.7 to 2017.7 @ *2018-07-11 13:04:01 UTC*
  - 04ba31147f Merge pull request #48513 from rallytime/merge-2017.7
  - a466a4880f Merge branch '2017.7.7' into '2017.7'
- **PR** #48514: (gtmanfred) upgrade dependencies for pytest tests @ *2018-07-10 20:53:55 UTC*
  - f0352ea95a Merge pull request #48514 from gtmanfred/2017.7
  - 739bf92599 upgrade dependencies for pytest tests
- **PR** #48491: (grokrecursion) Group gid fixed as integer @ *2018-07-10 19:31:40 UTC*
  - 5372b60137 Merge pull request #48491 from grokrecursion/group-fix-v2
  - 1556b37221 fixed pylint errors

  - f6d9177dab fixed indentation for pep8
  - b9556bf923 v2 try of gid converted to integer
- **PR** #48375: (Ch3LL) Add user and group to makedirs cmd in file.copy @ *2018-07-10 12:35:10 UTC*
  - 16d3daab3c Merge pull request #48375 from Ch3LL/file_copy
  - d16a790775 Add mode to _makedirs call in file.copy
  - 2cbcb4fd26 Add user and group to makedirs cmd in file.copy
- **PR** #48490: (KaiSforza) Cifixes @ *2018-07-09 15:20:46 UTC*
  - d38951b1b3 Merge pull request #48490 from KaiSforza/cifixes
  - 9fe7199af5 Mark failed if anything is found
  - 6749a6bf19 Stop archiving the docs
- **PR** #48472: (gtmanfred) use tox to run pylint @ *2018-07-07 02:46:46 UTC*
  - 22cd4206c4 Merge pull request #48472 from gtmanfred/2017.7
  - c7a3a7d8bd update jenkins tests to use tox for lint
  - 44aaac1d33 use tox to run pylint
- **PR** #48476: (twangboy) Fix dependencies @ *2018-07-06 23:43:26 UTC*
  - 27ea8f35ea Merge pull request #48476 from twangboy/fix_dependencies
  - b0087d425c Add license info
  - 935f9b560c Fix dependencies
- **ISSUE** #48336: (JuanManuelVizcainoAbad) file.directory (refs: #48399)
- **PR** #48399: (garethgreenaway) [2017.7] fixes to module/file.py @ *2018-07-06 18:02:29 UTC*
  - acf42864aa Merge pull request #48399 from garethgreenaway/48336_2017_7_ensure_chmod_setuid_with_chown
  - 8efd33320f Normalize the mode before we compare it.
  - f894f0ecb8 Setting the mode with setuid or setgid bits in addition to setting the owner and group will force the setuid & setgid bits to reset. This change ensures that we set the mode after setting the owner & group.
- **PR** #48471: (gtmanfred) Add some configurations to tox @ *2018-07-06 17:45:55 UTC*
  - 6166ff6b78 Merge pull request #48471 from gtmanfred/2017.7
  - 392ab4e51f Add some configurations to tox
- **PR** #48433: (discogestalt) Fix issue with redismod.hmset method @ *2018-07-06 15:55:44 UTC*
  - 6234d9b15d Merge pull request #48433 from discogestalt/fix-redis.hmset
  - 65817ac74d Use clean_kwargs method instead
  - f7fa7f57c6 Found another issue with redismod.hmset
  - 5624865f86 Fix issue with redismod.hmset method
- **PR** #48428: (terminalmage) Fix outputter detection in jobs.lookup_jid runner @ *2018-07-06 14:37:37 UTC*
  - aacee0fe2c Merge pull request #48428 from terminalmage/fix-jobs.lookup_jid
  - 6509aa9089 Fix outputter detection in jobs.lookup_jid runner
- **PR** #48429: (rallytime) Back-port #46824 to 2017.7 @ *2018-07-03 21:45:36 UTC*

  - **PR** #46824: (beornf) Regression to ignore retcodes on crontab calls (refs: #48429)
  - a7e52f0de2 Merge pull request #48429 from rallytime/bp-46824
  - 8b4486248d Added ignore_retcode to mock unit tests
  - f8beab71dd Regression to ignore retcodes on crontab calls
- **PR** #48432: (dwoz) Prepend test needs file.touch method @ *2018-07-03 21:32:48 UTC*
  - 4576ef20bc Merge pull request #48432 from dwoz/file-prepend-again
  - 349a2b279e Prepend test needs file.touch method
- **ISSUE** #48230: (whytewolf) file.managed source_hash case-sensitive for test=true (refs: #48422)
- **ISSUE** #38914: (hgfischer) Uppercase checksums are not accepted by archive.extracted (refs: #40754, #48422)
- **PR** #48422: (rallytime) Allow file.managed to work with uppercase source_hash in test=true mode @ *2018-07-03 16:40:59 UTC*
  - **PR** #40754: (lordcirth) file.manage_file: uppercase checksums now work (refs: #48422)
  - 422441505d Merge pull request #48422 from rallytime/fix-48230
  - a6abf85621 Allow file.managed to work with uppercase source_hash in test=true mode
  - 4c691ac57a Add regression test for Issue #48230
- **PR** #48410: (dwoz) Fix service dead test on windows @ *2018-07-03 14:17:29 UTC*
  - 83e387c951 Merge pull request #48410 from dwoz/service_dead_test_fix
  - 5c0923448d Fix service dead test on windows
- **PR** #48413: (twangboy) Fix archive.extracted to handle UNC paths @ *2018-07-03 14:14:46 UTC*
  - d3ba345da5 Merge pull request #48413 from twangboy/fix_47811
  - c5aad04140 handle unc paths
- **PR** #48416: (dwoz) Fix service integration test (py3) @ *2018-07-03 14:08:45 UTC*
  - 827ef2aac4 Merge pull request #48416 from dwoz/service_integration_fix
  - c43f150dcd Fix service integration test (py3)
- **PR** #48385: (Ch3LL) Update release versions for the 2017.7 branch @ *2018-07-03 13:39:29 UTC*
  - 10fe7d2b9b Merge pull request #48385 from Ch3LL/1update_version_doc_2017.7
  - 3cf335b0b9 Update release versions for the 2017.7 branch
- **PR** #48405: (Ch3LL) [2017.7] Remove In Progress Warning on 2017.7.7 @ *2018-07-03 13:32:10 UTC*
  - f8a9a037e6 Merge pull request #48405 from Ch3LL/rm_inprog_2017.7
  - 03ed5167fa [2017.7] Remove In Progress Warning on 2017.7.7
- **PR** #48409: (dwoz) Fix file state prepend test @ *2018-07-02 23:21:08 UTC*
  - 8ac867c168 Merge pull request #48409 from dwoz/test_prepend_fix
  - 9c7085b70f Fix file state prepend test
- **PR** #48396: (dwoz) Fix file.symlink state test for windows @ *2018-07-02 18:10:32 UTC*
  - 10e93bff7f Merge pull request #48396 from dwoz/symlink_test_fix
  - d3456d31e6 Fix file.symlink state test for windows

- **PR** #48362: (twangboy) Fix stacktrace when registry entries are missing @ *2018-06-30 13:20:01 UTC*
  - 7e25f26837 Merge pull request #48362 from twangboy/fix_48276
  - 77629f640a Fix module unit tests
  - b5ca560b73 Fix lint error
  - 210e280ecf Use explicit parameter names
  - 776db98a5c Handle missing registry entries
- **PR** #48383: (rallytime) Back-port #48379 to 2017.7 @ *2018-06-29 19:48:38 UTC*
  - **PR** #48379: (KaiSforza) Clean up the workspaces at the end for every job (refs: #48383)
  - 80222b6b7c Merge pull request #48383 from rallytime/bp-48379
  - 9fc7815594 Clean up the workspaces at the end for every job
- **PR** #48382: (rallytime) Back-port #48346 to 2017.7 @ *2018-06-29 19:45:12 UTC*
  - **PR** #48346: (KaiSforza) Use the right ssh key to destroy kitchen (refs: #48382)
  - aa68aa774a Merge pull request #48382 from rallytime/bp-48346
  - 6a70ba222e Use the right ssh key to destroy kitchen
- **PR** #48381: (rallytime) Back-port #48330 to 2017.7 @ *2018-06-29 19:43:46 UTC*
  - **PR** #48330: (KaiSforza) Add warnings plugin to replace violations (refs: #48381)
  - 9519f640e7 Merge pull request #48381 from rallytime/bp-48330
  - 6857bedcc2 Add warnings plugin to replace violations
- **PR** #48363: (dubb-b) Changing debug to info for logging @ *2018-06-28 20:45:17 UTC*
  - 14db0aa35d Merge pull request #48363 from dubb-b/2017.7
  - c0b962e28d Changing debug to info for logging
- **PR** #48352: (Ch3LL) Add missing key to accept_dict function in wheel docs @ *2018-06-28 13:54:37 UTC*
  - 5b29cd326f Merge pull request #48352 from Ch3LL/wheel_doc
  - 89ada68165 Add missing key to accept_dict function in wheel docs
- **PR** #48329: (rallytime) Trigger review requests for team-ssh for roster files @ *2018-06-28 13:53:42 UTC*
  - 02cf19ee3b Merge pull request #48329 from rallytime/codeowners-roster-files
  - 5e56615e00 Trigger review requests for team-ssh for roster files
- **ISSUE** #48316: (ralex) Wrong init system used in virtual module "service" on Manjaro (refs: #48349)
- **PR** #48349: (ralex) Disable the "service" module on Manjaro since it is using systemd @ *2018-06-28 13:32:37 UTC*
  - 53cf1794be Merge pull request #48349 from ralex/fix-manjaro-service-behaviour
  - 6cafce547e Disable the *service* module on Manjaro since it is using systemd
- **PR** #48324: (Ch3LL) Update release versions for the 2017.7 branch @ *2018-06-27 13:44:22 UTC*
  - 89dfcf3a4e Merge pull request #48324 from Ch3LL/update_version_doc_2017.7
  - 47845ba810 Update release versions for the 2017.7 branch
- **PR** #48331: (rallytime) Back-port #48215 to 2017.7 @ *2018-06-27 13:12:35 UTC*

  - **PR** #48215: (KaiSforza) Notify during tests using different credentials (refs: #48331)
  - ece4c30aac Merge pull request #48331 from rallytime/bp-48215
  - 9d6ba3e247 Set jobs to pending when they come in
  - 727d2b4ffe Use proper creds with githubNotify
  - d355861c5c Test using different credentials
- **PR** #48332: (rallytime) Back-port #48321 to 2017.7 @ *2018-06-27 13:12:15 UTC*
  - **PR** #48321: (KaiSforza) Jenkins needs the / to get the directory (refs: #48332)
  - ab05e00d8c Merge pull request #48332 from rallytime/bp-48321
  - d72af6ab4e Jenkins needs the / to get the directory
- **ISSUE** #42659: (szin2012) rest_tornado is not able to return normal result (refs: #48193, #48295)
- **PR** #48295: (rallytime) Back-port #48193 to 2017.7 @ *2018-06-26 23:42:18 UTC*
  - **PR** #48193: (jacksontj) Properly wait on returns in saltnado (refs: #48295)
  - 21ed5b97ce Merge pull request #48295 from rallytime/bp-48193
  - 352fe33fd6 Separate set_result() to a new line apart from Future() call
  - c0180ff33d Properly configure syndic in test case
  - d0a98534a9 Properly wait on returns in saltnado
- **PR** #48117: (twangboy) service.disabled on Windows @ *2018-06-26 23:35:59 UTC*
  - 9aa4687ab9 Merge pull request #48117 from twangboy/fix_48026
  - dd37f8fbc2 Disable services that are set to manual
- **ISSUE** #48122: (pasmon) Salt minion 2017.7.6 disregards "bin_env" in pip state (refs: #48189)
- **PR** #48207: (rallytime) Back-port #48189 to 2017.7 @ *2018-06-25 19:26:56 UTC*
  - **PR** #48189: (gtmanfred) If pip binary is passed to bin_env, use that pip binary (refs: #48207)
  - caf630487c Merge pull request #48207 from rallytime/bp-48189
  - e9d09e0375 Use old is_windows utils path on 2017.7
  - b965d6c9b3 If pip binary is passed to bin_env, use that pip binary
- **ISSUE** #45383: (jodok) RabbitMQ commands fail due to wrong locale (refs: #47453)
- **PR** #48293: (rallytime) Back-port #47453 to 2017.7 @ *2018-06-25 19:06:42 UTC*
  - **PR** #47453: (dqminh) dont reset system locale when running rabbitmqctl commands (refs: #48293)
  - 06a927b2aa Merge pull request #48293 from rallytime/bp-47453
  - e96ab6778e dont reset system locale when running rabbitmqctl commands
- **PR** #48219: (zer0def) Fix: LXC legacy configuration key warnings falsely report errors during state change @ *2018-06-25 13:46:07 UTC*
  - 95ef006e00 Merge pull request #48219 from zer0def/lxc-bootstrap-fixes
  - c77058560b Merge branch '2017.7' into lxc-bootstrap-fixes
  - d63cf3f072 Fixes another case of legacy configuration key usage warning getting in the way of changing container's state.
- **ISSUE** #46507: (nkv16786) Failed to return clean data (refs: #48234)

- **PR** #48234: (dwoz) Fix py2 thin dir issues @ *2018-06-25 13:33:02 UTC*
  - 3327181507 Merge pull request #48234 from dwoz/thin_dir
  - 70c603451b Fix py2 thin dir issues
- **ISSUE** #47984: (jeduardo) x509 module/state writing wrong certificate serial number to CRL (refs: #48080)
- **PR** #48080: (lusche) Bugfix #47984 messed up cert serial @ *2018-06-22 18:00:14 UTC*
  - 83d7d286c4 Merge pull request #48080 from lusche/2017.7
  - 917dc985fc #47984 remove the line completly
  - ba12ee947b Merge branch '2017.7' of https://github.com/saltstack/salt into 2017.7
  - dfb13d5051 Bugfix #47984 messed up cert serial
- **PR** #48232: (gtmanfred) do not expand kwargs for cloud.action @ *2018-06-22 15:49:21 UTC*
  - bccb4dcd46 Merge pull request #48232 from gtmanfred/cloud
  - d108112e1a do not expand kwargs for cloud.action
- **PR** #48238: (mirceaulinic) SDB cache module: AttributeError: 'Cache' object has no attribute 'set' @ *2018-06-22 15:37:22 UTC*
  - 056f43f663 Merge pull request #48238 from mirceaulinic/fix-sdb-cache
  - 51e5fbfa1d SDB cache module: AttributeError: 'Cache' object has no attribute 'set'
- **ISSUE** #48025: (onmeac) Autoloading Utility Modules? (refs: #48037)
- **PR** #48037: (terminalmage) Add "sync_mods" argument to state.apply/state.sls @ *2018-06-21 19:57:03 UTC*
  - 731ec0a11a Merge pull request #48037 from terminalmage/fix-custom-types-sync-docs
  - 052ae83c4b Update versionchanged
  - 8b1bd0eda2 Update test to reflect changed argument name
  - 5e75936198 Change 2018.3.2 to 2018.3.3
  - c53ad603fc Rename sync -> sync_mods per review suggestion
  - e4d67c5fd8 Update docs to include references to new "sync" argument
  - cb8e6f9fb8 Remove redundant mocking
  - bc3ad795e9 Add test for sync argument to state.sls
  - f81ccd1fdd Add sync option to state.apply/state.sls
  - 8289b07e24 Fix documentation on when custom types are synced
- **PR** #48249: (rallytime) Update release notes for 2017.7.7 @ *2018-06-21 18:30:39 UTC*
  - 0bae927048 Merge pull request #48249 from rallytime/2017.7.7-release-notes-update
  - 36032c8ee7 Update release notes for 2017.7.7
- **PR** #48242: (asnell) Add sample list data via command line pillar @ *2018-06-21 18:27:41 UTC*
  - 8e06471817 Merge pull request #48242 from asnell/asnell-patch-1
  - f66bf60073 Add sample list data via command line pillar
- **PR** #48226: (terminalmage) 2 fixes for when Salt is installed using -OO @ *2018-06-20 20:12:34 UTC*
  - a172f9de84 Merge pull request #48226 from terminalmage/fix-alias-docstring

  - fb237272f5 Don't display "None" in SaltInvocationError when Salt installed using -OO
  - 90c90f5d5c Fix docstring construction in alias_function when Salt installed using -OO
- **PR** #48227: (Ch3LL) Skip new sha256 files on repo.saltstack.com/windows @ *2018-06-20 20:12:09 UTC*
  - c6a0207cae Merge pull request #48227 from Ch3LL/fix_win_tests
  - 236773e3e9 Skip new sha256 files on repo.saltstack.com/windows
- **ISSUE** #48128: (bowmanjd-lms) apk.py Python 3 compatibility (refs: #48131)
- **PR** #48131: (bowmanjd-lms) Fix py3-incompatible dict.keys()[x] call in apk.py @ *2018-06-19 18:19:02 UTC*
  - 6929423528 Merge pull request #48131 from bowmanjd-lms/fix-apk-python3
  - f5d2835299 Merge branch '2017.7' into fix-apk-python3
  - 022f9cba50 Fix py3-incompatible dict.keys()[x] call in apk.py
- **ISSUE** #47901: (frogunder) 2017.7.6 - Exiting command with CTRL-C gives Error/Traceback (refs: #48185)
- **PR** #48185: (DmitryKuzmenko) Don't call .exception() on future unless it's done. @ *2018-06-19 18:16:27 UTC*
  - 868c17377f Merge pull request #48185 from DSRCorporation/bugs/47901_future_done_exception
  - 5f63316311 Merge branch '2017.7' into bugs/47901_future_done_exception
  - 9f2dbf94cb Don't call .exception() on future unless it's done.
- **PR** #48192: (twangboy) Use the –disable-pip-version-check option (2017.7) @ *2018-06-19 18:13:12 UTC*
  - 80a3e37c2e Merge pull request #48192 from twangboy/fix_pip_version_2017.7
  - 59ae2cc5fc Use –disable-pip-version-check for build_env_#.ps1
- **PR** #48201: (zer0def) Fixes to LXC bootstrap when alternate lxcpath is provided. @ *2018-06-19 12:56:40 UTC*
  - 12e2b8882a Merge pull request #48201 from zer0def/lxc-bootstrap-fixes
  - 6f6d3d40d4 Take lxcpath into account when bootstrapping new containers. Version comparison fixes.
- **PR** #48190: (terminalmage) Fix mount.vfstab unit test @ *2018-06-18 21:47:54 UTC*
  - e079fce38d Merge pull request #48190 from terminalmage/fix-vfstab-test
  - 8d70d14362 Fix mount.vfstab unit test
- **PR** #48115: (KaiSforza) Add jenkinsfiles to define tests in 2017 @ *2018-06-18 21:27:40 UTC*
  - c8b9cec986 Merge pull request #48115 from KaiSforza/pr-lint-2017
  - 494727ab39 Add docs, kitchen tests
  - 317023bb20 Move .jenkins to .ci for future
  - dedc313cee Make it parallel so we don't fail right away
  - cd13426726 Add a .jenkins file to run pylint
- **PR** #48174: (terminalmage) Add docker.update_mine config option @ *2018-06-18 19:04:54 UTC*
  - fc1752b721 Merge pull request #48174 from terminalmage/docker-update-mine
  - 9af09e0127 Mock config.get due to changes in _refresh_mine_cache
  - c5802ad465 Fix the version number in versionadded
  - d403ae58d7 Add unit test for docker.update_mine
  - 44c275698a Actually it's more than just add/remove that updates the mine

  - 0cb6996b07 Add release notes mention of docker.update_mine config option
  - de05097b20 Add docs for new config option
  - 04c55a9178 Add note in mine.get_docker docstring about new config item
  - 4e456255c0 Allow mine update to be disabled using new config option
- **ISSUE** #47031: (lejambon) Cannot set hwclock on UTC on CentOS 7 (refs: #47049, #48061)
- **PR** #48061: (garethgreenaway) [2017.7] Porting #47049 to 2017.7. @ *2018-06-15 17:15:50 UTC*
  - **PR** #47049: (lejambon) Centos set utc hwclock (refs: #48061)
  - df2a156338 Merge pull request #48061 from garethgreenaway/port_47049_2017_7
  - 7c472fed51 Fixing failing test_set_hwclock_aix test.
  - ccb0acc958 Porting #47049 to 2017.7.
- **PR** #48143: (Ch3LL) Add timeout argument to run_salt for ShellCase @ *2018-06-15 17:04:57 UTC*
  - 5ec3cf2dd4 Merge pull request #48143 from Ch3LL/fix_long_job
  - 48b5d2e9d3 Add timeout argument to run_salt for ShellCase
- **PR** #48135: (rallytime) Update the 2017.7.7 and 2017.7.8 release notes @ *2018-06-14 21:22:06 UTC*
  - **PR** #48134: (rallytime) Add release notes file for 2017.7.7 (refs: #48135)
  - e1e566d1f8 Merge pull request #48135 from rallytime/release-notes-2017.7
  - 7a97f157b3 Add missing *v* for tag reference
  - 2f2b69ed37 Add "in progress" notation to 2017.7.7 release notes
  - 06a1151a63 Add release notes file for 2017.7.7
  - 885b2862ce Move 2017.7.7 release notes to 2017.7.8
- **PR** #48105: (rallytime) [2017.7] Merge forward from 2017.7.6 to 2017.7 @ *2018-06-14 17:00:09 UTC*
  - ac9dabbfaa Merge pull request #48105 from rallytime/merge-2017.7
  - cdb45874de Merge branch '2017.7.6' into '2017.7'
- **PR** #48101: (rallytime) [2017.7] Update 2017.7.7 reference to 2017.7.8 @ *2018-06-14 13:14:09 UTC*
  - 2da56a69d9 Merge pull request #48101 from rallytime/update-doc-refs-2017.7
  - 7ba6f5fb36 Update 2017.7.7 reference to 2017.7.8
- **PR** #48091: (terminalmage) Reverse monkeypatching after test_symlink_list finishes @ *2018-06-13 18:02:53 UTC*
  - 5a0e3d46e7 Merge pull request #48091 from terminalmage/fix-file_roots-monkeypatching
  - 6fe711ad76 Reverse monkeypatching after test_symlink_list finishes
- **PR** #48088: (rallytime) Update release versions for the 2017.7 branch @ *2018-06-13 14:03:03 UTC*
  - 053b019a8f Merge pull request #48088 from rallytime/update_version_doc_2017.7
  - 1b8d1c936b Remove "in progress" info for 2017.7.6 release notes
  - 9a0f4d190a Update release versions for the 2017.7 branch
- **ISSUE** #48038: (austinpapp) jobs are not dedup'ing minion side (refs: #48075)
- **PR** #48075: (garethgreenaway) [2017.7] Ensure that the shared list of jids is passed @ *2018-06-13 13:25:35 UTC*

  - d6d2c5ee18 Merge pull request #48075 from garethgreenaway/48038_jobs_are_not_deduping_minion_side
  - c537b3275b Ensure that the shared list of jids is passed when creating the Minion. Fixes an issue when minions are pointed at multiple syndics.
- **PR** #48071: (terminalmage) Fix inaccurate gitfs_saltenv example in GitFS Walkthrough @ *2018-06-12 20:34:13 UTC*
  - aa33cc0b0c Merge pull request #48071 from terminalmage/gitfs-docs
  - 24545204b3 Fix inaccurate gitfs_saltenv example in GitFS Walkthrough
- **ISSUE** #48039: (twellspring) github.repo_present returns stacktrace, got multiple values for keyword argument profile (refs: #48040)
- **PR** #48053: (rallytime) Back-port #48040 to 2017.7 @ *2018-06-12 14:46:12 UTC*
  - **PR** #48040: (twellspring) states/github.py fix for incorrect positional argument (refs: #48053)
  - fb4ceacb88 Merge pull request #48053 from rallytime/bp-48040
  - 265b22b194 states/github.py fix for incorrect positional argument
- **PR** #48024: (gtmanfred) add pytest coverage and xml junits @ *2018-06-12 14:03:26 UTC*
  - f37dcaac6d Merge pull request #48024 from gtmanfred/2017.7
  - 56b074ab27 allow specifying a different state to run on converge
  - cc9c4b4d5a add pytest coverage and xml junits
- **PR** #48044: (twangboy) Add fail code for WinHTTP send/receive error @ *2018-06-11 13:49:29 UTC*
  - ffee3576ec Merge pull request #48044 from twangboy/wua_add_fail_code
  - d41d0c25eb Add faile code for WinHTTP send/receive error
- **PR** #48009: (dwoz) minionswarm runs on windows @ *2018-06-08 13:52:36 UTC*
  - 871a910fcb Merge pull request #48009 from dwoz/winswarmfix
  - 5027c7bb84 minionswarm runs on windows
- **PR** #47968: (gtmanfred) Clean up tox + pytest @ *2018-06-06 21:23:52 UTC*
  - 0dcaead36d Merge pull request #47968 from gtmanfred/2017.7
  - ae1b0d28bb pass LANG and HOME into tox commands
  - 8d15b93d00 remove tox virtualenvs from doc tests
  - e1872e2036 simplify tox.ini
  - 81bd01f5af add tests.txt for running tests with tox requirements
  - 556a2067fc fix masterapi test to use testing directory for configs
  - 5a41f484ef add EXPENSIVE_TESTS
  - 055cd5a6ba set DESTRUCTIVE_TESTS environment variable for pytest
- **PR** #47978: (twangboy) Add try/finally, fix typo in 7zip def file @ *2018-06-06 19:06:04 UTC*
  - 42dd6b83eb Merge pull request #47978 from twangboy/fix_test_pkg
  - 8bb36b3a4b Add try/finally, fix typo in 7zip def file
- **PR** #47959: (twangboy) Fix failing test when service doesn't exist @ *2018-06-06 19:05:43 UTC*
  - a67b21ef3d Merge pull request #47959 from twangboy/fix_test_service

  - 0cd47aa81e Remove unused import
  - d4b42540e3 Fix failing test when service doesn't exist
- **ISSUE** #30367: (paulfanelli) salt-run orchestration via state.sls always successful; it is ignoring state return value (refs: #47843)
- **PR** #47983: (garethgreenaway) [2017.7] Backporting #47843 @ *2018-06-06 19:03:15 UTC*
  - **PR** #47843: (garethgreenaway) [2018.3] orchestration results False when function is False (refs: #47983)
  - b81d482067 Merge pull request #47983 from garethgreenaway/backport_47843_2017_7
  - 7b7fb1f5d5 Adding missing test sls file.
  - 54e51bc627 Backporting #47843 to 2017.7, updating test for 2017.7.
- **PR** #47533: (twangboy) Fix issues with functions that use makedirs on Windows @ *2018-06-06 17:20:51 UTC*
  - d56ddad22c Merge pull request #47533 from twangboy/fix_47178
  - 24717cbc21 Change version added to 2017.7.7
  - 0592f1bac2 Fix issues with functions that user makedirs on Windows
- **PR** #47827: (twangboy) Fix issue when archive is on mapped drive @ *2018-06-06 17:18:00 UTC*
  - 5c56b8c755 Merge pull request #47827 from twangboy/fix_47791
  - fbbd91f09e Add more descriptive debug message
  - 365f81651b Fix deprecated exception handling
  - 8dd6710b93 Use local instead of network
  - 980d99d74b Fix issue when archive is on mapped drive
- **ISSUE** #47888: (mbunkus) postfix.set_main overwrites wrong keys (refs: #47966)
- **PR** #47966: (mbunkus) Fix postfix.set_main's assumption of prefix-free key names @ *2018-06-06 15:43:24 UTC*
  - 998a1c13d5 Merge pull request #47966 from mbunkus/fix-postfix-prefix-free-key-names-assumption
  - 0ae402d1b5 Fix postfix.set_main's assumption of prefix-free key names
- **PR** #47824: (twangboy) Fix issues with setup.py in Windows @ *2018-06-05 17:39:53 UTC*
  - df50ce7b1b Merge pull request #47824 from twangboy/fix_setup.py
  - 4538b3abb3 Remove 'len-as-condition' disablement
  - d4efcc3c8b Skip lint errors
  - 1b3977f8d4 Remove m2crypto download for Windows
- **PR** #47922: (damon-atkins) win_pkg: pkg.refresh_db verbose=True PY3 compatibility fix @ *2018-06-05 13:31:49 UTC*
  - 3e91a31134 Merge pull request #47922 from damon-atkins/2017.7_win_pkg_keys_fix
  - 7129203b1b Merge branch '2017.7' into 2017.7_win_pkg_keys_fix
- **PR** #47933: (terminalmage) Clarify pillar.get docs @ *2018-06-04 14:07:58 UTC*
  - 8008fca2f6 Merge pull request #47933 from terminalmage/add-pillar.get-docs
  - 4eb022b675 Clarify pillar.get docs
- **PR** #47944: (terminalmage) Replace use of deprecated argument name in git.detached docstring @ *2018-06-04 13:53:34 UTC*

  - ab565bfaa3 Merge pull request #47944 from terminalmage/fix-git.detached-docs
  - 1d082b4389 Replace use of deprecated argument name in git.detached docstring
  - 6253793cb6 Merge branch '2017.7' into 2017.7_win_pkg_keys_fix
- **PR** #47860: (terminalmage) Clean up doc build warnings, add best practices for writing states @ *2018-06-01 14:42:55 UTC*
  - a2b588299b Merge pull request #47860 from terminalmage/state-writing-docs
  - 8cb33d3dec Quiet the linter
  - af51e16f23 Use more elegant RST syntax
  - 91cd57d1e0 Update mocking to reflect changes in service module
  - b9ace5a859 Fix crappy mocking
  - bc9e3acef3 Lint fixes
  - d37f7e4f04 Add 2017.7.6 release notes
  - 9dddeeefab Improve documentation on syncing states
  - f465fa3ca7 Add best practices docs for writing states
  - 93ee5ee2b0 Fix all Sphinx warnings
  - 64b9b4d0b8 Clarify that name would override the id declaration
- **PR** #47877: (dwoz) Revert job chunk wait time change @ *2018-06-01 14:11:46 UTC*
  - d2f3377b5b Merge pull request #47877 from dwoz/sleep_audit
  - ebc7cde9cb Revert job chunk wait time change
    - 4ce0fe6546 win_pkg under py3 keys returns a view instead of a list, wrap keys in list()
- **PR** #47917: (dwoz) Fix windows tests suite breakage @ *2018-06-01 11:42:15 UTC*
  - **PR** #47500: (dwoz) Get the current username on windows (refs: #47917)
  - 735e92093e Merge pull request #47917 from dwoz/winsuite
  - 118601ebd6 Fix windows tests suite breakage
- **PR** #47822: (Ch3LL) Add user state integration tests to windows @ *2018-05-30 21:13:07 UTC*
  - 63efb76d51 Merge pull request #47822 from Ch3LL/win_user_test
  - 3a691b405f add user_home path for both windows and linux
  - 788abf771e Add user state integration tests to windows
- **PR** #47876: (doesitblend) Add file read windows @ *2018-05-30 14:03:55 UTC*
  - b9da4f1221 Merge pull request #47876 from doesitblend/add-file-read-windows
  - 3f7e7ec327 Add file.read function to Windows module
- **PR** #47882: (frogunder) add whoami test @ *2018-05-30 13:47:13 UTC*
  - cbe2ecfae8 Merge pull request #47882 from frogunder/45013
  - 38d114a2d2 add whoami test
- **PR** #47853: (dwoz) Fix for py3 ec2 cloud tests @ *2018-05-28 23:08:09 UTC*
  - 4b28e510f1 Merge pull request #47853 from dwoz/py3-cloud-test

  - c2f8aef7c5 Fix for py3 ec2 cloud tests
- **ISSUE** #47833: (blefeuvr) reactor.add | list | delete -> python exception (refs: #47847)
- **PR** #47847: (garethgreenaway) [2017.7] reactor runner documentation @ *2018-05-25 23:23:31 UTC*
  - 215efa6b73 Merge pull request #47847 from garethgreenaway/47833_reactor_runner_documentation
  - 03676712de Adding some addition documentation to the reactor runner indicating that the reactor system must be active prior to using it.
- **PR** #47846: (Ch3LL) Fix flaky refresh pillar integration test @ *2018-05-25 20:50:48 UTC*
  - 0bf651b470 Merge pull request #47846 from Ch3LL/p_refresh_test
  - 019edad8e4 Fix flaky refresh pillar integration test
- **PR** #47552: (twangboy) Show GPO settings, raise error if trying to set gpo managed settings @ *2018-05-25 20:41:22 UTC*
  - 799fce979d Merge pull request #47552 from twangboy/fix_46981
  - cffc9b7806 Merge branch '2017.7' into fix_46981
  - 81308a4a44 Add release notes for 2017.7.7
  - 008af0ac6b Fix unit tests
  - 87097eefb6 Add comments about how get is returning data
  - f1f1bfc5c1 Show GPO settings, raise error if trying to set gpo managed settings
- **ISSUE** #47690: (lomeroe) productname grain missing on Solaris (refs: #47691)
- **PR** #47691: (lomeroe) Update regexes for Solaris 'product' grain and copy to 'productname' grain @ *2018-05-25 20:38:18 UTC*
  - ffc2ebc373 Merge pull request #47691 from lomeroe/solaris_product_grains
  - 6889befc6b Merge branch '2017.7' into solaris_product_grains
  - 49053bc106 lint fix
  - 6f185c9179 another lint fix
  - dbffba6876 fix tons of errors in my tests
  - 8c9355d34c Lint fix
  - b29ec75da7 Update regexes in core grains for detecting the 'product' grain on Solaris Sparc systems. Additionally, copy the 'product' grain to 'productname' to be consistent with other OSes.
- **PR** #47778: (Ch3LL) Add test_pkg integration state tests to windows @ *2018-05-25 20:36:47 UTC*
  - 96b7c0cc76 Merge pull request #47778 from Ch3LL/win_state_pkg
  - e1786de6b0 Merge branch '2017.7' into win_state_pkg
  - f8c467d3e6 Fix text editor error
  - 872e162137 Add test_pkg integration state tests to windows
- **PR** #47823: (Ch3LL) Add win_dns module integration tests for windows @ *2018-05-25 20:34:03 UTC*
  - 6f934c2259 Merge pull request #47823 from Ch3LL/win_dns_test
  - 7848114d6a Add win_dns module integration tests for windows
- **PR** #47838: (rallytime) Back-port #47810 to 2017.7 @ *2018-05-25 20:33:44 UTC*

  - **PR** #47810: (rmcintosh) Fix Linode plan selection (refs: #47838)
  - e63e04602d Merge pull request #47838 from rallytime/bp-47810
  - bd2b62fa66 better debug message
  - 83565c55df Address PR feedback
  - 50bce3a2f3 make decode_linode_plan_label a private function
  - 2ba4fc4cea fix raising when a 'GB' format invalid plan is supplied
  - 3afb50d5a2 slight cleanup
  - 319fbd3406 match quotation mark types properly
  - 95e0202223 more consistent use of parens in logged warning
  - f52926ca87 log a warning when the user supplied a label we could decode but was not in the proper format
  - 121303d827 reduce complexity of get_plan_id by moving decoding of the user-supplied label to its own function
  - cf534c7314 Fix Linode plan selection
- **PR** #47805: (twangboy) Add sign.bat script for signing packages @ *2018-05-25 13:53:58 UTC*
  - 9ef6e60617 Merge pull request #47805 from twangboy/add_sign.bat
  - 66d8b0331a Add sign.bat script for signing packages
- **PR** #47500: (dwoz) Get the current username on windows (refs: #47917) @ *2018-05-24 19:14:07 UTC*
  - 93420950d3 Merge pull request #47500 from dwoz/winuser
  - 0ca5224cbf Merge branch '2017.7' into winuser
  - e5948902af Use salt utils method for this_user
  - a056a293f1 Centeralize test username lookup
  - 6a6ab69722 Get the current username on windows
- **ISSUE** #35960: (basepi) Beacons cannot be reloaded/refreshed without minion restart (refs: #40892)
- **PR** #47802: (rallytime) Back-port #40892 to 2017.7 @ *2018-05-24 16:34:35 UTC*
  - **PR** #40892: (jagguli) add minion function to reload beacon #35960 (refs: #47802)
  - 552526f309 Merge pull request #47802 from rallytime/bp-40892
  - bb357da084 add minion function to reload beacon #35960
- **PR** #47816: (Ch3LL) Add windows to service disable ERROR check in tests @ *2018-05-24 15:31:36 UTC*
  - c9cccf53b9 Merge pull request #47816 from Ch3LL/disable_windows
  - 2509d36888 Add windows to service disable ERROR check in tests
- **PR** #47773: (frogunder) add win_servermanager.list_available test @ *2018-05-23 19:44:14 UTC*
  - 2cb6634c6b Merge pull request #47773 from frogunder/45011
  - 7c9b0bda33 add win_servermanager.list_available test
- **PR** #47807: (dwoz) Try an even bigger timeout @ *2018-05-23 19:38:21 UTC*
  - e65dece202 Merge pull request #47807 from dwoz/winrmtests
  - 1f1cc1357a Increase instance size for cloud tests

  - be8dcd21f1 Try an even bigger timeout
- **PR** #47765: (meaksh) Prevent zypper from parsing repo configuration from not .repo files @ *2018-05-23 14:45:04 UTC*
  - 46618d2ce7 Merge pull request #47765 from meaksh/2017.7-fix-zypper-error-reading-repo
  - 4475ba19b8 Prevent zypper from parsing repo configuration from not .repo files
- **PR** #47781: (rallytime) Update cloud test profile and docs to use new Linode size lables @ *2018-05-23 13:09:13 UTC*
  - 0e87559ee3 Merge pull request #47781 from rallytime/update-linode-sizes
  - a90c1b760e Update cloud test profile and docs to use new Linode size lables
- **PR** #47748: (rallytime) [2017.7] Merge forward from 2017.7.6 to 2017.7 @ *2018-05-22 20:53:02 UTC*
  - 3ddc56cb9b Merge pull request #47748 from rallytime/merge-2017.7
  - b75f5ae38e Merge branch '2017.7.6' into '2017.7'
  - 17596f3025 Merge branch '2017.7.6' into '2017.7'
  - e458f4a031 Merge branch '2017.7.6' into '2017.7'
- **PR** #47779: (dwoz) Increase ec2 cloud timeouts to 1000 @ *2018-05-22 20:35:45 UTC*
  - 0ddc864f90 Merge pull request #47779 from dwoz/winrmtests
  - d5eafe9d53 Use common timeout variable
  - 7e948eb540 Increase ec2 cloud timeouts to 1000
- **PR** #47430: (baniobloom) Add catch for VPCAssociationNotFound @ *2018-05-22 14:54:01 UTC*
  - 17c4c8443c Merge pull request #47430 from baniobloom/vpc_association_not_found
  - 0f1d007f91 added catch for VPCAssociationNotFound
- **PR** #47525: (baniobloom) Trying to add a duplicate sec group rule true @ *2018-05-22 14:51:09 UTC*
  - 9c5dde1449 Merge pull request #47525 from baniobloom/duplicate_sec_group_rule
  - dcc35df489 if we are trying to add the same rule then we are already in the desired state, return true
- **PR** #47772: (gtmanfred) lock down dependencies for kitchen-salt @ *2018-05-22 14:21:49 UTC*
  - 536ce2fa42 Merge pull request #47772 from gtmanfred/2017.7
  - 67756a50fd lock down dependencies for kitchen-salt
- **ISSUE** #47717: (server-monitoring) Invalid output of pkg.list_upgrades on Fedora 27 Server (refs: #47749)
- **PR** #47749: (terminalmage) Fix "dnf list upgrades" parsing @ *2018-05-22 12:51:29 UTC*
  - 12b330f049 Merge pull request #47749 from terminalmage/issue47717
  - cd484bef69 Fix "dnf list upgrades" parsing
- **PR** #47755: (dwoz) Be explicit about winrm setting @ *2018-05-21 22:39:24 UTC*
  - 5cd18b9e70 Merge pull request #47755 from dwoz/winrmtests
  - 57dd89e6c3 Default to ec2 classic compatible images
  - f89668920a Be explicit about winrm setting
- **PR** #47668: (Ch3LL) Add pkg.latest_version windows test @ *2018-05-21 20:26:45 UTC*
  - f45a96ba1d Merge pull request #47668 from Ch3LL/win_pkg_test

  - a8981024de Add pkg.latest_version windows test
- **PR** #47750: (dwoz) Allow ssh_interface to default to public_ips @ *2018-05-21 17:14:11 UTC*
  - 7b12444dfa Merge pull request #47750 from dwoz/cloudtest
  - 63b722ba21 Allow ssh_interface to default to public_ips
- **PR** #47737: (dwoz) Cloud test fixup @ *2018-05-21 14:19:49 UTC*
  - 27fbb42210 Merge pull request #47737 from dwoz/win_timeout
  - e88833a07d Cloud test fixup
- **PR** #47729: (UtahDave) fix cli example to match function name @ *2018-05-19 11:31:21 UTC*
  - 1a93f060fb Merge pull request #47729 from UtahDave/fix_get_info_doc
  - 443a2d72a2 fix cli example to match function name
- **ISSUE** #47182: (MartinEmrich) state.highstate error with require_in: sls: (refs: #47682)
- **PR** #47682: (terminalmage) Fix traceback when excludes are present in an included SLS file @ *2018-05-18 16:47:47 UTC*
  - 00a13761c7 Merge pull request #47682 from terminalmage/issue47182
  - d0243e8f23 Suppress spurious lint failure
  - 3b449f11fc Add regression test for excludes issue
  - 28a7d2b81c Skip __exclude__ in find_sls_ids
- **ISSUE** #47696: (darkpixel) 2017.7.5 zpool.present does not check for properties=None before attempting to enumerate properties (refs: #47708)
- **PR** #47708: (darkpixel) Wrap properties loop with a check for None. Closes GH-47696 @ *2018-05-18 15:25:44 UTC*
  - 518f7bcc62 Merge pull request #47708 from darkpixel/47696-do-not-enumerate-none
  - 54e9bf9ec9 Merge branch '2017.7' into 47696-do-not-enumerate-none
- **PR** #47681: (rallytime) [2017.7] Merge forward from 2017.7.6 to 2017.7 @ *2018-05-17 21:30:02 UTC*
  - cad062eaa7 Merge pull request #47681 from rallytime/merge-2017.7
  - 6c06cb3ae3 Merge branch '2017.7.6' into '2017.7'
- **PR** #47683: (Ch3LL) Remove unnecessary setUp in states.test_user test for mac @ *2018-05-17 21:28:26 UTC*
  - fd2d45d9c2 Merge pull request #47683 from Ch3LL/mac_user_test
  - 5a1b25b9c7 Remove unnecessary setUp in states.test_user test for mac
- **PR** #47692: (dwoz) Default windows to m1.small for ec2-classic @ *2018-05-17 14:44:10 UTC*
  - f323799c42 Merge pull request #47692 from dwoz/win_instance_type
  - 20b6070d54 Default windows to m1.small for ec2-classic
- **ISSUE** #45125: (wes-novack) salt-cloud no longer picking up master setting from provider file (refs: #47665)
- **PR** #47665: (gtmanfred) use dictupdate to update the providers dictionary and merge lists @ *2018-05-16 15:59:17 UTC*
  - 990ece5cd5 Merge pull request #47665 from gtmanfred/cloud
  - fd8a02decb use dictupdate to update the providers dictionary and merge lists

- **PR** #47151: (zer0def) Allow interaction with default section in ConfigParser serializer @ *2018-05-16 13:53:44 UTC*
  - a2ed8cbb7f Merge pull request #47151 from zer0def/configparser-defaultsect
  - a8f2ad977e Merge branch '2017.7' into configparser-defaultsect
  - cca2a806c9 Made interaction with [DEFAULT] section in ConfigParser as sane as upstream permits.
- **ISSUE** #40846: (ghost) file.managed fails for long URLs (refs: #45808)
- **PR** #47673: (terminalmage) Backport #45808 to 2017.7 @ *2018-05-16 13:52:06 UTC*
  - **PR** #45808: (b1naryth1ef) [bugfix] Fix for oversized filenames in fileclient (refs: #47673)
  - e6fb8342ef Merge pull request #47673 from terminalmage/bp-45808
  - 454291ad62 Fix errors when attempting to cache files with long names or URLs
- **PR** #47670: (gtmanfred) add a pytest.ini and update a dependency for kitchen @ *2018-05-16 01:56:17 UTC*
  - b0446aab07 Merge pull request #47670 from gtmanfred/2017.7
  - dfd5a8715f add a pytest.ini and update a dependency for kitchen
- **PR** #47626: (gtmanfred) Implement Pipfile and remove extra requirements.txt files @ *2018-05-16 00:03:12 UTC*
  - 0d06da6944 Merge pull request #47626 from gtmanfred/2017.7
  - 418364a533 make dev_python27.txt have the same message as dev_python34.txt
  - d53b44d10f remove lock file
  - 493ed7f93d use the toml dict format
  - e0f7cc1447 add proxy tests decorator to pytest conftest
  - 0a621dd0ec add Pipfile for managing dependencies in salt
  - f78b81db94 simplify dev and base.txt to single files
- **PR** #47578: (Ch3LL) Ensure mac_service.disabled is correctly querying services @ *2018-05-15 18:26:37 UTC*
  - a07a8906a0 Merge pull request #47578 from Ch3LL/mac_service_disabled
  - 6032a01f55 change disable check for upstart service on ubuntu14
  - 33b4cfbc5b Change service assertion check for systemd platforms
  - 14896f9743 change codeauthor and class name
  - 502c5bdff5 Ensure mac_service.disabled is correctly querying services
- **PR** #47639: (rallytime) [2017.7] Merge forward from 2017.7.6 to 2017.7 @ *2018-05-14 17:58:25 UTC*
  - f63d801858 Merge pull request #47639 from rallytime/merge-2017.7
  - 1503f1020f Merge branch '2017.7.6' into '2017.7'
- **PR** #47643: (dwoz) Remove unwanted file @ *2018-05-14 17:53:57 UTC*
  - db0eef3e8b Merge pull request #47643 from dwoz/runas_detect
  - 8c5e54b8b3 Fix typo
  - 69bc88f1c6 Remove unwanted file
- **PR** #47601: (dwoz) Skip tests when we can not use runas @ *2018-05-14 15:26:36 UTC*
  - 6dd3ab9f06 Merge pull request #47601 from dwoz/runas_detect

  - 3235ac08cb use ignore-undefined-variable
  - 58911d510a Ignore pylint WindowsError
  - 17987d3c5a Better doc string
  - a48ac26573 Skip tests when we can not use runas
- **PR** #47611: (Ch3LL) Update debian install docs to reflect new latest debian releases @ *2018-05-11 16:32:53 UTC*
  - f61ccae627 Merge pull request #47611 from Ch3LL/deb_doc
  - 6d2ddd050f One more grammar fixup
  - 2d5ff01261 Grammar fix
  - 138847c9c4 Update debian install docs to reflect new latest debian releases
- **PR** #47598: (rallytime) Back-port #47568 to 2017.7 @ *2018-05-10 22:06:20 UTC*
  - **PR** #47568: (terminalmage) salt.serializers.yaml/yamlex: remove invalid multi_constructor (refs: #47598)
  - 9e2fe284e4 Merge pull request #47598 from rallytime/bp-47568-2017.7
  - 09458c65cf Add exception logging on serialize/deserialize exceptions
  - c62c855f9c salt.serializers.yaml/yamlex: remove invalid multi_constructor
- **PR** #47548: (Ch3LL) Catch Sysloghandler errors when log file does not exist @ *2018-05-10 20:23:25 UTC*
  - 8a7913ccc3 Merge pull request #47548 from Ch3LL/syslog
  - 31f13a4197 sysloghander: check for 3.5.4 python version
  - a020352a03 Catch Sysloghandler errors when log file does not exist
- **ISSUE** #47491: (doc987) mysql_query.run_file Doesn't Work when query_file = salt://… (refs: #47508)
- **PR** #47508: (gtmanfred) allow pulling the mysql_query.run_file to pull from the fileserver @ *2018-05-10 18:12:28 UTC*
  - 82fb6ba366 Merge pull request #47508 from gtmanfred/2017.7
  - 1a87e7455f allow pulling the mysql_query.run_file to pull from the fileserver
- **ISSUE** #47203: (aesposito91) NAPALM grains not updating (refs: #47513)
- **ISSUE** #46546: (ppalmieri) salt-minion installed on vEOS does not give vendor grain (refs: #47513)
- **PR** #47562: (rallytime) Back-port #47513 to 2017.7 @ *2018-05-10 15:19:04 UTC*
  - **PR** #47513: (frankiexyz) Fix #46546 eos napalm's grains issue (refs: #47562)
  - be29b58a09 Merge pull request #47562 from rallytime/bp-47513
  - 357bc084b3 fix #46546
- **PR** #47471: (meaksh) Fix inconsistency with "jid" on minion scheduled jobs and the returners output @ *2018-05-09 18:20:00 UTC*
  - d3121fcfa4 Merge pull request #47471 from meaksh/2017.7-fix-inconsistent-scheduled-jid-with-returners
  - f079939500 Do not override jid on returners, only sending back to master
- **PR** #47313: (Circuitsoft) Add Poky/Yocto support @ *2018-05-09 17:31:59 UTC*
  - 662f6086db Merge pull request #47313 from Circuitsoft/2017.7

  - 237560b745 Merge branch '2017.7' into 2017.7
- **PR** #47540: (terminalmage) Fix corner case where runas user's HOME env value is incorrect @ *2018-05-08 20:31:40 UTC*
  - 291d3d3bd8 Merge pull request #47540 from terminalmage/cmdmod-home-fix
  - e4b277f82e Fix corner case where runas user's HOME env value is incorrect
- **ISSUE** #24148: (abng88) Update gitfs docs to include tutorial on how to configure gitfs for salt-call (masterless) (refs: #47547)
- **PR** #47547: (terminalmage) Add masterless mode docs to gitfs tutorial @ *2018-05-08 20:24:29 UTC*
  - 49af577b3e Merge pull request #47547 from terminalmage/gitfs-tutorial
  - 806ffb298a Add masterless mode docs to gitfs tutorial
- **PR** #47538: (terminalmage) Fix/clarify some of the pip module documentation @ *2018-05-08 15:24:18 UTC*
  - 476ba053d1 Merge pull request #47538 from terminalmage/pip-docs
  - 98e8ffeba4 Fix/clarify some of the pip module documentation
- **PR** #47524: (dwoz) Gracefully handle blank lines in whitelist.txt @ *2018-05-08 15:04:23 UTC*
  - 7cc8d9d30a Merge pull request #47524 from dwoz/whitelistlines
  - 2605ff8712 Gracefully handle blank lines in whitelist.txt
- **PR** #47527: (dwoz) Test file name @ *2018-05-07 22:02:04 UTC*
  - f1a5b13072 Merge pull request #47527 from dwoz/test_file_name
  - 4aea7ca571 Update doc string
  - b3289e5a1c Fix test file name
- **PR** #47408: (Ch3LL) Skip status.diskusage integration tests on macsox @ *2018-05-07 19:06:57 UTC*
  - e9c1723175 Merge pull request #47408 from Ch3LL/skip_diskusage
  - 068da8ad7d Merge branch '2017.7' into skip_diskusage
  - 1a7ffb4e0d return error if diskusage not available
  - 10bd63a976 Skip status.diskusage integration tests on macsox
- **PR** #47480: (baniobloom) added handling for the aws error ConflictingDomainExists @ *2018-05-07 19:05:28 UTC*
  - 2f50ff7bc8 Merge pull request #47480 from baniobloom/ConflictingDomainExists
  - d22ed7dffa added handling for the aws error ConflictingDomainExists
- **PR** #47444: (terminalmage) Skip trying to render a template for a nonexistant SLS file @ *2018-05-07 13:48:24 UTC*
  - 7f53be6e92 Merge pull request #47444 from terminalmage/render_state-spurious-error
  - a1e9fe00fd Skip trying to render a template for a nonexistant SLS file
- **PR** #47478: (terminalmage) Rename pip state test modules to match naming convention @ *2018-05-07 13:13:13 UTC*
  - 50b9c4d79d Merge pull request #47478 from terminalmage/rename-pip-state-test
  - 9f7a9ebebd Rename pip state test modules to match naming convention
- **ISSUE** #47443: (skylerberg) Input validation does not raise SaltInvocationError in win_dsc.py (refs: #47505)

- **PR** #47505: (dwoz) Raise proper invocation errors @ *2018-05-07 13:05:22 UTC*
    - e78ac0f9b7 Merge pull request #47505 from dwoz/win_dsc_fixes
    - d1fcb40d1c Raise proper invocation errors
- **PR** #47499: (dwoz) Move kill process tree and re-use it @ *2018-05-06 20:54:22 UTC*
    - 0d4d5047d8 Merge pull request #47499 from dwoz/win_run_timeout_again
    - c6697b9f16 Move kill process tree and re-use it
- **PR** #47493: (dwoz) Add support for windows timeout to run_salt @ *2018-05-06 05:34:31 UTC*
    - 6a4d0380b1 Merge pull request #47493 from dwoz/win_run_timeout
    - fec1233dc4 Add support for windows timeout to run_salt
    - cb674fb1cb Enable opkg as pkgrepo handler on Poky
    - 7a58fd157e Enable opkg on non-NILinuxRT systems
    - 139360c55f Add Poky to OS Information Grains
    - d74057224d Un-normalize os_family in pkgrepo state
    - 95d4488f01 Merge branch '2017.7.6' into 47696-do-not-enumerate-none
    - 041e4c6ddb Wrap properties loop with a check for None. Closes GH-47696

28.2.29 In Progress: Salt 2017.7.9 Release Notes

Version 2017.7.9 is an **unreleased** bugfix release for *2017.7.0*. This release is still in progress and has not been released yet.

Salt Cloud Features

GCE Driver

The GCE salt cloud driver can now be used with GCE instance credentials by setting the configuration paramaters `service_account_private_key` and `service_account_private_email` to an empty string.

28.2.30 Salt 2016.11.0 Release Notes - Codename Carbon

New Features

Docker Introspection and Configuration

Major additions have been made to the Docker support in 2016.11.0. The new addition allows Salt to be executed within a Docker container without a minion running or installed in the container. This allows states to be run inside a container, but also all of Salt's remote execution commands to be run inside docker containers as well. This makes container introspection simple and powerful. See the tutorial on using this new feature here:

See *Salt in Docker Containers*.

Advanced Ceph Control

Our friends over at SUSE have delivered a powerful new tool to make the deployment of Ceph storage systems using Salt very easy. These new Ceph tools allow for a storage system to be easily defined using the new *ceph.quorum* state.

Thorium Additions and Improvements

The Thorium advanced reactor has undergone extensive testing and updates. These updates include many more Thorium states, a system for automating key management, the ability to use Thorium to easily replace old reactors and a great deal of stability and bug fixes.

State Rollback Using Snapper

Rollback has been one of the most prevalent requests for Salt. We have researched it extensively and concluded that the only way to accomplish truly reliable rollback would be to execute it at the filesystem layer. To accomplish this we have introduced Snapper integration into Salt States.

Snapper is a tool which allows for simple and reliable snapshots of the filesystem to be made. With the new *snapper_states* option set to *True* in the minion config a snapshot will be made before and after every Salt State run.

These snapshots can be viewed, managed and rolled back to via the *snapper* execution module.

Preserve File Perms in File States

This feature has been requested for years, the ability to set a flag and use the same file permissions for files deployed to a minion as the permissions set to the file on the master. Just set the *keep_mode* option on any file management state to *True*.

Ponies!

We all agreed that cowsay was just not good enough, install the *ponysay* command and the new *pony* outputter will work. Fun for the whole family!

Additional Features

- Minions can run in stand-alone mode to use beacons and engines without having to connect to a master. (Thanks @adelcast!)
- Added a `salt` runner to allow running salt modules via salt-run.

  ```
  salt-run salt.cmd test.ping
  # call functions with arguments and keyword arguments
  salt-run salt.cmd test.arg 1 2 3 a=1
  ```

- Added SSL support to Cassandra CQL returner. SSL can be enabled by setting `ssl_options` for the returner. Also added support for specifying `protocol_version` when establishing cluster connection.
- The `mode` parameter in the *`file.managed`* state, and the `file_mode` parameter in the *`file.recurse`* state, can both now be set to `keep` and the minion will keep the mode of the file from the Salt fileserver. This

works only with files coming from sources prefixed with `salt://`, or files local to the minion (i.e. those which are absolute paths, or are prefixed with `file://`). For example:

```
/etc/myapp/myapp.conf:
  file.managed:
    - source: salt://conf/myapp/myapp.conf
    - mode: keep

/var/www/myapp:
  file.recurse:
    - source: salt://path/to/myapp
    - dir_mode: 755
    - file_mode: keep
```

- The `junos` state module is now available. It has all the functions that are present in the `junos` execution module.
- The `junos` state module is now available. It has all the functions that are present in the `junos` execution module.
- The minion data cache is a pluggable data store now. It's configurable with *`cache`* option. Default is `localfs`.
- User names in `client_acl` support glob matching now.

New Top File Merging Strategy for States

A new strategy called `merge_all` has been added to provide a new way of merging top file matches when executing a *highstate*. See the *`top_file_merging_strategy`* documentation for further information.

In addition, the `same` merging strategy was not functioning as documented. This has now been corrected. While this is technically a bugfix, we decided to hold a change in top file merging until a feature release to minimize user impact.

Improved Archive Extraction Support

The *`archive.extracted`* state has been overhauled. Notable changes include the following:

- When enforcing ownership (with the `user` and/or `group` arguments), the `if_missing` argument no longer has any connection to which path(s) have ownership enforced. Instead, the paths are determined using the either the newly-added *`archive.list`* function, or the newly-added `enforce_ownership_on` argument.
- `if_missing` also is no longer required to skip extraction, as Salt is now able to tell which paths would be present if the archive were extracted. It should, in most cases, only be necessary in cases where a semaphore file is used to conditionally skip extraction of the archive.
- Password-protected ZIP archives are now detected before extraction, and the state fails without attempting to extract the archive if no password was specified.
- By default, a single top-level directory is enforced, to guard against 'tar-bombs'. This enforcement can be disabled by setting `enforce_toplevel` to `False`.
- The `tar_options` and `zip_options` arguments have been deprecated in favor of a single `options` argument.
- The `archive_format` argument is now optional. The ending of the `source` argument is used to guess whether it is a tar, zip or rar file. If the `archive_format` cannot be guessed, then it will need to be specified, but in many cases it can now be omitted.

- Ownership enforcement is now performed irrespective of whether or not the archive needed to be extracted. This means that the state can be re-run after the archive has been fully extracted to repair changes to ownership.

A number of new arguments were also added. See the docs py:func:*docs for the archive.extracted state <salt.states.archive.extracted>* for more information.

Additionally, the following changes have been made to the `archive` execution module:

- A new function (`archive.list`) has been added. This function lists the files/directories in an archive file, and supports a `verbose` argument that gives a more detailed breakdown of which paths are files, which are directories, and which paths are at the top level of the archive.
- A new function (`archive.is_encrypted`) has been added. This function will return `True` if the archive is a password-protected ZIP file, `False` if not. If the archive is not a ZIP file, an error will be raised.
- `archive.cmd_unzip` now supports passing a password, bringing it to feature parity with `archive.unzip`. Note that this is still not considered to be secure, and `archive.unzip` is recommended for dealing with password-protected ZIP archives.
- The default value for the `extract_perms` argument to `archive.unzip` has been changed to `True`.

Improved Checksum Handling in `file.managed`, `archive.extracted` States

When the `source_hash` argument for these states refers to a file containing checksums, Salt now looks for checksums matching the name of the source URI, as well as the file being managed. Prior releases only looked for checksums matching the filename being managed. Additionally, a new argument (`source_hash_name`) has been added, which allows the user to disambiguate ambiguous matches when more than one matching checksum is found in the `source_hash` file.

A more detailed explanation of this functionality can be found in the `file.managed` documentation, in the section for the new `source_hash_name` argument.

Note: This improved functionality is also available in the `2016.3` (Boron) release cycle, starting with the 2016.3.5 release.

Config Changes

The following default config values were changed:

- `gitfs_ssl_verify`: Changed from `False` to `True`
- `git_pillar_ssl_verify`: Changed from `False` to `True`
- `winrepo_ssl_verify`: Changed from `False` to `True`

Grains Changes

- All core grains containing `VMWare` have been changed to `VMware`, which is the official capitalization. Additionally, all references to `VMWare` in the documentation have been changed to `VMware` issue #30807. Environments using versions of Salt before and after Salt 2016.11.0 should employ case-insensitive grain matching on these grains.

```
{% set on_vmware = grains['virtual'].lower() == 'vmware' %}
```

- On Windows the `cpu_model` grain has been changed to provide the actual cpu model name and not the cpu family.

  Old behavior:

  ```
  root@master:~# salt 'testwin200' grains.item cpu_model
  testwin200:
      ----------
      cpu_model:
          Intel64 Family 6 Model 58 Stepping 9, GenuineIntel
  ```

  New behavior:

  ```
  root@master:~# salt 'testwin200' grains.item cpu_model
  testwin200:
      ----------
      cpu_model:
          Intel(R) Core(TM) i7-3520M CPU @ 2.90GHz
  ```

Beacons Changes

- The `loadavg` beacon now outputs averages as integers instead of strings. (Via issue #31124.)

Runner Changes

- Runners can now call out to *utility modules* via `__utils__`.
- ref:*Utility modules <writing-utility-modules>* (placed in `salt://_utils/`) are now able to be synced to the master, making it easier to use them in custom runners. A *`saltutil.sync_utils`* function has been added to the *`saltutil runner`* to facilitate the syncing of utility modules to the master.

Pillar Changes

- Thanks to the new *`saltutil.sync_utils`* runner, it is now easier to get ref:*utility modules <writing-utility-modules>* synced to the correct location on the Master so that they are available in execution modules called from Pillar SLS files.

Junos Module Changes

- The following new functionalities were added to the junos module
  - facts - Displays the facts gathered during the connection.
  - shutdown - Shut down or reboot a device running Junos OS.
  - install_config - Modify the configuration of a Junos device.
  - install_os - Install Junos OS software package.
  - zeroize - Remove all configuration information on the Routing Engines and reset all key values on a device.
  - file_copy - Copy file from proxy to the Junos device.

Network Automation: NAPALM

Beginning with 2016.11.0, network automation is included by default in the core of Salt. It is based on a the NAPALM library and provides facilities to manage the configuration and retrieve data from network devices running widely used operating systems such: JunOS, IOS-XR, eOS, IOS, NX-OS etc.

- see the complete list of supported devices.

The connection is established via the *NAPALM proxy*.

In the current release, the following modules were included:

- *NAPALM grains* - Select network devices based on their characteristics
- *NET execution module* - Networking basic features
- *NTP execution module*
- *BGP execution module*
- *Routes execution module*
- *SNMP execution module*
- *Users execution module*
- *Probes execution module*
- *NTP peers management state*
- *SNMP configuration management state*
- *Users management state*

Cisco NXOS Proxy Minion

Beginning with 2016.11.0, there is a proxy minion that can be used to configure nxos cisco devices over ssh.

- *Proxy Minion*
- *Execution Module*
- *State Module*

Cisco Network Services Orchestrator Proxy Minion

Beginning with 2016.11.0, there is a proxy minion to use the Cisco Network Services Orchestrator as a proxy minion.

- *Proxy Minion*
- *Execution Module*
- *State Module*

Junos Module Changes

- The following new functionalities were added to the junos module
  - facts - Displays the facts gathered during the connection.
  - shutdown - Shut down or reboot a device running Junos OS.
  - install_config - Modify the configuration of a Junos device.

  - install_os - Install Junos OS software package.
  - zeroize - Remove all configuration information on the Routing Engines and reset all key values on a device.
  - file_copy - Copy file from proxy to the Junos device.

Returner Changes

- Any returner which implements a *save_load* function is now required to accept a *minions* keyword argument. All returners which ship with Salt have been modified to do so.

Renderer Changes

Added the ability to restrict allowed renderers. Two new config parameters, `renderer_whitelist` and `renderer_blacklist` are introduced for this purpose.

eAuth Changes

- External auth modules' `auth` method can return an ACL list for the given username instead of `True`. This list should be in the same format as described in the *eAuth documentation*. It will be used for the user instead of one set in master config.

  Example of the `auth` method return that allows a user to execute functions in the `test` and `network` modules on the minions that match the `web*` target and allow access to `wheel` and `runner` modules:

  ```
  [{'web*': ['test.*',
             'network.*']},
   '@wheel',
   '@runner']
  ```

- External auth is supported by *salt-run* and *salt-key* now. Note that master must be started to use them with eAuth.

External Module Packaging

Modules may now be packaged via entry-points in setuptools. See *external module packaging* tutorial for more information.

Functionality Changes

- The `onfail` requisite now uses OR logic instead of AND logic. issue #22370
- The consul external pillar now strips leading and trailing whitespace. issue #31165
- The win_system.py state is now case sensitive for computer names. Previously computer names set with a state were converted to all caps. If you have a state setting computer names with lower case letters in the name that has been applied, the computer name will be changed again to apply the case sensitive name.
- The `mac_user.list_groups` function in the `mac_user` execution module now lists all groups for the specified user, including groups beginning with an underscore. In previous releases, groups beginning with an underscore were excluded from the list of groups.
- The `junos.call_rpc` function in the `junos` execution module can now be used to call any valid rpc. Earlier it used to call only "get_software_information".

- A new option for minions called `master_tries` has been added. This specifies the number of times a minion should attempt to contact a master to attempt a connection. This allows better handling of occasional master downtime in a multi-master topology.
- The default hash_type is now sha256 instead of md5. You will need to make sure both your master and minion share the same hash_type.
- Nodegroups consisting of a simple list of minion IDs can now also be declared as a yaml list. The below two examples are equivalent:

```
# Traditional way
nodegroups:
  - group1: L@host1,host2,host3

# New way (optional)
nodegroups:
  - group1:
    - host1
    - host2
    - host3
```

New Azure ARM Cloud Driver

A new cloud driver has been added for Azure ARM, aka, the Azure Resource Manager. The older Azure driver is still required to work with the older Azure API. This new driver works with the newer ARM API, which is managed via the newer Azure Portal website.

New Modules

Beacons

- *salt.beacons.avahi_announce*
- *salt.beacons.bonjour_announce*
- *salt.beacons.haproxy*
- *salt.beacons.status*

Clouds

- *salt.cloud.clouds.azurearm*

Engines

- `salt.engines.hipchat`
- *salt.engines.http_logstash*

Modules

- *salt.modules.boto_cloudwatch_event*

- *salt.modules.celery*
- *salt.modules.ceph*
- salt.modules.influx08
- *salt.modules.inspectlib.entities*
- *salt.modules.inspectlib.fsdb*
- *salt.modules.inspectlib.kiwiproc*
- *salt.modules.inspector*
- *salt.modules.libcloud_dns*
- *salt.modules.openstack_mng*
- *salt.modules.servicenow*
- *salt.modules.testinframod*
- *salt.modules.win_lgpo*
- *salt.modules.win_pki*
- *salt.modules.win_psget*
- *salt.modules.win_snmp*
- *salt.modules.xbpspkg*

Outputters

- *salt.output.pony*

Pillar

- *salt.pillar.csvpillar*
- *salt.pillar.http_json*
- *salt.pillar.makostack*

Returners

- *salt.returners.zabbix_return*

Runners

- *salt.runners.auth*
- *salt.runners.event*
- *salt.runners.smartos_vmadm*
- *salt.runners.vistara*

SDB

- *salt.sdb.env*

States

- *salt.states.boto_cloudwatch_event*
- *salt.states.csf*
- *salt.states.ethtool*
- *salt.states.influxdb08_database*
- *salt.states.influxdb08_user*
- *salt.states.libcloud_dns*
- *salt.states.snapper*
- *salt.states.testinframod*
- *salt.states.win_lgpo*
- *salt.states.win_pki*
- *salt.states.win_snmp*

Thorium

- *salt.thorium.calc*
- *salt.thorium.key*
- *salt.thorium.runner*
- *salt.thorium.status*
- *salt.thorium.wheel*

Deprecations

General Deprecations

- `env` to `saltenv`

  All occurrences of `env` and some occurrences of `__env__` marked for deprecation in Salt 2016.11.0 have been removed. The new way to use the salt environment setting is with a variable called `saltenv`:

  ```
  def fcn(msg='', env='base', refresh=True, saltenv='base', **kwargs):
  ```

  has been changed to

  ```
  def fcn(msg='', refresh=True, saltenv='base', **kwargs):
  ```

  - If `env` (or `__env__`) is supplied as a keyword argument to a function that also accepts arbitrary keyword arguments, then a new warning informs the user that `env` is no longer used if it is found. This new warning will be removed in Salt 2017.7.0.

```
def fcn(msg='', refresh=True, saltenv='base', **kwargs):
```

```
# will result in a warning log message
fcn(msg='add more salt', env='prod', refresh=False)
```

  - If `env` (or `__env__`) is supplied as a keyword argument to a function that does not accept arbitrary keyword arguments, then python will issue an error.

```
def fcn(msg='', refresh=True, saltenv='base'):
```

```
# will result in a python TypeError
fcn(msg='add more salt', env='prod', refresh=False)
```

  - If `env` (or `__env__`) is supplied as a positional argument to a function, then undefined behavior will occur, as the removal of `env` and `__env__` from the function's argument list changes the function's signature.

```
def fcn(msg='', refresh=True, saltenv='base'):
```

```
# will result in refresh evaluating to True and saltenv likely not being a string at all
fcn('add more salt', 'prod', False)
```

- Deprecations in `minion.py`:
  - The `salt.minion.parse_args_and_kwargs` function has been removed. Please use the `salt.minion.load_args_and_kwargs` function instead.

Cloud Deprecations

- The `vsphere` cloud driver has been removed. Please use the `vmware` cloud driver instead.
- The `private_ip` option in the `linode` cloud driver is deprecated and has been removed. Use the `assign_private_ip` option instead.
- The `create_dns_record` and `delete_dns_record` functions are deprecated and have been removed from the `digital_ocean` driver. Use the `post_dns_record` function instead.

Execution Module Deprecations

- The `blockdev` execution module had four functions removed:
  - dump
  - tune
  - resize2fs
  - wipe

  The `disk` module should be used instead with the same function names.
- The `boto_vpc` execution module had two functions removed, `boto_vpc.associate_new_dhcp_options_to_vpc` and `boto_vpc.associate_new_network_acl_to_subnet` in favor of more concise function names, `boto_vpc.create_dhcp_options` and `boto_vpc.create_network_acl`, respectively.

- The `data` execution module had `getval` and `getvals` functions removed in favor of one function, `get`, which combines the functionality of the removed functions.
- File module deprecations:
    - The `contains_regex_multiline` function was removed. Use `file.search` instead.
    - Additional command line options for `file.grep` should be passed one at a time. Please do not pass more than one in a single argument.
- The `lxc` execution module has the following changes:
    - The `run_cmd` function was removed. Use `lxc.run` instead.
    - The `nic` argument was removed from the `lxc.init` function. Use `network_profile` instead.
    - The `clone` argument was removed from the `lxc.init` function. Use `clone_from` instead.
    - passwords passed to the `lxc.init` function will be assumed to be hashed, unless `password_encrypted=False`.
    - The `restart` argument for `lxc.start` was removed. Use `lxc.restart` instead.
    - The old style of defining lxc containers has been removed. Please use keys under which LXC profiles should be configured such as `lxc.container_profile.profile_name`.
- The `env` and `activate` keyword arguments have been removed from the `install` function in the `pip` execution module. The use of `bin_env` replaces both of these options.
- `reg` execution module

  Functions in the `reg` execution module had misleading and confusing names for dealing with the Windows registry. They failed to clearly differentiate between hives, keys, and name/value pairs. Keys were treated like value names. There was no way to delete a key.

  New functions were added in 2015.5 to properly work with the registry. They also made it possible to edit key default values as well as delete an entire key tree recursively. With the new functions in place, the following functions have been deprecated:

    - read_key
    - set_key
    - create_key
    - delete_key

  Use the following functions instead:

    - for `read_key` use `read_value`
    - for `set_key` use `set_value`
    - for `create_key` use `set_value` with no `vname` and no `vdata`
    - for `delete_key` use `delete_key_recursive`. To delete a value, use `delete_value`.
- The `hash_hostname` option was removed from the `salt.modules.ssh` execution module. The `hash_known_hosts` option should be used instead.
- The `human_readable` option was removed from the `uptime` function in the `status` execution module. The function was also updated in 2015.8.9 to return a more complete offering of uptime information, formatted as an easy-to-read dictionary. This updated function replaces the need for the `human_readable` option.
- The `persist` kwarg was removed from the `win_useradd` execution module. This option is no longer supported for Windows. `persist` is only supported as part of user management in UNIX/Linux.

- The `zpool_list` function in the `zpool` execution module was removed. Use `list` instead.

Outputter Module Deprecations

- The `compact` outputter has been removed. Set `state_verbose` to `False` instead.

Runner Module Deprecations

- The `grains.cache` runner no longer accepts `outputter` or `minion` as keyword arguments. Users will need to specify an outputter using the `--out` option. `tgt` is replacing the `minion` kwarg.
- The `fileserver` runner no longer accepts the `outputter` keyword argument. Users will need to specify an outputter using the `--out` option.
- The `jobs` runner no longer accepts the `outputter` keyword argument. Users will need to specify an outputter using the `--out` option.
- `virt` runner module:
  - The `hyper` kwarg was removed from the `init`, `list`, and `query` functions. Use the `host` option instead.
  - The `next_hyper` function was removed. Use the `next_host` function instead.
  - The `hyper_info` function was removed. Use the `host_info` function instead.

State Module Deprecations

- The `env` and `activate` keyword arguments were removed from the `installed` function in the `pip` state module. The use of `bin_env` replaces both of these options.
- `reg` state module

  The `reg` state module was modified to work with the new functions in the execution module. Some logic was left in the `reg.present` and the `reg.absent` functions to handle existing state files that used the final key in the name as the value name. That logic has been removed so you now must specify value name (`vname`) and, if needed, value data (`vdata`).

  For example, a state file that adds the version value/data pair to the Software\Salt key in the HKEY_LOCAL_MACHINE hive used to look like this:

  ```
  HKEY_LOCAL_MACHINE\\Software\\Salt\\version:
    reg.present:
      - value: 2016.3.1
  ```

  Now it should look like this:

  ```
  HKEY_LOCAL_MACHINE\\Software\\Salt
    reg.present:
      - vname: version
      - vdata: 2016.3.1
  ```

  A state file for removing the same value added above would have looked like this:

  ```
  HKEY_LOCAL_MACHINE\\Software\\Salt\\version:
    reg.absent:
  ```

Now it should look like this:

```
HKEY_LOCAL_MACHINE\\Software\\Salt
  reg.absent:
    - vname: version
```

This new structure is important as it allows salt to deal with key default values which was not possible before. If vname is not passed, salt will work with the default value for that hivekey.

Additionally, since you could only delete a value from a the state module, a new function (`key_absent`) has been added to allow you to delete a registry key and all subkeys and name/value pairs recursively. It uses the new `delete_key_recursive` function.

For additional information see the documentation for the `reg` execution and state modules.

- `lxc` state module: The following functions were removed from the `lxc` state module:
  - `created`: replaced by the `present` state.
  - `started`: replaced by the `running` state.
  - `cloned`: replaced by the `present` state. Use the `clone_from` argument to set the name of the clone source.
- The `hash_hostname` option was removed from the `salt.states.ssh_known_hosts` state. The `hash_known_hosts` option should be used instead.
- The `always` kwarg used in the `built` function of the `pkgbuild` state module was removed. Use `force` instead.

Utils Module Deprecations

- The use of `jid_dir` and `jid_load` were removed from the `salt.utils.jid`. `jid_dir` functionality for job_cache management was moved to the `local_cache` returner. `jid_load` data is now retrieved from the `master_job_cache`.
- `ip_in_subnet` function in `salt.utils.network.py` has been removed. Use the `in_subnet` function instead.
- The `iam` utils module had two functions removed: `salt.utils.iam.get_iam_region` and `salt.utils.iam.get_iam_metadata` in favor of the aws utils functions `salt.utils.aws.get_region_from_metadata` and `salt.utils.aws.creds`, respectively.

28.2.31 Salt 2016.11.1 Release Notes

Version 2016.11.1 is a bugfix release for *2016.11.0*.

Statistics

- Total Merges: **89**
- Total Issue References: **29**
- Total PR References: **83**
- Contributors: **30** (Ch3LL, Da-Juan, DmitryKuzmenko, MTecknology, adelcast, attiasr, bbinet, cachedout, cro, dmurphy18, gtmanfred, isbm, jeanpralo, kraney, kstreee, lorengordon, mateiw, mirceaulinic, morsik, mschneider82, rallytime, rbjorklin, scott-w, sjorge, skizunov, techhat, terminalmage, thatch45, ticosax, whiteinge)

Changelog for v2016.11.0..v2016.11.1

Generated at: 2018-05-27 14:25:03 UTC

- **PR** #38186: (Ch3LL) add 2016.11.1 changelog to release notes
- **PR** #38182: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2016-12-09 21:25:47 UTC*
  - 23c039347e Merge pull request #38182 from rallytime/merge-2016.11
  - 627242ab5d Merge branch '2016.3' into '2016.11'
    - 65b2ad7b14 Merge pull request #38163 from Ch3LL/enabled_ec2_cloud
      - be74c45463 enabled ec2 cloud tests
    - b63f74e034 Merge pull request #38177 from vutny/fix-cp-get-file-str
      - a449980672 Correct *cp.get_file_str* docstring and add integration tests
    - 7596313be0 Merge pull request #38153 from vutny/master-includes-error-tolerance
      - cd0154ee93 Master config includes may contain errors and be safely skipped
- **ISSUE** #38094: (bfilipek) TypeError: object of type 'float' has no len() in grains.filter_by (refs: #38158)
- **PR** #38158: (cachedout) Fix type problem in grains.filter_by @ *2016-12-09 21:24:40 UTC*
  - 8355adc535 Merge pull request #38158 from cachedout/issue_38094
  - e8196e23c2 Lint, remove set literal
  - 9f4ebb3c18 Fix type problem in grains.filter_by
- **ISSUE** #38090: (jf) pkg.installed does not seem to refresh the repo database, no matter what (refs: #38113, #38156)
- **PR** #38156: (terminalmage) Remove rtag when windows minion refreshes early in state @ *2016-12-09 21:15:01 UTC*
  - 31a157d902 Merge pull request #38156 from terminalmage/fix-windows-refresh
  - 258bd4c2aa Remove rtag when windows minion refreshes early in state
- **ISSUE** #37981: (tazaki) Salt-cloud ec2 vpc securitygroupid always returning default (refs: #38183)
- **PR** #38183: (cro) Fix bad set operations when setting up securitygroups in AWS. @ *2016-12-09 21:12:10 UTC*
  - c638952684 Merge pull request #38183 from cro/fix_37891
  - 0527d6f25e Fix bad set operations when setting up securitygroups in AWS. Fixes #37891.
  - **PR** #38181: (rallytime) Reset socket default timeout to None (fixes daemons_tests failures)
- **PR** #38148: (whiteinge) Remove ssh_async from NetapiClient clients; it is not implemented @ *2016-12-09 18:49:42 UTC*
  - 7ccbedd2cc Merge pull request #38148 from whiteinge/no-ssh-async-client
  - cb58cd4795 Remove ssh_async from NetapiClient clients; it is not implemented
- **PR** #38160: (terminalmage) Update information about xz-utils in archive state/module docs @ *2016-12-09 18:34:03 UTC*
  - 8d4e194400 Merge pull request #38160 from terminalmage/update-archive-docs
  - 8e4ad3cff3 Update information about xz-utils in archive state/module docs
- **ISSUE** #38024: (Ch3LL) 2016.11.0 release notes missing azure arm reference (refs: #38164)

- **PR** #38164: (techhat) Add Azure ARM docs for 2016.11.0 @ *2016-12-09 18:00:22 UTC*
  - 05136f0d8c Merge pull request #38164 from techhat/azuredocs
  - 71b787e250 Add Azure ARM docs for 2016.11.0
- **PR** #38173: (rallytime) Bump some win* module deprecations from Nitrogen to Oxygen @ *2016-12-09 16:57:29 UTC*
  - e3c858cc28 Merge pull request #38173 from rallytime/update-win-deprecation-versions
  - 09a50b25e7 Bump some win* module deprecations from Nitrogen to Oxygen
- **PR** #38036: (terminalmage) archive.extracted: fix problems with overwrite arg @ *2016-12-08 19:08:41 UTC*
  - **PR** #37889: (isbm) Allow overwrite archives extraction (refs: #38036)
  - 827bf59999 Merge pull request #38036 from terminalmage/archive-extracted-override
  - a1c70c7b95 archive.extracted: fix problems with overwrite arg
- **PR** #38133: (terminalmage) Fix edge case in creation of trans tar for salt-thin @ *2016-12-08 17:47:26 UTC*
  - 50773a5f96 Merge pull request #38133 from terminalmage/zd1067
  - 71e0bd023f Fix edge case in creation of trans tar for salt-thin
- **PR** #38138: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2016-12-07 20:15:56 UTC*
  - 6026cb23b2 Merge pull request #38138 from rallytime/merge-2016.11
  - 28b56ea3b4 Merge branch '2016.3' into '2016.11'
  - 86091db647 Skip daemon unit tests when running on Python 2.6 (#38134)
- **PR** #38130: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2016-12-07 20:11:19 UTC*
  - 90478ef25e Merge pull request #38130 from rallytime/merge-2016.11
  - 4d7d9abb41 Merge branch '2016.3' into '2016.11'
    - d3d98fd4eb Merge pull request #38102 from rallytime/fix-38091
      - 4f79d5a0d1 Add False + msg tuple return if requests is missing for zenoss module
    - 8c8cbc2734 Merge pull request #38104 from rallytime/bp-36794
      - c906c8a0d5 Pylint fixes
      - da3ebf83e6 FreeBSD sysctl module now handels config_file parameter in show method
  - **PR** #38129: (Ch3LL) Fix beacon index
  - **PR** #38127: (rallytime) Add versionadded tags for network module funcs
- **ISSUE** #38042: (MTecknology) [2016.11.0] Invalid interfaces file produced by debian_ip module (refs: #38043)
- **PR** #38043: (MTecknology) Debian networking fix @ *2016-12-07 17:32:18 UTC*
  - fd06bab673 Merge pull request #38043 from MTecknology/2016.11
  - 6d5e132e44 Removing trailing whitespace from previous commit
  - f882674acf Adding some options that are valid for inet6 blocks.
  - 81cb688d4c Better check for dual stack.
  - 525c746274 May Cthulhu take mercy on my soul for this commit.
  - 300ca6047e I guess this makes the previous commit a bit redundant, but I'm not sure if I want to remove it.

  - 6e7fc39c68 This now seems absurdly obvious, but I'm not ruling out that I'll break everything.
  - 82d2b89e0c Rolling back unit test.
  - b3edbcfd05 Adding larger and more complete debian_ip unit test.
  - 3afd7b6cf4 Adding the valid/documented 'slaves' option.
  - b6b1adc091 Typo: missing closing parenthesis
  - 756e41caf2 Fixing a typo; line should not be commented
  - 32a1374748 Corrects expected return value
  - 88f9d9f22c Mostly whitespace & comment changes
  - 41ffb8d805 Removing redundant line
  - 3a8168667b Ensure iface_dict not being populated will not produce a stacktrace
  - 4de2cb2805 Corrects regression in debian_ip/debian_eth.jinja
- **PR** #38107: (cachedout) Status beacon should raise proper exception @ *2016-12-07 17:21:49 UTC*
  - **PR** #38088: (dmurphy18) Updated to match formulas and allow for missing functions (refs: #38107)
  - 4b9a7f2295 Merge pull request #38107 from cachedout/supercede_38088
  - 73d724845d Change to log.debug per Tom
  - da135b1b59 Fix docs
  - 792b422dc2 Pylint fix
  - 88e03bba6d Fix typo
  - a8ce153252 Status beacon should raise proper exception
- **PR** #38101: (lorengordon) Clarifies file.replace behavior on symlinks @ *2016-12-07 13:27:11 UTC*
  - da8f5ac0c6 Merge pull request #38101 from lorengordon/file-replace-note
  - 345990f2b0 Clarifies file.replace behavior on symlinks
- **ISSUE** #38090: (jf) pkg.installed does not seem to refresh the repo database, no matter what (refs: #38113, #38156)
- **PR** #38113: (terminalmage) Revert changes to refresh tag for pkg states @ *2016-12-07 13:11:14 UTC*
  - d47761f349 Merge pull request #38113 from terminalmage/issue38090
  - 9f347df012 Revert changes to refresh tag for pkg states
- **ISSUE** #37976: (t0nyhays) Error when status beacon fires (2016.11.0) (refs: #38120)
- **PR** #38120: (Da-Juan) Fix status beacon config default values @ *2016-12-07 13:08:33 UTC*
  - d4c34e0a58 Merge pull request #38120 from Da-Juan/2016.11
  - 7e4a35e8ad Fix status beacon config default values
- **PR** #38114: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2016-12-07 12:45:04 UTC*
  - 6868089a87 Merge pull request #38114 from rallytime/merge-2016.11
  - fec9dec23a Merge branch '2016.3' into '2016.11'
    - fbc87769b9 Merge pull request #38083 from twangboy/fix_refresh_db
      - 978af6d83c Remove only .sls files from the cached winrepo-ng

        - 9dcfdeef6b Merge pull request #38059 from rallytime/daemons-test-fix
            - eb372b27d8 Add missing "not" statement: The last syndic test should assertFalse()
            - 4e10f8e018 Call exec_test for the Syndic daemon in tests.unit.daemons_test.py
        - 9cd42b9b3f Merge pull request #38039 from rallytime/fix-37939
            - 1da7aacfbe Update unit tests to account for additional file.search call
            - 8a685b1820 Check to see if a line is already commented before moving on
            - f2c045520d Write an integration test demonstrating the issue
        - a34a763984 Merge pull request #38045 from terminalmage/issue38037
            - 65289503d9 Simplify logic for matching desired pkg arch with actual pkg arch
            - 3babbcda94 yumpkg.py: don't include non-upgrade versions found by "yum list available"
- **PR** #38109: (gtmanfred) mode needs to be an integer @ *2016-12-07 11:58:24 UTC*
    - b9920e54ee Merge pull request #38109 from gtmanfred/2016.11
    - 7546760eb3 mode needs to be an integer
- **PR** #38103: (rallytime) Back-port #37283 to 2016.11 @ *2016-12-06 23:12:59 UTC*
    - **PR** #37283: (jeanpralo) Handle docker-compose up to version 1.9.0 (refs: #38103)
    - **PR** #37215: (mschneider82) removed version check (refs: #37283)
    - fd77dcbd0f Merge pull request #38103 from rallytime/bp-37283
    - 11944df69b handle up to version 1.9.0
- **PR** #38057: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 (refs: #38059) @ *2016-12-06 23:11:41 UTC*
    - 5d9d6b9280 Merge pull request #38057 from rallytime/merge-2016.11
    - 342884018b Fix SaltKeyOptionParserTestCase test failures
    - 186e2d0d03 Don't allow libcloud mock module injection in unit/states/libcloud_dns_test.py either
    - d513a60189 Do not allow libcloud to be injected as a mock value in the libcloud_dns_test
    - 74a417e527 Update the mocked cloud configs to also include master configs
    - f2c8cb13d0 Better merge conflict resolution from the initial merge
    - 8fd53a4808 Merge branch '2016.3' into '2016.11'
        - 6724fe4871 Modify daemons test to use multiprocessing (#38034)
        - 6942d5d95b Merge pull request #37995 from rallytime/merge-2016.3
            - b44e17921c Merge branch '2015.8' into '2016.3'
            - 7a7e36728f Merge pull request #37978 from terminalmage/ext_pillar_first-docs
            - 61ed9a8657 Add clarifying language to ext_pillar_first docs
        - cd66c179cb fix broken yaml code block (#38002)
        - 3dd45fbedf Merge pull request #37912 from attiasr/fix_aws_response_encoding
            - ba4ec4e7f1 use Requests result encoding to encode the text
            - abe4eb3b98 fix encoding problem aws responses

- 69a74a4d2d Merge pull request #37950 from vutny/fix-starting-up-syndic
  - 7d9bc9abce syndic_master: correct default value, documentation and example config
  - 92a7c7ed1b Set default Salt Master address for a Syndic (like for a Minion)
- 7f269bc7f9 Add clarification on expr_form usage and future deprecation (#37964)
- 1001987f64 Catch possible exception from lsb_release (#37962)
- 330021cd8b Handle empty tokens safely (#37961)
- ea46639ce7 Merge pull request #37272 from vutny/fix-getting-default-logging-opts
  - e5ce52388a Fix description in the Salt Syndic usage info
  - 518a3dd7ee Add unit tests for Salt parsers processing logging options
  - 83d6a44254 Add *ssh_log_file* option to master config and documentation
  - c8a0915460 Fix configuration example and documentation for *syndic_log_file* option
  - e64dd3ed6b Correct default attributes for various parser classes
  - 82a2e216b3 Fix default usage string for Salt command line programs
  - 45dffa292f Fix readding and updating logfile and pidfile config options for Salt API
  - f47253c21b Fix reading and applying Salt Cloud default configuration
  - fad5bec936 Work with a copy of default opts dictionaries
  - b7c24811e5 Fix *log_level_logfile* config value type
  - 1bd76a1d96 Fix setting temporary log level if CLI option omitted
  - 121848cc77 Fix obtaining *log_granular_levels* config setting
  - 44cf07fec2 Make CLI options take precedence for setting up logfile_logger
  - 61afaf1792 Fix setting option attributes when processing *log_level* and *log_file*
  - 3c60e2388e Fix processing of *log_level_logfile* config setting
  - 55a0af5bbd Use attribute functions for getting/setting options and config values
  - c25f2d091e Fix getting Salt API default logfile option
  - f2422373c1 Remove processing of unused and undocumented *cli_*_log_** config options
  - 2065e8311c Get default logging level and file from default opts dict
- f2f957da6c Merge pull request #37925 from kontrolld/add-ipv6-centos-network
  - ac2b477412 Adding IPv6 functionality for CentOS /etc/sysconfig/network
- c07ad11279 Merge pull request #37899 from DSRCorporation/bugs/37059_schedule_task_hang
  - 9497748546 Clear functions context in schedule tasks for ZeroMQ.
- a55519db40 Merge pull request #37928 from techhat/issue37737
  - a09a60e89b Don't modify self.opts directly
- 9d17f1ce90 Merge pull request #37929 from gtmanfred/2016.3
  - c7d2c73503 add list_nodes_min to nova driver
- 3bb743b59f Merge pull request #37926 from kontrolld/fix-ipv6-centos-network
  - 3ed42e5b44 updated

        - 3b3bc4f239 Fixes no IPv6 functionality in /etc/sysconfig/network
      - 271170a9f3 Merge pull request #37921 from rallytime/merge-2016.3
        - 523a67c422 Merge branch '2015.8' into '2016.3'
        - 4cdc6cf5ec Update earlier release channels' docs with Carbon release notes (#37914)
        - d31491a7fe [2015.8] Update version numbers in doc config for 2016.11.0 release (#37918)
      - 6cd6429ac0 Merge pull request #37924 from cachedout/fix_gem_states
        - 894cca3427 Update test for new gem ver
  - **PR** #38112: (rallytime) Account for case where vim install already exists and is at an older version
- **ISSUE** #19502: (kt97679) salt-ssh fails to run state.highstate with custom master_tops (refs: #38021)
- **PR** #38021: (mateiw) Add master_tops support in salt-ssh @ *2016-12-06 14:26:22 UTC*
  - f8c67a9598 Merge pull request #38021 from mateiw/salt-ssh_master_tops
  - 65a0f102fd Add/remove newlines
  - 7037fa116d Add master_tops support in salt-ssh
  - **PR** #38084: (rallytime) Start release notes file for 2016.11.1 release
- **PR** #37878: (kstreee) Makes threads avoid blocking waiting while communicating using Zeromq. @ *2016-12-05 19:50:46 UTC*
  - 78295516e7 Merge pull request #37878 from kstreee/2016.11
  - 9103878c4f Fixes blocking waiting through implementing a socket pool class.
- **PR** #37987: (rbjorklin) consul_pillar support for limiting pillar exposure via minion targeting @ *2016-12-05 19:48:20 UTC*
  - **PR** #37985: (rbjorklin) consul_pillar support for limiting pillar exposure via minion targeting (refs: #37987)
  - 0809ccd429 Merge pull request #37987 from rbjorklin/consul-pillar-target
  - 5d0454a7ca Ignore W1401 (anomalous-backslash-in-string)
  - 2e929a5ecc Linting fixes
  - 171cab1726 Fixed possible incorrect behavior if target wasn't on start/end of str
  - 7440582ce8 consul_pillar support for limiting pillar exposure via minion targeting
- **ISSUE** #38062: (UtahDave) archive execution module not loading on Windows in 2016.11.0 (refs: #38067)
- **PR** #38067: (terminalmage) Remove virtual funcs for archive state/module @ *2016-12-05 16:37:23 UTC*
  - 83dcfe81ea Merge pull request #38067 from terminalmage/issue38062
  - 2e0f26a084 Remove virtual funcs for archive state/module
- **ISSUE** #38001: (tomlaredo) Regression on postgres_group.present ('postgres_group' __virtual__ returned False) (refs: #38023)
- **ISSUE** #37986: (marek-obuchowicz) Module postgres - wrong docs, doesn't work with debian 8.5 (refs: #38023)
- **ISSUE** #37935: (ipmb) Postgres module regression on 2016.11 (refs: #37946, #37993, #38023, #38058)
- **PR** #38058: (rallytime) Remove initdb dependency in postgres module @ *2016-12-04 04:19:02 UTC*
  - **PR** #38023: (gtmanfred) Expand error message for postgres states (refs: #38058)

  - **PR** #37993: (ticosax) Remove initdb dependency to consume postgres module. (refs: #38058)
  - c9933670f9 Merge pull request #38058 from rallytime/remove-init-db-dep
  - c1ceeca3d3 Remove initdb dependency in postgres module
- **ISSUE** #37969: (lordcirth) Archive.extracted fails if -user: root is specified (refs: #38004)
- **PR** #38004: (terminalmage) Fix regression in user/group mgmt for archive.extracted @ *2016-12-02 18:28:49 UTC*
  - 1ac53e5196 Merge pull request #38004 from terminalmage/issue37969
  - 23bb90a7ce Add integration test for archive.extracted with user/group set to root
  - e5ee721696 Don't use simple boolean check on uid/gid
- **ISSUE** #37941: (L4rS6) Outdated documentation for 2016.11.x (refs: #38051)
- **PR** #38051: (Ch3LL) add docs for hash_type change to sha256 @ *2016-12-02 18:11:36 UTC*
  - e90cbbef08 Merge pull request #38051 from Ch3LL/fix_hash_docs
  - e95f88fbe3 add docs for hash_type change to sha256
- **ISSUE** #38000: (morganwillcock) 2016.11.0: saltutil.runner returns a different dict structure and breaks template rendering (refs: #38028)
- **PR** #38028: (terminalmage) Pass full_return to saltutil.runner @ *2016-12-02 09:49:31 UTC*
  - 1b52289508 Merge pull request #38028 from terminalmage/issue38000
  - 9bf13d55b4 Pass full_return to saltutil.runner
- **ISSUE** #37980: (tveastman) Having 'git' in fileserver_backends and no gitfs_remotes defined causes a crash (refs: #38044)
- **PR** #38044: (terminalmage) Remove debugging code @ *2016-12-02 09:43:44 UTC*
  - 41c44ff684 Merge pull request #38044 from terminalmage/issue37980
  - f70a0409b3 Remove debugging code
- **PR** #38035: (dmurphy18) Updated to return status from make_repo similar to rpmbuild.py @ *2016-12-01 22:30:53 UTC*
  - 9661258f22 Merge pull request #38035 from dmurphy18/fix_debbuild
  - 3bca96e7f2 Updated to return status from make_repo similar to rpmbuild.py
- **ISSUE** #38001: (tomlaredo) Regression on postgres_group.present ('postgres_group' __virtual__ returned False) (refs: #38023)
- **ISSUE** #37986: (marek-obuchowicz) Module postgres - wrong docs, doesn't work with debian 8.5 (refs: #38023)
- **ISSUE** #37935: (ipmb) Postgres module regression on 2016.11 (refs: #37946, #37993, #38023, #38058)
- **PR** #38023: (gtmanfred) Expand error message for postgres states (refs: #38058) @ *2016-12-01 22:05:06 UTC*
  - 141b5c5656 Merge pull request #38023 from gtmanfred/2016.11
  - 1aa43eba80 Expand error message for postgres states
  - ac72ee600e Revert "Updated the bins_dir to default to pg_bin #37935"
- **PR** #38026: (rallytime) Back-port #38015 to 2016.11 @ *2016-12-01 19:16:15 UTC*
  - **PR** #38015: (morsik) Typo fix (refs: #38026)
  - 79486421f5 Merge pull request #38026 from rallytime/bp-38015

  - 11becf3e68 Typo fix
  - **PR** #38022: (DmitryKuzmenko) Added Carbon release notes. Fixed sphinx errors in the file.
  - **PR** #38011: (rallytime) Adjust code examples to use the actual bootstrap-salt.sh file name
- **ISSUE** #37940: (alex-zel) dockerng.sls_build fails on some distributions (refs: #37954)
- **PR** #37954: (gtmanfred) use sleep from path for docker.sls_build @ *2016-11-30 18:08:45 UTC*
  - 0a041277ea Merge pull request #37954 from gtmanfred/2016.11
  - 9caf0b406d use sleep from path for docker.sls_build
- **ISSUE** #37935: (ipmb) Postgres module regression on 2016.11 (refs: #37946, #37993, #38023, #38058)
- **PR** #37993: (ticosax) Remove initdb dependency to consume postgres module. (refs: #38058) @ *2016-11-30 18:08:13 UTC*
  - 4ef5c98845 Merge pull request #37993 from ticosax/remove-initdb-requirement
  - c5c7a53d72 Remove initdb dependency to consume postgres module.
- **PR** #37997: (cachedout) Update gem test for 2016.11 @ *2016-11-30 17:13:45 UTC*
  - 2e5565685c Merge pull request #37997 from cachedout/gem_test_carbon
  - 1d221aa91c Update gem test for 2016.11
- **ISSUE** #36723: (white-hat) ext_pillar_first option is broken in 2016.3 (refs: #36807)
- **ISSUE** #24501: (astehlik) Order in top.sls file is not respected for pillar data in local mode (refs: #31316)
- **ISSUE** #19332: (QuinnyPig) Nondeterminism in Pillar (refs: #31316)
- **PR** #37979: (terminalmage) Revert addition of pillar_roots_override_ext_pillar @ *2016-11-30 14:34:24 UTC*
  - **PR** #36807: (terminalmage) Fix pillar merging when ext_pillar_first is enabled (refs: #37979)
  - **PR** #31316: (kraney) Let ext_pillar_first determine the override order (refs: #37979)
  - ca3a9488f1 Merge pull request #37979 from terminalmage/revert-pillar-change
  - 6135dfa4dd Revert addition of pillar_roots_override_ext_pillar
  - **PR** #37970: (rallytime) Back-port #37958 to 2016.11
  - **PR** #37958: (mirceaulinic) Fix RST link format in Carbon release notes (refs: #37970)
  - **PR** #37971: (rallytime) Lint 2016.11 sooner rather than later
  - **PR** #37955: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 (refs: #37971)
- **ISSUE** #37935: (ipmb) Postgres module regression on 2016.11 (refs: #37946, #37993, #38023, #38058)
- **PR** #37946: (scott-w) Updated the bins_dir to default to pg_bin @ *2016-11-29 16:48:27 UTC*
  - 36f91408c5 Merge pull request #37946 from scott-w/37935-fix-bin-dir
  - d33d403969 Restored missing initdb #37935
  - a041b9f8e8 Use Salt deprecation warning #37935
  - a96789353f Updated the bins_dir to default to pg_bin #37935
- **PR** #37889: (isbm) Allow overwrite archives extraction (refs: #38036) @ *2016-11-29 16:18:57 UTC*
  - d8650c5474 Merge pull request #37889 from isbm/isbm-states-archive-fix
  - e67706bd29 Document the behaviour.

  - 1970814111 Prevent crash during externally changed archive permissions
  - 91b42578b2 Add overwrite option so the extraction of the archive can be always performed.
  - e6958f7f15 Remove nonsense comment and react on generally absent path name
- **PR** #37869: (isbm) Input sanitation (16.11) @ *2016-11-29 16:17:16 UTC*
  - e2b9e58d30 Merge pull request #37869 from isbm/isbm-input-sanitation-16.11
  - f9ec5d68af Use six instead of builtins
  - 203dfcb238 Use American spelling instead
  - 91ed307af9 Sanitise input for the keys and IDs
  - 86623f913d Add a stub for ID sanitiser (at the moment same as hostname)
  - 637144c841 Rename "general.py" to "sanitisers.py"
  - f2571fc8bf Add hostname sanitiser
  - 3ae086aff4 Add filename sanitiser
  - 816b1d1977 Add general sanitisers
- **PR** #37884: (isbm) Do not include "gpg-pubkey" packages, filtering by their name @ *2016-11-28 21:11:37 UTC*
  - e539a94a56 Merge pull request #37884 from isbm/isbm-zypper-gpgkey-pkg-filter
  - 038374a586 Do not include "gpg-pubkey" packages, filtering by their name
- **PR** #37882: (attiasr) multiple issues in boto_rds state and module @ *2016-11-28 21:09:11 UTC*
  - eb3d81a1de Merge pull request #37882 from attiasr/fix_missing_tags
  - 73b3c5fa1a Add newline
  - 166c42bc51 fix boto_rds.describe
  - ddd88ba047 fix boto_rds.describe parameters and subnetgroup_present
  - bfe7f92cb4 fix missing tags in call to boto_rds.exists
  - **PR** #37931: (rallytime) Remove release candidate doc ref from 2016.11.0 release notes
- **PR** #37930: (cachedout) Remove dictionary comprehension in netusers @ *2016-11-28 20:27:06 UTC*
  - 3d2dabc7b7 Merge pull request #37930 from cachedout/fix_comp
  - 670e83200b Remove dictionary comprehension in netusers
- **PR** #37923: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2016-11-28 19:55:03 UTC*
  - 0f8b187d15 Merge pull request #37923 from rallytime/merge-2016.11
  - da7f5518eb Don't let 2016.3 doc config changes overwrite the 2016.11 changes
  - dfedd1185a Merge branch '2016.3' into '2016.11'
    - c35ba1f390 Merge pull request #37916 from rallytime/doc-update-2016.3
      - bd40592289 [2016.3] Update version numbers in doc config for 2016.11.0 release
    - e13a2488c8 Merge pull request #37785 from Cloudtek/ddns-respect-trailing-dot
      - 262e3b3697 respect trailing dot in ddns name parameter
    - c03b389422 Merge pull request #37895 from fj40crawler/fix-augeas-return-for-test
      - ddc238df36 Fixed augeas_test.py to match True v.s. None for test_change_in_test_mode

      - ef75c459c0 Merge branch '2016.3' of github.com:saltstack/salt into fix-augeas-return-for-test
      - b0fe0cd256 Change return value for salt/states/augeas.py to be True instead of None for cases where salt is run with test=True. Fixes #37870
    - fdbc31e8d8 Merge pull request #37907 from Talkless/patch-2
      - 072a319490 Fix server trust in test run of svn.latest
    - f39fdf443f Merge pull request #37896 from toanju/2016.3
      - c95304188e rh networking: add missing values
    - ea935c5a91 Merge pull request #37886 from bdrung/fix-typos
      - 9a51ba5c5b Fix various spelling mistakes
    - 371b0a86d9 Merge pull request #37736 from dhaines/issue-37732
      - 7ef590a505 Update selinux.py
      - 516a67e6a3 fix indexing error
      - 4e49c1e991 fix typo
      - b16f2d8400 handle semodule version >=2.4 (#37732) and fix typo
    - 87aeb66fbf Merge pull request #37797 from clan/extfs
      - acf0f960ef check count of columns after split
    - f7c7109152 Merge pull request #37762 from twangboy/fix_chocolatey_state
      - 9696b6dfa5 Use keyword args instead of relying on ordering
      - 398eaa074d Add pre_versions to the available arguments
    - 56baa92d55 Merge pull request #37866 from meaksh/2016.3-bp-37149-36938-36784
      - 9d8d578109 Fix pkg.latest_version when latest already installed
      - ffca0d491c - acl.delfacl: fix position of -X option to setfacl
      - 3dfed6b841 Adjust linux_acl unit test argument ordering
      - f185ecdde1 core.py: quote style fixed
      - 8404d13424 Setting up OS grains for SLES Expanded Support (SUSE's Red Hat compatible platform)
    - d0cc7f0d56 Merge pull request #37863 from rallytime/bp-36893
      - 4c70534991 Add versionadded to reauth option in dockerng module
      - 5ca2c388c2 added documentation for the new reuth option in docker registry configuration
      - 5b0c11ab47 add option to force a reauth for a docker registry
    - b17a118e72 add multiline encryption documentation to nacl (#37847)
  - **PR** #37927: (thatch45) Add a release notes reference to the docker-sls tutorial
  - **PR** #37917: (rallytime) [2016.11] Update version numbers in doc config for 2016.11.0 release
- **PR** #37890: (bbinet) Fix support for extra_mods='six' to add six module to a thin.tgz tarball @ *2016-11-28 13:53:06 UTC*
  - ee00592995 Merge pull request #37890 from bbinet/fix-genthin-six

  - 7fceaa3476 Fix support for extra_mods='six' to add six module to a thin.tgz tarball
- **ISSUE** #37713: (aboe76) masterless minion can't call pillar.item from pillar stack (development branch) (refs: #37843)
  - **PR** #37843: (terminalmage) Don't skip pillar compilation when master_type=='disable'
  - **PR** #32521: (adelcast) Fix salt-call on standalone minion case (refs: #37843)
- **ISSUE** #37449: (thatch45) Allow TLS connections in the Tornado TCP transport (refs: #37776, #37859)
  - **PR** #37849: (skizunov) Eliminate warning when 'ssl' not set
  - **PR** #37776: (DmitryKuzmenko) Full TLS/SSL options support as provided by Tornado TCPServer. (refs: #37849)
- **ISSUE** #37449: (thatch45) Allow TLS connections in the Tornado TCP transport (refs: #37776, #37859)
  - **PR** #37859: (DmitryKuzmenko) TLS example config
  - **PR** #37841: (terminalmage) Clarify the master_type docs
  - **PR** #37831: (skizunov) PY3: Fix exception when handling connect exception in TCP transport
- **PR** #37829: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2016-11-22 15:26:00 UTC*
  - dd81d2fa67 Merge pull request #37829 from rallytime/merge-2016.11
  - 3d6d32edc5 Merge branch '2016.3' into '2016.11'
  - aa3748744c add missing chloginclass (#37827)
  - 0e74bad284 Update branch refs to more relevant branch (#37826)
  - 6a9b49c782 Add "names" option to file state docs: point users to highstate doc examples (#37823)
  - aaf587de63 Clarify keystone.user_present password state docs with default behavior (#37821)
  - c300863159 Add some dependency documentation to libvirt docs (#37820)
  - 485270f74e Merge pull request #37772 from bdrung/openssl1.1
    - 819c9658ed Support initializing OpenSSL 1.1
  - 4910912ffa Update orchestrate runner file.copy doc example (#37817)
  - c5d3d8b66a Merge pull request #37816 from rallytime/bp-32157
    - d9c297119e Add quotes to cron doc
  - 97e6b6aabe Merge pull request #37812 from rallytime/bp-37790
    - ca3b6e7874 Update proxmox.rst with more options and LXC
  - 27703c54bc Merge pull request #37811 from rallytime/bp-37789
    - ba3fef48e1 fix comment
    - a021f76a9b issue: 37751 Add documentation for option privileged
  - adac9d7c0c Merge pull request #37810 from rallytime/bp-37775
    - 2bed91437b Document *python* argument in *salt.states.virtualenv_mod*
  - **PR** #37794: (sjorge) network.routes should not raise exception if no interface
- **PR** #37815: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2016-11-21 20:22:49 UTC*
  - 628c4a3d27 Merge pull request #37815 from rallytime/merge-2016.11
  - c6b5fd3715 Merge branch '2016.3' into '2016.11'

    - 7de784411d Add nodegroup check to ckminions (#37763)
    - d674369efc Fix ip/port issue with salt-call (#37766)

- **ISSUE** #37449: (thatch45) Allow TLS connections in the Tornado TCP transport (refs: #37776, #37859)
- **PR** #37776: (DmitryKuzmenko) Full TLS/SSL options support as provided by Tornado TCPServer. (refs: #37849) @ *2016-11-21 20:11:52 UTC*
  - 0b30b93dbb Merge pull request #37776 from DSRCorporation/features/37449_tls
  - 6857b9b8b1 Documented new TLS/SSL settings.
  - e42898f2e3 Full TLS/SSL options support as provided by Tornado TCPServer.
- **PR** #37773: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2016-11-18 19:18:42 UTC*
  - 3835f91d99 Merge pull request #37773 from rallytime/merge-2016.11
  - c859fc9ec1 Merge branch '2016.3' into '2016.11'
  - c62ff6b023 Add thorium path to syspaths (#37767)
  - bff949f4e9 Merge pull request #37760 from hu-dabao/fix_cb_returner
    - de372f277e 1. returner no need to check whether the jid exists for external job cache setup 2. add full_ret to return doc so that the document will be informative 3. make ttl as a config attribute because salt-minion does not have keep_jobs attribute 4. add password into config attribute 5. update the documents accordingly
  - 1f976ac212 Merge pull request #37738 from terminalmage/issue36629
    - da46678c51 Allow pillar.get to retrieve fresh pillar data when saltenv passed
  - 7aee7fc63c Switch default filter tag for ONE resources from user only to all resources (#37745)
- **PR** #37764: (mirceaulinic) Doc fixes and *replace* feature @ *2016-11-18 03:15:31 UTC*
  - 6f0f70c9a3 Merge pull request #37764 from cloudflare/NET-UPDATE
  - c3f0202fdd Replace feature and doc fixes

28.2.32 Salt 2016.11.10 Release Notes

Version 2016.11.10 is a security release for *2016.11.0*.

Changes for v2016.11.9..v2016.11.10

Security Fix

CVE-2018-15751 Remote command execution and incorrect access control when using salt-api.

CVE-2018-15750 Directory traversal vulnerability when using salt-api. Allows an attacker to determine what files exist on a server when querying /run or /events.

Credit and thanks for discovery and responsible disclosure: nullbr4in, xcuter, koredge, loupos, blackcon, Naver Business Platform

28.2.33 Salt 2016.11.2 Release Notes

Version 2016.11.2 is a bugfix release for *2016.11.0*.

Statistics

- Total Merges: **157**
- Total Issue References: **34**
- Total PR References: **116**
- Contributors: **45** (Ch3LL, Cybolic, DmitryKuzmenko, UtahDave, Vaelatern, alex-zel, alxwr, amendlik, anlutro, aosagie, basdusee, bbinet, benediktwerner, cachedout, clinta, cro, dereckson, disaster123, ewapptus, ezh, folti, gmacon, gqgunhed, gtmanfred, kkoppel, lorengordon, martintamare, mcalmer, meaksh, mirceaulinic, mostafahussein, mvdwalle, rallytime, rbjorklin, scthi, sjorge, techhat, terminalmage, tsaridas, twangboy, vutny, wolfpackmars2, yhekma, yopito, yue9944882)

Security Fixes

CVE-2017-5192 local_batch client external authentication not respected

The `LocalClient.cmd_batch()` method client does not accept `external_auth` credentials and so access to it from salt-api has been removed for now. This vulnerability allows code execution for already-authenticated users and is only in effect when running salt-api as the `root` user.

CVE-2017-5200 Salt-api allows arbitrary command execution on a salt-master via Salt's ssh_client

Users of Salt-API and salt-ssh could execute a command on the salt master via a hole when both systems were enabled.

We recommend everyone upgrade to 2016.11.2 as soon as possible.

Changelog for v2016.11.1..v2016.11.2

Generated at: 2018-05-27 19:28:11 UTC

- **PR** #38859: (alxwr) fix parsing of sockstat -4 @ *2017-01-23 16:47:22 UTC*
  - ec59ae67c8 Merge pull request #38859 from alxwr/2016.11
  - 30fe5641c7 fix parsing of sockstat -4
- **PR** #38850: (techhat) Strip .p from cache file names @ *2017-01-23 16:28:46 UTC*
  - 5fe6db6201 Merge pull request #38850 from techhat/stripcache
  - 109cb62e76 Remove .p from test
  - 534aa3f527 Strip .p from cache file names
  - **PR** #38848: (Ch3LL) add 2016.11.2 changelog to release notes
- **PR** #38819: (twangboy) Remove *Users* from c:\salt [DO NOT MERGE FORWARD] @ *2017-01-20 20:17:35 UTC*
  - 4913c4f90c Merge pull request #38819 from twangboy/salt_perms_2016.11
  - eb04ed7eef Remove *User* from c:salt
- **PR** #38815: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-01-20 18:53:01 UTC*
  - a275b9714e Merge pull request #38815 from rallytime/merge-2016.11
  - ce6d1b103d Make sure we're using the opts dict mocking in parsers_test
  - 315b2c8712 Merge branch '2016.3' into '2016.11'
    - d14f0c64eb Merge pull request #38812 from rallytime/pyobjects-test

      - f3e84c1ab7 Update pyobjects test to be a list
    - 50f03f8057 Merge pull request #38813 from gtmanfred/2016.3
      - ce3472cec2 catch SIGPIPE in vmware connection
    - 23b8b47258 Merge pull request #38809 from twangboy/fix_hostname_2016.3
      - d57a51f9f9 Fix tests for get_hostname
      - 7ca3fd7484 Fix get_hostname to handle longer computer names
    - 1033bbdde8 Merge pull request #38808 from vutny/fix-38388
      - 9bd203ffcc Fix #38388
    - f3ae3cd5c8 Merge pull request #38668 from terminalmage/issue38604
      - 0ea97cdad9 Merge pull request #10 from cachedout/pr-38668
      - db81afc035 Munge retcode into return data for batching
      - a642a995dc Return the ret data from batch execution instead of raw data
    - c6a19a9e5a Merge pull request #38789 from rallytime/fix-38622
      - af41fe0c6e Update some saltenv refs to environment in salt.modules.state docs
    - e0bf700020 Merge pull request #38790 from cachedout/fix_pyobjects_test_typo
      - a66afb5f0f Fix typo in pyobjects test
    - 6e9785edea Merge pull request #38792 from rallytime/fix-38629
      - 1e125e2844 Update pillar tutorial lanuage regarding pillar_opts settings

- **PR** #38832: (terminalmage) archive.extracted: Identify symlinks when checking for incorrect types @ *2017-01-20 18:36:15 UTC*
  - efe1bf10e8 Merge pull request #38832 from terminalmage/issue38711
  - d10c068e25 Update archive state unit tests to reflect symlinks in archive.list
  - d6adfb6d12 Identify symlinks when looking for incorrect types
  - 09b9e95f7c archive.list: organize symlinks separately from files in verbose mode
  - e6483f096d Support removing symlinks in salt.utils.rm_rf
- **PR** #38726: (twangboy) Add VC Redist 2008 SP1 MFC to installer @ *2017-01-19 19:13:42 UTC*
  - 10a3d8b8dd Merge pull request #38726 from twangboy/vcredist
  - f00a65355d change extensions .ext to .exe
  - 98c40e278c Add VC Redist 2008 SP1 MFC to installer
- **PR** #38810: (UtahDave) Fix beacon doc @ *2017-01-18 21:37:21 UTC*
  - d5f2d92a4e Merge pull request #38810 from UtahDave/fix_beacon_doc_zd1035
  - dbe9edb806 fix reactor example.
- **PR** #38811: (techhat) Show a lot less data when requesting a VM @ *2017-01-18 21:08:03 UTC*
  - 88faf08a71 Merge pull request #38811 from techhat/sanvm
  - 47c19325cf Show a lot less data when requesting a VM
  - **PR** #38807: (Ch3LL) refine the os detection in archive test

- **PR** #38799: (aosagie) Parse ansible dynamic inventory output correctly @ *2017-01-18 15:32:47 UTC*
  - e3ca6881c8 Merge pull request #38799 from aosagie/fix-ansible-dynamic-roster
  - 26d6f699a7 Parse ansible dynamic inventory output correctly
- **PR** #38787: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-01-18 08:39:08 UTC*
  - 76df6a43f3 Merge pull request #38787 from rallytime/merge-2016.11
  - 2aad54c49f Merge branch '2016.3' into '2016.11'
    - 3417adc617 Merge pull request #38796 from saltstack/revert-38707-root_dir_fix-gh
      - cb080f3bbe Revert "Fixed prepending of root_dir override to the other paths"
  - 64d866f7ab Merge branch '2016.3' into '2016.11'
  - bab3479a3c Merge pull request #38585 from rallytime/follow-up-38527
    - 05587201b6 Pylint fix: add line at end of file
    - fa01367599 Keep a copy of the DEFAULT_API_OPTS and restore them after the test run
    - 2ad07634d9 Test clean up
    - fd2ee7db30 Add some simple unit tests for salt.config.api_config function
    - 3d2fefc83b Make sure the pidfile and log_file values are overriden by api opts
    - 1f6b540e46 Make sure the pidfile and log_file values are overriden by api opts
    - 04d307f917 salt-api no longer forces the default timeout
  - 0fb6bb7b77 Merge pull request #38707 from alexbleotu/root_dir_fix-gh
    - 0bac8c8be3 Fixed prepending of root_dir override to the other paths
  - 96c9dc10f7 Merge pull request #38774 from vutny/dev-test-docs
    - 4620dc4afa DOCS: add C++ compiler installation on RHEL required for bundled 0mq
  - aedfbb7a43 Merge pull request #38749 from vutny/pkg-build-better-exception-msg
    - 53f2be5b21 pkg build modules throw better exception message if keyid wasn't found
- **PR** #38660: (techhat) Don't force salt.cache to use cachedir from opts @ *2017-01-17 18:38:35 UTC*
  - 4e6146f65f Merge pull request #38660 from techhat/cachedir
  - be55b57abf One last fix
  - fc24b24998 Add correct function name
  - 9bbecf7960 Typo fix
  - 436ba28f08 Change getlist back to list (using _list)
  - ff734fe93b Default to CACHE_DIR in syspaths
  - 380abd3744 Add cachedir args to tests
  - deb08c0587 Not every module will need cachedir
  - 4489f7cac0 Don't force salt.cache to use cachedir from opts
- **ISSUE** #37948: (djacobs2016) ssh_known_hosts.present is failing when checking key/host (refs: #37982)
- **ISSUE** #33932: (folti) ssh_known_hosts.present: hashing global known hosts file makes it readable by root only (refs: #33933)

- **PR** #38667: (rallytime) Back-port #37982 to 2016.11 @ *2017-01-17 15:42:13 UTC*
  - **PR** #37982: (wolfpackmars2) Update ssh.py (refs: #38667)
  - **PR** #33933: (folti) ssh: keep original permissions, when hashing known_hosts (refs: #38667)
  - 89dc86e2bc Merge pull request #38667 from rallytime/bp-37982
  - be91e46a93 Update ssh.py
- **PR** #38759: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-01-17 15:22:01 UTC*
  - 751e14c523 Merge pull request #38759 from rallytime/merge-2016.11
  - 30e8a66fb0 Merge branch '2016.3' into '2016.11'
    - 8466b34e82 Merge pull request #38743 from rallytime/merge-2016.3
      - d24776f5e9 Merge branch '2015.8' into '2016.3'
      - 6869621ed1 Merge pull request #38731 from rallytime/merge-2015.8
      - 9eb191b6ac Pylint fix
      - b910499dbe Various follow up fixes
      - e8309a6bbf Add release notes for 2015.8.13
      - f881f366b7 Merge pull request #20 from rallytime/2015.8.12_follow_up-batch-tests
      - 34282322c0 Clean up tests and docs for batch execution
      - c80b20b957 Merge pull request #19 from whiteinge/batchclient
      - 3d8f3d18f6 Remove batch execution from NetapiClient and Saltnado
      - 97b0f64923 Lintfix
      - d1516664f7 Add explanation comment
      - 62f2c87080 Add docstring
      - 9b0a786aeb Explain what it is about and how to configure that
      - 5ea3579e10 Pick up a specified roster file from the configured locations
      - 3a8614c5df Disable custom rosters in API
      - c0e5a1171d Add roster disable flag
      - e9c59e9b8f Merge pull request #38602 from terminalmage/fix-boto-test
      - 3424a108ac Fix failing unit.states.boto_vpc_test.BotoVpcRouteTableTestCase.test_present_with_routes
    - a642cdef79 Merge pull request #38723 from rallytime/fix-38674
      - 706c885f55 Remove "event_publisher_pub_hwm" and "salt_event_pub_hwm" from config/__init__.py
    - fc545af10b Merge pull request #38669 from rallytime/update-bootstrap-script
      - 78ba76e34c Update bootstrap script verstion to latest release
    - 50d417f267 Merge pull request #38693 from twangboy/update_jinja
      - e0c7e5549b Update jinja2 to 2.9.4
    - f4233bb18d Merge pull request #38739 from vutny/fix-runtests-doc
      - b872bb63f6 DOCS: correct examples of running test suite

    - 51d4707071 DOCS: add links to File State Backups page where necessary (#38735)
    - 6d3717b9ee Proofread jinja_to_execution_module tutorial (#38720)
- **ISSUE** #38775: (charburns) Error using napalm netusers (refs: #38778)
- **PR** #38778: (mirceaulinic) Fix "Error using napalm netusers" @ *2017-01-17 15:20:27 UTC*
  - bb6291d93a Merge pull request #38778 from cloudflare/fix-38775
  - b3388f7162 Fix #38775
- **ISSUE** #38528: (MorphBonehunter) x509 make permissions configurable (refs: #38664)
- **ISSUE** #38081: (haraldrudell) x509 state or module cannot generate password protected private keys (refs: #38664)
- **PR** #38664: (clinta) X509 Improvements. Expose setting permissions, encrypted private keys, and combined key and cert management in one state @ *2017-01-17 02:20:18 UTC*
  - 6663107021 Merge pull request #38664 from clinta/x509-passphrase2
  - 77c78723fe pep8
  - a2b20ee518 No mutable default args, remove unneeded import
  - b48b85cc70 bug fixes
  - f62393b864 pep8
  - c8613243a1 change documentation
  - 9a0abde9ac expose passphrase functionality to state
  - e47a93d496 add passphrase to execution module
  - a4d6598f1e preserve detailed change reports
  - d0ad251778 combine private key and cert management
  - 3d1474d911 cross call file.managed to get permissions options
- **PR** #38682: (mirceaulinic) [2016.11.2/napalm] Better error message when NotImplementedError raised @ *2017-01-15 18:34:25 UTC*
  - bf6d74c98e Merge pull request #38682 from cloudflare/NotImplementedError-MSG
  - f847639dee Better error message when NotImplementedError raised
- **ISSUE** #37996: (stefan-as) influxdb_user.present does not pass client_args (refs: #38695)
- **PR** #38695: (rallytime) Pass in client_args when calling influxdb execution module funcs @ *2017-01-15 18:33:48 UTC*
  - df12e49d80 Merge pull request #38695 from rallytime/fix-37996
  - 05b0975888 Pass in client_args when calling influxdb execution module funcs
- **ISSUE** #38521: (vladvasiliu) State cloud.present on AWS: TypeError: 'NoneType' object is not iterable (refs: #38651)
- **ISSUE** #37981: (tazaki) Salt-cloud ec2 vpc securitygroupid always returning default (refs: #38183)
- **PR** #38651: (rallytime) Don't lose the set reference for ec2 securitygroup ids @ *2017-01-15 18:06:25 UTC*
  - **PR** #38183: (cro) Fix bad set operations when setting up securitygroups in AWS. (refs: #38651)
  - 834e5469fc Merge pull request #38651 from rallytime/fix-38521
  - 830c03cec6 Don't lose the set reference for ec2 securitygroup ids

- **ISSUE** #38216: (pgrishin) salt-run: can't get cache.grains (refs: #38659)
- **PR** #38659: (techhat) Turn None into an empty string (for minion matching) @ *2017-01-15 18:02:03 UTC*
  - 8b38cfea8d Merge pull request #38659 from techhat/issue38216
  - 4073c91584 Turn None into an empty string (for minion matching)
- **PR** #38703: (yhekma) The *test* option is only valid for the minion, not the master @ *2017-01-15 17:56:22 UTC*
  - 0ad5d22ad4 Merge pull request #38703 from yhekma/docfix
  - 57df3bf148 The *test* option is only valid for the minion, not the master
- **PR** #38718: (terminalmage) Fix for dynamic git_pillar when pillarenv is used @ *2017-01-15 14:37:30 UTC*
  - 8c1222e7db Merge pull request #38718 from terminalmage/zd909
  - 12bbea5a24 Fix for dynamic git_pillar when pillarenv is used
- **ISSUE** #38677: (yhekma) consul cache backend broken (refs: #38676)
- **PR** #38676: (yhekma) Removed overloading of list() @ *2017-01-15 05:42:13 UTC*
  - aae8b54860 Merge pull request #38676 from yhekma/2016.11
  - 3237d23e1c Localfs should also be changed of course
  - 9d9de67219 We do not want to overload the list() type because if we do, we turn this function into a recursive one, which results in an exception because set() cannot be concatenated with str ('/')
- **ISSUE** #38684: (rukender) 2016.11.1 :[ERROR][11182] Failed to import beacons avahi_announce (refs: #38713)
- **PR** #38713: (rallytime) Add NameError to exception in avahi_announce beacon @ *2017-01-15 05:33:04 UTC*
  - c246ab41c5 Merge pull request #38713 from rallytime/fix-38684
  - db60bed24c Add NameError to exception in avahi_announce beacon
- **PR** #38729: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-01-13 23:15:33 UTC*
  - 6c14774c04 Merge pull request #38729 from rallytime/merge-2016.11
  - 4e1e45d640 Merge branch '2016.3' into '2016.11'
  - 7b850d472d Merge pull request #38647 from gtmanfred/nova
    - 5be9b60851 add documentation about using keystoneauth for v3
    - 7b657ca4ae add the ability to use keystone v2 and v3
    - 5646ae1b34 add ability to use keystoneauth to authenitcate in nova driver
  - 383768d838 Merge pull request #38650 from rallytime/remove-ubuntu-ppa-docs
    - 30429b2e44 Remove the installation instructions for out-of-date community ppa
  - 7d9f56e3b5 Merge pull request #38657 from DSRCorporation/bugs/38087_syndic_event_format_fix
    - 594c33f396 Publish the 'data' field content for Syndic evets
  - 83987511fd Merge pull request #38649 from Ch3LL/test_apply_template
    - 47f8b68e0b fix unit.modules.file_test
- **ISSUE** #38631: (doitian) In Orchestration, kwargs are not passed to state.sls in masterless mode (refs: #38635)
- **PR** #38635: (lorengordon) Sends pass-through params to state module @ *2017-01-10 20:01:59 UTC*
  - cfd82d1631 Merge pull request #38635 from lorengordon/issue-38631

  - 14666138b9 Sends pass-through params to state module
- **PR** #38640: (mirceaulinic) Import napalm_base instead of napalm @ *2017-01-10 19:58:01 UTC*
  - 017094a207 Merge pull request #38640 from cloudflare/NAPALM-IMPORTS
  - 8f13f63880 Import napalm_base instead of napalm
- **PR** #38661: (techhat) Add sane cache defaults for minion and cloud @ *2017-01-10 19:55:15 UTC*
  - 79663132dd Merge pull request #38661 from techhat/sanedefault
  - aee40648ec Add a sane cache default for cloud
  - c9e01a36e7 Add a sane cache default for minions
- **PR** #38645: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-01-10 19:54:06 UTC*
  - b0ed91ce2d Merge pull request #38645 from rallytime/merge-2016.11
  - 7a668e9749 Merge branch '2016.3' into '2016.11'
  - 74ddc71be3 Merge pull request #38626 from saltstack/revert-37358-2016.3.3_issue37355
    - e912ac99c2 Revert "Fix/workaround for issue #37355"
  - 5e58b32934 Merge pull request #37358 from Firewire2002/2016.3.3_issue37355
    - 910da18bfd fixed typo
    - 4fbc5ddd06 fixed wrong renamed variable and spaces
    - 92366e646c issue #37355
    - 7dc87ab7b8 issue #37355
    - 2878180405 issue #37355
  - 6c2fe615aa Merge pull request #35390 from alexandr-orlov/2016.3
    - cd5ae17e8d fxd missed proper grains dictionary
  - 2579cfa42d Merge pull request #38618 from rallytime/bp-38579
    - 2052ecee2c Add copy import
    - 2c8845aaa0 add test for pillar.get() + default value
    - c2f98d2f04 ticket 38558: add unit test, deepcopy() only if necessary
    - 30ae0a1958 added deepcopy of default if merge=True
- **PR** #38627: (cachedout) Pr 38476 @ *2017-01-06 22:05:45 UTC*
  - **PR** #38476: (amendlik) Key fingerprints (refs: #38627)
  - d67f6937d7 Merge pull request #38627 from cachedout/pr-38476
  - 2a423ffedd Add changes to raetkey
  - 55ad9d6c6c Add hash_type argument to MultiKeyCLI.finger_all function
  - c8681269a4 Add hash_type argument to key module fingerprint functions
  - d0f4c300b7 Add hash_type argument to wheel fingerprint functions
  - e558ddcb18 Add finger_master function to wheel.key module
- **ISSUE** #38595: (yue9944882) Redis ext job cache occurred error (refs: #38610)

- **PR** #38610: (yue9944882) Fix #38595 - Unexpected error log from redis retuner in master's log @ *2017-01-06 21:47:21 UTC*
  - b13cd1370f Merge pull request #38610 from yue9944882/2016.11
  - 54325cf293 Fix #38595 - Unexpected error log from redis retuner in master's log
- **ISSUE** #36148: (alex-zel) Eauth error with openLDAP groups (refs: #38406)
- **PR** #38406: (alex-zel) Fix eauth error with openLDAP/389 directory server groups @ *2017-01-06 21:40:30 UTC*
  - 179d385003 Merge pull request #38406 from alex-zel/fix-eauth-groups-permissions
  - 6b9e9d8f89 Fix eauth error with openLDAP/389 directory server groups
- **PR** #38619: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-01-06 17:51:19 UTC*
  - 82e9b3d1a1 Merge pull request #38619 from rallytime/merge-2016.11
  - 0efb2d844e Merge branch '2016.3' into '2016.11'
    - da676cebd6 Merge pull request #38601 from terminalmage/pillar-get
      - 8613d7254d pillar.get: Raise exception when merge=True and default is not a dict
    - 224fc7712a Merge pull request #38600 from terminalmage/issue38459-2016.3
      - 8a45b13e76 Avoid errors when sudo_user is set
    - a376970f88 Merge pull request #38589 from tobithiel/fix_rvm_rbenv_warning
      - 9ec470b4a5 State Gem: fix incorrect warning about missing rvm/rbenv
    - 02e6a78254 Merge pull request #38567 from pass-by-value/pgjsonb_queue_changes_2016.3
      - 67879ebe65 Create queue if one doesn't exist
    - 0889cbdb31 Merge pull request #38587 from rallytime/fix-37498
      - 2a5880966f Change daemontools __virtualname__ from service to daemontools
- **PR** #38612: (sjorge) network.ifacestartswith throws exception on Solaris-like platforms @ *2017-01-06 17:20:32 UTC*
  - f64e003a69 Merge pull request #38612 from sjorge/2016.11-solaris-ifacestartswith
  - 26fae54f5b network.ifacestartswith throws exception on Solaris-like platforms
- **ISSUE** #37027: (sjorge) Solaris FQDN/UQDN and documentation/consistancy (refs: #38615)
- **PR** #38615: (sjorge) add note related to issue #37027 @ *2017-01-06 16:38:34 UTC*
  - 5820ceee16 Merge pull request #38615 from sjorge/2016.11-solarisdocs
  - fbdd32f46b add note related to issue #37027
- **PR** #38598: (terminalmage) Avoid errors when sudo_user is set (refs: #38600, #38599) @ *2017-01-05 23:16:22 UTC*
  - a27fdb46a7 Merge pull request #38598 from terminalmage/issue38459
  - b37f7ffa38 Avoid errors when sudo_user is set
- **PR** #38599: (terminalmage) archive.extracted: Prevent traceback when state.single cannot be run @ *2017-01-05 23:16:11 UTC*
  - **PR** #38598: (terminalmage) Avoid errors when sudo_user is set (refs: #38600, #38599)
  - d6b7019df6 Merge pull request #38599 from terminalmage/archive-results-handling

  - 9aceb8186d archive.extracted: Prevent traceback when state.single cannot be run
- **ISSUE** #38517: (basdusee) Slack.py engine 100% CPU load due to missing time.sleep(1) (refs: #38520)
- **PR** #38520: (basdusee) Fix issue #38517, added time.sleep(1) at line 227 in slack.py @ *2017-01-05 20:35:08 UTC*
  - d486b42ceb Merge pull request #38520 from basdusee/fix-issue-38517
  - e3a883c915 Small fix on the fix regarding indentation
  - 8adeae6f81 Fix issue #38517, added time.sleep(1) at line 227 in slack.py engine.
- **ISSUE** #38485: (wasabi222) bgp.config not working (refs: #38499)
- **PR** #38577: (mirceaulinic) Fix function headers as per #38499 @ *2017-01-05 18:41:33 UTC*
  - **PR** #38499: (mirceaulinic) Fix #38485 (refs: #38577)
  - 0706cde626 Merge pull request #38577 from cloudflare/PREP-2016.11.2
  - 62bee3c793 Fix function headers as per #38499
- **PR** #38578: (mirceaulinic) [2016.11] Port 5123f11 from develop into 2016.11.2 @ *2017-01-05 18:11:12 UTC*
  - 55d1747792 Merge pull request #38578 from cloudflare/PORT-5123f1
  - dea7866d57 Update net.load_template doc: 2016.11.2
- **ISSUE** #38462: (g-shockfx) Can't add beacon memusage on Windows (refs: #38584)
- **PR** #38584: (rallytime) Allow memusage beacon to load on Windows @ *2017-01-05 18:08:30 UTC*
  - be69bafe6e Merge pull request #38584 from rallytime/fix-38462
  - 1fe945df5e Allow memusage beacon to load on Windows
- **PR** #38570: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 (refs: #38585) @ *2017-01-05 14:28:38 UTC*
  - 14b643fd48 Merge pull request #38570 from rallytime/merge-2016.11
  - 30f14d15df Merge branch '2016.3' into '2016.11'
  - 7b74436d13 Merge pull request #38562 from rallytime/arch-install-docs
    - 8b1897ace9 Update arch installation docs with correct package name
  - 01860702cb Merge pull request #38560 from Ch3LL/fix_api_log
    - 1b45e9670b fix api logfile
  - 0056620a53 Merge pull request #38531 from rallytime/bp-33601
    - c36cb39825 remove the unnecessary double trigger
    - 38414493bf fix spacing lint error
    - 8c1defc710 Remove uncessary type from alias commands. Deduplicate alias handling to autodetect function selection. Add error reporting to slack connectivty problems. Cleanup slack's unicode conversion
    - c2f23bc45e Fix slack engine to run on python2.6
  - 50242c7f17 Merge pull request #38541 from techhat/issue38187
    - eae3a435dd Strip user:pass from cached URLs
  - 325dc56e59 Merge pull request #38554 from multani/fix/30454
    - 2e7f743371 yaml: support unicode serialization/deserialization

    - df76113c5c jinja: test the "yaml" filter with ordered dicts
    - f7712d417f Revert "Add yaml_safe filter"
  - 4ddbc2ecaa add note about pyVmomi locale workaround (#38536)
  - 1c951d152b fix gce image bug (#38542)
- **PR** #38509: (mostafahussein) Stop request from being processed if bad ip @ *2017-01-04 20:05:44 UTC*
  - 9a1550d336 Merge pull request #38509 from mostafahussein/2016.11
  - 8847289c3e remove commented code
  - 420817a963 Stop request from being processed if bad ip
- **ISSUE** #38518: (kkoppel) slack_notify.call_hook returns tracebacks (refs: #38522)
- **PR** #38522: (kkoppel) Fix usage of salt.utils.http.query in slack_notify.call_hook @ *2017-01-04 20:04:57 UTC*
  - bc07d420e9 Merge pull request #38522 from kkoppel/fix-issue-38518
  - ff1e7f0c71 Fix usage of salt.utils.http.query in slack_notify.call_hook
- **ISSUE** #38524: (rbjorklin) salt-api seems to ignore rest_timeout since 2016.11.0 (refs: #38585, #38527)
- **PR** #38527: (rbjorklin) salt-api no longer forces the default timeout (refs: #38585) @ *2017-01-04 17:10:15 UTC*
  - 42fef270ee Merge pull request #38527 from rbjorklin/api-timeout-fix
  - 0202f68820 salt-api no longer forces the default timeout
- **PR** #38529: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-01-04 17:06:57 UTC*
  - 1895eb7533 Merge pull request #38529 from rallytime/merge-2016.11
  - 85f470207c Merge branch '2016.3' into '2016.11'
    - ec60f9c721 Merge pull request #38487 from gtmanfred/2016.3
      - 048b9f6b9d add test
      - c480c11528 allow spaces in cron env
      - c529ec8c34 allow crons to have multiple spaces
    - c5ba11b5e0 Merge pull request #38491 from gtmanfred/timing
      - 79368c7528 Use UTC for timing in case timezone changes
    - 86f0aa0bb3 Merge pull request #38503 from jinm/issue_38472_jinm
      - 0cd9df299f Hash type fallback for file management
    - ed2ba4bd1b Merge pull request #38457 from bshelton229/git-latest-head-bug
      - 558e7a771a Stops git.latest checking for local changes in a bare repo
    - 36e21b22cb Merge pull request #38385 from dragon788/2016.3-double-dash
      - 86c4b56f47 Newline for lint compat
      - 9d9b686057 Address review comments, consistency of quotes
      - df9bd5e7f9 Use unambigous long names with double dashes
    - 59f2560d88 Merge pull request #38474 from cachedout/key_loop
      - de504538e1 Allow an existing ioloop to be passed to salt-key
    - 3d0c752acd Merge pull request #38467 from gtmanfred/2016.3

        - 7b7c6b3878 file.line fail with mode=delete
    - 940025d5c4 Merge pull request #38434 from slinn0/issue_38433_fixes
        - 22af87a3fc Fixes for https://github.com/saltstack/salt/issues/38433
    - e5eb51255b Update deprecation notices to the correct version (#38421)
    - 9ce53318df file.managed: Fix failure when filename contains unicode chars (#38415)
    - 2cdb59d055 Merge pull request #38419 from Ch3LL/fix_doc_scsi
        - 234043b8bb fix scsci docs example

- **PR** #38539: (twangboy) Fix DSC LCM Config int checks @ *2017-01-04 16:56:27 UTC*
  - ec4f118ca2 Merge pull request #38539 from twangboy/dsc_int_checks
  - 5657fd1956 Add repr flag for str
  - aea4219502 Fix DSC LCM Config int checks
- **PR** #38549: (meaksh) Adding multiple SUBVOLUME support and some fixes to the Snapper module @ *2017-01-04 15:32:30 UTC*
  - 53449c89a5 Merge pull request #38549 from meaksh/2016.11-snapper-multiple-subvolumen-support
  - ef26e93bb7 Some fixes and pylint
  - 1e6ba45db4 Fixes pre/post snapshot order to get the inverse status
  - 68d5475c1f Fixing Snapper unit tests for SUBVOLUME support
  - e9919a913f Removing posible double '/' from the file paths
  - 8b4f87f226 Updating and fixing the documentation
  - edea45272a Raises "CommandExecutionError" if snapper command fails
  - 3841e1143b Only include diff in the state response if *include_diff* is True
  - 7803e7716c Adds multiple SUBVOLUME support to the Snapper module
  - **PR** #38545: (rallytime) Move boto_vpc.describe_route_table deprecation version to Oxygen
- **PR** #38471: (twangboy) Fix Problem with win_service module @ *2017-01-01 20:30:21 UTC*
  - 5e80104a70 Merge pull request #38471 from twangboy/fix_win_service
  - 810471b9cd Fix problem with some services getting access denied
- **ISSUE** #38485: (wasabi222) bgp.config not working (refs: #38499)
- **PR** #38499: (mirceaulinic) Fix #38485 (refs: #38577) @ *2017-01-01 17:42:15 UTC*
  - 0a09049a2d Merge pull request #38499 from cloudflare/FIX-38485
  - 18018139f3 Fix #38485
- **PR** #38501: (mvdwalle) Do not assume every object is a server @ *2017-01-01 17:37:57 UTC*
  - 13f0b809df Merge pull request #38501 from mvdwalle/fix-gogrid-list-password
  - bd7dee9a10 Do not assume every object is a server
- **PR** #38461: (anlutro) Improvements/fixes to kapacitor task change detection @ *2016-12-29 17:08:47 UTC*
  - aa0c843553 Merge pull request #38461 from alprs/fix-kapacitor_changes
  - 52721e97d6 clean up and fix tests

  - 8648775c2a if task is not defined, it's not up to date
  - c3ab954c6e improvements/fixes to kapacitor task change detection
- **PR** #38473: (twangboy) Change OSX/OS X to macOS where possible @ *2016-12-29 16:35:11 UTC*
  - 2c51eb9d16 Merge pull request #38473 from twangboy/osx_to_macos
  - e96bfe8fa2 Change OSX/OS X to macOS where possible
- **PR** #38412: (bbinet) Update PillarStack stack.py to latest upstream version @ *2016-12-28 19:28:40 UTC*
  - 2497fb547e Merge pull request #38412 from bbinet/pillarstack-updates
  - b66b4bd060 Fix lint violations in stack.py
  - 6a30fe6aeb Update PillarStack stack.py to latest upstream version
- **PR** #38456: (twangboy) Gate Windows Specific Salt Utils @ *2016-12-28 18:44:33 UTC*
  - 5395d3210a Merge pull request #38456 from twangboy/gate_win_utils
  - d34d110a84 Fix lint, fix boto module
  - c20111142f Gate Windows Utils
- **PR** #38428: (gqgunhed) fixed typo: lq command-line syntax @ *2016-12-27 15:42:02 UTC*
  - 7c7799162b Merge pull request #38428 from gqgunhed/fix_lq_typo
  - d79d682e8b fixed typo: lq command-line syntax
- **ISSUE** #38443: (lorengordon) 2016.11 breaks file.managed on Windows (refs: #38444)
- **ISSUE** #34101: (windoverwater) archive.extracted breakage due to 2016.3.0 upgrade from 2015.8.10 (refs: #37368)
- **PR** #38444: (lorengordon) Adds new import required for *extract_hash* @ *2016-12-27 15:37:20 UTC*
  - **PR** #37368: (terminalmage) Overhaul archive.extracted state (refs: #38444)
  - f5984d0f81 Merge pull request #38444 from lorengordon/issue-38443
  - b2925ad7b7 Adds new import required for *extract_hash*
- **ISSUE** #38071: (luochun-95) remote execute is very slow (refs: #38167)
- **PR** #38167: (cachedout) Kill pkg_resources for CLI tools [DO NOT MERGE] @ *2016-12-22 22:11:22 UTC*
  - 4c4f07ca4c Merge pull request #38167 from cachedout/no_pkg_resources
  - ec6901720a Remove debugging
  - f28e33b9b6 Remove from all but salt cli
  - bb3af72317 Remove from salt-call
  - c676846066 Kill pkg_resources for CLI tools
- **PR** #38417: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2016-12-22 19:00:44 UTC*
  - 2fc8c154af Merge pull request #38417 from rallytime/merge-2016.11
  - efb8a8ddf5 Merge branch '2016.3' into '2016.11'
  - 27253522c8 Improve pillar documentation (#38407)
  - 423b1fdddf Merge pull request #38398 from terminalmage/issue38372
    - c80dbaa914 Fix call to file.get_managed in cron.file state

  - 5a33d1e697 Fix http.query when result has no text (#38382)
  - b74b5c7d38 Merge pull request #38390 from meaksh/2016.3-fix-try-restart-for-autorestarting-on-SUSE-systems
    - de6ec05ec0 add try-restart to fix autorestarting on SUSE systems
  - 2c3a39760a Merge pull request #38221 from UtahDave/fix_default_returner
    - 385640765b remove a blank line to satisfy linter
    - 9c248aa14c validate return opt, remove default.
    - 8bb37f9fe7 specify allowed types and default for "returner"
    - 11863a4bfe add examples of default minion returners
    - e7c6012655 add support for default returners using *return*
- **PR** #38342: (scthi) Bugfix ext pillar nodegroups @ *2016-12-22 16:47:42 UTC*
  - bbc149c67f Merge pull request #38342 from scthi/bugfix-ext-pillar-nodegroups
  - dba315c4b6 ext-pillar nodegroups works for all minions now.
- **PR** #38403: (terminalmage) git_pillar: Document the transition from env to saltenv in the jinja context @ *2016-12-22 16:34:48 UTC*
  - 453476d982 Merge pull request #38403 from terminalmage/document-saltenv
  - 0a72e0f0be git_pillar: Document the transition from env to saltenv in the jinja context
- **ISSUE** #38253: (gmacon) There was no error installing package 'setuptools' although it does not show when calling 'pip.freeze'. (refs: #38354)
- **PR** #38354: (gmacon) Use –all when calling pip.py @ *2016-12-20 20:40:21 UTC*
  - 12436efb54 Merge pull request #38354 from gmacon/pip-freeze-all
  - dca24b270e Use –all when calling pip.py
- **PR** #38348: (rallytime) Update autodoc topics for new modules added in 2016.11 @ *2016-12-20 20:36:20 UTC*
  - 68430b1fa6 Merge pull request #38348 from rallytime/mod-docs-2016.11
  - b31c2412ca Add __iter__ and next options to doc/conf.py
  - b8c16094c4 Revert "Move import/error messaging logic for snapper module into __virtual__()"
  - 640db5b5ac Move import/error messaging logic for snapper module into __virtual__()
  - 366271f459 Add snapper to state index doc module list
  - 135d254c80 Remove netapi autodoc files: they should not be added as their doc structure is different
  - 0006139aca Update autodoc topics for new modules added in 2016.11
- **PR** #38377: (DmitryKuzmenko) Implementation and docs for Consul key-value store plugin for minion data cache. @ *2016-12-20 20:36:02 UTC*
  - 6ee7b2bae7 Merge pull request #38377 from DSRCorporation/features/consul_cache
  - 6fb4430ae3 Configuration options and documentation for Consul data cache plugin.
  - dad748f57a Data cache plugin configuration documentation.
  - c7209cd90c Consul data cache plugin.
- **PR** #38373: (rallytime) Back-port #38212 to 2016.11 @ *2016-12-20 20:35:09 UTC*

  - **PR** #38212: (disaster123) ZMQ: add an option for zmq.BACKLOG to salt master (zmq_backlog) (refs: #38373)
  - f6d1b559bc Merge pull request #38373 from rallytime/bp-38212
  - 52fc6daac0 ZMQ: add an option for zmq.BACKLOG to salt master (zmq_backlog)
- **PR** #38374: (mirceaulinic) NAPALM proxy module: Fix optional_args key issue @ *2016-12-20 20:34:59 UTC*
  - 69c3f19fc1 Merge pull request #38374 from cloudflare/FIX-NAPALM-PROXY
  - 44169315d8 Fix optional_args key issue
- **ISSUE** #38048: (ezh) [2016.11.0] Salt-cloud throws TypeError exception (refs: #38073)
- **PR** #38073: (ezh) 2016.11 @ *2016-12-20 14:51:11 UTC*
  - 530f495955 Merge pull request #38073 from doublescoring/2016.11
  - 42d3d26f28 [38073] Fix test assertion
  - 9b37ead913 Fix broken os.write without string.encode
- **PR** #38344: (bbinet) Fix influxdb_database.present state @ *2016-12-20 13:57:45 UTC*
  - 67908d5aba Merge pull request #38344 from bbinet/fix-influx-createdb
  - c6b075d6f4 Fix influxdb_database.present state
- **PR** #38358: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2016-12-20 00:11:48 UTC*
  - 04d6898958 Merge pull request #38358 from rallytime/merge-2016.11
  - c6e191ad0d Remove doc markup references from 2016.11 branch
  - 513058945c Merge branch '2016.3' into '2016.11'
  - 09d9cff992 Merge pull request #38288 from terminalmage/archive-extracted-local-source-2016.3
    - 845e3d0e75 Update tests to reflect change in cache behavior
    - 5a08d7c70a archive.extracted: don't try to cache local sources (2016.3 branch)
  - bf37667f8a Merge pull request #38312 from cro/proxy_config_in_cfg
    - 2006c4000e Typo
    - 689d95b10f Backport feature allowing proxy config to live in pillar OR /etc/salt/proxy.
  - c83db5a785 Merge pull request #38320 from rallytime/cleanup-doc-refs
    - 62978cb7a0 Don't check the doc/conf.py file for doc markup refs
    - 770e732d76 Add a unit test to search for new doc markup refs
    - 5c42a361a0 Remove ":doc:" references from all doc/topics/installation/* files
    - 23bce1c929 Remove ":doc:" references from all doc/topics/releases/* files
    - 4aafa41d22 Remove ":doc:" references from a bunch of doc/* files
    - 02bfe7912c Remove more ":doc:" references from doc/* files
    - 6e32267d0c Remove ":doc:" references in salt/* files
  - **PR** #38285: (terminalmage) archive.extracted: don't try to cache local sources
- **PR** #37947: (vutny) Fix *salt-minion* initscript for RHEL5 (SysV) to pick up proper python version @ *2016-12-19 21:03:50 UTC*
  - 13414949e3 Merge pull request #37947 from vutny/fix-rhel5-minion-init

  - c94e798b8a SysV init script for rpm: get and show unique PIDs only
  - 8ff68c4128 Fix initscript for RHEL5 (SysV) to pick up proper python version
- **PR** #38106: (techhat) ”test” is not necessarily in opts, for thorium @ *2016-12-19 14:40:32 UTC*
  - 4d072ca689 Merge pull request #38106 from techhat/stateget
  - 5edc16f606 ”test” is not necessarily in opts, for thorium
- **PR** #38333: (amendlik) Suppress errors when checking if an alternative exists @ *2016-12-19 13:40:49 UTC*
  - a01fade604 Merge pull request #38333 from amendlik/states-alternatives
  - 8bfcd5bcd5 Adjust alternatives test for updated error message
  - 09dee3c611 Suppress errors when checking if an alternative exists
- **PR** #38340: (ewapptus) Backport PR #38251: Fixed nested orchestrate not respecting failures @ *2016-12-19 13:31:16 UTC*
  - **PR** #38251: (ewapptus) Fixed nested orchestrate not respecting failures (refs: #38340)
  - 15d3b476e9 Merge pull request #38340 from ewapptus/bp-38251
  - 266e0a465c Fixed nested orchestrate not respecting failures
- **PR** #38229: (mcalmer) provide kwargs of sls_build to dockerng.create @ *2016-12-18 13:13:10 UTC*
  - ecd441d090 Merge pull request #38229 from mcalmer/dockerng-sls_build-kwargs
  - e7292fabb7 make it explicit that we want to delete these keys
  - 4c710139b5 use default values for pop() to prevent KeyError raised
  - 455c18325c provide kwargs to dockerng.create to provide all features to sls_build as well
- **ISSUE** #36204: (stanvarlamov) Salt-Cloud: salt.runners.cloud.create exits with True on Python process (ec2.py) exception (refs: #37333)
- **PR** #38309: (ewapptus) Backport PR #37333: Fixed state.salt.runner() reporting success on exceptions @ *2016-12-18 12:39:53 UTC*
  - **PR** #37333: (benediktwerner) Fixed state.salt.runner() reporting success on exceptions (refs: #38309)
  - d2ce9c3e71 Merge pull request #38309 from ewapptus/bp-37333
  - a2b1259671 Fixed display of errors
  - 14a39f914e Fixed state.salt.runner return value on exceptions
- **PR** #38323: (rallytime) Update the Cloud Provider Specifics links in cloud docs @ *2016-12-18 12:30:49 UTC*
  - ebb9f6cbbc Merge pull request #38323 from rallytime/update-cloud-provider-links
  - 022caf23e9 Update the Cloud Provider Specifics links in cloud docs
- **PR** #38324: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2016-12-18 12:30:26 UTC*
  - 5bd7471e30 Merge pull request #38324 from rallytime/merge-2016.11
  - 5940db5b3f Merge branch ’2016.3’ into ’2016.11’
    - 6367ca7d2a Add nick to args for create_multi (#38281)
    - 235682b1e6 Merge pull request #38313 from dragon788/2016.3-chocolatey-fix
      - 1f5fc17551 Use machine readable output for list
      - cdbd2fbe3c Added limit-output to eliminate false packages

    - 9e78ddc80e Merge pull request #38279 from rallytime/fix-38174
      - 4a62d01577 Add docs for syndic_wait setting

- **ISSUE** #38246: (martintamare) Windows Minion unable to start via nssm (refs: #38247)
- **PR** #38325: (rallytime) Back-port #38247 to 2016.11 @ *2016-12-18 12:28:41 UTC*
  - **PR** #38247: (martintamare) fix(win_function): handle other language (refs: #38325)
  - 83523d2f73 Merge pull request #38325 from rallytime/bp-38247
  - 4b6c5438e3 fix(win_functions): syntax
  - e602f17e3d fix(win_function): handle other language
- **ISSUE** #30195: (Vaelatern) Add Void Linux support in Salt (refs: #31262, #38326)
- **PR** #38326: (yopito) fix runit init support (grain init) in 2016.11 @ *2016-12-18 12:07:25 UTC*
  - **PR** #31262: (Vaelatern) Add support for Void Linux (refs: #38326)
  - 54a2bb95de Merge pull request #38326 from yopito/fix-runit-init-support
  - 25b91bb686 fix detection of runit as init system (grain init)
  - **PR** #38322: (rallytime) Add azurearm module to doc index
- **PR** #38305: (dereckson) Avoid normalization call for normalized mode value @ *2016-12-16 17:31:25 UTC*
  - 1e4f299e7d Merge pull request #38305 from dereckson/fix-mode-extraneous-normalization
  - 573ac3565e Avoid normalization call for normalized mode value
  - **PR** #38291: (terminalmage) Improve documentation for archive.extracted in 2016.11
- **ISSUE** #37966: (Cybolic) salt-cloud EC2 instance can't be initiated (refs: #37967)
- **PR** #38298: (rallytime) Back-port #37967 to 2016.11 @ *2016-12-16 15:20:04 UTC*
  - **PR** #37967: (Cybolic) Fixed faulty logic preventing instance initialisation. (refs: #38298)
  - 3cf0135d50 Merge pull request #38298 from rallytime/bp-37967
  - 42d367f39d Fixed faulty logic preventing instance initialisation.
- **ISSUE** #38070: (ezh) [2016.11.0] Salt-cloud throws UnicodeDecodeError exception (refs: #38076)
- **PR** #38076: (ezh) Fix decoding of broken string from remote sources @ *2016-12-15 19:05:25 UTC*
  - f4f0036f30 Merge pull request #38076 from doublescoring/fix-2016.11-38070
  - 70c8db5489 Fix decoding of broken string from remote sources
- **PR** #38278: (rallytime) Back-port #38207 to 2016.11 @ *2016-12-15 18:09:27 UTC*
  - **PR** #38207: (tsaridas) remove empty strings from list but not ones with one empty space char (refs: #38278)
  - **PR** #38188: (tsaridas) fix for push_dir in different OS (refs: #38203, #38207)
  - 2ccab22c19 Merge pull request #38278 from rallytime/bp-38207
  - 5e8bf571d8 python3 compatibility and fix pylint
  - e0df047000 remove empty strings from list but not ones with one empty space char
- **PR** #38277: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2016-12-15 18:09:10 UTC*
  - a748e842a8 Merge pull request #38277 from rallytime/merge-2016.11

  - 49a3355915 Merge branch '2016.3' into '2016.11'
  - fc9e1dff35 Merge pull request #38248 from meaksh/salt-api-successfully-close-child-processes
    - ee6eae9855 Successfully exit of salt-api child processes when SIGTERM.
  - 3c718ed35e Merge pull request #38254 from terminalmage/check-pillarenv
    - fa9ad311c6 Also check if pillarenv is in opts
  - 6b9060c38f [2016.3] Bump latest release version to 2016.11.1 (#38256)
- **ISSUE** #38231: (tjuup) Typo: salt-key deleteed (refs: #38232)
- **PR** #38232: (rallytime) Strip final 'e' in key cmd to correct "deleteed" misspelling @ *2016-12-15 10:38:49 UTC*
  - 0af343e71f Merge pull request #38232 from rallytime/fix-38231
  - 26e1ee3650 Strip final 'e' in key cmd to correct "deleteed" misspelling
- **ISSUE** #38200: (sebw) selinux.mode doesn't return any output and doesn't persist (refs: #38236)
- **PR** #38236: (gtmanfred) SELINUXTYPE should not be changed @ *2016-12-15 10:37:06 UTC*
  - 6c1ca9dae7 Merge pull request #38236 from gtmanfred/2016.11
  - d1b070c894 clean up selinux unit test
  - 96eabd4939 SELINUXTYPE should not be changed
- **ISSUE** #38228: (vquiering) archive.extracted with options and user/group (refs: #38262)
- **PR** #38262: (terminalmage) Fix archive.extracted when –strip or –strip-components is in the options @ *2016-12-15 08:57:18 UTC*
  - fd32dc3e9b Merge pull request #38262 from terminalmage/issue38228
  - 6442f8a7b5 Add tests for –strip/–strip-components
  - c502e68f12 Detect –strip/–strip-components in tar options and handle properly
  - e95770594d Add strip_components arg to archive.list
- **PR** #38264: (mirceaulinic) Port #37862 into 2016.11 @ *2016-12-15 08:51:20 UTC*
  - **PR** #37862: (mirceaulinic) [2016.11.1] Docstring fixes and new features for napalm_network (refs: #38264)
  - b232bd8ce8 Merge pull request #38264 from cloudflare/PORT-37862
  - 28bbb73151 Import from napalm_base instead of napalm
  - 0a675afc40 Vice-versa docstring
  - 09c50176e2 More docfix
  - 215b8f38e2 Lint cleanup
  - **PR** #38260: (rallytime) Add 2016.11.2 release notes
  - **PR** #38257: (rallytime) [2016.11] Bump latest release version to 2016.11.1
  - **PR** #38233: (terminalmage) Correct an inaccurate warning when top_file_merging_strategy == merge_all
- **PR** #38234: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2016-12-13 18:28:02 UTC*
  - ba62fcf2ec Merge pull request #38234 from rallytime/merge-2016.11
  - 6a327d1367 Merge branch '2016.3' into '2016.11'

  - 004e46afe7 Merge pull request #38198 from vutny/unit-tests-require-libcloud-boto3
    - a6098bac1a Remove note about SaltTesting installation, now it is in the requirements
    - 004bff113e Add missing requirements for running unit tests: libcloud and boto3
  - 9d497bc74c Merge pull request #38213 from rallytime/skip-tls-test
    - bdb807fc7c Skip test_cert_info tls unit test on pyOpenSSL upstream errors
  - 203109dd17 Merge pull request #38224 from whiteinge/cors-options-unauthed
    - de4d3227ab Allow CORS OPTIONS requests to be unauthenticated
  - 721a5feccd Merge pull request #38223 from whiteinge/salt-api-root_dirs
    - bfbf390c0e Add root_dir to salt-api file paths
- **PR** #38205: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2016-12-12 18:13:18 UTC*
  - 7ead1ed336 Merge pull request #38205 from rallytime/merge-2016.11
  - e31f97cf71 Merge branch '2016.3' into '2016.11'
  - 70f7d22ad6 Merge pull request #38191 from terminalmage/issue38162
    - 1ae543a98a Clarify the fact that git_pillar.update does not fast-forward
  - 28171cbfc5 Merge pull request #38194 from vutny/integration-test-requirements-doc
    - e9f419ff64 Document the requirements for running ZeroMQ-based integration tests
  - a4ef037ab1 Merge pull request #38185 from rallytime/bp-38181
  - 609f814454 Reset socket default timeout to None (fixes daemons_tests failures)
- **PR** #38203: (rallytime) Back-port #38188 to 2016.11 @ *2016-12-12 17:48:51 UTC*
  - **PR** #38188: (tsaridas) fix for push_dir in different OS (refs: #38203, #38207)
  - 669409d681 Merge pull request #38203 from rallytime/bp-38188
  - 50d3200b12 removing not needed join
  - 7af708e1e7 fix for push_dir in different OS

28.2.34 Salt 2016.11.3 Release Notes

Version 2016.11.3 is a bugfix release for *2016.11.0*.

Statistics

- Total Merges: **137**
- Total Issue References: **49**
- Total PR References: **130**
- Contributors: **47** (Ch3LL, DmitryKuzmenko, MTecknology, The-Loeki, UtahDave, anlutro, arthru, axmetishe, bailsman, bobrik, cachedout, clinta, corywright, cro, dmaziuk, dmitrievav, dmurphy18, eliasp, eradman, ezh, gtmanfred, hu-dabao, hujunya, isbm, jak3kaj, janhorstmann, joe-niland, kevinanderson1, kstreee, l2ol33rt, lomeroe, mcalmer, meaksh, mirceaulinic, morganwillcock, nasenbaer13, nicholasmhughes, rallytime, sakateka, sergeizv, sjorge, techhat, terminalmage, thatch45, toanju, twangboy, vutny)

Changelog for v2016.11.2..v2016.11.3

Generated at: 2018-05-27 19:39:56 UTC

- **PR** #39536: (twangboy) Namespace 'status' functions in 'win_status' @ *2017-02-21 23:45:31 UTC*
  - **PR** #39005: (cro) Ungate the status.py module and raise unsupported errors in functions not executable on Windows. (refs: #39536)
  - 40f72db53e Merge pull request #39536 from twangboy/fix_win_status
  - d5453e2f9e Remove unused import (lint)
  - 837c32e673 Remove list2cmdline
  - c258cb3f73 Streamline wmic command returns for easier parsing
  - 6d2cf8171e Fix 'ping_master' function
  - d946d10597 Namespace 'status' functions in 'win_status'
- **PR** #39534: (rallytime) Fix breakage in aptpkg and dpkg execution modules @ *2017-02-21 20:31:15 UTC*
  - **PR** #39418: (anlutro) Allow aptpkg.info_installed on package names that aren't installed (refs: #39534)
  - dc8f578447 Merge pull request #39534 from rallytime/fix-pkg-function-specs
  - d34a8fe9dc Fix breakage in aptpkg and dpkg execution modules
- **ISSUE** #34712: (richardscollin) Salt Test Suite Error - develop (refs: #37366)
  - **PR** #39521: (vutny) Upgrade SaltTesting to run test suite for 2016.11 and add SaltPyLint
  - **PR** #37366: (eradman) dev_python*.txt: use current SaltTesting and SaltPyLint modules (refs: #39521)
- **PR** #39370: (twangboy) Gate win_osinfo and winservice @ *2017-02-17 23:53:58 UTC*
  - e4c71683d9 Merge pull request #39370 from twangboy/gate_win_utils
  - 167cdb3447 Gate windows specific imports, add __virtual__
  - e67387deb7 Add option to return a Non instantiated class
  - 315b0cc105 Clarify return value for win_osinfo
  - 994314ed3d Fix more docs
  - 2bbe3cbf49 Fix some docs
  - 4103563ee1 Merge branch 'gate_win_utils' of https://github.com/twangboy/salt into gate_win_utils
    - 24c1bd079d Remove extra newlines
  - 82a86ced55 Add helper function for winservice
  - 0051b5a5e2 Put the win_osinfo classes in a helper function
  - 4e08534877 Gate win_osinfo and winservice better
- **PR** #39486: (twangboy) Remove orphaned function list_configurable_policies @ *2017-02-17 22:21:50 UTC*
  - a3e71b6cce Merge pull request #39486 from twangboy/win_remove_orphaned
  - 1328055c4d Remove orphaned function list_configurable_policies
- **PR** #39418: (anlutro) Allow aptpkg.info_installed on package names that aren't installed (refs: #39534) @ *2017-02-17 18:34:19 UTC*
  - 87b269fc80 Merge pull request #39418 from alprs/fix-aptpkg_info_nonexistent_pkg

  - 246bf1e938 add failhard argument to various apt pkg functions
- **PR** #39438: (mirceaulinic) file.get_managed: refetch source when file hashsum is changed @ *2017-02-17 17:58:29 UTC*
  - e816d6c23e Merge pull request #39438 from cloudflare/fix_39422
  - 8453800639 file.get_managed: refetch cached file when hashsum chnaged
- **ISSUE** #39203: (dmaziuk) salt.users gecos field (refs: #39432)
- **PR** #39432: (dmaziuk) Quick and dirty fix for GECOS fields with more than 3 commas @ *2017-02-17 17:57:30 UTC*
  - a5fe8f0fa6 Merge pull request #39432 from dmaziuk/issue39203
  - 41c046308c Remove #
  - 4f877c6b6f Quick and dirty fix for GECOS fields with more than 3 commas
- **PR** #39484: (corywright) The Reactor docs should use pillar='{}' instead of 'pillar={}' @ *2017-02-17 17:50:57 UTC*
  - 3665229a5a Merge pull request #39484 from corywright/fix-reactor-docs-pillar-keyword-args
  - cc90d0d53f The Reactor docs should use pillar='{}' instead of 'pillar={}'
- **PR** #39456: (twangboy) Add salt icon to buildenv directory @ *2017-02-16 22:47:58 UTC*
  - 2e3a9c5e58 Merge pull request #39456 from twangboy/win_fix_icon
  - 8dd915dae4 Add salt icon to buildenv directory
- **PR** #39462: (twangboy) Use url_path instead of url_data.path @ *2017-02-16 22:44:18 UTC*
  - 63adc03484 Merge pull request #39462 from twangboy/win_fix_fileclient
  - a96bc13133 Use url_path instead of url_data.path
- **PR** #39458: (rallytime) Fix more warnings in doc build @ *2017-02-16 21:45:52 UTC*
  - e9b034f02f Merge pull request #39458 from rallytime/fixup-more-doc-build-warnings
  - e698bc3508 Fix more warnings in doc build
- **PR** #39437: (sakateka) Fixes about saltfile @ *2017-02-16 20:32:15 UTC*
  - e4f8c2bfb0 Merge pull request #39437 from sakateka/fixes_about_saltfile
  - ab68524d7a less pylint: salt/utils/parsers.py
  - 9e7d9dcc78 Revert "pylint: salt/utils/parsers.py"
  - f3f129c8f1 document ~/.salt/Saltfile
  - 33f3614b1e pylint: salt/utils/parsers.py
  - 0f36e10e7d expand config_dir and '~/.salt/Saltfile' as last resort
  - **PR** #39451: (Ch3LL) add 2016.11.3 changelog to release notes
- **ISSUE** #38032: (meggiebot) Add missing Carbon docs (refs: #39448)
- **PR** #39448: (gtmanfred) Add release notes for cisco proxy minions added in Carbon @ *2017-02-16 17:29:48 UTC*
  - 8e2cbd2307 Merge pull request #39448 from gtmanfred/2016.11
  - 3172e88700 Add release notes for cisco proxy minions added in Carbon
- **PR** #39428: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-02-16 00:01:15 UTC*
  - 070904b719 Merge pull request #39428 from rallytime/merge-2016.11

  - 2acb188ac9 Change `path` value from a tuple to a list
  - 6d78adbf08 Merge branch '2016.3' into '2016.11'
    - 4ff13acf8b salt.fileserver.roots: Fix regression in symlink_list (#39409)
    - 8b8ab8ef8e Merge pull request #39362 from dincamihai/cp-push-test-2016.3
      - 91383c5a19 Add cp.push test
    - 4b726f955b Merge pull request #39380 from joe-niland/quote-numeric-usernames
      - c2edfdd464 Quote numeric user names so pwd.getpwnam handles them properly
    - 1116d32df9 Merge pull request #39400 from meaksh/2016.3-fix-local-cache-issue
      - e7e559ef5c Prevents 'OSError' exception in case path doesn't exist
    - 6c854da1d4 Merge pull request #39300 from terminalmage/loader-optimization
      - d3e5d1525e Replace more usage of str.format in the loader
    - 5286b5ff1b Merge pull request #39337 from terminalmage/issue34428
      - a7d2135dc2 Don't re-walk the roots fileserver in symlink_list()
    - ce781deeb5 Merge pull request #39339 from cro/pillar_filetree_doc
      - 410810cea2 Clarification on external pillar usage.
    - fa3014393c Document the upstream RedHat bug with their pygit2 package (#39316)
- **ISSUE** #39360: (bbinet) file.symlink should not try to set ownership to root:root (refs: #39364)
- **PR** #39429: (rallytime) Back-port #39364 to 2016.11 @ *2017-02-15 21:27:21 UTC*
  - **PR** #39364: (gtmanfred) set default user variable to the user cmd runs as (refs: #39429)
  - 54a572e50c Merge pull request #39429 from rallytime/bp-39364
  - 157f4dcdf9 set default user variable to the user cmd runs as
- **PR** #39424: (twangboy) Fix problem with too many connection attempts in Windows @ *2017-02-15 18:51:35 UTC*
  - 881ebf2e93 Merge pull request #39424 from twangboy/win_fix_dos
  - d3f7dd7f50 Add sleep to eval_master
- **ISSUE** #30561: (jfindlay) salt-ssh fails with IPv6 address (refs: #39419, #38831)
- **ISSUE** #22984: (tomasfejfar) salt-ssh problem possibly related to ipv6 (refs: #39419, #38831)
- **PR** #39419: (The-Loeki) Backport Salt-SSH IPv6 fixes to 2016.11 @ *2017-02-15 17:33:13 UTC*
  - **PR** #38877: (The-Loeki) Salt-SSH client: Don't overwrite self.host w/IPv6 brackets (refs: #39419)
  - **PR** #38831: (The-Loeki) Salt-SSH deal with raw IPv6 addresses (refs: #39419, #38877)
  - 47872355a8 Merge pull request #39419 from The-Loeki/bp-ssh-ipv6
  - 4fc5626f16 Don't overwrite self.host w/IPv6 brackets
  - dd1223468b Salt-SSH deal with raw IPv6 addresses
- **PR** #39379: (terminalmage) win_pkg: remove all installed versions when no explicit version passed @ *2017-02-14 18:41:28 UTC*
  - 878946d0f8 Merge pull request #39379 from terminalmage/issue34821
  - fd9ab8e4e3 Remove extra newline

  - 5871825b9e win_pkg: remove all installed versions when no explicit version passed
- **PR** #39392: (anlutro) Make sure OrderedDict order is preserved in nested output @ *2017-02-14 17:50:15 UTC*
  - caffef88cf Merge pull request #39392 from alprs/fix-nested_output_ordered_dict
  - 625a770a23 make sure OrderedDict order is preserved in output
- **PR** #39378: (dmurphy18) Update make_repo in debbuild.py execution module to utilize timeout @ *2017-02-14 17:10:15 UTC*
  - f2459e3ce8 Merge pull request #39378 from dmurphy18/deb_pkg_fix
  - 4bd47cc18a Updated all make_repo loops to use timeout value for retries
- **ISSUE** #39358: (Kimamisa) Backport the RDS fix in Carbon (refs: #39369)
- **PR** #39369: (rallytime) Back-port #37338 to 2016.11 @ *2017-02-13 21:41:19 UTC*
  - **PR** #37338: (bailsman) Fix wait_status in boto_rds.create() (refs: #39369)
  - 99554d9d72 Merge pull request #39369 from rallytime/bp-37338
  - 2e7f6e8e37 Fix wait_status in boto_rds.create()
- **PR** #39303: (kstreee) Removes a redundant test case after removed 'batch' in 'netapi'. @ *2017-02-13 19:55:46 UTC*
  - 03ab8b1b5a Merge pull request #39303 from kstreee/fix-testcase-rm-batch-in-netapi
  - 51972d0724 Removes a redundant test case after removed 'batch' in 'netapi'.
- **PR** #39315: (Ch3LL) improve salt-run salt.cmd test @ *2017-02-13 19:00:14 UTC*
  - 60640f77d7 Merge pull request #39315 from Ch3LL/fix_run_salt_test
  - b3cbc5a408 improve salt-run salt.cmd test
- **ISSUE** #39243: (morganwillcock) win_system.reboot: can return True without rebooting (refs: #39311)
- **PR** #39311: (morganwillcock) win_system: return False from a skipped reboot @ *2017-02-13 18:59:11 UTC*
  - 2ca63a93cd Merge pull request #39311 from morganwillcock/skip-reboot
  - 0f3abb613d Clarify success for shutdown function
  - dcb4d05275 win_system: return False from a skipped reboot
- **PR** #39346: (joe-niland) Ignore non-HTTP IIS bindings @ *2017-02-13 18:18:36 UTC*
  - 082105fa84 Merge pull request #39346 from joe-niland/handle-iis-bindings
  - 8d5afdb0ae win_iis module: list_sites - when retrieving bindings, ignore bindigs whose protocols do not have host headers
- **ISSUE** #39321: (mgresser) Grain matching failing where grain value is an INT (refs: #39361)
- **PR** #39361: (gtmanfred) make sure both variables are strings. @ *2017-02-13 17:20:17 UTC*
  - a3a9a8e1ed Merge pull request #39361 from gtmanfred/2016.11
  - ee2275ad67 make sure both variables are strings.
  - **PR** #39341: (eliasp) Add creation/configuration of Salt PKI dirs to hacking docs
- **PR** #39317: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-02-10 23:07:32 UTC*
  - ce1f01f81a Merge pull request #39317 from rallytime/merge-2016.11
  - c1df446b7a Merge branch '2016.3' into '2016.11'

  - 9de559ff4e Merge pull request #39313 from rallytime/merge-2016.3
    - 0b8dddf12b Merge branch '2015.8' into '2016.3'
    - fc551bcf5d Merge pull request #39293 from sergeizv/grammar-fix
      - 70f2b586d3 Rewrap paragraph
      - e6ab5178ea Grammar fix
    - 8a1b45632a Merge pull request #39295 from sergeizv/typo-fix
      - 5d9f36d58d Fix typo
    - cfaafece34 Merge pull request #39296 from sergeizv/whitespace-fix
      - 1d4c1dc140 Whitespace fix in docs Makefile
    - 0b4dcf4a47 Merge pull request #39294 from sergeizv/fix-link
      - 04bde6eed2 Fix link in proxyminion guide
  - dd3ca0ecb0 Fix #38595 - Unexpected error log from redis retuner in master's log (#39299)
  - f16027d30e Merge pull request #39297 from cro/pg_returner_docs
    - 28bac649ae Typo
    - 19fedcdd23 Add doc to recommend pgjsonb for master job caches
  - 77e50ed8b7 Merge pull request #39286 from terminalmage/fix-pillarenv-precedence
    - 3cb9833e57 Allow minion/CLI saltenv/pillarenv to override master when compiling pillar
  - 52440416ca Merge pull request #39221 from lvg01/fix-bug-39220
    - e8a41d6341 Removes to early content stripping (stripping is allready done when needed with ident:true), fixes #39220
    - a4b169e0bd Fixed wrong logic, fixes #39220
  - 5a27207c57 Add warning for Dulwich removal (#39280)
- **ISSUE** #38451: (ezh) 2016.11 file.replace has multiple errors under python 3 (refs: #38464)
- **PR** #38464: (ezh) [38451] Fix file.replace 2016.11 @ *2017-02-09 23:07:49 UTC*
  - c3c621aab0 Merge pull request #38464 from doublescoring/fix-2016.11-38451
  - 81f0337338 [38451] Fix few bugs after review
  - 1bdab253ad [38451] Fix pylint W1699(incompatible-py3-code)
  - 3bfc6547da [38451] Fix file.replace to make it suitable to python 3
- **PR** #39291: (terminalmage) Add note about using saltenv jinja var in pillar top files @ *2017-02-09 21:43:50 UTC*
  - 6365211a6f Merge pull request #39291 from terminalmage/pillar-docs
  - fbd551e069 Add note about using saltenv jinja var in pillar top files
- **PR** #39281: (twangboy) Require VCRedist on 2008R2 and below instead of 2008 @ *2017-02-09 17:59:57 UTC*
  - a496ec2a16 Merge pull request #39281 from twangboy/win_installer
  - ef5078729a Capitalize the 'r' for 2008R2
  - 1b6bd634ac Require VCRedist on 2008R2 and below instead of 2008
- **PR** #39264: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-02-09 17:10:14 UTC*

  - db6140aa83 Merge pull request #39264 from rallytime/merge-2016.11
  - a9c2c106c1 Pylint fix
  - f6aad99db2 Merge branch '2016.3' into '2016.11'
    - 1b9217d636 Update jsonschema tests to reflect change in jsonschema 2.6.0 (#39260)
    - c1d16cc3d0 Better handling of enabled/disabled arguments in pkgrepo.managed (#39251)
    - 8e88f71dd9 Merge pull request #39227 from terminalmage/loader-optimization
      - c750662946 Loader optimzation
    - bc89b297f8 Merge pull request #39228 from gtmanfred/2016.3
      - afee047b08 default to utf8 encoding if not specified
    - d9b0671dbd Merge pull request #39231 from terminalmage/clarify-jenkins-depends
      - ad1b1255f2 Add clarification for jenkins execution module
    - ddcff89a84 Merge pull request #39232 from terminalmage/issue21342
      - c88896c277 Avoid recursion in s3/svn ext_pillars
    - ef4e437bbc Fix the win_ip_test failures (#39230)
    - df5f934c34 Merge pull request #39199 from rallytime/bp-39170
      - c129905310 Added missing source_hash_name argument in get_managed function Additional fix to #33187 Customer was still seeing errors, this should now work. Tested with 2015.8.13 and 2016.11.2
    - 2621c119fd Merge pull request #39206 from cachedout/issue_issue_37174
      - be31e0559c Ignore empty dicts in highstate outputter
    - dd440452ea Merge pull request #39209 from terminalmage/sorted-envs
      - e6dda4a625 Sort the return list from the fileserver.envs runner
    - 7bed68743e [2016.3] Pylint fix (#39202)
    - ab76054127 Merge pull request #39197 from cachedout/pr-38793
      - f3d35fb5c6 Lint fixes
      - 624f25b78d Fix for #38697

- **ISSUE** #39269: (alexharrington) Remount forced with lizardfs fuse filesystem due to device missmatch (refs: #39276)
- **ISSUE** #39106: (carsten-AEI) CVMFS fuse mount gets remounted every time (refs: #39276)
- **PR** #39276: (gtmanfred) _device_mismatch_ignored will never be True @ *2017-02-09 17:05:28 UTC*
  - 304eb19b18 Merge pull request #39276 from gtmanfred/2016.11
  - 6635a9fd3b _device_mismatch_ignored will never be True
- **PR** #39238: (dmurphy18) Update disk fstype, inodeusage, percent and mount.active functions for AIX support @ *2017-02-08 21:53:32 UTC*
  - 7611698474 Merge pull request #39238 from dmurphy18/fix_aix_disk_mount
  - a8a519c493 Removed space for pylint
  - 8fa0ffa427 Updates due to code review comments

  - 97c59a8d1c Updated mount functionality for active on AIX
  - 1a32b2cc89 Updated disk functionality for fstype, inodeuage and percent on AIX
- **PR** #39233: (rallytime) Various doc updates to fix warnings in doc build @ *2017-02-08 19:29:53 UTC*
  - 99bfa7dfee Merge pull request #39233 from rallytime/fixup-more-doc-build-warnings
  - 2f74dcf685 Various doc updates to fix warnings in doc build
- **PR** #39237: (axmetishe) fix rds subnet group creation @ *2017-02-08 19:04:31 UTC*
  - 59e927b520 Merge pull request #39237 from axmetishe/2016.11
  - 6f4be8b69c fix rds subnet group creation
  - **PR** #39234: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11
  - **PR** #39225: (terminalmage) Put legacy git_pillar on a deprecation path for Oxygen
- **ISSUE** #39078: (morganwillcock) setup.py: cannot install without setting global options (refs: #39180)
- **PR** #39180: (morganwillcock) setup.py: Remove global options from install command @ *2017-02-07 16:20:49 UTC*
  - 19c3d90a32 Merge pull request #39180 from morganwillcock/setup
  - d7e05091a2 Remove global options from Install
- **PR** #38863: (hujunya) fix django auth not work @ *2017-02-07 15:43:00 UTC*
  - a0907bc861 Merge pull request #38863 from hujunya/fix_django_auth
  - 2a99ff46d3 check if django_auth_path has been in sys.path
  - 933ebf15d7 fix pylint violations
  - 6b5a7f4b64 fix django auth not work
- **PR** #39198: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-02-06 21:01:56 UTC*
  - c3e541e0a2 Merge pull request #39198 from rallytime/merge-2016.11
  - 7ea5f7f82f Merge branch '2016.3' into '2016.11'
    - fa45cbc359 Merge pull request #39166 from Ch3LL/fix_boto_ec2_docs
      - 90af696331 fix boto ec2 module create_image doc
    - a40cb46249 Merge pull request #39173 from rallytime/restore-community-docs
      - 5aeddf42a0 Restore "Salt Community" doc section
- **ISSUE** #39059: (mirceaulinic) KeyError: 'multiprocessing' in the master logs (proxy minions) (refs: #39063)
- **PR** #39063: (mirceaulinic) Avoid KeyError: 'multiprocessing' in the master logs @ *2017-02-06 19:37:35 UTC*
  - 2a85d73f59 Merge pull request #39063 from cloudflare/ISS-39059
  - 7118eff034 Avoid KeyError: 'multiprocessing'
- **ISSUE** #38782: (lomeroe) win_lgpo unable to find some Administrative Template policies (refs: #38783, #39083, #39090)
- **ISSUE** #38761: (DaveOHenry) Cannot apply state that contains lgpo.set (refs: #39083, #39088)
- **ISSUE** #38689: (lomeroe) win_lgpo state fails to set single policy due to case sensitive check (refs: #39083, #39084, #38690)
- **ISSUE** #38100: (skjaro) Problem with win_lgpo.py in salt 2016.11.0 (refs: #39083, #39089, #38779)

- **ISSUE** #21485: (lorengordon) Feature Request: Manage Windows Local Security Policy Settings (refs: #36336)
- **PR** #39083: (lomeroe) Backport #36336 to 2016.11 @ *2017-02-06 18:50:52 UTC*
  - **PR** #36336: (lomeroe) add additional static policies to computer configuration policy class (refs: #39083)
  - 91c25bd651 Merge pull request #39083 from lomeroe/bp-36336
  - 03e5319124 Merge branch '2016.11' into bp-36336
  - 981ec89a4d update command line example to correct policy name
  - e2574da0b8 Fix/Add documentation, 80 char line lengths
  - 5e94a30a34 add additional static policies to computer configuration policy class duplicate code cleanup/misc code efficiencies
- **PR** #39153: (nicholasmhughes) Fix selinux.mode state config file handling @ *2017-02-06 18:37:34 UTC*
  - 30455079fe Merge pull request #39153 from nicholasmhughes/fix-selinux.mode-config-predictability
  - 8d8ba9c7d2 added the new getconfig function to the test
  - a6a24e1a1b Addressed edge case when attempting to set the config file to 'Disabled'. The state should only check the file, since the in-memory setting won't disappear until after reboot.
  - 6858658cc2 The selinux.mode state only checked the current status of SELinux in memory (getenforce) when determining if changes needed to be made. The /etc/selinux/config file could have a different value, and it would not be changed. This commit enhances idempotency of the state in regards to both the in-memory and configuration file enforcement of SELinux.
- **ISSUE** #38081: (haraldrudell) x509 state or module cannot generate password protected private keys (refs: #39159)
- **PR** #39159: (clinta) Csr crl passphrase @ *2017-02-06 18:36:05 UTC*
  - 7b5eb17cbe Merge pull request #39159 from clinta/csr-crl-passphrase
  - cf548ac717 Remove unnecessary pass
  - 4ebf7a3df4 Remove unnecessary pass statement
  - 6a8046970e fix csr bugs and pep8
  - 36dcf5f3da only overwrite if overwrite option is specified
  - 403000d375 recreate cert on bad password
  - 6497094ba7 passphrase for crl
  - 3ef809fb0f passphrase for csr
- **PR** #39162: (meaksh) Adding more function to Snapper module @ *2017-02-06 18:33:53 UTC*
  - b240468525 Merge pull request #39162 from meaksh/snapper-module-improvements
  - f950732fa0 pylint fixes
  - aa2f9906e0 Removing extra spaces
  - 9d6a33f257 Adds 'snapper.create_config' unit tests
  - d38ed505f8 Adds 'snapper.modify_snapshots' unit tests
  - d5496ccc99 Adds 'snapper.delete_snapshots' unit tests
  - 3eecb6076b Snapper: Adding support for creating configurations
  - 041e54d42a Snapper: Adding support for snapshot metadata modification

  - eaf5de9dce Snapper: Adding support for deleting snapshots
- **ISSUE** #38370: (tjyang) Salt-Cloud: There was a query error: Required field "deviceChange" not provided (not @optional) (refs: #39171)
- **PR** #39171: (techhat) Raise an error for a disk size that is too small @ *2017-02-06 18:19:46 UTC*
  - 6f9251ebed Merge pull request #39171 from techhat/issue38370
  - ec57a39c00 Typo
  - 2ed2932387 Clean up debug logs
  - 671282656a Raise an error for a disk size that is too small
- **PR** #39179: (mcalmer) fix error parsing @ *2017-02-06 17:57:00 UTC*
  - 036f36dc9b Merge pull request #39179 from mcalmer/fix-dockerng-error-parsing
  - 6750ccd78e fix error parsing
- **PR** #39189: (morganwillcock) Fix NetBSD sockstat parsing @ *2017-02-06 17:28:08 UTC*
  - 30f83156cb Merge pull request #39189 from morganwillcock/sockstat
  - 344d13eff5 Fix NetBSD sockstat example
  - 64b693195c Fix NetBSD sockstat parsing
- **ISSUE** #38003: (morganwillcock) salt.runners.cache functions seem to ignore minion targeting parameter (refs: #39141)
- **PR** #39141: (UtahDave) Don't overwrite the minion_ids var that was passed @ *2017-02-03 20:56:25 UTC*
  - 6a9704189f Merge pull request #39141 from UtahDave/fix_cache_lookup_ZD1187
  - 0340614d15 return all minions' grains if no tgt
  - f833bf3a79 Don't overwrite the minion_ids var that was passed
- **PR** #39164: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-02-03 17:57:07 UTC*
  - d19cee716f Merge pull request #39164 from rallytime/merge-2016.11
  - 6504bb6b02 Merge branch '2016.3' into '2016.11'
    - 9de08af950 Apply fix from #38705 to 2016.3 branch (#39077)
    - da3053ea9b update vmware getting started doc (#39146)
    - e78ca0f575 Fixing a weird edge case when using salt syndics and targetting via pillar. Without this fix the master of masters ends up in an infinite loop since the data returned from the minions is differently structured than if a sync was not in use. (#39145)
    - cd8077ab81 Merge pull request #38804 from alexbleotu/root_dir_fix-2016.3-gh
      - b3bdd3b04a Add missing whiteline
      - c7715acd53 Merge pull request #3 from cro/ab_rootdirfix
      - e8cbafaaf1 When running testsuite, salt.syspaths.ROOT_DIR is often empty.
      - b12dd44a26 Merge pull request #1 from cro/ab_rootdirfix
      - bffc537aca Remove extra if statements (rstrip will check for the presence anyway).
      - 97521b3468 Second attempt to fix prepending of root_dir to paths
    - 6ffeda3ee5 Clarify ipv6 option for minion and inteface for master, closes #39118 (#39131)

    - 646b9ea4e5 Don't abort pillar.get with merge=True if default is None (#39116)

- **PR** #39152: (twangboy) Remove files not needed by salt-minion @ *2017-02-03 17:11:11 UTC*
  - ed12512045 Merge pull request #39152 from twangboy/win_installer
  - 5ff8a14317 Fix problem deleting files
  - 4524dd49d4 Remove files not needed by salt-minion
- **ISSUE** #38691: (lomeroe) win_lgpo module throws a key error when run with return_not_configured=True (refs: #39085, #38666)
  - **PR** #39085: (lomeroe) Backport #38666 to 2016.11
  - **PR** #38666: (lomeroe) correct issue when running lgpo.get with return_not_configured=True (refs: #39085)
  - **PR** #39086: (lomeroe) Backport #38165 to 2016.11
  - **PR** #38165: (lomeroe) have _in_range_inclusive function attempt to convert a string to an i… (refs: #39086)
- **ISSUE** #38241: (frogunder) mine.get and salt-ssh gives error message (refs: #38970)
- **PR** #38970: (gtmanfred) when using local_cache we have to pass the list of minions @ *2017-02-02 19:24:39 UTC*
  - 4eec641b65 Merge pull request #38970 from gtmanfred/2016.11
  - ebb9df3ec7 when using local_cache we have to pass the list of minions
- **ISSUE** #39110: (morganwillcock) archive.extracted: 2016.11.2 returns state failure for some zip formats, if already extracted (refs: #39128)
  - **PR** #39128: (terminalmage) Fix archive.list on Windows
- **ISSUE** saltstack/salt#36712: (dmitrievav) s3.put function does not create s3 bucket (refs: #36714)
  - **PR** #39133: (rallytime) Back-port #36714 to 2016.11
  - **PR** #36714: (dmitrievav) s3.put can't create s3 bucket (refs: #39133)
- **ISSUE** #38689: (lomeroe) win_lgpo state fails to set single policy due to case sensitive check (refs: #39083, #39084, #38690)
  - **PR** #39084: (lomeroe) Backport #38690 to 2016.11
  - **PR** #38690: (lomeroe) correct checking of policy_class to compare with lower() version of t… (refs: #39084)
- **ISSUE** #38100: (skjaro) Problem with win_lgpo.py in salt 2016.11.0 (refs: #39083, #39089, #38779)
  - **PR** #39089: (lomeroe) Backport #38779 to 2016.11
  - **PR** #38779: (lomeroe) win_lgpo handle errors when 'encoding="unicode"' exists in ADMX file (refs: #39089)
- **ISSUE** #38782: (lomeroe) win_lgpo unable to find some Administrative Template policies (refs: #38783, #39083, #39090)
  - **PR** #39090: (lomeroe) Backport #38783 to 2016.11
  - **PR** #38783: (lomeroe) Perform a "starts-with" search to match ADML text names (refs: #39090)
- **ISSUE** #38761: (DaveOHenry) Cannot apply state that contains lgpo.set (refs: #39083, #39088)
  - **PR** #39088: (lomeroe) Backport #37262 to 2016.11
  - **PR** #37262: (lomeroe) correct issues in win_lgpo module (refs: #39088)

- **PR** #39122: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-02-01 21:41:59 UTC*
  - 50d72da3f6 Merge pull request #39122 from rallytime/merge-2016.11
  - a782b00ee1 Merge branch '2016.3' into '2016.11'
  - cc9b69b6bc Merge pull request #39091 from terminalmage/update-test-valid-docs
    - d76f0380d0 add debug logging for batch vars
    - b4afea2a25 Don't fail test if data is empty
    - b3a5d549c1 Account for trimmed value in 'salt -d' output
    - 909916c78e Run test_valid_docs in batches
  - bcee3d1ef6 Move fileclient tests to tests/integration/fileserver/fileclient_test.py (#39081)
  - 122422bc08 Bump openstack deprecation notice to Oxygen (#39067)
- **PR** #39087: (lomeroe) Backport #37375 to 2016.11 @ *2017-02-01 19:02:58 UTC*
  - **PR** #37375: (lomeroe) add updating gpt.ini file when ADM template policies are modified (gp... (refs: #39087)
  - f8a6863d98 Merge pull request #39087 from lomeroe/bp-37375
  - c3aaa536f3 _in_range_inclusive class method incorrectly called isinstance
  - ce263f9372 set_computer_policy and set_user_policy call "set" by the original function name (set) instead of the aliased function name set_
  - ff7d74bfb0 correct tool extension guid for user registry policies
  - 08f0078ef3 spelling correction
  - 5fc40485f7 add updating gpt.ini file when ADM template policies are modified (gpt.ini file must exist with proper data for ADM policies to apply)
- **PR** #39094: (rallytime) Add a bunch of missing doc module references @ *2017-02-01 18:56:27 UTC*
  - c4c6e701af Merge pull request #39094 from rallytime/doc-build-warnings
  - b866427f59 Add a bunch of missing doc module references
- **PR** #39108: (janhorstmann) [Bugfix] Fix state x509.crl_managed @ *2017-02-01 18:32:43 UTC*
  - d302bb747e Merge pull request #39108 from janhorstmann/fix-x509-state
  - 9f5c532510 [Bugfix] Fix state x509.crl_managed
- **ISSUE** #39100: (whytewolf) salt-run fileserver.update Exception (refs: #39107)
- **ISSUE** #39098: (FraaJad) state.event runner fails with TypeError: argument of type 'NoneType' is not iterable (refs: #39107)
- **ISSUE** #38638: (mirceaulinic) *salt.cmd* runner raises TypeError when function returns bool (refs: #39107)
- **PR** #39107: (mirceaulinic) Check if data['return'] is dict type @ *2017-02-01 18:21:46 UTC*
  - bf61ec9515 Merge pull request #39107 from cloudflare/FIX-38638
  - 7c34815979 Check if data['return'] is dict type
- **ISSUE** #39065: (jak3kaj) primary bonding option is not applied (refs: #39068, #39069)
  - **PR** #39069: (jak3kaj) Update primary bonding option in rh_ip.py
- **ISSUE** #39065: (jak3kaj) primary bonding option is not applied (refs: #39068, #39069)

  - **PR** #39068: (jak3kaj) Update primary bonding option in debian_ip.py
- **ISSUE** #38704: (nasenbaer13) Archive extracted fails when another state run is queued (refs: #38705)
- **PR** #39076: (terminalmage) Re-submit PR #38705 against 2016.11 branch @ *2017-01-31 20:11:55 UTC*
  - **PR** #38705: (nasenbaer13) Fix for #38704 archive extracted and dockerio states (refs: #39077, #39076)
  - 9836d7dd29 Merge pull request #39076 from terminalmage/pr-38705
  - 15db8d47ed Fix for #38704 archive extracted and dockerio states
- **ISSUE** #39057: (sergeizv) modules.linux_lvm.fullversion provides incomplete info (refs: #39058)
- **PR** #39058: (sergeizv) Fix salt.modules.linux_lvm.fullversion @ *2017-01-31 19:01:12 UTC*
  - 86b4b77bfe Merge pull request #39058 from sergeizv/fix-lvm-fullversion
  - e46c89f9ed Fix salt.modules.linux_lvm.fullversion
  - fb7ef99838 Fix mock emulating lvm version
- **ISSUE** #39051: (afletch) salt.roster.cache / salt.utils.cloud is_public_ip - incorrect public IP address (refs: #39066)
- **PR** #39066: (techhat) 127.0.0.0/8 is all loopback @ *2017-01-31 18:43:22 UTC*
  - 721b245f90 Merge pull request #39066 from techhat/issue39051
  - ea43bb8101 127.0.0.0/8 is all loopback
- **ISSUE** #39070: (sergeizv) modules.linux_lvm.pvcreate misbehaves if all submitted devices are already LVM PVs (refs: #39071)
- **PR** #39071: (sergeizv) Fix modules.linux_lvm.pvcreate on existing LVM PVs @ *2017-01-31 18:36:54 UTC*
  - c54d9f4e2a Merge pull request #39071 from sergeizv/fix-lvm-pvcreate
  - f1e3e86e6a Fix modules.linux_lvm.pvcreate on existing LVM PVs
  - 0f84ca2487 Add test for modules.linux_lvm.pvcreate on existing LVM PVs
  - 3967992bfd Fix test for modules.linux_lvm.pvcreate
- **PR** #39048: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-01-31 15:55:49 UTC*
  - 88b171f863 Merge pull request #39048 from rallytime/merge-2016.11
  - b2b3989773 Merge branch '2016.3' into '2016.11'
    - a24af5ac46 Merge pull request #39047 from rallytime/merge-2016.3
      - b732a1f646 Merge branch '2015.8' into '2016.3'
      - 56ccae6ff7 Add 2015.8.14 release notes file (#39046)
      - 5943fe65d3 Update 2015.8.13 release notes (#39037)
    - fef1b1133d Add 2016.3.6 release notes file (#39045)
    - 7c43f4ac32 [2016.3] Update release numbers for doc build (#39042)
    - ff324599d5 Update 2016.3.5 release notes (#39038)
    - 5b09dc4198 Merge pull request #39028 from terminalmage/clarify-delimiter-argument
      - f29ef071f3 Clarify delimiter argument
    - 1ff359fa58 Add CLI Example for rest_sample_utils.get_test_string function (#39030)
    - f13fb9ef1e Enable __proxy__ availability in states, highstate, and utils. Enable __utils__ availability in proxies. (#38899)

  - **PR** #39035: (cro) Add CLI Examples so tests will pass
  - **PR** #39044: (rallytime) Add 2016.11.3 release notes file
  - **PR** #39040: (rallytime) [2016.11] Update release numbers for doc build
  - **PR** #39039: (rallytime) Update 2016.11.2 release notes
  - **PR** #39005: (cro) Ungate the status.py module and raise unsupported errors in functions not executable on Windows. (refs: #39536)
- **PR** #39012: (terminalmage) Fix "invalid lexer" errors in docs build @ *2017-01-28 06:47:45 UTC*
  - e70904c480 Merge pull request #39012 from terminalmage/invalid-lexer
  - 868001baac Fix "invalid lexer" errors in docs build
- **PR** #39003: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-01-28 00:09:09 UTC*
  - cea0f32936 Merge pull request #39003 from rallytime/merge-2016.11
  - 76e95087fd Merge branch '2016.3' into '2016.11'
  - da96221741 Merge pull request #38951 from DSRCorporation/bugs/37938_fix_depends_decorator_memleak
    - 0b18f34678 Keep the only one record per module-function in depends decorator.
  - 85165edb70 Merge pull request #38982 from rallytime/fix-34780
    - 1583c5579a Set response when using "GET" method in s3 utils
  - cfdbc99e12 Merge pull request #38989 from alprs/docfix-state_pt3_environ
    - 52a9ad1c60 fix SLS in environment variable examples
  - 55e4d2572e Merge pull request #39000 from rallytime/skip-badload-test
    - 4b3ff0fe0f Skip the test_badload test until Jenkins move is complete
  - fe054eb772 Merge pull request #38995 from terminalmage/fix-pillar.item-docstring
    - 06d094dd8f Fix pillar.item docstring
- **ISSUE** #38853: (bobrik) file.serialize still expects show_diff instead of show_changes (refs: #38908)
- **PR** #38908: (bobrik) Deprecate show_diff for file.serialize to mimic file.managed, closes #38853 @ *2017-01-27 17:15:37 UTC*
  - 58543d5cbf Merge pull request #38908 from bobrik/show-changes-for-serialize
  - e0af212c1b Remove unnecessary blank lines
  - a08c1ca530 Deprecate show_diff for file.serialize to mimic file.managed, closes #38853
- **ISSUE** saltstack/salt-bootstrap#1021: (sjorge) salt-bootstrap missing salt-api.xml on smartos (refs: #38978)
- **PR** #38978: (sjorge) fixes saltstack/salt-bootstrap#1021 @ *2017-01-27 17:05:10 UTC*
  - 4b75dfac95 Merge pull request #38978 from sjorge/2016.11-bootstrap
  - 26eb35f99d fixes salt/salt-bootstrap`#1021`_
- **PR** #38991: (isbm) Isbm zypper state unknown pkg crash @ *2017-01-27 16:59:38 UTC*
  - b40f369d98 Merge pull request #38991 from isbm/isbm-zypper-state-unknown-pkg-crash
  - 35f620e1c8 Prevent crash on unknown to the repo package
- **PR** #38979: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-01-26 22:56:13 UTC*
  - 3e76662166 Merge pull request #38979 from rallytime/merge-2016.11

  - fdaa5ac1b0 Merge branch '2016.3' into '2016.11'
  - b66b6f6423 Merge pull request #38950 from mbom2004/2016.3
    - c09f39d6c9 Remove unused json import
    - 249efa3068 Fixed Logstash Engine in file logstash.py
  - a6c6e47842 Handle changing "is_default" value in moto package for boto test mock (#38973)
  - b965b5dcc2 Merge pull request #38952 from terminalmage/zd1168
    - 6b014e53fc Rename on_demand_pillar to on_demand_ext_pillar
    - d216f90c63 Document new on_demand_pillar option and add to config template
    - 426b20f02f Add documentation for on-demand pillar to pillar.ext docstring
    - 7b10274b6b Make on-demand ext_pillars tunable
    - d54723ccae Add on_demand_pillar config option
  - 2c4ad85a78 Merge pull request #38948 from rallytime/bump-template-context-deprecation
    - 749e0031d7 Bump the template context deprecation version to Oxygen
  - e4514ca7d8 Merge pull request #38946 from rallytime/bp-37632
    - ee37cdace9 Fix some lint
    - c08071e182 Fix versions report for server OSs
  - 953a20350a Merge pull request #38913 from Adaephon-GH/patch-1
    - e2f4a16fdd Removing trailing whitespace
    - 616292c6b1 Ignore plist files without Label key
  - 826dce1059 Merge pull request #38917 from twangboy/update_jinja_mac
    - 62e608b627 Update Jinja2 to 2.9.4
  - b27733cc33 Merge pull request #38925 from terminalmage/issue38540
    - 76392fc6ad Fix traceback when a netapi module uses wheel_async
    - bd4474fa62 Fix 'success' value for wheel commands
  - 618596f0cc Merge pull request #38926 from gtmanfred/2016.3
    - 9cae953c93 add note about pysss for pam eauth
- **PR** #38937: (arthru) Fix smtp ret require gnupg @ *2017-01-26 20:08:16 UTC*
  - 0660cc3cf2 Merge pull request #38937 from HashBangDev/fix-smtp-ret-require-gnupg
  - 399556b9fe Remove trailing whitespace
  - f308d13a17 log an error on gnupg absence instead of raising an exception
  - 0427879d19 fails if gpgowner is set in smtp returner config but the installation lacks gnupg module
  - 27449c5a9b smtp returner does not require gnupg to be installed
- **ISSUE** #38816: (grichmond-salt) Errors in cloud runners are not reliably being captured as failures. (refs: #38955)
- **PR** #38955: (techhat) Do a better job at error detection in runners @ *2017-01-26 20:00:18 UTC*
  - d947c5c449 Merge pull request #38955 from techhat/issue38816

  - ea8654f400 Typo
  - 94050ff716 Watch out for bools
  - 0142b0bcb3 Do a better job at error detection in runners
- **PR** #38953: (thatch45) fix an issue where thorium would remove keys of reattaching minions @ *2017-01-26 19:15:59 UTC*
  - 04a5b05c36 Merge pull request #38953 from thatch45/thorium_keyfix
  - 68e96b11ac This is faster and cleaner
  - 13d28a34a6 fix an issue where thorium would remove keys of reattaching minions
  - **PR** #38972: (rallytime) Add CLI Example for rest_sample_utils.get_test_string function (refs: #39030)
- **PR** #38957: (mcalmer) Fix timezone handling for rpm installtime @ *2017-01-26 18:41:15 UTC*
  - 27166fad4e Merge pull request #38957 from mcalmer/fix-rpm-install_date-timezone
  - c7da9f87b6 Fix timezone handling for rpm installtime
- **PR** #38965: (toanju) salt-cloud will use list_floating_ips for OpenStack @ *2017-01-26 16:44:12 UTC*
  - **PR** #34280: (kevinanderson1) salt-cloud will use list_floating_ips for Openstack (refs: #38965)
  - ec690a0a12 Merge pull request #38965 from toanju/2016.11
  - 1253ce9b63 salt-cloud will use list_floating_ips for OpenStack
- **PR** #38949: (clinta) Use signing passphrase as public passphrase when generating self-sign… @ *2017-01-25 20:20:58 UTC*
  - d906e8fadb Merge pull request #38949 from clinta/x509-passphrase-bug
  - c8697e38a8 Use signing passphrase as public passphrase when generating self-signed certificates
- **PR** #38929: (MTecknology) Fix psutil regressions in 2016.11 @ *2017-01-25 20:17:41 UTC*
  - de3b2cc97b Merge pull request #38929 from MTecknology/2016.11
  - 73a8c6d121 Load core grains only if required.
  - 4966011cb5 Modules might still be needed, even if psutil loads.
  - fb0432fd21 Fixes a regression with old versions of python-psutil.
- **PR** #38940: (isbm) Isbm sanitizers fix and unit test @ *2017-01-25 20:15:56 UTC*
  - 3ec806c003 Merge pull request #38940 from isbm/isbm-sanitizers-fix-and-unit-test
  - a112b790fe Fix typo
  - 47a16916c3 Add unit test
  - 046c5436eb Fix leading dots on sanitized hostname
- **PR** #38944: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-01-25 19:44:42 UTC*
  - e420763285 Merge pull request #38944 from rallytime/merge-2016.11
  - ee33a53a64 Merge branch '2016.3' into '2016.11'
    - 405d86a2ca Merge pull request #38847 from terminalmage/issue38825
      - 11a47803ce Use log.exception() instead
      - e40fac589a Catch MinionError in file.source_list
    - b5df104fc2 Merge pull request #38875 from terminalmage/issue36121

      - fbc4d2a2c4 reactor: ensure glob_ref is a string
      - 2e443d79a3 cp.cache_file: add note re: return for nonexistant salt:// path
    - e9ebec4d80 Merge pull request #38890 from cro/vmware_reset_vm_20163
      - 0146562fb4 Call correct function for resetting a VM
    - c3fbfcd231 Merge pull request #38883 from techhat/dontrequire
      - 67bc4d6687 Don't require text_out path to exist
    - 6430a45196 Merge pull request #38851 from terminalmage/docker-py-2.0
      - 3c061b21fe Support docker-py 2.0 in dockerng
    - ac8008d843 Merge pull request #38844 from cachedout/http_memory_leak
      - c46bf85518 Fix memory leak in HTTP client
    - dfe6dfe963 Merge pull request #38823 from gtmanfred/2016.3
      - f0a71e8707 pass pillar to compound matcher in match module
    - a04ab86da1 Merge pull request #38833 from Ch3LL/add_release_notes_2016.3.5
    - 374dc1ab88 skip 2016.3.5 due to :doc: references
    - 31f324c4ff add 2016.3.5 changelog to release notes

- **ISSUE** #38753: (alexbleotu) __*proxy*__ dunder is not injected when invoking the *salt* variable in sls files (refs: #38899, #38900, #38829)
- **ISSUE** #38557: (alexbleotu) Proxy not working on develop (refs: #38829)
- **ISSUE** #38265: (mirceaulinic) __*utils*__ object not available in proxy module (refs: #38899, #38900, #38829)
- **ISSUE** #32918: (mirceaulinic) Proxy minions reconnection (refs: #38829)
- **PR** #38900: (cro) Enable __proxy__ availability in states, highstate, and utils. Enable __utils__ for proxies. @ *2017-01-25 19:36:48 UTC*
  - **PR** #38899: (cro) Enable __proxy__ availability in states, highstate, and utils. Enable __utils__ for proxies. (refs: #38900)
  - **PR** #38829: (cro) MANY dunder variable fixes for proxies + proxy keepalive from @mirceaulinic (refs: #38899, #38900)
  - **PR** #37864: (mirceaulinic) Proxy keepalive feature (refs: #38829)
  - bd4889ac73 Merge pull request #38900 from cro/px_dunder_201611
  - 9a86fddfa1 Remove extra call to salt.loader.utils.
  - f4ba89735c Resolve merge conflict
- **PR** #38918: (thatch45) Thorium typos @ *2017-01-25 19:00:40 UTC*
  - f94b8798b6 Merge pull request #38918 from thatch45/thorium_typos
  - 0b4aca9145 fix some minor typos in the thorium docs
  - 58a18e2b58 Add test= True to the master so that thorium does not stack trace
- **ISSUE** #38543: (amendlik) salt –subset returns wrong number of minions (refs: #38919)
- **PR** #38919: (cachedout) Correctly pass subset to cmd_subset @ *2017-01-25 18:59:16 UTC*
  - 32fbb945b7 Merge pull request #38919 from cachedout/issue_38543

  - a555de7c56 Correctly pass subset to cmd_subset
- **PR** #38922: (twangboy) Fix 64bit detection, vcredist only on <= 2008 @ *2017-01-25 18:47:41 UTC*
  - 6b3c738bfd Merge pull request #38922 from twangboy/fix_vcredist
  - 214e1cc598 Fix 64bit detection, vcredist only on <= 2008
- **ISSUE** #38371: (syphernl) [2016.11.1] Scheduled highstates not returning to master (refs: #38923)
- **PR** #38923: (DmitryKuzmenko) Fixed broken __schedule_return handler. @ *2017-01-25 18:45:30 UTC*
  - **PR** #36202: (hu-dabao) for 36049, log current connected master and make status module more useful and efficient (refs: #38923)
  - 954658523b Merge pull request #38923 from DSRCorporation/bugs/38371_fix_schedule_return
  - b18f675486 Fixed broken __schedule_return handler.
- **PR** #38927: (l2ol33rt) Adding explicit install of python-systemd in jessie-backports on Debian Guide @ *2017-01-25 18:21:18 UTC*
  - 828e9bd8f9 Merge pull request #38927 from l2ol33rt/debian_doc_fix
  - 9cc9c6110d Adding explicit call to python-systemd in jessie-backports
- **ISSUE** #37413: (Snarfingcode666) Salt-cloud vmware missing reboot command (refs: #38890, #38887, #38889)
- **PR** #38889: (cro) Backport #38887 to 2016.11: Call correct function for resetting a VM @ *2017-01-24 15:20:29 UTC*
  - **PR** #38887: (cro) Enable resetting a VM via salt-cloud & VMware driver (refs: #38890, #38889)
  - 5ff5e97598 Merge pull request #38889 from cro/vmware_reset_vm_201611
  - 76a9920a6b Call correct function for resetting a VM
- **PR** #38891: (UtahDave) Proper function parameter default @ *2017-01-24 15:06:09 UTC*
  - 53d0aa8855 Merge pull request #38891 from UtahDave/fix_cassandra_protocol_version
  - c475609683 Proper function parameter default
- **PR** #38904: (terminalmage) Add top file merging docs to the master config file documentation @ *2017-01-24 14:59:26 UTC*
  - c680ee3174 Merge pull request #38904 from terminalmage/docs
  - 42a3652620 Add top file merging docs to the master config file documentation
- **PR** #38885: (meaksh) Increasing timeouts for running integrations tests @ *2017-01-23 18:59:50 UTC*
  - 41a3055213 Merge pull request #38885 from meaksh/2016.11-fix-tests-issues
  - 4311b0b6de Increasing timeouts for running integrations tests
- **PR** #38639: (isbm) Isbm disable custom roster for api 2016.11 @ *2017-01-23 18:59:11 UTC*
  - bde6d3eee7 Merge pull request #38639 from isbm/isbm-disable-custom-roster-for-api-2016.11
  - ffbd45062e Explain what it is about and how to configure that

28.2.35 Salt 2016.11.4 Release Notes

Version 2016.11.4 is a bugfix release for *2016.11.0*.

Statistics

- Total Merges: **276**
- Total Issue References: **63**
- Total PR References: **223**
- Contributors: **62** (Ch3LL, DennisHarper, DmitryKuzmenko, L4rS6, MasterNayru, Seb-Solon, The-Loeki, UtahDave, aabognah, alankrita, amontalban, ardakuyumcu, attiasr, bdrung, bewing, cachedout, cro, defanator, discountbin, dmurphy18, drawsmcgraw, eldadru, garethgreenaway, githubcdr, gtmanfred, hkrist, isbm, jbadson, jeanpralo, jettero, jinm, joe-niland, kaszuba, lomeroe, lorengordon, mateiw, mcalmer, mchugh19, meaksh, mirceaulinic, mlalpho, narendraingale2, nmadhok, rallytime, redbaron4, roaldnefs, s0undt3ch, skazi0, skizunov, smarsching, sofixa, sp1r, sthrasher, techhat, terminalmage, thatch45, thor, ticosax, twangboy, vutny, whiteinge, zer0def)

AIX Support Expanded

AIX support has been added for the following execution modules:

- *`user`*
- *`group`*
- *`network`*
- *`status`*
- *`timezone`*

Additionally, AIX is now supported in the *`disk.iostat`* remote-execution function, and the *`status`* beacon is now supported.

Minion Data Cache Enhancement

Memcache is now supported as a data store for the minion data cache.

Memcache is an additional cache layer that keeps a limited amount of data fetched from the minion data cache for a limited period of time in memory that makes cache operations faster. It doesn't make much sense for the `localfs` cache driver but helps for more complex drivers like `consul`.

For more details see `memcache_expire_seconds` and other `memcache_*` options in the master config reverence.

Docker Fixes

- Docker authentication has been re-organized. Instead of attempting a login for each push/pull (which was unnecessary), a new function called `dockerng.login` has been added, which authenticates to the registry and adds the credential token to the `~/.docker/config.json`. After upgrading, if you have not already performed a `docker login` on the minion using the docker CLI, you will need to run `dockerng.login` to login. This only needs to be done once.
- A bug in resolving the tag name for images in a custom registry (where a colon can appear in the image name, e.g. `myregistry.com:5000/image:tagname`) has been fixed. In previous releases, Salt would use the colon to separate the tag name from the image name, and if there was no colon, the default tag name of `latest` would be assumed. However, this caused custom registry images to be misidentified when no explicit tag name was passed (e.g. `myregistry.com:5000/image`). To work around this in earlier releases, simply specify the tag name.

Salt-Cloud Fixes

2016.11.0 added support for templating userdata files for the `ec2` driver, using the `renderer` option from the master config file. However, as the default renderer first evaluates jinja templating, followed by loading the data as a YAML dictionary, this results in unpredictable results when userdata files are comprised of non-YAML data (which they generally are).

2016.11.4 fixes this by only templating the userdata_file when it is explicitly configured to do so. This is done by adding a new optional parameter to the cloud profile called `userdata_template`. This option is used in the same way as the `template` argument in `file.managed` states, it is simply set to the desired templating renderer:

```
my-ec2-config:
  # Pass userdata to the instance to be created
  userdata_file: /etc/salt/my-userdata-file
  userdata_template: jinja
```

If no `userdata_template` option is set in the cloud profile, then salt-cloud will check for the presence of the master configuration parameter `userdata_renderer`. If this is also not set, then no templating will be performed on the userdata_file.

In addition, the other cloud drivers which support setting a `userdata_file` (`azurearm`, `nova`, and `openstack`) have had templating support added to bring them to feature parity with the ec2 driver's implementation of the `userdata_file` option.

Changelog for v2016.11.3..v2016.11.4

Generated at: 2018-05-27 19:46:47 UTC

- **PR** #40708: (Ch3LL) Add 2016.11.4 Release Note ChangeLog @ *2017-04-14 22:12:57 UTC*
  - e5cd6086a7 Merge pull request #40708 from Ch3LL/2016.11.4_release
  - d228fb6e02 Add 2016.11.4 Release Note ChangeLog
- **PR** #40685: (Ch3LL) Fix errno code for filecache test for other operating systems. @ *2017-04-14 16:54:25 UTC*
  - 77028a6c4e Merge pull request #40685 from Ch3LL/fix_mac_file
  - 9ea6e8b456 remove io and change to EROFS
  - 688791ff60 remove try-except and change errno
  - e30afc4c01 add exception type
  - acf333df08 change errno code for fileclient test
- **ISSUE** #40688: (jbadson) Syslog returner does not work with Python 2.6 (refs: #40689)
- **PR** #40689: (jbadson) Fixes bug that prevents syslog returner from working under Python 2.6 @ *2017-04-14 10:45:13 UTC*
  - bc70772f9d Merge pull request #40689 from jbadson/fix-syslog-returner
  - e5a3a7d217 Fixes bug that prevents syslog returner from working under Python 2.6
- **ISSUE** #40658: (sebw) State tomcat.war_deployed regression when WAR filename contains version (refs: #40690)
- **PR** #40690: (thor) Fixes #40658: even clearer and working(!) Tomcat version handling @ *2017-04-14 10:44:02 UTC*
  - 983d35ad38 Merge pull request #40690 from thor/2016.11-tomcat

  - 09145ea1a5 Fixes unindexed strfmt curly braces for python 2.6
  - b78fc46b91 Fixes #40658: clearer version handling
- **PR** #40686: (twangboy) Fix 'salt-minion' service for Win 10 Creators Update 1703 @ *2017-04-13 20:00:12 UTC*
  - 3cd9a50b22 Merge pull request #40686 from twangboy/fix_service
  - b6ac4aa86d Fix service for win10 update
- **PR** #40675: (gtmanfred) use loader for getting war version @ *2017-04-13 19:58:30 UTC*
  - ad4d6839fd Merge pull request #40675 from gtmanfred/2016.11
  - a61fc824c4 use loader for war extraction
- **ISSUE** #38497: (chrisLeeTW) local_batch client ignore external auth (refs: #40598)
- **PR** #40680: (rallytime) Back-port #40598 to 2016.11 @ *2017-04-13 19:58:16 UTC*
  - **PR** #40598: (mchugh19) Ensure batch uses passed eauth token or credentials (refs: #40680)
  - 7ea526f59e Merge pull request #40680 from rallytime/bp-40598
  - cc1643eb1f Fix netapi lint
  - e790930f5a re-add batch support to cherrypy saltapi
  - 6eec04b2db pop out of kwargs
  - 260dd84758 Create eauth dict for passing into batch class
  - 5fb8190d44 Ensure batch uses passed eauth token or credentials
- **PR** #40681: (cachedout) Allow status beacon to run on all operating systems @ *2017-04-13 19:33:10 UTC*
  - db68df23dd Merge pull request #40681 from cachedout/status_beacon
  - ecbb0d186f Allow status beacon to run on all operating systems
- **PR** #40678: (Ch3LL) fix test_fstype test for mac @ *2017-04-13 19:20:32 UTC*
  - 39dd6e284d Merge pull request #40678 from Ch3LL/fix_mac_fstype
  - 60724980ec fix test_fstype test for mac
- **PR** #40665: (rallytime) Back-port #35665 to 2016.11 @ *2017-04-12 21:06:36 UTC*
  - **PR** #35665: (sthrasher) Speed up /jobs for salt-api when run under cherrypy. (refs: #40665)
  - 6df76f6687 Merge pull request #40665 from rallytime/bp-35665
  - 0f897b2426 Switch from comprehension to logic used in jobs runner. This makes it easier to deal with potential unicode in returns.
  - 78dd629f09 Fix compat issues with /jobs return values.
  - 4778bc7365 Speed up /jobs for salt-api when run under cherrypy.
- **PR** #40666: (gtmanfred) make sure userdata is always defined in ec2 @ *2017-04-12 21:06:00 UTC*
  - 3e41a248a5 Merge pull request #40666 from gtmanfred/userdata
  - 5e92fd0948 make sure userdata is always defined in ec2
- **PR** #40662: (twangboy) Backport msi-conformant-version function @ *2017-04-12 18:49:23 UTC*
  - b245abbea5 Merge pull request #40662 from twangboy/backport_msi_versioning
  - 825832812b Backport msi-conformant-version function

- **ISSUE** #39868: (amontalban) archive.extracted issue when source_hash_update=True and extracted files does not exist (refs: #40551)
- **PR** #40551: (terminalmage) Fix four issues in archive.extracted state @ *2017-04-12 18:37:52 UTC*
  - 92b5f03beb Merge pull request #40551 from terminalmage/issue39868
  - a722ca9ccf archive.extracted: also cleanup fileclient's cached location
  - 5ea1f607b0 Fix mocking in unit tests
  - 8dfa51f31f Moar fixes for source_hash_update
  - 7103707d49 Remove unnecessary versionadded lines
  - a717881f53 Just get a hash for the source archive
  - 9da4eb18bf Check hash of cached source against source_hash before downloading archive
  - ad24faa59d Fix three issues in archive.extracted state
- **PR** #40637: (twangboy) Add unicode_literals import @ *2017-04-12 16:55:03 UTC*
  - 0638418d22 Merge pull request #40637 from twangboy/fix_unicode_issues
  - 021783dbae Add unicode_literals import
- **PR** #40651: (twangboy) Fix status.diskusage for Windows on Py3 @ *2017-04-12 16:21:29 UTC*
  - 491661f323 Merge pull request #40651 from twangboy/fix_diskusage_py3
  - 7c5079ec91 Correct capitalization problem with api call
- **ISSUE** #40624: (sumeetisp) Issue - grains.append (refs: #40631)
- **PR** #40631: (gtmanfred) if grain is defined as None still convert in append @ *2017-04-12 16:19:16 UTC*
  - 3aabd85e53 Merge pull request #40631 from gtmanfred/grains
  - b0bd99c26d add comment and unit test
  - b21bc7528f if grain is defined as None still convert in append
- **ISSUE** #40167: (alias454) file.replace diff results output showing additional characters (refs: #40629)
- **PR** #40629: (aabognah) Fixing issue # 40167 @ *2017-04-11 22:45:08 UTC*
  - 3737289bee Merge pull request #40629 from aabognah/fix-bug-40167
  - 28f7744cb6 Fixing issue # 40167 with file.replace where the diff output does not display correctly.
- **PR** #40646: (twangboy) Keep network.py execution module @ *2017-04-11 22:03:02 UTC*
  - 2a22bea290 Merge pull request #40646 from twangboy/fix_win_network
  - 0f7a81cd34 Keep network.py execution module
- **PR** #40645: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-04-11 20:59:13 UTC*
  - e1f5a5dfc3 Merge pull request #40645 from rallytime/merge-2016.11
  - 8de6497933 Merge branch '2016.3' into '2016.11'
    - 2ae9eaa176 Merge pull request #40638 from rallytime/bp-40571
      - 2d1c4be2df pkgrepo.managed: properly handle comments for debian
- **ISSUE** #40594: (anlutro) salt-ssh file.recurse adds a lot of unwanted directories (refs: #40642)
- **ISSUE** #38458: (duk3luk3) salt-ssh uses sudo to create cache dir, later fails to access it (refs: #40442)

- **PR** #40642: (DmitryKuzmenko) Correctly resolve relative cache path to absolute. @ *2017-04-11 20:43:57 UTC*
  - **PR** #40442: (gtmanfred) allow file_client to figure out cachedir (refs: #40642)
  - 6c4ae3c914 Merge pull request #40642 from DSRCorporation/bugs/40594_ssh_cachedir
  - 055256c518 Correctly resolve relative cache path to absolute.
- **ISSUE** #40075: (afletch) salt-ssh temporary files - insecure permissions (refs: #40609)
- **PR** #40609: (gtmanfred) stat_file when keep is set, instead of mirroring all file permissions @ *2017-04-11 18:48:47 UTC*
  - 8492cef7a5 Merge pull request #40609 from gtmanfred/2016.11
  - 6e34c2b5e5 stat file when placing it on server instead of caching
- **PR** #40620: (mateiw) SUSE specific changes to salt-api.service @ *2017-04-11 14:45:00 UTC*
  - 05ac613ecf Merge pull request #40620 from mateiw/2016.11-suse-saltapi-service
  - ee911a74b4 suse specific changes to salt-api.service
- **ISSUE** #39463: (githubcdr) Transport TCP minions don't reconnect/recover (refs: #40614)
- **PR** #40614: (gtmanfred) add retries on authentications of the salt minion reconnecting @ *2017-04-10 22:42:16 UTC*
  - b0a2414d68 Merge pull request #40614 from gtmanfred/tcp
  - a86b101ae6 add retries on authentications of the salt minion reconnecting
- **PR** #40606: (kaszuba) Use correct exec_driver in dockerng.sls module @ *2017-04-10 22:25:31 UTC*
  - f7e121a9ee Merge pull request #40606 from kaszuba/fix-dockerng-sls
  - 3a0d61f108 Use correct exec_driver in dockerng.sls module
- **ISSUE** #39863: (daswathn) Salt-Master not responding when the list of minions are high after upgrade to 2016.11.2 (refs: #40615)
- **PR** #40615: (rallytime) Call out to _pki_minions() once, rather than in a loop in _check_list_minions() @ *2017-04-10 22:22:18 UTC*
  - **PR** #34920: (cachedout) Key cache (refs: #40615)
  - b6cf948afe Merge pull request #40615 from rallytime/fix-39863
  - 1a9f03ab92 Call out to _pki_minions() once, rather than in a loop in _check_list_minions()
- **PR** #40588: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-04-07 19:30:14 UTC*
  - 4fa58be222 Merge pull request #40588 from rallytime/merge-2016.11
  - 5a419b8aae Merge branch '2016.3' into '2016.11'
  - 83f6d3d3bb Merge pull request #40567 from terminalmage/fix-pillar-get-merge-lists
    - cb4db56eb5 Allow pillar.get to merge list as well as dictionaries
  - a8304cd5a1 Merge pull request #40552 from terminalmage/fix-hash-type-refs
    - 8c61f333ae Don't use __opts__.get() for hash_type
  - 705e1d8a08 Merge pull request #40562 from terminalmage/fix-get-client
    - 7f1ef72f83 Fix dockerng _get_client() regression
  - 00f8ef0c55 Merge pull request #40548 from Ch3LL/fix_vultrpy

    - 7710355e3a check for salt install fail on vultur test
    - aae3d14ea4 fix vultr cloud race condition to match on 0*

- **PR** #40575: (rallytime) Back-port #40559 to 2016.11 @ *2017-04-07 15:42:26 UTC*
  - **PR** #40559: (jinm) Fix v3 for https://github.com/saltstack/salt/issues/38472 (refs: #40575)
  - 3d07f637ca Merge pull request #40575 from rallytime/bp-40559
  - 8280e5256e Fix v3 for https://github.com/saltstack/salt/issues/38472
- **PR** #40576: (rallytime) Back-port #40573 to 2016.11 @ *2017-04-07 15:20:11 UTC*
  - **PR** #40573: (ardakuyumcu) Fix typo in IAM state for managed policies (refs: #40576)
  - 9041ca2ba5 Merge pull request #40576 from rallytime/bp-40573
  - 12180808ee Fix typo in IAM state for managed policies
- **PR** #40563: (terminalmage) Merge-forward 2016.3 -> 2016.11 @ *2017-04-07 15:08:20 UTC*
  - **PR** #40562: (terminalmage) Fix dockerng _get_client() regression (refs: #40563)
  - **PR** #40481: (terminalmage) Backport auth and custom registry fixes from #40480 to 2016.3 branch (refs: #40563, #40562)
  - **PR** #40480: (terminalmage) Improved Docker auth handling and other misc. Docker improvements (refs: #40481)
  - f8bc423ef9 Merge pull request #40563 from terminalmage/merge-2016.3-2016.11
  - 0c608d7417 Add client_args_mock back to test
  - a7a78da984 remove unused imports
  - a6d68f50fe Merge remote-tracking branch 'upstream/2016.3' into merge-2016.3-2016.11
    - 0918311330 Don't mark files that already were deleted as errors
    - 51d88a16c8 Merge branch 'zer0def-fix-31363' into 2016.3
      - 7f3cbd5cf9 Merge branch 'fix-31363' of https://github.com/zer0def/salt into zer0def-fix-31363
      - 3c750c2b24 Changed rm_rf's argument to actually remove intended file. (refs #31363)
      - 9ed85f3c59 Remove directory content instead of directory itself when using *force_clone* in *git.latest* state. (refs #31363)
    - cfba4cb422 Merge pull request #40534 from terminalmage/issue39892
      - ad88c58a09 Check master's ssh_minion_opts for fileserver/pillar values and ignore them
    - 8da27c9e1d Merge pull request #40306 from terminalmage/issue40279
      - 57ace1f336 Merge branch 'issue40279' of https://github.com/terminalmage/salt into issue40279
      - 8bcdf1a761 Remove unused import for lint
      - 808ad76419 systemd.py: when getting all services, don't repeat gathering of systemd services
      - 2d219af67a Don't use context caching for gathering systemd services
    - 97caac4c0a Merge pull request #40481 from terminalmage/docker-auth-handling-2016.3
      - dcef1e0d4b Make sure we keep the cached client when clearing context

      - 1e2a04cfc5 Backport auth and custom registry fixes from #40480 to 2016.3 branch
    - e62603d5eb Merge pull request #40505 from gtmanfred/2016.3
      - 6e2f9080ca update docs for logging handlers

- **ISSUE** #39778: (Talkless) pkgrepo.managed state always report changes with test=True on APT system (refs: #40571)
- **PR** #40571: (terminalmage) pkgrepo.managed: properly handle comments for debian (refs: #40638) @ *2017-04-06 21:55:46 UTC*
  - fd757fffa3 Merge pull request #40571 from terminalmage/issue39778
  - 191610482d pkgrepo.managed: properly handle comments for debian
- **ISSUE** #40278: (UtahDave) cloud.action giving errors on 2016.11.1 (refs: #40572)
- **PR** #40572: (rallytime) Clean out kwargs dict in cloud.action before calling cloud driver function @ *2017-04-06 21:53:40 UTC*
  - b1698e830e Merge pull request #40572 from rallytime/fix-40278
  - c978486452 Clean out kwargs dict in cloud.action before calling cloud driver function
- **ISSUE** #39842: (smarsching) File module removes trailing newline on Windows (refs: #39882)
- **PR** #39882: (smarsching) Fix handling of trailing newlines on Windows @ *2017-04-06 21:12:24 UTC*
  - 62d8ad2b4b Merge pull request #39882 from smarsching/issue-39842
  - d485d1af44 Fix context for _splitlines_preserving_trailing_newline.
  - 76cb7bf612 Fix trailing newlines on Windows (#39842).
- **PR** #40451: (isbm) Fileclient testcase (2016.11) @ *2017-04-06 19:53:31 UTC*
  - ae13de622a Merge pull request #40451 from isbm/isbm-fileclient-testcase-2016.11
  - 74c65557dd Add space before in-lint comment for lint
  - 35fcb8b52d Fix race condition on cache directory creation
  - aba94495a5 Lintfix (Py3 code compat)
  - 9f9dc6e4e7 Add unit test case for fileclient
- **ISSUE** #40084: (podstava) profile fields in azurearm salt-cloud need to be actualized to sources (refs: #40564)
- **PR** #40564: (techhat) Update Azure ARM docs @ *2017-04-06 18:17:32 UTC*
  - 74366c57a4 Merge pull request #40564 from techhat/azuredocs
  - 08d071bc68 Update Azure ARM docs
- **ISSUE** #40005: (vutny) *ssh_known_hosts.present* does not support SHA256 key fingerprints (refs: #40543)
- **PR** #40543: (rallytime) Add the "fingerprint_hash_type" option to ssh state and module @ *2017-04-05 21:21:16 UTC*
  - cb9dcb1e1b Merge pull request #40543 from rallytime/fix-40005
  - 1ef81e6a55 Add the "fingerprint_hash_type" option to ssh state and module
- **PR** #40540: (DmitryKuzmenko) A quick fix for Cache has no 'list' attribute. @ *2017-04-05 18:50:18 UTC*
  - **PR** #40494: (rallytime) [develop] Merge forward from 2016.11 to develop (refs: #40540)
  - 3f0695575a Merge pull request #40540 from DSRCorporation/bugs/40494_merge_forward_cache_list_fix

  - c0fd5634cf A quick fix for Cache has no 'list' attribute.
- **ISSUE** #32662: (anlutro) salt-cloud: allow templating of EC2 userdata, similar to deploy script (refs: #32698)
- **PR** #40464: (terminalmage) salt-cloud: Do not pass userdata_file through yaml renderer @ *2017-04-05 17:32:07 UTC*
  - **PR** #32698: (techhat) Allow EC2 userdata to be templated (refs: #40464)
  - 28fc048030 Merge pull request #40464 from terminalmage/userdata-renderer
  - 84ee693006 Nova and openstack don't accept base64-encoded userdata
  - 73f4c43e2a Allow for userdata_template to be disabled in a cloud_profile
  - 78b4798b1b Update compile_template test to use StringIO
  - 5f7c5613ce Properly handle renderers which return StringIO objects
  - d551b0d857 Bring in salt.utils.stringio from develop branch
  - 6a6ef0adf8 Move userdata templating to salt.utils.cloud
  - b440d0c679 Update 2016.11.4 release notes for userdata_renderer -> userdata_template
  - a6183d93d3 Preserve windows newlines in salt.template.compile_template()
  - 04f02df5fe Try to read compiled template as StringIO
  - 79cc253bbf Only template the userdata_file if explicitly configured to do so
  - b580654f85 Update cloud docs to reflect userdata_renderer -> userdata_template
  - a6064fb2e4 Rename userdata_renderer -> userdata_template in master config docs
  - 50f2b2831f Remove userdata_renderer value
  - cc2186f35a Add templating support for other cloud drivers that support userdata_file
  - be8d34c59b ec2: Add support for using userdata_renderer to template userdata_file
  - eddbd41265 Openstack did not have templating support for userdata_file before 2016.11.4
  - a85a416c72 Add userdata_renderer fix info to 2016.11.4 release notes
  - 111188742a Add documentation for userdata_renderer
  - 9ee2dcfc2d Add userdata_renderer master config param
- **PR** #40530: (dmurphy18) Update release information for 2016.11.4 for additional AIX support @ *2017-04-05 16:20:22 UTC*
  - 990bde4c07 Merge pull request #40530 from dmurphy18/aix_docupd
  - fd93caf206 Added further support for functionality on AIX for 2016.11.4
  - 17b58917f2 Update release information for new AIX support
- **PR** #40528: (dmurphy18) Allow for nightly build designations in Salt versions @ *2017-04-04 20:34:26 UTC*
  - 4d932691f1 Merge pull request #40528 from dmurphy18/salt_nightlybuild
  - d62a119fc1 Allow for nightly build designations in Salt versions
- **ISSUE** #37699: (gstachowiak) Artifactory state. Incorrect timeout error reporting. (refs: #40465)
- **PR** #40465: (rallytime) Artifactory Execution & State Module: Fixup Error Handling @ *2017-04-04 20:12:21 UTC*
  - 0ed385210f Merge pull request #40465 from rallytime/fix-37699

  - 8f084f7056 Update unit test to look for actual string comment
  - ef664b46ae Artifactory State: Only wrap main function call to module in try/except and wrap exc comment in str()
  - f1015e3900 Artifactory Module: catch URLErrors as well as HTTPErrors
- **ISSUE** #39275: (yhekma) Cache backend gets hit a *lot* (refs: #40497, #40429)
- **PR** #40497: (DmitryKuzmenko) Memcache documentation and minor updates. @ *2017-04-04 19:55:18 UTC*
  - **PR** #40429: (DmitryKuzmenko) MemCache - a minion data cache booster. (refs: #40497, #40468)
  - 7a04ed2439 Merge pull request #40497 from DSRCorporation/features/39275_memcache
  - 82c45b1a52 Memcache documentation and minor updates.
- **ISSUE** #38683: (gstachowiak) require/order/failhard combination error (refs: #40504)
- **PR** #40504: (rallytime) Group checks for failhard setting in () in state.check_failhard function @ *2017-04-04 19:53:48 UTC*
  - d654de52ed Merge pull request #40504 from rallytime/fix-38683
  - ede4c28887 Group checks for failhard setting in () in state.check_failhard function
- **PR** #40503: (thatch45) first pass at adding support for pycryptodome installed as @ *2017-04-04 19:39:02 UTC*
  - 4d5d7d9712 Merge pull request #40503 from thatch45/2016.11
  - e21fd54d1b fix lint on the lint ignores…
  - 60113248b1 pycryptodome adds RSA to the key header which the openssl
  - 206dec63ff fix the cryptodome version lookup for the versions report
  - d3b77092b5 good catch
  - 31c6a10d1b first pass at adding support for pycryptodome installed as
- **PR** #40525: (dmurphy18) Add support for disk.iostat on AIX @ *2017-04-04 19:31:41 UTC*
  - 0dd92c63ea Merge pull request #40525 from dmurphy18/aix_dskiostat
  - 712537272b Added support on AIX for disk.iostat
- **PR** #40496: (rallytime) Back-port #40415 to 2016.11 @ *2017-04-04 17:19:39 UTC*
  - **PR** #40415: (defanator) Fix boto_vpc.create_route() to work with interface_id (refs: #40496)
  - a6291b17c1 Merge pull request #40496 from rallytime/bp-40415
  - f8b3006898 Fix boto_vpc.create_route() to work with interface_id
- **ISSUE** #39275: (yhekma) Cache backend gets hit a *lot* (refs: #40497, #40429)
- **PR** #40468: (techhat) Add __func_alias__ back in @ *2017-04-04 17:02:43 UTC*
  - **PR** #40429: (DmitryKuzmenko) MemCache - a minion data cache booster. (refs: #40497, #40468)
  - 3eb8e0baf1 Merge pull request #40468 from techhat/cachealias
  - 6ec0baa9a0 Swap around aliases
  - 76e54a2900 Add __func_alias__ back in
- **ISSUE** #29104: (adithep) Merging Order warning (refs: #39109)
- **PR** #39109: (bdrung) Fix top_file_merging_strategy warning if env_order is set @ *2017-04-04 14:20:56 UTC*
  - 8c0befaa8b Merge pull request #39109 from bdrung/fix-merge-order-warning

  - fbf8fcfa98 Simplify _get_envs() by using list comprehensions
  - 74a3b066ea Fix top_file_merging_strategy warning if env_order is set
  - ec219b5f42 Remove duplicate client_envs variable definitions
  - 6279f7c120 fix do to pre correct on python randome function
  - 66b9515af7 Fix up the doc for failover clarity
- **PR** #40495: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-04-03 18:36:16 UTC*
  - 02a1f642ab Merge pull request #40495 from rallytime/merge-2016.11
  - 8111909bb1 Merge branch '2016.3' into '2016.11'
  - 3d45a004b0 Merge pull request #40427 from terminalmage/clarify-master-tops-docs
    - bda781d8f9 Grammar fix
    - 0d7b0c4ef0 Improve the master_tops documentation
    - d27340a9f2 Add saltutil.sync_tops runner func
- **PR** #40466: (dmurphy18) Support for execution module status on AIX @ *2017-04-01 00:28:51 UTC*
  - ac82972cb3 Merge pull request #40466 from dmurphy18/aix_status
  - 7c0b30d9a4 Support for AIX
- **ISSUE** #39275: (yhekma) Cache backend gets hit a *lot* (refs: #40497, #40429)
- **PR** #40429: (DmitryKuzmenko) MemCache - a minion data cache booster. (refs: #40497, #40468) @ *2017-03-31 20:21:00 UTC*
  - fdb0250c95 Merge pull request #40429 from DSRCorporation/features/39275_memcache
  - 4475d1757d In-memory minion data cache.
- **ISSUE** #38458: (duk3luk3) salt-ssh uses sudo to create cache dir, later fails to access it (refs: #40442)
- **PR** #40442: (gtmanfred) allow file_client to figure out cachedir (refs: #40642) @ *2017-03-31 20:14:27 UTC*
  - 31d4e6949c Merge pull request #40442 from gtmanfred/salt-ssh
  - 8367735063 allow file_client to figure out cachedir
- **PR** #40456: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-03-31 17:51:11 UTC*
  - 0cfcd188a9 Merge pull request #40456 from rallytime/merge-2016.11
  - 0da4c46b68 Merge branch '2016.3' into '2016.11'
  - c26f4cc76c Merge pull request #40371 from terminalmage/pr-40344
    - a8bcaa73d7 Force use of posixpath when joining salt fileserver paths in gitfs
    - cafa08d8e0 Add ability for salt.utils.path_join to force the use of posixpath
  - df9df82959 Merge pull request #40379 from rallytime/tests-for-39855
    - 96259d6c63 Lint fix
    - 4f7ac1431e Create a unit test for the _replace_auth_key function in the ssh module
- **PR** #40443: (gtmanfred) prepend ssh_log_file with root_dir @ *2017-03-31 09:23:46 UTC*
  - 8617be9c6d Merge pull request #40443 from gtmanfred/sshlog
  - 7f6046deec prepend ssh_log_file with root_dir

- **PR** #40376: (nmadhok) Backporting changes in vmware cloud driver from develop branch to 2016.11 branch @ *2017-03-30 22:35:13 UTC*
  - 132d8b7b88 Merge pull request #40376 from nmadhok/2016.11
  - dd62310941 Adding unit tests for vmware_test
  - 36edf0af64 Add additional VMware related exceptions
  - 034ef30f7c Remove old vmware unit tests
  - 7c144888da Backporting changes in vmware cloud driver from develop branch to 2016.11 branch
- **ISSUE** #39692: (djsly) tuned module and state are broken on 7.3 families. (refs: #40387, #39719, #39768)
- **PR** #40387: (redbaron4) More complete fix for 39692 @ *2017-03-30 22:29:05 UTC*
  - dfaa670b66 Merge pull request #40387 from redbaron4/fix-39692
  - 77a40a0c44 Lint fixes
  - 8c1adfafd5 More complete fix for 39692
- **ISSUE** #7287: (dragozov) django.loaddata treats fixture list as arguments and prepends "–" for each (refs: #40404)
- **PR** #40404: (roaldnefs) Fix for fixtures in the djangomod module @ *2017-03-30 22:26:09 UTC*
  - 313d21626f Merge pull request #40404 from roaldnefs/fix-djangomod-loaddata
  - 92285cb045 Fix for fixtures in the djangomod module
- **PR** #40416: (lorengordon) Adds some missing file functions on Windows @ *2017-03-30 22:22:44 UTC*
  - 5379899442 Merge pull request #40416 from lorengordon/win-file-funcs
  - 8edaf25e10 Adds some missing file functions on Windows
- **ISSUE** #40417: (lorengordon) *temp.file* does not close the file handle (refs: #40418)
- **PR** #40418: (lorengordon) Closes handle to temporary file before returning the path @ *2017-03-30 22:22:03 UTC*
  - 1f5d6b88f9 Merge pull request #40418 from lorengordon/close-temp-file
  - 7baf2809cf Closes handle to temporary file before returning the path
- **PR** #40430: (twangboy) Fix logic for __virtual__ in win_dsc and win_psget @ *2017-03-30 22:06:16 UTC*
  - 5c78d55eab Merge pull request #40430 from twangboy/fix_virtual
  - 08e95ce4f0 Add logging on __virtual__ failures
  - 43ecb1a597 Fix logic for __virtual__
- **PR** #40431: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-03-30 21:37:39 UTC*
  - b855f29928 Merge pull request #40431 from rallytime/merge-2016.11
  - d5576d75e7 Merge branch '2016.3' into '2016.11'
  - b6770fd81f Merge pull request #40407 from aesdana/fix_rabbitmq_version_check
    - 4c0763fa2f Added split to cut off debian_revision from rabbitmq-server version Fixes #40396
  - d4fb45d9f8 Merge pull request #40424 from terminalmage/fix-open-filehandle
    - 66251263cf Fix open filehandles
  - 8708096365 Merge pull request #40399 from terminalmage/docker-py_version
    - 14c6575655 Add docker-py version to the versions report

  - ff1266b3a6 Merge pull request #40391 from Ch3LL/2016.3.7_release_notes
    - f532ec5288 initial 2016.3.7 release notes
  - 96bf9427b0 Merge pull request #40368 from Ch3LL/bump_version_3
    - a02fa7dd1f [2016.3] Bump previous version to 2016.3.6
- **PR** #40401: (roaldnefs) fix Ubuntu notation in docs/faq.rst @ *2017-03-29 20:28:31 UTC*
  - 7d900d31ea Merge pull request #40401 from roaldnefs/fix-doc-faq
  - 21f161fecc fix Ubuntu notation in docs/faq.rst
- **ISSUE** #29028: (kevins9) state.sls fails to render state with pillar data: Jinja variable 'dict object' has no attribute (refs: #37795)
- **PR** #40390: (rallytime) Back-port #37795 to 2016.11 @ *2017-03-29 19:05:12 UTC*
  - **PR** #37795: (jettero) please tell me where is the "error: 'dict' object has no …" (refs: #40390)
  - 70a3f963ec Merge pull request #40390 from rallytime/bp-37795
  - 1ba15577bd Pylint fix
  - ec65924659 please tell me where is the "error: 'dict' object has no attribute 'seek'" ??
- **PR** #40395: (rallytime) Handle AttributeError for dockerng_mod.docker attempt fails and docker is installed @ *2017-03-29 17:47:11 UTC*
  - f8fbfff7dc Merge pull request #40395 from rallytime/catch-attribute-error-docker-test
  - 99c8dcc18e Handle AttributeError for dockerng_mod.docker attempt fails and docker is installed
- **PR** #40362: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-03-28 22:50:32 UTC*
  - d7d3d68035 Merge pull request #40362 from rallytime/merge-2016.11
  - 4f1543c2a1 Merge branch '2016.3' into '2016.11'
    - 1381f97292 Merge pull request #40264 from meaksh/2016.3-gather_job_timeout-fix
      - 68dccae5b4 Makes sure "gather_job_timeout" is an integer
- **PR** #40372: (zer0def) Fixes related to cache directory argument changes in pip>=6. @ *2017-03-28 22:48:41 UTC*
  - 2febd05896 Merge pull request #40372 from zer0def/pip-cache-fixes
  - d68067f1d7 Merge remote-tracking branch 'main/2016.11' into pip-cache-fixes
  - 4f23a23ca8 Fixed the *test_install_download_cache_argument_in_resulting_command* to accomodate introduced cache directory argument fixes and renamed it to *test_install_download_cache_dir_arguments_in_resulting_command*.
  - 9d0f94eeba Fixed unnecessary API changes introduced with suggested changes.
- **PR** #40369: (Ch3LL) [2016.11] Bump previous version to 2016.3.6 @ *2017-03-28 18:50:39 UTC*
  - 6162698c87 Merge pull request #40369 from Ch3LL/bump_version_11
  - 7597d96edb [2016.11] Bump previous version to 2016.3.6
- **ISSUE** #40322: (Whissi) ssh_auth.absent: Wrong comment when test=True (refs: #40333)
- **ISSUE** #40321: (Whissi) state.alternatives: Wrong comment when test=True (refs: #40333)
- **PR** #40333: (gtmanfred) fix some test=True comments @ *2017-03-28 16:11:01 UTC*
  - 2d2cb5b837 Merge pull request #40333 from gtmanfred/2016.11

  - 5596620dfb fix some test=True comments
- **PR** #40347: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-03-28 02:39:31 UTC*
  - bb37f133fc Merge pull request #40347 from rallytime/merge-2016.11
  - e77e86db3a Merge branch '2016.3' into '2016.11'
  - 17ab1da0ab Merge pull request #40345 from twangboy/fix_osx_build
    - 3207d670c5 Fix osx build
  - 7ab10491ab Merge pull request #40338 from UtahDave/fix_cherrypy_ssl_error_link
    - 280b501950 Upstream cherrypy moved to Github from Bitbucket
  - **PR** saltstack/salt#40332: (zer0def) Fixes related to cache directory argument changes in pip>=6. (refs: #40346)
- **PR** #40346: (cachedout) Revert "Fixes related to cache directory argument changes in pip>=6." @ *2017-03-27 23:17:29 UTC*
  - a572b46183 Merge pull request #40346 from saltstack/revert-40332-pip-cache-fixes
  - b4753d1a5a Revert "Fixes related to cache directory argument changes in pip>=6."
- **ISSUE** #40296: (L4rS6) Wrong documentation in mount.mounted (refs: #40326)
- **PR** #40326: (L4rS6) Update mount state documentation (Fixes: #40296) @ *2017-03-27 23:15:53 UTC*
  - a91bab867e Merge pull request #40326 from L4rS6/update-mount-state-doc
  - a717c527a1 Update mount state documentation (Fixes: #40296)
- **PR** #40328: (L4rS6) Fixes wrong compared extra_mount_ignore_fs_keys key. @ *2017-03-27 23:14:22 UTC*
  - ca2980cfb0 Merge pull request #40328 from L4rS6/fix-mount-state-extra-ignore-fs-key
  - f0f68b9033 Fixes wrong compared extra_mount_ignore_fs_keys key.
- **PR** #40329: (isbm) Merge tops (backport) @ *2017-03-27 23:13:47 UTC*
  - 3a6c5d0297 Merge pull request #40329 from isbm/isbm-merge-tops-201611
  - a762c9edda Merge output from master_tops
- **PR** #40285: (rallytime) Dockerng unit tests fixes: isolate global variables @ *2017-03-27 23:05:03 UTC*
  - 2b7b2f1cb4 Merge pull request #40285 from rallytime/docker-test-fixes
  - 0f263a52e0 Mock out the get_client_args mocks in the dockerng module tests more aggressively
  - f1352fe253 Add one more dockerng.version mock that was missed previously
  - 0d31d2c4d1 Add a couple more patches for docker.version information
  - a9c5eebaf0 Clean up dockerng unit tests to avoid global variables and fixup some patching
- **PR** #40341: (twangboy) Fix service.create, fix docs @ *2017-03-27 21:46:19 UTC*
  - 01efc842c1 Merge pull request #40341 from twangboy/fix_win_service
  - 6736457ec8 Docs for create
  - 652cf08f8a Fix service.create, fix docs
- **PR** #40332: (zer0def) Fixes related to cache directory argument changes in pip>=6. @ *2017-03-27 21:01:15 UTC*
  - 8eabcca6dc Merge pull request #40332 from zer0def/pip-cache-fixes
  - 7976840100 Fixes related to cache directory changes in pip>=6.

- **PR** #40337: (Ch3LL) Add archive.extracted with use_cmd_unzip argument @ *2017-03-27 21:00:23 UTC*
  - ceba1b9bc6 Merge pull request #40337 from Ch3LL/add_unzip_test
  - 8b21b4c8bb add use_cmd_unzip test
- **PR** #40312: (rallytime) Update minion data cache documentation @ *2017-03-27 20:56:55 UTC*
  - a192597ec2 Merge pull request #40312 from rallytime/cache-docs
  - 5363e0b58b Update minion data cache documentation
- **PR** #40315: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-03-27 15:11:25 UTC*
  - 7f16754619 Merge pull request #40315 from rallytime/merge-2016.11
  - c65d602f60 Merge branch '2016.3' into '2016.11'
    - 7c21153d3a Merge pull request #40300 from meaksh/2016.3-adding-timeouts-parameters-to-cmd_batch
      - 9174e6f281 Fixes testing opts dict for batch unit tests
      - b1de79abcf Adds custom 'timeout' and 'gather_job_timeout' to 'local_batch' client
- **PR** #40313: (techhat) Add minimum and maximum to calls to calc @ *2017-03-27 14:54:15 UTC*
  - a9a73bf8dc Merge pull request #40313 from techhat/calcref
  - 7106a86258 Use named kwargs
  - 822f3b81c3 Add minimum and maximum to calls to calc
- **ISSUE** #40247: (eldadru) boto_rds.delete wait_for_deletion checks rds status incorrectly and always loop until timeout (refs: #40277)
- **PR** #40277: (eldadru) Fixing boto_rds.py delete() wait_for_deletion, if statement was inco… @ *2017-03-24 22:29:25 UTC*
  - 9d0762deca Merge pull request #40277 from eldadru/Fix-40247-boto_rds-delete-wait-for-deletion-failure
  - 3c15a32764 Fixing boto_rds.py delete() wait_for_deletion, if statement was incorrectly checking the return value of boto_rds.py exists() method.
- **PR** #40280: (bewing) Clean up temporary file in net.load_template @ *2017-03-24 22:27:04 UTC*
  - **PR** #40273: (bewing) Clean up temporary file in net.load_template (refs: #40280)
  - 6c29c81d01 Merge pull request #40280 from bewing/bp_40273
  - f028e939f5 Clean up temporary file in net.load_template
- **ISSUE** #37972: (ebauman) salt-run execution for master with no AAAA record adds significant execution time (refs: #40310)
- **PR** #40310: (gtmanfred) add warning when no host/dns record is found for fqdn_ip @ *2017-03-24 21:55:20 UTC*
  - 839b620f32 Merge pull request #40310 from gtmanfred/2016.11
  - cff027ddc6 add warning when no host/dns record is found for fqdn
- **PR** #40288: (dmurphy18) Execution module network support for AIX @ *2017-03-24 20:10:36 UTC*
  - eb86d55478 Merge pull request #40288 from dmurphy18/aix_network
  - b53a95dab1 Further update to us in similar to review comments
  - 59c0bdc14d Updated for review comments
  - 031c9457ba Execution module network support for AIX

- **PR** #40308: (rallytime) Back-port #38835 to 2016.11 @ *2017-03-24 19:00:46 UTC*
  - **PR** #38835: (UtahDave) Cache docs (refs: #40308)
  - 4928026253 Merge pull request #40308 from rallytime/bp-38835
  - 3ba50d3c52 add info about what is cached
  - 77e8f6aff9 fix config example
  - 61f2fa9339 Add documentation for the Minion data cache
- **PR** #40287: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-03-24 16:50:23 UTC*
  - 12a9fc43c9 Merge pull request #40287 from rallytime/merge-2016.11
  - 77415369cc Merge branch '2016.3' into '2016.11'
  - 0e2d52c3ea Merge pull request #40260 from lubyou/fix-join_domain
    - 1cb15d1ea8 use win32api.FormatMessage() to get the error message for the system code
  - 0c62bb37d3 Merge pull request #40275 from UtahDave/2016.3local
    - 9f0c9802c2 remove reference to auth_minion.
  - 57ce474d73 Merge pull request #40265 from terminalmage/issue40219
    - 1a731e0216 Pop off the version when aggregating pkg states
    - 0055fda3e9 Properly aggregate version when passed with name
    - 62d76f50fc Don't aggregate both name/pkgs and sources in pkg states
  - b208630d85 Merge pull request #40201 from sergeizv/cloud-roster-fixes-2016.3
    - d87b377ad2 cloud roster: Don't stop if minion wasn't found in cloud cache index
    - a6865e0283 cloud roster: Check whether show_instance succeeded on node
    - 1b45c8e8c2 cloud roster: Check provider and profile configs for ssh_username
    - a18250b2e4 cloud roster: Return proper target name
    - 637930b2b3 cloud roster: Fix extracting instance's info
    - dd1d3aac74 cloud roster: Work with custom conf dir
- **PR** #40250: (techhat) Add wait_for_fun() to set_tags() @ *2017-03-23 16:42:13 UTC*
  - **PR** #40225: (techhat) Add wait_for_fun() to set_tags() (refs: #40250)
  - b7f9100e6d Merge pull request #40250 from techhat/settags
  - baff7a046d Add wait_for_fun() to set_tags()
- **ISSUE** #39976: (peterhirn) win_lgpo missing policies, eg. *Prevent the usage of OneDrive for file storage* (refs: #40255, #40253)
- **PR** #40255: (lomeroe) backport #40253 @ *2017-03-23 16:36:44 UTC*
  - **PR** #40253: (lomeroe) correct method of getting 'text' of the XML object to compare to the … (refs: #40255)
  - 904e144ae4 Merge pull request #40255 from lomeroe/fix_39976_2016.11
  - 0e9f5820cc backport #40253
- **PR** #40240: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-03-23 14:14:11 UTC*
  - **PR** #40237: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 (refs: #40240)

  - 720a362c7a Merge pull request #40240 from rallytime/merge-2016.11
  - 5c5b74b09a Merge branch '2016.3' into '2016.11'
    - 35ced607dd Merge pull request #40226 from terminalmage/issue40149
      - 2a8df9384c Fix wrong errno in systemd.py
    - 24c4ae9c21 Merge pull request #40232 from rallytime/update-release-notes
      - 2ead188b4f Update release notes for 2016.3.6
    - c59ae9a82c Merge pull request #39855 from Foxlik/use_regex_to_compare_authorized_keys
      - d46845a5b6 Add newline at end of file
      - d4a3c8a66a Use regular expression instead of split when replacing authorized_keys
    - fd10430018 Merge pull request #40221 from rallytime/bp-39179
      - 07dc2de084 fix error parsing
    - a27a2cc3bb Merge pull request #40206 from cro/sign_pub_take2
      - 01048de83f leave sign_pub_messages off on minion by default.
      - a82b005507 Leave sign_pub_messages off by default.
    - d1abb4cbaa Merge pull request #40193 from rallytime/bp-40117
      - cf1857904b More optimization.
      - 5a08266814 Removed debug statemnt
      - f557f7c6bb Added fix for issue 39393
      - bb62278b73 Reverting changes.
      - a9107cde44 Added if condition for broken link.
    - 0f1ff4d4a8 Merge pull request #40196 from twangboy/win_fix_deps
      - 6761527793 Update dependencies for PyOpenSSL
    - b0501515cb Merge pull request #40184 from terminalmage/link-reactor-example
      - a42be82993 Link to minion start reactor example from FAQ.

- **ISSUE** #39445: (systemtrap) state file.copy for directories does not set ownership recursively (refs: #40030)
- **PR** #40231: (rallytime) Back-port #40030 to 2016.11 @ *2017-03-22 23:14:40 UTC*
  - **PR** #40030: (narendraingale2) Added changes for fix_39445 (refs: #40231)
  - c40376250f Merge pull request #40231 from rallytime/bp-40030
  - 4d1c687cbd Using lchown insted of chown.
  - 52b3d986b5 Added changes for fix_39445
  - **PR** saltstack/salt#40225: (techhat) Add wait_for_fun() to set_tags() (refs: #40239)
- **PR** #40239: (cachedout) Revert "Add wait_for_fun() to set_tags()" @ *2017-03-22 22:59:16 UTC*
  - e39f5cbf40 Merge pull request #40239 from saltstack/revert-40225-waitforfun
  - 95bdab87b4 Revert "Add wait_for_fun() to set_tags()"
- **PR** #40225: (techhat) Add wait_for_fun() to set_tags() (refs: #40250) @ *2017-03-22 18:15:35 UTC*
  - 11d2f5abec Merge pull request #40225 from techhat/waitforfun

  - 89b5010883 Add wait_for_fun() to set_tags()
- **PR** #40172: (dmurphy18) Fix solaris network @ *2017-03-22 17:41:56 UTC*
  - c8cfbb7df6 Merge pull request #40172 from dmurphy18/fix_solaris_network
  - a6218b9484 Updated use of tail on Solaris and Sun-like OS
  - 90e6a1d8f6 Further update to support correct tail in network for Solaris
  - 5b6d33dd70 Fix use of correct tail on Solaris for active_tcp
- **PR** #40210: (rallytime) Skip flaky test for now @ *2017-03-22 16:34:41 UTC*
  - e9a4e8548b Merge pull request #40210 from rallytime/test-skip
  - 0ba773d86b Skip flaky test for now
- **ISSUE** #40204: (sofixa) InfluxDB returner present on salt-minion(installed via salt-bootstrap and updated via apt-get) has a bug (refs: #40209)
- **PR** #40209: (sofixa) change InfluxDB get_version to expect status code 204 @ *2017-03-21 21:42:26 UTC*
  - 0b00489eb2 Merge pull request #40209 from sofixa/2016.11
  - e1cc7234ff change InfluxDB get_version to expect status code 204
- **ISSUE** #39775: (mirceaulinic) Proxy *mine_interval* config ignored (refs: #39935, #saltstack/salt‘#39935‘_, #39776)
  - **PR** saltstack/salt#39935: (cro) Add special token to insert the minion id into the default_include path (refs: #40202)
- **PR** #40202: (cro) Revert ”Add special token to insert the minion id into the default_include path” @ *2017-03-21 21:37:33 UTC*
  - 66bc680d0a Merge pull request #40202 from saltstack/revert-39935-namespace_proxy_cfg
  - bb71710747 Revert ”Add special token to insert the minion id into the default_include path”
- **PR** #40199: (whiteinge) Ponysay emergency hotfix @ *2017-03-21 21:10:21 UTC*
  - d8f0b79997 Merge pull request #40199 from whiteinge/ponysay-emergency-hotfix
  - 85ea61b544 Add depends note
  - 5a271acfdc Fix ponysay outputter hardcoded path
- **PR** #40194: (terminalmage) Change imports for dockerng tests @ *2017-03-21 19:34:55 UTC*
  - 82cee58e72 Merge pull request #40194 from terminalmage/fix-docker-test-imports
  - 6caedb0de8 Change imports for dockerng tests
- **PR** #40189: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-03-21 18:02:51 UTC*
  - 0b512f9ffb Merge pull request #40189 from rallytime/merge-2016.11
  - a55c4138a8 Merge branch ’2016.3’ into ’2016.11’
    - d4e6c58526 Merge pull request #40182 from terminalmage/dockerng-mod_watch-stopped
      - 4629a26fb7 Add support for ”stopped” state to dockerng’s mod_watch
    - a0b4082484 Merge pull request #40171 from Ch3LL/2016.3.6_release
      - 9c6d8d892f additional PRs/issues for 2016.3.6 release notes
    - 33ba7821f7 Merge pull request #40120 from sergeizv/gce-expand-node-fix

        - 9d0fbe7e01 gce: Exclude GCENodeDriver objects from _expand_node result
      - 48843977c3 Merge pull request #40122 from meaksh/2016.3-yum-downloadonly-support
        - 067f3f77c2 Adding downloadonly support to yum/dnf module
      - 60e1d4e2f3 Merge pull request #40159 from cro/sign_pub
        - e663b761fb Fix small syntax error
        - 0a0f46fb14 Turn on sign_pub_messages by default. Make sure messages with no 'sig' are dropped with error when sign_pub_messages is True.
- **ISSUE** #39779: (sp1r) Pillar scheduling is broken (refs: #40034)
- **ISSUE** #38523: (MorphBonehunter) schedule not changed on pillar update after minion restart (refs: #40034)
- **ISSUE** #36134: (Ch3LL) carbon: multi-master with failover does not failover when master goes down (refs: #36437)
- **PR** #40034: (sp1r) Disallow modification of jobs from pillar with schedule execution module @ *2017-03-21 16:36:34 UTC*
  - **PR** #36437: (DmitryKuzmenko) Keep the schedule jobs in ONE place. (refs: #40034)
  - d9cb222aa8 Merge pull request #40034 from sp1r/fix-pillar-scheduling
  - 595f786327 fix evaluating jobs when "pillar" is missing in opts
  - 9d5db1910c fix initial data structure for schedule tests
  - d3a2489c9c schedule tests to ensure pillar jobs are not modified
  - 27385ff49c added a check ensuring schedule is a dict before merging
  - 14d71918b2 Fixes #39779
- **PR** #40160: (eldadru) Fix this issue: https://github.com/saltstack/salt/issues/40073, descr... @ *2017-03-20 21:37:43 UTC*
  - 257c862c52 Merge pull request #40160 from eldadru/fix-issue-40073-boto-rds-describe-empty-dict
  - 954c871332 Fix this issue: https://github.com/saltstack/salt/issues/40073, describe return dictionary returned empty , probably as result of incorrect past merge (see discussion on issue)
- **PR** #40162: (rallytime) Make sure the tornado web server is stopped at the end of the test class @ *2017-03-20 20:35:21 UTC*
  - aec504173a Merge pull request #40162 from rallytime/archive-integration-test-fixes
  - dd193cc740 Make sure the tornado web server is stopped at the end of the test class
- **PR** #40158: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-03-20 20:34:23 UTC*
  - 461e15f0f4 Merge pull request #40158 from rallytime/merge-2016.11
  - 88f3ebd7e9 Remove extra "connect" kwarg caught by linter
  - f4d4768a6d Merge branch '2016.3' into '2016.11'
    - 28e4fc17b6 Merge pull request #40123 from twangboy/win_fix_network
      - 06dfd55ef9 Adds support for inet_pton in Windows to network util
    - 35ddb79f59 Merge pull request #40141 from bobrik/fallback-resolve
      - af1545deed Use the first address if cannot connect to any

- **PR** #40165: (rallytime) Don't try to run the dockerng unit tests if docker-py is missing @ *2017-03-20 20:33:19 UTC*
  - b235f0953f Merge pull request #40165 from rallytime/gate-docker-unit-tests
  - f32d8a8683 Don't try to run the dockerng unit tests if docker-py is missing
- **PR** #40085: (mirceaulinic) VRF arg and better doc for ping and traceroute @ *2017-03-20 19:48:57 UTC*
  - db9fb58b82 Merge pull request #40085 from cloudflare/fix-ping-tr
  - 6cbdd61b54 Strip trailing whitespaces
  - 897a2a37c3 VRF arg and better doc for ping and traceroute
- **PR** #40095: (skizunov) dns_check should not try to connect when connect=False @ *2017-03-17 17:31:42 UTC*
  - 3bac06f099 Merge pull request #40095 from skizunov/develop2
  - 880790f743 dns_check should not try to connect when connect=False
- **PR** #40096: (skizunov) When building up the 'master_uri_list', do not try to connect @ *2017-03-17 17:13:41 UTC*
  - 31da90edd9 Merge pull request #40096 from skizunov/develop3
  - eb9a0a6fd1 When building up the 'master_uri_list', do not try to connect
- **PR** #40111: (eldadru) Fixing simple issue 40081 - the key parameter of the method create ov… @ *2017-03-17 17:00:03 UTC*
  - 5303386d93 Merge pull request #40111 from eldadru/fix-issue-40081-boto-rds-create-overwritten-key-parameter
  - 78b5d112d7 Fixing simple issue 40081 - the key parameter of the method create overwritten by internal loop.
- **PR** #40118: (rallytime) Add CLI Example for dockerng.get_client_args @ *2017-03-17 16:34:13 UTC*
  - d2e376e8f2 Merge pull request #40118 from rallytime/cli-example
  - bb496bb7f4 Add CLI Example for dockerng.get_client_args
- **PR** #40097: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-03-17 15:17:08 UTC*
  - baef5009aa Merge pull request #40097 from rallytime/merge-2016.11
  - ef1ff38f8d Merge branch '2016.3' into '2016.11'
  - 116201f345 Merge pull request #40059 from terminalmage/fix-virtualenv-traceback
    - e3cfd29d6b Fix traceback when virtualenv.managed is invoked with nonexistant user
  - a01b52b9a3 Merge pull request #40090 from rallytime/bp-40056
    - ae012db87a update mention bot blacklist
  - d1570bba4c Merge pull request #40057 from cachedout/ollie_blacklist
    - 0ac2e83d37 Merge branch '2016.3' into ollie_blacklist
    - 5592c680b5 More mentionbot blacklists
- **ISSUE** #39771: (mirceaulinic) Empty __proxy__ dunder inside scheduler (refs: #40077)
- **PR** #40077: (mirceaulinic) Fix #39771 (Empty __proxy__ dunder inside scheduler) @ *2017-03-16 20:56:02 UTC*
  - 9ef3e070c2 Merge pull request #40077 from cloudflare/fix-39771
  - cd319e7e39 Add proxy kwarg to scheduler

  - c6e6dd1a04 ProxyMinion: correctly build the scheduler
- **PR** #40088: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-03-16 19:58:44 UTC*
  - b12720a56f Merge pull request #40088 from rallytime/merge-2016.11
  - 626bd03885 Merge branch '2016.3' into '2016.11'
  - d36bdb1a6e Merge pull request #40070 from Ch3LL/2016.3.6_release
    - a1f8b49bd1 update 2016.3.6 release notes with additional PR's
  - 8dcffc7751 Merge pull request #40018 from meaksh/2016.3-handling-timeouts-for-manage.up-runner
    - 9f5c3b7dcd Allows to set custom timeouts for 'manage.up' and 'manage.status'
    - 2102d9c75c Allows to set 'timeout' and 'gather_job_timeout' via kwargs
  - 22fc5299a2 Merge pull request #40038 from velom/fix-pip-freeze-parsing
    - 3fae91d879 correctly parse "pkg_name===version" from pip freeze
  - 3584f935fa Merge pull request #40053 from saltstack/rh_ip_patch
    - 219947acdb Update rh_ip.py
  - 837432d3d2 Merge pull request #40041 from terminalmage/issue40011
    - 5b5d1b375c Fix transposed lines in salt.utils.process
- **PR** #40055: (rallytime) Update "yaml" code-block references with "jinja" where needed @ *2017-03-16 16:30:38 UTC*
  - 703ab23953 Merge pull request #40055 from rallytime/doc-build-warnings
  - 72d16c9fa9 Update "yaml" code-block references with "jinja" where needed
- **PR** #40072: (meaksh) [2016.11] Allows overriding 'timeout' and 'gather_job_timeout' to 'manage.up' runner call @ *2017-03-16 15:31:46 UTC*
  - **PR** #40018: (meaksh) Allows overriding 'timeout' and 'gather_job_timeout' to 'manage.up' runner call (refs: #40072)
  - e73a1d0e54 Merge pull request #40072 from meaksh/2016.11-handling-timeouts-for-manage.up-runner
  - 40246d3723 Allows to set custom timeouts for 'manage.up' and 'manage.status'
  - ad232fdc01 Allows to set 'timeout' and 'gather_job_timeout' via kwargs
- **PR** #40045: (terminalmage) Fix error when chhome is invoked by user.present state in Windows @ *2017-03-15 19:00:41 UTC*
  - 2f28ec26ee Merge pull request #40045 from terminalmage/fix-windows-user-present
  - 359af3bb2b Fix error when chhome is invoked by user.present state in Windows
- **PR** #40047: (rallytime) Back-port #40000 to 2016.11 @ *2017-03-15 17:47:37 UTC*
  - **PR** #40000: (skizunov) Fix exception in salt-call when master_type is 'disable' (refs: #40047)
  - 4067625676 Merge pull request #40047 from rallytime/bp-40000
  - 11766c7259 Fix exception in salt-call when master_type is 'disable'
- **PR** #40023: (jeanpralo) We need to match on .p not just strip '.p' otherwise it will remove a… @ *2017-03-14 23:14:56 UTC*
  - 86f7195e0e Merge pull request #40023 from jeanpralo/fix-minions-cant-finish-by-char-p

  - d7b0c8ae88 We need to match on .p not just strip '.p' otherwise it will remove any p from the string even if we have no dot
- **PR** #40025: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-03-14 23:14:33 UTC*
  - 277bd17ff2 Merge pull request #40025 from rallytime/merge-2016.11
  - 029f28bbd5 Merge branch '2016.3' into '2016.11'
  - ee7f3b1200 Merge pull request #40021 from Ch3LL/2016.3.6_release
    - f3e7e4fb2a Add 2016.3.6 Release Notes
  - 26895b7be2 Merge pull request #40016 from terminalmage/fix-grains-test
    - 0ec81a4cde Fixup a syntax error
    - 5d84b40bfd Attempt to fix failing grains tests in 2016.3
  - 0c61d064ad Merge pull request #39980 from vutny/cmd-run-state-bg
    - a81dc9dfc1 [2016.3] Allow to use *bg* kwarg for *cmd.run* state function
  - b042484455 Merge pull request #39994 from rallytime/ulimits-dockerng-version
    - 37bd800fac Add a versionadded tag for dockerng ulimits addition
  - e125c94ba5 Merge pull request #39988 from terminalmage/dockerng-timeout
    - bd2519ed1b Add comment explaining change from #39973
- **PR** #40020: (dmurphy18) Full support for execution module timezone on AIX @ *2017-03-14 21:05:31 UTC*
  - 8db74fb275 Merge pull request #40020 from dmurphy18/aix_timezone
  - aabbbffd45 Full support to execution module timezone on AIX
  - 16d5c7ce4a WIP: timezone support for AIX
- **PR** #39924: (dmurphy18) Add AIX support for user and group execution modules @ *2017-03-14 21:04:02 UTC*
  - 60066da614 Merge pull request #39924 from dmurphy18/salt_aix_fixMar
  - 5077c989bb Updated changes file for added AIX support
  - 8e107bd43e WIP: support for useradd on AIX
  - 2f87d727d6 WIP: group support for AIX
- **PR** #40010: (jettero) S3 bucket path broken @ *2017-03-14 19:01:01 UTC*
  - cd73eafec8 Merge pull request #40010 from jettero/s3-bucket-path-broken
  - acee5bf7c8 clarify this, because it messes people up in the mailing lists, and myself briefly before I thought about it
  - 8102ac8e3c same here
  - 21b79e00be In order for the heredoc to be correct, bucket and path have to default to ", not None
- **PR** #39991: (terminalmage) Document the fact that the checksum type can be omitted in file.managed states @ *2017-03-14 15:58:11 UTC*
  - 61f1fb04c5 Merge pull request #39991 from terminalmage/source_hash-docs
  - 537fc36029 Document the fact that the checksum type can be omitted in file.managed states
- **PR** #39984: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-03-13 18:30:16 UTC*
  - 53d14d8ad9 Merge pull request #39984 from rallytime/merge-2016.11

  - ef6f4b15ca Merge branch '2016.3' into '2016.11'
  - cd0336e868 Merge pull request #39973 from terminalmage/dockerng-timeout
    - 869416e7db Don't use docker.Client instance from context if missing attributes
- **PR** #39967: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-03-10 23:45:33 UTC*
  - 31c00740e7 Merge pull request #39967 from rallytime/merge-2016.11
  - 3022466615 Merge branch '2016.3' into '2016.11'
  - 282c607d26 Merge pull request #39962 from cachedout/disable_mentionbot_delay_3
    - 7a638f204b Disable mention bot delay on 2016.3
  - 1e0c88ae08 Merge pull request #39937 from cachedout/gpg_zypper
    - 13ed0d1209 Fix –non-gpg-checks in zypper module
- **PR** #39963: (cachedout) Mention bot delay disable for 2016.11 @ *2017-03-10 20:25:25 UTC*
  - 269a2fd739 Merge pull request #39963 from cachedout/disable_mentionbot_delay_11
  - 5fcea05691 Mention bot delay disable for 2016.11
- **ISSUE** #7997: (shantanub) Proper way to upgrade salt-minions / salt-master packages without losing minion connectivity (refs: #39952)
- **PR** #39952: (vutny) Fix #7997: describe how to upgrade Salt Minion in a proper way @ *2017-03-10 18:41:57 UTC*
  - 6350b07384 Merge pull request #39952 from vutny/doc-faq-minion-upgrade-restart
  - d989d749d6 Fix #7997: describe how to upgrade Salt Minion in a proper way
- **ISSUE** #39775: (mirceaulinic) Proxy *mine_interval* config ignored (refs: #39935, #saltstack/salt‘#39935‘_, #39776)
- **PR** #39935: (cro) Add special token to insert the minion id into the default_include path @ *2017-03-10 17:51:55 UTC*
  - dc7d4f4224 Merge pull request #39935 from cro/namespace_proxy_cfg
  - e4aef54c73 Add special token to insert the minion id into the default_include path
- **PR** #39936: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-03-10 17:05:04 UTC*
  - 9503a1d0c6 Merge pull request #39936 from rallytime/merge-2016.11
  - c8b5d390b5 Merge branch '2016.3' into '2016.11'
    - 4526fc6e08 Merge pull request #39929 from terminalmage/pr-39770-2016.3
      - cf0100dabe Scrap event-based approach for refreshing grains
    - 111110caf8 Merge pull request #39919 from The-Loeki/patch-1
      - 170cbadc54 CIDR matching supports IPv6, update docs
    - caf10e9988 Merge pull request #39899 from techhat/cleanupdisks
      - baf4579e63 Update cleanup function for azure
    - fcf95f3654 Merge pull request #39871 from terminalmage/squelch-import-warning
      - 2b2ec69d04 Squelch warning for pygit2 import
    - f223fa8906 Merge pull request #39794 from cachedout/clean_monitor_socket_shutdown

        - 2e683e788b Clean up errors which might be thrown when the monitor socket shuts down
    - 4002dc1947 Merge pull request #39819 from terminalmage/top-file-matching-docs
        - 7178e77eee Improve the Top File matching docs
    - c08aaeb7fd Merge pull request #39820 from ni3mm4nd/beacons_topic_doc_typo
        - 804b12048c Add missing apostrophe
    - cbd2a4e3cc Merge pull request #39826 from cachedout/yubikey_fix
        - 6125eff02d Add group func to yubikey auth
    - f575ef459f Merge pull request #39624 from drawsmcgraw/39622
        - 13da50be33 Fix indention lint errors
        - 545026352f Address issue 39622
    - 1f3619c1e5 Merge pull request #39796 from cachedout/master_shutdown
        - e31d46c1b8 Stop the process manager when it no longer has processes to manage
    - 53341cf152 Merge pull request #39791 from gtmanfred/2016.3
        - 3ab4f843bf load runners if role is master
    - c234c25092 Merge pull request #39784 from sergeizv/fix-39782
        - b71c3fe13c Revert "cloud.clouds.ec2: cache each named node (#33164)"
    - 4ee59be22c Merge pull request #39766 from rallytime/fix-ipv6-connection
        - 65b239664e Restore ipv6 connectivity and "master: <ip>:<port>" support

- **ISSUE** #38121: (Da-Juan) Beacon configuration doesn't work as a list (refs: #39932, #39930)
- **PR** #39932: (rallytime) Cherry-pick the beacon fixes made in #39930 to 2016.11 @ *2017-03-10 00:21:09 UTC*
    - **PR** #39930: (s0undt3ch) Moar Py3 and a fix for #38121 (refs: #39932)
    - 899e037f0a Merge pull request #39932 from rallytime/cp-beacon-fixes
    - 4a52cca926 Pylint fixes
    - 4627c4ea6d Code cleanup and make sure the beacons config file is deleted after testing
    - c7fc09f97d Support the new list configuration format.
    - be06df9b64 Remove **args, **kwargs*. Not needed, not useful.
    - 4a242829ee These tests aren't even using mock!
    - 6408b123e7 These tests are not destructive
    - 50e51b5b9d The beacons configuration is now a list. Handle it!
- **PR** #39933: (hkrist) Fixed rawfile_json returner output format. @ *2017-03-10 00:20:52 UTC*
    - 2e68edee4a Merge pull request #39933 from hkrist/fix-rawfile_json_returner-format
    - 4d0ddcd110 Fixed rawfile_json returner output format. It outputted python object instead of standard json.
- **PR** #39934: (dmurphy18) Correct comment lines output from execution module's host.list_hosts @ *2017-03-10 00:20:14 UTC*
    - fb0dc33c42 Merge pull request #39934 from dmurphy18/fix_host_list
    - e7b9a45079 Correct comment lines output got list_hosts

- **PR** #39900: (twangboy) Namespace the line function properly in win_file @ *2017-03-09 22:19:12 UTC*
  - a6f88d03df Merge pull request #39900 from twangboy/win_fix_line
  - 462bdecd33 Namespace the line function properly in win_file
- **ISSUE** #37741: (discountbin) Check in file.replace state for ignore_if_missing (refs: #37743, #39910)
- **PR** #39910: (rallytime) Back-port #37743 to 2016.11 @ *2017-03-09 22:16:58 UTC*
  - **PR** #37743: (discountbin) Adding check for ignore_if_missing param when calling _check_file. (refs: #39910)
  - 77ecff4e02 Merge pull request #39910 from rallytime/bp-37743
  - ca306c0860 Replace pass with updated comment for return
  - 1a78878b47 Adding check for ignore_if_missing param when calling _check_file.
- **PR** #39770: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-03-09 22:00:17 UTC*
  - c2d4d17589 Merge pull request #39770 from rallytime/merge-2016.11
  - dbaea3de73 Remove extra refresh reference that snuck in
  - d9f48ac6ea Don't shadow refresh_pillar
  - d86b03dc90 Remove manual refresh code from minion.py
  - a7e419e35f Scrap event-based approach for refreshing grains
  - 776a9431b9 Merge branch '2016.3' into '2016.11'
    - a24da31131 Merge pull request #39761 from cachedout/issue_33187
      - c2df29edb2 Properly display error in jboss7 state
    - 0888bc32ef Merge pull request #39728 from rallytime/update-release-ver-2016.3
      - c9bc8af8f2 [2016.3] Bump latest release version to 2016.11.3
    - b52dbeec68 Merge pull request #39619 from terminalmage/zd1207
      - c7dfb494a6 Fix mocking for grains refresh
      - 7e0ced3b45 Properly hand proxy minions
      - 692c456da3 Add a function to simply refresh the grains
- **PR** #39872: (techhat) Add installation tips for azurearm driver @ *2017-03-07 23:18:04 UTC*
  - 801ff28053 Merge pull request #39872 from techhat/fixdocs
  - 35440c5936 Add installation tips for azure
  - 2a1ae0bf2e Change example master in azure docs
- **PR** #39837: (terminalmage) Fix regression in archive.extracted when it runs file.directory @ *2017-03-07 04:09:51 UTC*
  - 6d0f15c31a Merge pull request #39837 from terminalmage/more-issue39751
  - 0285ff3c7d Fix regression in archive.extracted when it runs file.directory
- **PR** #39858: (techhat) Reorder keys that were being declared in the wrong place @ *2017-03-07 03:51:56 UTC*
  - 68752a2a18 Merge pull request #39858 from techhat/statuskey
  - 507a4f7f93 Reorder keys that were being declared in the wrong place
- **ISSUE** #38830: (danielmotaleite) salt-ssh: vault fails to use config (refs: #38943)

- **PR** #39862: (rallytime) Back-port #38943 to 2016.11 @ *2017-03-07 03:34:40 UTC*
  - **PR** #38943: (thatch45) When we generate the pillar we should send in the master opts (refs: #39862)
  - 49c8faa141 Merge pull request #39862 from rallytime/bp-38943
  - e21b16c002 try it with a different init sequence
  - 92cac0ff8b make it a deepcopy
  - 58cb8cd4f5 make sure to copy the top dict reference since we are moding it
  - a0b671ea43 When we generate the pillar we should send in the master opts
- **PR** #39852: (rallytime) Back-port #39651 to 2016.11 @ *2017-03-06 21:18:34 UTC*
  - **PR** #39651: (DennisHarper) Checking Instance when calling a function that can return None (refs: #39852)
  - 8ecc719f90 Merge pull request #39852 from rallytime/bp-39651
  - bb5ddbe18c Checking instance exists in master._get_cached_minion_data when cache.fetch returns None
  - 79f2a7cbb7 Update __init__.py
  - e2a232921d Checking instance exists in master._get_cached_minion_data when cache.fetch returns None
  - 838774291d Update __init__.py
  - ff6f63e9dd Checking instance exists in master._get_cached_minion_data when cache.fetch returns None
  - 855f87554c Checking instance exists in master._get_cached_minion_data when cache.fetch returns None
- **ISSUE** #39052: (githubcdr) Minion restart very slow since 2016.11.2 (refs: #39104)
- **PR** #39851: (rallytime) Back-port #39104 to 2016.11 @ *2017-03-06 21:17:43 UTC*
  - **PR** #39104: (githubcdr) Do not use name resolving for –notrim check (refs: #39851)
  - 897275ae5f Merge pull request #39851 from rallytime/bp-39104
  - 6539dbdbca Do not use name resolving for –notrim check
- **ISSUE** #38231: (tjuup) Typo: salt-key deleteed (refs: #39799)
- **PR** #39799: (Ch3LL) Fix deleteed message when key is deleted @ *2017-03-03 05:17:43 UTC*
  - d0440e2a2a Merge pull request #39799 from Ch3LL/fix_salt_key_msg
  - 8346682cf7 Fix deleteed message when key is deleted
- **ISSUE** #38962: (gstachowiak) Broken /jobs in salt-api in salt 2016.11.1 (Carbon) (refs: #39472)
- **PR** #39472: (whiteinge) Update _reformat_low to not run kwarg dicts through parse_input @ *2017-03-02 17:46:20 UTC*
  - 9f70ad7164 Merge pull request #39472 from whiteinge/_reformat_low-update
  - d11f5381a4 Add RunnerClient test for old/new-style arg/kwarg parsing
  - ec377ab379 Reenable skipped RunnerClient tests
  - 27f7fd9ad4 Update _reformat_low to run arg through parse_input
  - 5177153459 Revert parse_input change from #32005
- **PR** #39727: (terminalmage) salt.modules.state: check gathered pillar for errors instead of in-memory pillar @ *2017-03-02 17:06:43 UTC*

  - 7dfc4b572a Merge pull request #39727 from terminalmage/issue39627
  - 3bb0ebd872 Update tests for PR 39727
  - c334b59c96 salt.modules.state: check gathered pillar for errors instead of in-memory pillar
  - 97dd8a13d9 Ensure that ext_pillar begins with pillar_override if ext_pillar_first is True
  - f951266944 Add log message for successful makostack processing
- **ISSUE** #39775: (mirceaulinic) Proxy *mine_interval* config ignored (refs: #39935, #saltstack/salt‘#39935‘_, #39776)
- **PR** #39776: (mirceaulinic) WIP: Save _schedule.conf under <proxy ID> dir @ *2017-03-02 16:27:45 UTC*
  - 965f474316 Merge pull request #39776 from cloudflare/proxy-schedule
  - 35b8b8fd64 Save _schedule.conf under <minion ID> dir
- **PR** #39788: (cachedout) Disable one API test that is flaky @ *2017-03-02 16:17:31 UTC*
  - 555f1473f6 Merge pull request #39788 from cachedout/disable_api_test
  - 523e377b33 Disable one API test that is flaky
- **PR** #39762: (terminalmage) Fix regression in file.get_managed @ *2017-03-02 02:59:34 UTC*
  - 793979cbe6 Merge pull request #39762 from terminalmage/issue39751
  - 64db0b8563 Add integration tests for remote file sources
  - f9f894d981 Fix regression in file.get_managed when skip_verify=True
  - 28651a6699 Remove next(iter()) extraction
- **ISSUE** #35088: (Modulus) salt/cloud/ec2.py encoding problems. (refs: #37912)
- **PR** #39767: (rallytime) Back-port #38316 to 2016.11 @ *2017-03-02 02:54:57 UTC*
  - **PR** #38316: (mlalpho) salt utils aws encoding fix (refs: #39767)
  - **PR** #37912: (attiasr) fix encoding problem aws responses (refs: #38316)
  - 91a9337ab3 Merge pull request #39767 from rallytime/bp-38316
  - 1dcf018df7 requests api says Response.encoding can sometimes be None http://docs.python-requests.org/en/master/api/#requests.Response.encoding and result.text.encode() doesn’t accept None and expects a string.
- **ISSUE** #39692: (djsly) tuned module and state are broken on 7.3 families. (refs: #40387, #39719, #39768)
- **PR** #39768: (rallytime) Back-port #39719 to 2016.11 @ *2017-03-02 02:54:40 UTC*
  - **PR** #39719: (Seb-Solon) Support new version of tuned-adm binary (refs: #39768)
  - 4a01bd64ea Merge pull request #39768 from rallytime/bp-39719
  - d7cb70f203 Enh: Support new version of tuned-adm binary
- **PR** #39760: (Ch3LL) Initial 2016.11.4 Release Notes Doc @ *2017-03-01 18:43:39 UTC*
  - 780457f934 Merge pull request #39760 from Ch3LL/2016.11.4_notes
  - 1853c998c4 add initial 2016.11.4 release notes
- **PR** #39731: (twangboy) Add docs for Kwargs in pkg.refresh_db @ *2017-02-28 22:02:59 UTC*
  - 0147f78ab5 Merge pull request #39731 from twangboy/win_pkg_docs
  - 423e6f7448 Add docs for Kwargs in pkg.refresh_db

- **ISSUE** #39710: (huangfupeng) schedule.add parameter can not use "after" (refs: #39734)
- **PR** #39734: (garethgreenaway) Missing parameter in the schedule.add function @ *2017-02-28 20:43:08 UTC*
  - fce2d184f3 Merge pull request #39734 from garethgreenaway/39710_missing_schedule_add_parameter
  - 63eb610245 Per #39710, missing parameter in the schedule.add function
- **PR** #39729: (rallytime) [2016.11] Bump latest release version to 2016.11.3 @ *2017-02-28 18:08:25 UTC*
  - 7b4865c058 Merge pull request #39729 from rallytime/update-release-ver-2016.11
  - b5a7111ad9 [2016.11] Bump latest release version to 2016.11.3
- **PR** #39721: (vutny) DOCS: add 2nd level header for advanced targeting methods @ *2017-02-28 17:57:46 UTC*
  - 47e494fe07 Merge pull request #39721 from vutny/doc-targeting
  - 1d86cf1161 DOCS: add 2nd level header for advanced targeting methods
- **ISSUE** #39683: (alankrita) Error in Saltstack's rest auth "Authentication module threw 'status' " (refs: #39711)
- **PR** #39711: (alankrita) Fix error in Saltstack's rest auth "Authentication module threw 'status' " @ *2017-02-28 15:56:09 UTC*
  - d39b679d82 Merge pull request #39711 from alankrita/fix-rest-eauth
  - ee426562a7 Fix error in Saltstack's rest auth "Authentication module threw 'status' "
- **PR** #39699: (techhat) Strip shabang line from rendered HTTP data @ *2017-02-28 00:05:01 UTC*
  - 3940321462 Merge pull request #39699 from techhat/httpshabang
  - 559eb93576 Strip shabang line from rendered HTTP data
- **PR** #39694: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-02-27 22:13:49 UTC*
  - 00f121eade Merge pull request #39694 from rallytime/merge-2016.11
  - 756f1de2d2 Merge branch '2016.3' into '2016.11'
    - 3f8b5e6733 Merge pull request #39487 from bobrik/mode-docs
      - 41ef69b3ca Document default permission modes for file module
    - f7389bf1f5 Merge pull request #39641 from smarsching/issue-39169-2016.3
      - 88c2d9a540 Fix return data structure for runner (issue #39169).
    - fc970b6a16 Merge pull request #39633 from terminalmage/fix-systemd-typo
      - ca54541abe Add missing unit test for disable func
      - 17109e1522 Fix misspelled argument in salt.modules.systemd.disable()
    - 53e78d67f6 Merge pull request #39613 from terminalmage/fix-docs
      - 9342eda377 Fix inaccurate documentation
- **ISSUE** #39642: (drawsmcgraw) boto_vpc.nat_gateway_present does not honor the allocation_id parameter like the module does (refs: #39643)
- **PR** #39643: (drawsmcgraw) issue 39642 - boto_vpc.nat_gateway_present should accept parameter al… @ *2017-02-27 20:19:09 UTC*
  - 2c919e31d6 Merge pull request #39643 from drawsmcgraw/39642
  - 56d9adfbf6 issue 39642 - boto_vpc.nat_gateway_present should accept parameter allocation_id.
- **PR** #39666: (terminalmage) Rewrite the test_valid_docs test @ *2017-02-26 20:14:33 UTC*

  - df013c5f31 Merge pull request #39666 from terminalmage/test_valid_docs
  - 5a3c099e4f Rewrite the tests_valid_docs test
- **PR** #39662: (The-Loeki) Py3 compat: Force minions to be a list for local serialized caches @ *2017-02-26 02:36:46 UTC*
  - a29a7be7f8 Merge pull request #39662 from The-Loeki/py3cachefix
  - b02ef984f7 Add comment
  - 0fe5c90a05 Py3 compat: Force minions to be a list for local serialized caches
- **PR** #39644: (vutny) Improve and align dockerng execution module docs @ *2017-02-25 04:16:28 UTC*
  - bd6efd18b1 Merge pull request #39644 from vutny/dockerng-docs
  - c4988e874e Improve and align dockerng execution module docs
- **PR** #39516: (jettero) Prevent spurious "Template does not exist" error @ *2017-02-24 23:41:36 UTC*
  - fffab54078 Merge pull request #39516 from jettero/give-pillarenv-tops-similar-treatment
  - 8fe48fa5c4 prevent billions of inexplicable lines of this:
- **PR** #39654: (skizunov) Fix issue where compile_pillar failure causes minion to exit @ *2017-02-24 22:47:52 UTC*
  - be9629b180 Merge pull request #39654 from skizunov/develop2
  - 9f80bbce07 Fix issue where compile_pillar failure causes minion to exit
- **PR** #39653: (cachedout) Use salt's ordereddict for comparison @ *2017-02-24 22:46:24 UTC*
  - e63cbbaab9 Merge pull request #39653 from cachedout/26_odict
  - 91eb7210bb Use salt's ordereddict for comparison
- **ISSUE** #38836: (toanctruong) file.managed with S3 Source errors out with obscure message (refs: #39609, #39589)
- **PR** #39609: (gtmanfred) intialize the Client stuff in FSClient @ *2017-02-24 18:50:55 UTC*
  - 0bc6027e68 Merge pull request #39609 from gtmanfred/2016.11
  - 0820620ef8 intialize the Client stuff in FSClient
- **PR** #39615: (skizunov) Bonjour/Avahi beacons: Make sure TXT record length is valid @ *2017-02-24 18:47:05 UTC*
  - 28035c07b3 Merge pull request #39615 from skizunov/develop2
  - b1c7e9b505 Bonjour/Avahi beacons: Make sure TXT record length is valid
- **PR** #39617: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-02-24 16:07:55 UTC*
  - e9410fb669 Merge pull request #39617 from rallytime/merge-2016.11
  - 13622899d2 Merge branch '2016.3' into '2016.11'
  - 4e2b852f83 Merge pull request #39600 from vutny/state-file-docs
    - 9b0427c27a state.file: drop non-relevant examples for *source_hash* parameter
  - ed83420417 Merge pull request #39584 from cachedout/mentionbot_docs
    - 652044b18f A note in the docs about mentionbot
  - d3e50b4f2f Merge pull request #39583 from cachedout/mentionbot_blacklist
    - 62491c900d Add empty blacklist to mention bot

- **ISSUE** #38758: (bobrik) Remote state execution is much slower on 2016.11.1 compared to 2016.3.4 (refs: #39505)
- **ISSUE** #33575: (anlutro) File states seem slower in 2016.3, especially on first cache retrieval (refs: #33896)
- **ISSUE** #29643: (matthayes) Can't get batch mode and –failhard to work as expected (refs: #31164)
- **ISSUE** #28569: (andrejohansson) Reactor alert on highstate fail (refs: #31164)
- **PR** #39505: (cachedout) Threadsafety option for context dictionaries @ *2017-02-23 19:38:13 UTC*
  - **PR** #37378: (skizunov) Fix *__context__* to properly sandbox (refs: #39505)
  - **PR** #33896: (DmitryKuzmenko) Don't deep copy context dict values. (refs: #39505)
  - **PR** #31164: (DmitryKuzmenko) Issues/29643 fix invalid retcode (refs: #33896)
  - 0d31201e08 Merge pull request #39505 from cachedout/issue_38758
  - 1dba2f9cb0 Add warning in docs
  - 9cf654b72c Threadsafety option for context dictionaries
- **PR** #39507: (joe-niland) Detect IIS version and vary certificate association command depending on version @ *2017-02-23 19:15:40 UTC*
  - c0d4357f46 Merge pull request #39507 from joe-niland/iis-7-cert-binding
  - c94f0b8c62 Fix additional issue whereby existing certificate bindings were not found in IIS 7.5, due to the fact that IIS earlier than 8 doesn't support SNI
  - 18effe0103 Detect IIS version and vary certificate association command depending on version
- **PR** #39565: (terminalmage) states.file.patch/modules.file.check_hash: use hash length to determine type @ *2017-02-23 19:14:28 UTC*
  - e6f5e8a474 Merge pull request #39565 from terminalmage/issue39512
  - cbdf905b9f Update test to reflect new state comment
  - 650dbaca4e states.file.patch/modules.file.check_hash: use hash length to determine type
- **PR** #39591: (mcalmer) fix case in os_family for Suse @ *2017-02-23 19:07:17 UTC*
  - 53e22b8f15 Merge pull request #39591 from mcalmer/fix-case-in-os_family
  - 81bd96e32d fix case in os_family for Suse
- **ISSUE** #38452: (jf) file.line with mode=delete does not preserve ownership of a file (refs: #39592)
- **PR** #39592: (skazi0) Ensure user/group/file_mode after line edit @ *2017-02-23 18:40:05 UTC*
  - aee43f7fa4 Merge pull request #39592 from skazi0/line-user-fix
  - baf84b4430 Ensure user/group/file_mode after line edit
- **PR** #39596: (ticosax) Reduce scope of try except StopIteration wrapping @ *2017-02-23 18:16:17 UTC*
  - 6ab4151213 Merge pull request #39596 from ticosax/reduce-scope-catehed-exception
  - 54cdacb680 Reduce scope of try except StopIteration wrapping
- **ISSUE** #38836: (toanctruong) file.managed with S3 Source errors out with obscure message (refs: #39609, #39589)
- **PR** #39610: (rallytime) Back-port #39589 to 2016.11 @ *2017-02-23 17:48:03 UTC*
  - **PR** #39589: (MasterNayru) Allow masterless minions to pull files from S3 (refs: #39610)
  - b1c3b84862 Merge pull request #39610 from rallytime/bp-39589

  - 83ec174d44 Set utils property explicitly for FSClient
  - 3889006149 Allow masterless minions to pull files from S3
- **PR** #39606: (rallytime) [2016.11] Pylint: add missing import @ *2017-02-23 16:39:55 UTC*
  - fe15ed9b92 Merge pull request #39606 from rallytime/lint-2016.11
  - 71164348e7 [2016.11] Pylint: add missing import
- **PR** #39573: (thatch45) Added a few more comments to the ssl docs @ *2017-02-23 02:17:13 UTC*
  - **PR** #39554: (DmitryKuzmenko) Cosmetic: support bool value for 'ssl' config option. (refs: #39573)
  - **PR** #39528: (thatch45) Add better ssl option docs (refs: #39554)
  - 5987c4e30e Merge pull request #39573 from thatch45/ssl_docs
  - b230c35eac This should be good to go now
- **PR** #39577: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-02-23 02:10:12 UTC*
  - b8e321cbec Merge pull request #39577 from rallytime/merge-2016.11
  - 397c756a01 Merge branch '2016.3' into '2016.11'
    - 8352e6b44b Merge pull request #39579 from rallytime/fix-lint
      - 65889e1f30 [2016.3] Pylint: Remove unused import
    - 43dba3254c Merge pull request #39578 from cachedout/2016.3
      - 344499eef7 Add mention-bot configuration
  - c52cecd856 Fix syntax error leftover from incomplete merge-conflict resolution
  - 7b9b3f700d Merge branch '2016.3' into '2016.11'
    - 8f7a0f9d96 Merge pull request #39542 from twangboy/gate_ssh_known_hosts
      - c90a52ef27 Remove expensive check
      - 6d645cae0e Add __virtual__ function
    - c10965833a Merge pull request #39289 from bobrik/autodetect-ipv6
      - 2761a1b244 Move new kwargs to the end of argument list
      - 0df6b922e7 Narrow down connection exception to socket.error
      - e8a2cc0488 Do no try to connect to salt master in syndic config test
      - af9578631e Properly log address that failed to resolve or pass connection check
      - 9a34fbeba9 Actually connect to master instead of checking route availability
      - c494839c65 Avoid bare exceptions in dns_check
      - 29f376676d Rewrite dns_check to try to connect to address
      - 55965ce505 Autodetect IPv6 connectivity from minion to master
    - 3fb928b63a Merge pull request #39569 from s0undt3ch/2016.3
      - 49da135abd Don't use our own six dictionary fixes in this branch
    - 91e3319df8 Merge pull request #39508 from dincamihai/openscap
      - 9fedb84607 Always return oscap's stderr
      - 0ecde2cd02 Include oscap returncode in response

    - fbe2194a93 Merge pull request #39562 from terminalmage/issue30802
      - c50374041d Add ulimits to dockerng state/exec module
      - da42040c1a Try the docker-py 2.0 client name first
    - 01d4a84a2f dockerng.get_client_args: Fix path for endpoint config for some versions of docker-py (#39544)
- **PR** #39574: (Ch3LL) Update 2016.11.3 release notes @ *2017-02-23 00:10:23 UTC*
  - cff9334929 Merge pull request #39574 from Ch3LL/update_release_notes
  - c0f8c35df7 fix reference to set in docs
  - 663f6f159d add additional PRs to 2016.11.3 release notes
- **PR** #39528: (thatch45) Add better ssl option docs (refs: #39554) @ *2017-02-22 18:29:47 UTC*
  - b492f7094c Merge pull request #39528 from thatch45/ssl_docs
  - c357e37831 Add minion config
  - 539bb2aa80 Add better ssl option docs
- **ISSUE** saltstack/salt#35869: (amontalban) timezone.system state fails on FreeBSD when /etc/localtime does not exists (refs: #39532)
- **PR** #39532: (amontalban) Fix case when /etc/localtime is a file and it is not updated @ *2017-02-22 18:28:54 UTC*
  - 0dad49cdff Merge pull request #39532 from amontalban/corner_case_35869
  - f0d3c16547 Fix case when /etc/localtime is a file and it is not updated
- **PR** #39540: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-02-22 18:24:01 UTC*
  - 9cfaf3b599 Merge pull request #39540 from rallytime/merge-2016.11
  - 49fe4e891e Merge branch '2016.11' into '2016.11'
  - c613d19e76 Merge branch '2016.3' into '2016.11'
  - dff35b58f8 Merge pull request #39498 from terminalmage/pr-39483
    - 20b097a745 dockerng: compare sets instead of lists of security_opt
  - 6418e725ed Merge pull request #39497 from terminalmage/docker-compat-fixes
    - cbd0270bac docker: make docker-exec the default execution driver
    - a6a17d58aa Handle docker-py 2.0's new host_config path
  - 9c4292fb4e Merge pull request #39423 from dincamihai/openscap
    - 9d13422ac1 OpenSCAP module
  - 7dd2502360 Merge pull request #39464 from gtmanfred/2016.3
    - f829d6f9fc skip false values from preferred_ip
  - db359ff2c3 Merge pull request #39460 from cachedout/win_dism_test_fix
    - e652a45592 Fix mocks in win_disim tests
  - 9dbfba9b57 Merge pull request #39426 from morganwillcock/dism
    - a7d5118262 Return failure when package path does not exist
  - 56162706e3 Merge pull request #39431 from UtahDave/fix_grains.setval_performance
    - 391bbecd90 add docs

        - 709c197f84 allow sync_grains to be disabled on grains.setval
    - 239e16e612 Merge pull request #39405 from rallytime/fix-39304
        - bd1fe03ce7 Update :depends: docs for boto states and modules
    - 415102f346 Merge pull request #39411 from rallytime/fix-38762
        - e13febe58d Update external_cache docs with other configuration options
    - 7e1803b617 Update docs on upstream EPEL7 pygit2/libgit2 issues (#39421)
- **PR** #39554: (DmitryKuzmenko) Cosmetic: support bool value for 'ssl' config option. (refs: #39573) @ *2017-02-22 16:59:03 UTC*
    - **PR** #39528: (thatch45) Add better ssl option docs (refs: #39554)
    - 56fe2f198e Merge pull request #39554 from DSRCorporation/bugs/ssl_bool
    - 7a6fc11291 Cosmetic: support bool value for 'ssl' config option.
- **PR** #39560: (vutny) [CLOUD] Log error when private/public IP was not detected @ *2017-02-22 16:49:46 UTC*
    - cf37f83565 Merge pull request #39560 from vutny/cloud-detect-ips
    - 567bb50884 [CLOUD] Log error when private/public IP was not detected

28.2.36 Salt 2016.11.5 Release Notes

Version 2016.11.5 is a bugfix release for *2016.11.0*.

Statistics

- Total Merges: **82**
- Total Issue References: **23**
- Total PR References: **80**
- Contributors: **32** (BenoitKnecht, Ch3LL, DmitryKuzmenko, Enquier, SolarisYan, UtahDave, alexproca, benediktwerner, bobrik, brd, cachedout, clinta, corywright, cro, danlsgiga, drawsmcgraw, ezh, gtmanfred, isbm, jf, jleproust, lorengordon, nevins-b, oeuftete, peter-funktionIT, rallytime, rkgrunt, senthilkumar-e, sjorge, skizunov, terminalmage, twangboy)

Patched Packages

Due to the critical nature of issue issue #41230 we have decided to patch the 2016.11.5 packages with PR #41244. This issue affects all calls to a salt-minion if there is an ipv6 nameserver set on the minion's host. The patched packages on repo.saltstack.com will divert from the v2016.11.5 tag and pypi packages due to the patches applied to the packages.

Changelog for v2016.11.4..v2016.11.5

Generated at: 2018-05-27 20:12:47 UTC

- **PR** #41134: (twangboy) Fix *pkg.install* on Windows on 2016.11 @ *2017-05-09 15:10:19 UTC*
    - a10f0146a4 Merge pull request #41134 from twangboy/fix_get_msiexec
    - d808a60129 Remove redundant if statement

  - b4d6d5a927 Fix for version_num of None and Latest
  - 0f31822a83 Fix problem when use_msiexec is a bool
- **ISSUE** #41100: (frogunder) Exception occurred in runner jobs.list_jobs (refs: #41102)
- **PR** #41102: (gtmanfred) don't pass jid to list_jobs @ *2017-05-08 17:45:40 UTC*
  - 4ecab68bb9 Merge pull request #41102 from gtmanfred/2016.11
  - 83057d0f0f don't pass jid to list_jobs
  - 505cb45722 Merge branch 'fix-file-blockreplace-diff-in-test-mode' of https://github.com/L4rS6/salt into L4rS6-fix-file-blockreplace-diff-in-test-mode
  - de9f66b448 show chanages in file.blockreplace function in testing mode. also used same programming style as in file.managed function: (ret['changes']['diff'] = ret['pchanges']['diff'])
- **PR** #41103: (lorengordon) Adds a get_route() function to win_network.py @ *2017-05-06 06:19:42 UTC*
  - 2af89beb53 Merge pull request #41103 from lorengordon/win.get_route
  - 93ce5644ea Adds test for win_network.get_route
  - b9cbbc0290 Adds a get_route() function to win_network.py
- **PR** #41098: (rallytime) Back-port #41088 to 2016.11 @ *2017-05-05 19:04:03 UTC*
  - **PR** #41088: (sjorge) Fix docs for zfs state module (refs: #41098)
  - 2f9b5a4074 Merge pull request #41098 from rallytime/bp-41088
  - dc6cd2ea45 Fix docs for zfs state module
- **PR** #41097: (rallytime) Back-port #41079 to 2016.11 @ *2017-05-05 19:03:43 UTC*
  - **PR** #41079: (brd) Remove an extra colon that is causing rendering issues (refs: #41097)
  - 2123001f32 Merge pull request #41097 from rallytime/bp-41079
  - 845b49c304 Remove and extra colon that is causing rendering issues
- **PR** #41093: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-05-05 17:22:09 UTC*
  - ff6fa2b120 Merge pull request #41093 from rallytime/merge-2016.11
  - a670eaa1db Merge branch '2016.3' into '2016.11'
  - 247400c44e Merge pull request #41083 from rallytime/git-state-fix
    - b85ee48ff4 Git state: head_ref should be head_rev in "latest" function
- **PR** #41084: (rallytime) Skip the test_salt_documentation_arguments_not_assumed test for Arch @ *2017-05-04 21:56:29 UTC*
  - **PR** #41074: (rallytime) Skip integration.shell.matcher.MatchTest.test_salt_documentation test for Arch (refs: #41084)
  - 4c2e636cd1 Merge pull request #41084 from rallytime/disable-matcher-test-arch
  - da811fe505 Skip the correct test for the matcher tests in Arch
  - b9d1ce9aed Revert "Skip integration.shell.matcher.MatchTest.test_salt_documentation test for Arch"
- **PR** #41069: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-05-04 20:34:58 UTC*
  - 1179720327 Merge pull request #41069 from rallytime/merge-2016.11
  - 08c58919cb Merge branch '2016.3' into '2016.11'

        - 69418092bd Merge pull request #41070 from rallytime/lint-2016.3
            - 486e2ba62e Pylint: remove extra line in mac_system module
    - db70b2d42e Pylint: remove extra line in mac_system module
    - 855d157aa6 Merge branch '2016.3' into '2016.11'
        - 3101694d71 Merge pull request #41048 from willkil/mac_system_non_root
            - b65b82a750 mac_system: return False for non-root user
- **PR** #41074: (rallytime) Skip integration.shell.matcher.MatchTest.test_salt_documentation test for Arch (refs: #41084) @ *2017-05-04 19:26:16 UTC*
    - 9d638abc62 Merge pull request #41074 from rallytime/disable-matcher-test-arch
    - 9eb482d5c7 Skip integration.shell.matcher.MatchTest.test_salt_documentation test for Arch
- **PR** #41078: (Ch3LL) Add 2016.11.5 release notes and change log @ *2017-05-04 19:00:58 UTC*
    - 72c854d9ac Merge pull request #41078 from Ch3LL/add_2016.11.5_release
    - 96ed815687 Add 2016.11.5 release notes and change log
- **PR** #40879: (peter-funktionIT) Update win_pki.py @ *2017-05-04 16:12:00 UTC*
    - eac8401e90 Merge pull request #40879 from peter-funktionIT/2016.11
    - 80fa9e5b76 Update win_pki.py
    - a48b05f158 Update win_pki.py
    - 3a4e6d9d91 Update win_pki.py
- **ISSUE** #40928: (sokratisg) Orchestration runner, highstate and environment question (refs: #41036)
- **PR** #41036: (terminalmage) Do not force effective saltenv when running states via orchestration @ *2017-05-04 15:44:14 UTC*
    - 547a9738db Merge pull request #41036 from terminalmage/issue40928
    - 72ef34c420 Do not force effective saltenv when running states via orchestration
- **PR** #41039: (terminalmage) Look for currently-running python's pip first @ *2017-05-04 15:43:52 UTC*
    - 6e2458e171 Merge pull request #41039 from terminalmage/improve-pip-bin
    - effe8b9432 Look for currently-running python's pip first
- **PR** #41049: (Ch3LL) fix integration wheel test_gen test @ *2017-05-04 15:33:59 UTC*
    - ff39613a53 Merge pull request #41049 from Ch3LL/fix_wheel_test
    - ba223827b9 fix integration wheel test_gen test
- **PR** #41054: (terminalmage) Update package targets for Arch pkg tests @ *2017-05-04 14:59:42 UTC*
    - 4e4b3514b4 Merge pull request #41054 from terminalmage/salt-jenkins-315
    - ee493bae47 Update package targets for Arch pkg tests
- **PR** #41046: (twangboy) Fix pkg.remove @ *2017-05-04 14:58:57 UTC*
    - 62dff52820 Merge pull request #41046 from twangboy/fix_pkg_remove
    - 2af38e5564 Use target instead of version_num
- **PR** #41045: (terminalmage) Clarify gitfs docs @ *2017-05-03 22:24:55 UTC*
    - 2b47b7bec6 Merge pull request #41045 from terminalmage/clarify-gitfs-docs

  - c757eda331 Clarify gitfs docs

- **PR** #41032: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-05-03 19:31:58 UTC*
  - 819007cd00 Merge pull request #41032 from rallytime/merge-2016.11
  - d26fd0bbf4 Merge branch '2016.3' into '2016.11'
    - b00acb0034 Merge pull request #41011 from terminalmage/docker-refresh-credentials
      - b8d1dcc307 Use proposed docker-py reload_config() func
- **ISSUE** #35699: (jleproust) LVM state fails to add new device, volume group name is empty string (refs: #41007)
- **PR** #41007: (jleproust) Recognize LVM2 pv with empty vg as orphan @ *2017-05-03 18:24:51 UTC*
  - d7fbd38474 Merge pull request #41007 from jleproust/fix_lvm_empty_vg
  - 3b9a845145 Recognize LVM2 pv with empty vg as orphan
- **PR** #41029: (rallytime) Back-port #38565 to 2016.11 @ *2017-05-03 17:05:10 UTC*
  - **PR** #38565: (drawsmcgraw) Update management of ip addresses for salt cloud azurearm module (refs: #41029)
  - 4eab962e9e Merge pull request #41029 from rallytime/bp-38565
  - 2df93ae3ab Update management of ip addresses. - Assign static, private IP addresses. - Ability to not assign a public IP to a VM.
- **PR** #41012: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-05-02 22:06:40 UTC*
  - 97500f078d Merge pull request #41012 from rallytime/merge-2016.11
  - fc756c595c Merge branch '2016.3' into '2016.11'
  - 19894f68ca Merge pull request #40724 from cro/minion_key_revoke_cfg
    - cbc70195c0 Change message level when minion requests key revoke and feature is turned off.
    - 65ea8997b7 Add allow_minion_key_revoke config option
    - 8920495943 Add config option to prevent minions from revoking their own keys.
  - 129859f79b Merge pull request #40952 from terminalmage/fix-dockerng.login-docs
    - dfbbeb5946 Fix documentation for docker login function in pre-nitrogen release branches
- **PR** #40726: (benediktwerner) Fixed minion keys remaining pending after auto signing and fixed typo (Resubmitted to 2016.3) @ *2017-05-02 16:57:34 UTC*
  - e210eaead4 Merge pull request #40726 from benediktwerner/fix-minions-remain-pending-after-autosign-and-typo
  - 82c144e960 Fixed minion keys remaining pending after auto signing and fixed typo
- **PR** #40960: (danlsgiga) Fix consul module "AttributeError: 'dict' object has no attribute 'json'" @ *2017-05-02 16:16:57 UTC*
  - 4f342e2fe5 Merge pull request #40960 from danlsgiga/2016.11
  - 6e4cc6db47 Fix consul module "AttributeError: 'dict' object has no attribute 'json'"
- **PR** #40963: (twangboy) Fix fullname parameter for add function @ *2017-05-02 16:08:59 UTC*
  - c3b329b398 Merge pull request #40963 from twangboy/fix_win_useradd
  - 5371b6b85e Fix fullname parameter for add function
- **PR** #40995: (twangboy) Remove unused code fragments @ *2017-05-02 15:31:58 UTC*

  - d79c033239 Merge pull request #40995 from twangboy/remove_utils
  - 8c01aacd9b Remove unused code fragments
- **ISSUE** #40981: (ezh) docker-events engine is broken with modern docker (refs: #40982)
- **PR** #40991: (rallytime) Back-port #40982 to 2016.11 @ *2017-05-01 22:31:30 UTC*
  - **PR** #40982: (ezh) Fix docker_events field handling (refs: #40991)
  - c6162876d6 Merge pull request #40991 from rallytime/bp-40982
  - 8fcb7205db Fix docker_events field handling
- **ISSUE** #40965: (weirdbricks) salt-cloud sudo failing (refs: #40987)
- **PR** #40987: (gtmanfred) get sudo_password correctly @ *2017-05-01 19:39:55 UTC*
  - 3fb24929c6 Merge pull request #40987 from gtmanfred/2016.11
  - 2ed694cac6 get sudo_password correctly
- **ISSUE** #40988: (santzi) status.netdev tx_bytes is always zero (refs: #40992)
- **PR** #40992: (gtmanfred) fix bug in status.netdev @ *2017-05-01 19:38:35 UTC*
  - ecbac138d1 Merge pull request #40992 from gtmanfred/netdev
  - a9eed7f1c9 fix bug in status.netdev
- **ISSUE** #40976: (sjorge) smtp.send_msg state oddities (refs: #40993)
- **PR** #40993: (gtmanfred) smtp state can use profile or sender @ *2017-05-01 19:35:47 UTC*
  - d852320d34 Merge pull request #40993 from gtmanfred/smtp
  - 068ebfd9ec smtp state can use profile or sender
- **PR** #40958: (rallytime) Back-port #40939 to 2016.11 @ *2017-04-28 18:01:17 UTC*
  - **PR** #40939: (Ch3LL) Allow vmware to query deploy arg from opts (refs: #40958)
  - fc26fb8a05 Merge pull request #40958 from rallytime/bp-40939
  - 3e9394862f allow vmware to query deploy arg from opts
- **ISSUE** saltstack/salt#34640: (nevins-b) utils.shlex_split removing quotes which are required for augeas (refs: #‘saltstack/salt#34643‘_)
  - **PR** saltstack/salt#34643: (nevins-b) fix augeas module so shlex doesn’t strip quotes (refs: #38115)
- **PR** #40957: (rallytime) Back-port #38115 to 2016.11 @ *2017-04-28 18:01:02 UTC*
  - **PR** #38115: (cro) Revert ”fix augeas module so shlex doesn’t strip quotes” (refs: #40957)
  - a586e12180 Merge pull request #40957 from rallytime/bp-38115
  - eb889173b0 Revert ”fix augeas module so shlex doesn’t strip quotes”
- **ISSUE** #40635: (promorphus) Orchestrate + Batches returns false failed information (refs: #40905)
- **PR** #40905: (rkgrunt) Fixed issue with parsing of master minion returns when batching is en… @ *2017-04-28 17:52:32 UTC*
  - 00a15eba60 Merge pull request #40905 from rkgrunt/40635
  - 4f9c92a012 Fixed issue with parsing of master minion returns when batching is enabled.
- **PR** #40954: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-04-28 16:41:49 UTC*
  - bb50d4f646 Merge pull request #40954 from rallytime/merge-2016.11

  - 7f31e41aa6 Merge branch '2016.3' into '2016.11'
  - 55a399583e Merge pull request #40930 from rallytime/bp-40811
    - 3ccb553f9f get config_dir based off conf_file
  - 7bc01be859 Merge pull request #40927 from terminalmage/docs
    - 8c078f144c Add additional note about quoting within load_yaml
    - 123b5cdc11 Add documentation for PyYAML's loading of time expressions
  - 7eab9c6cf4 Merge pull request #40891 from terminalmage/pip-installed
    - 75e6bc0aa3 Fix two issues with pip.install
- **ISSUE** #39531: (ypid) Use yaml.safe_* instaed of yaml.load / yaml.dump / … (refs: #40751)
- **PR** #40751: (rallytime) Use Salt's SaltYamlSafeLoader and SafeOrderedDumper classes for yaml.load/dump @ *2017-04-28 12:56:06 UTC*
  - 909d519ddb Merge pull request #40751 from rallytime/fix-39531
  - 85dc4164f5 Don't change the salt.utils.jinja yaml Dumper class
  - 4fe6ac93c6 Add extra line for lint
  - 55cfa12975 Use salt.utils.yamldumper with SafeOderedDumper as the Dumper in yaml.dump
  - 62c4d37c2f Use salt.utils.yamlloader with SaltYamlSafeLoader as the Loader with yaml.load
- **ISSUE** #37307: (szjur) Minions run every job twice and open 2 connections to the same syndic - apparently after reconnection between masters (refs: #40861)
- **PR** #40861: (DmitryKuzmenko) Don't run status.master while minion is failing-over. @ *2017-04-28 12:14:56 UTC*
  - 18fdd8cc34 Merge pull request #40861 from DSRCorporation/bugs/37307_minion_run_jobs_twice
  - f0d46d04af Don't run status.master while minion is failing-over.
- **PR** #40923: (terminalmage) aptpkg: fix temp pkg unhold when version is specified @ *2017-04-28 11:59:54 UTC*
  - 62cb7b1ae6 Merge pull request #40923 from terminalmage/aptpkg-install-fix-unhold
  - 6dda4f2bc3 aptpkg: fix temp pkg unhold when version is specified
- **ISSUE** #40908: (nicksloan) If master_port is a string the minion cannot connect and prints an unhelpful error message (refs: #40933)
- **ISSUE** #39118: (bobrik) Minion ipv6 option is not documented (refs: #39289)
- **PR** #40933: (gtmanfred) allow master_port to be a string @ *2017-04-28 11:54:58 UTC*
  - **PR** #39289: (bobrik) Autodetect IPv6 connectivity from minion to master (refs: #40933)
  - 9d92ba7878 Merge pull request #40933 from gtmanfred/2016.11
  - 194423c08e allow master_port to be a string
- **ISSUE** #40912: (razed11) IPV6 Warning when ipv6 set to False (refs: #40934)
- **PR** #40934: (gtmanfred) Only display IPvX warning if role is master @ *2017-04-28 11:53:50 UTC*
  - d5e0b8b655 Merge pull request #40934 from gtmanfred/ipv6
  - 7855cd6ce6 Only display IPvX warning if role is master
- **ISSUE** #40881: (stamak) 2016.11 SoftLayer salt-cloud driver connects on private IP instead of public IP (refs: #40935)

- **PR** #40935: (gtmanfred) Attempt to connect to public ip address in softlayer @ *2017-04-28 11:43:57 UTC*
    - 8fdfe4ece6 Merge pull request #40935 from gtmanfred/softlayer
    - d6eb11410f Attempt to connect to public ip address in softlayer
- **PR** #40936: (terminalmage) Add dockerng fixes to 2016.11.4 release notes @ *2017-04-27 19:54:16 UTC*
    - 7404309bec Merge pull request #40936 from terminalmage/release_notes
    - e494ae43e5 Add dockerng fixes to 2016.11.4 release notes
- **ISSUE** #33093: (gtmanfred) [salt-cloud][nova] race condition when assigning floating ips to cloud servers (refs: #37696)
- **PR** #40929: (rallytime) Back-port #37696 to 2016.11 @ *2017-04-27 17:43:26 UTC*
    - **PR** #37696: (SolarisYan) if vm state is not ACTIVE, it will fail (refs: #40929)
    - a622518ad2 Merge pull request #40929 from rallytime/bp-37696
    - 1a28722c5a Pylint fix
    - 8e0a9864c5 if vm state is not ACTIVE, associate floating ip to it will fail.So we should wait for state of vm is ACTIVE,then associate the assigned floating ip to it
- **PR** #40921: (corywright) Make salt.auth.rest heading consistent with all other salt.auth documentation @ *2017-04-27 17:36:47 UTC*
    - f88ce8e4de Merge pull request #40921 from corywright/consistent-salt-auth-headings
    - 2995a05c2b Make salt.auth.rest heading consistent with all other salt.auth documentation
- **ISSUE** #37824: (dxiri) SSLError Trying to use v3 API of Openstack Newton as provider. (refs: #40752)
- **PR** #40752: (Enquier) Add ability to specify a custom SSL certificate or disable SSL verification in KeystoneAuth v3 @ *2017-04-27 17:29:09 UTC*
    - 26be306b5c Merge pull request #40752 from Enquier/nova_ssl_2
    - 817f49296e fixing lint errors in keystone auth error
    - f683636c61 fix trailing whitespace
    - 4a70b8c0cc fixing minor error in security_groups security groups parser had incorrect split action which caused errors
    - c9d6f8e5ed adding note in documentation
    - c24dfe3fba adding support for cacert verification
    - bfaf5e322d Merge pull request #5 from saltstack/2016.11
- **ISSUE** #40845: (e-senthilkumar) /jobs call is broken in 2016.11.4 (refs: #40894)
- **PR** #40894: (senthilkumar-e) Fix for broken /jobs/<jid> in 2016.11.4 @ *2017-04-27 11:33:00 UTC*
    - 0f2ec1e1db Merge pull request #40894 from senthilkumar-e/broken_jobs_api_fix
    - 2f55b26e08 Fixing the pylint issue
    - fb607bab75 Fix for broken /jobs/<jid> in 2016.11.4
- **PR** #40876: (BenoitKnecht) states: sqlite3: fix table_present with multi-line schema @ *2017-04-26 15:21:19 UTC*
    - ea55c15367 Merge pull request #40876 from BenoitKnecht/fix-sqlite3-table-present-with-multiline-schema
    - 2ca627d02d states: sqlite3: fix table_present with multi-line schema

- **ISSUE** #40741: (clinta) Regression in 2016.11.3. File.managed downloads every time. (refs: #40742)
- **PR** #40742: (clinta) Fix #40741 @ *2017-04-25 22:52:06 UTC*
  - e09bafdceb Merge pull request #40742 from clinta/40741
  - 72bf5af9e6 Set sfn if cached_sum == source_sum
- **PR** #40859: (skizunov) Fix TCP Transport to work with Tornado 4.5 @ *2017-04-25 04:29:00 UTC*
  - 5249496f74 Merge pull request #40859 from skizunov/develop2
  - 958ecdace8 Fix TCP Transport to work with Tornado 4.5
- **PR** #40862: (gtmanfred) status should be an int @ *2017-04-24 23:11:31 UTC*
  - ca80f287af Merge pull request #40862 from gtmanfred/2016.11
  - 87ec1da771 status should be an int
- **PR** #40865: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-04-24 23:06:28 UTC*
  - c95341959d Merge pull request #40865 from rallytime/merge-2016.11
  - 53ad3159cc Merge branch '2016.3' into '2016.11'
  - 2a71dc3552 Merge pull request #40854 from Ch3LL/11.4_release_2016.3
    - 889540a313 [2016.3] Bump latest release version to 2016.11.4
  - b5f67f0750 Merge pull request #40822 from lordcirth/fix-rsync-changes
    - 1b304bb476 Extra space before inline comment
    - ea4592de91 rsync.py: Don't return changes when clean
- **PR** #40855: (Ch3LL) [2016.11] Bump latest release version to 2016.11.4 @ *2017-04-24 17:37:47 UTC*
  - 7861f12df8 Merge pull request #40855 from Ch3LL/11.4_release_2016.11
  - e7b604339d [2016.11] Bump latest release version to 2016.11.4
- **PR** #40817: (isbm) Some UT for cloud @ *2017-04-23 10:01:40 UTC*
  - 25b62aee47 Merge pull request #40817 from isbm/isbm-skip-false-values-from-preferred-ip-201611
  - 7c5714b90b Describe debug information
  - e0210ff8cb Reformat idents, fix typos
  - fb777e3f3e PEP8: fix unused variable
  - b2e85de85d Fix lint, typos and readability
  - 116c96a4b7 Fix UT parameter changes
  - 61558f08e7 Lintfix E0602
  - ed84420df0 Add unit test for node ip filtering
  - 82582cff77 Skip test, if libcloud is not around
  - f005d53c56 Fix name error exception
  - b668e60b4c Move out nested function for testing purposes
  - 5e574a24d9 Add unit test for nova connector
  - 181d0780d0 Lintfix
  - 8e9ce1a68d Move out nested function to be unit-testable

  - cd43805770 Add initial unit test for openstack cloud module
  - 177f31446d Add fake preferred IP function for testing
  - d1aeb13ac7 Move out openstack's nested function to be testable
- **PR** #40824: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-04-21 20:03:10 UTC*
  - 50ddf219a6 Merge pull request #40824 from rallytime/merge-2016.11
  - f31f9512b8 Merge branch '2016.3' into '2016.11'
    - 3b9ebeb98f Merge pull request #40754 from lordcirth/fix-uppercase-checksums
      - c80c792704 remove too many newlines for lint
      - a7d8f375e8 file.manage_file: uppercase checksums now work
- **PR** #40811: (UtahDave) get config_dir based off conf_file if __opts__['config_dir'] doesn't exist (refs: #40930) @ *2017-04-21 17:44:42 UTC*
  - d6e26d18cb Merge pull request #40811 from UtahDave/2016.11local
  - 9f6e2e9c92 get config_dir based off conf_file
- **PR** #40820: (gtmanfred) remove deprecated firstgen rackspace cloud driver @ *2017-04-21 17:42:19 UTC*
  - ddedf05b7d Merge pull request #40820 from gtmanfred/2016.11
  - b60a8d013a remove rackspace from index
  - 559aa1d8b6 remove deprecated firstgen rackspace cloud driver
- **PR** #40797: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-04-20 19:42:04 UTC*
  - 2ab42489df Merge pull request #40797 from rallytime/merge-2016.11
  - 22500a7261 Merge branch '2016.3' into '2016.11'
    - 623e2eb61f Merge pull request #40791 from a-powell/s3-util-get-memory-fix
      - 36f6521014 Merge remote-tracking branch 'upstream/2016.3' into s3-util-get-memory-fix
      - 04637cd4eb Fixing objects being loaded into memory when performing a GET request with a local file specified.
- **PR** #40800: (rallytime) Back-port #40720 to 2016.11 @ *2017-04-20 19:41:41 UTC*
  - **PR** #40720: (oeuftete) Call tornado.httputil.url_concat compatibly (refs: #40800)
  - ced839f841 Merge pull request #40800 from rallytime/bp-40720
  - 6c0124ae21 Call tornado.httputil.url_concat compatibly
- **ISSUE** #19137: (jeffclay) MSI installer(s) for windows minion (refs: #40716)
- **PR** #40785: (alexproca) win_pkg: backport 2016.11 add msiexec override to enable selection of 32 or 64 msiexec.exe @ *2017-04-20 16:45:14 UTC*
  - **PR** #40716: (alexproca) win_pkg: add msiexec override to enable selection of 32 or 64 msiexec.exe (refs: #40785)
  - 5388ffa7a2 Merge pull request #40785 from alexproca/backport-winexec-selection
  - 91cafd5094 Add option to select 32 or 64 version of msiexec
- **PR** #40796: (terminalmage) Fix inaccurate nodegroup docs @ *2017-04-20 16:08:22 UTC*
  - f0f135c71d Merge pull request #40796 from terminalmage/fix-nodegroup-docs

  - f99259a6eb Fix inaccurate nodegroup docs
- **ISSUE** #40737: (jf) Fix consul_pillar documentation: 'root=' canNOT start with a slash (refs: #40760)
- **PR** #40769: (rallytime) Back-port #40760 to 2016.11 @ *2017-04-19 20:23:22 UTC*
  - **PR** #40760: (jf) Fix 'root=/…' references in consul_pillar documentation: 'keys should not start with a forward slash'! (refs: #40769)
  - d8f78550d9 Merge pull request #40769 from rallytime/bp-40760
  - 71ac15fc4c Fix 'root=/…' references in consul_pillar documentation: 'keys should not start with a forward slash'!
- **PR** #40756: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-04-19 17:47:51 UTC*
  - 61f8de43df Merge pull request #40756 from rallytime/merge-2016.11
  - 0e087323f1 Merge branch '2016.3' into '2016.11'
  - f4f3ee69ba Merge pull request #40721 from gtmanfred/2016.3
    - 58b88859b3 unset the bitwise instead of toggle
- **ISSUE** #29602: (multani) cloud.action start raises "got an unexpected keyword argument 'kwargs'" (refs: #40735)
- **PR** #40735: (rallytime) Handle stacktraces in cloud.action function in module and runner @ *2017-04-18 20:05:06 UTC*
  - 3557b5140e Merge pull request #40735 from rallytime/handle-cloud-traces
  - 87154a95a4 Use *log.error* instead of *log.err*
  - b35bf919a3 Handle stacktraces in cloud.action function in module and runner
- **PR** #40745: (cro) Backport *Add support for specifying a datastore for new disks.* PR #36457 @ *2017-04-18 20:00:51 UTC*
  - **PR** #36457: (cro) Add support for specifying a datastore for new disks. (refs: #40745)
  - e700d8183b Merge pull request #40745 from cro/vmware_disk_datastore_bp
  - 1460f82ce4 Remove leftover conflict markers (oops! :-/)
  - b26be652dd Remove leftover conflict markers (oops! :-/)
  - 096f063464 Remove leftover conflict markers (oops! :-/)
  - d24078d1a0 Add docs for "datastore" param for disks
  - 500d6b281d Document validity of a datastore key inside a disk definition.
  - 7608b10225 Add support for specifying a datastore for new disks.
- **PR** #40740: (cro) Backport pr #39802 to add random_startup_delay @ *2017-04-18 19:47:55 UTC*
  - **PR** #39802: (cachedout) A random startup delay option for minions (refs: #40740)
  - 78dbab01dc Merge pull request #40740 from cro/minion_delay_start
  - 2ab95b7dd5 Set minion test to use default opts
  - 785e6060a9 Add requested docs
  - 8ab321f934 A random startup delay option for minions
- **PR** #40728: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-04-17 21:13:15 UTC*
  - a48ecc4a5c Merge pull request #40728 from rallytime/merge-2016.11

  - 0a5e05a6e5 Merge branch '2016.3' into '2016.11'
  - bf8bb0fde6 Merge pull request #40719 from rallytime/bp-40714
    - d6c436246b Make salt.modules.pw_user.get_loginclass return string rather than dict
  - 4145d33e46 Merge pull request #40718 from terminalmage/fix-docstring
  - 14e8b85da5 Fix copypasta in the pw_user docstring
- **ISSUE** #36967: (gmykhailiuta) S3fs objects list gets truncated (refs: #40707)
- **PR** #40707: (gtmanfred) Use markers when s3 bucket list is truncated @ *2017-04-17 16:45:21 UTC*
  - c5cbfc2e63 Merge pull request #40707 from gtmanfred/2016.11
  - 1932f7265d Use markers when s3 bucket list is truncated

28.2.37 Salt 2016.11.6 Release Notes

Version 2016.11.6 is a bugfix release for *2016.11.0*.

Statistics

- Total Merges: **137**
- Total Issue References: **58**
- Total PR References: **153**
- Contributors: **49** (BenoitKnecht, Ch3LL, Enquier, F30, Foxlik, The-Loeki, UtahDave, abednarik, alex-zel, arif-ali, automate-solutions, axmetishe, bdrung, cachedout, cro, darenjacobs, dmurphy18, dschaller, epcim, gareth-greenaway, github-abcde, gtmanfred, isbm, jettero, jmarinaro, kiorky, lomeroe, lordcirth, lorengordon, lubyou, mcalmer, moio, onlyanegg, peter-funktionIT, pkazmierczak, pprkut, rallytime, ricohouse, seanjnkns, sebw, skizunov, svinota, t0fik, terminalmage, tmeneau, tonybaloney, twangboy, whiteinge, yannj-fr)

Changelog for v2016.11.5..v2016.11.6

Generated at: 2018-05-27 20:18:17 UTC

- **PR** #41861: (twangboy) Fix problems with get_rule and delete_rule @ *2017-06-20 20:37:23 UTC*
  - afc61ffe63 Merge pull request #41861 from twangboy/fix_win_firewall
  - 78892074f5 Fix problems with get_rule and delete_rule
- **ISSUE** #41778: (frogunder) 2016.11.6 - TCP Transport gives Exception (refs: #41787)
- **PR** #41787: (skizunov) Fix #41778 @ *2017-06-20 20:11:23 UTC*
  - **PR** #41436: (skizunov) TCP transport: Fix occasional errors when using salt command (refs: #41787)
  - 938d4fddf1 Merge pull request #41787 from skizunov/develop3
  - 2ffd20cede Fix #41778
- **PR** #41812: (skizunov) TCP: Fix salt-master in bad state if remote side closed connection @ *2017-06-20 19:46:53 UTC*
  - 03b6ae5ea8 Merge pull request #41812 from skizunov/develop4
  - 736420eb83 TCP: Fix salt-master in bad state if remote side closed connection

- **PR** #41857: (dmurphy18) Modified support for deprecated netstat being removed by utilizing ss @ *2017-06-20 18:46:27 UTC*
  - cf2252bcea Merge pull request #41857 from dmurphy18/netstat_fix
  - 017fbdbc53 Modified support for deprecated netstat being removed by utilizing ss
- **ISSUE** #40878: (joewreschnig) SSH modules spam warning about MD5 fingerprints when there aren't any (refs: #41837)
- **ISSUE** #40005: (vutny) *ssh_known_hosts.present* does not support SHA256 key fingerprints (refs: #40543)
- **PR** #41837: (rallytime) Add fingerpint_hash_type option to ssh_auth state and related functions @ *2017-06-20 18:14:53 UTC*
  - **PR** #40543: (rallytime) Add the "fingerprint_hash_type" option to ssh state and module (refs: #41837)
  - 12ec5f9f23 Merge pull request #41837 from rallytime/fix-40878
  - 48ff5d2a62 Add fingerpint_hash_type option to ssh_auth state and related functions
- **PR** #41839: (cro) Extend proxy to jinja @ *2017-06-19 23:03:00 UTC*
  - e7fc30f482 Merge pull request #41839 from cro/extend_proxy_to_jinja
  - 172d3520ea Merge branch 'extend_proxy_to_jinja' of github.com:cro/salt into extend_proxy_to_jinja
    - 2e4a0633da Extend __proxy__ to jinja as *proxy* (like __salt__->salt)
  - 2ffad2af35 Extend __proxy__ to jinja as *proxy* (like __salt__->salt)
- **ISSUE** #41733: (sumeetisp) Salt Rest Api call (refs: #41786)
- **ISSUE** #40845: (e-senthilkumar) /jobs call is broken in 2016.11.4 (refs: #41786)
- **ISSUE** #38962: (gstachowiak) Broken /jobs in salt-api in salt 2016.11.1 (Carbon) (refs: #39472)
- **PR** #41786: (whiteinge) Runner arg parsing regressions @ *2017-06-19 23:00:07 UTC*
  - **PR** #39472: (whiteinge) Update _reformat_low to not run kwarg dicts through parse_input (refs: #41786)
  - 58387b127a Merge pull request #41786 from whiteinge/runner-arg-parsing-regressions
  - bf15c0bb5f Restore sending __current_eauth_* through to the function
  - 6be975da2c Fix regressions from not calling load_args_and_kwargs
  - 9d1cc1a176 Add test to check that runners ignore invalid kwargs
- **PR** #41776: (gtmanfred) npm 5.0.0 added a second line after fsevents @ *2017-06-19 16:53:43 UTC*
  - be0e9abedb Merge pull request #41776 from gtmanfred/2016.11
  - 733a2279ca npm 5.0.0 added a second line after fsevents
- **ISSUE** #32400: (rallytime) Document Default Config Values (refs: #41783)
- **PR** #41783: (rallytime) Add a bunch of config options to the various master/minion files that are missing @ *2017-06-19 16:42:54 UTC*
  - d94d4e4d19 Merge pull request #41783 from rallytime/config-doc-updates
  - c828ad803a Add a bunch of config options to the various master/minion files that are missing
- **PR** #41816: (twangboy) Upgrade psutil to version 5.2.2 @ *2017-06-17 01:51:29 UTC*
  - 2c681887d3 Merge pull request #41816 from twangboy/update_psutil_req
  - 8b4e3ad77d Upgrade psutil to version 5.2.2

- **ISSUE** #41785: (UtahDave) Using master tops without a top.sls file causes extra errors in minion log (refs: #41803)
- **PR** #41803: (terminalmage) Don't log an error when no top.sls is found @ *2017-06-16 22:49:08 UTC*
  - 3e5fe7ca4b Merge pull request #41803 from terminalmage/issue41785
  - f9f4d49f05 Don't log an error when no top.sls is found
- **PR** #41801: (terminalmage) Don't take hostname from name param when creating docker container (2016.11 branch) @ *2017-06-16 17:02:02 UTC*
  - d12bc4ee68 Merge pull request #41801 from terminalmage/issue41781-2016.11
  - 8236d3e1c3 Don't take hostname from name param when creating docker container (2016.11 branch)
- **PR** #41768: (rallytime) Manually back-port the changes in PR #41615 @ *2017-06-15 20:41:45 UTC*
  - **PR** #41615: (Ch3LL) Fix get_hwclock_aix test on MacOSX (refs: #41768)
  - 87e2e72d94 Merge pull request #41768 from rallytime/bp-41615
  - b6cc0b6bf0 Manually backport the changes in PR #41615
- **PR** #41740: (terminalmage) Fix spurious error when glob/regex used in publisher_acl @ *2017-06-15 15:14:56 UTC*
  - 36cb223ab2 Merge pull request #41740 from terminalmage/zd1532
  - e5f3d08751 Fix spurious error when glob/regex used in publisher_acl
- **PR** #41749: (terminalmage) Fix bug in pkg_resource.parse_targets when version passed @ *2017-06-15 15:05:52 UTC*
  - 126a36747b Merge pull request #41749 from terminalmage/parse_targets
  - 698806fb09 No need to manually create pkg_params dict when name and version passed
  - 7484bcc6c6 parse_targets: include version in packed return data
- **PR** #41753: (rallytime) Back-port #41449 to 2016.11 @ *2017-06-14 22:16:10 UTC*
  - **PR** #41449: (sebw) Fix state "svn.latest" diff output in test mode (refs: #41753)
  - 2c24012ded Merge pull request #41753 from rallytime/bp-41449
  - fae41c2875 Adjusting SVN unit test
  - eac6b151eb Improved SVN output in test mode
- **PR** #41750: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-06-14 22:15:41 UTC*
  - e685858269 Merge pull request #41750 from rallytime/merge-2016.11
  - 89834e49c2 Merge branch '2016.3' into '2016.11'
    - c5a79a1ea6 Merge pull request #41695 from xiaoanyunfei/fixRecursion
      - c54fde69a0 fix spell error
      - bc855b4711 fix swallow exception
      - c0b1f57fc0 add exception
      - aacf9f0a44 fix maximum recursion depth exceeded
    - 4b43ba3366 Merge pull request #41585 from cro/sign_minion_messages
      - 628f709c3c Correct test–caching requires files on disk but the test just supplies what would have been read from disk.

        - 687872a488 Lint
        - dadf4b851c Add documentation to the example master and minion configuration files. Move minion event signing to a saner place. Enable dropping messages when signature does not verify or when minion is not adding the signature to its payloads.
        - e44673cdae Add caching of key.
        - c3917d1e91 Fat finger fix.
        - 3b9326fda7 Sign_minion_messages support
- **PR** #41756: (Ch3LL) Add Change Log to 2016.11.6 Release Notes @ *2017-06-14 20:57:08 UTC*
    - 36cc8f1e35 Merge pull request #41756 from Ch3LL/2016.11.6_release
    - fa368f21ac Add Change Log to 2016.11.6 Release Notes
- **ISSUE** #40155: (grichmond-salt) State module boto_vpc not working with boto 2 (refs: #41692)
- **PR** #41692: (rallytime) Add boto and boto3 version dependencies to boto_vpc state docs @ *2017-06-14 19:05:07 UTC*
    - edcafc6a26 Merge pull request #41692 from rallytime/fix-40155
    - 539c1b0692 Add boto and boto3 version dependencies to boto_vpc state docs
- **ISSUE** #39918: (kivoli) Enabling list merging leads to multiplying of unique list items (refs: #40902)
- **PR** #40902: (lorengordon) Removes duplicates when merging pillar lists and adds pillar.get override for pillar_merge_lists @ *2017-06-14 18:39:09 UTC*
    - bdaeb55a77 Merge pull request #40902 from lorengordon/pillar-get-merge-lists
    - 6e35673fe3 Preserves order when removing duplicates
    - 18eda7084c Updates list merge tests to check for sorted, unique lists
    - 74bf91f99e Sorts the list when removing duplicates
    - 26a4b1b17f Adds pillar.get param to control list merge/overwrite behavior
    - ed04bae94c Removes duplicate values when merging lists
- **ISSUE** #32743: (tonybaloney) Issue with salt-cloud on OpenSUSE (refs: #41723)
- **PR** #41723: (rallytime) Support apache-libcloud work-around for issue #32743 for versions older than 2.0.0 @ *2017-06-14 17:13:38 UTC*
    - **PR** #40837: (tonybaloney) Upgrade apache-libcloud package dependency for 2.0 (refs: #41723)
    - 203ec6730f Merge pull request #41723 from rallytime/libcloud-support
    - 1e9a06000b Bump version check down to 1.4.0 and use distutils.version lib
    - a30f654b04 Support apache-libcloud work-around for issue #32743 for versions older than 2.0.0
- **ISSUE** #41654: (Enquier) Nova Cloud module doesn't work for python-novaclient 8.0.0+ (refs: #41655)
- **PR** #41655: (Enquier) Allow Nova cloud module to set a specific floating ip address @ *2017-06-14 16:44:05 UTC*
    - 62dbf5083c Merge pull request #41655 from Enquier/nova-cloud-set_ip_address
    - 293bc64158 Removed empty debug log
    - 3d9871fe11 Cleaning up, removing debugging tests
    - c78e5feea9 Fixing error message
    - 404dffb6b8 Debugging variable format

  - 6fa3b976a5 removing string call
  - 005995e1b0 modifying variable calls
  - 9e5e7a38ec Testing variable changes
  - 05e240f37f Debugging Format of floating_ip variable
  - 366aca00a8 Adding Max version check for Nova since Cloud no longer operates at higher versions
  - 6f66c9d10c Fixing response of floating_ip_show to align with other floating ip's. Spelling fix
  - 58459adbe8 Adding ability to set a Floating IP by a specific IP address
- **PR** #41731: (terminalmage) Clarify that archive_format is required pre-2016.11.0 @ *2017-06-14 15:05:21 UTC*
  - 82eab84883 Merge pull request #41731 from terminalmage/docs
  - d3f4ea1a84 Clarify that archive_format is required pre-2016.11.0
- **PR** #41663: (skizunov) Don't invoke lspci if enable_lspci is False @ *2017-06-13 21:19:42 UTC*
  - b6d27beac2 Merge pull request #41663 from skizunov/develop3
  - 154d6ce59e Don't invoke lspci if enable_lspci is False
- **ISSUE** #40446: (sumeetisp) [Documentation] include list of kwargs for ec2.create_volume in cloud driver (refs: #41693)
- **PR** #41693: (rallytime) Document available kwargs for ec2.create_volume function @ *2017-06-13 19:51:10 UTC*
  - 46b8d5dc4b Merge pull request #41693 from rallytime/fix-40446
  - 569eb2bf7e Document available kwargs for ec2.create_volume function
- **ISSUE** #41691: (jdonofrio728) Can't pass integers as cmd.run environment variables (refs: #41696)
- **PR** #41696: (terminalmage) Handle a few edge/corner cases with non-string input to cmd.run @ *2017-06-13 18:48:56 UTC*
  - aab55d304a Merge pull request #41696 from terminalmage/issue41691
  - 0623e40d33 Apparently some funcs are passing tuples to cmd.run_*
  - cdbfb94cfe Handle a few edge/corner cases with non-string input to cmd.run
- **PR** #41697: (terminalmage) Resubmit #41545 against 2016.11 branch @ *2017-06-13 16:10:37 UTC*
  - **PR** #41545: (kiorky) Make print_cli resilient on slow systems (refs: #41697)
  - 97897d7a7a Merge pull request #41697 from terminalmage/pr-41545
  - faaacf88bf Use error name instead of error number
  - 7eacda5cbf Make print_cli resilient on slow systems
- **ISSUE** #40605: (sumeetisp) Salt-run manage.bootstrap (refs: #41711)
- **PR** #41711: (rallytime) Update deprecated version info in manage.bootstrap func for root_user @ *2017-06-13 16:04:32 UTC*
  - 09260d7c08 Merge pull request #41711 from rallytime/fix-40605
  - 903c2ffca5 Update deprecated version info in manage.bootstrap fucn for root_user
- **ISSUE** #39668: (mirceaulinic) Master scheduled job not recorded on the event bus (refs: #41658)
- **PR** #41658: (garethgreenaway) Fixes to the salt scheduler @ *2017-06-13 16:00:57 UTC*
  - d563b3e345 Merge pull request #41658 from garethgreenaway/39668_schedule_runners_fire_events

  - d688a1cd88 Enable jobs scheduled on the master to fire their return data to the event bus
- **PR** #41706: (twangboy) Add missing batch files @ *2017-06-13 15:32:53 UTC*
  - 3c3b9343b7 Merge pull request #41706 from twangboy/batch_files
  - 0d4be0220b Add batch files for master
- **PR** #41710: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-06-13 15:11:38 UTC*
  - 1afc4adc5a Merge pull request #41710 from rallytime/merge-2016.11
  - 5150916556 Merge branch '2016.3' into '2016.11'
  - 5058b0de1f Merge pull request #41707 from terminalmage/master-tops-docs
    - 6ec9dfb7f3 Update version in master-tops docs
  - 1c1964d807 Merge pull request #41689 from yannj-fr/fix-41688
    - a47eddccd2 Fix #41688 : fix mkfs command linux-swap support
- **PR** #41702: (gtmanfred) npm 5 and greater requires –force for cache clean @ *2017-06-12 23:21:56 UTC*
  - 5d763b9b7f Merge pull request #41702 from gtmanfred/2016.11
  - 8bd19fcc17 fix version number
  - 0fa380f75c npm 5 and greater requires –force for cache clean
- **ISSUE** #41668: (yannj-fr) Parted modules mkfs command does not work with NTFS (refs: #41670)
- **PR** #41704: (rallytime) Back-port #41670 to 2016.11 @ *2017-06-12 23:20:31 UTC*
  - **PR** #41670: (yannj-fr) fixes #41668 ntfs case problem in parted module (refs: #41704)
  - f6519e7f80 Merge pull request #41704 from rallytime/bp-41670
  - 8afc8792d1 fixes #41668 ntfs case problem in parted module
- **ISSUE** #39939: (martinschipper) Relative symlinks are changed with file.recurse 2016.11.3 (refs: #41700)
- **PR** #41700: (terminalmage) roots: return actual link destination when listing symlinks @ *2017-06-12 22:07:03 UTC*
  - 0b89377dce Merge pull request #41700 from terminalmage/issue39939
  - bdbb265a0b roots: return actual link destination when listing symlinks
- **PR** #41699: (rallytime) Remove note about version incompatibility with salt-cloud @ *2017-06-12 19:44:28 UTC*
  - 7cf47f9651 Merge pull request #41699 from rallytime/troubleshooting-doc-update
  - c91ca5f809 Remove note about version incompatibility with salt-cloud
- **ISSUE** #40410: (DarrenDai) Targeting Minions by IP Range via restful API doesn't work (refs: #41694)
- **PR** #41694: (rallytime) Add ipcidr options to "Allowed Values" list in LocalClient expr_form docs @ *2017-06-12 19:06:16 UTC*
  - d68a6316b8 Merge pull request #41694 from rallytime/fix-40410
  - 6de9da1d5d Add ipcidr options to "Allowed Values" list in LocalClient expr_form docs
- **ISSUE** #41365: (lubyou) file.managed chokes on windows paths when source_hash is set to the URI of a file that contains source hash strings (refs: #41659)
- **PR** #41659: (lubyou) Use re.escape to escape paths before handing them to re.match @ *2017-06-12 18:10:53 UTC*
  - 80d4a3ab98 Merge pull request #41659 from lubyou/41365-fix-file-managed

  - d49a1579b0 Use re.escape to escape paths, before handing them to re.match
  - ac240facca use correct variable
  - c777eba2c1 Use re.escape to escape paths, before handing them to re.match
- **PR** #41661: (whiteinge) Add note about avoiding the *-i* flag for the /keys endpoint @ *2017-06-09 15:03:40 UTC*
  - 564d5fd9d3 Merge pull request #41661 from whiteinge/rest_cherrypy-keys-headers
  - a66ffc9d3e Add note about avoiding the *-i* flag for the /keys endpoint
- **ISSUE** #41651: (Sakorah) pkg.installed fails when unholding and test=true (refs: #41660)
- **PR** #41660: (garethgreenaway) Fix to modules/aptpkg.py for unheld @ *2017-06-09 14:53:23 UTC*
  - 38424f3e3e Merge pull request #41660 from garethgreenaway/41651_fixing_aptpkg_held_unheld_with_test
  - 30da2370a4 Fix when test=True and packages were being set to unheld.
- **PR** #41656: (rallytime) Back-port #41575 to 2016.11 @ *2017-06-08 22:43:23 UTC*
  - **PR** #41575: (dschaller) Fix 41562 (refs: #41656)
  - a308b960d8 Merge pull request #41656 from rallytime/bp-41575
  - 4374e6b034 Replace "tbd" with release version information
  - 81413896d1 Lint: Add index numbers to format {} calls
  - 384570384e only list top level npm modules during {un)install
- **PR** #41456: (bdrung) Fix pkgrepo.managed always return changes for test=true @ *2017-06-08 18:21:05 UTC*
  - e6d37b5f3e Merge pull request #41456 from bdrung/fix-pkgrepo.managed-changes-check
  - d3ce7bf05f Fix pkgrepo.managed always return changes for test=true
  - 1592687294 Document aptpkg architectures parameter
- **ISSUE** #41478: (jf) security / information leak with consul pillar when subsitution values are not present (refs: #41530)
- **PR** #41530: (gtmanfred) Set default for consul_pillar to None @ *2017-06-08 18:13:15 UTC*
  - 721e5b6cb9 Merge pull request #41530 from gtmanfred/2016.11
  - 2a4633ce16 Set default for consul_pillar to None
- **ISSUE** #41629: (lubyou) salt.states.cmd.script: Parameter "args" is overwritten if "name/id" contains spaces (refs: #41638)
- **PR** #41638: (gtmanfred) don't overwrite args if they are passed to the script @ *2017-06-08 17:48:48 UTC*
  - 8926d1c731 Merge pull request #41638 from gtmanfred/cmdscript
  - 6c7d68b97d don't overwrite args if they are passed to the script
- **PR** #41639: (dmurphy18) Update notrim check, netstat takes minutes if large number connections @ *2017-06-07 23:03:24 UTC*
  - ecb09b8694 Merge pull request #41639 from dmurphy18/minion_netstat_check
  - 7ab3319090 Update notrim check, netstat takes minutes if large number connections - 260K
- **ISSUE** #38894: (amendlik) salt.runner and salt.wheel ignore test=True (refs: #41309, #41611)
- **PR** #41611: (garethgreenaway) Additional fixes to states/saltmod.py @ *2017-06-07 22:58:24 UTC*
  - 2913a33b27 Merge pull request #41611 from garethgreenaway/41309_right_return_res

  - fda41ede76 Updating result values to be None for test cases.
  - 003f2d9323 Following the documentation, when passed the test=True argument the runner and wheel functions should return a result value of False.
- **ISSUE** #41626: (ruiaylin) When onlyif and bg are used together the (refs: #41637)
- **PR** #41637: (gtmanfred) never run bg for onlyif or unless cmd states @ *2017-06-07 17:37:47 UTC*
  - 334a5fc2a0 Merge pull request #41637 from gtmanfred/cmd
  - 40fb6c6249 never run bg for onlyif or unless cmd states
- **PR** #41255: (lordcirth) linux_syctl.default_config(): only return path, don't create it @ *2017-06-07 14:13:07 UTC*
  - 34dd9ea862 Merge pull request #41255 from lordcirth/fix-sysctl-test-11
  - 0089be4440 linux_sysctl: use dirname() as suggested
  - 262d95e41d linux_syctl.default_config(): only return path, don't create it
  - 277232b3ac linux_sysctl.persist(): create config dir if needed
- **ISSUE** #35481: (giany) global_identifier does not work when using Softlayer driver (refs: #41551)
- **PR** #41616: (rallytime) Back-port #41551 to 2016.11 @ *2017-06-06 22:44:09 UTC*
  - **PR** #41551: (darenjacobs) Update __init__.py (refs: #41616)
  - 4cf577771b Merge pull request #41616 from rallytime/bp-41551
  - 53bca96328 Update __init__.py
- **PR** #41552: (Enquier) Adding logic so that update_floatingip can dissassociate floatingip's @ *2017-06-06 18:25:56 UTC*
  - 846ca54688 Merge pull request #41552 from Enquier/neutron-floatingip-remove
  - aeed51c1e3 Adding port=None default and documentation
  - fcce05e1e4 Adding logic so that update_floatingip can dissassociate floatingip's Previously update_floatingip would cause an error if port is set to None.
- **PR** #41569: (gtmanfred) Check all entries in result @ *2017-06-06 18:18:17 UTC*
  - b720ecb732 Merge pull request #41569 from gtmanfred/fix_test_result_check
  - 19ea5481b6 remove test that never passed
  - e2a4d5e1e2 Check all entries in result
- **ISSUE** #41540: (UtahDave) archive.extracted fails on second run (refs: #41599)
- **PR** #41599: (garethgreenaway) Fixes to modules/archive.py @ *2017-06-06 18:02:14 UTC*
  - d9546c6283 Merge pull request #41599 from garethgreenaway/41540_fixes_to_archive_module
  - 66a136e6d8 Fixing issues raised in #41540 when a zip file is created on a Windows system. The issue has two parts, first directories that end up in the archive end up in the results of aarchive.list twice as they show up as both files and directories because of the logic to handle the fact that Windows doesn't mark them as directories. This issue shows up when an extraction is run a second time since the module verified the file types and the subdirectory is not a file. The second issue is related to permissions, if Salt is told to extract permissions (which is the default) then the directory and files end up being unreadable since the permissions are not available. This change sets the permissions to what the default umask for the user running Salt is.
- **ISSUE** #40950: (idokaplan) Import certificate (refs: #41453, #41383)

- **PR** #41453: (peter-funktionIT) Update win_pki.py @ *2017-06-06 17:15:55 UTC*
    - **PR** #41383: (peter-funktionIT) Update win_pki.py (refs: #41453)
    - 10ac80ee96 Merge pull request #41453 from peter-funktionIT/fix_win_pki_state_import_cert
    - d146fd029c Update win_pki.py
    - ef8e3ef569 Update win_pki.py
- **PR** #41557: (dmurphy18) Add symbolic link for salt-proxy service similar to other serivce files @ *2017-06-06 17:13:52 UTC*
    - 3335fcbc7d Merge pull request #41557 from dmurphy18/fix-proxy-service
    - ffe492d6a9 Add symbolic link salt-proxy service similar to other service files
- **PR** #41597: (rallytime) Back-port #41533 to 2016.11 @ *2017-06-06 15:15:09 UTC*
    - **PR** #41533: (svinota) unit tests: add pyroute2 interface dict test (refs: #41597)
    - 65ed230f45 Merge pull request #41597 from rallytime/bp-41533
    - 535b8e8d8e Update new pyroute2 unit test to conform with 2016.11 branch standards
    - 5c86dee73c unit tests: test_pyroute2 – add skipIf
    - 026b39493f unit tests: add encoding clause into test_pyroute2
    - 9ab203d54b unit tests: fix absolute imports in test_pyroute2
    - 1f507cfa7a unit tests: add pyroute2 interface dict test
- **PR** #41596: (rallytime) Back-port #41487 to 2016.11 @ *2017-06-06 02:44:17 UTC*
    - **PR** #41487: (svinota) clean up *change* attribute from interface dict (refs: #41596)
    - bf8aed153d Merge pull request #41596 from rallytime/bp-41487
    - 7b497d9ec6 clean up *change* attribute from interface dict
- **ISSUE** #41435: (seanjnkns) 2016.11: Keystone.endpoint_present overwrites all interfaces (refs: #41509)
- **PR** #41509: (seanjnkns) Add keystone V3 API support for keystone.endpoint_present|absent @ *2017-06-03 03:01:05 UTC*
    - cc6c98a8d8 Merge pull request #41509 from seanjnkns/fix-keystone-v3-endpoint_present
    - 095e5949a3 Fix unit tests for PR #41509
    - eb7ef3c856 Add keystone V3 API support for keystone.endpoint_present|get, endpoint_absent|delete.
- **ISSUE** #38061: (Ch3LL) x509.crl_managed ValueError when digest is not specified in the module (refs: #41539)
- **PR** #41539: (gtmanfred) allow digest to be empty in create_crl @ *2017-06-02 17:00:04 UTC*
    - 0a08649637 Merge pull request #41539 from gtmanfred/x509
    - 0989be8919 allow digest to be empty in create_crl
- **ISSUE** #41154: (mephi42) archive.extracted outputs password embedded in archive URL (refs: #41561)
- **PR** #41561: (terminalmage) Redact HTTP basic authentication in archive.extracted @ *2017-06-02 15:33:14 UTC*
    - 3ae8336895 Merge pull request #41561 from terminalmage/issue41154
    - cbf8acbafc Redact HTTP basic authentication in archive.extracted
- **PR** #41436: (skizunov) TCP transport: Fix occasional errors when using salt command (refs: #41787) @ *2017-06-01 16:37:43 UTC*

  - 39840bfe4e Merge pull request #41436 from skizunov/develop2
  - 07d5862773 unit.transport.tcp_test: Clean up channel after use
  - 4b6aec7154 Preserve original IO Loop on cleanup
  - 892c6d4d24 TCP transport: Fix occasional errors when using salt command
- **ISSUE** #41335: (sypherndl) [2016.11.5] ssh_auth.present: IndexError: list index out of range (refs: #41337)
- **PR** #41337: (Foxlik) Fix #41335 - list index out of range on empty line in authorized_keys @ *2017-05-31 19:59:17 UTC*
  - 06ed4f077b Merge pull request #41337 from Foxlik/2016.11
  - 916fecb64f modify ssh_test.py, to check empty lines and comments in authorized_keys #41335
  - 011d6d65e7 Fix #41335 - list index out of range on empty line in authorized_keys
- **PR** #41512: (twangboy) Use psutil where possible in win_status.py @ *2017-05-31 19:56:00 UTC*
  - 1ace72d871 Merge pull request #41512 from twangboy/fix_win_status
  - 582d09b484 Get psutil import
  - fd88bb277f Remove unused imports (lint)
  - 41a39dff00 Use psutil where possible
- **PR** #41490: (t0fik) Backport of SELinux module installation and removal @ *2017-05-31 19:38:00 UTC*
  - 683cc5f414 Merge pull request #41490 from jdsieci/2016.11_selinux
  - e2fbada1c1 Backport of SELinux module installation and removal
- **PR** #41522: (jettero) Sadly, you can't have '.'s and '$'s in dict keys in a mongodb doc. @ *2017-05-31 15:55:24 UTC*
  - 2e7e84b8f2 Merge pull request #41522 from jettero/mongodb-keys-are-stupid
  - 12648f5439 dang, thought I already got that. Apparently only got the bottom one. This should do it.
  - 7c4a763518 ugh, forgot about this lint too. This one looks especially terrible.
  - c973988d8d forgot about the linter pass … fixed
  - da0d9e4045 Sadly, you can't have '.'s and '$'s in dict keys in a mongodb doc.
- **ISSUE** #41504: (mtkennerly) Can't set REG_DWORD registry value larger than 0x7FFFFFFF (refs: #41506)
- **PR** #41506: (gtmanfred) check for integer types @ *2017-05-31 00:48:21 UTC*
  - 30ad4fd9a0 Merge pull request #41506 from gtmanfred/2016.11
  - 5fe2e9bbf5 check for integer types
- **PR** #41469: (Ch3LL) Fix keep_jobs keyerror in redis returner @ *2017-05-30 18:37:42 UTC*
  - 06ef17dec3 Merge pull request #41469 from Ch3LL/fix_redis_error
  - 8ee1251a3a Fix keep_jobs keyerror in redis returner
- **PR** #41473: (twangboy) Fix win_firewall execution and state modules @ *2017-05-30 18:35:24 UTC*
  - 7a09b2b678 Merge pull request #41473 from twangboy/fix_win_firewall
  - e503b455c3 Fix lint error
  - d3f0f8bcd2 Fix win_firewall execution and state modules
- **PR** #41499: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-05-30 18:06:03 UTC*

  - f635cb11c4 Merge pull request #41499 from rallytime/merge-2016.11
  - 20d893d397 Merge branch '2016.3' into '2016.11'
  - 964b1ee027 Merge pull request #41439 from terminalmage/salt-cp-base64
    - ebf6cc78c7 base64 encode binary data sent using salt-cp
- **PR** #41464: (rallytime) Back-port #39850 to 2016.11 @ *2017-05-26 21:22:44 UTC*
  - **PR** #39850: (epcim) Fix endpoint handling per region (refs: #41464)
  - 83f1e48241 Merge pull request #41464 from rallytime/bp-39850
  - 9b84b751b2 Pylint fixes
  - 6db8915021 Endpoint handling per region, fixes #35874 - extend tests for multiple regions - region arg by default set to None - print verbose changes to be exec.
- **PR** #41443: (UtahDave) use proper arg number @ *2017-05-26 20:36:37 UTC*
  - 960c5767fa Merge pull request #41443 from UtahDave/fix_args_masterpy
  - dfbdc275ca use proper arg number
- **ISSUE** #41341: (lorengordon) TypeError traceback in network.system with retain_settings=True (refs: #41350)
- **PR** #41350: (lorengordon) Supports quoted values in /etc/sysconfig/network @ *2017-05-26 16:22:03 UTC*
  - 88c28c18c3 Merge pull request #41350 from lorengordon/issue-41341
  - f2f6da7039 Supports quoted values in /etc/sysconfig/network
- **PR** #41398: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-05-26 15:17:49 UTC*
  - 824f2d3b69 Merge pull request #41398 from rallytime/merge-2016.11
  - 2941e9c923 Merge pull request #22 from terminalmage/merge-2016.11
    - 087a958afc base64 encode binary data sent using salt-cp
  - 503f925275 Add missing import
  - d2d9a3d29f Merge branch '2016.3' into '2016.11'
    - d617c9fe72 Merge pull request #41265 from terminalmage/issue41234
      - edf552fe9a Update PKG_TARGETS for RHEL-based distros
      - 0ecc7b9b20 yumpkg: fix latest_version() when showdupesfromrepos=1 set in /etc/yum.conf
    - 26bd914580 Merge pull request #41316 from Ch3LL/update_latest_2016.3
      - 520740d862 [2016.13] Bump latest release version to 2016.11.5
    - 18898b7d1f Merge pull request #41216 from terminalmage/issue16592
      - 0e15fdbb1a Update salt-cp integration test to reflect recent changes
      - 10dc695cc4 Make salt-cp work with larger files
      - c078180539 Make KeyErrors more specific when interpreting returns
      - fc401c9eb4 Add generator functions for reading files
- **PR** #41442: (UtahDave) use proper arg number @ *2017-05-26 13:42:50 UTC*
  - ec08064b99 Merge pull request #41442 from UtahDave/fix_args
  - 0324833c9e use proper arg number

- **ISSUE** #37824: (dxiri) SSLError Trying to use v3 API of Openstack Newton as provider. (refs: #41397, #40752)
- **ISSUE** #36548: (abonillasuse) openstack auth with nova driver (refs: #38647)
- **PR** #41397: (Enquier) Updating Nova/Neutron modules to support KeystoneAuth and SSLVerify @ *2017-05-25 21:16:14 UTC*
  - **PR** #40752: (Enquier) Add ability to specify a custom SSL certificate or disable SSL verification in KeystoneAuth v3 (refs: #41397)
  - **PR** #38647: (gtmanfred) Allow novaclient to use keystoneauth1 sessions for authentication (refs: #41397)
  - 22096d9213 Merge pull request #41397 from Enquier/neutron-ssl-verify
  - d25dcf61d5 Small error in nova that was preventing execution
  - 0e7a1009ed Updated module docs to include changes made
  - 05e0192665 Adding missing os_auth_system
  - 4e0f4981e4 allow service_type to be specified default is now 'network'
  - 991e84343f Added non-profile and defaults for Neutron
  - c93f112c9b Updating Nova Module to include use_keystone Auth
  - 66ab1e5184 Re-adding neutron dependency check
  - cce07eefc2 Updating Neutron module to suport KeystoneAuth
- **ISSUE** #34460: (Ch3LL) Receive an error when using salt-api to call a runner (refs: #41409)
- **PR** #41409: (garethgreenaway) Fixes to ipc transport @ *2017-05-25 21:06:27 UTC*
  - 14a58cf536 Merge pull request #41409 from garethgreenaway/34460_fixes_ipc_transport
  - 5613b72dfe Updating the exception variable to be more in line with the rest of the exception code
  - 41eee8b333 Fixing a potential lint issue
  - 760d561dfa Fixing a potential lint issue
  - c11bcd0d12 Changing the approaching and including an except for the action socket.error exception, then logging a trace log if error number is 0 and an error log otherwise.
  - 3f950596f4 Fixing lint issues.
  - f3a6531a69 On occasion an exception will occur which results in the event not returning properly, even though the wire_bytes is correctly populated. In this situation, we log to trace and continue. #34460
- **PR** #41421: (UtahDave) Correct doc to actually blacklist a module @ *2017-05-25 21:01:46 UTC*
  - 824428700d Merge pull request #41421 from UtahDave/fix_blacklist_docs
  - 5eb27571a0 Correct doc to actually blacklist a module
- **ISSUE** #41353: (rmarchei) Orchestrate runner needs saltenv on 2016.11.5 (refs: #41431)
- **PR** #41431: (terminalmage) Fix regression in state orchestration @ *2017-05-25 18:44:53 UTC*
  - b98d5e00d4 Merge pull request #41431 from terminalmage/issue41353
  - 16eae64cca Fix regression in state orchestration
- **ISSUE** #41338: (ricohouse) Exception not raised when running config compare and the device (Juniper) returns error (refs: #41429)
- **PR** #41429: (ricohouse) Issue #41338: Return false when compare config fails @ *2017-05-25 17:18:02 UTC*
  - eeff3dd7fb Merge pull request #41429 from ricohouse/fix-compare-bug

  - 9b61665c4c Issue #41338: Return false when compare config fails
- **PR** #41414: (Ch3LL) Update bootstrap script verstion to latest release(v2017.05.24) @ *2017-05-24 19:51:49 UTC*
  - 561a416cf3 Merge pull request #41414 from Ch3LL/update_bootstrap
  - d8c03eef60 Update bootstrap script verstion to latest release(v2017.05.24)
- **PR** #41336: (mcalmer) fix setting and getting locale on SUSE systems @ *2017-05-24 17:46:08 UTC*
  - 88fd3c0ed9 Merge pull request #41336 from mcalmer/fix-locale-on-SUSE
  - f30f5c8a25 fix unit tests
  - 428baa9bce fix setting and getting locale on SUSE systems
- **PR** #41393: (rallytime) Back-port #41235 to 2016.11 @ *2017-05-24 16:08:56 UTC*
  - **PR** #41235: (moio) rest_cherrypy: remove sleep call (refs: #41393)
  - 4265959647 Merge pull request #41393 from rallytime/bp-41235
  - c79c0e3f43 rest_cherrypy: remove sleep call
- **PR** #41394: (rallytime) Back-port #41243 to 2016.11 @ *2017-05-24 16:00:17 UTC*
  - **PR** #41243: (arif-ali) Remove the keys that don't exist in the new change (refs: #41394)
  - 83f54694f9 Merge pull request #41394 from rallytime/bp-41243
  - a5351302af Lint fix
  - 05fadc0af3 Remove the keys that don't exist in the new change
- **PR** #41401: (bdrung) Add documentation key to systemd service files @ *2017-05-24 15:49:54 UTC*
  - 3a45ac30f0 Merge pull request #41401 from bdrung/systemd-service-documentation-key
  - 3f7f30895d Add documentation key to systemd service files
- **PR** #41404: (bdrung) Fix typos @ *2017-05-24 14:42:44 UTC*
  - d34333c30b Merge pull request #41404 from bdrung/fix-typos
  - 33a7f8b2ec Fix typos
- **PR** #41388: (bdrung) Do not require sphinx-build for cleaning docs @ *2017-05-23 19:32:41 UTC*
  - 3083764195 Merge pull request #41388 from bdrung/clean-doc-without-sphinx
  - 5b79a0a9f8 Do not require sphinx-build for cleaning docs
- **ISSUE** #41362: (automate-solutions) On AWS EC2: salt-cloud -f delete_keypair ec2 keyname=mykeypair doesn't delete the keypair (refs: #41364)
- **PR** #41364: (automate-solutions) Fix issue #41362 invalid parameter used: KeyName.1 instead of KeyName @ *2017-05-23 17:32:10 UTC*
  - 842875e590 Merge pull request #41364 from automate-solutions/fix-issue-41362
  - cfd8eb7a87 Set DescribeKeyPairs back to KeyName.1 according to documentation
  - 6a82ddc6fc Fix issue #41362 invalid parameter used: KeyName.1 instead of KeyName
- **ISSUE** #40950: (idokaplan) Import certificate (refs: #41453, #41383)
- **PR** #41383: (peter-funktionIT) Update win_pki.py (refs: #41453) @ *2017-05-23 17:26:43 UTC*
  - 92f94e66bc Merge pull request #41383 from peter-funktionIT/fix-win_pki-get_cert_file
  - 4d9bd06176 Update win_pki.py

- **PR** #41113: (cro) Rescue proxy_auto_tests PR from git rebase hell @ *2017-05-22 17:05:07 UTC*
  - **PR** #39575: (cro) WIP: Proxy auto test, feedback appreciated (refs: #41113)
  - 1ba95684a9 Merge pull request #41113 from cro/proxy_auto_test2
  - 19db038b99 Fix test–use proxy_config instead of minion_config
  - 7749ceadb6 Change default proxy minion opts so only the proxy-specific ones are listed, and the rest are taken from DEFAULT_MINION_OPTS.
  - 106394c80c Lint.
  - 3be90cc9f4 Rescue proxy_auto_tests PR from git rebase hell
- **PR** #41360: (cro) Sysrc on FreeBSD, YAML overeager to coerce to bool and int @ *2017-05-22 15:54:31 UTC*
  - 375892d910 Merge pull request #41360 from cro/sysrc_fix
  - 6db31ce52a Fix problem with sysrc on FreeBSD, YAML overeager to coerce to bool and int.
- **ISSUE** #41190: (jheidbrink) Cannot extract tar.xz archive when it exceeds size of /tmp (refs: #41372)
- **PR** #41372: (terminalmage) Don't use intermediate file when listing contents of tar.xz file @ *2017-05-22 15:36:45 UTC*
  - 01b71c75c1 Merge pull request #41372 from terminalmage/issue41190
  - 1f08936d9c Remove unused import
  - 68cb897520 Replace reference to fileobj
  - 788874408a Remove '*' from mode
  - 3d4b833627 Don't use intermediate file when listing contents of tar.xz file
- **PR** #41373: (alex-zel) Allow HTTP authentication to ES. @ *2017-05-22 15:32:09 UTC*
  - 5edfcf972c Merge pull request #41373 from alex-zel/patch-3
  - 3192eab128 Allow HTTP authentication to ES.
- **ISSUE** #40748: (djhaskin987) Consul backend minion cache does not work (refs: #41287)
- **PR** #41287: (garethgreenaway) Fix to consul cache @ *2017-05-19 18:32:56 UTC*
  - 29bd7f48b7 Merge pull request #41287 from garethgreenaway/40748_2016_11_consul
  - 5039fe12fb Removing chdir as it is no needed with this change
  - 4550c3ce49 Updating the code that is pulling in the list of cached minions to use self.cache.list instead of relying on checking the local file system, which only works for the localfs cache method. #40748
- **ISSUE** #38894: (amendlik) salt.runner and salt.wheel ignore test=True (refs: #41309, #41611)
- **PR** #41309: (garethgreenaway) Adding test argument for runners & wheel orchestration modules @ *2017-05-19 18:26:09 UTC*
  - 672aaa88d3 Merge pull request #41309 from garethgreenaway/38894_allowing_test_argument
  - e1a88e8bf7 Allowing test=True to be passed for salt.runner and salt.wheel when used with orchestration
- **ISSUE** #41306: (lomeroe) win_lgpo does not properly pack group policy version number in gpt.ini (refs: #41319, #41307)
- **PR** #41319: (lomeroe) backport #41307 to 2016.11, properly pack version numbers into single @ *2017-05-19 18:25:00 UTC*
  - **PR** #41307: (lomeroe) properly pack/unpack the verison numbers into a number (refs: #41319)

  - 140b0427e1 Merge pull request #41319 from lomeroe/bp_41307
  - 4f0aa577a5 backport 41307 to 2016.11, properly pack version numbers into single number
- **PR** #41327: (Ch3LL) Add 2016.11.6 Release Notes @ *2017-05-19 18:05:09 UTC*
  - 6bdb7cca7d Merge pull request #41327 from Ch3LL/add_2016.11.6_release
  - e5fc0aeb9c Add 2016.11.6 Release Notes
- **PR** #41329: (lorengordon) Corrects versionadded for win_network.get_route @ *2017-05-19 17:47:57 UTC*
  - 1faffd3932 Merge pull request #41329 from lorengordon/doc-fix
  - 3c471247f0 Corrects versionadded for win_network.get_route
- **PR** #41322: (Ch3LL) Add patched packages warning to 2016.11.5 release notes @ *2017-05-18 21:53:26 UTC*
  - 6ca65592da Merge pull request #41322 from Ch3LL/fix_release_2016.11.5_notes
  - 9a1bf4205f fix url refs in rst
  - cde008ff77 Add patched packages warning to 2016.11.5 release notes
- **PR** #41208: (pkazmierczak) Fix: zypper handling of multiple version packages @ *2017-05-18 15:44:26 UTC*
  - 9f359d841f Merge pull request #41208 from pkazmierczak/pkazmierczak-zypper-multiple-ver-pkgs
  - d411a91676 Reverted back to cascading with statements for python 2.6 compat
  - 7204013653 Compacted with statements in the unit test.
  - 6c4c08042c Added unit tests and copied the behavior to .upgrade method, too.
  - 5f952007f6 Fix: zypper handling of multiple version packages
- **PR** #41317: (Ch3LL) [2016.11] Bump latest release version to 2016.11.5 @ *2017-05-18 15:34:13 UTC*
  - bcef99adb6 Merge pull request #41317 from Ch3LL/update_latest_2016.11
  - cdb072c207 [2016.11] Bump latest release version to 2016.11.5
- **PR** #41232: (axmetishe) Add basic auth for SPM @ *2017-05-17 19:08:56 UTC*
  - b8ddd7ee08 Merge pull request #41232 from axmetishe/2016.11
  - 76104f23b4 Add basic auth for SPM
- **PR** #41236: (BenoitKnecht) states: cron: show correct changes when using *special* @ *2017-05-17 18:51:58 UTC*
  - 7bdb66d969 Merge pull request #41236 from BenoitKnecht/2016.11
  - 33211d032e states: cron: show correct changes when using *special*
- **PR** #41269: (isbm) Bugfix: Unable to use "127" as hostname for the Minion ID @ *2017-05-17 18:31:15 UTC*
  - 1c1e092f56 Merge pull request #41269 from isbm/isbm-minion-id-127-name
  - 5168ef8959 Add unit test for hostname can be started from 127
  - 0d0354198b Harden to 127. IP part
  - d9c8324a6b Unit test for accepting hosts names as 127
  - 65b03c667b Bugfix: unable to use 127 as hostname
- **PR** #41289: (garethgreenaway) Fixing consul cache @ *2017-05-17 16:54:12 UTC*
  - d0fa31d4ca Merge pull request #41289 from garethgreenaway/2016_11_5_fix_consul_cache_ls
  - 780a28c9a0 Swapping the order in the func_alias so the ls function is available.

- **ISSUE** #41291: (lomeroe) win_lgpo does not properly convert large decimal values in regpol data (refs: #41301, #41303)
- **PR** #41303: (lomeroe) backport #41301 – properly convert packed string to decimal values @ *2017-05-17 16:32:22 UTC*
  - **PR** #41301: (lomeroe) properly convert packed string to decimal values (refs: #41303)
  - 6566648948 Merge pull request #41303 from lomeroe/bp-41301
  - f4b93f9d9a properly convert packed string to decimal values
- **ISSUE** #41231: (kaihowl) PR #30777 misses an update to the documentation for pkg.installed and hold:true (refs: #41251)
- **ISSUE** #30733: (ealphonse) version-controlled packages with hold: True can no longer be upgraded by salt (refs: #30777)
- **PR** #41283: (terminalmage) Backport #41251 to 2016.11 @ *2017-05-16 18:01:17 UTC*
  - **PR** #41251: (abednarik) Update apt module regarding upgrade against hold packages. (refs: #41283)
  - **PR** #30777: (abednarik) Fix update apt hold pkgs (refs: #41251)
  - 44598617be Merge pull request #41283 from terminalmage/bp-41251
  - ed03ca534f Update apt module regarding upgrade against hold packages.
- **PR** #41181: (gtmanfred) add resolving extra flags to yum upgrade @ *2017-05-16 04:07:47 UTC*
  - d8e9676fcf Merge pull request #41181 from gtmanfred/2016.11
  - 2ca71713b1 use six and clean_kwargs
  - c9bf09a5a1 add resolving extra flags to yum upgrade
- **ISSUE** #40177: (eldadru) libcloud_dns state "global name '__salt__' is not defined" in salt.cmd runner (refs: #40246)
- **PR** #41220: (rallytime) Back-port #40246 to 2016.11 @ *2017-05-15 17:59:38 UTC*
  - **PR** #40246: (tonybaloney) Fix libcloud_dns state module bug (refs: #41220)
  - 75942235f0 Merge pull request #41220 from rallytime/bp-40246
  - 79f1bb2bba Remove unused/duplicate imports leftover from merge-conflict resolution
  - 2f610680e5 remove unused imports
  - 9b7de2e7d7 fix unit tests
  - 49d94559ab linting
  - 4b260a4594 linting
  - 41d1adab5f fix up tests
  - b3822e03fc add fixes for incorrectly importing modules directly instead of using __salt__
- **ISSUE** #41230: (RealKelsar) 2016.11.5 IPv6 nameserver in resolv.conf leads to minion exception (refs: #41244)
- **ISSUE** #40912: (razed11) IPV6 Warning when ipv6 set to False (refs: #40934)
- **PR** #41244: (cachedout) Fix ipv6 nameserver grains @ *2017-05-15 17:55:39 UTC*
  - **PR** #40934: (gtmanfred) Only display IPvX warning if role is master (refs: #41244)
  - 53d5b3e816 Merge pull request #41244 from cachedout/fix_ipv6_nameserver_grains
  - f745db1a43 Lint

  - 6e1ab69710 Partial revert of #40934
  - 88f49f9146 Revert "Only display IPvX warning if role is master"
- **PR** #41242: (pprkut) Fix changing a mysql user to unix socket authentication. @ *2017-05-15 17:00:06 UTC*
  - 895fe582eb Merge pull request #41242 from M2Mobi/mysql_socket_auth
  - 7d8359766d Fix changing a mysql user to unix socket authentication.
- **ISSUE** #40940: (djhaskin987) When *state_aggregate* is set to *True*, the *latest* keyword doesn't work with pkg.installed (refs: #41101)
- **PR** #41101: (terminalmage) Fix "latest" keyword for version specification when used with aggregation @ *2017-05-15 16:52:35 UTC*
  - 50d8fde123 Merge pull request #41101 from terminalmage/issue40940
  - 7fe64219ae Add rtag check to integration test for pkg.refresh_db
  - 88a08aa3bf Add comments to explain what removing the rtag file actually does
  - 92011dbe5f Fix "latest" keyword for version specification when used with aggregation
- **ISSUE** #34775: (babilen) Please allow users to disable branch environment mapping in GitFS (refs: #41144)
- **PR** #41146: (terminalmage) gitfs: Backport performance fixes for getting tree objects @ *2017-05-12 17:35:47 UTC*
  - **PR** #41144: (terminalmage) gitfs: Add two new options to affect saltenv mapping (refs: #41146)
  - 049712ba53 Merge pull request #41146 from terminalmage/backport-get_tree-performance-improvement
  - f9d6734afe gitfs: Backport performance fixes for getting tree objects
- **ISSUE** #41135: (shallot) gpg renderer doesn't seem to work with salt-ssh, tries to execute gpg on the minion? (refs: #41161)
- **PR** #41161: (The-Loeki) gpg renderer: fix gpg_keydir always reverting to default @ *2017-05-12 17:19:07 UTC*
  - 4215a0b99d Merge pull request #41161 from The-Loeki/2016.11
  - 24946fef18 gpg renderer: fix gpg_keydir always reverting to default
- **ISSUE** #41162: (onlyanegg) Elasticsearch module functions should pass hosts and profile to index_exists() (refs: #41163)
- **PR** #41163: (onlyanegg) Elasticsearch - pass hosts and profile to index_exists() @ *2017-05-12 17:18:06 UTC*
  - 5b10fc58ba Merge pull request #41163 from onlyanegg/elasticsearch-pass_profile_to_index_exists
  - 7f512c701b Pass hosts and profile to index_exists() method
- **ISSUE** #41185: (jmarinaro) package name collisions in chocolatey state (refs: #41186)
- **PR** #41186: (jmarinaro) Fix package name collisions in chocolatey state @ *2017-05-12 17:01:31 UTC*
  - d433cf850d Merge pull request #41186 from jmarinaro/fix-chocolatey-package-collision
  - 229f3bf9f3 apply changes to uninstalled function
  - ffd4c7ef04 Fix package name collisions in chocolatey state
- **PR** #41189: (github-abcde) utils/minions.py: Fixed case where data is an empty dict resulting in... @ *2017-05-12 16:32:25 UTC*
  - bb5ef41ce0 Merge pull request #41189 from github-abcde/utils-minions-fix

  - 853dc5406c utils/minions.py: Fixed case where data is an empty dict resulting in errors.
- **PR** #41104: (Ch3LL) Add test to query results of /jobs call in api @ *2017-05-10 20:11:08 UTC*
  - b136b15330 Merge pull request #41104 from Ch3LL/add_jobs_test
  - dac16583b7 add test to query results of /jobs call in api
- **PR** #41170: (lomeroe) Backport #41081 to 2016.11 @ *2017-05-10 19:58:52 UTC*
  - **PR** #41081: (lomeroe) Update win_dns_client to use reg.read_value and set_value (refs: #41170)
  - ca18b4df93 Merge pull request #41170 from lomeroe/bp-41081
  - 2af89f2165 update mock data
  - b7fa115a59 update win_dns_client tests with correct module names
  - 4d05a22675 Update win_dns_client to use reg.read_value and set_value
- **PR** #41173: (twangboy) Add silent action to MsgBox for Path Actions @ *2017-05-10 19:57:06 UTC*
  - d7ec37b003 Merge pull request #41173 from twangboy/fix_installer
  - 24b11ffdc2 Add release notes
  - 96918dcfa6 Add silent action to MsgBox for Path Actions
- **PR** #41158: (Ch3LL) 2016.11.5 release notes: add additional commits @ *2017-05-09 22:41:40 UTC*
  - 88e93b7fe5 Merge pull request #41158 from Ch3LL/update_2016.11.5
  - 28371aa035 2016.11.5 release notes: add additional commits
- **PR** #41148: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-05-09 20:23:28 UTC*
  - d2ae7deff2 Merge pull request #41148 from rallytime/merge-2016.11
  - aba35e20dd Merge branch '2016.3' into '2016.11'
    - 2969153097 Merge pull request #41122 from terminalmage/masterless-env_cache-fix
      - be732f0577 gitfs: refresh env cache during update in masterless
    - b8f0a4f108 Merge pull request #41123 from terminalmage/gitfs-vsts-note
      - f6a16956a0 Add note on lack of support for VSTS in older libssh2 releases.
    - 8f79b6f537 Merge pull request #41090 from bbinet/rdurations_float
    - fd48a63653 rdurations should be floats so that they can be summed when profiling
- **PR** #41147: (rallytime) Back-port #39676 to 2016.11 @ *2017-05-09 18:40:44 UTC*
  - **PR** #39676: (F30) Fix comments about the "hash_type" option (refs: #41147)
  - 2156395b2e Merge pull request #41147 from rallytime/bp-39676
  - 5b55fb2452 Fix comments about the "hash_type" option
- **PR** #40852: (isbm) Isbm fix coregrains constants bsc#1032931 @ *2017-05-09 18:35:46 UTC*
  - a2f359fa13 Merge pull request #40852 from isbm/isbm-fix-coregrains-constants-bsc#1032931
  - f3b12a3f5b Do not use multiple variables in "with" statement as of lint issues
  - 35a8d99934 Disable the test for a while
  - 76cb1b7150 Rewrite test case for using no patch decorators
  - f71af0b625 Fix lint issues

  - 0e6abb3e37 Add UT on set_hw_clock on Gentoo
  - a2b1d4638c Add UT for set_hwclock on Debian
  - 5356a0821a Bugfix: use correct grain name for SUSE platform
  - 88e8184702 Add UT set_hwclock on SUSE
  - 0cd590f927 Fix UT names
  - bee94ade63 Add UT for set_hwclock on RedHat
  - dfe2610d05 Add UT for set_hwclock on Arch
  - d000a8a6f5 Add UT for set_hwclock on solaris
  - d2614aedaa Fix docstrings
  - 6d782191dc Add UT for set_hwclock on AIX
  - d303e0dd8a Add UT for AIX on get_hwclock
  - 86f2d83781 Add UT on Solaris
  - c3cafed6d5 Add UT for Debian on get_hwclock
  - d337c09357 Add UT for RedHat/SUSE platforms on get_hwclock
  - 501a59ca7e Bugfix: use correct grain for SUSE and RedHat platform
  - f25dc5c56c Add UT for get_hwclock on SUSE platform
  - 08e00c865c Remove dead code
  - 1216a0bf12 Add UT for get_hwclock on UTC/localtime
  - 39332c71d3 Remove duplicate code
  - 58676c568d Add UT for Debian on set_zone
  - 1b9ce37b1b Add UT for gentoo on set_zone
  - cf7f766a68 Bugfix: use correct os_family grain value for SUSE series
  - 6ed9be985e Adjust UT to use correct grain for SUSE series
  - ce4c836a60 Add UT for set_zone on SUSE series
  - 155a498b49 Doc fix
  - a40876cdac Remove unnecessary mock patch
  - ffab2db213 Fix doc for RH UT
  - 72388f7ae2 Add UT for RedHat's set_zone
  - 11595d3a42 Refactor with setup/teardown
  - ce6a06de98 Bugfix: use correct grain constant for platform
  - 28072c9e41 Adjust the test so it is using the right grain for SUSE systems
  - 7a0e4be4f8 Add unit test for get_zone and various platforms

- **ISSUE** #41105: (terminalmage) ssl_verify gitfs/git_pillar option does not work with pygit2 (refs: #41111)
- **PR** #41111: (terminalmage) Allow "ssl_verify: False" to work with pygit2 @ *2017-05-09 17:56:12 UTC*
  - 6fa41dc89d Merge pull request #41111 from terminalmage/issue41105
  - 8c6410e3cd Add notices about ssl_verify only working in 0.23.2 and newer

  - 98ce829729 Support ssl_verify in pygit2
  - f73c4b7167 Add http(s) auth config docs for GitPython
- **PR** #41008: (cro) Look in /opt//_lib instead of just /opt/local/lib on Illumos distros. @ *2017-05-09 16:56:00 UTC_
  - 81add1b944 Merge pull request #41008 from cro/rsax_smos
  - a4f7aa145e Look for libcrypto in both /opt/tools and /opt/local on Illumos-based distros.
- **PR** #41124: (gtmanfred) add user_data to digitalocean @ *2017-05-09 16:47:42 UTC*
  - c649725e9b Merge pull request #41124 from gtmanfred/do
  - 2370d9316b add user_data to digital ocean
- **ISSUE** #41125: (tmeneau) service.running returns True if enable=None and init script returns 0 (refs: #41127)
- **PR** #41127: (tmeneau) Fix incorrect service.running state response when enable=None and init script returns 0 @ *2017-05-09 16:43:35 UTC*
  - d0a3fcf33a Merge pull request #41127 from xetus-oss/fix-41125-service-running
  - d8766562c9 fix incorrect service.running success response

28.2.38 Salt 2016.11.7 Release Notes

Version 2016.11.7 is a bugfix release for *2016.11.0*.

Security Fix

CVE-2017-12791 Maliciously crafted minion IDs can cause unwanted directory traversals on the Salt-master

This release corrects a flaw in minion ID validation which could allow certain minions to authenticate to a master despite not having the correct credentials. To exploit the vulnerability, an attacker must create a salt-minion with an ID containing characters that will cause a directory traversal. Credit for discovering the security flaw goes to: Vernhk@qq.com

28.2.39 Salt 2016.11.8 Release Notes

Version 2016.11.8 is a bugfix release for *2016.11.0*.

Statistics

- Total Merges: **171**
- Total Issue References: **68**
- Total PR References: **202**
- Contributors: **61** (AFriemann, Ch3LL, CorvinM, Da-Juan, DmitryKuzmenko, UtahDave, abulford, amalleo25, amendlik, aneeshusa, aogier, arount, arthurlogilab, astronouth7303, binocvlar, blarghmatey, cachedout, clem-compilatio, corywright, cri-epita, damon-atkins, davidjb, dglloyd, dmurphy18, ferringb, garethgreenaway, gdubroeucq, gilbsgilbs, goten4, gtmanfred, isbm, jagguli, kevinanderson1, kojiromike, kstreee, leeclemens, lomeroe, lorengordon, lubyou, mcarlton00, meaksh, morganwillcock, nhavens, pabloh007, rallytime, remi-jouannet, renner, root360-AndreasUlm, s-sebastian, sarcasticadmin, sbojarski, shengis, tdutrion, terminal-mage, toanju, twangboy, ushmodin, viktorkrivak, vutny, whiteinge, xiaoanyunfei)

Security Fix

CVE-2017-14695 Directory traversal vulnerability in minion id validation in SaltStack. Allows remote minions with incorrect credentials to authenticate to a master via a crafted minion ID. Credit for discovering the security flaw goes to: Julian Brost (julian@0x4a42.net)

CVE-2017-14696 Remote Denial of Service with a specially crafted authentication request. Credit for discovering the security flaw goes to: Julian Brost (julian@0x4a42.net)

Anonymous Binds and LDAP/Active Directory

When `auth.ldap.anonymous` is set to `False`, the bind password can no longer be empty.

Changelog for v2016.11.7..v2016.11.8

Generated at: 2018-05-27 20:23:07 UTC

- **PR** #43508: (rallytime) Back-port #43333 to 2016.11.8 @ *2017-09-14 21:40:19 UTC*
  - **PR** #43333: (damon-atkins) Docs are wrong cache_dir (bool) and cache_file (str) cannot be passed as params + 1 bug (refs: #43508)
  - a648f75949 Merge pull request #43508 from rallytime/bp-43333
  - d4981a2717 Update doco
  - a7c8b9e048 Update win_pkg.py
  - 1d6dc6fb72 Docs are wrong cache_dir (bool) and cache_file (str) cannot be passed on the cli (#2)
- **PR** #43434: (rallytime) Add 2016.11.8 release notes @ *2017-09-11 17:06:29 UTC*
  - e7009877bc Merge pull request #43434 from rallytime/2016.11.8-release-notes
  - 68f529ee5e Add 2016.11.8 release notes
- **PR** #43271: (twangboy) Fix minor formatting issue @ *2017-08-30 18:35:12 UTC*
  - cf21f91fb2 Merge pull request #43271 from twangboy/win_fix_pkg.install
  - 91b062f564 Fix formatting issue, spaces surrounding +
- **PR** #43228: (twangboy) Win fix pkg.install @ *2017-08-30 14:26:21 UTC*
  - 3a0b02f3ae Merge pull request #43228 from twangboy/win_fix_pkg.install
  - 13dfabb1ce Fix regex statement, add .
  - 31ff69f0ad Add underscore to regex search
  - 3cf2b6575c Fix spelling
  - ed030a35a5 Use regex to detect salt-minion install
  - e5daff495a Fix pkg.install
- **PR** #43191: (viktorkrivak) Fix apache.config with multiple statement @ *2017-08-28 18:13:44 UTC*
  - b4c689dff5 Merge pull request #43191 from viktorkrivak/fix-apache-config-multi-entity
  - c15bcbe1cc Merge remote-tracking branch 'upstream/2016.11' into fix-apache-config-multi-entity

  - 4164047951 Fix apache.config with multiple statement At this moment when you post more than one statement in config only last is used. Also file is rewrited multiple times until last statement is written. Example: salt '*' apache.config /etc/httpd/conf.d/ports.conf config="[{'Listen': '8080'}, {'Proxy': "Something"}]" Ends only with Proxy Something and ignore Listen 8080, This patch fix this issue.
- **ISSUE** #42279: (dafyddj) win_lgpo matches multiple policies due to startswith() (refs: #43154, #43116)
- **PR** #43154: (lomeroe) Backport #43116 to 2016.11 @ *2017-08-28 16:40:41 UTC*
  - **PR** #43116: (lomeroe) Fix 42279 in develop (refs: #43154)
  - b90e59ede9 Merge pull request #43154 from lomeroe/bp-43116-2016.11
  - 8f593b0b02 verify that files exist before trying to remove them, win_file.remove raises an exception if the file does not exist
  - 33a30bac06 correcting bad format statement in search for policy to be disabled
  - acc3d7ac82 correct fopen calls from salt.utils for 2016.11's utils function
  - 2da1cdd109 lint fix
  - 61bd12c0de track xml namespace to ensure policies w/duplicate IDs or Names do not conflict
  - f232bed9f9 add additional checks for ADM policies that have the same ADMX policy ID (#42279)
- **ISSUE** #42642: (githubcdr) state.augeas (refs: #42669, #43202)
- **PR** #43202: (garethgreenaway) Reverting previous augeas module changes @ *2017-08-28 13:14:27 UTC*
  - 5308c27f9f Merge pull request #43202 from garethgreenaway/42642_2016_11_augeas_module_revert_fix
  - ef7e93eb3f Reverting this change due to it breaking other uses.
- **ISSUE** #43101: (aogier) genesis.bootstrap fails if no pkg AND exclude_pkgs (which can't be a string) (refs: #43103)
- **PR** #43103: (aogier) genesis.bootstrap deboostrap fix @ *2017-08-25 20:48:23 UTC*
  - f16b7246e4 Merge pull request #43103 from aogier/43101-genesis-bootstrap
  - db94f3bb1c better formatting
  - e5cc667762 tests: fix a leftover and simplify some parts
  - 13e5997457 lint
  - 216ced69e5 allow comma-separated pkgs lists, quote args, test deb behaviour
  - d8612ae006 fix debootstrap and enhance packages selection/deletion via cmdline
- **ISSUE** #42329: (jagguli) State git.latest does not pull latest tags (refs: #42663)
- **PR** #42663: (jagguli) Check remote tags before deciding to do a fetch #42329 @ *2017-08-25 20:14:32 UTC*
  - 4863771428 Merge pull request #42663 from StreetHawkInc/fix_git_tag_check
  - 2b5af5b59d Remove refs/tags prefix from remote tags
  - 3f2e96e561 Convert set to list for serializer
  - 2728e5d977 Only include new tags in changes
  - 4b1df2f223 Exclude annotated tags from checks
  - 389c037285 Check remote tags before deciding to do a fetch #42329
- **ISSUE** #43198: (corywright) disk.format_ needs to be aliased to disk.format (refs: #43199)
- **PR** #43199: (corywright) Add *disk.format* alias for *disk.format_* @ *2017-08-25 19:21:07 UTC*

  - 4193e7f0a2 Merge pull request #43199 from corywright/disk-format-alias
  - f00d3a9ddc Add *disk.format* alias for *disk.format_*
- **ISSUE** saltstack/salt-jenkins#495: (Ch3LL) npm tests failing (refs: #43196)
- **PR** #43196: (gtmanfred) Pin request install to version for npm tests @ *2017-08-25 18:43:06 UTC*
  - 5471f9fe0c Merge pull request #43196 from gtmanfred/2016.11
  - ccd2241777 Pin request install to version
- **ISSUE** #43143: (abulford) git.detached does not fetch if rev is missing from local (refs: #43178)
- **PR** #43178: (terminalmage) git.detached: Fix traceback when rev is a SHA and is not present locally @ *2017-08-25 13:58:37 UTC*
  - ace2715c60 Merge pull request #43178 from terminalmage/issue43143
  - 2640833400 git.detached: Fix traceback when rev is a SHA and is not present locally
- **PR** #43179: (terminalmage) Fix missed deprecation @ *2017-08-24 22:52:34 UTC*
  - 12e9507b9e Merge pull request #43179 from terminalmage/old-deprecation
  - 3adf8ad04b Fix missed deprecation
- **PR** #43171: (terminalmage) Add warning about adding new functions to salt/utils/__init__.py @ *2017-08-24 19:10:23 UTC*
  - b595440d90 Merge pull request #43171 from terminalmage/salt-utils-warning
  - 7b5943a31a Add warning about adding new functions to salt/utils/__init__.py
- **PR** #43173: (Ch3LL) Add New Release Branch Strategy to Contribution Docs @ *2017-08-24 19:04:56 UTC*
  - 4f273cac4f Merge pull request #43173 from Ch3LL/add_branch_docs
  - 1b24244bd3 Add New Release Branch Strategy to Contribution Docs
- **PR** #43151: (ushmodin) state.sls hangs on file.recurse with clean: True on windows @ *2017-08-23 17:25:33 UTC*
  - **PR** #42969: (ushmodin) state.sls hangs on file.recurse with clean: True on windows (refs: #43151)
  - 669b376abf Merge pull request #43151 from ushmodin/2016.11
  - c5841e2ade state.sls hangs on file.recurse with clean: True on windows
- **PR** #42986: (renner) Notify systemd synchronously (via NOTIFY_SOCKET) @ *2017-08-22 16:52:56 UTC*
  - ae9d2b7985 Merge pull request #42986 from renner/systemd-notify
  - 79c53f3f81 Fallback to systemd_notify_call() in case of socket.error
  - f1765472dd Notify systemd synchronously (via NOTIFY_SOCKET)
- **ISSUE** #43036: (mcarlton00) Linux VMs in Bhyve aren't displayed properly in grains (refs: #43037)
- **PR** #43037: (mcarlton00) Issue #43036 Bhyve virtual grain in Linux VMs @ *2017-08-22 16:43:40 UTC*
  - b420fbe618 Merge pull request #43037 from mcarlton00/fix-bhyve-grains
  - 73315f0cf0 Issue #43036 Bhyve virtual grain in Linux VMs
- **PR** #43100: (vutny) [DOCS] Add missing *utils* sub-dir listed for *extension_modules* @ *2017-08-22 15:40:09 UTC*
  - 0a86f2d884 Merge pull request #43100 from vutny/doc-add-missing-utils-ext
  - af743ff6c3 [DOCS] Add missing *utils* sub-dir listed for *extension_modules*
- **ISSUE** #15171: (JensRantil) Maximum recursion limit hit related to requisites (refs: #42985)

- **PR** #42985: (DmitryKuzmenko) Properly handle *prereq* having lost requisites. @ *2017-08-21 22:49:39 UTC*
  - e2bf2f448e Merge pull request #42985 from DSRCorporation/bugs/15171_recursion_limit
  - 651b1bab09 Properly handle *prereq* having lost requisites.
- **PR** #43092: (blarghmatey) Fixed issue with silently passing all tests in Testinfra module @ *2017-08-21 20:22:08 UTC*
  - e51333306c Merge pull request #43092 from mitodl/2016.11
  - d4b113acdf Fixed issue with silently passing all tests in Testinfra module
- **PR** #43060: (twangboy) Osx update pkg scripts @ *2017-08-21 20:06:12 UTC*
  - 77a443ce8e Merge pull request #43060 from twangboy/osx_update_pkg_scripts
  - ef8a14cdf9 Remove /opt/salt instead of /opt/salt/bin
  - 2dd62aa1da Add more information to the description
  - f44f5b70dc Only stop services if they are running
  - 3b62bf953c Remove salt from the path
  - ebdca3a0f5 Update pkg-scripts
- **ISSUE** #42869: (abednarik) Git Module : Failed to update repository (refs: #43064)
- **PR** #43064: (terminalmage) Fix race condition in git.latest @ *2017-08-21 14:29:52 UTC*
  - 1b1b6da803 Merge pull request #43064 from terminalmage/issue42869
  - 093c0c2f77 Fix race condition in git.latest
- **ISSUE** #42041: (lorengordon) pkg.list_repo_pkgs fails to find pkgs with spaces around yum repo enabled value (refs: #43054)
- **PR** #43054: (lorengordon) Uses ConfigParser to read yum config files @ *2017-08-18 20:49:44 UTC*
  - **PR** #42045: (arount) Fix: salt.modules.yumpkg: ConfigParser to read ini like files. (refs: #43054)
  - 96e8e836d1 Merge pull request #43054 from lorengordon/fix/yumpkg/config-parser
  - 3b2cb81a72 fix typo in salt.modules.yumpkg
  - 38add0e4a2 break if leading comments are all fetched
  - d7f65dc7a7 fix configparser import & log if error was raised
  - ca1b1bb633 use configparser to parse yum repo file
- **PR** #43048: (rallytime) Back-port #43031 to 2016.11 @ *2017-08-18 12:56:04 UTC*
  - **PR** #43031: (gtmanfred) use a ruby gem that doesn't have dependencies (refs: #43048)
  - 43aa46f512 Merge pull request #43048 from rallytime/bp-43031
  - 35e45049e2 use a ruby gem that doesn't have dependencies
- **PR** #43023: (terminalmage) Fixes/improvements to Jenkins state/module @ *2017-08-18 01:33:10 UTC*
  - ad89ff3104 Merge pull request #43023 from terminalmage/fix-jenkins-xml-caching
  - 33fd8ff939 Update jenkins.py
  - fc306fc8c3 Add missing colon in *if* statement
  - 822eabcc81 Catch exceptions raised when making changes to jenkins
  - 91b583b493 Improve and correct exception raising

  - f096917a0e Raise an exception if we fail to cache the config xml
- **PR** #43026: (rallytime) Back-port #43020 to 2016.11 @ *2017-08-17 23:19:46 UTC*
  - **PR** #43020: (gtmanfred) test with gem that appears to be abandoned (refs: #43026)
  - 2957467ed7 Merge pull request #43026 from rallytime/bp-43020
  - 0eb15a1f67 test with gem that appears to be abandoned
- **ISSUE** #40490: (alxwr) saltstack x509 incompatible to m2crypto 0.26.0 (refs: #42760)
- **PR** #43033: (rallytime) Back-port #42760 to 2016.11 @ *2017-08-17 22:24:43 UTC*
  - **PR** #42760: (AFriemann) Catch TypeError thrown by m2crypto when parsing missing subjects in c… (refs: #43033)
  - 4150b094fe Merge pull request #43033 from rallytime/bp-42760
  - 3e3f7f5d8e Catch TypeError thrown by m2crypto when parsing missing subjects in certificate files.
- **PR** #43032: (rallytime) Back-port #42547 to 2016.11 @ *2017-08-17 21:53:50 UTC*
  - **PR** #42547: (blarghmatey) Updated testinfra modules to work with more recent versions (refs: #43032)
  - b124d3667e Merge pull request #43032 from rallytime/bp-42547
  - ea4d7f4176 Updated testinfra modules to work with more recent versions
- **ISSUE** #42992: (pabloh007) docker.save flag push does is ignored (refs: #43027)
- **PR** #43027: (pabloh007) Fixes ignore push flag for docker.push module issue #42992 @ *2017-08-17 19:55:37 UTC*
  - a88386ad44 Merge pull request #43027 from pabloh007/fix-docker-save-push-2016-11
  - d0fd949f85 Fixes ignore push flag for docker.push module issue #42992
- **ISSUE** #42627: (taigrrr8) salt-cp no longer works. Was working a few months back. (refs: #42890)
- **PR** #42890: (DmitryKuzmenko) Make chunked mode in salt-cp optional @ *2017-08-17 18:37:44 UTC*
  - 51d16840bb Merge pull request #42890 from DSRCorporation/bugs/42627_salt-cp
  - cfddbf1c75 Apply code review: update the doc
  - afedd3b654 Typos and version fixes in the doc.
  - 9fedf6012e Fixed 'test_valid_docs' test.
  - 999388680c Make chunked mode in salt-cp optional (disabled by default).
- **PR** #43009: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-08-17 18:00:09 UTC*
  - b3c253cdfa Merge pull request #43009 from rallytime/merge-2016.11
  - 566ba4fe76 Merge branch '2016.3' into '2016.11'
    - 13b8637d53 Merge pull request #42942 from Ch3LL/2016.3.6_follow_up
      - f281e1795f move additional minion config options to 2016.3.8 release notes
      - 168604ba6b remove merge conflict
      - 8a07d95212 update release notes with cve number
      - 149633fdca Add release notes for 2016.3.7 release
      - 7a4cddcd95 Add clean_id function to salt.utils.verify.py
    - bbb1b29ccb Merge pull request #42954 from Ch3LL/latest_2016.3

        - b551e66744 [2016.3] Bump latest and previous versions
      - 5d5edc54b7 Merge pull request #42949 from Ch3LL/2016.3.7_docs
        - d75d3741f8 Add Security Notice to 2016.3.7 Release Notes

- **PR** #43021: (terminalmage) Use socket.AF_INET6 to get the correct value instead of doing an OS check @ *2017-08-17 17:57:09 UTC*
  - **PR** #43014: (Ch3LL) Change AF_INET6 family for mac in test_host_to_ips (refs: #43021)
  - 37c63e7cf2 Merge pull request #43021 from terminalmage/fix-network-test
  - 4089b7b1bc Use socket.AF_INET6 to get the correct value instead of doing an OS check
- **PR** #43019: (rallytime) Update bootstrap script to latest stable: v2017.08.17 @ *2017-08-17 17:56:41 UTC*
  - 8f6423247c Merge pull request #43019 from rallytime/bootstrap_2017.08.17
  - 2f762b3a17 Update bootstrap script to latest stable: v2017.08.17
- **PR** #43014: (Ch3LL) Change AF_INET6 family for mac in test_host_to_ips (refs: #43021) @ *2017-08-17 16:17:51 UTC*
  - ff1caeee68 Merge pull request #43014 from Ch3LL/fix_network_mac
  - b8eee4401e Change AF_INET6 family for mac in test_host_to_ips
- **PR** #42968: (vutny) [DOCS] Fix link to Salt Cloud Feature Matrix @ *2017-08-16 13:16:16 UTC*
  - 1ee9499d28 Merge pull request #42968 from vutny/doc-salt-cloud-ref
  - 44ed53b1df [DOCS] Fix link to Salt Cloud Feature Matrix
- **ISSUE** #38839: (DaveOHenry) Invoking runner.cloud.action via reactor sls fails (refs: #42291)
- **PR** #42291: (vutny) Fix #38839: remove *state* from Reactor runner kwags @ *2017-08-15 23:01:08 UTC*
  - 923f9741fe Merge pull request #42291 from vutny/fix-38839
  - 5f8f98a01f Fix #38839: remove *state* from Reactor runner kwags
- **ISSUE** #42644: (stamak) nova salt-cloud -P Private IPs returned, but not public. Checking for misidentified IPs (refs: #42940)
- **PR** #42940: (gtmanfred) create new ip address before checking list of allocated ips @ *2017-08-15 21:47:18 UTC*
  - c20bc7d515 Merge pull request #42940 from gtmanfred/2016.11
  - 253e216a8d fix IP address spelling
  - bd63074e7a create new ip address before checking list of allocated ips
- **PR** #42959: (rallytime) Back-port #42883 to 2016.11 @ *2017-08-15 21:25:48 UTC*
  - **PR** #42883: (rallytime) Fix failing boto tests (refs: #42959)
  - d6496eca72 Merge pull request #42959 from rallytime/bp-42883
  - c6b9ca4b9e Lint fix: add missing space
  - 5597b1a30e Skip 2 failing tests in Python 3 due to upstream bugs
  - a0b19bdc27 Update account id value in boto_secgroup module unit test
  - 60b406e088 @mock_elb needs to be changed to @mock_elb_deprecated as well
  - 6ae1111295 Replace @mock_ec2 calls with @mock_ec2_deprecated calls
- **PR** #42944: (Ch3LL) [2016.11] Add clean_id function to salt.utils.verify.py @ *2017-08-15 18:06:12 UTC*

  - 6366e05d0d Merge pull request #42944 from Ch3LL/2016.11.6_follow_up
  - 7e0a20afca Add release notes for 2016.11.7 release
  - 63823f8c3e Add clean_id function to salt.utils.verify.py
- **PR** #42952: (Ch3LL) [2016.11] Bump latest and previous versions @ *2017-08-15 17:23:02 UTC*
  - 49d339c976 Merge pull request #42952 from Ch3LL/latest_2016.11
  - 74e7055d54 [2016.11] Bump latest and previous versions
- **PR** #42950: (Ch3LL) Add Security Notice to 2016.11.7 Release Notes @ *2017-08-15 16:50:23 UTC*
  - b0d2e05a79 Merge pull request #42950 from Ch3LL/2016.11.7_docs
  - a6f902db40 Add Security Notice to 2016.11.77 Release Notes
- **PR** #42836: (aneeshusa) Backport salt.utils.versions from develop to 2016.11 @ *2017-08-14 20:56:54 UTC*
  - **PR** #42835: (aneeshusa) Fix typo in utils/versions.py module (refs: #42836)
  - c0ff69f88c Merge pull request #42836 from lyft/backport-utils.versions-to-2016.11
  - 86ce7004a2 Backport salt.utils.versions from develop to 2016.11
- **PR** #42919: (rallytime) Back-port #42871 to 2016.11 @ *2017-08-14 20:44:00 UTC*
  - **PR** #42871: (amalleo25) Update joyent.rst (refs: #42919)
  - 64a79dd5ac Merge pull request #42919 from rallytime/bp-42871
  - 4e46c968e6 Update joyent.rst
- **ISSUE** #42803: (gmcwhistler) master_type: str, not working as expected, parent salt-minion process dies. (refs: #42848)
- **ISSUE** #42753: (grichmond-salt) SaltReqTimeout Error on Some Minions when One Master in a Multi-Master Configuration is Unavailable (refs: #42848)
- **PR** #42918: (rallytime) Back-port #42848 to 2016.11 @ *2017-08-14 20:43:43 UTC*
  - **PR** #42848: (DmitryKuzmenko) Execute fire_master asynchronously in the main minion thread. (refs: #42918)
  - bea8ec1098 Merge pull request #42918 from rallytime/bp-42848
  - cdb48126f7 Make lint happier.
  - 62eca9b00b Execute fire_master asynchronously in the main minion thread.
- **PR** #42861: (twangboy) Fix pkg.install salt-minion using salt-call @ *2017-08-14 19:07:22 UTC*
  - 52bce329cb Merge pull request #42861 from twangboy/win_pkg_install_salt
  - 0d3789f0c6 Fix pkg.install salt-minion using salt-call
- **PR** #42798: (s-sebastian) Update return data before calling returners @ *2017-08-14 15:51:30 UTC*
  - b9f4f87aa5 Merge pull request #42798 from s-sebastian/2016.11
  - 1cc86592ed Update return data before calling returners
- **ISSUE** #41976: (abulford) dockerng network states do not respect test=True (refs: #41977)
- **PR** #41977: (abulford) Fix dockerng.network_* ignoring of tests=True @ *2017-08-11 18:37:20 UTC*
  - c15d0034fe Merge pull request #41977 from redmatter/fix-dockerng-network-ignores-test
  - 1cc2aa503a Fix dockerng.network_* ignoring of tests=True

- **PR** #42886: (sarcasticadmin) Adding missing output flags to salt cli docs @ *2017-08-11 18:35:19 UTC*
  - 3b9c3c5671 Merge pull request #42886 from sarcasticadmin/adding_docs_salt_outputs
  - 744bf954ff Adding missing output flags to salt cli
- **PR** #42882: (gtmanfred) make sure cmd is not run when npm isn't installed @ *2017-08-11 17:53:14 UTC*
  - e5b98c8a88 Merge pull request #42882 from gtmanfred/2016.11
  - da3402a53d make sure cmd is not run when npm isn't installed
- **PR** #42788: (amendlik) Remove waits and retries from Saltify deployment @ *2017-08-11 15:38:05 UTC*
  - 5962c9588b Merge pull request #42788 from amendlik/saltify-timeout
  - 928b523797 Remove waits and retries from Saltify deployment
- **PR** #42877: (terminalmage) Add virtual func for cron state module @ *2017-08-11 15:33:09 UTC*
  - 227ecddd13 Merge pull request #42877 from terminalmage/add-cron-state-virtual
  - f1de196740 Add virtual func for cron state module
- **PR** #42859: (terminalmage) Add note about git CLI requirement for GitPython to GitFS tutorial @ *2017-08-11 14:53:03 UTC*
  - ab9f6cef33 Merge pull request #42859 from terminalmage/gitpython-git-cli-note
  - 35e05c9515 Add note about git CLI requirement for GitPython to GitFS tutorial
- **ISSUE** saltstack/salt-jenkins#475: (rallytime) Arch is failing npm cache test (refs: #42856)
- **ISSUE** #41770: (Ch3LL) NPM v5 incompatible with salt.modules.cache_list (refs: #42856)
- **PR** #42856: (gtmanfred) skip cache_clean test if npm version is >= 5.0.0 @ *2017-08-11 13:39:20 UTC*
  - 682b4a8d14 Merge pull request #42856 from gtmanfred/2016.11
  - b458b89fb8 skip cache_clean test if npm version is >= 5.0.0
- **PR** #42864: (whiteinge) Make syndic_log_file respect root_dir setting @ *2017-08-11 13:28:21 UTC*
  - 01ea854029 Merge pull request #42864 from whiteinge/syndic-log-root_dir
  - 4b1f55da9c Make syndic_log_file respect root_dir setting
- **PR** #42851: (terminalmage) Backport #42651 to 2016.11 @ *2017-08-10 18:02:39 UTC*
  - **PR** #42651: (gtmanfred) python2- prefix for fedora 26 packages (refs: #42851)
  - 2dde1f77e9 Merge pull request #42851 from terminalmage/bp-42651
  - a3da86eea8 fix syntax
  - 6ecdbcec1d make sure names are correct
  - f83b553d6e add py3 for versionlock
  - 21934f61bb python2- prefix for fedora 26 packages
- **ISSUE** #42683: (rgcosma) Gluster module broken in 2017.7 (refs: #42806)
- **PR** #42806: (rallytime) Update doc references in glusterfs.volume_present @ *2017-08-10 14:10:16 UTC*
  - c746f79a3a Merge pull request #42806 from rallytime/fix-42683
  - 8c8640d6b8 Update doc references in glusterfs.volume_present
- **PR** #42829: (twangboy) Fix passing version in pkgs as shown in docs @ *2017-08-10 14:07:24 UTC*

  - 27a8a2695a Merge pull request #42829 from twangboy/win_pkg_fix_install
  - 83b9b230cd Add winrepo to docs about supporting versions in pkgs
  - 81fefa6e67 Add ability to pass version in pkgs list
- **PR** #42838: (twangboy) Document requirements for win_pki @ *2017-08-10 13:59:46 UTC*
  - 3c3ac6aeb2 Merge pull request #42838 from twangboy/win_doc_pki
  - f0a1d06b46 Standardize PKI Client
  - 7de687aa57 Document requirements for win_pki
- **PR** #42805: (rallytime) Back-port #42552 to 2016.11 @ *2017-08-09 22:37:56 UTC*
  - **PR** #42552: (remijouannet) update consul module following this documentation https://www.consul.... (refs: #42805)
  - b3e2ae3c58 Merge pull request #42805 from rallytime/bp-42552
  - 5a91c1f2d1 update consul module following this documentation https://www.consul.io/api/acl.html
- **ISSUE** #42731: (infoveinx) http.query template_data render exception (refs: #42804)
- **PR** #42804: (rallytime) Back-port #42784 to 2016.11 @ *2017-08-09 22:37:40 UTC*
  - **PR** #42784: (gtmanfred) only read file if ret is not a string in http.query (refs: #42804)
  - d2ee7934ed Merge pull request #42804 from rallytime/bp-42784
  - dbd29e4aaa only read file if it is not a string
- **PR** #42826: (terminalmage) Fix misspelling of "versions" @ *2017-08-09 19:39:43 UTC*
  - 4cbf8057b3 Merge pull request #42826 from terminalmage/fix-spelling
  - 00f93142e4 Fix misspelling of "versions"
- **PR** #42786: (Ch3LL) Fix typo for template_dict in http docs @ *2017-08-08 18:14:50 UTC*
  - de997edd90 Merge pull request #42786 from Ch3LL/fix_typo
  - 90a2fb66a2 Fix typo for template_dict in http docs
- **ISSUE** #42600: (twangboy) Unable to set 'Not Configured' using win_lgpo execution module (refs: #42795, #42744)
- **PR** #42795: (lomeroe) backport #42744 to 2016.11 @ *2017-08-08 17:17:15 UTC*
  - **PR** #42744: (lomeroe) fix #42600 in develop (refs: #42795)
  - bf6153ebe5 Merge pull request #42795 from lomeroe/bp-42744_201611
  - 695f8c1ae4 fix #42600 in develop
- **ISSUE** #42747: (whiteinge) Outputters mutate data which can be a problem for Runners and perhaps other things (refs: #42748)
- **PR** #42748: (whiteinge) Workaround Orchestrate problem that highstate outputter mutates data @ *2017-08-07 21:11:33 UTC*
  - 61fad97286 Merge pull request #42748 from whiteinge/save-before-output
  - de60b77c82 Workaround Orchestrate problem that highstate outputter mutates data
- **PR** #42764: (amendlik) Fix infinite loop with salt-cloud and Windows nodes @ *2017-08-07 20:47:07 UTC*
  - a4e3e7e786 Merge pull request #42764 from amendlik/cloud-win-loop

  - f3dcfca4e0 Fix infinite loops on failed Windows deployments
- **ISSUE** #42690: (ChristianBeer) git.latest state with remote set fails on first try (refs: #42694)
- **PR** #42694: (gtmanfred) allow adding extra remotes to a repository @ *2017-08-07 18:08:11 UTC*
  - da85326ad4 Merge pull request #42694 from gtmanfred/2016.11
  - 1a0457af51 allow adding extra remotes to a repository
- **ISSUE** #42642: (githubcdr) state.augeas (refs: #42669, #43202)
- **PR** #42669: (garethgreenaway) [2016.11] Fixes to augeas module @ *2017-08-06 17:58:03 UTC*
  - 7b2119feee Merge pull request #42669 from garethgreenaway/42642_2016_11_augeas_module_fix
  - 24413084e2 Updating the call to shlex_split to pass the posix=False argument so that quotes are preserved.
- **PR** #42629: (xiaoanyunfei) tornado api @ *2017-08-03 22:21:20 UTC*
  - 30725769ed Merge pull request #42629 from xiaoanyunfei/tornadoapi
  - 1e13383b95 tornado api
- **PR** #42655: (whiteinge) Reenable cpstats for rest_cherrypy @ *2017-08-03 20:44:10 UTC*
  - **PR** #33806: (cachedout) Work around upstream cherrypy bug (refs: #42655)
  - f0f00fcee1 Merge pull request #42655 from whiteinge/rest_cherrypy-reenable-stats
  - deb6316d67 Fix lint errors
  - 6bd91c8b03 Reenable cpstats for rest_cherrypy
- **ISSUE** #42686: (gilbsgilbs) Unable to set multiple RabbitMQ tags (refs: #42693)
- **PR** #42693: (gilbsgilbs) Fix RabbitMQ tags not properly set. @ *2017-08-03 20:23:08 UTC*
  - 21cf15f9c3 Merge pull request #42693 from gilbsgilbs/fix-rabbitmq-tags
  - 78fccdc7e2 Cast to list in case tags is a tuple.
  - 287b57b5c5 Fix RabbitMQ tags not properly set.
- **ISSUE** #41433: (sbojarski) boto_cfn.present fails when reporting error for failed state (refs: #42574)
- **PR** #42574: (sbojarski) Fixed error reporting in "boto_cfn.present" function. @ *2017-08-01 17:55:29 UTC*
  - f2b0c9b4fa Merge pull request #42574 from sbojarski/boto-cfn-error-reporting
  - 5c945f10c2 Fix debug message in "boto_cfn._validate" function.
  - 181a1beecc Fixed error reporting in "boto_cfn.present" function.
- **PR** #42623: (terminalmage) Fix unicode constructor in custom YAML loader @ *2017-07-31 19:25:18 UTC*
  - bc1effc4f2 Merge pull request #42623 from terminalmage/fix-unicode-constructor
  - fcf45889dd Fix unicode constructor in custom YAML loader
- **PR** #42515: (gtmanfred) Allow not interpreting backslashes in the repl @ *2017-07-28 16:00:09 UTC*
  - cbf752cd73 Merge pull request #42515 from gtmanfred/backslash
  - cc4e45656d Allow not interpreting backslashes in the repl
- **ISSUE** #42456: (gdubroeucq) Use yum lib (refs: #42586)
- **PR** #42586: (gdubroeucq) [Fix] yumpkg.py: add option to the command "check-update" @ *2017-07-27 23:52:00 UTC*

  - 549495831f Merge pull request #42586 from gdubroeucq/2016.11
  - 9c0b5cc1d6 Remove extra newline
  - d2ef4483e4 yumpkg.py: clean
  - a96f7c09e0 yumpkg.py: add option to the command "check-update"
- **ISSUE** #41982: (abulford) dockerng.network_* matches too easily (refs: #41988)
- **PR** #41988: (abulford) Fix dockerng.network_* name matching @ *2017-07-27 21:25:06 UTC*
  - 6b45debf28 Merge pull request #41988 from redmatter/fix-dockerng-network-matching
  - 9eea796da8 Add regression tests for #41982
  - 3369f0072f Fix broken unit test test_network_absent
  - 0ef6cf634c Add trace logging of dockerng.networks result
  - 515c612808 Fix dockerng.network_* name matching
- **PR** #42339: (isbm) Bugfix: Jobs scheduled to run at a future time stay pending for Salt minions (bsc#1036125) @ *2017-07-27 19:05:51 UTC*
  - 4b16109122 Merge pull request #42339 from isbm/isbm-jobs-scheduled-in-a-future-bsc1036125
  - bbba84ce2d Bugfix: Jobs scheduled to run at a future time stay pending for Salt minions (bsc#1036125)
- **ISSUE** #23516: (dkiser) BUG: cron job scheduler sporadically works (refs: #42077)
- **PR** #42077: (vutny) Fix scheduled job run on Master if *when* parameter is a list @ *2017-07-27 19:04:23 UTC*
  - **PR** #41973: (vutny) Fix Master/Minion scheduled jobs based on Cron expressions (refs: #42077)
  - 6c5a7c604a Merge pull request #42077 from vutny/fix-jobs-scheduled-with-whens
  - b1960cea44 Fix scheduled job run on Master if *when* parameter is a list
- **PR** #42414: (vutny) DOCS: unify hash sum with hash type format @ *2017-07-27 18:48:40 UTC*
  - f9cb536589 Merge pull request #42414 from vutny/unify-hash-params-format
  - d1f2a93368 DOCS: unify hash sum with hash type format
- **ISSUE** #42375: (dragonpaw) salt.modules.*.__virtualname__ doens't work as documented. (refs: #42523)
- **PR** #42523: (rallytime) Add a mention of the True/False returns with __virtual__() @ *2017-07-27 18:13:07 UTC*
  - 535c922511 Merge pull request #42523 from rallytime/fix-42375
  - 685c2cced6 Add information about returning a tuple with an error message
  - fa466519c4 Add a mention of the True/False returns with __virtual__()
- **PR** #42527: (twangboy) Document changes to Windows Update in Windows 10/Server 2016 @ *2017-07-27 17:45:38 UTC*
  - 0df0e7e749 Merge pull request #42527 from twangboy/win_wua
  - 0373791f2a Correct capatlization
  - af3bcc927b Document changes to Windows Update in 10/2016
- **PR** #42551: (binocvlar) Remove '-s' (–script) argument to parted within align_check function @ *2017-07-27 17:35:31 UTC*
  - 69b06586da Merge pull request #42551 from binocvlar/fix-lack-of-align-check-output
  - c4fabaa192 Remove '-s' (–script) argument to parted within align_check function

- **ISSUE** #42403: (astronouth7303) [2017.7] Pillar empty when state is applied from orchestrate (refs: #42433)
- **PR** #42573: (rallytime) Back-port #42433 to 2016.11 @ *2017-07-27 13:51:21 UTC*
  - **PR** #42433: (terminalmage) Only force saltenv/pillarenv to be a string when not None (refs: #42573)
  - 9e0b4e9faf Merge pull request #42573 from rallytime/bp-42433
  - 0293429e24 Only force saltenv/pillarenv to be a string when not None
- **PR** #42571: (twangboy) Avoid loading system PYTHON* environment vars @ *2017-07-26 22:48:55 UTC*
  - e931ed2517 Merge pull request #42571 from twangboy/win_add_pythonpath
  - d55a44dd1a Avoid loading user site packages
  - 9af1eb2741 Ignore any PYTHON* environment vars already on the system
  - 4e2fb03a95 Add pythonpath to batch files and service
- **ISSUE** #42371: (tsaridas) Minion unresponsive after trying to failover (refs: #42387)
- **PR** #42387: (DmitryKuzmenko) Fix race condition in usage of weakvaluedict @ *2017-07-25 20:57:42 UTC*
  - de2f397041 Merge pull request #42387 from DSRCorporation/bugs/42371_KeyError_WeakValueDict
  - e721c7eee2 Don't use *key in weakvaluedict* because it could lie.
- **ISSUE** #41955: (root360-AndreasUlm) rabbitmq 3.6.10 changed output => rabbitmq-module broken (refs: #41968)
- **PR** #41968: (root360-AndreasUlm) Fix rabbitmqctl output sanitizer for version 3.6.10 @ *2017-07-25 19:12:36 UTC*
  - 641a9d7efd Merge pull request #41968 from root360-AndreasUlm/fix-rabbitmqctl-output-handler
  - 76fd941d91 added tests for rabbitmq 3.6.10 output handler
  - 3602af1e1b Fix rabbitmqctl output handler for 3.6.10
- **ISSUE** #42477: (aikar) Invalid ssh_interface value prevents salt-cloud provisioning without reason of why (refs: #42479)
- **PR** #42479: (gtmanfred) validate ssh_interface for ec2 @ *2017-07-25 18:37:18 UTC*
  - 66fede378a Merge pull request #42479 from gtmanfred/interface
  - c32c1b2803 fix pylint
  - 99ec634c6b validate ssh_interface for ec2
- **ISSUE** #42405: (felrivero) The documentation is incorrectly compiled (PILLAR section) (refs: #42516)
- **PR** #42516: (rallytime) Add info about top file to pillar walk-through example to include edit.vim @ *2017-07-25 17:01:12 UTC*
  - a925c7029a Merge pull request #42516 from rallytime/fix-42405
  - e3a6717efa Add info about top file to pillar walk-through example to include edit.vim
- **ISSUE** #42417: (clem-compilatio) salt-cloud - openstack - "no more floating IP addresses" error - but public_ip in node (refs: #42509)
- **PR** #42509: (clem-compilatio) Fix _assign_floating_ips in openstack.py @ *2017-07-24 17:14:13 UTC*
  - 1bd5bbccc2 Merge pull request #42509 from clem-compilatio/fix-42417
  - 72924b06b8 Fix _assign_floating_ips in openstack.py
- **PR** #42464: (garethgreenaway) [2016.11] Small fix to modules/git.py @ *2017-07-21 21:28:57 UTC*

  - 4bf35a74de Merge pull request #42464 from garethgreenaway/2016_11_remove_tmp_identity_file
  - ff24102d51 Uncomment the line that removes the temporary identity file.
- **ISSUE** #42357: (Giandom) Salt pillarenv problem with slack engine (refs: #42443)
- **PR** #42443: (garethgreenaway) [2016.11] Fix to slack engine @ *2017-07-21 15:48:57 UTC*
  - e2120dbd0e Merge pull request #42443 from garethgreenaway/42357_pass_args_kwargs_correctly
  - 635810b3e3 Updating the slack engine in 2016.11 to pass the args and kwrags correctly to LocalClient
- **ISSUE** #42198: (shengis) state sqlite3.row_absent fail with "parameters are of unsupported type" (refs: #42200)
- **PR** #42200: (shengis) Fix #42198 @ *2017-07-21 14:47:29 UTC*
  - 8262cc9054 Merge pull request #42200 from shengis/sqlite3_fix_row_absent_2016.11
  - 407b8f4bb3 Fix #42198 If where_args is not set, not using it in the delete request.
- **ISSUE** #42413: (goten4) Invalid error message when proxy_host is set and tornado not installed (refs: #42424)
- **PR** #42424: (goten4) Fix error message when tornado or pycurl is not installed @ *2017-07-20 21:53:40 UTC*
  - d9df97e5a3 Merge pull request #42424 from goten4/2016.11
  - 1c0574d05e Fix error message when tornado or pycurl is not installed
- **PR** #42350: (twangboy) Fixes problem with Version and OS Release related grains on certain versions of Python (2016.11) @ *2017-07-19 17:07:26 UTC*
  - 42bb1a64ca Merge pull request #42350 from twangboy/win_fix_ver_grains_2016.11
  - 8c048403d7 Detect Server OS with a desktop release name
- **PR** #42356: (meaksh) Allow to check whether a function is available on the AliasesLoader wrapper @ *2017-07-19 16:56:41 UTC*
  - 0a72e56f6b Merge pull request #42356 from meaksh/2016.11-AliasesLoader-wrapper-fix
  - 915d94219e Allow to check whether a function is available on the AliasesLoader wrapper
- **PR** #42368: (twangboy) Remove build and dist directories before install (2016.11) @ *2017-07-19 16:47:28 UTC*
  - 10eb7b7a79 Merge pull request #42368 from twangboy/win_fix_build_2016.11
  - a7c910c31e Remove build and dist directories before install
- **PR** #42370: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-07-18 22:39:41 UTC*
  - 016189f62f Merge pull request #42370 from rallytime/merge-2016.11
  - 0aa5dde1de Merge branch '2016.3' into '2016.11'
  - e9b0f20f8a Merge pull request #42359 from Ch3LL/doc-update-2016.3
    - dc85b5edbe [2016.3] Update version numbers in doc config for 2017.7.0 release
- **PR** #42360: (Ch3LL) [2016.11] Update version numbers in doc config for 2017.7.0 release @ *2017-07-18 19:23:30 UTC*
  - f06a6f1796 Merge pull request #42360 from Ch3LL/doc-update-2016.11
  - b90b7a7506 [2016.11] Update version numbers in doc config for 2017.7.0 release
- **ISSUE** #32400: (rallytime) Document Default Config Values (refs: #42319)
- **PR** #42319: (rallytime) Add more documentation for config options that are missing from master/minion docs @ *2017-07-18 18:02:32 UTC*

  - e0595b0a0f Merge pull request #42319 from rallytime/config-docs
  - b40f980632 Add more documentation for config options that are missing from master/minion docs
- **ISSUE** #42333: (b3hni4) Getting "invalid type of dict, a list is required" when trying to configure engines in master config file (refs: #42352)
- **PR** #42352: (CorvinM) Multiple documentation fixes @ *2017-07-18 15:10:37 UTC*
  - 78940400e3 Merge pull request #42352 from CorvinM/issue42333
  - 526b6ee14d Multiple documentation fixes
- **PR** #42353: (terminalmage) is_windows is a function, not a propery/attribute @ *2017-07-18 14:38:51 UTC*
  - b256001760 Merge pull request #42353 from terminalmage/fix-git-test
  - 14cf6ce322 is_windows is a function, not a propery/attribute
- **ISSUE** #41116: (hrumph) FAQ has wrong instructions for upgrading Windows minion. (refs: #42264)
- **PR** #42264: (rallytime) Update minion restart section in FAQ doc for windows @ *2017-07-17 17:40:40 UTC*
  - 866a1febb4 Merge pull request #42264 from rallytime/fix-41116
  - bd638880e3 Add mono-spacing to salt-minion reference for consistency
  - 30d62f43da Update minion restart section in FAQ doc for windows
- **ISSUE** #42194: (jryberg) pkg version: latest are now broken, appending <package>-latest to filename (refs: #42275)
- **PR** #42275: (terminalmage) pkg.installed: pack name/version into pkgs argument @ *2017-07-17 17:38:39 UTC*
  - 9a707088ad Merge pull request #42275 from terminalmage/issue42194
  - 663874908a pkg.installed: pack name/version into pkgs argument
- **ISSUE** #41721: (sazaro) state.sysrc broken when setting the value to YES or NO (refs: #42269)
- **PR** #42269: (rallytime) Add some clarity to "multiple quotes" section of yaml docs @ *2017-07-17 17:38:18 UTC*
  - e588f235e0 Merge pull request #42269 from rallytime/fix-41721
  - f2250d474a Add a note about using different styles of quotes.
  - 38d9b3d553 Add some clarity to "multiple quotes" section of yaml docs
- **ISSUE** #42152: (dubb-b) salt-cloud errors on Rackspace driver using -out=yaml (refs: #42282)
- **PR** #42282: (rallytime) Handle libcloud objects that throw RepresenterErrors with –out=yaml @ *2017-07-17 17:36:35 UTC*
  - 5aaa214a75 Merge pull request #42282 from rallytime/fix-42152
  - f032223843 Handle libcloud objects that throw RepresenterErrors with –out=yaml
- **ISSUE** #42295: (lubyou) file.absent fails on windows if the file to be removed has the "readonly" attribute set (refs: #42308)
- **PR** #42308: (lubyou) Force file removal on Windows. Fixes #42295 @ *2017-07-17 17:12:13 UTC*
  - fb5697a4bc Merge pull request #42308 from lubyou/42295-fix-file-absent-windows
  - 026ccf401a Force file removal on Windows. Fixes #42295
- **ISSUE** #42267: (gzcwnk) salt-ssh not creating ssh keys automatically as per documentation (refs: #42314)
- **PR** #42314: (rallytime) Add clarification to salt ssh docs about key auto-generation. @ *2017-07-17 14:07:49 UTC*

  - da2a8a518f Merge pull request #42314 from rallytime/fix-42267
  - c406046940 Add clarification to salt ssh docs about key auto-generation.
- **ISSUE** #41936: (michaelkarrer81) git.latest identity does not set the correct user for the private key file on the minion (refs: #41945)
- **PR** #41945: (garethgreenaway) Fixes to modules/git.py @ *2017-07-14 17:46:10 UTC*
  - acadd54013 Merge pull request #41945 from garethgreenaway/41936_allow_identity_files_with_user
  - 44841e5626 Moving the call to cp.get_file inside the with block to ensure the umask is preserved when we grab the file.
  - f9ba60eed8 Merge pull request #1 from terminalmage/pr-41945
    - 1b6026177c Restrict set_umask to mkstemp call only
  - 68549f3496 Fixing umask to we can set files as executable.
  - 4949bf3ff3 Updating to swap on the new salt.utils.files.set_umask context_manager
  - 8faa9f6d92 Updating PR with requested changes.
  - 494765e939 Updating the git module to allow an identity file to be used when passing the user parameter
- **ISSUE** #42240: (casselt) empty_password in user.present always changes password, even with test=True (refs: #42289)
- **PR** #42289: (CorvinM) Multiple empty_password fixes for state.user @ *2017-07-14 16:14:02 UTC*
  - **PR** #41543: (cri-epita) Fix user creation with empty password (refs: #42289)
  - f90e04a2bc Merge pull request #42289 from CorvinM/bp-41543
  - 357dc22f05 Fix user creation with empty password
- **PR** #42123: (vutny) DOCS: describe importing custom util classes @ *2017-07-12 15:53:24 UTC*
  - a91a3f81b1 Merge pull request #42123 from vutny/fix-master-utils-import
  - 6bb8b8f98c Add missing doc for `utils_dirs` Minion config option
  - f1bc58f6d5 Utils: add example of module import
- **PR** #42261: (rallytime) Some minor doc fixes for dnsutil module so they'll render correctly @ *2017-07-11 23:14:53 UTC*
  - e2aa5114e4 Merge pull request #42261 from rallytime/minor-doc-fix
  - 8c76bbb53d Some minor doc fixes for dnsutil module so they'll render correctly
- **PR** #42262: (rallytime) Back-port #42224 to 2016.11 @ *2017-07-11 23:14:25 UTC*
  - **PR** #42224: (tdutrion) Remove duplicate instruction in Openstack Rackspace config example (refs: #42262)
  - 3e9dfbc9cc Merge pull request #42262 from rallytime/bp-42224
  - c31ded341c Remove duplicate instruction in Openstack Rackspace config example
- **ISSUE** #42137: (kiemlicz) cmd.run with multiple commands - random order of execution (refs: #42181)
- **PR** #42181: (garethgreenaway) fixes to state.py for names parameter @ *2017-07-11 21:21:32 UTC*
  - 7780579c36 Merge pull request #42181 from garethgreenaway/42137_backport_fix_from_2017_7
  - a34970b45b Back porting the fix for 2017.7 that ensures the order of the names parameter.
- **PR** #42253: (gtmanfred) Only use unassociated ips when unable to allocate @ *2017-07-11 20:53:51 UTC*

  - **PR** #38965: (toanju) salt-cloud will use list_floating_ips for OpenStack (refs: #42253)
  - **PR** #34280: (kevinanderson1) salt-cloud will use list_floating_ips for Openstack (refs: #38965)
  - 72537868a6 Merge pull request #42253 from gtmanfred/2016.11
  - 53e25760be Only use unassociated ips when unable to allocate
- **PR** #42252: (UtahDave) simple docstring updates @ *2017-07-11 20:48:33 UTC*
  - b2a4698b5d Merge pull request #42252 from UtahDave/2016.11local
  - e6a9563d47 simple doc updates
- **ISSUE** #42232: (astronouth7303) Half of dnsutil refers to dig (refs: #42235)
- **PR** #42235: (astronouth7303) Abolish references to *dig* in examples. @ *2017-07-10 20:06:11 UTC*
  - 781fe13be7 Merge pull request #42235 from astronouth7303/patch-1-2016.3
  - 4cb51bd03a Make note of dig partial requirement.
  - 08e7d8351a Abolish references to *dig* in examples.
- **PR** #42215: (twangboy) Add missing config to example @ *2017-07-07 20:18:44 UTC*
  - 83cbd76f16 Merge pull request #42215 from twangboy/win_iis_docs
  - c07e22041a Add missing config to example
- **PR** #42211: (terminalmage) Only pass a saltenv in orchestration if one was explicitly passed (2016.11) @ *2017-07-07 20:16:35 UTC*
  - 274946ab00 Merge pull request #42211 from terminalmage/issue40928
  - 22a18fa2ed Only pass a saltenv in orchestration if one was explicitly passed (2016.11)
- **PR** #42173: (rallytime) Back-port #37424 to 2016.11 @ *2017-07-07 16:39:59 UTC*
  - **PR** #37424: (kojiromike) Avoid Early Convert ret['comment'] to String (refs: #42173)
  - 89261cf06c Merge pull request #42173 from rallytime/bp-37424
  - 01addb6053 Avoid Early Convert ret['comment'] to String
- **ISSUE** #39365: (dglloyd) service.running fails if sysv script has no status command and enable: True (refs: #39366)
- **PR** #42175: (rallytime) Back-port #39366 to 2016.11 @ *2017-07-06 19:51:47 UTC*
  - **PR** #39366: (dglloyd) Pass sig to service.status in after_toggle (refs: #42175)
  - 3b17fb7f83 Merge pull request #42175 from rallytime/bp-39366
  - 53f7b987e8 Pass sig to service.status in after_toggle
- **PR** #42172: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-07-06 18:16:29 UTC*
  - ea16f47f0a Merge pull request #42172 from rallytime/merge-2016.11
  - b1fa332a11 Merge branch '2016.3' into '2016.11'
    - 8fa1fa5bb1 Merge pull request #42155 from phsteve/doc-fix-puppet
      - fb2cb78a31 Fix docs for puppet.plugin_sync so code-block renders properly and sync is spelled consistently
- **PR** #42176: (rallytime) Back-port #42109 to 2016.11 @ *2017-07-06 18:15:35 UTC*
  - **PR** #42109: (arthurlogilab) [doc] Update aws.rst - add Debian default username (refs: #42176)

  - 6307b9873f Merge pull request #42176 from rallytime/bp-42109
  - 686926daf7 Update aws.rst - add Debian default username
- **PR** #42095: (terminalmage) Add debug logging to dockerng.login @ *2017-07-06 17:13:05 UTC*
  - 28c4e4c3b7 Merge pull request #42095 from terminalmage/docker-login-debugging
  - bd27870a71 Add debug logging to dockerng.login
- **ISSUE** #42116: (terminalmage) CLI pillar override regression in 2017.7.0rc1 (refs: #42119)
- **PR** #42119: (terminalmage) Fix regression in CLI pillar override for salt-call @ *2017-07-06 17:02:52 UTC*
  - 2b754bc5af Merge pull request #42119 from terminalmage/issue42116
  - 9a268949e3 Add integration test for 42116
  - 1bb42bb609 Fix regression when CLI pillar override is used with salt-call
- **ISSUE** #42114: (clallen) saltenv bug in pillar.get execution module function (refs: #42121)
- **PR** #42121: (terminalmage) Fix pillar.get when saltenv is passed @ *2017-07-06 16:52:34 UTC*
  - 8c0a83cbb5 Merge pull request #42121 from terminalmage/issue42114
  - d14291267f Fix pillar.get when saltenv is passed
- **PR** #42094: (terminalmage) Prevent command from showing in exception when output_loglevel=quiet @ *2017-07-06 16:18:09 UTC*
  - 687992c240 Merge pull request #42094 from terminalmage/quiet-exception
  - 47d61f4edf Prevent command from showing in exception when output_loglevel=quiet
- **ISSUE** #42115: (nomeelnoj) Installing EPEL repo breaks salt-cloud (refs: #42163)
- **PR** #42163: (vutny) Fix #42115: parse libcloud "rc" version correctly @ *2017-07-06 16:15:07 UTC*
  - dad255160c Merge pull request #42163 from vutny/fix-42115
  - b27b1e340a Fix #42115: parse libcloud "rc" version correctly
- **PR** #42164: (Ch3LL) Fix kerberos create_keytab doc @ *2017-07-06 15:55:33 UTC*
  - 2a8ae2b3b6 Merge pull request #42164 from Ch3LL/fix_kerb_doc
  - 7c0fb248ec Fix kerberos create_keytab doc
- **PR** #42141: (rallytime) Back-port #42098 to 2016.11 @ *2017-07-06 15:11:49 UTC*
  - **PR** #42098: (twangboy) Change repo_ng to repo-ng (refs: #42141)
  - 678d4d4098 Merge pull request #42141 from rallytime/bp-42098
  - bd80243233 Change repo_ng to repo-ng
- **PR** #42140: (rallytime) Back-port #42097 to 2016.11 @ *2017-07-06 15:11:29 UTC*
  - **PR** #42097: (gtmanfred) require large timediff for ipv6 warning (refs: #42140)
  - c8afd7a3c9 Merge pull request #42140 from rallytime/bp-42097
  - 9c4e132540 Import datetime
  - 1435bf177e require large timediff for ipv6 warning
- **PR** #42142: (Ch3LL) Update builds available for rc1 @ *2017-07-05 21:11:56 UTC*
  - c239664c8b Merge pull request #42142 from Ch3LL/change_builds

  - e1694af39c Update builds available for rc1

- **PR** #42078: (damon-atkins) pkg.install and pkg.remove fix version number input. @ *2017-07-05 06:04:57 UTC*
  - 4780d7830a Merge pull request #42078 from damon-atkins/fix_convert_flt_str_version_on_cmd_line
  - 09d37dd892 Fix comment typo
  - 7167549425 Handle version=None when converted to a string it becomes 'None' parm should default to empty string rather than None, it would fix better with existing code.
  - 4fb2bb1856 Fix typo
  - cf55c3361c pkg.install and pkg.remove on the command line take number version numbers, store them within a float. However version is a string, to support versions numbers like 1.3.4
- **PR** #42105: (Ch3LL) Update releasecanddiate doc with new 2017.7.0rc1 Release @ *2017-07-04 03:14:42 UTC*
  - 46d575acbc Merge pull request #42105 from Ch3LL/update_rc
  - d4e7b91608 Update releasecanddiate doc with new 2017.7.0rc1 Release
- **ISSUE** #41885: (astronouth7303) Recommended pip installation outdated? (refs: #42099)
- **PR** #42099: (rallytime) Remove references in docs to pip install salt-cloud @ *2017-07-03 22:13:44 UTC*
  - d38548bbbd Merge pull request #42099 from rallytime/fix-41885
  - c2822e05ad Remove references in docs to pip install salt-cloud
- **ISSUE** #42076: (abulford) dockerng.volume_present test looks as though it would cause a change (refs: #42086)
- **PR** #42086: (abulford) Make result=true if Docker volume already exists @ *2017-07-03 15:48:33 UTC*
  - 81d606a8cb Merge pull request #42086 from redmatter/fix-dockerng-volume-present-result
  - 8d549685a7 Make result=true if Docker volume already exists
- **ISSUE** #25842: (shikhartanwar) Running salt-minion as non-root user to execute sudo commands always returns an error (refs: #42021)
- **PR** #42021: (gtmanfred) Set concurrent to True when running states with sudo @ *2017-06-30 21:02:15 UTC*
  - 7160697123 Merge pull request #42021 from gtmanfred/2016.11
  - 26beb18aa5 Set concurrent to True when running states with sudo
- **PR** #42029: (terminalmage) Mock socket.getaddrinfo in unit.utils.network_test.NetworkTestCase.test_host_to_ips @ *2017-06-30 20:58:56 UTC*
  - b784fbbdf8 Merge pull request #42029 from terminalmage/host_to_ips
  - 26f848e111 Mock socket.getaddrinfo in unit.utils.network_test.NetworkTestCase.test_host_to_ips
- **PR** #42055: (dmurphy18) Upgrade support for gnupg v2.1 and higher @ *2017-06-30 20:54:02 UTC*
  - e067020b9b Merge pull request #42055 from dmurphy18/handle_gnupgv21
  - e20cea6350 Upgrade support for gnupg v2.1 and higher
- **PR** #42048: (Ch3LL) Add initial 2016.11.7 Release Notes @ *2017-06-30 16:00:05 UTC*
  - 74ba2abc48 Merge pull request #42048 from Ch3LL/add_11.7
  - 1de5e008a0 Add initial 2016.11.7 Release Notes
- **PR** #42024: (leeclemens) doc: Specify versionadded for SELinux policy install/uninstall @ *2017-06-29 23:29:50 UTC*
  - ca4e619edb Merge pull request #42024 from leeclemens/doc/selinux

  - b63a3c0fae doc: Specify versionadded for SELinux policy install/uninstall
  - **PR** saltstack/salt#41961: (cachedout) Allow docs to be built under Python 3 (refs: #42028)
- **PR** #42030: (whiteinge) Re-add msgpack to mocked imports @ *2017-06-29 20:47:59 UTC*
  - **PR** #42028: (whiteinge) Revert "Allow docs to be built under Python 3" (refs: #42030)
  - 50856d0e28 Merge pull request #42030 from whiteinge/revert-py3-doc-chagnes-pt-2
  - 18dfa9893c Re-add msgpack to mocked imports
  - **PR** saltstack/salt#41961: (cachedout) Allow docs to be built under Python 3 (refs: #42028)
- **PR** #42028: (whiteinge) Revert "Allow docs to be built under Python 3" (refs: #42030) @ *2017-06-29 19:47:46 UTC*
  - 53031d2f55 Merge pull request #42028 from saltstack/revert-41961-py3_doc
  - 5592e6e5d4 Revert "Allow docs to be built under Python 3"
- **ISSUE** #42013: (dusto) Misspelled nozeroconf in salt/modules/rh_ip.py (refs: #42017)
- **PR** #42017: (lorengordon) Fixes typo "nozerconf" -> "nozeroconf" @ *2017-06-29 17:30:48 UTC*
  - 1416bf70b9 Merge pull request #42017 from lorengordon/issue-42013
  - b6cf5f2528 Fixes typo nozerconf -> nozeroconf
- **PR** #41906: (terminalmage) Better support for numeric saltenvs @ *2017-06-29 17:19:33 UTC*
  - 0ebb50b601 Merge pull request #41906 from terminalmage/numeric-saltenv
  - 2d798de982 Better support for numeric saltenvs
- **PR** #41995: (terminalmage) Temporarily set the umask before writing an auth token @ *2017-06-29 01:09:48 UTC*
  - 6a3c03c2d5 Merge pull request #41995 from terminalmage/token-umask
  - 4f54b0069f Temporarily set the umask before writing an auth token
- **PR** #41999: (terminalmage) Update IP address for unit.utils.network_test.NetworkTestCase.test_host_to_ips @ *2017-06-29 01:01:31 UTC*
  - e3801b0e78 Merge pull request #41999 from terminalmage/fix-network-test
  - fb6a93314f Update IP address for unit.utils.network_test.NetworkTestCase.test_host_to_ips
- **ISSUE** #18659: (whiteinge) mod_aggregate not working for list-form configuration (refs: #41991)
- **PR** #41991: (Da-Juan) Accept a list for state_aggregate global setting @ *2017-06-29 00:58:59 UTC*
  - a7f38929cb Merge pull request #41991 from Da-Juan/fix-state_aggregate-list
  - c9075b8f84 Accept a list for state_aggregate setting
- **PR** #41993: (UtahDave) change out salt support link to SaltConf link @ *2017-06-29 00:55:20 UTC*
  - 7424f879a3 Merge pull request #41993 from UtahDave/2016.11local
  - bff050ad52 change out salt support link to SaltConf link
- **PR** #41987: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-06-28 20:19:11 UTC*
  - 3b9ccf09d7 Merge pull request #41987 from rallytime/merge-2016.11
  - 48867c4a82 Merge branch '2016.3' into '2016.11'
    - c589eae03f Merge pull request #41981 from Ch3LL/11.6_3

        - 2516ae1349 [2016.3] Bump latest release version to 2016.11.6

- **PR** #41985: (rallytime) Back-port #41780 to 2016.11 @ *2017-06-28 20:18:57 UTC*
    - **PR** #41780: (ferringb) Fix salt.util.render_jinja_tmpl usage for when not used in an environmnet (refs: #41985)
    - 768339d734 Merge pull request #41985 from rallytime/bp-41780
    - 8f8d3a473a Fix salt.util.render_jinja_tmpl usage for when not used in an environment.
- **ISSUE** #34963: (craigafinch) Incorrect behavior or documentation for comments in salt.states.pkgrepo.managed (refs: #41820)
- **PR** #41986: (rallytime) Back-port #41820 to 2016.11 @ *2017-06-28 20:18:43 UTC*
    - **PR** #41820: (nhavens) Fix yum repo file comments to work as documented in pkgrepo.managed (refs: #41986)
    - bd9090c0bf Merge pull request #41986 from rallytime/bp-41820
    - 72320e35b9 Fix yum repo file comments to work as documented in pkgrepo.managed
- **PR** #41973: (vutny) Fix Master/Minion scheduled jobs based on Cron expressions (refs: #42077) @ *2017-06-28 16:39:02 UTC*
    - a31da52635 Merge pull request #41973 from vutny/fix-croniter-scheduled-jobs
    - 148788e652 Fix Master/Minion scheduled jobs based on Cron expressions
- **PR** #41980: (Ch3LL) [2016.11] Bump latest release version to 2016.11.6 @ *2017-06-28 15:35:11 UTC*
    - 689ff93349 Merge pull request #41980 from Ch3LL/11.6_11
    - fe4f5711d5 [2016.11] Bump latest release version to 2016.11.6
- **PR** #41961: (cachedout) Allow docs to be built under Python 3 @ *2017-06-27 21:11:54 UTC*
    - 82b1eb28ab Merge pull request #41961 from cachedout/py3_doc
    - 7aacddf6ef Allow docs to be built under Python 3
- **PR** #41948: (davidjb) Fix Composer state's *name* docs; formatting @ *2017-06-27 17:51:29 UTC*
    - **PR** #41933: (davidjb) Fix Composer state's *name* docs and improve formatting (refs: #41948)
    - f0eb51df17 Merge pull request #41948 from davidjb/patch-9
    - 0e4b3d9a42 Fix Composer state's *name* docs; formatting
- **PR** #41914: (vutny) archive.extracted: fix hash sum verification for local archives @ *2017-06-26 17:59:27 UTC*
    - e28e10ded2 Merge pull request #41914 from vutny/fix-archive-extracted-local-file-hash
    - 54910fe55f archive.extracted: fix hash sum verification for local archives
- **PR** #41912: (Ch3LL) Allow pacman module to run on Manjaro @ *2017-06-26 15:35:20 UTC*
    - 76ad6ff064 Merge pull request #41912 from Ch3LL/fix_manjaro
    - e4dd72a3e7 Update os_name_map in core grains for new manjaro systems
    - aa7c839fc5 Allow pacman module to run on Manjaro
- **ISSUE** #38093: (DmitryKuzmenko) Make threads avoid blocking waiting while communicating using TCP transport. (refs: #41516)
- **PR** #41516: (kstreee) Implements MessageClientPool to avoid blocking waiting for zeromq and tcp communications. @ *2017-06-26 14:41:38 UTC*

  - **PR** #37878: (kstreee) Makes threads avoid blocking waiting while communicating using Zeromq. (refs: #41516)
  - ff67d47a2e Merge pull request #41516 from kstreee/fix-blocking-waiting-tcp-connection
  - df96969959 Removes redundant closing statements.
  - 94b9ea51eb Implements MessageClientPool to avoid blocking waiting for zeromq and tcp communications.
- **PR** #41888: (Ch3LL) Add additional commits to 2016.11.6 release notes @ *2017-06-22 16:19:00 UTC*
  - c90cb6798a Merge pull request #41888 from Ch3LL/change_release
  - 4e1239d980 Add additional commits to 2016.11.6 release notes
- **PR** #41882: (Ch3LL) Add pycryptodome to crypt_test @ *2017-06-21 19:51:10 UTC*
  - 4a326444fe Merge pull request #41882 from Ch3LL/fix_crypt_test
  - 6f70dbd0e1 Add pycryptodome to crypt_test
- **PR** #41877: (Ch3LL) Fix netstat and routes test @ *2017-06-21 16:16:58 UTC*
  - 13df29ed9b Merge pull request #41877 from Ch3LL/fix_netstat_test
  - d2076a6c93 Patch salt.utils.which for test_route test
  - 51f7e107dc Patch salt.utils.which for test_netstat test
- **ISSUE** #41367: (lubyou) certutil.add_store does not work on non english windows versions or on Windows 10 (localised or English) (refs: #41566)
- **PR** #41566: (morganwillcock) win_certutil: workaround for reading serial numbers with non-English languages @ *2017-06-21 15:40:29 UTC*
  - 66f8c83c93 Merge pull request #41566 from morganwillcock/certutil
  - c337d52d0c Fix test data for test_get_serial, and a typo
  - 7f6961378e test and lint fixes
  - 8ee48432f4 Suppress output of crypt context and be more specifc with whitespace vs. serial
  - 61f817d172 Match serials based on output position (fix for non-English languages)
- **PR** #41679: (terminalmage) Prevent unnecessary duplicate pillar compilation @ *2017-06-21 15:32:42 UTC*
  - 4d0f5c433d Merge pull request #41679 from terminalmage/get-top-file-envs
  - a916e8da49 Improve normalization of saltenv/pillarenv usage for states
  - 02f293a19c Update state unit tests to reflect recent changes
  - b7e5c11165 Don't compile pillar data when getting top file envs
  - 8d6fdb7c9a Don't compile pillar twice for salt-call
  - d2abfbf4ed Add initial_pillar argument to salt.state
  - 70186de532 salt.pillar: rename the "pillar" argument to "pillar_override"
- **ISSUE** #39668: (mirceaulinic) Master scheduled job not recorded on the event bus (refs: #41658)
- **ISSUE** #12653: (pengyao) salt schedule doesn't return jobs result info to master (refs: #41853)
- **PR** #41853: (vutny) Fix master side scheduled jobs to return events @ *2017-06-20 22:06:29 UTC*
  - **PR** #41695: (xiaoanyunfei) fix max RecursionError, Ellipsis (refs: #41853)

  - **PR** #41658: (garethgreenaway) Fixes to the salt scheduler (refs: #41853)
  - 29b0acc3a2 Merge pull request #41853 from vutny/fix-master-schedule-event
  - e206c381c6 Fix master side scheduled jobs to return events

28.2.40 Salt 2016.11.9 Release Notes

Version 2016.11.9 is a bugfix release for *2016.11.0*.

Statistics

- Total Merges: **143**
- Total Issue References: **60**
- Total PR References: **167**
- Contributors: **54** (Ch3LL, UtahDave, VertigoRay, akissa, aogier, arthtux, austinpapp, basepi, benediktwerner, bobrik, brejoc, cachedout, cetanu, corywright, creideiki, cro, cruscio, damon-atkins, dayid, defanator, dereckson, dijit, doesitblend, garethgreenaway, gtmanfred, gurubert, gvengel, jfindlay, johnj, jubrad, junovitch, lomeroe, lordcirth, lorengordon, mattLLVW, meaksh, moio, msummers42, mtkennerly, nicholasmhughes, oeuftete, rallytime, rasathus, roaldnefs, rossengeorgiev, seanjnkns, senthilkumar-e, techhat, terminalmage, twangboy, vernondcole, vutny, whiteinge, whytewolf)

Windows Changes

`pkg` Execution Module‘

Significate changes (PR #43708 & #45390, damon-atkins) have been made to the pkg execution module. Users should test this release against their existing package sls definition files.

- *`pkg.list_available`* no longer defaults to refreshing the winrepo meta database.
- *`pkg.install`* without a `version` parameter no longer upgrades software if the software is already installed. Use `pkg.install version=latest` (or simply use a *`pkg.latest`* state to get the old behavior.
- *`pkg.list_pkgs`* now returns multiple versions if software installed more than once.
- *`pkg.list_pkgs`* now returns `Not Found` when the version is not found instead of `(value not set)` which matches the contents of the sls definitions.
- *`pkg.remove`* will wait up to 3 seconds (normally about a second) to detect changes in the registry after removing software, improving reporting of version changes.
- *`pkg.remove`* can remove `latest` software, if `latest` is defined in sls definition.
- Documentation was update for the execution module to match the style in new versions, some corrections as well.
- All install/remove commands are prefix with cmd.exe shell and cmdmod is called with a command line string instead of a list. Some sls files in saltstack/salt-winrepo-ng expected the commands to be prefixed with cmd.exe (i.e. the use of &).
- Some execution module functions results, now behave more like their Unix/Linux versions.

`cmd` Execution Module

Due to a difference in how Python's `subprocess.Popen()` spawns processes on Windows, passing the command as a list of arguments can result in problems. This is because Windows' *CreateProcess* requires the command to be passed as a single string. Therefore, `subprocess` will attempt to re-assemble the list of arguments into as string. Some escaped characters and quotes can cause the resulting string to be incorrectly-assembled, resulting in a failure to execute the command.

Salt now deals with these cases by joining the list of arguments correctly and ensuring that the command is passed to `subprocess.Popen()` as a string.

Changelog for v2016.11.8..v2016.11.9

Generated at: 2018-05-27 20:28:05 UTC

- **PR** #45638: (twangboy) Win fix shell info @ *2018-01-23 22:38:22 UTC*
  - 10812969f0 Merge pull request #45638 from twangboy/win_fix_shell_info
  - 872da3ffba Only convert text types in the list_values function
  - 0e41535cdb Fix reg.py to only convert text types to unicode
  - 3579534ea5 Fix issue with detecting powershell
- **PR** #45564: (Ch3LL) Add PR changes to 2016.11.9 Release Notes @ *2018-01-19 21:36:05 UTC*
  - 2d1dd1186e Merge pull request #45564 from Ch3LL/r-notes-2016
  - 325f4cbcda Add PR changes to 2016.11.9 Release Notes
- **PR** #45563: (Ch3LL) Update man pages for 2016.11.9 @ *2018-01-19 21:19:00 UTC*
  - 28e4398150 Merge pull request #45563 from Ch3LL/man_2016
  - 529bc0c680 update release number for salt-call man page 2016.11.9
  - 11b7222148 Update man pages for 2016.11.9
- **PR** #45532: (gtmanfred) fix mock for opensuse @ *2018-01-18 22:48:30 UTC*
  - 654df0f526 Merge pull request #45532 from gtmanfred/2016.11.9
  - 6c26025664 fix mock for opensuse
- **PR** #45518: (gtmanfred) fix last 2016.11.9 failing tests @ *2018-01-18 12:03:50 UTC*
  - 571c33aa39 Merge pull request #45518 from gtmanfred/2016.11.9
  - 5455d2dee6 fix centos 6 pip test
  - 40255194b0 fix fedora pkg test
- **ISSUE** #45394: (dmurphy18) git.latest fails when "depth" is used with a non-default branch (refs: #45399)
- **PR** #45443: (rallytime) Back-port #45399 to 2016.11.9 @ *2018-01-17 14:53:58 UTC*
  - **PR** #45399: (terminalmage) Fix git.latest failure when rev is not the default branch (refs: #45443)
  - 4e0a0eec1f Merge pull request #45443 from rallytime/bp-45399-2016.11.9
  - 919e92c911 Fix git.latest failure when rev is not the default branch
- **ISSUE** #45432: (TheBigBear) winrepo-ng fault pkg.refresh_db doesn't work - it processes ANY stray .git meta-data *.sls files present on minion (refs: #45493)

- **PR** #45493: (damon-atkins) win_pkg: pkg.refresh_db report an issue if a sls pkg definition does not contain a dict instead of aborting @ *2018-01-17 14:52:03 UTC*
  - ebd4db66b8 Merge pull request #45493 from damon-atkins/2016.11_fix_sls_defintion_wrong_type
  - af108440df win_pkg lint space after ,
  - c6e922a236 win_pkg lint issues
  - f4627d7a80 fix quote i.e. change ‘ to ’
  - 6938a4c099 pkg.refresh_db report an issue if a sls pkg definition id not a dict instead of aborting.
- **PR** #45446: (rallytime) Back-port #45390 to 2016.11.9 @ *2018-01-16 20:08:38 UTC*
  - **PR** #45390: (damon-atkins) win_pkg: fix pkg.remove, pkg.list_pkgs (refs: #45446)
  - 7322efba92 Merge pull request #45446 from rallytime/bp-45390
  - 69f045ea24 lint too-many-blank-lines
  - 10a7501ede Update release notes
  - 6f2affe01c fix pkg.remove, pkg.list_pkgs
- **PR** #45424: (twangboy) Fix some issues with reg.py @ *2018-01-13 19:34:47 UTC*
  - b0ece9f4d4 Merge pull request #45424 from twangboy/win_reg
  - 30f06205f7 Fix some issues with reg.py
- **PR** #45327: (lomeroe) Backport #44861 to 2016.11 @ *2018-01-08 21:10:41 UTC*
  - **PR** #44861: (twangboy) Fix win_lgpo for unknown values (refs: #45327)
  - 0959ae4ea3 Merge pull request #45327 from lomeroe/bp-44861_2016.11
  - 784139f734 Check for values other than 0 or 1
- **PR** #45268: (damon-atkins) Fix pkg.install packagename version=latest i.e. if on an old version is installed @ *2018-01-08 17:34:15 UTC*
  - a6db5f95f0 Merge pull request #45268 from damon-atkins/2016.11_win_pkg_pkg_install_latest
  - 325a9f0f66 Update 2016.11.9.rst
  - 4da9200b9c Update 2016.11.9.rst
  - 126aee36ac Update 2016.11.9.rst
  - 1c01967943 Update 2016.11.9.rst
  - a0d89882b8 Fix pkg.install packagename version=latest i.e. if on an old version upgrade to the latest
- **PR** #45256: (rallytime) Back-port #45034 to 2016.11 @ *2018-01-04 14:25:42 UTC*
  - **PR** #45034: (brejoc) Fix for pidfile removal logging (refs: #45256)
  - 1c5e905b61 Merge pull request #45256 from rallytime/bp-45034
  - 68f971b38f Apply test fixes from #45034 to parsers_test.py
  - 9454236694 Fix for pidfile removal logging
- **ISSUE** saltstack/salt-jenkins#598: (rallytime) [oxygen] CentOS 7 is failing ~ 20 tests in the integration.ssh.test_state.SSHStateTest (refs: #45209)
- **PR** #45235: (rallytime) Back-port #45209 to 2016.11 @ *2018-01-02 20:20:15 UTC*
  - **PR** #45209: (gtmanfred) enable UsePAM for ssh tests (refs: #45235)

  - b75f50afe3 Merge pull request #45235 from rallytime/bp-45209
  - 2d0a9bbf7e enable UsePAM for ssh tests
- **PR** #44965: (gtmanfred) check if VALUE is a string_type @ *2018-01-02 16:42:39 UTC*
  - 3ab962b01a Merge pull request #44965 from gtmanfred/2016.11
  - a5d8a6340e check if VALUE is a string_type
- **ISSUE** #27160: (martinadolfi) salt.states.mount persistence error using spaces in route (refs: #45232)
- **PR** #45232: (rasathus) Backport #27160 to 2016.11 @ *2018-01-02 15:48:22 UTC*
  - 40fb30f63f Merge pull request #45232 from rasathus/2016.11
  - 7a2bd8f49b Merge branch '2016.11' into 2016.11
- **ISSUE** #44516: (doesitblend) Windows PY3 Minion Returns UTF16 UnicodeError (refs: #44944, #45161)
- **PR** #45161: (lomeroe) Backport #44944 to 2016.11 @ *2017-12-30 13:19:35 UTC*
  - **PR** #44944: (lomeroe) win_lgpo registry.pol encoding updates (refs: #45161)
  - 707ef55175 Merge pull request #45161 from lomeroe/bp-44944_2016.11
  - 0a4c6b5a83 remove references to six.unichr
  - f3196d795d lint fixes for static regexes
  - 11b637d108 lint fixes
  - c14d6282ad do not decode registry.pol file wholesale, but instead decode individual elements of the file
- **ISSUE** #45188: (jak3kaj) salt state status.process always returns false (refs: #45199)
- **PR** #45199: (gtmanfred) status.pid returns pid ids not process names @ *2017-12-28 19:06:11 UTC*
  - 6f52034e08 Merge pull request #45199 from gtmanfred/status
  - fb07f9ea7d status.pid returns pid ids not process names
- **ISSUE** #44728: (casselt) Nodegroups can not be defined by glob with ? or seq (refs: #45118)
- **PR** #45118: (garethgreenaway) [2016.11] Fix to allow nodegroups to include sequences @ *2017-12-27 18:49:10 UTC*
  - d3381e27d0 Merge pull request #45118 from garethgreenaway/44728_nodegroups_seq
  - 0ff811de70 Swapping import to be the old path for 2016.11
  - b3e2f388f5 Fix to allow nodegroups to include sequences
- **PR** #45127: (twangboy) Fix issue with 1641 return code @ *2017-12-22 15:18:28 UTC*
  - f969aca3a3 Merge pull request #45127 from twangboy/win_fix_pkg
  - 14639739f2 Fix issue with 1641 return code
- **PR** #45137: (twangboy) Catch correct error type in list_keys and list_values @ *2017-12-22 14:45:22 UTC*
  - dc357b39f0 Merge pull request #45137 from twangboy/win_fix_reg_tests
  - b6f4ef8d73 Catch correct error type in list_keys and list_values
- **PR** #45130: (rallytime) Resolve groups for salt api @ *2017-12-21 20:38:32 UTC*
  - 0aa1662731 Merge pull request #45130 from rallytime/api-groups
  - 2dcc8df845 Resolve groups for salt api

- **PR** #45114: (twangboy) Move pam library load to try/except block @ *2017-12-21 14:37:17 UTC*
    - 7dc3cc4641 Merge pull request #45114 from twangboy/win_fix_pam
    - cf5eae1f77 Move pam library load to try/except block
- **ISSUE** #45049: (vernondcole) salt cloud module documentation is missing from the index. (refs: #45070)
- **PR** #45100: (rallytime) Back-port #45070 to 2016.11 @ *2017-12-20 14:55:01 UTC*
    - **PR** #45070: (vernondcole) insert clouds modules in index (refs: #45100)
    - 7e128e8f15 Merge pull request #45100 from rallytime/bp-45070
    - 0bdb46dab9 add clouds modules to index
- **PR** #45098: (rallytime) Back-port #45092 to 2016.11 @ *2017-12-20 14:40:51 UTC*
    - **PR** #45092: (terminalmage) Fix integration.states.test_pip.PipStateTest.test_pip_installed_weird_install (refs: #45098)
    - bdf93f339d Merge pull request #45098 from rallytime/bp-45092
    - 80b6bd6813 Fix integration.states.test_pip.PipStateTest.test_pip_installed_weird_install
- **ISSUE** #41044: (pirxthepilot) user.present 'date' parameter is not applying (refs: #44078)
- **PR** #44078: (rossengeorgiev) user.present: allow date param to be 0 @ *2017-12-19 15:59:29 UTC*
    - 324b7d4058 Merge pull request #44078 from rossengeorgiev/fix-41044
    - a81a6fe23c fix #41044; allow for date param to be 0
- **PR** #44970: (rallytime) Update bootstrap script to latest release: 2017.12.13 @ *2017-12-19 15:49:05 UTC*
    - 48a59761df Merge pull request #44970 from rallytime/update-bootstrap-script
    - b2c8057427 Update bootstrap script to latest release: 2017.12.13
- **ISSUE** #45036: (dijit) Quiet installation of packaged minions fails due to redistributable not being quietly installed [py3] [Windows] (refs: #45040)
- **PR** #45069: (rallytime) Back-port #45040 to 2016.11 @ *2017-12-19 14:25:57 UTC*
    - **PR** #45040: (dijit) Installation Fails on headless machines. (refs: #45069)
    - 637fdaed58 Merge pull request #45069 from rallytime/bp-45040
    - aa438e1605 Installation Fails on headless machines.
        - de53c45c29 Backport #27160 to 2016.11
- **ISSUE** #41286: (arthtux) boto_vpc.accept_vpc_peering_connection wait a object (refs: #41305)
- **PR** #44969: (rallytime) Back-port #41305 to 2016.11 @ *2017-12-15 17:22:18 UTC*
    - **PR** #41305: (arthtux) correct accept_vpc_peering_connection (refs: #44969)
    - 4d6d640381 Merge pull request #44969 from rallytime/bp-41305
    - 5c4bee43dc correct accept_vpc_peering_connection
- **PR** #45031: (terminalmage) Fix invalid exception class in mysql returner @ *2017-12-15 15:00:15 UTC*
    - 10de468f13 Merge pull request #45031 from terminalmage/fix-mysql-returner
    - f3bd12c27c Fix invalid exception class in mysql returner
- **ISSUE** #44820: (msteed) Custom returner breaks manage runner (refs: #44958)
- **PR** #44972: (terminalmage) Backport #44958 to 2016.11 branch @ *2017-12-14 16:56:02 UTC*

  - **PR** #44958: (terminalmage) Fix a race condition in manage runner (refs: #44972)
  - 9a7406207f Merge pull request #44972 from terminalmage/bp-44958
  - a416bf0112 No need to manually do connect_pub, use listen=True in run_job
  - 3ec004bd2e Fix a race condition in manage runner
- **ISSUE** #44378: (llua) minion: infinite loop during start when schedule key is null (refs: #44385)
- **PR** #44385: (gtmanfred) schedule should be a dict in opts @ *2017-12-12 20:44:02 UTC*
  - 1032ca3290 Merge pull request #44385 from gtmanfred/schedule
  - 9e15c38da2 add comma
  - 855d933cb7 schedule should be a dict
- **ISSUE** #44734: (cruscio) Documentation inconsistency for minion ping_interval timing (refs: #44770)
- **PR** #44770: (cruscio) Fix minion ping_interval documentation @ *2017-12-11 19:50:19 UTC*
  - 68d901b12c Merge pull request #44770 from cruscio/2016.11
  - e2682bf441 Fix minion ping_interval documentation
- **ISSUE** #44292: (andrew-regan) grains['virtual_subtype'] assignment for Docker broken on Mac (refs: #44335)
- **PR** #44335: (gtmanfred) add docker-ce to docker subtype grains check @ *2017-12-10 17:17:49 UTC*
  - d4ab55ec47 Merge pull request #44335 from gtmanfred/2016.11
  - 3f1268d67f fix patching for python 2.6
  - 1d0bd5bb32 Merge branch '2016.11' into 2016.11
  - f02b02032d Merge pull request #4 from terminalmage/pr-44335
    - b4eb1527a6 Add test for PR 44335
  - a30af3252e add docker-ce to docker subtype grains check
- **ISSUE** #44530: (roaldnefs) Identifier not working in salt.states.cron when special is used (refs: #44579)
- **PR** #44579: (roaldnefs) Fix bug in cron module and state - Fixes #44530 @ *2017-12-07 20:18:27 UTC*
  - bb1f8dceaf Merge pull request #44579 from roaldnefs/fix-cron-identifier
  - df73a4c051 Merge branch '2016.11' into fix-cron-identifier
- **PR** #44852: (damon-atkins) win_pkg fix spelling typos and minion option 2016.11 @ *2017-12-06 16:49:17 UTC*
  - af0131fa1f Merge pull request #44852 from damon-atkins/2016.11_win_pkg_typo_n_fix
  - 0e7c19084f Lint: Remove extra whitespace
  - 7c7e21f94d Fix spelling typo, and fix backwards campatible minion option for repo location
- **ISSUE** #44365: (icycle77) file.managed appears to ignore source_hash check (refs: #44794)
- **PR** #44794: (terminalmage) Fix regression in file.managed when source_hash used with local file @ *2017-12-04 14:23:29 UTC*
  - 88c0d66b4e Merge pull request #44794 from terminalmage/issue44365
  - 3b8b6f25e6 Remove debugging line
  - 153bf45b03 Fix regression in file.managed when source_hash used with local file
- **ISSUE** #35777: (rallytime) Properly deprecate template context data in Fluorine (refs: #44738)

- **ISSUE** #35523: (rallytime) Come up with a reasonable alternative for lxc.edited_conf (refs: #44738)
- **PR** #44738: (rallytime) Bump some deprecation warnings from Oxygen to Fluorine @ *2017-12-01 23:10:08 UTC*
  - c8bb9dfbbb Merge pull request #44738 from rallytime/bump-oxygen-warnings
  - ead3c569e1 Bump deprecation warnings from Oxygen to Fluorine
- **ISSUE** #44730: (msciciel) State network.routes could not add route without gateway on centos7 (refs: #44741)
- **PR** #44741: (gtmanfred) if gateway is not specified use iface @ *2017-12-01 23:09:03 UTC*
  - 88e3aab00d Merge pull request #44741 from gtmanfred/rhip
  - 439dc8dce6 if gateway is not specified use iface
- **ISSUE** #31405: (SEJeff) Salt leaves tmp file when file.managed dest file is immutable (refs: #44699)
- **PR** #44699: (jfindlay) utils/files.py remove temp file upon move failure @ *2017-12-01 15:03:54 UTC*
  - 97e0cf569c Merge pull request #44699 from jfindlay/attr_file
  - 9e5a40ea7c Merge branch '2016.11' into attr_file
  - 5c34607f6c utils/files remove temp file upon move failure
- **ISSUE** #44556: (doesitblend) –static option doesn't return highstate output (refs: #44714)
- **PR** #44714: (rallytime) Allow –static option to display state runs with highstate output @ *2017-12-01 14:31:19 UTC*
  - 7434e0afdf Merge pull request #44714 from rallytime/fix-44556
  - 1bbe1abeb2 Allow –static option to display state runs with highstate output
- **PR** #44517: (whytewolf) Publish port doc missing @ *2017-11-28 21:50:19 UTC*
  - 998d714ee7 Merge pull request #44517 from whytewolf/publish_port_doc_missing
  - 4b5855283a missed one place where i didnt chanbge master_port from my copy to publish_port
  - e4610baea5 update doc to have publish port
- **PR** #41279: (Ch3LL) Add fqdn and dns core grain tests @ *2017-11-27 21:28:10 UTC*
  - 6169b52749 Merge pull request #41279 from Ch3LL/add_grain_tests
  - 1b64f15692 Merge branch '2016.11' into add_grain_tests
  - 095f1b7d7a Merge branch '2016.11' into add_grain_tests
  - 9ea4db4224 mock socket.getaddrinfo
  - 78a07e30f4 add more fqdn tests and remove some of the mocking
  - 5dbf4144ce add ipv6 in opts
  - eabc1b4f9c Add fqdn and dns core grain tests
    - 3ec4329307 Merge branch '2016.11' into fix-cron-identifier
- **ISSUE** #44544: (creideiki) pgjsonb returner sets wrong timezone on timestamps in database when using Python 2 (refs: #44563)
- **PR** #44563: (creideiki) Send Unix timestamps to database in pgjsonb returner @ *2017-11-21 17:44:32 UTC*
  - dc6de050a9 Merge pull request #44563 from creideiki/pgjsonb-timestamps-44544
  - 231e412ca4 Merge branch '2016.11' into pgjsonb-timestamps-44544

- **ISSUE** #44601: (rallytime) CherryPy 12.0 removed support for "engine.timeout_monitor.on" config option (refs: #44602)
- **PR** #44602: (rallytime) Handle timeout_monitor attribute error for new versions of CherryPy @ *2017-11-20 21:38:40 UTC*
  - 4369df020b Merge pull request #44602 from rallytime/fix-44601
  - ff303fd060 Handle timeout_monitor/TimeoutError issues for new versions of CherryPy
- **PR** #44604: (lorengordon) Documents the exclude argument in state execution module @ *2017-11-20 18:19:18 UTC*
  - 4a4756fc37 Merge pull request #44604 from lorengordon/doc-exclude
  - c4a6c40eb3 Documents the exclude argument in state execution module
  - 15c445e6b9 Send Unix timestamps to database in pgjsonb
    - 99fa05a456 Fix for bug in cron state
    - 97328faeac Fix for bug in cron module
- **PR** #44434: (whytewolf) add a note that describes grain rebuilding on restart and refresh @ *2017-11-14 11:21:54 UTC*
  - 91d46d4cfc Merge pull request #44434 from whytewolf/1837
  - d148e39dda change from md to rst for code reference
  - 955e305bda fix bad english, as requested by cachedout
  - 7256fcc1c9 update note to take into account grains_cache
  - 7a2981585e Merge branch '2016.11' into 1837
  - aca0405b26 add a note that describes grain rebuilding on restart and refresh
- **ISSUE** #41474: (dmaziuk) state.file.* line endings (refs: #44321)
- **PR** #44321: (gvengel) Fix file.line diff formatting. @ *2017-11-13 19:36:39 UTC*
  - a3bd99317f Merge pull request #44321 from gvengel/fix-file-line-diff-output
  - 69a50204a6 Add newline for lint.
  - ef7b6bbb81 Fixed issue with file.line on Windows running Python 2.
  - 8f89c99fa5 Fix FileModuleTest setUp and tearDown to work on Windows.
  - 3ac5391f5f Namespace missing functions for file.line on Windows.
  - b2b8f075b9 Fixed test to work on Windows.
  - 5a5a2dd026 Added integration test for issue #41474
  - 24d7315f1a Fix file.line diff formatting.
- **ISSUE** #43417: (damon-atkins) win_pkg: pkg.install and pkg.remove general issues (refs: #43708)
- **PR** #43708: (damon-atkins) Merge Ready : Backport develop win_pkg to 2016.11 with additional bug fixes @ *2017-11-13 19:33:41 UTC*
  - 9ca563718d Merge pull request #43708 from damon-atkins/2016.11_43417_Backport_and_Fixes
  - 04d03ea6b8 Updated comment
  - 1dd565e585 Merge remote branch 'upstream/2016.11' into 2016.11_43417_Backport_and_Fixes
  - dd48ba2616 Merge remote branch 'upstream/2016.11' into 2016.11_43417_Backport_and_Fixes

  - a0d08598bf dco fix
  - 9467899fc6 Merge remote branch 'upstream/2016.11' into 2016.11_43417_Backport_and_Fixes
  - 6dc180fd0e doco fixes
  - 2496a42ea4 lint fix
  - 2c937fbe19 Merge remote branch 'upstream/2016.11' into 2016.11_43417_Backport_and_Fixes
  - c9c8c48a4d all remove/install commands are passed to cmd.exe /s /c and commands are passed as strings to cmdmod
  - 350244bd93 typo in comments and doc strings.
  - ec31f5a9bd 2017.11/develop version() was ignoring saltenv setting.
  - b314549a32 Backport of devlop to 2016.11 with additional bug fixes
- **ISSUE** #44423: (mtkennerly) The win_path.exists state cannot prepend to the very start of the PATH (refs: #44424)
- **PR** #44477: (rallytime) Back-port #44424 to 2016.11 @ *2017-11-13 17:33:29 UTC*
  - **PR** #44424: (mtkennerly) Fix #44423: Handle index=None and index=0 distinctly in the win_path.exists state (refs: #44477)
  - 68ea22188e Merge pull request #44477 from rallytime/bp-44424
  - 4a9f8dcc96 Fix #44423: Handle index=None and index=0 distinctly
- **ISSUE** #44034: (seanjnkns) salt-call pillar overrides broken in 2016.11.8 and 2017.7.2 (refs: #44483)
- **PR** #44483: (terminalmage) salt-call: account for instances where __pillar__ is empty @ *2017-11-13 17:30:36 UTC*
  - 2c89050a24 Merge pull request #44483 from terminalmage/issue44034
  - a9db8becea salt-call: account for instances where __pillar__ is empty
- **PR** #44489: (whytewolf) update log-granular-levels to describe what they are filtering on @ *2017-11-13 17:27:37 UTC*
  - b5c2028680 Merge pull request #44489 from whytewolf/1956_log-granular-levels
  - 9cdeb4e903 update log-granular-levels to describe what they are filtering on
- **PR** #44193: (twangboy) Fix reg.py for use with LGPO module @ *2017-11-10 19:01:17 UTC*
  - ea07f9c54c Merge pull request #44193 from twangboy/win_fix_reg
  - 44d6d9f46d Remove unused import (lint)
  - f7502436bd Fix various issues
  - 221e6e3b91 make salt.utils.to_unicode return none when passed none
  - ce41acc788 Fix many issues with reg.py
  - 4a19df1f7f Use six.text_type instead of str
  - 1b12acd303 Check type before casting
  - 03fa37b445 Cast vdata to it's proper type
- **PR** #43863: (nicholasmhughes) Atomicfile only copies mode and not user/group perms @ *2017-11-10 18:47:55 UTC*
  - ed8da2450b Merge pull request #43863 from nicholasmhughes/fix-atomicfile-permission-copy

  - ea852ec5d3 remove index use with stat module attributes
  - dbeeb0e917 fixes #38452 atomicfile only copies mode and not user/group perms
- **ISSUE** #39901: (seanjnkns) network.managed ipaddrs ignored (refs: #44260)
- **PR** #44260: (seanjnkns) Fixes #39901 for RH/CentOS 7 @ *2017-11-07 23:14:59 UTC*
  - a66cd67d15 Merge pull request #44260 from seanjnkns/issue-39901
  - ed8cccf457 #39901: Fix pylint
  - 43c81dfdee #39901: Add unit tests
  - 613d500876 Merge branch '2016.11' into issue-39901
  - b97e8046ca Utilize salt.utils.validate.net.* and _raise_error_iface
  - 6818f3631d Fixes #39901 for RH/CentOS 7
- **PR** #44383: (gtmanfred) switch salt-jenkins over to saltstack for kitchen-salt tests @ *2017-11-03 19:56:48 UTC*
  - 5e289f42ba Merge pull request #44383 from gtmanfred/2016kitchen
  - b65f4ea4ea switch salt-jenkins over to saltstack
- **PR** #44173: (twangboy) Use google style docstrings in win_system.py @ *2017-10-31 17:56:34 UTC*
  - cab54e34b5 Merge pull request #44173 from twangboy/win_system_docs
  - 8e111b413d Fix some of the wording and grammer errors
  - a12bc5ae41 Use google style docstrings
- **PR** #44304: (jfindlay) states.cron identifier defaults to name @ *2017-10-31 16:39:47 UTC*
  - 7aaea1d179 Merge pull request #44304 from jfindlay/cron_id
  - cc038c5bec states.cron identifier defaults to name
- **ISSUE** #44313: (rossengeorgiev) salt-ssh: –user option missing from the cli documentation (refs: #44322)
- **PR** #44322: (rossengeorgiev) updated CLI docs for salt-ssh @ *2017-10-30 21:39:23 UTC*
  - e4dbbde734 Merge pull request #44322 from rossengeorgiev/saltssh-docs-update
  - b18f2e5a6d fix program name and description for –static
  - 5b10918f02 updated CLI docs for salt-ssh
- **PR** #44345: (gtmanfred) remove binding from erb template rendering @ *2017-10-30 20:57:43 UTC*
  - 4e6f09e3eb Merge pull request #44345 from gtmanfred/2016kitchen
  - 79b8b2d0bf remove binding
- **PR** #44342: (gtmanfred) render template files platforms.yml and driver.yml @ *2017-10-30 20:04:00 UTC*
  - 209847c8c2 Merge pull request #44342 from gtmanfred/2016kitchen
  - c50508f0b7 render template files platforms.yml and driver.yml
- **ISSUE** #44336: (corywright) Docs for archive.tar should not use leading dash for tar options (refs: #44339)
- **PR** #44339: (corywright) Remove leading dash from options in archive.tar docs (2016.11) @ *2017-10-30 19:00:34 UTC*
  - 1be65224cb Merge pull request #44339 from corywright/issue-44336-fix-archive-tar-docs-2016-11
  - 9c1c35a59f Remove leading dash (-) from options in archive.tar documentation

- **ISSUE** #44272: (gurubert) [patch] win_service.stop() fails (refs: #44295)
- **PR** #44295: (gurubert) fixes issue #44272 @ *2017-10-27 14:28:57 UTC*
  - bebc33daf5 Merge pull request #44295 from HeinleinSupport/issue44272
  - f972715a45 fixes issue #44272
- **PR** #44286: (gtmanfred) use our git repo for kitchen-salt @ *2017-10-25 19:27:32 UTC*
  - e7ca9f8407 Merge pull request #44286 from gtmanfred/2016.11
  - 193e715e37 use our git repo for kitchen-salt
- **PR** #44259: (gtmanfred) begin switching in kitchen-salt for running the test suite @ *2017-10-25 13:30:35 UTC*
  - 8a1ea165af Merge pull request #44259 from gtmanfred/2016.11
  - 56a3ad8f68 fix pylint comments
  - 4add666db1 add comment to Gemfile and move copyartifacts
  - b4c8f7eb57 fix pylint
  - 392fd4f837 try newest salttesting
  - 79251287d0 add logging
  - 38963d5a82 use transport if not set in state_file
  - 10e309a64f which vagrant should go to stderr
  - 9307564de0 fix output columns
  - 2da22f87e1 test opennebula
  - 9f38f16905 add opennebula to Gemfile
  - 7465f9b27a add script for copying back artifacts
  - 255118cfd7 run tests with kitchen
- **PR** #44268: (twangboy) Fix typo @ *2017-10-25 13:01:35 UTC*
  - 9d6bc8509b Merge pull request #44268 from twangboy/win_fix_lgpo_typo
  - a6a4c10a77 Fix typo
- **PR** #44269: (terminalmage) Fix log message in salt.utils.gitfs @ *2017-10-25 13:00:58 UTC*
  - 0beb65a283 Merge pull request #44269 from terminalmage/fix-log-message
  - bc9cd65496 Fix log message in salt.utils.gitfs
- **ISSUE** #44155: (rhoths) file.directory with clean not triggering listener in test mode (refs: #44160)
- **PR** #44160: (gtmanfred) add changes to test return @ *2017-10-23 14:35:21 UTC*
  - 304dd2529d Merge pull request #44160 from gtmanfred/directory
  - a7d3d668f4 missed removing changes in the next test
  - ac0b5ec440 fix test
  - d3d00c3e62 add changes to test return
- **PR** #44205: (rallytime) Back-port #44177 to 2016.11 @ *2017-10-23 14:09:07 UTC*
  - **PR** #44177: (senthilkumar-e) Fixing default redis.host in documentation (refs: #44205)
  - e10395483d Merge pull request #44205 from rallytime/bp-44177

  - b9940f8521 Fixing default redis.host in documentation
- **ISSUE** #44140: (vtolstov) incorrect network interfaces settings with network.managed under debian jessie (refs: #44167)
- **PR** #44167: (garethgreenaway) Fixes to modules/debian_ip @ *2017-10-20 14:25:39 UTC*
  - 09ddfd0c08 Merge pull request #44167 from garethgreenaway/44140_debian_ip_fixes
  - 5f7555846f When looping through the various pre, post, up and down commands put them into the interface dict using the right internet family variable.
- **PR** #43830: (rallytime) Back-port #43644 to 2016.11 @ *2017-10-19 22:57:51 UTC*
  - **PR** #43644: (defanator) Several fixes for RDS DB parameter group management (refs: #43830)
  - 9f9e936b52 Merge pull request #43830 from rallytime/bp-43644
  - 12845ae802 Several fixes for RDS DB parameter group management
- **ISSUE** #43936: (oeuftete) manage.present still reports *lost* minion (refs: #43994)
- **ISSUE** #38367: (tyeapple) logic error in connected_ids function of salt/utils/minions.py when using include_localhost=True (refs: #43994)
- **PR** #43994: (oeuftete) Fix manage.present to show lost minions @ *2017-10-19 22:27:59 UTC*
  - 07db6a3d8b Merge pull request #43994 from oeuftete/fix-manage-runner-presence
  - f3980d7d83 Fix manage.present to show lost minions
- **ISSUE** #44150: (rossengeorgiev) version param in pkg.installed broken in 2016.11.8/2017.7.2 in EL6-7 (refs: #44188)
- **PR** #44188: (terminalmage) yumpkg: Check pkgname instead of name to see if it is a kernel pkg @ *2017-10-19 22:20:35 UTC*
  - a07537e258 Merge pull request #44188 from terminalmage/issue44150
  - 0692f442db yumpkg: Check pkgname instead of name to see if it is a kernel pkg
- **ISSUE** #43427: (tylerjones4508) Salt-Cloud There was a profile error: invalid literal for int() with base 10: (refs: #44089)
- **PR** #44158: (rallytime) Back-port #44089 to 2016.11 @ *2017-10-19 20:38:15 UTC*
  - **PR** #44089: (cetanu) Catch on empty Virtualbox network addr #43427 (refs: #44158)
  - 715edc0cea Merge pull request #44158 from rallytime/bp-44089
  - 534faf0b7a Catch on empty Virtualbox network addr #43427
- **ISSUE** #43307: (marek-knappe) Filesystem creation is failing on newly created LV (refs: #44029)
- **PR** #44131: (rallytime) Back-port #44029 to 2016.11 @ *2017-10-17 15:05:39 UTC*
  - **PR** #44029: (msummers42) addresses issue #43307, disk.format_ to disk.format (refs: #44131)
  - 0cd493b691 Merge pull request #44131 from rallytime/bp-44029
  - bebf301976 fixed test addressing issue #43307, disk.format_ to disk.format
  - b4ba7ae2fc addresses issue #43307, disk.format_ to disk.format
- **ISSUE** #44087: (mfussenegger) Using state.highstate with *terse=true* prevents useful error output (refs: #44093)
- **PR** #44093: (gtmanfred) don't filter if return is not a dict @ *2017-10-16 19:13:19 UTC*
  - 3a68e356f8 Merge pull request #44093 from gtmanfred/fix-44087

  - 5455c5053b fix pylint
  - f749cafa25 don't filter if return is not a dict
- **PR** #44122: (cachedout) Add note about GPG signing to PR template @ *2017-10-16 19:09:38 UTC*
  - c785d7a847 Merge pull request #44122 from cachedout/gpg_pr_template
  - e41e3d76be Typo fix
  - 37c7980880 Add note about GPG signing to PR template
- **PR** #44124: (rallytime) [2016.11] Merge forward from 2016.11.8 to 2016.11 @ *2017-10-16 19:07:14 UTC*
  - bf90ea1f51 Merge pull request #44124 from rallytime/merge-2016.11
  - 59861291c8 Merge branch '2016.11.8' into '2016.11'
    - 57623e2abe Merge pull request #44028 from rallytime/bp-44011
      - 89e084bda3 Do not allow IDs with null bytes in decoded payloads
      - 206ae23f15 Don't allow path separators in minion ID
- **PR** #44097: (gtmanfred) OpenNebula does not require the template_id to be specified @ *2017-10-16 18:36:17 UTC*
  - 13f3ffa83a Merge pull request #44097 from gtmanfred/openneb
  - c29655b2c2 Merge branch '2016.11' into openneb
  - bd2490b149 OpenNebula does not require the template_id to be specified
- **PR** #44110: (roaldnefs) Format fix code example local returner doc @ *2017-10-16 15:57:50 UTC*
  - ac3e4df964 Merge pull request #44110 from roaldnefs/fix-doc-local-returner
  - efd58f7594 Merge branch '2016.11' into fix-doc-local-returner
- **PR** #44092: (techhat) Made sure that unicoded data is sent to sha256() @ *2017-10-13 21:20:12 UTC*
  - c960ca32c2 Merge pull request #44092 from techhat/awsunicode
  - bbd9db4d00 One more encoding
  - 0e8b325667 Apparently __salt_system_encoding__ is a thing
  - 1e7211838d Use system encoding
  - 1af21bbe5e Made sure that unicoded data is sent to sha256()
- **ISSUE** #43581: (jcourington) cherrypy stats issue (refs: #44021)
- **PR** #44021: (whiteinge) Also catch cpstats AttributeError for bad CherryPy release ~5.6.0 @ *2017-10-12 18:11:41 UTC*
  - **PR** #42655: (whiteinge) Reenable cpstats for rest_cherrypy (refs: #44021)
  - **PR** #33806: (cachedout) Work around upstream cherrypy bug (refs: #42655)
  - d89c317d96 Merge pull request #44021 from whiteinge/cpstats-attribute-error
  - bf14e5f578 Also catch cpstats AttributeError for bad CherryPy release ~5.6.0
- **PR** #44025: (dayid) Typo correction of lover to lower @ *2017-10-11 17:31:45 UTC*
  - bbdabe242a Merge pull request #44025 from dayid/lover_typo
  - 385980c21a Merge branch '2016.11' of https://github.com/saltstack/salt into lover_typo
  - 266dc00a23 Typo correction of lover to lower

- **PR** #44030: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 @ *2017-10-11 13:01:42 UTC*
    - d8f3891a5e Merge pull request #44030 from rallytime/merge-2016.11
    - 53eaf0d75c Merge branch '2016.3' into '2016.11'
    - 64fd839377 Merge pull request #44010 from Ch3LL/2016.3.7_follow_up
        - 9a00302cd8 fix 2016.3.7 release notes merge conflict
        - 63da1214db Do not allow IDs with null bytes in decoded payloads
        - ee792581fc Don't allow path separators in minion ID
        - 8aab65c718 fix 2016.3.7 release notes merge conflict
    - bd73dcb02c Merge pull request #43977 from Ch3LL/3.8_sec
    - 5fb3f5f6b1 Add Security Notes to 2016.3.8 Release Notes
- **PR** #44011: (Ch3LL) Security Fixes for 2016.11.8 (refs: #44028) @ *2017-10-10 20:04:36 UTC*
    - 0dbf41e79e Merge pull request #44011 from Ch3LL/2016.11.7_follow_up
    - c0149101c0 Do not allow IDs with null bytes in decoded payloads
    - 19481423dd Don't allow path separators in minion ID
- **PR** #44023: (Ch3LL) Add 2016.11.9 Release Note File @ *2017-10-10 20:03:03 UTC*
    - d61300df20 Merge pull request #44023 from Ch3LL/11.9rn
    - 7f9015eb41 Add 2016.11.9 Release Note File
- **PR** #44019: (benediktwerner) Added missing docs to the tutorial index and fixed spelling mistake @ *2017-10-10 19:57:06 UTC*
    - 9ff53bf63a Merge pull request #44019 from benediktwerner/2016.11
    - bc53598027 Fixed spelling mistake in salt_bootstrap tutorial
    - 6c30344824 Added missing tutorial docs to the tutorial index
- **PR** #43955: (meaksh) Enable a new '–with-salt-version' parameter for the "setup.py" script @ *2017-10-10 17:36:52 UTC*
    - 364523f5f8 Merge pull request #43955 from meaksh/2016.11-fix-2291
    - a81b78381b Merge branch '2016.11' into 2016.11-fix-2291
    - 44bc91bb98 Enable '–with-salt-version' parameter for setup.py script
- **ISSUE** #43945: (bobrik) kmod.present doesn't work with compiled-in modules (refs: #43962)
- **PR** #43962: (bobrik) Report built-in modiles in kmod.available, fixes #43945 @ *2017-10-10 16:31:39 UTC*
    - fec714b91d Merge pull request #43962 from bobrik/kmod-built-in
    - 95ab901553 Report built-in modiles in kmod.available, fixes #43945
- **PR** #43960: (cro) Require that bindpw be non-empty when auth.ldap.anonymous is False @ *2017-10-09 23:09:02 UTC*
    - e434c39c4e Merge pull request #43960 from cro/ldap_nopw_bind2
    - 962a20cf4b Require that bindpw be non-empty if auth.ldap.anonymous=False
    - 9df3d91d8f Release notes blurb for change to bindpw requirements
- **PR** #43991: (Ch3LL) Add Security Notes to 2016.3.8 Release Notes @ *2017-10-09 22:00:25 UTC*

  - e9dfda2177 Merge pull request #43991 from Ch3LL/3.8_sec_2
  - 1977df8462 Add Security Notes to 2016.3.8 Release Notes
- **ISSUE** #42947: (rossengeorgiev) Zenoss state changes production state even when test=true (refs: #43968)
- **PR** #43968: (rossengeorgiev) fix zenoss state module not respecting test=true @ *2017-10-09 21:27:31 UTC*
  - 2346d2691e Merge pull request #43968 from rossengeorgiev/fix-zenoss-prod_state
  - e6d31c1ea6 fix zenoss state module not respecting test=true
- **PR** #43776: (Ch3LL) [2016.11] Bump latest and previous versions @ *2017-10-09 17:22:15 UTC*
  - 8d56a5ac45 Merge pull request #43776 from Ch3LL/2016.11.8_docs
  - f72bc00000 [2016.11] Bump latest and previous versions
- **PR** #43976: (Ch3LL) Add Security Notes to 2016.11.8 Release Notes @ *2017-10-09 17:20:54 UTC*
  - 21bf71c3f5 Merge pull request #43976 from Ch3LL/11.8_sec
  - f0c3184288 Add Security Notes to 2016.11.8 Release Notes
- **PR** #43973: (terminalmage) Fix grains.has_value when value is False @ *2017-10-09 14:59:20 UTC*
  - 1d5397ab5b Merge pull request #43973 from terminalmage/fix-grains.has_value
  - bf45ae6e6a Fix grains.has_value when value is False
- **PR** #43888: (rallytime) Back-port #43841 to 2016.11 @ *2017-10-05 20:09:58 UTC*
  - **PR** #43841: (austinpapp) add -n with netstat so we don't resolve IPs (refs: #43888)
  - 9ac3f2ea7b Merge pull request #43888 from rallytime/bp-43841
  - 87d676f08a add -n with netstat so we don't resolve
- **PR** #43916: (dereckson) Fix typo in salt-cloud scaleway documentation @ *2017-10-05 18:58:00 UTC*
  - f880ac4c08 Merge pull request #43916 from dereckson/fix-typo-cloud-scaleway
  - 15b8b8a9f4 Fix typo in salt-cloud scaleway documentation
- **PR** #43884: (UtahDave) Update SaltConf banner per Rhett's request @ *2017-10-04 13:08:30 UTC*
  - 2ab7549d48 Merge pull request #43884 from UtahDave/2016.11local
  - e3b2857285 Merge branch '2016.11' into 2016.11local
- **PR** #43869: (terminalmage) Only join cmd if it's not a string @ *2017-10-03 16:25:07 UTC*
  - 4b882d4272 Merge pull request #43869 from terminalmage/issue43522
  - fe28b0d4fb Only join cmd if it's not a string
  - 8c671fd0c1 Update SaltConf banner per Rhett's request
- **ISSUE** #43373: (rgcosma) use keyword breaks sls_id (refs: #43707)
- **PR** #43707: (terminalmage) Add missing support for use/use_in requisites to state.sls_id @ *2017-10-01 14:07:53 UTC*
  - a2161efda3 Merge pull request #43707 from terminalmage/issue43373
  - 3ebde1895f Merge branch '2016.11' into issue43373
  - e580ed4caa Merge branch '2016.11' into issue43373
  - 5b3be6e8af Fix failing unit test

  - f73764481b Add missing support for use/use_in requisites to state.sls_id
- **PR** #43807: (terminalmage) cmdmod: Don't list-ify string commands on Windows @ *2017-09-29 02:48:36 UTC*
  - 85b3aa332a Merge pull request #43807 from terminalmage/issue43522
  - d8708bf698 cmdmod: Don't list-ify string commands on Windows
- **PR** #43768: (vutny) Fix Pylint deprecated option warnings @ *2017-09-28 12:27:36 UTC*
  - ea8d273c2b Merge pull request #43768 from vutny/fix-pylint-deprecation-warnings
  - f8b3fa9da1 Merge branch '2016.11' into fix-pylint-deprecation-warnings
- **ISSUE** #40311: (cralston0) –hide-timeout used with –output json –static produces unparseable JSON (refs: #43772)
- **PR** #43772: (gtmanfred) dont print Minion not responding with quiet @ *2017-09-27 15:39:18 UTC*
  - 1a8cc60bb4 Merge pull request #43772 from gtmanfred/2016.11
  - 0194c60960 dont print Minion not responding with quiet
- **PR** #43747: (rallytime) Add GPG Verification section to Contributing Docs @ *2017-09-26 21:25:37 UTC*
  - 9dee896fb9 Merge pull request #43747 from rallytime/gpg-verification
  - 7a70de19f4 Merge branch '2016.11' into gpg-verification
- **ISSUE** #43729: (The-Loeki) Docker events engine broken on newer docker.py (refs: #43733)
- **PR** #43733: (terminalmage) Allow docker_events engine to work with newer docker-py @ *2017-09-26 16:47:40 UTC*
  - 1cc3ad1c8d Merge pull request #43733 from terminalmage/issue43729
  - 6e5c99bda0 Allow docker_events engine to work with newer docker-py
- **ISSUE** #42082: (stamak) [salt.utils.gitfs][CRITICAL] Invalid gitfs configuration parameter 'saltenv' in remote git+ssh://git@ourgitserver/ourgitrepo.git. (refs: #43458)
- **PR** #43458: (terminalmage) Fix missing PER_REMOTE_ONLY in cache.clear_git_lock runner @ *2017-09-26 14:39:01 UTC*
  - 5d38be4ff7 Merge pull request #43458 from terminalmage/issue42082
  - 5f90812b12 Fix missing PER_REMOTE_ONLY in cache.clear_git_lock runner
    - 23bb4a5dde Add GPG Verification section to Contributing Docs
- **ISSUE** #43650: (rallytime) Review contributing documentation and the merge-forward process (refs: #43727)
- **ISSUE** #42706: (blarghmatey) Parallel Cache Failure (refs: #43018)
- **PR** #43727: (rallytime) Revise "Contributing" docs: merge-forwards/release branches explained! @ *2017-09-26 12:43:16 UTC*
  - **PR** #43018: (jubrad) Update state.py (refs: #43727)
  - 023a563657 Merge pull request #43727 from rallytime/fix-43650
  - babad12d83 Revise "Contributing" docs: merge-forwards/release branches explained!
- **PR** #43648: (rallytime) Handle VPC/Subnet ID not found errors in boto_vpc module @ *2017-09-22 17:40:43 UTC*
  - f46c858f25 Merge pull request #43648 from rallytime/handle-boto-vpc-errors
  - 54842b5012 Handle VPC/Subnet ID not found errors in boto_vpc module
    - 651ed16ad3 Fix Pylint deprecated option warnings

- **PR** #43575: (akissa) Fix CSR not recreated if key changes @ *2017-09-21 17:52:01 UTC*
  - 9dba34aa06 Merge pull request #43575 from akissa/fix-csr-not-recreated-if-key-changes
  - b1b4dafd39 Fix CSR not recreated if key changes
- **ISSUE** #42165: (arount) top_file_merging_strategy: merge does not works (refs: #43415)
- **PR** #43672: (rallytime) Back-port #43415 to 2016.11 @ *2017-09-21 16:38:56 UTC*
  - **PR** #43415: (mattLLVW) Fix env_order in state.py (refs: #43672)
  - 1d4fa48209 Merge pull request #43672 from rallytime/bp-43415
  - 3fb42bc238 Fix env_order in state.py
- **PR** #43673: (rallytime) Back-port #43652 to 2016.11 @ *2017-09-21 16:37:36 UTC*
  - **PR** #43652: (VertigoRay) Salt Repo has Deb 9 and 8 (refs: #43673)
  - ff832ee607 Merge pull request #43673 from rallytime/bp-43652
  - d91c47c6f0 Salt Repo has Deb 9 and 8
- **PR** #43677: (terminalmage) Fix RST headers for runners (2016.11 branch) @ *2017-09-21 16:35:57 UTC*
  - 365cb9fba8 Merge pull request #43677 from terminalmage/runners-docs-2016.11
  - 2fd88e94fa Fix RST headers for runners (2016.11 branch)
- **PR** #43534: (twangboy) Fixes removal of double-quotes by shlex_split in winrepo for 2016.11 @ *2017-09-21 14:39:42 UTC*
  - be38239e5d Merge pull request #43534 from twangboy/win_fix_pkg.install_2016.11
  - 1546c1ca04 Add posix=False to call to salt.utils.shlex_split
  - **PR** #43663: (moio) multiprocessing minion option: documentation fixes (develop) (refs: #43661)
- **PR** #43661: (moio) multiprocessing minion option: documentation fixes (2016.11) @ *2017-09-21 13:02:27 UTC*
  - 0d3fd3d374 Merge pull request #43661 from moio/2016.11-multiprocessing-doc-fix
  - 625eabb83f multiprocessing minion option: documentation fixes
- **PR** #43646: (brejoc) Added tests for pid-file deletion in DaemonMixIn @ *2017-09-20 19:21:54 UTC*
  - 6b4516c025 Merge pull request #43646 from brejoc/2016.11.4-pidfile-tests
  - 96f39a420b Fixed linting
  - 08fba98735 Fixed several issues with the test
  - 3a089e450f Added tests for pid-file deletion in DaemonMixIn
- **PR** #43591: (rallytime) [2016.11] Merge forward from 2016.11.8 to 2016.11 @ *2017-09-19 16:18:34 UTC*
  - cfb1625741 Merge pull request #43591 from rallytime/merge-2016.11
  - 57b9d642c2 Merge branch '2016.11.8' into '2016.11'
    - e83421694f Merge pull request #43550 from twangboy/osx_fix_preinstall_2016.11.8
    - 1b0a4d39d2 Fix logic in */etc/paths.d/salt* detection
- **PR** #43572: (vutny) cloud.action: list_nodes_min returns all EC2 instances @ *2017-09-18 20:36:44 UTC*
  - 8671b91f62 Merge pull request #43572 from vutny/fix-salt-cloud-list-min-instance-set
  - 21966e7ce8 cloud.action: list_nodes_min returns all instances

- **PR** #43461: (twangboy) Add */norestart* switch to vcredist install @ *2017-09-12 20:33:46 UTC*
  - f2b86fa2db Merge pull request #43461 from twangboy/win_norestart
  - 2d269d1a76 Change all comment markers to '#'
  - d80aea16cb Handle ErrorCodes returned by VCRedist installer
  - fb31e9a530 Add /norestart switch to vcredist install
- **ISSUE** #43267: (brejoc) OSError - Can't delete PIDfile when not root (refs: #43366)
- **PR** #43366: (brejoc) Catching error when PIDfile cannot be deleted @ *2017-09-12 15:31:16 UTC*
  - 90e8ca9c36 Merge pull request #43366 from brejoc/2016.11.pidfile-fix
  - 6e3eb76c79 Removed unused format argument
  - daf4948b3d Catching error when PIDfile cannot be deleted
- **ISSUE** #43386: (rajvidhimar) Scheduler's job_kwargs not working as expected. (refs: #43442)
- **PR** #43442: (garethgreenaway) [2016.11] Fixes to scheduler __pub values in kwargs @ *2017-09-12 15:16:20 UTC*
  - a6c458607a Merge pull request #43442 from garethgreenaway/43386_2016_11_schedule_kwargs_pub
  - e637ecbe86 Merge branch '2016.11' into 43386_2016_11_schedule_kwargs_pub
  - 6114df8dc3 Adding a small check to ensure we do not continue to populate kwargs with __pub_ items from the kwargs item.
- **ISSUE** #43223: (rallytime) Properly deprecate describe_route_table function in boto_vpc module (refs: #43445)
- **PR** #43456: (rallytime) Add Neon to version list @ *2017-09-12 15:00:27 UTC*
  - **PR** #43445: (rallytime) Bump deprecation warning for boto_vpc.describe_route_table (refs: #43456)
  - 3c429299f9 Merge pull request #43456 from rallytime/43445_follow_up
  - 35c1d8898d Add Neon to version list
- **PR** #43441: (meaksh) Use $HOME to get the user home directory instead using '~' char @ *2017-09-11 21:25:20 UTC*
  - 6db7a721c0 Merge pull request #43441 from meaksh/2016.11-salt-bash-completion-fix
  - be4f26ab21 Use $HOME to get the user home directory instead using '~' char
- **ISSUE** #43223: (rallytime) Properly deprecate describe_route_table function in boto_vpc module (refs: #43445)
- **PR** #43445: (rallytime) Bump deprecation warning for boto_vpc.describe_route_table (refs: #43456) @ *2017-09-11 21:23:28 UTC*
  - 05fff44a50 Merge pull request #43445 from rallytime/bump-deprecation-warning
  - c91cd1c6d9 Bump deprecation warning for boto_vpc.describe_route_table
- **PR** #43432: (rallytime) Back-port #43419 to 2016.11 @ *2017-09-11 17:36:37 UTC*
  - **PR** #43419: (gtmanfred) make cache dirs when spm starts (refs: #43432)
  - c57dc5f0e3 Merge pull request #43432 from rallytime/bp-43419
  - c471a29527 make cache dirs when spm starts
- **ISSUE** #43387: (aogier) genesis.bootstrap debootstrap fails if no qemu specified (refs: #43390)
- **PR** #43390: (aogier) better qemu_static parameter mangle in deboostrap management, tests @ *2017-09-11 13:18:30 UTC*

  - 57cccd75d0 Merge pull request #43390 from aogier/43387-genesis-qemu
  - 496f14a7e7 forgot to mock the proper one
  - 51c7a1ba00 only check if static_qemu is_executable()
  - 70642e495d better qemu_static parameter mangle in deboostrap management, tests
- **ISSUE** #43338: (LEMNX) virtualenv never-download (refs: #43356)
- **PR** #43356: (gtmanfred) never-download got readded @ *2017-09-07 17:46:05 UTC*
  - 6106aec696 Merge pull request #43356 from gtmanfred/2016.11
  - 3f19b247f3 Add handler.messages back in for test comparison
  - 9911b04208 fix test
  - 3c6ae99a77 never-download got readded
- **PR** #43325: (doesitblend) mine_interval option is minutes not seconds @ *2017-09-07 16:58:11 UTC*
  - e638fac54e Merge pull request #43325 from doesitblend/salt-mine-doc-fix
  - 1e94d0ac3a Lint: Remove trailing whitespace
  - 51af8f8757 Fix mine_interval phrasing in default file
  - ba0cdd4536 Fix phrasing for mine_interval description
  - 9ff03c2d43 Update Salt Mine documentation to show that the mine_interval option is configured in minutes.
- **ISSUE** #43086: (aogier) pylint: Instance of 'tuple' has no 'extend' member (no-member) (refs: #43105)
- **PR** #43105: (aogier) groupadd module: string does not have attribute 'extend', plus homogeneous *cmd* parm building @ *2017-09-06 15:49:44 UTC*
  - fc587f784a Merge pull request #43105 from aogier/43086-no-member
  - 5111cf8bad Merge branch '2016.11' into 43086-no-member
- **PR** #43333: (damon-atkins) Docs are wrong cache_dir (bool) and cache_file (str) cannot be passed as params + 1 bug @ *2017-09-06 14:21:35 UTC*
  - d97a680372 Merge pull request #43333 from damon-atkins/2016.11
  - 92de2bb498 Update doco
  - fc9c61d12e Update win_pkg.py
  - c91fc14704 Merge branch '2016.11' into 2016.11
  - cb3af2bbbd Docs are wrong cache_dir (bool) and cache_file (str) cannot be passed on the cli (#2)
- **ISSUE** #43295: (V3XATI0N) salt.cache.redis_cache does not actually work. (refs: #43329)
- **PR** #43361: (rallytime) Back-port #43329 to 2016.11 @ *2017-09-05 23:23:01 UTC*
  - **PR** #43329: (johnj) Fix #43295, better handling of consul initialization (refs: #43361)
  - 0c986f5eba Merge pull request #43361 from rallytime/bp-43329
  - b09e5b4379 Fix #43295, better handling of consul initialization issues
- **ISSUE** #35840: (junovitch) preserve_minion_cache is broken in 2016.3+ (refs: #42903)
- **PR** #42903: (junovitch) Fix 'preserve_minion_cache: True' functionality (fixes #35840) @ *2017-09-05 22:57:14 UTC*

  - 22287439e6 Merge pull request #42903 from junovitch/issue-35840-fix-preserve-minion-cache-2016.11
  - c9d4fdbd45 Merge branch '2016.11' into issue-35840-fix-preserve-minion-cache-2016.11
  - 93a68e32a5 Merge branch '2016.11' into issue-35840-fix-preserve-minion-cache-2016.11
  - 079f097985 Fix 'preserve_minion_cache: True' functionality (fixes #35840)
- **PR** #43360: (terminalmage) Fix failing tests in Fedora @ *2017-09-05 22:23:13 UTC*
  - 4860e10757 Merge pull request #43360 from terminalmage/sj-496
  - 433bca14b1 Fix KeyError in yumpkg configparser code on Python 3
  - f6c16935d8 Move –showduplicates before repository-packages
- **PR** #43244: (rallytime) Update release branch section with a few more details @ *2017-09-05 20:27:59 UTC*
  - 4ba2dbe41e Merge pull request #43244 from rallytime/release-branch-clarifications
  - 0d5a46dbaa Update release branch section with a few more details
- **ISSUE** #43348: (9maf4you) network.managed doesn't work on CentOS 7 (refs: #43359)
- **PR** #43359: (gtmanfred) ipaddr_start ipaddr_end for el7 @ *2017-09-05 19:44:24 UTC*
  - 1a012eb3d7 Merge pull request #43359 from gtmanfred/ipaddr
  - 23d9abb560 ipaddr_start ipaddr_end for el7
- **PR** #43247: (rallytime) Back-port various mention bot settings to 2016.11 @ *2017-09-05 18:17:54 UTC*
  - **PR** #43206: (rallytime) Always notify tkwilliams when changes occur on boto files (refs: #43247)
  - **PR** #43183: (basepi) Add basepi to userBlacklist for mention bot (refs: #43247)
  - **PR** #42923: (rallytime) Always notify ryan-lane when changes occur on boto files (refs: #43247)
  - 8f88111be8 Merge pull request #43247 from rallytime/mentionbot-backports
  - 2b85757d73 Always notify tkwilliams when changes occur on boto files
  - 40b5a29f90 Add basepi to userBlacklist for mention bot
  - bad8f56969 Always notify ryan-lane when changes occur on boto files
- **PR** #43277: (rallytime) Add CODEOWNERS file @ *2017-09-01 16:56:53 UTC*
  - 02867fdcd2 Merge pull request #43277 from rallytime/owners-file
  - 2b4da0f0e7 Add CODEOWNERS file
- **PR** #43312: (lordcirth) cron docs: Remind user to use quotes for special strings @ *2017-09-01 16:24:15 UTC*
  - 1c1c484479 Merge pull request #43312 from lordcirth/fix-cron-docs
  - ec94a13750 cron docs: Remind user to use quotes for special strings
- **PR** #43290: (lordcirth) Clarify file.py docs @ *2017-09-01 14:30:04 UTC*
  - 0d1ed4b750 Merge pull request #43290 from lordcirth/fix-file-path-docs
  - 14a4591854 file.py docs: correct group and mode
  - d4214ca283 file.py docs: specify absolute paths
- **PR** #43274: (terminalmage) Use six.integer_types instead of int @ *2017-08-30 21:32:42 UTC*
  - 26ff89539e Merge pull request #43274 from terminalmage/fix-int-types
  - d533877743 Use six.integer_types instead of int

  - 42a118ff56 fixed cmd composition and unified his making across module
  - 881f1822f2 Format fix code example local returner doc

28.2.41 Salt 2016.3.0 Release Notes - Codename Boron

Known Issues

Warning: Some Salt Masters may need to apply a patch for Default Job Cache to prevent a possible crash

An issue exists that prevents the Salt master from cleaning the default job cache. This issue can cause an overconsumption of resources resulting in a crash. 2016.3.0 Salt masters should apply the patch in PR #33555. This issue will be addressed in 2016.3.1.

- issue #33516: When upgrading from 2015.8.10 to 2016.3.0 on centos7/redhat7 salt-minion must be restarted twice.
- issue #33517: SPM does not work on amazon linux 2015 in 2016.3.0.

Backwards-incompatible Changes

- The default path for the *`extension_modules`* master config option has been changed. Prior to this release, the location was a directory named `extmods` in the Salt cachedir. On most platforms, this would put the *`extension_modules`* directory in `/var/cache/salt/extmods`. It has been moved one directory down, into the master cachedir. On most platforms, this is `/var/cache/salt/master/extmods`. Most users won't have to worry about this, but those who have been manually placing custom runners into `/var/cache/salt/extmods/runners`, or outputters into `/var/cache/salt/extmods/output`, etc. will be affected by this. To transition, it is recommended not to simply move the extmods directory into `/var/cache/salt/master`, but to copy the custom modules into the salt fileserver under `salt://_runners`, `salt://_output`, etc. and sync them using the functions in the new *`saltutil runner`*.
- The `pkg.check_db` function has been removed for yum/dnf.

Core Changes

- The `onchanges` requisite now fires if **any** watched state changes. issue #19592.
- The `ext_pillar` functions **must** now accept a minion ID as the first argument. This stops the deprecation path started in Salt 0.17.x. Before this minion ID first argument was introduced, the minion ID could be retrieved accessing `__opts__['id']` losing the reference to the master ID initially set in opts. This is no longer the case, `__opts__['id']` will be kept as the master ID.
- Custom types can now be synced to the master using the new *`saltutil runner`*. Before, these needed to manually be placed under the *`extension_modules`* directory. This allows custom modules to easily be synced to the master to make them available when compiling Pillar data. Just place custom runners into `salt://_runners`, custom outputters into `salt://_output`, etc. and use the functions from the *`saltutil runner`* to sync them.
- The `client_acl` configuration options were renamed to `publisher_acl`.
- Added a new `--config-dump` option (issue #26639).
- TCP Transport presence events were updated to work with a NAT (PR #30629).

- A *minion_pillar_cache* setting was added to save rendered pillar data to cachedir for later use when file_client is set to local (PR #30428).
- Added the ability for binary data (such as a license key) to be distributed via pillar using the *file.managed* (issue #9569).
- Scheduled jobs now include `success` and `retcode` (issue #24237).
- The `saltversioninfo` grain was changed from a string to a list to enable reading values by index. (PR #30082).
- A *pillar_merge_lists* option was added to enable recursively merging pillar lists by aggregating them instead of replacing them (PR #30062).
- Grain values reported by Debian 8 (jessie) when lsb-release is installed were updated for consistency (PR #28649).
- A new option for minions called master_tries has been added. This specifies the number of times a minion should attempt to contact a master to attempt a connection. This allows better handling of occasional master downtime in a multi-master topology.
- The default directory for deploying the salt-thin tarball has changed for salt-ssh. It is now /var/tmp instead of /tmp. Users may also wish to delete any directories in /tmp ending with *_salt/*. (issue #32771)

External Module Packaging

Modules may now be packaged via entry-points in setuptools. See *external module packaging* tutorial for more information.

Cloud Changes

- Refactored the OpenNebula driver and added numerous `--function` and `--action` commands to enhance Salt support for image, template, security group, virtual network and virtual machine management in OpenNebula.
- Added execution/state modules to support the deployment of AWS cognito identity pools (PR #31094).
- Added ability to set tags and listener policies on a AWS ELB (PR #27552).

Platform Changes

- Renamed modules related to macOS. The following module filenames were changed. The virtual name remained unchanged.
- PR ##30558: renamed osxdesktop.py to mac_desktop.py
- PR ##30557: renamed macports.py to mac_ports.py
- PR ##30556: renamed darwin_sysctl.py to mac_sysctl.py
- PR ##30555: renamed brew.py to mac_brew.py
- PR ##30552: renamed darwin_pkgutil.py to mac_pkgutil.py

Package Support

- Ubuntu Xenial: Packages for Ubuntu Xenial (16.04) are available for 2016.3.0 and onwards. See repo.saltstack.com for more information. Note that Xenial comes with Debian's packaged version of Salt 2015.8.8 and official repo.saltstack.com packages are available for 2015.8 releases beginning with Salt 2015.8.11.

Proxy Minion Changes

The deprecated config option `enumerate_proxy_minions` has been removed.

As mentioned in earlier documentation, the `add_proxymodule_to_opts` configuration variable defaults to `False` in this release. This means if you have proxymodules or other code looking in `__opts__['proxymodule']` you will need to set this variable in your `/etc/salt/proxy` file, or modify your code to use the *__proxy__* injected variable.

The `__proxyenabled__` directive now only applies to grains and proxy modules themselves. Standard execution modules and state modules are not prevented from loading for proxy minions.

Support has been added to Salt's loader allowing custom proxymodules to be placed in `salt://_proxy`. Proxy minions that need these modules will need to be restarted to pick up any changes. A corresponding utility function, `saltutil.sync_proxymodules`, has been added to sync these modules to minions.

Enhancements in grains processing have made the `__proxyenabled__` directive somewhat redundant in dynamic grains code. It is still required, but best practices for the `__virtual__` function in grains files have changed. It is now recommended that the `__virtual__` functions check to make sure they are being loaded for the correct proxytype, example below:

```
def __virtual__():
    '''
    Only work on proxy
    '''
    try:
        if salt.utils.is_proxy() and \
           __opts__['proxy']['proxytype'] == 'ssh_sample':
            return __virtualname__
    except KeyError:
        pass

    return False
```

Note: `salt.utils.is_proxy()` has been renamed to `salt.utils.platform.is_proxy` as of the Oxygen release.

The try/except block above exists because grains are processed very early in the proxy minion startup process, sometimes earlier than the proxy key in the `__opts__` dictionary is populated.

Grains are loaded so early in startup that no dunder dictionaries are present, so `__proxy__`, `__salt__`, etc. are not available. Custom grains located in `/srv/salt/_grains` and in the salt install grains directory can now take a single argument, `proxy`, that is identical to `__proxy__`. This enables patterns like

```
def get_ip(proxy):
    '''
    Ask the remote device what IP it has
    '''
    return {'ip':proxy['proxymodulename.get_ip']()}
```

Then the grain `ip` will contain the result of calling the `get_ip()` function in the proxymodule called `proxymodulename`.

Proxy modules now benefit from including a function called `initialized()`. This function should return `True` if the proxy's `init()` function has been successfully called. This is needed to make grains processing easier.

Finally, if there is a function called `grains` in the proxymodule, it will be executed on proxy-minion startup and its contents will be merged with the rest of the proxy's grains. Since older proxy-minions might have used other methods to call such a function and add its results to grains, this is config-gated by a new proxy configuration option called `proxy_merge_grains_in_module`. This defaults to `False` in this release. It will default to True in the release after next. The next release is codenamed **Carbon**, the following is **Nitrogen**.

The example proxy minions `rest_sample` and `ssh_sample` have been updated to reflect these changes.

Syndic Updates

A major performance and management issue was found and fixed in the syndic. This makes the Salt Syndic substantially more reliable and performant. Please make sure that the syndic and the master of masters which syndics attach to are updated, otherwise the syndic fixes alone can cause minor performance issues with older master of masters. Please update masters first, then syndics. Minions do not need to be updated for this fix to work.

Module Changes

- *`file execution module`*: `show_diff` is deprecated in favor of `show_changes`. (PR #30988)
- *`reg execution module`*:
  - Removed the following deprecated functions from the reg module (PR #30956):
    - read_key
    - set_key
    - create_key
    - delete_key
  - Removed force parameter from reg state module
  - Fixed virtual function in state
  - Improved error information for `reg.delete_value` function
- *`jboss7 execution module`*: `deployed` function was decoupled from Artifactory by removing Artifactory-specific functionality. Note that the changes in some of the function arguments break existing state files, see issue #30515 and PR #3080 for details.
- *`pkg state module`*: The `wait` function was removed, the functionality was replaced with the `onchanges` requisite (PR #30297).
- *`firewalld state module`*: A `permanent` argument was added `add_port`. Note that `permanent` defaults to `True`, which changes previous behavior (PR #30275). A `bind` function was also added that allows binding zones to interfaces and sources (PR #29497).
- *`journald beacon module`*: The event string was updated to include a tag. Note this might impact existing reactors based on this beacon. (PR #30116).
- *`postgres_privileges state module`*: The default value of the `prepend` argument was changed from `None` to `public`.
- *`zenoss execution module`*: The `add_device` function was updated with a default value of `1000` for `prod_state` to match the documentation (PR #28924).

- The etcd execution module, state module, returner module, and util module were refactor (PR #28599). This refactor changes error returns for several functions (primarily edge cases):
  - get: Used to return " on key-not-found. Now returns None.
  - set: Used to return " on issues setting keys. Now returns None.
  - ls: Used to return {path: {}} on key-not-found. Now returns None.
  - Tree: Used to return {} on key-not-found. Now returns None.
- *smartos_virt execution module*: Updated to use most of the new smartos_vmadm (PR #28284).
- *apache_conf state module*, *apache_module state module*, and *apache_site state module*: the `enable` and `disable` functions were renamed to `enabled` and `disabled`, respectively. In PR #33562, these functions were readded and properly deprecated and will be removed in Salt 2017.7.0. This fix will be available in 2016.3.1. As a workaround, try

```
apache_module.enable{{ 'd' if grains.saltversioninfo == [2016, 3, 0] else '' }}
```

New Features

Thorium - Provisional New Reactor

The 2016.3 release introduces the new Thorium Reactor. This reactor is an experimental new feature that implements a flow programming interface using the salt state system as the engine. This means that the Thorium reactor uses a classic state tree approach to create a reactor that can aggregate event data from multiple sources and make aggregate decisions about executing reactions.

This feature is both experimental and provisional, it may be removed and APIs may be changed. This system should be considered as ambitious as the Salt State System in that the scope of adding a programmable logic engine of this scale into the event systems is non trivial.

See *Thorium Complex Reactor*.

Improved Mac OS Support

Improved Solaris Support

A lot of work was done to improve support for SmartOS. This work also resulted in improvements for Solaris and illumos as SmartOS.

- rewrite of *vmadm module* (SmartOS)
- rewrite of *imgadm module* (SmartOS)
- deprecation of *virt module* in favor of vmadm (SmartOS)
- implemented *smartos state* (SmartOS)
- improved *zpool module* add SmartOS, illumos and Solaris support
- improved *zfs module* add SmartOS, illumos and Solaris support
- implemented *zpool state*
- implemented *zfs state* implemented *solaris_system* system module to provide better Solaris support (PR #30519)
- other minor fixes to grains, localmod, ...

Tornado Transport

Important: The Tornado Transport wire protocol was changed in 2016.3, making it incompatible with 2015.8 (PR #29339).

Windows DSC Integration (Experiemental)

Dimension Data Cloud Support

A SaltStack Cloud driver for Dimension Data Public Cloud, provides the driver functionality to service automation for any of the Dimension Data Public Cloud locations:

- Deploy new virtual machines
- List and query virtual machine images
- Destroy and query virtual machines

Documentation of the Dimension Data SaltStack integration is found on developer.dimensiondata.com

Minion Blackout

During a blackout, minions will not execute any remote execution commands, except for *`saltutil.refresh_pillar`*. Blackouts are enabled using a special pillar key, `minion_blackout` set to `True`.

See *Minion Blackout*.

Splunk Returner

A Splunk Returner that uses HTTP Event Collector is now available (PR #30718).

SQLCipher Pillar Module

Support was added for retrieving pillar data via queries to SQLCiper databases (PR #29782).

New Modules

The following list contains a link to the new modules added in this release.

Beacons

- *`beacons.adb`*
- *`beacons.glxinfo`*
- *`beacons.memusage`*
- *`beacons.network_settings`*
- *`beacons.proxy_example`*

- *beacons.salt_proxy*

Engines

- *engines.docker_events*
- *engines.redis_sentinel*
- *engines.slack*
- *engines.sqs_events*
- *engines.thorium*

Execution Modules

- *modules.bcache*
- *modules.beacons*
- *modules.boto_cloudtrail*
- *modules.boto_datapipeline*
- *modules.boto_iot*
- *modules.boto_lambda*
- *modules.boto_s3_bucket*
- *modules.chronos*
- `modules.cytest`
- *modules.dockercompose*
- `modules.dsc`
- *modules.ethtool*
- *modules.github*
- *modules.infoblox*
- *modules.iwtools*
- `modules.jenkins`
- *modules.linux_ip*
- *modules.mac_assistive*
- `modules.mac_brew`
- `modules.mac_defaults`
- *modules.mac_desktop*
- *modules.mac_keychain*
- *modules.mac_pkgutil*
- `modules.mac_ports`
- *modules.mac_power*

- *modules.mac_service*
- *modules.mac_shadow*
- *modules.mac_softwareupdate*
- *modules.mac_sysctl*
- *modules.mac_system*
- *modules.mac_timezone*
- *modules.mac_xattr*
- *modules.marathon*
- *modules.minion*
- *modules.openvswitch*
- *modules.opkg*
- *modules.philips_hue*
- *modules.proxy*
- *modules.pushbullet*
- *modules.restartcheck*
- *modules.s6*
- *modules.salt_proxy*
- `modules.ssh_package`
- *modules.ssh_service*
- *modules.sysfs*
- *modules.vboxmanage*
- *modules.win_certutil*
- *modules.win_dism*
- *modules.win_dism*
- *modules.win_license*
- *modules.win_iis*
- *modules.win_task*
- *modules.zabbix*

Pillar

- *pillar.http_yaml*
- *pillar.stack*

Proxy

- *proxy.chronos*
- *proxy.junos*
- *proxy.marathon*
- `proxy.phillips_hue`
- *proxy.ssh_sample*

Roster

- *roster.range*

States

- *states.apache_conf*
- *states.apache_site*
- *states.boto_cloudtrail*
- *states.boto_datapipeline*
- *states.boto_iot*
- `states.boto_lamda`
- *states.boto_s3_bucket*
- *states.chocolatey*
- *states.chronos_job*
- *states.firewall*
- *states.github*
- *states.gpg*
- *states.grafana_dashboard*
- *states.grafana_datasource*
- `states.infoblox`
- *states.jenkins*
- *states.mac_assistive*
- `states.mac_defaults`
- *states.mac_keychain*
- *states.mac_xattr*
- *states.marathon_app*
- *states.openvswitch_bridge*
- *states.openvswitch_port*
- *states.postgres_cluster*

- *states.proxy*
- *states.salt_proxy*
- *states.virt*
- *states.win_certutil*
- *states.win_dism*
- *states.win_license*
- *states.zabbix_host*
- *states.zabbix_hostgroup*
- *states.zabbix_user*
- *states.zabbix_usergroup*

28.2.42 Salt 2016.3.1 Release Notes

Version 2016.3.1 is a bugfix release for *2016.3.0*.

Statistics

- Total Merges: **87**
- Total Issue References: **23**
- Total PR References: **58**
- Contributors: **25** (abednarik, amontalban, anlutro, babilen, cachedout, clburlison, danslimmon, eliasp, glomium, jacobhammons, jfindlay, kev009, lomeroe, michalsuba, neil-williamson, onorua, opdude, rallytime, sjorge, terminalmage, thatch45, ticosax, tomlaredo, twangboy, zigarn)

Final Release of Debian 7 Packages

Regular security support for Debian 7 ended on April 25th, 2016. As a result, 2016.3.1 and 2015.8.10 will be the last Salt releases for which Debian 7 packages are created.

Changelog for v2016.3.0..v2016.3.1

Generated at: 2018-05-27 04:31:54 UTC

- **PR** #33883: (jfindlay) add 2016.3.1 release notes
- **PR** #33866: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-06-08 19:49:56 UTC*
    - be20ce1bbf Merge pull request #33866 from rallytime/merge-2016.3
    - 595d4f2ac3 Fixup new groupadd tests for syntax change in 2016.3
    - c5b4ec0b0f Merge branch '2015.8' into '2016.3'
        - ec09095c45 Merge pull request #33827 from cachedout/issue_33810
            - 9d36f1e474 Fix broken locate.locate function
        - f7b3d0eda0 Merge pull request #33839 from cachedout/fix_pkgresource_test_stacktrace

    - 435547a747 Fix another unit test stacktrace in pkg_resource
- 5f081ef31c Merge pull request #33840 from cachedout/remove_matcher_unit_tests
    - 6297448377 Remove matcher tests
- cda032dab2 Merge pull request #33836 from cachedout/fix_winserver_manager_test
    - 453fb1ac91 Fixing more stupid unit tests
- 1db559afe9 Merge pull request #33805 from jfindlay/pkg_tests
    - 0c069ddc95 states.pkg int tests: skip if pkg mgr unavailable
- 3984b65486 Merge pull request #33808 from jfindlay/gem_tests
    - f7c19a1a58 modules.gem int tests: relax version checks
    - 6af47d2ba7 modules.gem int tests: remove pkgs before testing install
- c30d8a8c61 Merge pull request #33770 from jfindlay/service_tests
    - f13f914755 states.service: add integration tests
    - 90aee79c39 states.service.mod_watch: update unit test
    - d210a92f09 states.service.mod_watch: update sfun and force docs
- 7fdfbe9a28 Merge pull request #33691 from jtand/gem_integration_test
    - ff2dae103d ubuntu doesn't install default gems when ruby is installed
    - 504df9a65a Fixed lint error
    - 0cb1bfa0d3 Removed extra :
    - 86f59b3e80 Made more pythonic
    - 2f36f34981 Fixed salt.util import. Added status check to make sure external resource is available
    - 400a71ec33 Removed redundancies
    - 91db411bea A couple lint fixes
    - c97f3319b9 Add check for gem binary
    - 210aceb402 Refactored tests to not use return messages
    - 9d437bd45d Removed artifact from testing
    - 134e1fa888 Fixed typos, and added destructiveTest decorator
    - 37bc3ad8fd Fixed typo, uninstalled to uninstall
    - 5b23b91ac6 Integration test for gem module
- bb4194bb79 Merge pull request #33777 from sodium-chloride/2015.8-2016-0604-1939
    - c1fd830a1a Fix minor docstring issue of arg being missing
- c749aea409 Merge pull request #33759 from cachedout/issue_31219
    - 15a39f8646 Catch no minions exception in batch mode
- 47d668e071 Merge pull request #33719 from cachedout/fixup_33653
    - 635efa248b Change to just surround the mkdir
    - 21b7123a60 Catch oserror for race condition

        - 11e39e7203 Merge pull request #33712 from meaksh/fix-for-groupadd-module-failures-in-SLE11-2015.8
            - ab738416ba pylint fix
            - bf27e5d36e test_members cleanup
            - ba815dbf76 improvements on groupadd unit tests
            - 3bbc5ae0d9 one line is better
            - a53dc192c9 fix groupadd module for sles11 systems
        - 2c450a7494 Merge pull request #33718 from rallytime/bp-33700
            - a6a446121a Fix speed issue
            - a41146730a Fix incorrect args passed to timezone.set_hwclock
        - b07701f0a0 Merge pull request #33727 from terminalmage/issue33725
            - d8ba7ed5a5 Fix git_pillar edge case for remote repos without a master branch
        - 015e50cec8 Merge pull request #33728 from jfindlay/test_state_test
            - 87e018af2a states.test.configurable_test_state: add unit tests
            - c2d0679c4b states.test.configurable_test_state: refactor change_data
            - f06ff1af1f states.test.configurable_test_state test mode
        - 1cf8fe3f1d Merge pull request #33729 from twangboy/fix_win_servermanager
            - 2de91d166f Fix docstring
            - 9870479d99 Add exclude option to state
            - 50bd76e206 Add exclude option
        - 6c150d840d Merge pull request #33743 from vutny/drop-debian-community-repo-doc
            - 8621f5be54 Debian installation docs: drop section about community-maintained repository
        - 56c0a42e12 Create missing jid dir if it doesn't exist (#33653)
        - 8a566ff4b9 Merge pull request #33654 from twangboy/fix_win_servermanager
            - 6c7b21676a Fix lint and tests
            - 4775e6bdf0 Add additional params to state
            - b0af32346d Add additional params to install and remove
        - 996ff56dd4 Merge pull request #33679 from terminalmage/issue33424
            - 9da40c4437 Append empty dictionaries for saltenvs with no top file
            - 5eb1b3ca62 Only compile the template contents if they evaluate to True

- **ISSUE** #33843: (richardscollin) 2016.3 Test Suite TCP Error (refs: #33860)
- **PR** #33860: (cachedout) Allow socket closes when the socket is disconnected @ *2016-06-08 18:26:16 UTC*
    - 669aa92d59 Merge pull request #33860 from cachedout/issue_33843
    - 2c88e22c07 Use errno
    - e7de99dd0e Correct silly mistake
    - 7a46360a13 Allow socket closes when the socket is disconnected

- **ISSUE** #33818: (saltuser) 2016.3.0 minion default log level INFO (refs: #33821, #33861)
  - **PR** #33861: (cachedout) Set master and cloud to log level warning
  - **PR** #33821: (cachedout) Restore deafault log level to warning (refs: #33861)
- **PR** #33698: (opdude) Vsphere fixes @ *2016-06-08 14:12:17 UTC*
  - a3202f1ad6 Merge pull request #33698 from Unity-Technologies/vsphere-fixes
  - 8ff5906fad Revert "Fix a bug when creating a new VM and changing the network info"
  - 636f4c00f0 Make sure we only use GetConnection if we are using a proxy salt minion
  - 64e9334d56 Fix a bug with self signed certificates and creating a new VM
  - 7834aeda7d Fix a bug when creating a new VM and changing the network info
- **PR** #33771: (twangboy) Additional functionality to win_dism.py @ *2016-06-08 13:58:20 UTC*
  - 01aaf3e2a9 Merge pull request #33771 from twangboy/win_dism
  - 9be45fe37a Fix some more lint
  - 421dc97957 Fix/add unit tests for state
  - 8d66fac74c Add missing unit tests
  - 60f856f73d Fix unit tests for module
  - b574947afe Fix some lint errors
  - a32774c07d Add salt.utils.compare_lists
  - 7ff7050705 Fix incorrect parameters in the state
  - b8ee89f18e Fix typos
  - 10458d8a70 Remove multiple lookups, faster
  - d9b848c0d9 Change to dict instead of error
  - 6510e0a5b0 Add restart option
  - da8562dbc8 Add quiet and norestart options
  - 946371bf1f Handle errors, ensure add/remove
  - 7e6382a8b2 Use list instead of string for cmd
  - fbdd28f144 Add state functions
  - 90a4ee3d96 Merge branch '2016.3' of https://github.com/saltstack/salt into win_dism
  - 00c24abe1f Add get functions
  - c6621053fd Add additional functions
  - 36507845b6 Update documentation
- **ISSUE** #33649: (tyhunt99) 2016.3.0 dockerng state fails comparing cmd configuration (refs: #33851)
- **PR** #33851: (ticosax) [dockerng] Add support for edge case when *Cmd* and *Entrypoint* can't be blanked @ *2016-06-08 13:52:40 UTC*
  - f546e47552 Merge pull request #33851 from ticosax/fix-entrypoint-support
  - 0d40e1c4f3 Add support for edge case when Cmd and Entrypoint can't be blanked
- **ISSUE** #33818: (saltuser) 2016.3.0 minion default log level INFO (refs: #33821, #33861)

- **PR** #33821: (cachedout) Restore deafault log level to warning (refs: #33861) @ *2016-06-07 16:51:46 UTC*
  - 3f6d06a060 Merge pull request #33821 from cachedout/issue_33818
  - 52f1f77a38 Restore deafault log level to warning
- **ISSUE** #33578: (ohauer) 2016.3.0 FreeBSD Failed to load grains defined in grain file disks.disks in function <function disks at 0x80cff9320>, error: (refs: #33604, #33767)
- **PR** #33767: (amontalban) Fix #33604 implementation when 'geom disk list' does not output rotat… @ *2016-06-07 14:41:56 UTC*
  - **PR** #33604: (kev009) Fix #33578 disks grain (refs: #33767)
  - 3e48b701e3 Merge pull request #33767 from amontalban/2016.3
  - b8c0dd5b4c Fix #33604 implementation when 'geom disk list' does not output rotationrate. #33578
- **PR** #33806: (cachedout) Work around upstream cherrypy bug @ *2016-06-07 14:39:57 UTC*
  - a84588c788 Merge pull request #33806 from cachedout/cherrypy_1444
  - 1b537d41b6 Work around upstream cherrypy bug
- **ISSUE** #33754: (zerthimon) boto_s3_bucket.present is not idempotent (refs: #33776)
- **PR** #33776: (danslimmon) Fixed ACL user comparison. Resolves #33754. @ *2016-06-06 11:11:15 UTC*
  - 94f98b4ab8 Merge pull request #33776 from danslimmon/s3-bucket-idempotency-33754
  - 35b84f1877 Fixed bug where _prep_acl_for_compare() would edit but not return
  - f87bc347fd Fixed ACL user comparison. Resolves #33754.
- **ISSUE** #33741: (jopohl) pkg.install: ERROR: Zypper command failure: Unknown option '–no-refresh' (refs: #33763)
- **PR** #33763: (abednarik) Insert –no-refresh before install in Zypper. @ *2016-06-06 10:53:27 UTC*
  - a92e155a04 Merge pull request #33763 from abednarik/abednarik_zypper_no_refresh_fix
  - 7c909a1d7f Insert –no-refresh before install in Zypper.
- **ISSUE** #33647: (closepin) Pillars passed from command-line override pillar subtrees instead of merging (refs: #33764)
- **PR** #33764: (terminalmage) Merge instead of update pillar overrides @ *2016-06-06 10:52:22 UTC*
  - 306848a2d7 Merge pull request #33764 from terminalmage/issue33647
  - 914003c995 Merge instead of update pillar overrides
- **PR** #33772: (danslimmon) Fixed spelling of "through" @ *2016-06-06 10:50:54 UTC*
  - b37a862b70 Merge pull request #33772 from danslimmon/trough-through
  - ea3498aedc Fixed spelling of "through"
- **ISSUE** #33614: (knuta) grains.has_key() always returns false in 2016.3.0 (refs: #33651)
- **PR** #33651: (cachedout) Restore grains context to renderers @ *2016-06-03 20:48:44 UTC*
  - a8d9221631 Merge pull request #33651 from cachedout/issue_33614
  - 5518e1dd14 Fix whitespace
  - 7b50e1766e Better fix
  - 4e18ff7000 Restore grains context to renderers

- **PR** #33757: (cachedout) Reminder not to return non-serializable data from states @ *2016-06-03 19:23:54 UTC*
  - daf462e430 Merge pull request #33757 from cachedout/state_set_doc
  - 500d4ccec2 Reminder not to return non-serializable data from states
- **ISSUE** #33605: (morganwillcock) win_pkg: UnicodeEncodeError where DisplayName includes "Español" (refs: #33670)
- **PR** #33670: (rallytime) Handle non-ascii package names in state.format_log @ *2016-06-03 16:16:53 UTC*
  - a5684ed123 Merge pull request #33670 from rallytime/fix-33605
  - 59bd51f4c8 Update test to correct iteration
  - a580d1c6e0 Add unit test for format_log change
  - e68097445c Revert "Track down more unicode instances and add a test"
  - 9729aed262 Track down more unicode instances and add a test
  - ae332d1f88 Handle non-ascii package names in state.format_log
- **ISSUE** #33588: (whytewolf) rabbitmq_user.present error (refs: #33641)
- **PR** #33723: (rallytime) Back-port #33641 to 2016.3 @ *2016-06-03 16:07:53 UTC*
  - **PR** #33641: (glomium) check rabbitmq version and use different api to validate a users pass… (refs: #33723)
  - 56eab363ff Merge pull request #33723 from rallytime/bp-33641
  - 77a51a00a3 pylint W0141, W0702
  - f8518939a7 check rabbitmq version and use different api to validate a users password
- **ISSUE** #32059: (fuzzy-id) dockerng fails with: create_container() got an unexpected keyword argument 'binds' (refs: #33748)
- **PR** #33748: (ticosax) HostConfig has been introduced by docker api version 1.15 @ *2016-06-03 15:28:40 UTC*
  - c2b970789c Merge pull request #33748 from ticosax/adjust-api-version-host-config
  - 134e4a9abf HostConfig has been intoriduced by docker api version 1.15
- **PR** #33745: (eliasp) Typo (*privilages* → *privileges*) @ *2016-06-03 15:14:37 UTC*
  - e08c685a6c Merge pull request #33745 from eliasp/2016.3-typo-privilages-privileges
  - 646bc426c6 Typo (*privilages* → *privileges*)
- **ISSUE** #33537: (anlutro) apache_module state functions changed names with no deprecation warning or backward compatibility (refs: #33562)
- **PR** #33562: (jfindlay) states.apache_*: readd and deprecate enable and disable @ *2016-06-02 19:51:37 UTC*
  - **PR** #29651: (zigarn) Deb apache fixes (refs: #33562)
  - 5f4c6902aa Merge pull request #33562 from jfindlay/apache_funcs
  - 9b0eb858a6 add note and workaround to release notes
  - 17306bfc69 states.apache_*: readd and deprecate enable and disable
- **ISSUE** #33632: (rbjorklin) dockerng.volume_present: Dryrun isn't dry (refs: #33659)
- **PR** #33659: (danslimmon) Added test mode to states.dockerng. Resolves #33632. @ *2016-06-02 17:45:49 UTC*
  - d3253effe9 Merge pull request #33659 from danslimmon/dockerng-dryrun-33632

  - ef885c1b7e Added test mode to dockerng.volume_present. Resolves #33632.
- **PR** #33696: (clburlison) Update mac native package for upcoming release @ *2016-06-02 17:44:01 UTC*
  - 1d6582b659 Merge pull request #33696 from clburlison/2016.3-pkg-fix
  - b483d1d8a6 Update mac native package for upcoming release
- **PR** #33710: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-06-02 16:19:00 UTC*
  - 78966f5f30 Merge pull request #33710 from rallytime/merge-2016.3
  - b7accb0b3b Merge branch '2015.8' into '2016.3'
  - c8dc70b96a Merge pull request #33685 from jfindlay/get_url_test
    - 2b5035fdc0 modules.cp.get_url: add test for https://
  - 5e022ff29c Merge pull request #33581 from dincamihai/2015.8
    - 788730ea72 DRY test
    - 1d3769ccfa Improve zypper_patcher_config looks
    - 42d8d4195c Assert only gpgautoimport: True works
    - ced75e8e62 Reverse if conditions and rename variable
    - 80bfbe5c52 Reduce dicts and lists to one line where possible
    - 1d5d6d7d60 Update test method names to pass pylint
    - c7ae5907ee Call zypper refresh after adding/modifying a repository
  - 069ee15b7c Merge pull request #33681 from rallytime/bp-33599
    - 45143a599b use requests streaming for uploads/downloads to file (return_bin unchanged) allows downloading files larger than amount of memory (non-stream reads into memory before writing to disk or uploading)
    - 4a9b23f03f first go at having requests use streaming for get/put requests
  - 13537c4891 Merge pull request #33396 from babilen/issue-33393
    - 57e0475cd4 Make pip InstallationError import more robust
    - 291a3e21fa Remove duplicated code.
  - 7bce4ece1a Merge pull request #33652 from terminalmage/zh723
    - 411841603a Lower the log level for failed auths
  - 504989388a Merge pull request #33615 from danslimmon/mysql-traceback-33582
    - 180099ae9f Wrote test for broken server connection
    - c6c3ff02e3 Added some error checking to resolve #33582.
- **ISSUE** #32916: (giannello) file.managed memory usage with s3 sources (refs: #33599, #33682)
  - **PR** #33682: (lomeroe) backport #33599 to 2016.3
  - **PR** #33599: (lomeroe) Fix s3 large file download (refs: #33681, #33682)
- **ISSUE** #33532: (Routhinator) 2016.3 breaks existing formulas that work on 2015.8 (refs: #33688)
  - **PR** #33688: (terminalmage) Undo __repr__() and __str__() parts of d5a7dcc
- **ISSUE** #29265: (mbochenk) mysql_user.present does not work with MySQL 5.7 (refs: #33690, #32440, #30603)
  - **PR** #33690: (neil-williamson) Remove explicit PW column default from mysql_user

  - **PR** #32440: (neil-williamson) Automatically detect correct MySQL password column for 5.7 and fix setting passwords (refs: #33690)
  - **PR** #30603: (michalsuba) addressing #29265 (refs: #32440)
- **PR** #33680: (rallytime) Back-port #32942 to 2016.3 @ *2016-06-01 22:14:20 UTC*
  - **PR** #32942: (onorua) Make tornado raise error configurable (refs: #33680)
  - c725854596 Merge pull request #33680 from rallytime/bp-32942
  - 09751ecb04 Make tornado raise error configurable (#32942)
- **PR** #33677: (twangboy) Pass kwargs to cmd.run @ *2016-06-01 20:20:08 UTC*
  - 9571dad678 Merge pull request #33677 from twangboy/fix_runas
  - 4272afe0d5 Pass kwargs to cmd.run
- **ISSUE** #33529: (djneades) pkg.latest completely broken on FreeBSD in salt-ssh 2016.3 (refs: #33648)
- **PR** #33648: (terminalmage) salt.modules.pkgng: Fix incorrect usage of _pkg() @ *2016-06-01 16:37:46 UTC*
  - d566ec4b31 Merge pull request #33648 from terminalmage/issue33529
  - 4ad80d29b6 salt.modules.pkgng: Fix incorrect usage of _pkg()
- **PR** #33646: (jfindlay) Fix more tmp paths on MacOS @ *2016-06-01 16:36:33 UTC*
  - e92d6e214f Merge pull request #33646 from jfindlay/mac_tests
  - c53a727c18 tests.runtests: use globally-determined tempdir
  - 8295b48459 test.integration: use hard /tmp on MacOS
- **PR** #33656: (cachedout) Fix indentation error in minion.py @ *2016-06-01 16:23:20 UTC*
  - **PR** #33076: (cachedout) Avoid second grains load on windows multiprocessing (refs: #33656)
  - 9603cd3c0d Merge pull request #33656 from cachedout/fix_33076
  - 8259d4091f Fix indentation error in minion.py
- **PR** #33637: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-05-31 18:43:17 UTC*
  - b379dc57fd Merge pull request #33637 from rallytime/merge-2016.3
  - cd05d2bed1 Fix zypper test
  - 74a7b78e00 Merge branch '2015.8' into '2016.3'
    - b47182e47c Merge pull request #33558 from twangboy/fix_win_servermanager
      - 62a6bde0ea Fix comment when already installed
      - 79bc7195dc Fix unit tests
      - 56a6f6bb83 Fix changes
      - 8ebe99ec5e Fix restart_needed
      - 6e478cbda0 Add restart needed
      - 72ebf26616 Add missing import
      - 193583be96 Use dictionary compare for changes in remove
      - 1ae7dd76c1 Use dictionary compare for changes
    - 58d89d66e3 Merge pull request #33555 from cachedout/issue_33544

    - fe7ee7a470 Fix crashing Maintenence process
  - d052908729 Merge pull request #33501 from meaksh/zypper-download-check-signature-2015.8
    - eaaef25c79 lint issue fixed
    - 6b6febb211 unit tests for rpm.checksum() and zypper.download()
  - e2d0c4abb1 Merge pull request #33513 from rallytime/fix-33319
    - 81c1471209 Add a section to the jinja docs about escaping jinja
  - fabc15e616 Merge pull request #33520 from jacobhammons/release-notes.8
    - 42e358af7d Updated version numbers in the docs for the 2016.3.0 release
- **PR** #33638: (rallytime) Back-port #33613 to 2016.3
- **PR** #33613: (abednarik) Updated apache_module for backward compatible. (refs: #33638)

- **ISSUE** #33590: (morganwillcock) ini_manage.options_absent: only works in test mode (TypeError: unhashable type: 'list') (refs: #33606)
- **PR** #33606: (danslimmon) Fixed ini.options_absent. Resolves #33590. @ *2016-05-31 15:51:35 UTC*
  - 23506f8279 Merge pull request #33606 from danslimmon/ini-optionsabsent-33590
  - fb13852102 Fixed ini.options_absent. Resolves #33590.
- **ISSUE** #33578: (ohauer) 2016.3.0 FreeBSD Failed to load grains defined in grain file disks.disks in function <function disks at 0x80cff9320>, error: (refs: #33604, #33767)
- **PR** #33604: (kev009) Fix #33578 disks grain (refs: #33767) @ *2016-05-31 15:17:37 UTC*
  - 44e8c9e720 Merge pull request #33604 from kev009/fix-33578
  - e452ec514e Ignore cdroms in disks grain
  - 8bf0290024 Make disks grain datatyper more resilient
  - **PR** #33631: (babilen) Fix 'virt' state names in cloud controller tutorial
- **PR** #33603: (sjorge) allow esky packages to be build on base64 2015Q4 @ *2016-05-29 00:36:02 UTC*
  - e9a0c9304a Merge pull request #33603 from sjorge/2016.3-smartos-esky
  - 1064102394 add no-wheel, instructions were failing for someone testing due to wheel being used nog producing an egg
  - c85e03ecf7 allow for newer pyzmq in esky packages
  - 1620b8c0fa allow esky packages to be build on base64 2015Q4
- **ISSUE** #33565: (jamesp9) Typo in states/virtualenv_mod.py (refs: #33576)
- **PR** #33576: (tomlaredo) Fix #33565 (typo causes invalid syntax) @ *2016-05-27 16:46:35 UTC*
  - afd3c1b9bd Merge pull request #33576 from rodacom/2016.3
  - 9f7d81e0cc Fix #33565
- **ISSUE** #33530: (kluoto) Centos7 pkg.upgrade failure on 2016.3 (refs: #33549)
- **PR** #33549: (thatch45) Fix for #33530 @ *2016-05-26 19:26:01 UTC*
  - 71145ddda7 Merge pull request #33549 from thatch45/33530
  - b906859fce Fix for #33530
- **PR** #33538: (anlutro) Fix a KeyError if group is provided but not user in cmd states @ *2016-05-26 17:58:05 UTC*

  - 4831c6a353 Merge pull request #33538 from alprs/fix-cmd_user_runas_deprecation_bug
  - c738a0de76 fix a KeyError if group is provided but not user
- **ISSUE** #33543: (arthurlogilab) Thorium documentation is incorrectly formated and appears partially on docs.saltstack.com (refs: #33550)
- **PR** #33550: (jacobhammons) Fixes display of thorium docs @ *2016-05-26 17:57:05 UTC*
  - 5287a1b8c8 Merge pull request #33550 from saltstack/jacobhammons-patch-1
  - 65df3a6fa2 Refs #33543
- **PR** #33509: (twangboy) Detect System Architecture for Mac Build @ *2016-05-26 14:40:54 UTC*
  - 3a95f8a977 Merge pull request #33509 from twangboy/fix_arch
  - 7844059dcf Handle system architecture
- **PR** #33522: (jfindlay) rework modules.mac_brew.latest_version to work around brew version inconsistency @ *2016-05-26 14:19:25 UTC*
  - 0bc881b4da Merge pull request #33522 from jfindlay/mac_pkg
  - 2781377b17 modules.mac_brew: update unit tests
  - 0ed3598fc9 modules.mac_brew int tests: add latest_version test
  - 8789c2d06d modules.mac_brew int tests: add list_upgrades,info_installed
  - be381e0fc9 modules.mac_brew int tests: move decorators to class
  - fa3ec8a2bf modules.mac_brew.latest_version: refactor to use standard methods
  - 58492c29cf modules.mac_brew: add info_installed function
  - 9abf8f4832 modules.mac_brew.list_upgrades: use brew's json output
  - 77a4f5b01e modules.mac_brew: move retcode check to _call_brew
- **PR** #33519: (jacobhammons) New doc site layout, 2016.3.0 release note known issue additions @ *2016-05-26 13:53:21 UTC*
  - 518713f5e5 Merge pull request #33519 from jacobhammons/2016.3.0rel
  - a424c38f5d New doc site layout, 2016.3.0 release note known issue additions
- **PR** #33508: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-05-25 20:34:49 UTC*
  - 9199101ef2 Merge pull request #33508 from rallytime/merge-2016.3
  - a5e0141eda Merge branch '2015.8' into '2106.3'
  - 5a6b037cbd Merge pull request #33507 from rallytime/merge-2015.8
    - 03b0c97520 Merge branch '2015.5' into '2015.8'
    - 6f7fda0354 Merge pull request #33486 from jtand/2015.5
      - d1e210fff8 Merge branch '2015.5' of https://github.com/saltstack/salt into 2015.5
      - ee2ae0ea8a Added docstring examples to glance.image_schema and schema_get
      - 59e90064e6 modules.swift.head does not have a body. Should not be checked for a docstring right now.
    - f72ec1479b Merge pull request #33482 from rallytime/pillar-opts-docs
      - 087564528d Add pillar_opts docs to master.rst

    - dc644b145d Merge pull request #33488 from rallytime/fix-18752
      - b0a9f4181f Add docs for the syndic_finger config
    - a4e84aa7d2 Merge pull request #33454 from scubahub/2015.5
      - df3c0b8e78 Correct (and make consistent) determination of the test flag.
    - 3a52ace673 manage account information for pam (#33473)
  - ee76be3b0b Merge pull request #33503 from rallytime/fix-15252
    - cfc07f7641 Add docs about minion config file in standalone minion docs
  - e9b648e461 Merge pull request #33474 from cachedout/issue_29451
    - aa2bac3a0d Remove debugging
    - 68d8050cb8 Fix diskusage beacon
  - 3bfb6bf719 Merge pull request #33465 from meaksh/check-if-job-returns-successfully-2015.8
    - 9deb70fd8e jobs.exit_success() now works parsing the results of jobs.lookup_id()
    - 7ba40c4f31 jobs.exit_success allow to check if a job has executed and exit successfully
  - 70eb7b66f3 Merge pull request #33487 from jtand/glance_doc_fixes
    - 0b1cae05d9 Added docstring examples to glance methods and nova.list
    - ebf1256545 Don't need to check swift.head due to it having no body
  - 56ea979916 Merge pull request #33481 from rallytime/fix-33423
    - 7fd3e8f361 Fix docs about etcd config options and add pillar_opts doc
  - 2394cdc4bf Merge pull request #33490 from rallytime/fix-16319
    - 0c5548f9d1 Document the postgres.psql_query function
  - ede232f0f1 Merge pull request #33480 from jfindlay/service_doc
    - 29c00a1b1b states.service: clarify function description language
    - 6a9ae09e79 states.service.__virtual__: add load fail reason
  - 4f96cc1f54 Return full pending computer name (#33483)
  - a89be5e9d4 Use six.string_types in jobs runner (#33499)
  - 2e24a04565 Merge pull request #33491 from BlaineAtAffirm/2015.8
    - 7599b18995 fix jobs.list_jobs failing with search_target
  - 1861af427e Merge pull request #33478 from rallytime/bp-32484
    - 042f17efa4 Only unsub if we have a jid
  - b8154b678e Merge pull request #33457 from rallytime/doc-formatting
    - 82f8f3efff Make doc formatting consistent and use correct versionadded
  - 1dfa95651c Don't allow a "repo" kwarg for pkgrepo.managed (#33477)
  - b4071b07f1 Allow for config entry to be a list in a dict for beacons (#33476)
  - 9f56ab4c45 Merge pull request #33469 from meaksh/zypper-download-check-signature-2015.8
    - a65071a6d1 simpler rpm.checksum function
    - 80fe303e38 Renamed check_sig to checksum and some refactoring

    - d56e3f4258 bugfix: showing errors when a package download fails using zypper pkg.download
    - 8a21b9149e check the signature of downloaded RPM files
  - 00f9090928 Add docs about PyYAML's 1024 character limitations for simple keys (#33459)
  - 3b12f396b4 Prevent several minion processes on the same machine (#33464)
  - c8b4f338d8 Make –gpg-auto-import-keys a global param when calling zypper (#33432)
  - 0c4e38ced4 Fix the saltutil.wheel function and add integration tests (#33414)
- **PR** #33505: (twangboy) Fix build script where pip didn't work @ *2016-05-25 18:15:27 UTC*
  - a43ffadcb7 Merge pull request #33505 from twangboy/fix_build_script
  - 7d78e5d612 Fix build script where pip wouldn't work
- **PR** #33076: (cachedout) Avoid second grains load on windows multiprocessing (refs: #33656) @ *2016-05-25 17:10:06 UTC*
  - 4cf40da7d7 Merge pull request #33076 from cachedout/win_grains
  - dab9825c88 Fix indentation error
  - b14e2cce9e Avoid second grains load on windows multiprocessing

28.2.43 Salt 2016.3.2 Release Notes

Version 2016.3.2 is a bugfix release for *2016.3.0*.

Statistics

- Total Merges: **200**
- Total Issue References: **66**
- Total PR References: **177**
- Contributors: **52** (Ch3LL, DarkKnightCZ, DmitryKuzmenko, Inveracity, abalashov, abednarik, adelcast, ajacoutot, amendlik, anlutro, aphor, artxki, bbinet, bensherman, cachedout, christoe, clinta, cro, dmurphy18, dongweiming, eliasp, eradman, farcaller, garethgreenaway, glomium, gtmanfred, isbm, jacobhammons, jacobweinstock, jfindlay, jmacfar, jnhmcknight, justinta, l2ol33rt, lomeroe, meaksh, nulfox, opdude, peterdemin, rallytime, s0undt3ch, secumod, sjmh, sjorge, terminalmage, thatch45, themalkolm, ticosax, tmehlinger, twangboy, vutny, whiteinge)

Returner Changes

- Any returner which implements a `save_load` function is now required to accept a `minions` keyword argument. All returners which ship with Salt have been modified to do so.

Changelog for v2016.3.1..v2016.3.2

Generated at: 2018-05-27 04:37:58 UTC

- **PR** #34988: (rallytime) Update release notes with new changes @ *2016-07-27 15:54:16 UTC*
  - 721e6dcce8 Merge pull request #34988 from rallytime/release-notes-update
  - a2aae987a6 Update release notes with new changes

- **PR** #34946: (anlutro) Fix virtualenv behavior when requirements files are in subdirectories @ *2016-07-27 14:43:27 UTC*
  - d63ac1671c Merge pull request #34946 from alprs/fix-venv_reqs_subdir
  - f773d63cbb normalize requirements path to be absolute
  - bdec73bb03 remove unnecessary os.path.basename logic
- **PR** #34957: (sjmh) Don't fall through to checking auth entries @ *2016-07-26 22:16:17 UTC*
  - f765faa3aa Merge pull request #34957 from sjmh/2016.3
  - 0095dbe530 Don't fall through to checking auth entries
- **PR** #34971: (cachedout) Increase timeout for grains test @ *2016-07-26 22:11:29 UTC*
  - 2d3b95dec9 Merge pull request #34971 from cachedout/increase_timeout_grains_test
  - 82d271b43a Increase timeout for grains test
- **ISSUE** saltstack/salt#34873: (Cashwini) Scheduler on master does not recognize the date strings supported by python dateutil (refs: #34951)
- **ISSUE** #34873: (Cashwini) Scheduler on master does not recognize the date strings supported by python dateutil (refs: #34951)
- **PR** #34951: (vutny) Fix #34873 @ *2016-07-26 17:07:48 UTC*
  - f23e8c525e Merge pull request #34951 from vutny/fix-schedule-dateutil
  - 0faa490991 Fix job scheduling using `when` parameter (by `python-dateutil`)
- **PR** #34935: (rallytime) Avoid UnboundLocalError in beacons module @ *2016-07-26 17:01:23 UTC*
  - **PR** #34894: (rallytime) [develop] Merge forward from 2016.3 to develop (refs: #34935)
  - deb1331601 Merge pull request #34935 from rallytime/beacons-mod-cleanup
  - 97a36ef367 Avoid UnboundLocalError in beacons module
  - **PR** #34956: (cachedout) Increase all run_script timeouts to 30s
- **PR** #34933: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-07-25 22:09:05 UTC*
  - 5d194f2d17 Merge pull request #34933 from rallytime/merge-2016.3
  - 8b295fe4ea Merge branch '2015.8' into '2016.3'
  - ec8fc058d4 Master performance improvement (#34916)
  - 34dc2fd792 Merge pull request #34911 from cachedout/backport_34906
    - 8becec2f4f Backport #34906
  - 6ccc27f697 Merge pull request #34898 from hrumph/list_upgrades_refresh
    - acd4b1a23b Fixes #33620
  - 5c13ee0e72 Merge pull request #34606 from isbm/isbm-config-reading-exit-2015.8
    - 5f5b802c0c Add option to master config reader on ignoring system exit for wrong configuration
    - 6fc677f177 Ignore minion config errors everywhere but the minion itself
    - 8699194647 Remove deprecation: BaseException.message deprecated as of 2.6
    - 0e65cfec91 Fix lint: E8302
    - 67faa56bf1 Use Salt default exit codes instead of hard-coded values

    - a84556e596 Exit immediately on configuration error
    - 43d965907c Raise an exception on any found wrong configuration file
    - 30ed728d05 Cover exception handling in the utils.parsers
    - 5e8c0c6bdb Introduce configuration error exception
- **ISSUE** #34760: (nate-byrnes) XenServer 7 needs correct provider setup for services. (refs: #34915)
  - **PR** #34915: (abednarik) Update service_rh provider to exclude XenServer >= 7.
- **PR** #34926: (rallytime) Lint #34923 @ *2016-07-25 14:53:42 UTC*
  - **PR** #34923: (eliasp) Handle exception when no Slack API key was provided (refs: #34926)
  - a7e7ec6d25 Merge pull request #34926 from rallytime/lint-34923
  - b3514abf1b Lint fixes for #34923
  - 69afcc4060 Handle exception when no Slack API key was provided
- **ISSUE** saltstack/salt#34908: (Ch3LL) Cannot start proxy minion due to keyerror in grains (refs: #34910)
- **PR** #34910: (cachedout) Fix grains error on proxy minions @ *2016-07-22 23:05:46 UTC*
  - c663c8bb5b Merge pull request #34910 from cachedout/proxy_grains
  - 0970ebace8 Fix grains error on proxy minions
  - **PR** #34864: (jmacfar) Check for version in list of installed versions
- **ISSUE** saltstack/salt#34816: (msdogado) VirtuozzoLinux not realized as RedHat by pkg (refs: #‘saltstack/salt‘#34878‘_‘_, #34878)
  - **PR** saltstack/salt#34878: (abednarik) Add VirtuozzoLinux is yumpkg enable list. (refs: #34902)
  - **PR** #34902: (rallytime) Back-port #34878 to 2016.3
  - **PR** #34878: (abednarik) Add VirtuozzoLinux is yumpkg enable list. (refs: #34902)
- **ISSUE** saltstack/salt#34893: (msdogado) rpm VirtuozzoLinux not working (refs: #34901)
- **PR** #34901: (rallytime) Add VirtuozzoLinux to the list of enabled distros for rpm.py @ *2016-07-22 22:23:48 UTC*
  - ad640cc046 Merge pull request #34901 from rallytime/fix-34893
  - 45e2ce10a4 Add VirtuozzoLinux to the list of enabled distros for rpm.py
- **ISSUE** saltstack/salt#34890: (msdogado) VirtuozzoLinux enabling services not working (refs: #34900)
- **PR** #34900: (rallytime) Add VirtuozzoLinux to enabled platforms list in rh_service.py @ *2016-07-22 22:21:20 UTC*
  - 5aa532f98b Merge pull request #34900 from rallytime/fix-34890
  - 12824487cc Add VirtuozzoLinux to enabled platforms list in rh_service.py
- **PR** #34887: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-07-22 18:04:47 UTC*
  - ebebfa647f Merge pull request #34887 from rallytime/merge-2016.3
  - 109b368d19 Merge branch ’2015.8’ into ’2016.3’
    - fb223e1bd4 Invalidate the target cache very quickly (#34862)
    - 1ca1367289 Fail git.latest states with uncommitted changes when force_reset=False (#34869)
    - 4f4381e5b9 Merge pull request #34859 from cachedout/fix_wheel_test
      - b4be66dedf Fix wheel test

- **PR** #34632: (eliasp) Try to create the log directory when not present yet @ *2016-07-22 17:34:31 UTC*
  - eba34f7f4c Merge pull request #34632 from eliasp/2016.3-create-logdir-when-needed
  - 9c89470661 Try to create the log directory when not present yet
- **PR** #34854: (rallytime) Remove string_types import from state compiler @ *2016-07-22 17:20:15 UTC*
  - 965f517889 Merge pull request #34854 from rallytime/cleanup-state-imports
  - 73d3075ce9 Remove string_types import from state compiler
- **ISSUE** saltstack/salt#26171: (HG00) salt-ssh from python2.6 master to python2.7 minion fails on "from _elementtree import *" (refs: #34865)
- **PR** #34865: (thatch45) This needs discussion, since this breaks SUSE @ *2016-07-22 17:19:34 UTC*
  - 584d7606d4 Merge pull request #34865 from thatch45/break_suse
  - 6c5f363921 This needs discussion, since this breaks SUSE
- **PR** #34858: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-07-21 21:01:17 UTC*
  - aaede31f66 Merge pull request #34858 from rallytime/merge-2016.3
  - 9227c3dd26 Merge branch '2015.8' into '2016.3'
  - acc9e31c02 Merge pull request #34822 from thatch45/ssh_fixes
    - b5de492143 fix #34798
    - 5ad6bd7307 fix #34796
  - 5d91139bc9 Merge pull request #34847 from cachedout/pwall
    - 2c8298dc6e Profile logging
    - 3affafa2e9 Add an option to skip the verification of client_acl users
  - 07d1d36653 Merge pull request #34827 from thatch45/34691
    - 1ccf35eca4 fix beacon list to include all beacons being processed
  - b375720251 Merge pull request #34833 from rallytime/bp-28521
    - e50a6783ce SPM: packaging doesn't work in Python 2.6. Fixed.
  - 042646582f Merge pull request #34823 from rallytime/bp-25276
    - a028796eff copy spm.1 man page during setup Refs #25213
  - 6c35d88268 Fix #34648 (#34828)
- **PR** #34852: (rallytime) Skip GCE unit tests - causes test suite to hang @ *2016-07-21 17:52:31 UTC*
  - b3d8143d36 Merge pull request #34852 from rallytime/skip-gce-tests
  - 15b4f5a8b3 Skip GCE unit tests - causes test suite to hang
  - **PR** #34850: (rallytime) Update 2016.3.2 release notes
- **ISSUE** #34215: (rvora) salt-master crashes every few days (refs: #34683)
- **PR** #34844: (vutny) Fix getting total available memory without *psutil* installed @ *2016-07-21 17:12:38 UTC*
  - **PR** #34683: (cachedout) Fix publisher leak (refs: #34844)
  - 650674d14a Merge pull request #34844 from vutny/fix-grains-load-in-config-init
  - 4dc7827020 Fix comment in master config, prevents the service from starting

  - b4cfebb107 Fix Salt failure after merge of #34683
  - **PR** #34848: (rallytime) Update release notes for 2016.3.2
- **ISSUE** saltstack/salt#34345: (edgan) Salt master mode's and salt-ssh mode's top.sls processing aren't the same (refs: #34837)
- **ISSUE** #34345: (edgan) Salt master mode's and salt-ssh mode's top.sls processing aren't the same (refs: #34837)
- **PR** #34837: (thatch45) Fix #34345 @ *2016-07-21 14:36:15 UTC*
  - 52a95b2ea3 Merge pull request #34837 from thatch45/34345
  - 1e8c585cd3 Fix #34345
- **ISSUE** saltstack/salt#32591: (AndrewPashkin) "RuntimeError: maximum recursion depth exceeded" in salt/utils/lazy.py, using Salt-SSH (refs: #34838)
- **PR** #34838: (thatch45) Check if a valid value is passed to unlyif/unless @ *2016-07-21 14:34:29 UTC*
  - 96450ac74d Merge pull request #34838 from thatch45/unless_valid
  - 1f34299a84 Check if a valid value is passed to unlyif/unless
- **ISSUE** saltstack/salt#32525: (anlutro) state.show_low_sls not working in salt-ssh (refs: #34840)
- **PR** #34840: (thatch45) update the state wrapper to include show_low_sls @ *2016-07-21 14:34:02 UTC*
  - 3a5ef86d58 Merge pull request #34840 from thatch45/state_update_ssh
  - 77dce3920c update the state wrapper to include show_low_sls
- **ISSUE** #34762: (aphor) zpool state module needs support for disk vdev (refs: #34791, #34770)
- **PR** #34842: (sjorge) 2016.3 zpool cleanup and fixes @ *2016-07-21 14:32:56 UTC*
  - **PR** #34770: (aphor) zpool state module needs support for disk vdev #34762 (refs: #34842)
  - 5f67318fd7 Merge pull request #34842 from sjorge/2016.3-zpool-simplifaction
  - a7ff9524b0 drop parsing of vdevs, error passthrough from zpool cli
  - 25d6c8139b eliminate hardcoded vdev type from zpool state
  - 47b8dc946c salt.states.zpool - work with updates exec module
  - a5a98845c7 salt.module.zpool - fix bug with properties on/off being parsed as true/false
  - dd64494a19 salt.modules.zpool - drop vdev types to make it more future proof, fallback to zpool cli error messages
  - **PR** #34825: (thatch45) keep this beacon from stack tracing at the loader
- **PR** #34824: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-07-20 20:54:35 UTC*
  - b9db0b0036 Merge pull request #34824 from rallytime/merge-2016.3
  - 094731f4b6 Merge branch '2015.8' into '2016.3'
  - 98fa4a404e Merge pull request #34818 from jtand/mysql_state_integration_test_cleanup
    - 9abb6f91bb Skip mysql state test if mysqladmin is not available
  - 6636f2b449 Merge pull request #34803 from junovitch/issue_24744
    - 64c850410f salt/state.py: set 'chunk['order'] = 0' with 'order: first'; fixes #24744
- **PR** #34670: (isbm) Add "osmajorrelease" grain (2016.3) @ *2016-07-20 14:39:38 UTC*
  - 62ef8fdb35 Merge pull request #34670 from isbm/isbm-osmajorrelease-grain-suse

  - a6bcbd615f Lintfix PEP8: E262
  - 110a422d5a Keep osmajorrelease as a string type for 2016.3 release
  - 208fd33b48 Add unit test for osmajorrelease grain
  - 9a6b2175c6 Implement "osmajorrelease" by killing spaghetti
- **ISSUE** #34215: (rvora) salt-master crashes every few days (refs: #34683)
- **PR** #34683: (cachedout) Fix publisher leak (refs: #34844) @ *2016-07-20 13:57:10 UTC*
  - 6ca9ffa7c7 Merge pull request #34683 from cachedout/issue_34215
  - ccd53e9214 Lint
  - 76eb46fb08 Document master setting
  - 0dfe3aaf31 Set up dynamic config
  - 3cfb82cdd4 Fix silly error
  - 35a845fff5 Only set IPC with write buffer if set
  - b2d636017d Add IPC to minion opts
  - 2c1c92c48e Lint
  - c4395ae84e Dial down default buffer and apply to just write buffer
  - 3e3e2a997e Typo
  - 78f6251c09 Correct issues with config
  - c138cc03e3 Configuration settings for IPC buffers
- **ISSUE** #34762: (aphor) zpool state module needs support for disk vdev (refs: #34791, #34770)
- **PR** #34791: (sjorge) salt.state.zpool tweaks @ *2016-07-19 20:56:47 UTC*
  - 49ab3fd2b5 Merge pull request #34791 from sjorge/zpool-state-tweaks
  - d48c6d2dcb accomidate use of "fake" vdev type disk, this behavior may be broken later if a disk vdev ever gets added to the cli tools. improve documentation explaining how to create a striped pool without the "fake" vdev type
- **PR** #34784: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-07-19 16:30:18 UTC*
  - 1617a7058a Merge pull request #34784 from rallytime/merge-2016.3
  - 3e032dc397 Merge branch '2015.8' into '2016.3'
  - 58021035a9 Merge pull request #34773 from randomed/mysql-returner-startup/2015.8
    - 0cd55eb7d7 Add jid=req handling for mysql returner. It should also store the return jid into the jid list table.
  - 10a1af9949 Remove unnedeed config test (#34751)
  - f19caac8e4 Merge pull request #34754 from cachedout/disable_mid_test
    - 46901c6e65 Disable test
  - 81f29006f2 Merge pull request #34741 from rallytime/bp-34726
    - d949110993 Loop over updated keys in non recursive update
- **ISSUE** saltstack/salt#34630: (bdrung) Spelling errors (refs: #34756, #34722)
- **ISSUE** saltstack/salt#33923: (pavankumar2203) Salt module certutil install doesnt work (refs: #34756)

  - **PR** #34756: (jacobhammons) Rebuild man pages
- **ISSUE** saltstack/salt#27980: (rayba) salt-cloud 2015.5.0 azure provider could not be loaded (refs: #34746)
- **PR** #34746: (rallytime) Update azure lib dep to match the one in cloud.clouds.msazure @ *2016-07-18 18:54:40 UTC*
  - 2a9738f00d Merge pull request #34746 from rallytime/azure-version
  - ead3eb1606 Update azure lib dep to match the one in cloud.clouds.msazure
- **PR** #34744: (justinta) Test valid docs fix @ *2016-07-18 18:22:47 UTC*
  - c0e2657c8e Merge pull request #34744 from jtand/test_valid_docs_fix
  - 4fe33a7695 add directives example to ldap3.modify
  - 6fa40a0d46 Add cli examples for ldap3 module
  - b94e0dd95a ipset.long_range doesn't need a docstring
- **PR** #34740: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-07-18 16:28:56 UTC*
  - d4adf98b85 Merge pull request #34740 from rallytime/merge-2016.3
  - 7d106c78f0 Merge branch '2015.8' into '2016.3'
    - e9e5bbe38b Merge pull request #34721 from rallytime/fix-34703
      - 9c803d05a5 Add output_file option to master config docs
- **PR** #34607: (isbm) Bugfix: Exit on configuration read (backport) @ *2016-07-18 15:15:21 UTC*
  - efc7599f85 Merge pull request #34607 from isbm/isbm-config-reading-exit-2016.3
  - fb7542f920 Add option to master config reader on ignoring system exit for wrong configuration
  - abd10b5782 Ignore minion config errors everywhere but the minion itself
  - e5f43e6711 Remove deprecation: BaseException.message deprecated as of 2.6
  - 23d1031a09 Fix lint: E8302
  - 6b660678fa Use Salt default exit codes instead of hard-coded values
  - 0c2d3511c9 Exit immediately on configuration error
  - c5de6c8c4a Raise an exception on any found wrong configuration file
  - 575767022b Cover exception handling in the utils.parsers
  - 2cf696671f Introduce configuration error exception
  - **PR** saltstack/salt#34607: (isbm) Bugfix: Exit on configuration read (backport) (refs: #34739)
- **PR** #34739: (cachedout) Remove unnedeed config test @ *2016-07-18 15:15:15 UTC*
  - d0e0c0186b Merge pull request #34739 from cachedout/remove_config_test
  - 4625ee65b8 Remove unnedeed config test
- **ISSUE** saltstack/salt#34630: (bdrung) Spelling errors (refs: #34756, #34722)
- **PR** #34722: (rallytime) Various spelling fixes @ *2016-07-16 19:49:54 UTC*
  - abf5b976ed Merge pull request #34722 from rallytime/fix-34630
  - cca9446c37 Various spelling fixes
- **PR** #34714: (sjmh) Fix ldap auth for function matches @ *2016-07-16 19:49:12 UTC*

  - 922cc5a8a7 Merge pull request #34714 from sjmh/fix/ldap_auth
  - d4144039bc Fix ldap auth for function matches
- **PR** #34720: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-07-16 19:22:28 UTC*
  - 40626d786a Merge pull request #34720 from rallytime/merge-2016.3
  - c2130d5a04 Merge branch '2015.8' into '2016.3'
  - 08d00f3a61 Merge pull request #34689 from Azidburn/fix_pkg_sources
    - 2c0fc919b3 fix second run problems with pkg.installed using sources
  - 4cb1ded520 Merge pull request #34695 from isbm/isbm-zypper-product-boolean-values
    - 5ed5142fbc Update test data for 'registerrelease' and 'productline' fields
    - 21444ee240 Bugfix: return boolean only for 'isbase' and 'installed' attributes
  - aaa6f7d80a update 2015.8.11 release notes (#34682)
- **ISSUE** #34661: (chrimi) Cron State documentation lacks information of "New in" for special parameter in cron.present (refs: #34707)
  - **PR** #34707: (rallytime) Add versionadded to "special" option in cron.present state
- **PR** #34696: (isbm) Bugfix: Zypper *pkg.list_products* returns False on some empty values (2016.3) @ *2016-07-15 21:18:21 UTC*
  - 51fce770a5 Merge pull request #34696 from isbm/isbm-zypper-product-boolean-values-2016.3
  - 96021e257c Update test data for 'registerrelease' and 'productline' fields
  - 337eee33ac Bugfix: return boolean only for 'isbase' and 'installed' attributes
- **PR** #34702: (farcaller) Fixed dockerng.list_tags @ *2016-07-15 20:50:35 UTC*
  - 45045f6900 Merge pull request #34702 from farcaller/fixtags
  - 032e35a28e Fixed dockerng.list_tags
- **ISSUE** saltstack/salt#34548: (Inveracity) win_dsc.set_lcm_config does not set multiple values, missing semicolon (refs: #34549, #saltstack/salt`#34549`_)
  - **PR** saltstack/salt#34549: (Inveracity) fixes multiple values in mof configuration (refs: #34681)
  - **PR** #34681: (rallytime) Back-port #34549 to 2016.3
  - **PR** #34549: (Inveracity) fixes multiple values in mof configuration (refs: #34681)
- **PR** #34679: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-07-14 20:59:45 UTC*
  - d57507dde8 Merge pull request #34679 from rallytime/merge-2016.3
  - 0c566dce89 Merge branch '2015.8' into '2016.3'
  - 3192e1674b Merge pull request #34676 from cachedout/partial_revert_34644
    - 64a154826a Revert "Modify lodaer global test to use populated dunders"
  - 3b6f1089b2 Merge pull request #34601 from lorengordon/clarify-doc
    - bfe0dd0b8a Clarifies the proper way to reference states
  - bc63f25a6f Lint 34644 (#34651)
  - 50360263c5 Adjust the mine test a little bit to give it a better chance of success (#34647)
  - 8a0209101e Merge pull request #34642 from jtand/mysql_integration_cleanup

    - dd1559a599 Check that mysqladmin exists before running mysql integration tests.
  - 3e612c3794 Merge pull request #34618 from jtand/network_integration_fix
    - 34bcf9ccfc Changed network state test to use test=True
    - b2616833b0 Some small changes
    - ed59113e94 Change network state integration test to use test=True
- **ISSUE** saltstack/salt#33452: (Ch3LL) Digital Ocean does not return anything on deletion (refs: #34605)
- **PR** #34605: (gtmanfred) catch error if no dns domains exist @ *2016-07-14 15:20:46 UTC*
  - b88c39e1d2 Merge pull request #34605 from gtmanfred/2016.3
  - 37b0943539 catch error if no dns domains exist
- **PR** #34557: (jacobweinstock) handle jboss cli expression type in the parsing of output @ *2016-07-14 15:09:49 UTC*
  - b3dc6031fe Merge pull request #34557 from jacobweinstock/jboss7_cli-handle-expression-type
  - 1945153399 handle jboss cli expression type in the parsing of the output
  - **PR** #34652: (rallytime) Spelling fixes found in sqlite3 pillar docs
- **ISSUE** saltstack/salt#34382: (amontalban) Exception: unsupported operand type(s) for -: 'str' and 'int' (refs: #34565)
- **ISSUE** #34554: (stjack99) num_cpus grain missing with Salt 2016.3.1 on FreeBSD 10.x (refs: #34565)
  - **PR** #34565: (Ch3LL) add num_cpus grain to freebsd
  - **PR** #34621: (justinta) Suse Leap doesn't have 'man'
- **PR** #34619: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-07-12 21:52:01 UTC*
  - 61f5045a0d Merge pull request #34619 from rallytime/merge-2016.3
  - f734afd0b0 Merge branch '2015.8' into '2016.3'
  - 9f123543e5 Merge pull request #34617 from rallytime/merge-2015.8
    - 3026df346f Merge branch '2015.5' into '2015.8'
    - 57df38e685 Update github IP for ssh state integration tests (#34592)
    - 2e1007254b Avoid circular imports when calling salt.utils functions (#34584)
  - b90ae407f9 Add support for edge case when Cmd and Entrypoint can't be blanked (#34593)
  - 12b579c4e3 When sorting list actual_data, make it a list (#34590)
  - 7dd8035c62 Gate docker unit test to check for docker (#34591)
  - ae38c874da Add a bunch of documentation on getting files from other environments (#34560)
  - 91e0656d44 Merge pull request #34531 from terminalmage/issue34397
    - d0fec1b8f6 salt/modules/zypper.py: accept ignore_epoch argument
    - 5ae9463c1f salt/modules/yumpkg.py: accept ignore_epoch argument
    - c2791117af salt/modules/rpm.py: accept ignore_epoch argument
    - c5de8b880d salt/modules/ebuild.py: accept ignore_epoch argument
    - 4ee8e8f037 salt/modules/aptpkg.py: accept ignore_epoch argument
    - 5b123b403c Pass ignore_epoch to salt.utils.compare_versions()

    - 07368fac40 Accept ignore_epoch argument for salt.utils.compare_versions()
  - e99befad47 Merge pull request #34545 from terminalmage/docker-exec-driver
    - dd5838e242 Handle cases where Docker Remote API returns an empty ExecutionDriver
  - **PR** #34585: (rallytime) [2016.3] Avoid salt.utils circular imports when using "from"
- **PR** #34616: (jacobhammons) Adds a mock required for the network settings beacon @ *2016-07-12 19:09:30 UTC*
  - c8bdfb272d Merge pull request #34616 from jacobhammons/network-settings-mock
  - 5e2ddb5eb0 Adds a mock required for the network settings beacon
- **PR** #34553: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-07-11 19:36:26 UTC*
  - d8c8b4ac6f Merge pull request #34553 from rallytime/merge-2016.3
  - 815c8b38d5 Merge branch '2015.8' into '2016.3'
    - 7120d43df0 Merge pull request #34546 from rallytime/rename-boto-secgroup-test
      - f8a3622be7 Rename unit.states.boto_secgroup to unit.states.boto_secgroup_test
    - ca92061821 Merge pull request #34537 from rallytime/rename-simple-test
      - ceefb6e34c Rename tests.unit.simple to tests.unit.simple_test
    - fbab2f8c2b [2015.8] Update bootstrap script to latest stable (#34527)
    - 6b8c76af83 Prevent many errors in the test suite in loader tests (#34521)
    - c2f296c95b Fix wrong order of retention_policy_exists (#34507)
    - 685df80929 Merge pull request #34518 from terminalmage/fix-pkg.latest-test
      - 4aef44ecdf Fix pkg.latest integration test for non-LTS ubuntu
- **PR** #34569: (eliasp) Minor doc fixes for PostgreSQL states @ *2016-07-11 14:02:13 UTC*
  - 5b002e11b4 Merge pull request #34569 from eliasp/2016.3-postgres-doc
  - 221da29ef5 Typo (*defaul* → *default*)
  - ba3d7c624b Add code formatting
  - b3409c97a2 Fix typo (*seens* → *seen*)
- **PR** #34524: (terminalmage) yumpkg: Avoid spurious logging in pkg.upgrade @ *2016-07-07 22:06:01 UTC*
  - 7e1abd77ba Merge pull request #34524 from terminalmage/yumpkg-upgrade-logging
  - 40992f0790 yumpkg: Avoid spurious logging in pkg.upgrade
- **ISSUE** #34439: (edgan) Fast memory leak on ctrl-c out of salt '*' state.highstate (refs: #34490)
  - **PR** #34490: (cachedout) Fix master crash on ctl-c for long-running job
- **PR** #34520: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-07-07 19:22:40 UTC*
  - b9e87620f5 Merge pull request #34520 from rallytime/merge-2016.3
  - 27988dde48 Merge branch '2015.8' into '2016.3'
  - a516f116d1 Merge pull request #34513 from cachedout/lower_loader_log
    - 733c5d00c0 Lower the log level for modules which cannot be loaded to trace
  - 63f0451041 Merge pull request #34498 from rallytime/bootstrap-tutorial-doc-fix
    - 23c5739c3b Use -O in wget develop example in bootstrap tutorial

  - 3ebba020b6 Rename some unit test files by adding _test (#34503)
  - 8722257b82 Improve top file merging documentation (#34505)
  - 6ce7cb9616 Gracefully handle non-XML output in GlusterFS execution module. (#34492)
  - 75299456be Use skipTest for network state integration test (#34489)
  - 0f3f87fbc5 Update dnsmasq.get_config docs to use correct config_file param. (#34488)
- **ISSUE** #34224: (tehsu) salt-cloud to rackspace uses public ip instead of private (refs: #34499)
- **PR** #34499: (gtmanfred) remove unnecessary block parsing ip addrs for nova @ *2016-07-07 16:23:46 UTC*
  - 58f46eae15 Merge pull request #34499 from gtmanfred/2016.3
  - 019671d4c2 remove unnecessary block parsing ip addrs for nova
  - **PR** #34468: (twangboy) Use Python 2.7.12 for Windows Build
  - **PR** #34493: (twangboy) Use Python 2.7.12 for Mac Build
- **PR** #34486: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-07-06 17:28:31 UTC*
  - 95094c73ff Merge pull request #34486 from rallytime/merge-2016.3
  - 2b307b7ea1 Merge branch '2015.8' into '2016.3'
  - e2f576e847 Merge pull request #34462 from terminalmage/git-describe-always
    - 6ef7ee198e Restrict use of --always to git 1.5.6 and newer
    - c554b22fc8 modules/git: added --always parameter for git.describe().
  - 85f1f18239 Merge pull request #34467 from rallytime/bp-34457
    - 746883741f Only access key metadata if we found key metadata
  - 9e15337b74 Merge pull request #34432 from twangboy/fix_file.append
    - 13f11fddce Remove refactoring code
    - 78f7c530bb Remove unit tests, integration tests written
    - b83392edea Remove len() in favor of boolean test
    - 4373408163 Fix line error
    - 2479b53e2f Fix erroneous report on newline code
    - 75b6ed1fd5 Change back to binary read
    - 65753cff6d Use os.linesep instead of n
    - a55d63f086 Fix object names
    - 3e2fe12e5e Add new line if missing
    - 0b7821c8db Fix file.append state
  - 91e095bb41 Merge pull request #34429 from terminalmage/pkg-latest-versioncheck
    - 667f31a72a Skip version checking for targeted packages in pkg.latest state
  - 0a264597ca Forgot reference to inotify (#34455)
- **ISSUE** #33915: (mattglv) Orchestration runner output on Success vs Failures in 2016.3.0 (refs: #34459)
- **PR** #34459: (terminalmage) Ignore retcode when formatting highstate output @ *2016-07-06 03:59:23 UTC*
  - 7867d49193 Merge pull request #34459 from terminalmage/issue33915

  - 82a70e015f Ignore retcode when formatting highstate output
- **ISSUE** #34371: (erikgrinaker) git.detached does not work with commit ID as ref (refs: #34463)
- **PR** #34463: (terminalmage) states/git: pass required cwd parameter to git.describe. @ *2016-07-06 03:59:05 UTC*
  - ae6902290a Merge pull request #34463 from terminalmage/issue34371
  - f981a5646a states/git: pass required cwd parameter to git.describe.
- **ISSUE** #34395: (artxki) Nonfunctioning default_password in states.postgres_user.present (refs: #34436)
- **PR** #34466: (rallytime) Back-port #34436 to 2016.3 @ *2016-07-06 03:57:15 UTC*
  - **PR** #34436: (artxki) Fix #34395 Nonfunctional default_password in states.postgres_user.present (refs: #34466)
  - 8f8a6d2f68 Merge pull request #34466 from rallytime/bp-34436
  - e97c00b018 Fix #34395 Nonfunctional default_password in states.postgres_user.present
  - **PR** #34453: (justinta) Arch linux does not have osrelease or osmajorrelease grains
- **ISSUE** #33697: (asloboda-cisco) Client clash with Tornado IOLoop (refs: #34456)
- **PR** #34456: (thatch45) Be more careful when making the SMinion @ *2016-07-05 18:41:57 UTC*
  - fc67a4e216 Merge pull request #34456 from thatch45/2016.3
  - edd6b95c60 we need to be more careful when making the SMinion
- **PR** #34452: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-07-05 17:49:19 UTC*
  - 72b4d6b52c Merge pull request #34452 from rallytime/merge-2016.3
  - 91120dba01 Merge branch '2015.8' into '2016.3'
  - 7bb0868c66 Merge pull request #34451 from rallytime/merge-2015.8
    - 55a91e22be Merge branch '2015.5' into '2015.8'
    - 8c72ee56e4 Merge pull request #34435 from cachedout/backport_config_dir_integration
      - 0e2c71a537 Backport change to integraiton test suite
  - e65d1ae374 Merge pull request #34401 from terminalmage/rpm-version_cmp
    - 7cefd4182d Use rpmdev-vercmp as a fallback for version comparison on RHEL5
  - 5ddf417432 Merge pull request #34366 from steverweber/fix_servicerestart
    - 7847c39024 Update service.py
  - 485454febb Merge pull request #34426 from cro/inotify-linux-only
    - 54a02f25ba Document that inotify is Linux only
- **PR** #34427: (twangboy) Automated signing fixes for Ubuntu 16.04, 14.04, 12.04 (for dmurphy) @ *2016-07-05 15:18:46 UTC*
  - 7508d291d2 Merge pull request #34427 from twangboy/sign_fx
  - c804480982 Add changes suggested by @cachedout
  - 494deda074 Automated signing fixes for Ubuntu 16.04, 14.04, 12.04
- **ISSUE** #34379: (UtahDave) variable referenced before assignment (refs: #34400)
- **PR** #34400: (cachedout) Fix uninitialized value @ *2016-07-01 17:42:55 UTC*
  - b3875f397d Merge pull request #34400 from cachedout/issue_34379

  - b413f05a4f Fix uninitialized value
- **PR** #34404: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-07-01 15:02:09 UTC*
  - d1cd36ab2b Merge pull request #34404 from rallytime/merge-2016.3
  - 8398de0baf Merge branch '2015.8' into '2016.3'
  - fe18bbb527 Merge pull request #34392 from cro/salt-cloud-doc-clarify
    - 6cce575d40 Clarify that salt-cloud doesn't get installed by bootstrap
  - 45b8fb10d7 Merge pull request #34373 from jtand/network_state_integration_test
    - 1d24053e36 network.system sls file
    - 4a9e6af542 network.routes sls file
    - 76c90b2ef6 network.managed sls file
    - 84a36369fa Added network state integration test
  - d6af1de0b7 Optimize pkg integration tests and add a couple new tests (#34377)
- **PR** #34292: (twangboy) Fix runas function for System Account @ *2016-06-30 18:25:09 UTC*
  - ad63b1d3d3 Merge pull request #34292 from twangboy/fix_runas
  - 433f300eba Enable all privileges
  - 5584cc2c6f Handle users that aren't admin
  - e9d2402c0b Fix runas function for System Account
- **PR** #34388: (rallytime) Back-port #34378 to 2016.3 @ *2016-06-30 17:50:48 UTC*
  - **PR** #34378: (adelcast) network_settings.py: fix documentation (refs: #34388)
  - be9a831ef6 Merge pull request #34388 from rallytime/bp-34378
  - 2040dbeca5 network_settings.py: fix documentation
  - **PR** #34352: (cro) Esxi dvs
  - **PR** #34386: (rallytime) Beacon network docs
- **PR** #34376: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-06-30 14:31:28 UTC*
  - 5a44b077a0 Merge pull request #34376 from rallytime/merge-2016.3
  - 3149da1bcc Merge branch '2015.8' into '2016.3'
  - af8ef1e461 Merge pull request #34368 from rallytime/merge-2015.8
    - 3bce0cb510 Merge branch '2015.5' into '2015.8'
    - 970aaa46d4 Merge pull request #34252 from gtmanfred/2015.5
      - 82183f1572 return list of nodes for lxc driver when called directly
  - 94e094652c Back-port #34324 to 2015.8 (#34344)
  - 11dc0203b0 Making salt-ssh pass proper return codes for jinja rendering errors (#34342)
- **PR** #34365: (sjorge) fixes computenode_* grains on SmartOS compute nodes @ *2016-06-29 17:55:24 UTC*
  - 3808d849fe Merge pull request #34365 from sjorge/2016.3-fix-broken-smartos-grains
  - 3ff895cacf fixes computenode_* grains on SmartOS compute nodes

- **PR** #34353: (cro) Remove proxy check and additional GetConnection–this makes the proxy… @ *2016-06-29 14:54:47 UTC*
  - 65efb55917 Merge pull request #34353 from cro/pyvmomi-ssl-fail
  - 14ea29f446 Remove proxy check and additional GetConnection–this makes the proxy fail to start. Need to check to see if proxy memory leak is back.
- **PR** #34348: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-06-28 23:28:45 UTC*
  - c89d1ad27f Merge pull request #34348 from rallytime/merge-2016.3
  - c87a108a12 Don't forget the pylint disables for range
  - 359e8ca2ce Pylint fixes
  - f9ab8ba46d Merge branch '2015.8' into '2016.3'
    - f6bd1ad47e Revert py3modernize lint changes (#34339)
    - 046bdaa9f2 Merge pull request #34306 from ghedo/iptables_flush_table
      - 882c6c9c86 Do not force 'filter' table when flushing
    - 0c60feac02 Doc clarifications to file modules, addition of new *profile* log level to docs, fixed example in dnsmasq (#34323)
    - b793426c23 Remove unnecessarily-disabled sanity check (#34325)
    - c5890a0eca Merge pull request #34335 from rallytime/merge-2015.8
      - 2296587536 Merge branch '2015.5' into '2015.8'
      - 6cce545d92 Merge pull request #34313 from rallytime/bootstrap-2015.5
      - c7db73be92 [2015.5] Update to latest bootstrap script v2016.06.27
    - a6d3cc637b Typo in dockerio doc (#34319)
    - dd4c937009 Merge pull request #34312 from rallytime/bootstrap-2015.8
      - 944a393f89 [2015.8] Update to latest bootstrap script v2016.06.27
    - 91703d2dc4 Merge pull request #34307 from rallytime/fix-test-example
      - f44a0543fe Fix test example in integration testing docs
- **ISSUE** #34255: (tmehlinger) fire_event requisite does not work in orchestration states (refs: #34256, #34343)
  - **PR** #34343: (rallytime) Back-port #34256 to 2016.3
  - **PR** #34256: (tmehlinger) detect running from master in State.event method (refs: #34343)
- **PR** #34338: (themalkolm) Add listen/listen_in support to stateconf.py @ *2016-06-28 21:50:14 UTC*
  - 0b9cb602fe Merge pull request #34338 from themalkolm/patch-2
  - cd63541325 Add listen/listen_in support to stateconf.py
- **PR** #34283: (sjorge) 2016.3 mount vfstab support @ *2016-06-28 19:23:39 UTC*
  - 80a659bb51 Merge pull request #34283 from sjorge/2016.3-mount-fstab
  - b8c6948cd5 fixes broken rm_fstab test due to missing __grain__.kernel
  - d633e774ea actually do the cleanup, oops
  - 987c240850 minor cleanup
  - c3667203bf add test for vfstab

  - 80e9d1d278 set __grains__ for fstab unit test
  - f0f5d449c3 mount.vfstab implemented on Solaris like platforms
  - 4398e8841b undo some changes to mount.fstab and mount.rm_fstab, create mount.vfstab and mount.rm_vfstab
  - 133d3bb2bb mount.set_fstab errors out on Solaris like platforms
  - c0863fb024 mount.rm_fstab works with Solaris like platforms
  - 151799ea74 initial vfstab support (Solaris like platforms)
- **ISSUE** #34321: (Ch3LL) Raspberry Pi salt-minion missing osmajorrelease grain (refs: #34322)
- **PR** #34322: (Ch3LL) add osmajorrelease grain for raspbian @ *2016-06-28 19:08:39 UTC*
  - 75aad073a9 Merge pull request #34322 from Ch3LL/add_grains_majorrelease_test
  - 693cc61aa4 add osmajorrelease to ubuntu and fix pylint
  - 2fc3e8a54b add osmajorrelease grain for raspbian
- **PR** #34337: (clinta) Change merge-if-exists logic to properly report changes @ *2016-06-28 18:41:56 UTC*
  - 81547f413d Merge pull request #34337 from clinta/serialize-merge
  - ebe7def2fb Change merge-if-exists logic to properly report changes
- **PR** #34300: (vutny) Make apache.configfile state handle the Options list correctly @ *2016-06-28 18:34:45 UTC*
  - affc65dc79 Merge pull request #34300 from vutny/fix-apache-vhost-options
  - 52001afdde Fix apache.configfile state example
  - 64a9442e38 apache.config: correctly output a list of the Options
- **ISSUE** #33588: (whytewolf) rabbitmq_user.present error (refs: #34333)
  - **PR** #34333: (rallytime) Back-port #33734 to 2016.3
  - **PR** #34304: (rallytime) Back-port #33734 to 2016.3 (refs: #34333)
  - **PR** #33734: (glomium) modules/rabbitmq.py version checking had a logical error (refs: #34333, #34304)
- **ISSUE** #34329: (clinta) file.serialize merge_if_exists fails: 'function' object has no attribute 'deserialize' (refs: #34330)
  - **PR** #34330: (clinta) fix #34329
- **ISSUE** #34170: (rodoyle) ps.top raises ValueError "too many values to unpack" when psutil > 4.1.0 (refs: #34318)
  - **PR** #34318: (rallytime) Back-port #32182 to 2016.3
  - **PR** #32182: (dongweiming) Fix psutil.cpu_times unpack error (refs: #34318)
- **PR** #34311: (rallytime) [2016.3] Update to latest bootstrap script v2016.06.27 @ *2016-06-27 18:59:27 UTC*
  - 1398b1c51e Merge pull request #34311 from rallytime/bootstrap-2016.3
  - 75aa7047bc [2016.3] Update to latest bootstrap script v2016.06.27
- **ISSUE** #34129: (onorua) fqdn_ip4 and fqdn_ip6 are empty on 2016.3+ (refs: #34284)
- **PR** #34284: (rallytime) Don't require 'domain' to be present before checking fqdn_ip* grains @ *2016-06-27 17:06:17 UTC*
  - dc8462451d Merge pull request #34284 from rallytime/fix-34129
  - 5f45a8ff73 Don't require 'domain' to be present before checking fqdn_ip* grains

- **ISSUE** #30493: (sjorge) salt.modules.status mostly broken on solaris like operating systems. (refs: #34296)
- **PR** #34296: (sjorge) 2016.3 status module now works on Solaris like platforms @ *2016-06-27 16:49:41 UTC*
  - 259935d6d2 Merge pull request #34296 from sjorge/2016.3-module.status
  - a26340c555 make status.all_status work on Solaris like platforms
  - 33e24fa697 make status.cpustats work on Solaris like platforms
  - d214e9c776 correctly cast to int for status.netdevs on Solaris like platforms
  - b74761b52d make status.cpuinfo support Solaris like platforms and OpenBSD
  - 2cd76d5ab5 make status.diskstats work on Solaris like platforms
  - 3211538830 make status.diskusage work on Solaris like platforms
  - a12b311a62 make status.netdev compatible with Solaris like platforms
  - 3bc01458aa make status.netstats compatible with Solaris like platforms
  - 25678901fa avoid KeyError in ping_master
  - 81d7fc98d8 make status.vmstats work on Solaris like platforms and OpenBSD
- **PR** #34281: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-06-24 21:51:43 UTC*
  - 376bec0455 Merge pull request #34281 from rallytime/merge-2016.3
  - ae8ad9329c Merge branch '2015.8' into '2016.3'
  - d235b1245b Merge pull request #34233 from thegoodduke/for_2015.8_ipset
    - 4da5e35bf4 ipset: fix the comment containing blank
  - 65c5675a3f Merge pull request #34257 from rallytime/fix-34037
    - d7a5e9b10e Remove test that doesn't actually test anything
    - c4c037d600 Use 'config_dir' setting instead of CONFIG_DIR in gpg renderer
  - 203870f147 Merge pull request #34274 from clinta/2015.8
    - 6572454918 Don't escape source before calling managed
  - a59dc85a15 Merge pull request #34258 from rallytime/merge-2015.8
    - ea914b67cd Merge branch '2015.5' into '2015.8'
    - 8d5ed91980 Merge pull request #34225 from richardscollin/fix-win-set-datetime
      - 6286771ef7 Fix win_system.set_system_date_time
    - cb1e8bf082 Merge pull request #34232 from thegoodduke/for_2015.5_ipset
      - 344eb60762 ipset: fix commont containing blank
- **PR** #34271: (opdude) Fixed symlinks on windows where the slashes don't match @ *2016-06-24 17:05:25 UTC*
  - 805171c949 Merge pull request #34271 from Unity-Technologies/hotfix/windows_symlinks
  - e0a1a55431 Fixed symlinks on windows where the slashes don't match
- **ISSUE** #14915: (johngrasty) SmartOS/OmniOS - mount module fails. (refs: #34254)
  - **PR** #34254: (sjorge) Fix for #14915
- **PR** #34259: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-06-24 14:27:30 UTC*
  - 39579ce5f6 Merge pull request #34259 from rallytime/merge-2016.3

  - eeaf3cc1e7 Merge branch '2015.8' into '2016.3'
  - 92962957c8 Merge pull request #34093 from terminalmage/issue33873
    - 5edb45d746 win_pkg: refresh pkg database if refresh=True passed to version() or list_pkgs()
    - 0078adee35 Catch CommandExecutionError in pkg states
  - cb5399787c Merge pull request #34136 from meaksh/salt-suse-os-detection-2015.8
    - 97f1958863 some cleanup and renaming
    - 72c8e5d78f better way to check for openSUSE Leap
    - 548971bdc9 Fix for SUSE OS grains in 2015.8
- **PR** #34134: (meaksh) Fixed behavior for SUSE OS grains in 2016.3 (refs: #34136) @ *2016-06-23 20:24:51 UTC*
  - **PR** #33903: (meaksh) Fetching grains['os'] from /etc/os-release on SUSE systems if it is possible (refs: #34134)
  - 3acda896f2 Merge pull request #34134 from meaksh/salt-suse-os-detection
  - 23ce0b431b some cleanup and renaming
  - 516bbc454d better way to check for openSUSE Leap
  - 44eda2ad9f Fix for openSUSE Tumbleweed
  - 0d4a710d86 fixes for fopen mock and some os_release_map for SLES11SP3
  - d6410a03b8 unit tests for SUSE os grains detection
  - 47ecb7013b Normalization of osfullname grain for openSUSE
  - 9c81f434fa one clause to set OS grain from CPE_NAME
  - d78d57b717 Test fixed: get OS grain from /etc/os-release if possible
  - d80e0532ff fix: osarch_mock
  - db00ec756d osarch mock for unit test
  - dabc5cab7e lint fix
  - 9ac514724b testing if SUSE os grain is set from /etc/os-release
  - bc671336a7 Getting the 'os' grain from CPE_NAME inside /etc/os-release for SUSE and openSUSE
  - 64af4d4145 Adding SLES_SAP to OS_FAMILY_MAP
- **ISSUE** #34137: (christoe) Win_task info function broken (refs: #34159)
- **ISSUE** #34135: (christoe) Arguments to Windows task creation module are not used (refs: #34159)
- **PR** #34159: (christoe) Fixes to the win_task module @ *2016-06-23 17:54:53 UTC*
  - 5f42fd4486 Merge pull request #34159 from christoe/2016.3
  - f4143669db Fixes #34135, Fixes #34137
- **PR** #34223: (peterdemin) Fixed typo in filtering LDAP's potential_ous @ *2016-06-23 17:26:31 UTC*
  - 0a0267149f Merge pull request #34223 from peterdemin/bugfix-eauth-ldap-expanding
  - 8bb03ec109 Fixed typo in filtering LDAP's potential_ous
- **PR** #34239: (vutny) file.find module: fix handling of broken symlinks @ *2016-06-23 17:25:17 UTC*
  - f74f176bd5 Merge pull request #34239 from vutny/file-find-broken-symlinks

  - 7e164c4f86 file.find module: fix handling of broken symlinks
- **PR** #34229: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-06-22 22:57:00 UTC*
  - 4157f6fd39 Merge pull request #34229 from rallytime/merge-2016.3
  - 940ac86d4e Merge branch '2015.8' into '2016.3'
  - 56c7267631 fix regression from #33681 which causes pulling a list of s3 objects via s3.query to fail (#34208)
  - 02eb331494 Fix a pair of gitfs bugs (#34218)
  - 6d643cd528 Merge pull request #34182 from rallytime/fix-34043
    - b7d49c5052 Handle child PIDs differently depending on the availability of psutils
  - 5d3ec31564 Clarify pkg.list_repo_pkgs docstring for held packages (#34188)
  - 5bca5c42f1 Change target for dockerng assuming default status to Nitrogen release (#34206)
- **ISSUE** #33879: (Ch3LL) saltutil.wheel minions.connected does not return anything with remote minions (refs: #34214)
- **PR** #34214: (rallytime) Update saltutil.wheel docs to specify remote vs local minion behavior @ *2016-06-22 19:22:30 UTC*
  - b5ea1495af Merge pull request #34214 from rallytime/fix-33879
  - 1be05f6a87 Update saltutil.wheel docs to specify remote vs local minion behavior
- **ISSUE** #34074: (fooka03) Unable to use S3 file backend with 2016.3.1 on Ubuntu 14.04 or 16.04 (refs: #34209, #34208)
- **ISSUE** #32916: (giannello) file.managed memory usage with s3 sources (refs: #33599, #33682)
- **PR** #34209: (lomeroe) fix regression in s3.query from #33682 @ *2016-06-22 18:50:19 UTC*
  - **PR** #33682: (lomeroe) backport #33599 to 2016.3 (refs: #34209)
  - **PR** #33599: (lomeroe) Fix s3 large file download (refs: #33681, #33682)
  - 4205fd605c Merge pull request #34209 from lomeroe/fix_s3_utils_regression_33682
  - a2b99703b1 fix regression in s3.query from #33682
- **PR** #34222: (cachedout) Lint 34200 @ *2016-06-22 18:48:54 UTC*
  - **PR** #34200: (secumod) Fix parted module set CLI example (refs: #34222)
  - 05a4785c8c Merge pull request #34222 from cachedout/lint_34200
  - eadf80a56f Linted #34200
  - 2cd0433f8d Fix parted module set CLI example
- **PR** #34197: (eliasp) Make *module.ssh.recv_known_host()* more resilient against hosts not returning a key @ *2016-06-22 17:26:02 UTC*
  - 0cbdb73fc5 Merge pull request #34197 from eliasp/2016.3-salt.modules.ssh.recv_known_host-empty_results
  - 82c4b1229e Make *module.ssh.recv_known_host()* more resilient against hosts not returning a key
- **ISSUE** #34199: (DarkKnightCZ) cmdmod.exec_all doesn't work with Windows PowerShell (refs: #34201)
- **ISSUE** #34196: (DarkKnightCZ) Salt call cmdmod.exec_code_all fails on Windows minion due to invalid file mode (refs: #34198)

- **PR** #34201: (DarkKnightCZ) Suffix temp file with .sr1 and add mandatory argument when executing PowerShell script @ *2016-06-22 17:21:24 UTC*
  - **PR** #34198: (DarkKnightCZ) Don't use binary mode for cmdmod.exec_code (refs: #34201)
  - 606ae3c886 Merge pull request #34201 from DarkKnightCZ/cmdmod-34199
  - 05748743bc Suffix temp file with .sr1 and add -File argument when executing PowerShell code via cmdmod.exec_code
- **ISSUE** #34196: (DarkKnightCZ) Salt call cmdmod.exec_code_all fails on Windows minion due to invalid file mode (refs: #34198)
- **PR** #34198: (DarkKnightCZ) Don't use binary mode for cmdmod.exec_code (refs: #34201) @ *2016-06-22 17:14:06 UTC*
  - cb704b780b Merge pull request #34198 from DarkKnightCZ/cmdmod-34196
  - 04553cd3de Don't use binary mode for cmdmod.exec_code
- **PR** #34172: (dmurphy18) Support for building with local packages on Debian and Ubuntu @ *2016-06-22 16:36:44 UTC*
  - 0578a2f87d Merge pull request #34172 from dmurphy18/debbuild_deps
  - f7f8a5d33f Fixed pylint issues
  - 82fa276141 Support for building with local packages on Debian and Ubuntu
  - **PR** #34194: (vutny) Correct the docstrings formatting in pkgbuild modules and state
  - **PR** #34056: (vutny) Make rpmbuild module work on non-RPM based GNU/Linux systems (refs: #34194)
- **PR** #34186: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-06-21 23:26:34 UTC*
  - a8429c2595 Merge pull request #34186 from rallytime/merge-2016.3
  - 318c2ed6a1 Merge branch '2015.8' into '2016.3'
  - 1c4369d093 Merge pull request #34184 from rallytime/merge-2015.8
    - 8e36e90966 Merge branch '2015.5' into '2015.8'
    - 5411ebb3b4 Merge pull request #34141 from jtand/boto_vpc_test_fix
      - b7ac6c735a Moved imports to top, out of _get_moto_version function
      - 02f9ba99ba Updated version check. Moved check into it's own function
      - d445026c56 Updated test to work with new moto version. Changed strings to unicode
  - c059d6c08c Merge pull request #34176 from rallytime/bp-34103
    - 2e5e7ed03c Fix diskusage beacon
  - 5cbaaed167 Merge pull request #34179 from terminalmage/issue34114
    - 86d1b8e864 Raise the correct exception when gitfs lockfile is empty
  - 67deded119 Merge pull request #34178 from terminalmage/remove-comment
    - 4965be72b1 Remove unnecesssary comment
  - 6387d1636e fix salt –summary to count not responding minions correctly (#34165)
  - e5949ea6f1 doc: add missing dot (#34175)
  - 47595d6795 Typo fix (#34174)
  - 3669048654 Merge pull request #34077 from rallytime/grains-tests

    - 2199bb8a78 Add integration tests for grains.append
    - 37cfe70724 Add some grains targeting tests

- **ISSUE** #34162: (ryanwalder) salt-call default loglevel regression (refs: #34173)
- **PR** #34173: (rallytime) Update docs to match log_level default @ *2016-06-21 17:15:53 UTC*
  - 3413c494bd Merge pull request #34173 from rallytime/fix-34162
  - f577681f0b Update docs to match log_level warning default
- **ISSUE** #34094: (avandendorpe) cron.file is broken (refs: #34095)
- **PR** #34095: (rallytime) Back-port #32396 to 2016.3 @ *2016-06-21 16:12:39 UTC*
  - **PR** #32396: (eradman) Unbreak cron.file (refs: #34095)
  - c596bf5744 Merge pull request #34095 from rallytime/bp-32396
  - 074b6ab5c2 Correct pylint error
  - 20ff5c879a Unbreak cron.file
- **PR** #34108: (l2ol33rt) Make dockerng.absent state honor test=true @ *2016-06-21 15:55:29 UTC*
  - b98687875f Merge pull request #34108 from l2ol33rt/docker_absent_dryrun
  - 5598cb4a21 Make docker.absent honor test=true
- **ISSUE** #34012: (viq) States mount.* fail on OpenBSD's tmpfs (refs: #34133)
- **PR** #34133: (rallytime) Back-port #34057 to 2016.3 @ *2016-06-21 15:53:46 UTC*
  - **PR** #34057: (ajacoutot) _active_mounts_openbsd: unbreak output for special filesystems (refs: #34133)
  - a75386a669 Merge pull request #34133 from rallytime/bp-34057
  - f7be5e182b _active_mounts_openbsd: unbreak output for special filesystems
- **PR** #34156: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-06-21 15:52:59 UTC*
  - dd989dac78 Merge pull request #34156 from rallytime/merge-2016.3
  - b061b86946 Merge branch '2015.8' into '2016.3'
    - 65fba5b4d7 Merge pull request #34142 from isbm/isbm-getid-loglevel-shift
      - 236a67b702 Move log message from INFO to DEBUG.
    - 79a719b719 Update documentation on "refresh" behavior in pkg states (#34100)
    - 6d0d52fa86 modules.pkg int tests: skip refresh_db upon error (#34072)
- **PR** #34110: (garethgreenaway) Fixes to git module & state module related to identity file @ *2016-06-21 15:52:17 UTC*
  - b302cb03ef Merge pull request #34110 from garethgreenaway/git_needs_saltenv_for_identity
  - 68092cdc8c When specifying the SSH identity to use with Git as a salt URL, eg. salt://files/identity, if that file exists outside of the default base environment the file won't be accessible so we need to include the saltenv.
- **ISSUE** #34120: (rmohta) Correct package name to systemd-python for RHEL 7 in docs.saltstack.com (refs: #34138)
- **ISSUE** #31402: (vutny) [repo] systemd-python required package is missing from RHEL7 archive (refs: #34138)
- **PR** #34138: (rallytime) Update package dep note to systemd-python for RHEL7 install @ *2016-06-21 15:51:24 UTC*

  - 6c3405755a Merge pull request #34138 from rallytime/fix-34120
  - 73f3e12ce6 Update package dep note to systemd-python for RHEL7 install
  - **PR** #34166: (vutny) Fix YAML indentation in Apache state docstrings
  - **PR** #34098: (terminalmage) Restore old refresh logic
  - **PR** #34087: (bbinet) Encourage to report issues to upstream PillarStack project
- **PR** #34075: (jfindlay) modules.inspectlib.kiwiproc: import gate lxml @ *2016-06-17 15:36:08 UTC*
  - 9da592a297 Merge pull request #34075 from jfindlay/import_xml
  - f882a72348 modules.inspectlib.kiwiproc: import gate lxml
- **PR** #34056: (vutny) Make rpmbuild module work on non-RPM based GNU/Linux systems (refs: #34194) @ *2016-06-17 15:14:51 UTC*
  - 52b852216a Merge pull request #34056 from vutny/rpmbuild-support-debian
  - 8ff36d4f2b Expose virtual pkgbuild module as rpmbuild on non-RPM based systems if all required utilities are in place
  - 758f5cd77c Make rpmbuild module work on Debian GNU/Linux and derivatives
- **PR** #34073: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-06-16 23:02:32 UTC*
  - f2a2c2bf53 Merge pull request #34073 from rallytime/merge-2016.3
  - f6bfaede21 Merge branch '2015.8' into '2016.3'
  - 1b76de1557 Merge pull request #34069 from rallytime/test-minion-return-message
    - 60561ac6fc Add a test to check for disconnected minion messaging
  - 3119693dac Merge pull request #34048 from terminalmage/issue30100
    - 715e7af8a4 Ensure only one fileserver update in a masterless run
  - dd03024931 Merge pull request #34011 from rallytime/bp-33948-2015.8
    - a4660d1ff7 Warn when custom returners don't have minions kwarg in save_load
    - 78befde62f Add note to release notes about returner minions kwarg change
    - 4e7f35fa36 Fix loop over cache in auth checking!
    - 06963e0505 Save an entire minion cache traversal on each master pub
  - bca437128e Fixed a bug in the consul.py module that was preventing services (#34051)
  - 8ba117c7f6 Merge pull request #34045 from jacobhammons/release-prev
    - 43b4a12aa2 Updated latest release version
  - f9bfcde61f Always make changes to minion config if set (#34020)
  - e25dba49e2 More YAML indentation fixes in state module examples (#34030)
  - 5b5eae4ca9 Merge pull request #34018 from rallytime/merge-2015.8
    - 77f44f3087 Merge branch '2015.5' into '2015.8'
    - 871f7966ce Lint fix for #34000 (#34005)
    - f758e42172 Fix incorrectly written test (#34000)
    - cf6281b4cf Add loader.utils() example to calling minion_mods (#33953)
    - 6b98e8a9ea Merge pull request #33880 from terminalmage/zh744

      - ea726d11c8 pkg.uptodate: Pass kwargs to pkg.list_upgrades
      - de90b35d2b salt/modules/zypper.py: add fromrepo support to list_upgrades
      - 35fbb06df5 salt/modules/win_pkg.py: add kwargs to list_upgrades
      - bf5505f425 salt/modules/solarisips.py: add kwargs to list_upgrades
      - 6e89a8be98 salt/modules/pkgutil.py: add kwargs to list_upgrades
      - 5179dbcec4 salt/modules/pacman.py: add kwargs to list_upgrades
      - 46e5a52784 salt/modules/macports.py: add kwargs to list_upgrades
      - 76143b76ca salt/modules/ebuild.py: add kwargs to list_upgrades
      - b40fc9bc62 salt/modules/brew.py: add kwargs to list_upgrades
      - 4f11c16d86 salt/modules/aptpkg.py: add fromrepo support to list_upgrades
    - cb88960ed1 Merge pull request #33904 from rallytime/bp-33806
      - 638ccf501d Work around upstream cherrypy bug
  - 7d940aed1f states.file: fix indentation in YAML examples (#34003)
  - 4c7fac0aaa Remove loader test for pam module (#34002)
  - c4dab6a074 Merge pull request #33990 from jacobhammons/community-projects
    - b20213fd79 Adds links to several current Salt-related projects Removes the salt_projects.rst file which hasn't been updated in a long time, this is replaced by the updated topics/projects/index.rst file Adds a note about Salt Pack to the installation doc
  - 444c15792c Merge pull request #33983 from twangboy/fix_docs_join_domain
    - b057be04b4 Fix typo, more documentation
    - d8c2f3e57a Clarify the *account_exists* parameter
  - 9bd2317992 Merge pull request #33951 from jfindlay/gem_tests
    - 2eb633ccad modules.gem int tests: only check known installed gems
    - 9f3e18b037 modules.gem int tests: (un)install a non-core gem
  - 53baae6eb1 Merge pull request #33984 from jfindlay/disk_capacity
    - 6cbe31e6c2 states.disk: rewrite unit tests
    - 82c77b533f states.disk.status: validate percent values
    - aedc4e15e5 states.disk: add documentation
  - fa5efb6a69 Merge pull request #33985 from rallytime/more-batch-tests
    - 3e7ab8c7b3 Write some more simple batch command tests
  - 6080846cce acl.ClientACL: add unit tests (#33684)
- **ISSUE** #33831: (astehlik) file.managed state should not download a file if the checksum did not change (refs: #34010)
- **PR** #34010: (terminalmage) Do not cache remote files if they are already cached @ *2016-06-16 21:03:47 UTC*
  - 790384f413 Merge pull request #34010 from terminalmage/issue33831
  - 636d081ae0 Do not cache remote files if they are already cached
- **PR** #34009: (rallytime) Back-port #33948 to 2016.3 + add log message (refs: #34011) @ *2016-06-16 21:01:09 UTC*

  - **PR** #33948: (cachedout) Save an entire minion cache traversal on each master pub (refs: #34011, #34009)
  - dd26d6fd74 Merge pull request #34009 from rallytime/bp-33948
  - 239af9ae5e Warn when custom returners don't have minions kwarg in save_load
  - c776d2d795 Add note to release notes about returner minions kwarg change
  - 5f696082e3 Fix loop over cache in auth checking!
  - 180c312715 Save an entire minion cache traversal on each master pub
- **ISSUE** #33927: (phil123456) Salt - windows minion cannot do anything (refs: #33941)
- **PR** #33941: (cachedout) Don't call os.getppid() on Windows @ *2016-06-16 20:56:17 UTC*
  - 5f4ef46d2f Merge pull request #33941 from cachedout/issue_33927
  - 5fe889c7f1 Don't call os.getppid() on Windows
- **PR** #34067: (jacobhammons) Fixes doc refresh bug on chrome mobile. @ *2016-06-16 18:44:12 UTC*
  - fa253aa62b Merge pull request #34067 from jacobhammons/mobile-fix
  - ce027fd769 Fixes doc refresh bug on chrome mobile.
  - **PR** #34050: (rallytime) Back-port #34026 to 2016.3
  - **PR** #34026: (bensherman) removed method that doesn't exist (refs: #34050)
- **PR** #33987: (isbm) inspectlib cleanup @ *2016-06-15 22:09:31 UTC*
  - 73ff11585e Merge pull request #33987 from isbm/isbm-inspectlib-cleanup
  - e36821510f Fix documentation: add an example how to export system to the Kiwi
  - fe300ccf73 Lintfix
  - 96423076b1 Add unit test for file tree
  - 8975036b27 Add get_unmanaged_files test
  - be5f12fcaf Add initial unit test for inspectlib.collector.Inspector
  - 652c96d7e7 Stop build (not implemented yet)
  - 58e85ea0ab Refactor class caller
  - 878f67674a Sort package names
  - c31818b4aa Fix lint: PEP8 multiplication of 4.
  - c87fff3680 Add root-only warning when exporting system with Kiwi
  - 9bd80f02fc Implement users Kiwi export
  - e191f338c7 Cleanup code
  - 80f45defae Implement packages and patterns gathering
  - ad45a265f5 Add Debian support for the repo generator
  - 6280ad137e Semifix: sometimes SQLite3 is locked. TODO: a proper handling required.
  - 51567ab61d Implement SUSE repositories export
  - e4ac113927 Add Kiwi support to the collector/inspector
  - eceeb4ecf2 Add ability to specify an additional PID file
  - f522a91ac6 Add ISO/image build (stub) and export to the Kiwi

  - bb19684606 Add Kiwi processor exception
  - 805e2ce204 Add Kiwi exported (initial)
  - a52f9f7107 Add default configuration
- **ISSUE** #34038: (Ch3LL) user.list_users does not work on smartos (refs: #34042)
  - **PR** #34042: (sjorge) fix #34038
- **PR** #34025: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-06-15 19:41:00 UTC*
  - f546a00dc9 Merge pull request #34025 from rallytime/merge-2016.3
- **PR** #34044: (jacobhammons) Updated latest release to 2016.3.1 @ *2016-06-15 19:20:28 UTC*
  - 3035520594 Merge pull request #34044 from jacobhammons/3.1
  - a4b67fd1e9 Updated latest release to 2016.3.1 Clean up installation instructions code-block type updates Add link to jinja tutorial
  - **PR** #34014: (jnhmcknight) fix launch config creation params
  - **PR** #34021: (twangboy) Always make changes to minion config if set (2016.3)
- **PR** #34031: (eliasp) *states.postgres_privileges* expects a real list, not a comma-separated string @ *2016-06-15 16:34:04 UTC*
  - 5f90717fd3 Merge pull request #34031 from eliasp/2016.3-salt.states.postgres_privileges-doc-priv-list
  - d3198ea538 *states.postgres_privileges* expects a real list, not a comma-separated string for *privileges*
- **ISSUE** #33023: (cmclaughlin) rest_cherrypy eauth can't handle some characters (refs: #33995)
- **ISSUE** #23522: (nbirnel) Update the best practices documentation to include simpler examples of of jinja dictionaries (refs: #33995)
- **ISSUE** #12470: (whiteinge) Document how to (and not NOT to) use Jinja in states (refs: #33995)
- **ISSUE** #10480: (gravyboat) Create documentation that talks about using Jinja specifically for Salt. (refs: #33995)
- **ISSUE** #10206: (rabits) Jinja import: Jinja variable 'salt' is undefined (refs: #33995)
- **PR** #33995: (jacobhammons) Understanding Jinja topic, Jinja doc issues. @ *2016-06-14 02:00:29 UTC*
  - 1132bc5d0b Merge pull request #33995 from jacobhammons/doc-fixes
  - 887a415138 Adds new Understanding Jinja topic, and fixes several Jinja doc issues. Removes the "Full list of builtin …" from each module reference list, leaving just the module type for scanability.
- **PR** #33900: (amendlik) Document sudo policy for gitfs post-recieve hook @ *2016-06-14 01:04:35 UTC*
  - a400f6a6c3 Merge pull request #33900 from amendlik/gitfs-hook-doc
  - b4a28e2684 Add clarifying documentation about the need for sudo in the git hook
  - 1046279cb7 Document sudo policy for gitfs post-recieve hook
- **PR** #33980: (twangboy) Use full path to python.exe @ *2016-06-14 00:46:14 UTC*
  - 28c886edd0 Merge pull request #33980 from twangboy/fix_build
  - dd7d55afb9 Use full path to python.exe
- **PR** #33993: (s0undt3ch) Call *sys.exit()* instead of *exit()* @ *2016-06-14 00:30:46 UTC*
  - 26fee377ec Merge pull request #33993 from s0undt3ch/2016.3
  - 34f7d90d9f Call *sys.exit()* instead of *exit()*

- **PR** #33976: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-06-13 19:29:40 UTC*
  - 2e934cffef Merge pull request #33976 from rallytime/merge-2016.3
  - 19d49d94f2 Merge branch '2015.8' into '2016.3'
    - a74f1b8077 ZD 762 (#33942)
    - 0281d491c6 Merge pull request #33946 from rallytime/bp-33698
      - 5fdfed1cb9 Make sure we only use GetConnection if we are using a proxy salt minion
      - 1505c5724b Fix a bug with self signed certificates and creating a new VM
    - dff3f51955 Merge pull request #33952 from rallytime/fix-33911
      - 03b7cbbd2c Add base argument to salt-ssh grains wrapper for filter_by func
    - 4a8064918a Adds a "Generated on <timestamp>" line to the footer of each doc html page in the doc (#33962)
- **ISSUE** #33868: (abalashov) Returner configuration override options don't work for scheduled jobs (schedule module) (refs: #33912)
- **PR** #33912: (abalashov) utils/schedule.py:handle_func() - Fix for accessing returner configur… @ *2016-06-13 17:18:04 UTC*
  - 8d8ed59b85 Merge pull request #33912 from abalashov/abalashov/fix-schedule-returner-config
  - b5a4f8b313 utils/schedule.py:handle_func() - Fix for accessing returner configuration attributes 'return_config' and 'return_kwargs'.
- **PR** #33945: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-06-13 15:44:30 UTC*
  - 81e16bb93f Merge pull request #33945 from rallytime/merge-2016.3
  - b4ab322ce1 Merge branch '2015.8' into '2016.3'
    - b3ec39d644 Correct issue with ping on rotate with minion cache (#33765)
    - 378dd7ca06 Merge pull request #33888 from jfindlay/random_check
      - 6acee3cc30 modules.random_org._query: only return text if present
      - 82f95429db modules.random_org unit tests: skip if random.org down
      - 1f9422e0cd utils.http.query: also except gaierror with tornado
    - 2dc1914e7c Add connecting_settings to boto_elb state attributes list (#33936)
    - 91a2184f2d Wait for up to a minute for sync_after_install (#33917)
    - ef6da0be5d Merge pull request #33877 from rallytime/merge-2015.8
    - 398534a9e7 Fix ret return from merge-conflict resolution
    - b8e4706074 Merge branch '2015.5' into '2015.8'
    - cdda593c50 Merge pull request #33829 from terminalmage/update-versionchanged
      - f7028eb1c6 Update versionchanged directive
    - b8e6c144d8 Merge pull request #33833 from terminalmage/issue33645
      - 91745c2a67 Support syncing pillar modules to masterless minions
    - e061788e81 Merge pull request #33814 from terminalmage/archive-extracted-xz
      - 897a716df2 Support extraction of XZ archives in archive.extracted state

        - fa983e91cf Merge pull request #33778 from sodium-chloride/2015.5-2016-0604-1938
            - a5fb6d7a69 Fix minor docstring issues
        - b9133326c8 Merge pull request #33726 from jtand/sysmod_skip_valid_docs_glance
            - ebee8a89af glance.warn_until shouldn't be checked for a doc string
        - 137f0b19f3 Merge pull request #33611 from TargetHolding/2015.5
        - 1dd15a603b solve' TypeError: expected string or buffer' in json/decoder.py
        - eaf42ca892 solve AttributeError: 'module' object has no attribute 'exception'

- **PR** #33960: (nulfox) Fix mongo get_load to return full mongo record instead of non-existant 'load' key @ *2016-06-13 15:37:46 UTC*
    - 68d261fe5b Merge pull request #33960 from mecarus/2016.3
    - d622133a49 The jid load comes in directly, not as 'load' key. Should return the mongo record directly without accessing keys
- **PR** #33961: (jacobhammons) 2016.3.0 known issues update @ *2016-06-13 02:59:21 UTC*
    - 8f56406507 Merge pull request #33961 from jacobhammons/release
    - 2cf787d4ba 2016.3.0 known issues update
    - **PR** #33908: (ticosax) [boto_lambda] handle ommitted Permissions parameter
- **ISSUE** #33575: (anlutro) File states seem slower in 2016.3, especially on first cache retrieval (refs: #33896)
- **ISSUE** #29643: (matthayes) Can't get batch mode and –failhard to work as expected (refs: #31164)
- **ISSUE** #28569: (andrejohansson) Reactor alert on highstate fail (refs: #31164)
- **PR** #33896: (DmitryKuzmenko) Don't deep copy context dict values. @ *2016-06-10 15:32:54 UTC*
    - **PR** #31164: (DmitryKuzmenko) Issues/29643 fix invalid retcode (refs: #33896)
    - 16b5e9dcc1 Merge pull request #33896 from DSRCompany/issues/33575_do_not_deep_copy_context
    - 8e34d0a9c3 Don't deep copy context dict values.
- **ISSUE** #3077: (torhve) Client ACL and external auth system should have support for limiting functions to certain arguments (refs: #29153)
- **PR** #33905: (rallytime) Back-port #33847 to 2016.3 @ *2016-06-10 15:22:34 UTC*
    - **PR** #33847: (whiteinge) Add docs for arg/kwarg eauth matching (refs: #33905)
    - **PR** #29153: (DmitryKuzmenko) ACL limit args (refs: #33847)
    - 01323322b0 Merge pull request #33905 from rallytime/bp-33847
        - b6ebd7b6ef Add docs for arg/kwarg eauth matching
    - 261baeb5b5 Ensure tht pillar have freshest grains (#33910)
    - 00e016ecfc Add note about Xenial packages to 2016.3.0 release notes (#33870)

28.2.44 Salt 2016.3.3 Release Notes

Version 2016.3.3 is a bugfix release for *2016.3.0*.

Statistics

- Total Merges: **108**
- Total Issue References: **26**
- Total PR References: **115**
- Contributors: **36** (The-Loeki, abednarik, cachedout, cro, deniszh, dkruger, dmurphy18, eliasp, farcaller, galet, gtmanfred, hu-dabao, isbm, jacobhammons, jacobweinstock, jfindlay, justinta, kstreee, lubyou, markuskramerIgitt, meaksh, miihael, mzupan, nishigori, rallytime, s0undt3ch, skizunov, tankywoo, terminalmage, thatch45, theredcat, ticosax, tonybaloney, twangboy, vutny, whiteinge)

Known Issues

issue #36055: Salt Cloud events (`salt/cloud`) are not generated on the master event bus when provisioning cloud systems.

Bootstrap Issue #973: `python-futures` is not installed when installing from a git tag on RedHat-based distributions. `Python futures` is needed when

running Salt with the TCP transport. This is fixed on the `develop` branch of the salt-bootstrap repo and the fix will be included in the upcoming release of salt-bootstrap, but is a bug in the bootstrap release that ships with this version of Salt. Please see the salt-bootstrap repo for more information on how to update your bootstrap version.

Changelog for v2016.3.2..v2016.3.3

Generated at: 2018-05-27 04:47:36 UTC

- **PR** #35603: (rallytime) Make sure version label is correct in header
- **PR** #35602: (rallytime) Update release notes for 2016.3.3
- **ISSUE** #35102: (TheBigBear) Exception raised when processing __virtual__ function for mac_system - (mac os x installation relies on un-installed 'mac_service_helper.sh') (refs: #35580)
- **PR** #35580: (twangboy) Fix mac_service attempts to parse non-plist files @ *2016-08-19 09:24:38 UTC*
  - 9683bb3c58 Merge pull request #35580 from twangboy/fix_35102
  - 4122e66ed5 Handle malformed plist files
  - 52feff9309 Fix mac_service attempts to parse non-plist files
- **PR** #35586: (hu-dabao) Fix 35420, add run_on_start in build_schedule_item @ *2016-08-19 09:23:32 UTC*
  - c4ec94d6e8 Merge pull request #35586 from hu-dabao/fix-35420
  - 2d3a882cc2 fix 35420, add run_on_start in build_schedule_item, remove redundancy of enabled
  - **PR** #35583: (terminalmage) Fix localemod tests
- **PR** #35579: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-08-18 22:00:41 UTC*
  - d1339fd9f5 Merge pull request #35579 from rallytime/merge-2016.3
  - 00dff9dcbd Merge branch '2015.8' into '2016.3'
  - 26a7f7d9f7 Merge pull request #35577 from terminalmage/unit-file-changes
    - 6cb0fb47f3 pkg/salt-syndic.service: change Type to notify
    - 175ba99e0e pkg/salt-minion.service: remove KillMode, change Type to notify

-
  -
    - 540ec28954 pkg/salt-master.service: remove KillMode
    - 69fad464ab pkg/salt-api.service: change Type to notify
- **PR** #35571: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-08-18 19:50:03 UTC*
  - f7a18234db Merge pull request #35571 from rallytime/merge-2016.3
  - 2930df924e Update localemod_test systemd.sd_booted mock to use salt.utils.systemd.booted
  - e61b04a707 Merge branch '2015.8' into '2016.3'
    - 2a5d1a0eee fix-35384, fix the logic caused by wrong indent (#35566)
    - feb852f8c0 Clarify config.get docstring (#35492)
    - 205d8e2e7b Merge pull request #35483 from gtmanfred/2015.8
      - 2d8ec1e9db use __opts__ in salt.utils.cloud for cache functions
    - 70fa2d0901 Merge pull request #35546 from whiteinge/salt-api-eauth-fail-gracefully
      - eb3574adae Don't fail hard if the user's permissions cannot be found
      - ec597bd54c Change groups check in token to look for truthy values
    - 61fec6caa9 add missing glob import (#35525)
    - 0e3f2fc6cb Whitespace fix for 2015.8 (#35540)
    - fd3274c800 Merge pull request #35510 from terminalmage/issue33516
      - 5b5f19d269 Update zypper unit test to reflect call to config.get
      - 2730edb516 Add note about systemd-run usage in package states
      - e2d9e87e10 salt/modules/systemd.py: Use systemd-run –scope where needed
      - 22919a25bc Notify systemd on salt-api start
      - a40b3f8a08 Notify systemd on syndic start
      - e847d3af30 Notify systemd on minion start
      - d648887afc salt/modules/zypper.py: Use systemd-run –scope where needed
      - 2e17976722 salt/modules/yumpkg.py: Use systemd-run –scope where needed
      - 86b59c1e74 salt/modules/pacman.py: Use systemd-run –scope where needed
      - e32d92c6d5 salt/modules/ebuild.py: Use systemd-run –scope where needed
      - c7d21d3ae3 salt/modules/aptpkg.py: Use systemd-run –scope where needed
      - f83e0ef242 Add unit tests for salt.utils.systemd
      - 5b12f030c6 Add func to salt.utils.systemd to tell if scopes are available
  - **PR** #35573: (rallytime) Back-port #33337 to 2016.3
  - **PR** #33337: (mzupan) adding the () to make changes work (refs: #35573)
- **PR** #35572: (terminalmage) Fix poor formatting in pkg state docs @ *2016-08-18 18:15:52 UTC*
  - 73b549ed00 Merge pull request #35572 from terminalmage/docs
  - 7d7a7de9e6 Fix poor formatting in pkg state docs
  - **PR** #35545: (hu-dabao) fix-35384, fix cmd.run unless (refs: #35566)
  - **PR** saltstack/salt#35463: (skizunov) Make *auth_timeout* user configurable again (refs: #35489)

- **PR** #35489: (rallytime) Back-port #35463 to 2016.3 @ *2016-08-18 07:16:03 UTC*
  - **PR** #35463: (skizunov) Make *auth_timeout* user configurable again (refs: #35489)
  - f2eb3dc105 Merge pull request #35489 from rallytime/bp-35463
  - bbf7ce121b Remove final self.MINION_CONNECT_TIMEOUT ref
  - cf2e2daab9 Make *auth_timeout* user configurable again
  - **PR** #35538: (thatch45) Treat python XML as an optdep
- **PR** #35526: (thatch45) Always deploy the thin to /var/tmp @ *2016-08-17 19:44:26 UTC*
  - e2bd575461 Merge pull request #35526 from thatch45/ssh_W_tmp
  - a381f02cfe Always deploy the thin to /var/tmp
- **PR** #35522: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-08-17 18:07:16 UTC*
  - 8b770869e4 Merge pull request #35522 from rallytime/merge-2016.3
  - ff212d8976 Whitespace fix
  - c305d8d99b Merge branch '2015.8' into '2016.3'
  - b3b28cb760 Might be a good idea to be able to download the software we make (#35513)
  - 9f87081cef Merge pull request #35302 from Ch3LL/add_job_cache_test
    - ccb2a5cadf remove unused imports
    - 512ae81dfd remove TMP and add integration.TMP
    - c9b7c3cf80 need to add returners option in other places
    - 7316df7a02 fix pylint
    - 50a4f0fe6a fix comment
    - 6837acf742 add job cache integration tests
  - 1c82c6bee5 Merge pull request #35512 from cachedout/fixup_35419
    - 253662541a Fix import
    - f16a30786b Fixes consul.agent_service_register which was broken for registering service checks.
  - e1a373fa4c Merge pull request #35497 from deepakhj/2015.8
    - 685db4ab88 Fix spacing
  - 4048255ed6 Merge pull request #35508 from terminalmage/update-docstring
    - 67c945fce0 Add Carbon to versionadded for git.diff
- **PR** #35516: (rallytime) Back-port #34441 to 2016.3 @ *2016-08-17 15:47:23 UTC*
  - **PR** #34441: (markuskramerIgitt) Copy and delete silently, do not list each file (refs: #35516)
  - e86a39a115 Merge pull request #35516 from rallytime/bp-34441
  - e47c661cb0 Copy and delete silently, do not list each file
  - **PR** saltstack/salt#34502: (markuskramerIgitt) Windows installer build scripts will exit on error (refs: #35517)
- **PR** #35517: (rallytime) Back-port #34502 to 2016.3 @ *2016-08-17 15:47:10 UTC*
  - **PR** #34502: (markuskramerIgitt) Windows installer build scripts will exit on error (refs: #35517)

  - 45080d9860 Merge pull request #35517 from rallytime/bp-34502
  - 32da48df08 setup.py will not print each individual file
  - 698a076a39 Completely remove Python and verify
  - 7406bd22a6 Errors will stop the scripts
- **PR** #35429: (tankywoo) Fix iptables target options with no arguments @ *2016-08-17 10:05:17 UTC*
  - c1deb945d7 Merge pull request #35429 from tankywoo/fix-iptables-target-options
  - 914eb60d51 Fix iptables target options with no arguments
- **ISSUE** #35458: (iggy) SALT.STATES.APACHE_MODULE needs version annotations (refs: #35495)
- **PR** #35495: (rallytime) Use correct deprecated notation instead of a warning for apache_module.enable state function. @ *2016-08-17 09:36:40 UTC*
  - 678759ba6c Merge pull request #35495 from rallytime/fix-35458
  - 9bae3d09a6 Use correct deprecated notation instead of a warning.
- **ISSUE** #35336: (Sylvain303) documentation state.file.managed parameter template not reflecting TEMPLATE_REGISTRY (refs: #35360, #35498, #35406, #saltstack/salt‘#35360‘_)
  - **PR** saltstack/salt#35360: (rallytime) Add all template registery templates to file.managed docs (refs: #35406)
- **PR** #35498: (rallytime) Add supported templates list to all template doc references in file state @ *2016-08-17 09:33:36 UTC*
  - **PR** #35406: (rallytime) Provide links to the renderers in the template docs (refs: #35498)
  - **PR** #35360: (rallytime) Add all template registery templates to file.managed docs (refs: #35498)
  - 5bd44b10a7 Merge pull request #35498 from rallytime/file-state-docs
  - 6190b2d738 Add supported templates list to all template doc references in file state
- **PR** #35487: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-08-16 18:36:21 UTC*
  - 6df4648765 Merge pull request #35487 from rallytime/merge-2016.3
  - c6c82be1de Merge branch '2015.8' into '2016.3'
  - bfe7107a87 Update bootstrap script to latest stable (2016.08.16) (#35486)
  - 240fc12863 Merge pull request #35413 from cachedout/issue_35296
    - fb8a12d677 Fix silly error
    - 3646cf1afa Additional checks on master and integration test
    - 09efde7634 Splat the list into os.path.join
    - fc0d5878bc Set file_recv on test master
    - 81c4d136c5 Transition file push paths to lists
  - c3319b2a8b Merge pull request #35476 from cachedout/issue_35380
    - c05fcf33d1 Fixup SSH bug where sudo without sudo user would break
  - 004778c966 Merge pull request #35471 from terminalmage/issue34479
    - e243c63e43 win_pkg: Fix traceback when package is not installed
  - 5c9428c32d Merge pull request #35448 from isbm/isbm-zypper-106-fix

    - dd82e6a848 Add ignore_repo_failure option to suppress zypper's exit code 106 on unavailable repos
  - 1473474b04 Merge pull request #35451 from isbm/isbm-zypper-mod_repo-unchanged
    - 8790197d86 Fix Unit test for suppressing the exception removal on non-modified repos
    - 3f00c6997a Remove zypper's raise exception if mod_repo has no arguments and/or no changes
  - a8c4f17f50 Merge pull request #35453 from theothergraham/fix_CacheDisk
    - ae5b233d51 fixes #34279
  - d8c35b5260 Merge pull request #35459 from thatch45/shim_fix
    - 10037b00cb Some environments refuse to return the command output
  - 38b60a32e5 [2015.8] Update bootstrap script to latest stable (2016.08.15) (#35460)
- **ISSUE** #34161: (bobrik) Salt command can hang forever because of one broken minion (refs: #35446)
  - **PR** #35446: (cachedout) Make salt-client aware of edge-case where saltutil might be broken
- **ISSUE** #35422: (ViaviSolutions) aptpkg.py: install_recommends: True does not force "–install-recommends" (refs: #35449)
- **PR** #35449: (dkruger) aptpkg will specify –install-recommends if enabled by the SLS @ *2016-08-16 01:38:56 UTC*
  - f90ecbb15e Merge pull request #35449 from dkruger/fix-35422
  - f54bf445b5 aptpkg will specify –install-recommends if enabled by the SLS
- **ISSUE** #33367: (supertom) [salt-cloud] libcloud >= 1.0.0 incompatible regarding node_state (refs: #33518)
- **PR** #35467: (rallytime) Back-port #33518 to 2016.3 @ *2016-08-16 01:17:01 UTC*
  - **PR** #35235: (rallytime) Back-port #33518 to 2016.3 (refs: #35467)
  - **PR** #33518: (tonybaloney) Fix libcloud bug #33367 (refs: #35235, #35467)
  - d2dd78e25b Merge pull request #35467 from rallytime/bp-33518
  - e427815caf fix clrf
  - be41a400fa commit fix
  - 06530b5461 add a test to check existing functionality is broken
  - **PR** #35461: (rallytime) [2016.3] Update bootstrap script to latest stable (2016.08.15)
- **PR** #35456: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-08-15 19:16:23 UTC*
  - 9b2c075611 Merge pull request #35456 from rallytime/merge-2016.3
  - 6a86a12294 Merge branch '2015.8' into '2016.3'
  - d75005c519 Fix cp.push_dir pushing empty dirs (#35442)
  - 09925ba353 Minor doc fixup (#35436)
  - a0b128a85a Merge pull request #35132 from sjorge/2015.8-35121
    - 5cb38c8ae0 switch to fpread().splitlines(), as per @lorengordon suggestion
    - 634f1dded5 fixes #35121, causing lots of mayham (onchange) with 2016.3.2 for me
  - **PR** saltstack/salt#35308: (farcaller) Actually fixed dockerng.list_tags (refs: #35447)
  - **PR** saltstack/salt#34702: (farcaller) Fixed dockerng.list_tags (refs: #35447)

  - **PR** #35447: (ticosax) [dockerng] RepoTags can be also be None with docker 1.12
  - **PR** #34702: (farcaller) Fixed dockerng.list_tags (refs: #‘saltstack/salt#35308‘_)
- **ISSUE** saltstack/salt#35403: (randomed) Setting ext_job_cache breaks on salt-master (refs: #35427)
  - **PR** #35427: (cachedout) Correct errant call to argspec from master. Fix ext_job_cache.
- **ISSUE** #35423: (Ch3LL) Stacktrace when running state.sls against an sls does not exist (refs: #35428)
  - **PR** #35428: (cachedout) Resolve stacktrace logged by highstate outputter if sls cannot be found
- **PR** #35412: (s0undt3ch) Only allow one sync read to happen at a time. @ *2016-08-12 23:57:29 UTC*
  - 607169a01b Merge pull request #35412 from s0undt3ch/2016.3
  - f54b3cc514 Only allow one sync read to happen at a time.
- **ISSUE** #35336: (Sylvain303) documentation state.file.managed parameter template not reflecting TEMPLATE_REGISTRY (refs: #35360, #35498, #35406, #saltstack/salt‘#35360‘_)
  - **PR** saltstack/salt#35360: (rallytime) Add all template registery templates to file.managed docs (refs: #35406)
  - **PR** #35406: (rallytime) Provide links to the renderers in the template docs (refs: #35498)
- **PR** #35393: (deniszh) No need to run ddns update every time @ *2016-08-12 12:40:36 UTC*
  - b3e9e98b40 Merge pull request #35393 from deniszh/2016.3_fix35350
  - 6f2f080f4a No need to run dns update every time
- **PR** #35407: (hu-dabao) [Fix-35094] None will not be added to grains which generate [none] @ *2016-08-12 12:34:05 UTC*
  - a5fe05b7f9 Merge pull request #35407 from hu-dabao/fix-35094
  - a23108f795 None will not be added to grains which generate [none]
- **PR** #35411: (eliasp) modules.event.send(): Prevent backtrace for masterless Minions @ *2016-08-12 12:29:02 UTC*
  - 4dc776ffbf Merge pull request #35411 from eliasp/2016.3-modules.event-handle-file_client-opt
  - 8d7244bdd9 modules.event.send(): Also check for *file_client* and *use_master_when_local* opts
- **PR** #35395: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-08-11 20:59:37 UTC*
  - c032506e6b Merge pull request #35395 from rallytime/merge-2016.3
  - 0d1aa630f1 Lint fix
  - cf038ee3fe Merge branch ’2015.8’ into ’2016.3’
  - d9c20c0456 Update freebsd.rst (#35394)
  - a375dd7e1f Clean up open filehandles (#35359)
  - 9ea7a34c30 Merge pull request #35339 from isbm/isbm-2015.8-minion-importerror-fix
    - 12af60b7be Fix continuous minion restart if a dependency wasn’t installed
  - fd9b05ace4 Merge pull request #35357 from twangboy/file.recurse.clean.2015.8
    - d328ec0157 Fix file.recurse with clean: True
- **ISSUE** #35226: (mathieubouchard) Do not throw an exception when an invalid requisite is set (refs: #35373)
  - **PR** #35373: (cachedout) Raise SaltRenderError on bad requisite
- **PR** #35352: (twangboy) Fix file.recurse with clean: True on Windows (2016.3) @ *2016-08-11 00:46:11 UTC*

  - 72f3548671 Merge pull request #35352 from twangboy/file.recurse.clean
  - ecad616d08 Fix file.recurse with clean: True
- **PR** #35356: (jfindlay) document log levels and warn on all logging below info @ *2016-08-11 00:45:56 UTC*
  - 0fcfc70cc8 Merge pull request #35356 from jfindlay/log_levels
  - 2fc3a55338 utils.verify.verify_log: warn at all levels less than info
  - 72a3f18a2e log.setup: minor optimization
  - 66332510c6 doc.ref.configuration.logging: document log levels
  - 93616eff3e doc.ref.configuration.logging: fix formatting
  - 472a2d31de doc.ref.configuration.logging: cleanup formatting
- **PR** #35358: (twangboy) Update libsodium deps @ *2016-08-11 00:36:30 UTC*
  - 2f7be03053 Merge pull request #35358 from twangboy/update_libsodium_deps
  - d120a8906f Add vcredist 14 dlls
- **ISSUE** #35336: (Sylvain303) documentation state.file.managed parameter template not reflecting TEMPLATE_REGISTRY (refs: #35360, #35498, #35406, #saltstack/salt‘#35360‘_)
- **PR** #35360: (rallytime) Add all template registery templates to file.managed docs (refs: #35498) @ *2016-08-11 00:35:20 UTC*
  - f9e03b9c59 Merge pull request #35360 from rallytime/fix-35336
  - 30badb5402 Add all template registery templates to file.managed docs
- **ISSUE** #24745: (The-Loeki) RFC: disk versus blockdev (refs: #24893)
  - **PR** saltstack/salt#25267: (jfindlay) Disk module improvements (refs: #35361)
- **PR** #35362: (rallytime) Correct deprecation version tags @ *2016-08-11 00:34:38 UTC*
  - **PR** #35361: (rallytime) Blockdev deprecations (refs: #35362)
  - **PR** #25267: (jfindlay) Disk module improvements (refs: #35362)
  - **PR** #24893: (The-Loeki) Contribution: Disk module improvements (refs: #‘saltstack/salt‘#25267‘_‘_, #25267)
  - 3c628d3cbc Merge pull request #35362 from rallytime/correct-deprecated-tag
  - 507827a014 Correct deprecation version tags
- **PR** #35347: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-08-10 20:07:42 UTC*
  - 87e29188c0 Merge pull request #35347 from rallytime/merge-2016.3
  - a651962e95 Merge branch ’2015.8’ into ’2016.3’
  - 4618b433e9 Merge pull request #35323 from thatch45/ssh_crazy
    - 8a5b47b5d7 Collect all error data from the wfuncs call
    - 11864c31b7 supress a stack trace to show clean ssh error
    - 9fbfa282fa wow this solves an issue!
  - cfae862972 Merge pull request #35325 from kev009/fbsd-netstat-route
    - 0d49dd3c29 Fix fbsd netstat route on fbsd 10+
  - 244c3bd495 Pass port to ssh.check_known_host, closes #35264 (#35301)

  - 243909f39d file.recurse: Do not convert octal mode string to int (#35309)
  - **PR** #35334: (cachedout) Restore random_master functionality
- **PR** #35331: (hu-dabao) fix 35165, salt-run jobs.exit_success jid is broken @ *2016-08-10 11:50:10 UTC*
  - 78dfd18ec6 Merge pull request #35331 from hu-dabao/fix-35165
  - 4dcce18d01 fix 35165, salt-run jobs.exit_success jid is broken
- **PR** #35318: (rallytime) Remove legacy compat docs in mysql pillar since the code was removed already @ *2016-08-10 11:34:48 UTC*
  - **PR** #30913: (justinta) Deprecated code removed. (refs: #35318)
  - fcca0b9333 Merge pull request #35318 from rallytime/remove-deprecation-docs
  - 75f205e485 Remove legacy compat docs in mysql pillar since the code was removed already
- **PR** #35329: (hu-dabao) sys.doc will skip all not connected minions @ *2016-08-10 11:18:22 UTC*
  - 3446dc9ec6 Merge pull request #35329 from hu-dabao/fix-tiny-salt-cli
  - 4b806a70ea sys.doc will skip all not connected minions
- **PR** #35306: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-08-09 18:50:44 UTC*
  - 31f7d307a7 Merge pull request #35306 from rallytime/merge-2016.3
  - 2d3eadfe49 Merge branch '2015.8' into '2016.3'
  - 2efc1b333b Merge pull request #35290 from terminalmage/issue35051
    - d621aa7b61 Update runner/wheel unit tests to reflect new key in ret dict
    - 90c12a9c7b Add __orchestration__ key to orch returns for runner/wheel funcs
    - 7b8c3b86e7 Suppress error about invalid changes data for orchestration jobs
    - 54a1704d6c Suppress event for wheel/runner funcs executed from orchestration
    - f409f62bf2 Accept print_event option in WheelClient.cmd()
    - b42b25ccce Add cmd func for RunnerClient
    - 480065fe00 Add print_event option to client mixins
- **PR** #35229: (lubyou) Ignore import error for pwd module in mac_shadow @ *2016-08-09 15:48:16 UTC*
  - 94529d0578 Merge pull request #35229 from lubyou/fix-mac_shadow
  - b45039c240 Do not blindly ignore import failures
  - c1d5670b79 Ignore import error for pwd module
- **PR** #35227: (isbm) Isbm osfinger ubuntu fix @ *2016-08-09 15:38:31 UTC*
  - ce7aeb6ca4 Merge pull request #35227 from isbm/isbm-osfinger-ubuntu-fix
  - fe5da97283 Lintfix: E8303
  - 6eea62d4ec Add a deprecation warning
  - 4dc45f2509 Add grains unit test for Ubuntu systems
  - 3904e4b81c Bugfix: Ubuntu osfinger should contain also minor version
  - a69f97f9ad Bugfix: use oscodename if lsb_distrib_codename key exists empty.
- **PR** #35286: (hu-dabao) fix 34425, a bug that sys.doc cannot output format @ *2016-08-09 09:50:12 UTC*

  - 47e328f755 Merge pull request #35286 from hu-dabao/fix-34425
  - 86fb359f58 fix 34425, a bug that sys.doc cannot output format
- **ISSUE** #27294: (stenstad) salt-cloud should support Openstack Identitiy v3 for authentication (refs: #35213)
- **PR** #35275: (rallytime) Back-port #35213 to 2016.3 @ *2016-08-09 00:02:43 UTC*
  - **PR** #35213: (gtmanfred) add identity v3 support to openstack driver (refs: #35275)
  - d79cb1b4ec Merge pull request #35275 from rallytime/bp-35213
  - 9b9fc508cc add identity v3 support to openstack driver
  - **PR** #35278: (dmurphy18) Increase timeout for siging to 10 seconds when signing rpm packages
- **PR** #35276: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-08-08 18:20:29 UTC*
  - 959a00e4b7 Merge pull request #35276 from rallytime/merge-2016.3
  - 2b4c156df1 Merge branch '2015.8' into '2016.3'
  - f8158124d5 Merge pull request #35211 from cachedout/issue_31074
    - 6f53232e6d Better error handling and a workaround for group mismatch.
    - 5b56a4acf7 Docs
    - ae04e7aaeb Initial POC
  - 3e4eb13daa Merge pull request #35271 from bobrik/default-output-profile
    - 6cdee21036 Default state_output_profile to True everywhere, closes #35166
  - 673e1aa1aa Merge pull request #35233 from terminalmage/issue32719
    - 730a077041 Do not attempt to get fqdn_ip{4,6} grains when ipv{4,6} grains are empty
  - cdf3c0fe73 Merge pull request #35202 from multani/fix/test-doc
    - 1642dba5d1 doc: fix broken links in the test documentation page
  - e1331cd2a3 Merge pull request #35236 from rallytime/bp-35119
    - 9ade78de7b Revise unnecessary code duplication
    - 7c15f5b20a Fix formatting
    - 64f93f8938 Assume two EVRs are equal if E and V are equal but one R is missing.
  - 4f2b8aa5b6 Merge pull request #35240 from derekmaciel/bp-35225
    - 9ed47f713a Add missing documentation for pkg.installed
  - 4bcfaa97d0 Merge pull request #35241 from terminalmage/gitfs-fixes
    - e05648cc2d Break from loop when file is found
    - 6764a88601 Ensure that failed recursion results in no blob object being returned
  - f6d7360e0b Merge pull request #35245 from rallytime/bp-35039
    - 51ab9cd6d4 Add saltenv support to module.run
  - d65a5c7134 Merge pull request #35249 from terminalmage/issue35214
    - bcd5129e9f Fix regression in git.latest when update is fast-forward
    - e2e8bbbfde Add integration test for #35214
- **ISSUE** #35003: (edgan) rabbitmq_user.present broken on Ubuntu 16.04 Xenial (refs: #35232)

- **ISSUE** #34481: (L4rS6) rabbitmq_user.present with password keyword throws exception (refs: #35232)
- **ISSUE** #33588: (whytewolf) rabbitmq_user.present error (refs: #35232)
- **PR** #35274: (rallytime) Lint fixes for 2016.3 branch @ *2016-08-08 16:45:41 UTC*
  - **PR** #35232: (theredcat) fix rabbitmq version detection using a package-agnostic version (refs: #35274)
  - 157939d5b0 Merge pull request #35274 from rallytime/lint-2016.3
  - 0d3d711e9c Lint fixes for 2016.3 branch
- **PR** #35269: (meaksh) Checksum validation for zypper pkg.download in 2016.3 and develop @ *2016-08-08 14:45:16 UTC*
  - c58bb18624 Merge pull request #35269 from meaksh/checksum-during-zypper-pkg-download-for-2016.3-and-develop
  - 18700e821e unit tests for rpm.checksum() and zypper.download()
  - c3f29ab205 checksum validation during zypper pkg.download
- **PR** #35197: (vutny) Make *pkgbuild.repo* state recognize *createrepo* command return code @ *2016-08-06 23:20:47 UTC*
  - d3f2ce2a1a Merge pull request #35197 from vutny/pkgbuild-repo-failure-detection
  - a5f6630e97 Make *pkgbuild.repo* state recognize *createrepo* command return code
- **ISSUE** #34446: (mirceaulinic) Proxy minions & straight minion using the same caching directory (refs: #35178)
- **PR** #35178: (cro) Add append_minionid_config_dirs option @ *2016-08-06 22:21:14 UTC*
  - f004b831d2 Merge pull request #35178 from cro/proxy_cache_fix2
  - 84cc7d67c0 Add documentation for append_minionid_config_dirs.
  - f0961e741e Merge with 2016.3
- **ISSUE** #35234: (Sylvain303) Bug: module disk.wipe dont wipe the filesystem information (refs: #35253)
- **PR** #35259: (cachedout) Fixup 35253 @ *2016-08-06 21:59:48 UTC*
  - **PR** #35253: (abednarik) Fix disk.wipe missing option. (refs: #35259)
  - 6eb1c48469 Merge pull request #35259 from cachedout/fixup_35253
  - 104116f464 Add release notes and include entry about disk.wipe fix
  - 6714e8f386 Fix mock call in disk wipe test
- **ISSUE** #35234: (Sylvain303) Bug: module disk.wipe dont wipe the filesystem information (refs: #35253)
- **PR** #35253: (abednarik) Fix disk.wipe missing option. (refs: #35259) @ *2016-08-06 21:55:01 UTC*
  - 4e7d7f8e4c Merge pull request #35253 from abednarik/disk_wipe_fix
  - ff33df4ba1 Fix disk.wipe missing option.
- **PR** #35206: (hu-dabao) Make the log level back to warning for unclassified exc @ *2016-08-06 21:40:38 UTC*
  - eeede82109 Merge pull request #35206 from hu-dabao/fix-exc-log
  - 676be7d711 Make the log level back to warning for unclassified exc
- **PR** #35196: (isbm) Deprecate status.uptime one version later @ *2016-08-06 08:39:40 UTC*
  - 21808e27d5 Merge pull request #35196 from isbm/isbm-too-fast-uptime-deprecation
  - 6f3a32dace Deprecate status.uptime one version later

- **PR** #35207: (eliasp) Handle exceptions in *_get_virtual()* and in *_get_virtual()* consumers @ *2016-08-06 08:29:08 UTC*
  - 100645e557 Merge pull request #35207 from eliasp/2016.3-modules.aptpkg-handle-exceptions
  - 2f11df98ca Handle exceptions in *_get_virtual()* and in *_get_virtual()* consumers
- **ISSUE** #35003: (edgan) rabbitmq_user.present broken on Ubuntu 16.04 Xenial (refs: #35232)
- **ISSUE** #34481: (L4rS6) rabbitmq_user.present with password keyword throws exception (refs: #35232)
- **ISSUE** #33588: (whytewolf) rabbitmq_user.present error (refs: #35232)
- **PR** #35232: (theredcat) fix rabbitmq version detection using a package-agnostic version (refs: #35274) @ *2016-08-06 08:13:02 UTC*
  - 7302a8a6e5 Merge pull request #35232 from theredcat/fix-rabbitmq-version-detection
  - f75eb2ecc7 Fix runas in code order and make the check_password work with the new >3.5.7 version
  - 4d8119b88b fix rabbitmq version detection using a package-agnostic version
- **PR** #35244: (rallytime) Back-port #31677 to 2016.3 @ *2016-08-06 07:53:28 UTC*
  - **PR** #31677: (miihael) Return correct value for services that must be enabled in Systemd (refs: #35244)
  - 2e9fa3799c Merge pull request #35244 from rallytime/bp-31677
  - 45d563d5ac Return correct value for services that must be enabled in Systemd, not in SysV
- **PR** #35182: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-08-03 20:57:29 UTC*
  - bd0496eef5 Merge pull request #35182 from rallytime/merge-2016.3
  - c35974f78f Merge branch '2015.8' into '2016.3'
  - 67d8dd0fd0 Don't discard running beacons config when listing becaons (#35174)
  - 3754550dd2 Add missing CLI Examples to aws_sqs module funcs (#35173)
  - 4967ed275f doc version update to 2015.8.11, updates to release notes (#35145)
- **PR** #35150: (rallytime) Start release notes for 2016.3.3 @ *2016-08-03 13:46:31 UTC*
  - f9f92ad326 Merge pull request #35150 from rallytime/2016.3.3-release-notes
  - a64026fc99 Start release notes for 2016.3.3
- **PR** #35157: (hu-dabao) master returned from func should be a string as designed so far @ *2016-08-03 13:29:16 UTC*
  - 518ecf897a Merge pull request #35157 from hu-dabao/func-return-string
  - a7506af4c9 master returned from func should be a string as designed so far
  - **PR** #35147: (jacobhammons) doc version updated to 2016.3.2
- **PR** #35136: (s0undt3ch) Don't restart processes if the manager is not set to restart them @ *2016-08-02 18:40:05 UTC*
  - dc7d7db3d5 Merge pull request #35136 from s0undt3ch/2016.3
  - 7b8bf2d2b4 Don't restart processes if the manager is not set to restart them
- **PR** #35133: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-08-02 18:06:18 UTC*
  - bf04bd3316 Merge pull request #35133 from rallytime/merge-2016.3
  - 0d5f0b6016 Merge branch '2015.8' into '2016.3'

  - 81845ee31d Merge pull request #35114 from terminalmage/git_pillar-env-remap-docs
    - 5951554e9f Add clarification docs on a common git_pillar misconfiguration
  - 88a9fb1b31 Merge pull request #34768 from hrumph/bad-installed-state
    - e1fcb8311d Put pkg.latest_version in try/except structure Move refreshed or refresh to different spot (just for code tidyness)
    - e0b6261659 changed name of varibale 'refreshed' to 'was_refreshed'
    - 340110b4b4 Move check for rtag to outermost-nesting in function
    - ac67c6b493 Lint fix
    - 0435a1375e Get rid of repetition in code by using new "refreshed" variable instead
    - 3b1dc978e2 Lint fix
    - a9bd1b92b9 lint fixes
    - 71d69343ef Fixes #34767
  - 343576408f Merge pull request #35043 from rallytime/new-release-notes
    - bdcc81a384 Start release notes file for 2015.8.12
- **PR** #35120: (kstreee) The '_handle_event_socket_recv' function in Salt Api is missing first data of stream. @ *2016-08-02 16:22:50 UTC*
  - dd91006ed7 Merge pull request #35120 from kstreee/fix-missing-first-stream-data
  - 28f793caac Fix missing first data in stream when subscribing stream using a function 'read_async'.
  - **PR** saltstack/salt#35011: (nishigori) Fix docstring for code-block of rst (refs: #35131)
  - **PR** #35131: (rallytime) Back-port #35011 to 2016.3
  - **PR** #35011: (nishigori) Fix docstring for code-block of rst (refs: #35131)
- **PR** #35110: (hu-dabao) Do not return job status back to master for master_alive and master_failback schedules @ *2016-08-02 07:49:46 UTC*
  - 77b1f43b0d Merge pull request #35110 from hu-dabao/master-check-lighter
  - 3a3b66e27d dont return job status back to master for master_alive and master_failback schedules
- **PR** #35104: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-08-01 18:56:43 UTC*
  - 94a983f129 Merge pull request #35104 from rallytime/merge-2016.3
  - dda2c32325 Merge branch '2015.8' into '2016.3'
    - 848bf0272f Merge pull request #35050 from terminalmage/fix-saltdev-arg
      - 40cfa7cf17 Avoid needlessly running 2 argspecs in salt.utils.format_call()
      - fd186b7e4c Pass environment as 'saltdev' if runner/wheel func accepts a saltdev argument
      - 951b52ab93 Pass __env__ from saltmod orch states to to saltutil.{runner,wheel}
    - 2144178ae0 Merge pull request #35066 from jfindlay/postgres_log
      - c2c442234f returners.postgres_local_cache: do not log in __virtual__
    - 7121618142 Merge pull request #35024 from bobrik/daemon-reload-fix
      - c300615e9d Cache systemd unit update check per unit, closes #34927
    - 865c29f126 Expressly deny a minion if a key cannot be found instead of raising stacktrace (#35026)

- **ISSUE** #32761: (notpeter) Ubuntu 16.04 Xenial Xerus Support (refs: #‘saltstack/salt#33870‘_)
  - **PR** saltstack/salt#33870: (rallytime) Add note about Xenial packages to 2016.3.0 release notes (refs: #35105)
- **PR** #35105: (rallytime) Update 2016.3.0 release notes with repo.saltstack.com Xenial pkg availability @ *2016-08-01 17:26:55 UTC*
  - 6c056a829e Merge pull request #35105 from rallytime/update-2016.3.0-release-notes
  - fbaff3e98e Update 2016.3.0 release notes with repo.saltstack.com Xenial pkg availability
- **PR** #35059: (vutny) Add *fun_args* field to events generated by execution of Master modules @ *2016-08-01 13:01:42 UTC*
  - 1f8a0fd1e7 Merge pull request #35059 from vutny/event-function-args
  - 19d080445b Add *fun_args* field to events generated by execution of Master modules
- **PR** #34955: (lubyou) force dism to always output english text @ *2016-08-01 12:54:03 UTC*
  - d137c4b986 Merge pull request #34955 from lubyou/fix-dism-on-non-english-systems
  - 63c974a3d0 add missing comma
  - 775ea73578 fix unit tests
  - 51869807f1 force dism to always output english text
- **PR** #35078: (jacobweinstock) added missing non-keyword argument skip_verify to __get_artifact func… @ *2016-08-01 12:22:47 UTC*
  - ff7ddf0b68 Merge pull request #35078 from jacobweinstock/fix-missing-non-keyword-argument
  - c40314ba80 added missing non-keyword argument skip_verify to __get_artifact function
- **PR** #35008: (hu-dabao) Fix multimaster failover on more than two masters and failback behaviour @ *2016-07-29 16:34:37 UTC*
  - 878e200cd9 Merge pull request #35008 from hu-dabao/fix-multimaster
  - 12da890910 Fix multimaster failover on more than two masters and failback behaviour
- **ISSUE** saltstack/salt#33536: (murzick) pkgrepo.managed does not disable a yum repo with ”disabled: True” (refs: #35055)
- **ISSUE** #33536: (murzick) pkgrepo.managed does not disable a yum repo with ”disabled: True” (refs: #35055)
- **PR** #35055: (galet) #33536 pkgrepo.managed does not disable a yum repo with ”disabled: True” @ *2016-07-29 15:40:15 UTC*
  - 11ed147448 Merge pull request #35055 from galet/2016.3
  - d70796bbfe #33536 pkgrepo.managed does not disable a yum repo with ”disabled: True”
- **PR** #35039: (whiteinge) Add saltenv support to module.run (refs: #35245) @ *2016-07-29 14:01:03 UTC*
  - ebaee39b2b Merge pull request #35039 from whiteinge/module-run-saltenv
  - 7ef287e09e Add saltenv support to module.run
- **PR** #35046: (eliasp) Prevent backtrace in *salt.states.network* @ *2016-07-29 13:59:09 UTC*
  - 32ed78a399 Merge pull request #35046 from eliasp/2016.3-salt.states.network-prevent-backtrace
  - 1542cd5124 Prevent backtrace in *salt.states.network*
- **PR** #35054: (lubyou) Only fail user lookup is the user parameter is required @ *2016-07-29 13:58:41 UTC*

  - f34bb7a8de Merge pull request #35054 from lubyou/fix-win_dacl-disable_inheritance
  - 1e4e856fb2 Only fail user lookup is the user parameter is required
- **PR** #35029: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-07-28 18:01:35 UTC*
  - bee303db51 Merge pull request #35029 from rallytime/merge-2016.3
  - 65f9365ee0 Merge branch '2015.8' into '2016.3'
  - 2b511f3013 Merge pull request #35000 from rallytime/bp-33875
    - 35696ad637 Pylint fix
    - f9fd6ddd8a Fixup #33875
    - 56b1f6c651 Fix naive fileserver map diff algorithm
  - 837bc6ba7d Merge pull request #34994 from rallytime/bp-34835
    - 9268a793de same thing for the mine in salt-ssh
    - 3e11e19714 Fix the mine in salt ssh
  - **PR** #35021: (terminalmage) Don't add '.' to strerror when passed string ends in ? or !
  - **PR** #34983: (eliasp) modules.slack.post_message: Allow sending messages to direct-message …
- **PR** #34996: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-07-27 19:03:04 UTC*
  - 05cfb1cefc Merge pull request #34996 from rallytime/merge-2016.3
  - a7d4f93697 Merge branch '2015.8' into '2016.3'
    - b58c663d8d Merge pull request #34991 from cachedout/ssh_timeout
      - 39cd8da399 Lint diff against salt-testing
      - 443e5cdde2 Add timeout to ssh tests
    - 5f8370ae8d Refine errors in client (#34976)
    - a83cdf9339 Merge pull request #34831 from thatch45/recoverssh
    - fa73041a49 If the thin does not match, then redeploy, don't error
  - **PR** #34987: (eliasp) salt.states.slack: check correct result attribute
- **ISSUE** saltstack/salt#34526: (danielmotaleite) salt-ssh + mine = weird error (refs: #34835, #saltstack/salt'#34835'_)
- **PR** #34835: (thatch45) Make the mine and publish combine minion and master opts in salt-ssh (refs: #34994) @ *2016-07-27 17:25:26 UTC*
  - edeb0eda36 Merge pull request #34835 from thatch45/34526_
  - 1d2477df05 same thing for the mine in salt-ssh
  - 6b6c5ff278 Fix the mine in salt ssh

28.2.45 Salt 2016.3.4 Release Notes

Version 2016.3.4 is a bugfix release for *2016.3.0*.

Statistics

- Total Merges: **275**
- Total Issue References: **119**
- Total PR References: **374**
- Contributors: **80** (BenoitKnecht, Ch3LL, DavidWittman, DmitryKuzmenko, Jlin317, Kimamisa, UtahDave, aaronm-cloudtek, abednarik, ahammond, alertedsnake, alexander-bauer, amontalban, basepi, bl4ckcontact, bx2, cachedout, clarkperkins, clinta, cro, damon-atkins, danlsgiga, darkalia, dmurphy18, do3meli, edhgoose, efficks, eliasp, eradman, fix7, galet, goestin, gtmanfred, hrumph, hu-dabao, isbm, jackywu, jacobhammons, jbonachera, jf, jfindlay, jizhilong, justinta, kstreee, l2ol33rt, lomeroe, lorengordon, maximeguillet, meaksh, mikeadamz, mirceaulinic, morganwillcock, mrproper, multani, nvtkaszpir, oba11, onorua, opdude, orymate, oz123, pass-by-value, pbdeuchler, rallytime, roosri, silenius, skizunov, slinn0, stanislavb, swiftgist, techhat, terminalmage, thatch45, theredcat, ticosax, twangboy, vutny, whiteinge, xbglowx, xiaoanyunfei, yhekma)

Known Issues

The Salt Minion does not clean up files in `/tmp` when rendering templates. This potentially results in either running out of disk space or running out of inodes. Please see issue #37541 for more information. This bug was fixed with PR #37540, which will be available in the `2016.3.5` release of Salt.

The release of the `bootstrap-salt.sh` script that is included with 2016.3.4 release has a bug in it that fails to install salt correctly for git installs using tags in the 2015.5 branch. This bug has not been fixed in the salt-bootstrap repository yet, but the previous bootstrap release (v2016.08.16) does not contain this bug.

Changes

- The *disk.wipe* execution module function has been modified so that it correctly wipes a disk.
- Add ability to clone from a snapshot to the VMWare salt-cloud driver.
- Add ability to specify disk backing mode in the VMWare salt cloud profile.

Changelog for v2016.3.3..v2016.3.4

Generated at: 2018-05-27 04:56:54 UTC

- **PR** #37285: (rallytime) Update 2016.3.4 release notes
- **ISSUE** #37281: (frogunder) 2016.3.4: Raet Transport not working (refs: #37282)
- **PR** #37282: (thatch45) add cpub to raet event for compat @ *2016-10-27 21:33:48 UTC*
  - 3b62a89e45 Merge pull request #37282 from thatch45/raet_cpub
  - 90f778dbc1 Add func for compat with main event system
  - 8e52f425e4 add cpub to raet event for compat
  - **PR** #37278: (jfindlay) update 2016.3.4 release notes
- **PR** #37252: (vutny) Set logging level to 'info' for message about init system detection @ *2016-10-27 06:15:01 UTC*
  - d0ce3de50c Merge pull request #37252 from vutny/suppress-init-grain-error
  - 3f20cc01ed Set logging level to 'info' for message about init system detection

  - **PR** #37259: (rallytime) [2016.3] Update man pages for the 2016.3 branch
- **PR** #37257: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-10-26 17:32:36 UTC*
  - 2087cfce90 Merge pull request #37257 from rallytime/merge-2016.3
  - 2066f10d7b Merge branch '2015.8' into '2016.3'
    - f49ccdf30f Merge pull request #37234 from rallytime/bp-37167
      - a7dbb5bfc9 fixes pkgrepo for fedora>22 saltstack/salt#31240
- **ISSUE** #37238: (cmclaughlin) Restarting master causes minion to hang (refs: #37254)
- **ISSUE** #37192: (Ch3LL) 2016.3.4: Windows minion does not kill process running in foreground (refs: #37254)
- **ISSUE** #37191: (Ch3LL) 2016.3.4: Multi-Master Failover minion hangs when masters not running (refs: #37254)
- **ISSUE** #35480: (jelenak) 200 processes of salt-master (2016.3.2) (refs: #36184, #36555, #37254)
- **PR** #37254: (DmitryKuzmenko) Bugs/37191 minion hangs @ *2016-10-26 16:28:41 UTC*
  - ea6155c3f4 Merge pull request #37254 from DSRCorporation/bugs/37191_minion_hangs
  - 9ee24b2d70 Revert "Don't set the *daemon* flag for LoggingQueue process."
- **ISSUE** #37187: (darkalia) Supervisord is considered as "systemd" in grains (refs: #37218)
- **PR** #37218: (darkalia) Issue #37187 Do not parse first /proc/1/cmdline binary if it's not *b… @ *2016-10-26 01:41:03 UTC*
  - d1a6bb72ac Merge pull request #37218 from darkalia/37187_supervisor_2016.3
  - a8dfc6bb96 Issue #37187 Do not parse first /proc/1/cmdline binary if it's not *bin/init and set supervisord
- **PR** #37239: (Ch3LL) Fix cloud tests timeout @ *2016-10-26 01:11:52 UTC*
  - 760ed9f56d Merge pull request #37239 from Ch3LL/fix_cloud_timeout
  - 394fccf556 fix run_cloud timeout
  - 23947c5944 change timeout for cloud tests
- **PR** #37244: (rallytime) Update bootstrap release to 2016.10.25 @ *2016-10-26 00:46:29 UTC*
  - 6c5f619398 Merge pull request #37244 from rallytime/update-bootstrap
  - f728a5bc7b Update bootstrap release to 2016.10.25
  - **PR** saltstack/salt#36334: (pass-by-value) Add ability to specify disk backing mode for VMware cloud profile (refs: #37245)
- **PR** #37245: (rallytime) Back-port #36334 to 2016.3 @ *2016-10-26 00:41:00 UTC*
  - **PR** #36334: (pass-by-value) Add ability to specify disk backing mode for VMware cloud profile (refs: #37245)
  - bb7caf8c42 Merge pull request #37245 from rallytime/bp-36334
  - f64ca3c442 Update release notes and version added
  - 0a3d266d6b Add ability to specify disk backing mode for VMware cloud profile
- **ISSUE** #37132: (bl4ckcontact) Incorrect flag defined for disabling AD computer account in win_system.py (refs: #37154)
- **PR** #37233: (rallytime) Back-port #37154 to 2016.3 @ *2016-10-25 18:32:56 UTC*
  - **PR** #37154: (bl4ckcontact) modules.win_system.py: Fix flag disabling AD Computer objects (refs: #37233)

  - 3c94315d35 Merge pull request #37233 from rallytime/bp-37154
  - 849af162f1 modules.win_system.py: Fix flag disabling AD Computer objects
- **PR** #37232: (rallytime) Back-port #37153 to 2016.3 @ *2016-10-25 18:32:41 UTC*
  - **PR** #37153: (eradman) Update configuration examples for Joyent (refs: #37232)
  - 94852f2eb1 Merge pull request #37232 from rallytime/bp-37153
  - 3829b7592f Update configuration examples for Joyent
- **PR** #37228: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-10-25 18:22:57 UTC*
  - a913eed92a Merge pull request #37228 from rallytime/merge-2016.3
  - b99d6733b6 Merge branch '2015.8' into '2016.3'
  - d608465d77 Merge pull request #37178 from isbm/isbm-fix-saltapi-ssh-crash
    - 44da411c3a Do not prematurily raise an exception, let the main loop take care of it instead
    - ee48deeded Do not restart the whole thing if roster is not around
    - b8f4e46920 Fix PEP8
- **PR** #37213: (cachedout) More salttesting fixes @ *2016-10-25 07:53:33 UTC*
  - 6aaf6bf399 Merge pull request #37213 from cachedout/more_salttesting_fixes
  - 0bbf06bd86 Lint fix
  - f609917760 Workaround for utils
  - a6a24c2b3b Workaround for tornado test startup error
  - 88bcfa2c0a Fix TCP test
- **ISSUE** #37194: (sjorge) function_cache in modules.mine docs? (refs: #37207)
- **PR** #37207: (cachedout) Correct documentation for mine_functions @ *2016-10-25 07:25:09 UTC*
  - b448455c31 Merge pull request #37207 from cachedout/issue_37194
  - 9fcdf6da94 Correct documentation for mine_functions
- **ISSUE** #37182: (Ch3LL) 2016.3.4: multi-master minion stack trace when killed with ctrl+c (refs: #37208)
- **PR** #37208: (cachedout) Give multimion a process manager and its own destroy method @ *2016-10-25 07:24:52 UTC*
  - a5e1c041cc Merge pull request #37208 from cachedout/issue_37182
  - 1449770b0b Give multimion a process manager and its own destroy method
- **PR** #37206: (cachedout) Address transport test hang @ *2016-10-25 05:25:55 UTC*
  - e19ee88b6b Merge pull request #37206 from cachedout/transport_test_hang
  - c4393d5e9e Address transport test hang
- **PR** #37179: (isbm) Fix Salt-API ssh crash (2016.3) @ *2016-10-25 04:52:19 UTC*
  - 6737fd3ad9 Merge pull request #37179 from isbm/isbm-fix-saltapi-ssh-crash-2016-3
  - 28edda457e Do not prematurily raise an exception, let the main loop take care of it instead
  - 372f2bbd93 Do not restart the whole thing if roster is not around
  - 8d1450cc47 Fix PEP8

- **ISSUE** saltstack/salt#37176: (guettli) docs for "load tags" explains "import_yaml" (refs: #37183)
- **PR** #37183: (gtmanfred) load tags should reference the actual load tags @ *2016-10-25 04:38:00 UTC*
  - 815dfd1c04 Merge pull request #37183 from gtmanfred/2016.3
  - 1b7b4b1a0c load tags should reference the actual load tags
- **PR** #37188: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-10-25 04:37:26 UTC*
  - ca63376c97 Merge pull request #37188 from rallytime/merge-2016.3
  - ccb664050d Merge branch '2015.8' into '2016.3'
  - b3e79dcd51 Merge pull request #37139 from awerner/fix-spm-download-remote-download
    - a606a42575 Minor style change
    - e3916813bb Download spm package from remote repository and save it to cache directory
  - 35b4494157 Merge pull request #37162 from rallytime/bp-36823
    - 3032a542d9 Use NotifyAccess=all in all unit files
    - 4826995973 Remove EnvironmentFile and Restart lines from unit files
    - 3be15694d2 Use Type=notify for debian systemd units
    - d58fda6f67 Use control-group default for killmode
  - **PR** #37186: (rallytime) Pylint fix for 2016.3
  - **PR** #37175: (cachedout) Fix test hang (refs: #37186)
- **PR** #37175: (cachedout) Fix test hang (refs: #37186) @ *2016-10-24 09:55:37 UTC*
  - 0d7af935e5 Merge pull request #37175 from cachedout/fix_test_hange
  - 0fecb5ff2e Remove sleep. Thanks @s0undt3ch
  - cedc609503 Fix test suite hang on salt testing
- **ISSUE** #36866: (sjorge) [2016.11.0rc1] salt-master <> salt-minion communication borken due to master_alive_interval (refs: #37144, #37117, #37142)
- **PR** #37144: (DmitryKuzmenko) Bugs/36866 salt minion communication broken 2016.3 @ *2016-10-24 03:19:06 UTC*
  - **PR** #37142: (DmitryKuzmenko) status.master: don't fail if host_to_ips returns None (refs: #37144)
  - **PR** #37117: (DmitryKuzmenko) Updated host_to_ip to return all the IPs instead of the first one. (refs: #37144, #37142)
  - 334313ec64 Merge pull request #37144 from DSRCorporation/bugs/36866_salt-minion_communication_broken_2016.3
  - 87c2e93e40 Don't fail if host_to_ips returns None.
  - f625e6d3a9 Updated host_to_ip to return all the IPs instead of the first one.
- **PR** #37158: (jfindlay) add mock for *status.uptime* unit test (refs: #37157) @ *2016-10-24 03:13:53 UTC*
  - **PR** #37157: (jfindlay) Implement *status.uptime* on macOS (refs: #37158)
  - c5d81a8ade Merge pull request #37158 from jfindlay/mac_skip_uptime
  - 094eac06eb modules.status.uptime unit test: mock on linux
- **ISSUE** #37037: (mikeadamz) schedule state always reports changed when running in highstate (refs: #37098)

- **PR** #37161: (rallytime) Back-port #37098 to 2016.3 @ *2016-10-24 03:13:14 UTC*
  - **PR** #37098: (mikeadamz) Add run_on_start to SCHEDULE_CONF (refs: #37161)
  - e51f90b459 Merge pull request #37161 from rallytime/bp-37098
  - 36bc2a1ded Add run_on_start to SCHEDULE_CONF
- **PR** #37159: (rallytime) Back-port #37107 to 2016.3 @ *2016-10-22 13:55:47 UTC*
  - **PR** #37107: (do3meli) use versionadded and deprecated warnings in apache_module (refs: #37159)
  - b5025c044e Merge pull request #37159 from rallytime/bp-37107
  - c63126a2f0 removed trailing whitespaces in apache_module.py
  - a812cbfea7 use versionadded and deprecated warnings in apache_module
- **PR** #37163: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-10-22 13:52:37 UTC*
  - 1e520b3f48 Merge pull request #37163 from rallytime/merge-2016.3
  - 8fff95b3b4 Merge branch '2015.8' into '2016.3'
  - a5335a2f15 Merge pull request #37137 from awerner/fix-spm-msgpack
    - 52d47cece9 SPM-METADATA are now loaded as yaml from remote URLs
  - 8c46d69251 Merge pull request #37109 from meaksh/zypper-distupgrade-support-2015.8
    - 330f830c9b Disables 'novendorchange' for old SLEs versions
    - 01b0a6917c Minor pylint fixes
    - 7dbb0bd252 Unit tests fixes
    - e89982b6d2 Improves 'dryrun' outputting. Setting 'novendorchange' as not supported for SLE11
    - c5a34cbadf Adds multiple repositories support to 'fromrepo' parameter
  - 38fdd28962 Merge pull request #37087 from vutny/gpg-fix-short-keyid
    - c589cba8a9 salt.modules.gpg: allow getting keys by short key ID
  - 3a37a22366 Merge pull request #37088 from meaksh/zypper-distupgrade-support-2015.8
    - c0641a4027 Fix in log message
    - a092a974da Refactor: Cleanup and pylint fixes
    - 1331ae5c72 Unit tests for zypper upgrade and dist-upgrade
    - 4bcfef2ba2 Adding 'dist-upgrade' support to zypper module
  - 2f29e9e956 Merge pull request #37090 from zer0def/silence-prereq-supervisord-warnings
    - 6a4bfbb485 Silence warnings about "__prerequired__" being an invalid kwarg when using *prereq*. (no refs)
  - **PR** #37150: (rallytime) Allow the minion test daemons a couple of tries to connect to the master
  - **PR** #37152: (rallytime) Add note about salt-bootstrap known issue for 2016.3.4
- **PR** #37135: (aaronm-cloudtek) Fix example signing policy in salt.states.x509 docs @ *2016-10-21 11:45:24 UTC*
  - 8de7b39b5e Merge pull request #37135 from Cloudtek/x509-docs-fix
  - ce87f7311b Fix example signing policy in salt.states.x509 docs
- **PR** #37140: (vutny) pkgbuild.repo: fix GPG signing with *use_passphrase=False @ 2016-10-21 09:37:54 UTC*

  - 41ae90d3c3 Merge pull request #37140 from vutny/pkgbuild-repo-sign-with-no-passphrase
  - 409a3100a7 pkgbuild.repo: fix GPG signing with *use_passphrase=False*
- **PR** #37071: (vutny) pkgbuild.repo: add *timeout* parameter for waiting passphrase prompt @ *2016-10-21 05:20:26 UTC*
  - 96a1292a7e Merge pull request #37071 from vutny/pkgbuild-repo-gpg-sign-timeout
  - cfc3a0ed92 pkgbuild.repo: add *timeout* parameter for waiting passphrase prompt
- **ISSUE** saltstack/salt#31454: (johje349) Salt Mine memory leak (refs: #36024)
- **ISSUE** #37018: (tsaridas) get events from python (refs: #37115)
- **ISSUE** #31454: (johje349) Salt Mine memory leak (refs: #36720)
- **PR** #37115: (DmitryKuzmenko) Backport/36720 fix race condition @ *2016-10-21 05:16:15 UTC*
  - **PR** #36720: (skizunov) Fix race condition when returning events from commands (refs: #37115)
  - **PR** #36024: (DmitryKuzmenko) Don't subscribe to events if not sure it would read them. (refs: #36720)
  - 274120300d Merge pull request #37115 from DSRCorporation/backport/36720_fix_race_condition
  - d7e3209e13 For IPCClient, remove entry from instance map on close
  - 82e27634a7 Fix race condition when returning events from commands
- **PR** #37119: (jfindlay) log.setup: only assign user if defined @ *2016-10-21 05:14:55 UTC*
  - **PR** #36203: (xiaoanyunfei) fix owner of MultiprocessingLoggingQueue (refs: #37119)
  - 169a82e62b Merge pull request #37119 from jfindlay/log_proc_user
  - 8c29949a0e log.setup: only assign user if defined
  - 1d503f032c tests.integration: pass opts as a dict
  - **PR** #37126: (Ch3LL) fix digital ocean image name in profile
  - **PR** #37125: (jfindlay) add 2016.3.4 release notes
- **PR** #37120: (rallytime) Back-port #36246 to 2016.3 @ *2016-10-20 19:38:32 UTC*
  - **PR** #36418: (rallytime) Back-port #36246 to 2016.3 (refs: #37120)
  - **PR** #36246: (twangboy) Fix test_issue_6833_pip_upgrade_pip test on OS X (refs: #36418, #37120)
  - 2a35f57be8 Merge pull request #37120 from rallytime/bp-36246
  - f1c8d98119 Skip weird_install test on Mac OS X
  - 90de794290 Fix test_issue_6833_pip_upgrade_pip test on OSX
- **PR** #37103: (cachedout) Remove unnecessary sleep from unit.utils.process_test.TestProcessMana… @ *2016-10-20 08:45:07 UTC*
  - 0b87e7890a Merge pull request #37103 from cachedout/fix_proc_test
  - d7aebd1877 Remove unnecessary sleep from unit.utils.process_test.TestProcessManager.test_restarting
- **PR** #36823: (terminalmage) Update debian systemd unit files to use default KillMode, Type=notify (refs: #37162) @ *2016-10-20 05:54:42 UTC*
  - **PR** #36806: (l2ol33rt) Deb systemd should use control-group default for killmode (refs: #36823)
  - 326bbd5e30 Merge pull request #36823 from terminalmage/pr-36806
  - fb6e545f78 Use NotifyAccess=all in all unit files

  - 0ccf789172 Remove EnvironmentFile and Restart lines from unit files
  - ddd44e9b13 Use Type=notify for debian systemd units
  - 036d73f31b Use control-group default for killmode
- **PR** #37030: (isbm) Fix status.uptime for Solaris 9, 10 and 11. @ *2016-10-20 05:52:53 UTC*
  - 0c40e71e17 Merge pull request #37030 from isbm/isbm-solaris-status-fix
  - 7d7b5ef9a9 Lintfix: E8303 too many blank lines
  - c11940d14c Fix status.uptime for Solaris 9, 10 and 11.
- **PR** #37101: (rallytime) [2016.3] Merge forward from 2016.3 to carbon @ *2016-10-20 05:39:24 UTC*
  - eb88c73222 Merge pull request #37101 from rallytime/merge-2016.3
  - b445a5e579 Merge branch '2015.8' into '2016.3'
  - 68eeb29783 Add warning about GitPython 2.0.9 incompatibility with Python 2.6 (#37099)
  - 39d59ab0df Merge pull request #36880 from vutny/cp-get-salt-url
    - d1ab98b459 cp.get_url: update usage doc and add tests for *file://* URL with *dest=None*
    - c7cf79e959 cp.get_url: add note and test for https:// URL with `dest=None`
    - ff55f77179 cp.get_url: write more verbose docstring
    - 94a34a08ba cp.get_url: add integration tests
    - 983f82fcf4 cp.get_url: fix variable type check
    - b33f4d7b93 cp.get_url: log error message if no file could be fetched from *salt://* URL
    - 99cf3038cc cp.get_url: fix *dest=None* behaviour with *salt://* URL
- **PR** #36958: (twangboy) Fix bug where cmd.powershell fails to return @ *2016-10-19 16:03:58 UTC*
  - 8d44efed78 Merge pull request #36958 from twangboy/fix_cmd_powershell
  - 427be7b422 Add versionadded
  - d8e0e0e482 Fix missing comma
  - 7b46d04a84 Add note about increased completion times
  - 9365581a36 Clarify docs, add depth option
- **PR** #37086: (cachedout) Make salt-call a first-class citizen for multi-master @ *2016-10-19 15:19:09 UTC*
  - beb54b3ffa Merge pull request #37086 from cachedout/mm_req
  - 7dc15c1a48 Lint utils
  - 9bbe3c998b Lint error in publish
  - e22a3d2be6 Add multi-master support to publish.publish
  - 7f141ba38c Add function to search for substr in list
  - 007eef84d7 Extend support to event.fire_master
  - 8171c73b00 Multi-master support for salt-call
- **ISSUE** #36814: (martin-helmich) x509.create_csr creates invalid CSR (refs: #36898)
- **PR** #36898: (clinta) X509 fixes @ *2016-10-19 03:03:43 UTC*
  - 6b94153ea6 Merge pull request #36898 from clinta/x509-fixes

  - e732fe7725 fix docs on CSR state
  - 9b6f1a336c fix quotes and remove dependency on pkg_resources
  - eb4433d1ae return early if there are no requested extensions in the csr
  - d00cf8ef87 allow specifying digest for crl
  - dd50705e58 fix #36814
- **ISSUE** #34872: (cbuechler) "Minion did not return" executing state with long running command, 2016.3 regression (refs: #37025)
- **PR** #37025: (cro) Make salt.utils.minion._check_cmdline work on OSes without /proc. @ *2016-10-19 03:00:10 UTC*
  - a32b8cd741 Merge pull request #37025 from cro/freebsd_no_proc
  - 1ac87e0efd Make salt.utils.minion._check_cmdline work on OSes without /proc.
- **PR** #37050: (twangboy) Fix service state for Windows (DO NOT MERGE FORWARD) @ *2016-10-19 02:46:27 UTC*
  - **PR** #36923: (twangboy) Fix service state for Windows (refs: #37050)
  - e09d9f85c5 Merge pull request #37050 from twangboy/fix_win_service_state
  - b3b688e298 Fix tests
  - 1e1ee786c9 Set service to manual if disabled on start
  - **PR** saltstack/salt#29322: (mrproper) add http proxy support for tornado (refs: #37076)
- **PR** #37076: (jfindlay) Document proxy settings @ *2016-10-19 02:30:27 UTC*
  - 5e998638a4 Merge pull request #37076 from jfindlay/proxy_doc
  - 7328df68f5 doc.topic.tutorials.http.query: add proxy section
  - 331072b35d doc.topic.tutorials.http.query: add subheadings
  - 478def4923 doc.ref.configuration.minion: add proxy vars
- **ISSUE** #37001: (phil123456) URGENT : archive.extracted does not work anymore (refs: #37081)
- **PR** #37081: (terminalmage) Fix archive.extracted remote source_hash verification @ *2016-10-19 02:22:22 UTC*
  - 9ec366833e Merge pull request #37081 from terminalmage/issue37001
  - a3c4deeb82 Fix archive.extracted remote source_hash verification
- **ISSUE** #35097: (jwhite530) Minions die with "un-handled exception from the multiprocessing process" (refs: #37064)
- **PR** #37064: (cachedout) Unify job check in scheduler @ *2016-10-19 02:08:06 UTC*
  - 67faee1f94 Merge pull request #37064 from cachedout/issue_35097
  - 980ba892c9 Unify job check in scheduler
- **PR** #37072: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-10-18 15:23:17 UTC*
  - 7ef10f6de6 Merge pull request #37072 from rallytime/merge-2016.3
  - 78a144f19a Merge branch '2015.8' into '2016.3'
  - 7dd91c2880 Merge pull request #37053 from rallytime/update-fedora-install-docs
    - 24e0f5e024 Update the Fedora installation docs

  - 4eb0a89b7c remove options from pylint (#37054)
- **PR** #37049: (terminalmage) Further clarification on new grains docs from #37028 @ *2016-10-18 01:47:57 UTC*
  - **PR** #37028: (damon-atkins) Update topics/grains doco, about considerations before adding a Grain (refs: #37049)
  - 71fd01ab8d Merge pull request #37049 from terminalmage/grains-docs
  - 854586c6a4 Add one more paragraph
  - a0502a7b90 Restructure grain writing docs
  - 4e419e90ac Further clarification on new grains docs from #37028
- **ISSUE** saltstack/salt#18419: (jasonrm) salt-cloud fails to run as non-root user (refs: #35483)
- **ISSUE** #34806: (jerrykan) salt-cloud ignores sock_dir when firing event (refs: #35483)
- **PR** #37057: (rallytime) [2016.3] Update salt.utils.cloud references to __utils__ for cache funcs @ *2016-10-18 01:31:43 UTC*
  - **PR** #35483: (gtmanfred) use __utils__ in salt.cloud (refs: #35855, #37057, #36070)
  - 9a6671ce69 Merge pull request #37057 from rallytime/cloud-utils-cleanup
  - d0dc7d4e55 [2016.3] Update salt.utils.cloud references to __utils__ for cache funcs
- **PR** #36977: (twangboy) Remove whitespace from string commands @ *2016-10-17 22:32:03 UTC*
  - f8cd7b7b28 Merge pull request #36977 from twangboy/fix_cmd_run
  - 6586050736 Move strip to powershell block, add -NoProfile
- **PR** #37048: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-10-17 16:05:48 UTC*
  - 9378b22d80 Merge pull request #37048 from rallytime/merge-2016.3
  - 5efd6d3df9 Merge branch '2015.8' into '2016.3'
  - 7f5aced50e Merge pull request #36972 from zer0def/supervisor-state-fixes
    - 53801c6e80 Mitigates failure reports when making sure an existing supervisor process group is running, despite success.
  - 4e2ad07b0f Prevent source files in /tmp from being deleted by file.managed states (#37023)
  - 4e9824a65e args does not always exist (#37019)
- **PR** #37028: (damon-atkins) Update topics/grains doco, about considerations before adding a Grain (refs: #37049) @ *2016-10-17 09:54:21 UTC*
  - 104a153a1f Merge pull request #37028 from damon-atkins/update_topics_grains_doco
  - 01e83a715e doc/topics/grains Update doco on when a grain should be created
  - a0e1fcc951 Add information to consider before adding a Grain to doco's for Grains
- **PR** #37012: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-10-14 18:07:03 UTC*
  - c30656814d Merge pull request #37012 from rallytime/merge-2016.3
  - a7c9a72104 Merge branch '2015.8' into '2016.3'
  - c6254d59fd Merge pull request #36807 from terminalmage/issue36723
    - 7d60e73308 Fix pillar merging when ext_pillar_first is enabled
  - e2bc94b029 cp.get_file_str: do not fail if file not found (#36936)

- **ISSUE** #34397: (jaredhanson11) ignore_epoch needs to be passed through to version_cmp functions (refs: #34531)
  - **PR** #37007: (skizunov) opkg: Support ignore_epoch argument in version comparisons
  - **PR** #34531: (terminalmage) Support ignore_epoch argument in version comparisons (refs: #37007)
- **PR** #36808: (gtmanfred) allow for closing stuff in beacons (refs: #36835) @ *2016-10-14 15:50:09 UTC*
  - 8b3e65448d Merge pull request #36808 from gtmanfred/beacons
  - 727d4f309a allow for closing stuff in beacons
  - **PR** #36993: (terminalmage) Make helper funcs private
- **ISSUE** #27316: (efficks) Extracted state with zip format failed on Windows (refs: #27317)
- **ISSUE** #27207: (PredatorVI) archive.extracted state not preserving file permissions (refs: #33906)
- **ISSUE** #26569: (ssgward) Add support for password-protected zip files in archive.extracted on Windows (refs: #31116)
- **ISSUE** #23822: (sidcarter) Zip file extracted permissions are incorrect (refs: #25128)
  - **PR** saltstack/salt#36539: (jfindlay) Prefer archive.cmd_unzip (refs: #‘saltstack/salt‘#36648‘_‘_, #36648)
- **PR** #36986: (jfindlay) modules.archive.unzip: zipfile is stdlib @ *2016-10-13 21:38:00 UTC*
  - **PR** #36648: (jfindlay) Integration tests for archive execution module (refs: #36986)
  - **PR** #33906: (lomeroe) Archive unzip permissions (refs: #36539, #saltstack/salt‘#36539‘_)
  - **PR** #31116: (UtahDave) Add password support for zip files in archive module and state (refs: #36539, #saltstack/salt‘#36539‘_)
  - **PR** #27764: (basepi) Merge forward from 2015.8 to develop (refs: #36539, #saltstack/salt‘#36539‘_)
  - **PR** #27317: (efficks) State unzip should use unzip command instead of unzip_cmd. (refs: #36539, #saltstack/salt‘#36539‘_)
  - **PR** #25128: (stanislavb) Use cmd_unzip to preserve permissions (refs: #36539, #saltstack/salt‘#36539‘_)
  - a75761de87 Merge pull request #36986 from jfindlay/arch_test
  - 2ec2684860 modules.archive.unzip: zipfile is stdlib
- **ISSUE** #36422: (rippiedoos) No error Reporting for (yum)pkg.upgrade (refs: #‘saltstack/salt#36450‘_)
  - **PR** saltstack/salt#36980: (rallytime) Skip pkg.upgrade test if pkg install/upgrade has problems (refs: #36981)
  - **PR** saltstack/salt#36450: (terminalmage) Normalize pkg.upgrade and raise CommandExecutionError on failure (refs: #36981, #‘saltstack/salt#36980‘_)
- **PR** #36981: (rallytime) Skip pkg.upgrades test on distros other that Suse in 2016.3 @ *2016-10-13 21:29:36 UTC*
  - c7595b84a7 Merge pull request #36981 from rallytime/upgrades-test-fix
  - a5ae737057 Skip pkg.upgrades test on distros other that Suse in 2016.3
- **ISSUE** #36671: (wrigtim) systemd.py available() breaks on latest LSB-compliant versions of systemd (refs: #36755)
- **PR** #36755: (terminalmage) systemd.py: check retcode for service availability in systemd >= 231 @ *2016-10-13 19:41:50 UTC*
  - 6b782c15e1 Merge pull request #36755 from terminalmage/issue36671
  - d916c2b49c Handle cases where retcode/output feature is backported

  - b3364646ad Update systemd module unit tests
  - a2439acbc9 systemd.py: check retcode for service availability in systemd >= 231
- **ISSUE** #36746: (Ch3LL) Carbon: When killing a job jid output missing (refs: #36750)
- **PR** #36750: (terminalmage) Add the CLI client and pub_data as class attributes @ *2016-10-13 19:38:33 UTC*
  - 10d255c511 Merge pull request #36750 from terminalmage/issue36746
  - 0e7c600e02 Only display Ctrl-c message on SIGINT
  - 9025be48c5 Include the jid (when available) in SystemExit message on Ctrl-c
  - 9c9f1f620b Add the CLI client and pub_data as class attributes
- **ISSUE** #36240: (hrumph) win_certutil add_store state not installing certificates (refs: #36241)
- **PR** #36241: (hrumph) Fixes #36240 @ *2016-10-12 23:28:35 UTC*
  - 3ac9ced202 Merge pull request #36241 from hrumph/cert_problem
  - 51230fc263 Merge pull request #1 from rallytime/pr-36241
    - 32846794c8 Update mocks for failing tests in win_certutil_test
  - b26578d1ac Fixes #36240
- **PR** #36950: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-10-12 20:57:41 UTC*
  - c1f84388d6 Merge pull request #36950 from rallytime/merge-2016.3
  - 1d3ce45ac0 Merge branch '2015.8' into '2016.3'
    - 2ccc44f314 Merge pull request #36914 from rallytime/suse_show_link
      - b8ffd9f53f Allow alternatives.show_link function to work on Suse distros
  - 5362e5183e Merge branch '2015.8' into '2016.3'
  - fe2f094838 salt.modules.gpg: initialize GnuPG home dir with correct ownership (#36824)
  - 4b21cca909 Fix race condition in which files were removed during a file.directory (#36928)
  - 7838d8d3f9 Remove "Targeting with Executions" section from docs (#36925)
  - a56bf8bd2d Update references to future default value change that was reverted (#36924)
- **PR** #36948: (rallytime) Back-port #36943 to 2016.3 @ *2016-10-12 18:33:02 UTC*
  - **PR** #36943: (orymate) doc: document what the argument of salt –subset means (refs: #36948)
  - 7e2128c05d Merge pull request #36948 from rallytime/bp-36943
  - d2f8f18430 doc: document what the argument of salt –subset means
- **PR** #36946: (rallytime) Back-port #36892 to 2016.3 @ *2016-10-12 18:32:35 UTC*
  - **PR** #36892: (nvtkaszpir) Update tutorial.rst (refs: #36946)
  - f43a10252d Merge pull request #36946 from rallytime/bp-36892
  - 94c97ee726 Update tutorial.rst
- **ISSUE** #35198: (goestin) beacons modules: service fails (refs: #35199)
- **PR** #36945: (rallytime) Back-port #35199 to 2016.3 @ *2016-10-12 18:31:16 UTC*
  - **PR** #35199: (goestin) fix for issue #35198 (refs: #36945)
  - 5c70669ac0 Merge pull request #36945 from rallytime/bp-35199

  - 390b906c2f adhere pep8 e713
  - 79c9905fc5 Re-added accidentally removed line 85
  - 8bba13896a Fixed issue #35198 now without deprecated code.
  - 1241d87f1d fix for issue #35198
  - **PR** #36949: (terminalmage) Fix versionadded
  - **PR** #36930: (jfindlay) return opennebula errors to user
- **PR** #36929: (rallytime) [yumpkg] Skip test_pkg_upgrade_has_pending_upgrades if there are no upgrades @ *2016-10-11 22:55:49 UTC*
  - 6ea1f59058 Merge pull request #36929 from rallytime/fix-pending-upgrade-test
  - 32829b9474 [yumpkg] Skip test_pkg_upgrade_has_pending_upgrades if there are no upgrades
- **ISSUE** #36906: (sjorge) [docs] comments about targetting execution still correct? (refs: #36926, #‘salt-stack/salt‘#36925‘_‘_, #36925)
  - **PR** saltstack/salt#36925: (rallytime) Remove ”Targeting with Executions” section from docs (refs: #36926)
  - **PR** #36926: (rallytime) [2016.3] Remove ”Targeting with Executions” section from docs
- **PR** #36915: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-10-11 19:49:42 UTC*
  - b7f87e0aed Merge pull request #36915 from rallytime/merge-2016.3
  - 971c27cba2 Merge branch ’2015.8’ into ’2016.3’
  - f3443fb992 Properly handle ”shared” arg in git.init when it is a bool (#36912)
  - bdbf1619cb Check for test=True in salt.wait_for_event orchestration events (#36897)
  - **PR** #36820: (BenoitKnecht) Fix diff output of test runs for Debian slave interfaces
- **ISSUE** #36855: (edwardsdanielj) Issue with setting up schedule job via state.apply (refs: #36894)
  - **PR** #36894: (jfindlay) states.schedule: splay is not ordereddict
- **PR** #36885: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-10-10 19:30:25 UTC*
  - 86ac8bd680 Merge pull request #36885 from rallytime/merge-2016.3
  - c09b9d6e6a Merge branch ’2015.8’ into ’2016.3’
    - 3ce4897b97 Merge pull request #36857 from terminalmage/systemd-unit-tests
      - 7c78d6f419 Add unit tests for systemd scope usage
  - **PR** #36889: (terminalmage) salt-ssh: Try ”command -v” before falling back to ”which”
- **ISSUE** #36804: (Ch3LL) CARBON: error when using pkg.installed with url source (refs: #36830)
  - **PR** #36830: (terminalmage) fileclient: Change queryarg comparison from None to simple boolean check
- **PR** #36853: (rallytime) Back-port #33939 to 2016.3 @ *2016-10-07 21:44:33 UTC*
  - **PR** #33939: (bx2) Removed *!*-password check for salt-cloud vultr provider (refs: #36853)
  - 6a6bdf3e3f Merge pull request #36853 from rallytime/bp-33939
  - efbc09c1a6 Removed *!*-password check
- **PR** #36852: (rallytime) Back-port #36743 to 2016.3 @ *2016-10-07 21:35:43 UTC*
  - **PR** #36743: (do3meli) corrected OS Name in openbsd_sysctl module load error message (refs: #36852)

  - 01348bde18 Merge pull request #36852 from rallytime/bp-36743
  - 899130d11f corrected OS Name in module load error message
- **PR** #36844: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-10-07 19:20:31 UTC*
  - 0b7661244d Merge pull request #36844 from rallytime/merge-2016.3
  - 1c3a9a3ee9 Merge branch '2015.8' into '2016.3'
    - 3e6b16de2b Merge pull request #36786 from cachedout/fixup_36676
      - 3c93134e57 Typo
      - 13eb463bd9 Fixup altererernatives module
    - c126f2e132 Merge pull request #36757 from cachedout/issue_33841
      - 4bce452500 Resolve issue with minion failing to restart on failure
    - 89f9fc8c0d Merge pull request #36749 from jacobhammons/file-dict
      - 71f91b3a50 Fixes the cli examples to reference the correct function
    - 804a2a1ab0 Merge pull request #36730 from rallytime/bp-36028
      - 4be4f900ee Back-port #36028 to 2015.8
- **PR** #36835: (jfindlay) unify and expand beacon documentation @ *2016-10-07 15:59:34 UTC*
  - **PR** #36808: (gtmanfred) allow for closing stuff in beacons (refs: #36835)
  - dc5d821be6 Merge pull request #36835 from jfindlay/beacon_doc
  - b2eccdefd5 doc.topics.beacons: reflow text at 80 chars
  - b181f9890d doc.topics.{reactor|beacons}: unify examples, many minor edits
  - 28b4e30009 doc.glossary: use parenthesis
  - 82cf39db00 doc.glossary: add JID
  - cc071b75cb doc.glossary: add idempotent
- **ISSUE** #36787: (maximeguillet) postgres.* calls fail with postgresql 9.6 and .psqlrc custom file (refs: #36789)
- **PR** #36789: (maximeguillet) Fix behavior of psql -c option with postgresql 9.6 @ *2016-10-06 11:24:51 UTC*
  - 1284de27fc Merge pull request #36789 from maximeguillet/fix-psqlrc-pg9.6
  - b59c23bef1 Fix one remaining postgresql tests linked to #36787.
  - 8b92ae2061 Fix postgresql tests using position in the argument list of psql.
  - 21f2a17a07 Fix postgresql tests by adding –no-psqlrc option introduced by #36787.
  - 574e30e915 Fix behavior of psql -c option with postgresql 9.6
- **ISSUE** #36579: (scubahub) No error generated when reactor file does not exist. (refs: #36797)
- **PR** #36797: (cachedout) Error on reaction with missing SLS file @ *2016-10-06 11:19:27 UTC*
  - a1d59f4d2f Merge pull request #36797 from cachedout/issue_36579
  - 6ce4653fa3 Error on reaction with missing SLS file
- **ISSUE** saltstack/salt#36788: (damon-atkins) pillar/libvirt.py assume certtool is available and works everytime (refs: #36803)
- **PR** #36803: (gtmanfred) do not load libvirt pillar if certtool is unavailable @ *2016-10-06 11:15:14 UTC*

  - b75130be2d Merge pull request #36803 from gtmanfred/2016.3
  - 2183737085 do not load libvirt pillar if certtool is unavailable
- **PR** #36815: (BenoitKnecht) Fix glance.image_present state @ *2016-10-06 10:29:44 UTC*
  - 39148dc711 Merge pull request #36815 from BenoitKnecht/fix-glance-image-present-state-2016.3
  - 342eee444d states: glance: handle image list instead of dict
  - 02b91ecf15 states: glance: import keystone exceptions from new location
- **ISSUE** #36738: (edhgoose) rpmdev-vercmp throws lots of warnings on Amazon Linux (refs: #36739)
- **PR** #36754: (terminalmage) Base rpmdev-vercmp comparison result on retcode @ *2016-10-05 12:50:23 UTC*
  - **PR** #36739: (edhgoose) Add support for rpmdevtools returning < / > / == (refs: #36754)
  - 81c935f210 Merge pull request #36754 from terminalmage/issue36738
  - 928c99d2f7 Base rpmdev-vercmp comparison result on retcode
  - **PR** saltstack/salt#36728: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 (refs: #36785)
- **PR** #36785: (cachedout) Fixup merge forward #36728 @ *2016-10-05 11:02:16 UTC*
  - **PR** #36728: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 (refs: #36785)
  - 4bdb997dae Merge pull request #36785 from cachedout/pr-36728
  - 118ba8a772 Update alternatives module to strip newline chars
  - 24b8bba145 Merge branch '2015.8' into '2016.3'
    - a01a68d4be Merge pull request #36676 from vutny/redhat-alternatives-detect-fail
      - bba9d0d105 *alternatives.install* state: detect *alternatives* command failed
    - eab4fd563a Merge pull request #36700 from terminalmage/update-faq
      - 3d15eedfe0 Add additional information about onchanges/onchanges_in
      - 57ecbe6c53 Update minion restart example to use onchanges instead of cmd.wait
- **ISSUE** #36766: (bx2) salt-cloud (vultr) throws NameError: global name '__opts__' is not defined (refs: #36768)
- **PR** #36768: (gtmanfred) add __utils__ to vultr cloud provider @ *2016-10-05 06:59:27 UTC*
  - 90cca6b135 Merge pull request #36768 from gtmanfred/2016.3
  - 9df2fd11dd add __utils__ to vultr cloud provider
- **PR** #36764: (cachedout) Another bit of detection for failed pip tests @ *2016-10-04 13:05:29 UTC*
  - 8ff69bf0c7 Merge pull request #36764 from cachedout/more_pip_test_fixing
  - b9f5343449 Another bit of detection for failed pip tests
- **ISSUE** #27316: (efficks) Extracted state with zip format failed on Windows (refs: #27317)
- **ISSUE** #27207: (PredatorVI) archive.extracted state not preserving file permissions (refs: #33906)
- **ISSUE** #26569: (ssgward) Add support for password-protected zip files in archive.extracted on Windows (refs: #31116)
- **ISSUE** #23822: (sidcarter) Zip file extracted permissions are incorrect (refs: #25128)
  - **PR** saltstack/salt#36722: (rallytime) Skip cmd_unzip test if salt.utils.which('zip') isn't available (refs: #36747)
  - **PR** saltstack/salt#36648: (jfindlay) Integration tests for archive execution module (refs: #36747)

  - **PR** saltstack/salt#36539: (jfindlay) Prefer archive.cmd_unzip (refs: #‘saltstack/salt‘#36648‘_‘_, #36648)
- **PR** #36747: (jfindlay) modules.archive integration tests: check for gzip, rar @ *2016-10-04 11:47:32 UTC*
  - **PR** #33906: (lomeroe) Archive unzip permissions (refs: #36539, #saltstack/salt‘#36539‘_)
  - **PR** #31116: (UtahDave) Add password support for zip files in archive module and state (refs: #36539, #saltstack/salt‘#36539‘_)
  - **PR** #27764: (basepi) Merge forward from 2015.8 to develop (refs: #36539, #saltstack/salt‘#36539‘_)
  - **PR** #27317: (efficks) State unzip should use unzip command instead of unzip_cmd. (refs: #36539, #saltstack/salt‘#36539‘_)
  - **PR** #25128: (stanislavb) Use cmd_unzip to preserve permissions (refs: #36539, #saltstack/salt‘#36539‘_)
  - 5c0cbfc4c6 Merge pull request #36747 from jfindlay/arch_test
  - b5fcca9983 modules.archive int tests: check for gzip, rar
- **PR** #36744: (cachedout) Fix issue where test suite could hang on shutdown @ *2016-10-03 15:37:00 UTC*
  - 93f1daa4ce Merge pull request #36744 from cachedout/fix_test_shutdown
  - cdf2a56564 Fix issue where test suite could hang on shutdown
- **ISSUE** saltstack/salt#32490: (davegiles) __proxy__ not available when called from state.sls_id, fine from state.highstate (refs: #36696)
- **PR** #36696: (cro) pass __proxy__ in state.sls_id @ *2016-10-01 09:37:50 UTC*
  - 6fa9ec36d2 Merge pull request #36696 from cro/proxy_in_sls_id
  - 891004f3be try/except for when __proxy__ is not injected.
  - e8e53d60be pass __proxy__ in state.sls_id
- **PR** #36716: (vutny) salt.modules.ini_manage: fix creating options in empty file @ *2016-10-01 09:35:11 UTC*
  - e0b288feb3 Merge pull request #36716 from vutny/fix-ini-manage
  - 73eb773fb0 salt.modules.ini_manage: fix creating options in empty file
- **ISSUE** #29421: (scbunn) pillar data leaks through environments (refs: #36435, #saltstack/salt‘#36435‘_)
  - **PR** saltstack/salt#36628: (yhekma) Update doc to reflect the version where 'none' was added as a pillar_… (refs: #36724)
  - **PR** saltstack/salt#36435: (yhekma) Add "none" as a pillar merging strategy (refs: #‘saltstack/salt‘#36628‘_‘_, #36628)
- **PR** #36724: (rallytime) Back-port #36628 to 2016.3 @ *2016-10-01 09:33:43 UTC*
  - **PR** #36628: (yhekma) Update doc to reflect the version where 'none' was added as a pillar_… (refs: #36724)
  - 97713b09f5 Merge pull request #36724 from rallytime/bp-36628
  - 3bb2cb6379 Update doc to reflect the version where 'none' was added as a pillar_source_merging_strategy
  - **PR** saltstack/salt#36643: (roosri) a small, and unfortunate error (refs: #36725)
- **PR** #36725: (rallytime) Back-port #36643 to 2016.3 @ *2016-10-01 09:33:13 UTC*
  - **PR** #36643: (roosri) a small, and unfortunate error (refs: #36725)
  - 8e7529764b Merge pull request #36725 from rallytime/bp-36643

  - c5b8e442f9 a small, and unfortunate error
- **PR** #36726: (rallytime) Back-port #36722 to 2016.3 @ *2016-10-01 09:32:53 UTC*
  - **PR** #36722: (rallytime) Skip cmd_unzip test if salt.utils.which('zip') isn't available (refs: #36726)
  - cf32c59b6a Merge pull request #36726 from rallytime/bp-36722
  - 5904cc04c6 Skip cmd_unzip test if salt.utils.which('zip') isn't available
- **ISSUE** saltstack/salt#36718: (Ch3LL) Error when using archive.zip on python2.6 (refs: #36719)
  - **PR** #36719: (Ch3LL) fix python26 archive zip module
  - **PR** saltstack/salt#36616: (cro) Zypper fix test (refs: #36699)
- **PR** #36699: (cachedout) Fix error in test @ *2016-09-30 11:28:18 UTC*
  - 7d022a3f39 Merge pull request #36699 from cachedout/fixup_36616
  - 16f5bb70ec Remove line that checks against unordered keys
  - 0e9148293a Fix error in test
- **ISSUE** #36669: (jackywu) fix bug of including loopback addr will never work (refs: #36670)
- **PR** #36670: (jackywu) fix bug for including loopback addr @ *2016-09-30 10:21:53 UTC*
  - 0aa35596c0 Merge pull request #36670 from jackywu/2016.3
  - 48d2d512d8 fix bug for including loopback addr
- **ISSUE** #36692: (lorengordon) Expose *ignore_if_missing* param to the file.replace state (refs: #36694)
- **PR** #36694: (lorengordon) Exposes *ignore_if_missing* to file.replace state module @ *2016-09-30 10:12:27 UTC*
  - 0e8c9abe8d Merge pull request #36694 from lorengordon/issue-36692
  - 35f3bb3a8a Exposes *ignore_if_missing* to file.replace state module
  - **PR** saltstack/salt#35356: (jfindlay) document log levels and warn on all logging below info (refs: #36686)
- **PR** #36686: (jfindlay) log levels doc: try long form table @ *2016-09-29 18:21:47 UTC*
  - c089ac6c67 Merge pull request #36686 from jfindlay/log_levels
  - 4dd4fc94dc log levels doc: try long form table
- **PR** #36690: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-09-29 17:46:24 UTC*
  - e0a851b2f1 Merge pull request #36690 from rallytime/merge-2016.3
  - 7fc38c9aca Merge branch '2015.8' into '2016.3'
  - 7d1972bd5c Merge pull request #36684 from rallytime/merge-2015.8
    - 838722d225 Merge branch '2015.5' into '2015.8'
    - 8f1ba2fa26 Merge pull request #36678 from rallytime/merge-2015.5
    - 51240ecb13 Merge branch '2014.7' into '2015.5'
    - 86dc3dc9f7 Merge pull request #36641 from fuzzy-id/fix-lvm-thin-argument
    - 740516aace fix thin argument for 'lvm.lv_create'
- **PR** #36680: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-09-29 16:49:16 UTC*
  - f95dd696e5 Merge pull request #36680 from rallytime/merge-2016.3
  - 3e4ac617d3 Merge branch '2015.8' into '2016.3'

  - e4c5d0bfd6 Merge pull request #36664 from cachedout/remove_useless_size_check
    - 3d098c64ef Remove possible race between grains dumps in test
  - 8cfe371a5a Merge pull request #36663 from cachedout/skip_pip_tests_on_download_fail
    - 0c7fb91dc5 Fix error
  - e3f8618982 Merge pull request #36662 from cachedout/skip_pip_tests_on_download_fail
    - 0bbc60ccd7 Skip over tests where upstream pip isn't there
  - 3249a11e71 Merge pull request #36661 from cachedout/fix_grain_test_race
    - 2dcb92134d Fix race between minion job timeout and cli test timeout
  - b0190f248e Merge pull request #36660 from cachedout/fix_2068_issue_test
    - f4906fe771 Fix test not to rely on external resources
- **ISSUE** #28125: (peter-slovak) [2015.8] support for __env__ in Git external pillar (refs: #36659)
- **PR** #36659: (terminalmage) Support dynamic env in new-style git_pillar @ *2016-09-29 05:04:38 UTC*
  - eab1680f3f Merge pull request #36659 from terminalmage/issue28125
  - 45352b36bd Support dynamic env in new-style git_pillar
- **ISSUE** #34927: (bobrik) Salt does not run "systemd daemon-reload" on unit override (refs: #36538)
- **PR** #36538: (clinta) daemon-reload on call to service.avaliable @ *2016-09-29 02:28:00 UTC*
  - 0c2bd4b66b Merge pull request #36538 from clinta/daemon-reload
  - 833beb9b36 Merge pull request #1 from terminalmage/pr-36538
    - c4060ba2c1 Move check for service availability to a helper function
  - 20c2c91bba daemon-reload on call to service.avaliable
- **PR** #36616: (cro) Zypper fix test @ *2016-09-29 02:26:22 UTC*
  - d8a61eb9f6 Merge pull request #36616 from cro/zypper_fix_test
  - b618a5c07d Remove debugging
  - 3870589462 Test for pkg.upgrade. Most robust on Suse but better than nothing elsewhere
  - 867638ff48 Test for pkg.upgrade. Most robust on Suse but better than nothing elsewhere
- **PR** #36621: (terminalmage) Fix shadowed builtins @ *2016-09-29 02:25:54 UTC*
  - ccd92d22d2 Merge pull request #36621 from terminalmage/fix-shadowed-builtins
  - 62729eff8d Update tests to include fix for renamed function
  - 283aca8f2a Update test to reflect new function signature
  - 0f158b5edd Fix shadowed builtins
  - **PR** saltstack/salt#36618: (onorua) Fix memory leak for 0mq transport in case of TCP DDOS (refs: #36636)
- **PR** #36636: (rallytime) Back-port #36618 to 2016.3 @ *2016-09-29 02:23:09 UTC*
  - **PR** #36618: (onorua) Fix memory leak for 0mq transport in case of TCP DDOS (refs: #36636)
  - 24f82b2809 Merge pull request #36636 from rallytime/bp-36618
  - 275845c3d2 Fix memory leak for 0mq transport
- **ISSUE** #27316: (efficks) Extracted state with zip format failed on Windows (refs: #27317)

- **ISSUE** #27207: (PredatorVI) archive.extracted state not preserving file permissions (refs: #33906)
- **ISSUE** #26569: (ssgward) Add support for password-protected zip files in archive.extracted on Windows (refs: #31116)
- **ISSUE** #23822: (sidcarter) Zip file extracted permissions are incorrect (refs: #25128)
  - **PR** saltstack/salt#36539: (jfindlay) Prefer archive.cmd_unzip (refs: #‘saltstack/salt‘#36648‘_‘_, #36648)
- **PR** #36648: (jfindlay) Integration tests for archive execution module (refs: #36986) @ *2016-09-29 02:16:54 UTC*
  - **PR** #33906: (lomeroe) Archive unzip permissions (refs: #36539, #saltstack/salt‘#36539‘_)
  - **PR** #31116: (UtahDave) Add password support for zip files in archive module and state (refs: #36539, #saltstack/salt‘#36539‘_)
  - **PR** #27764: (basepi) Merge forward from 2015.8 to develop (refs: #36539, #saltstack/salt‘#36539‘_)
  - **PR** #27317: (efficks) State unzip should use unzip command instead of unzip_cmd. (refs: #36539, #saltstack/salt‘#36539‘_)
  - **PR** #25128: (stanislavb) Use cmd_unzip to preserve permissions (refs: #36539, #saltstack/salt‘#36539‘_)
  - 750ff8220c Merge pull request #36648 from jfindlay/arch_test
  - cc4d958557 modules.archive: add integration tests
  - 99bf89447b modules.archive: add opts arg to g(un)zip
  - c1219e68c5 modules.archive.unzip: depend on zipfile module
  - 315b031de9 modules.archive: use less redundant message
  - **PR** saltstack/salt#36389: (cachedout) Pr 36386 (refs: #36650)
- **PR** #36650: (rallytime) Revert ”Pr 36386” @ *2016-09-29 02:11:15 UTC*
  - **PR** #36386: (xiaoanyunfei) fix salt-api’s default opts were covered by salt-master #35734 (refs: #36389, #36650, #saltstack/salt‘#36389‘_)
  - **PR** #35734: (xiaoanyunfei) fix salt-api’s default opts were covered by salt-master (refs: #36386)
  - 91aa464d5d Merge pull request #36650 from saltstack/revert-36389-pr-36386
  - 33ef5bffe6 Revert ”Pr 36386”
- **ISSUE** #36304: (Ch3LL) stack trace when transport is not a currently supported transport (refs: #36646)
- **PR** #36646: (rallytime) Provide an error message when invalid transport is set @ *2016-09-28 22:52:11 UTC*
  - ab5c0e9e65 Merge pull request #36646 from rallytime/fix-36304
  - ae021d6dec Provide an error message when invalid transport is set
- **PR** #36635: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-09-28 21:10:25 UTC*
  - 6d9b28506c Merge pull request #36635 from rallytime/merge-2016.3
  - 787c1f557e Pylint fix
  - da574e5b03 Merge branch ’2015.8’ into ’2016.3’
  - f0d561a229 Merge pull request #36632 from isbm/isbm-thin-modules-config-15.8
    - 975f8bb27d Add extra-mods options to the Salt-Thin via SSH CLI
    - a441b35588 Add documentation about Salt Thin configuration
    - 3bfb17ee62 Add a description of the thin/min parameters to the master config

        - 3d878f9da5 Get the thin Salt with configured extra modules on SSH
        - 2be9330be6 Add thin options to the master config.
        - 58577d342e Generate thin with configured extrta modules

- **ISSUE** #36553: (nilliams) states.hg.latest claims to succeed despite errors (refs: #36620)
- **PR** #36620: (rallytime) Don't allow mercurial states to return True with errors @ *2016-09-28 05:50:50 UTC*
    - 83da81cdfd Merge pull request #36620 from rallytime/fix-36553
    - a828bdd0b8 Update test mocks for cmd.run_all dicts
    - 3904dfc5a8 Don't allow mercurial states to return True with errors
- **PR** #36622: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-09-28 05:39:43 UTC*
    - 1c001d0ee1 Merge pull request #36622 from rallytime/merge-2016.3
    - 90c66ef756 Merge branch '2015.8' into '2016.3'
        - 9b9e167b47 Merge pull request #36562 from kiorky/s2015.8
            - 47c3d03035 Fix pkg.latest_version using localized output
        - 4ab52ae0f6 Merge pull request #36607 from vutny/detect-service-fail
            - c4f899b3b3 *salt.states.service*: detect that service failed to start/stop
        - 5de036b56c Merge pull request #36611 from multani/2015.8
            - 79fdc12395 jinja: fix YAML terminator removal in Jinja's "yaml" filter
        - 6e36191fc4 Fix trust key 2015.8 (#36540)
- **PR** #36520: (twangboy) Fix cmd.script runas for Windows @ *2016-09-28 04:07:00 UTC*
    - e7def534b1 Merge pull request #36520 from twangboy/fix_cmd.script_runas
    - 377ced5c24 Remove directory in Windows with runas
    - 25d52efeac Fix mkdir
    - 18d41f7711 Add mkdir
    - 9d55bff914 Use cachedir for Windows
- **ISSUE** saltstack/salt#32368: (vitaliyf) Low timeout values causes duplicate commands to execute (refs: #36564)
- **PR** #36564: (DmitryKuzmenko) Improve and fix *_check_cache_minions* @ *2016-09-28 02:50:54 UTC*
    - 798bf3086b Merge pull request #36564 from DSRCorporation/bugs/32368_grains_match_bug
    - be61f97db3 Minor: syntax error fixes.
    - 29660ed672 Improve and fix *_check_cache_minions*
- **PR** #36606: (danlsgiga) Add support for ACL Tokens in consul_pillar with the option consul.token @ *2016-09-28 02:46:03 UTC*
    - 133705d567 Merge pull request #36606 from danlsgiga/consul_pillar_token
    - a5907c9c89 Add support for ACL Tokens in consul_pillar with the option consul.token
- **PR** #36613: (slinn0) Remove file.check_managed_changes when not needed (backport of PR #36589 to 2016.3) @ *2016-09-28 02:35:56 UTC*
    - **PR** #36589: (slinn0) Do not generate pchanges in file.managed unless test=True (refs: #36613)
    - b365f1e34d Merge pull request #36613 from slinn0/2016.3_36588_fixes

  - d9da5cb2d4 Backport of PR #36589 / Issue #36588 to 2016.3 branch.
- **PR** #36609: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-09-27 18:34:22 UTC*
  - e23af98d97 Merge pull request #36609 from rallytime/merge-2016.3
  - f15d4a38bd Merge branch '2015.8' into '2016.3'
  - 57ec792f6b Merge pull request #36550 from rickyninja/2015.8
    - f9ef30aabe Add version_cmp for FreeBSD pkg.
- **PR** #36595: (cachedout) Remove tests which no longer apply @ *2016-09-27 07:38:15 UTC*
  - 25fa754d94 Merge pull request #36595 from cachedout/issue_7754_fix
  - 3a83b0bd16 Remove tests which no longer apply
- **ISSUE** #36586: (gehzumteufel) Documentation update (refs: #36594)
- **PR** #36594: (cachedout) Update boostrap docs to recent versions of Ubuntu @ *2016-09-27 06:18:49 UTC*
  - aed98f47de Merge pull request #36594 from cachedout/issue_36586
  - 1e6a60ab01 Update boostrap docs to recent versions of Ubuntu
- **PR** #36585: (twangboy) Add pyOpenSSL to req file for Windows @ *2016-09-27 05:49:42 UTC*
  - c79f525863 Merge pull request #36585 from twangboy/add_pyopenssl
  - 5fc63a1054 Add pyOpenSSL to req file for Windows
- **ISSUE** #36568: (lkx007) cp.push remove_source problem (refs: #36572)
  - **PR** #36572: (cachedout) Fix salt.utils.rm_rf to delete files too
- **ISSUE** #36491: (cro) pkg.upgrade does not upgrade on Leap 42.1 or Tumbleweed (refs: #36495)
- **PR** #36495: (cro) Fix pkg.upgrade for zypper @ *2016-09-26 10:02:39 UTC*
  - d0dd92b037 Merge pull request #36495 from cro/zypper_fix
  - 6c5807c4be Fix pkg.upgrade for zypper
- **ISSUE** #27316: (efficks) Extracted state with zip format failed on Windows (refs: #27317)
- **ISSUE** #27207: (PredatorVI) archive.extracted state not preserving file permissions (refs: #33906)
- **ISSUE** #26569: (ssgward) Add support for password-protected zip files in archive.extracted on Windows (refs: #31116)
- **ISSUE** #23822: (sidcarter) Zip file extracted permissions are incorrect (refs: #25128)
- **PR** #36539: (jfindlay) Prefer archive.cmd_unzip @ *2016-09-26 10:02:11 UTC*
  - **PR** #33906: (lomeroe) Archive unzip permissions (refs: #36539, #saltstack/salt‘#36539‘_)
  - **PR** #31116: (UtahDave) Add password support for zip files in archive module and state (refs: #36539, #saltstack/salt‘#36539‘_)
  - **PR** #27764: (basepi) Merge forward from 2015.8 to develop (refs: #36539, #saltstack/salt‘#36539‘_)
  - **PR** #27317: (efficks) State unzip should use unzip command instead of unzip_cmd. (refs: #36539, #saltstack/salt‘#36539‘_)
  - **PR** #25128: (stanislavb) Use cmd_unzip to preserve permissions (refs: #36539, #saltstack/salt‘#36539‘_)
  - 4bca246a27 Merge pull request #36539 from jfindlay/arch_perms
  - d64ae48783 states.archive: use archive.cmd_unzip when possible

  - 928a7891b4 modules.archive.unzip: log a warning about perms
- **ISSUE** #36514: (nilliams) salt.stages.hg errors when -identity option is used (refs: #36546)
- **PR** #36546: (rallytime) Mercurial Module: Pass the identity_path portion as own arg @ *2016-09-26 09:44:30 UTC*
  - ab50cde391 Merge pull request #36546 from rallytime/fix-36514
  - 9afe76759e Mercurial Module: Pass the identity_path portion as own arg
- **ISSUE** #35480: (jelenak) 200 processes of salt-master (2016.3.2) (refs: #36184, #36555, #37254)
- **PR** #36555: (DmitryKuzmenko) Bugs/35480 master shutdown @ *2016-09-26 09:25:43 UTC*
  - aea55fce61 Merge pull request #36555 from DSRCorporation/bugs/35480_master_shutdown
  - 6ad2998715 Wait for kill in ProcessManager should be greater in main process than in subprocess.
  - c9c45a5d79 Don't set the *daemon* flag for LoggingQueue process.
- **PR** #36542: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-09-23 22:47:26 UTC*
  - a1e0afe1c7 Merge pull request #36542 from rallytime/merge-2016.3
  - 861a001749 Merge branch '2015.8' into '2016.3'
  - 07c9d040c0 Fixup the rabbitmq_user state test failure (#36541)
- **ISSUE** #29421: (scbunn) pillar data leaks through environments (refs: #36435, saltstack/salt#36435)
  - **PR** #36532: (rallytime) Back-port #36435 to 2016.3
  - **PR** #36435: (yhekma) Add "none" as a pillar merging strategy (refs: #36532)
  - **PR** #36535: (rallytime) Be explicit about the salt.utils.templates import
  - **PR** #36537: (rallytime) Wrap the entire GrainsAppendTestCase class with destructiveTest
- **PR** #36529: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-09-23 16:42:42 UTC*
  - 55cf4d6a04 Merge pull request #36529 from rallytime/merge-2016.3
  - 52cf40db8c Merge branch '2015.8' into '2016.3'
  - 1c3758544c Merge pull request #36441 from twangboy/update_setup
    - fc4a03a75d Check for existing library on Windows
- **PR** #36483: (dmurphy18) Isolate sun IPv6 fix to Sun OS only @ *2016-09-23 09:24:54 UTC*
  - 03491634ff Merge pull request #36483 from dmurphy18/aix_fix_ipv6
  - b68f982c6a Updated check as per code review
  - cbcdb472fe Isolate SUN IPv6 fix to Sun Os only
- **ISSUE** #36279: (alertedsnake) state.postgres_privileges should allow grants to ALL tables/sequences. (refs: #36280)
- **PR** #36280: (alertedsnake) Feature/2016.3 better postgresql grants @ *2016-09-23 07:55:32 UTC*
  - **PR** #36249: (alertedsnake) Quote postgres privilege target names (refs: #36280)
  - 654fa8d770 Merge pull request #36280 from jwplayer/feature/2016.3-better-postgresql-grants
  - e7a597da00 Bugfix: don't concatenate when not needed
  - ba60b7972a Additional documentation.
  - 8b877f014d 'All' grants for PostgreSQL.

- **PR** #36508: (twangboy) Fix chocolatey @ *2016-09-23 07:36:03 UTC*
  - 8104d5c92a Merge pull request #36508 from twangboy/fix_chocolatey
  - a7c858d9ab Fix retcodes
  - feadd827a7 Add additional functionality to upgrade
  - fb5eb4dc03 Fix retcodes, add upgrade function
- **PR** #36519: (terminalmage) Rewrite minionfs walkthrough @ *2016-09-23 05:19:59 UTC*
  - 364f74dfc9 Merge pull request #36519 from terminalmage/docs
  - 2df51ce3e9 Rewrite minionfs walkthrough
  - cc9d41fb0e Change items in minionfs blacklist/whitelist example
- **PR** #36505: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-09-22 17:37:33 UTC*
  - 6f54e16cdf Merge pull request #36505 from rallytime/merge-2016.3
  - 5bd4d6430b Merge branch '2015.8' into '2016.3'
  - bf6195b9a6 *postgres_extension* state: small corrections in docstrings (#36500)
  - b021ea5d40 Merge pull request #36464 from vutny/postgres-tablespace-options
    - 580aed87b9 Fix *options* parameter processing in *postgres_tablespace.present*
- **ISSUE** #35813: (UtahCampusD) Empty dictionary returned from grains.items command within local client (refs: #36496)
- **PR** #36496: (cachedout) Add repr to namespacedict @ *2016-09-22 04:34:11 UTC*
  - 464c4305f9 Merge pull request #36496 from cachedout/namespace_repr
  - 333842c319 Add repr to namespacedict
- **PR** #36474: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-09-22 04:16:58 UTC*
  - a0f838af36 Merge pull request #36474 from rallytime/merge-2016.3
  - 8805b57a1e Merge branch '2015.8' into '2016.3'
    - 41d3c09857 Merge pull request #35433 from terminalmage/issue34790
      - 71b51f49ba Add integration tests for PR #35433
      - 82515eccde Add an additional hint for cases where rev == 'HEAD'
      - 4b7e2f9475 git.latest: Add a hint for possible rev changes resulting in non-fast-forward failures
    - 87263b9387 Merge pull request #36445 from notpeter/salt_cloud_iam_role
      - 469d1a61fe Remove (required).
      - 98449e66f5 Better docs for use-instance-role-credentials.
- **ISSUE** #36475: (amendlik) GitFS online documentation is missing a section present in the code (refs: #36478)
- **PR** #36478: (rallytime) Add the "bash" option to the "code-block"directive. @ *2016-09-22 04:15:14 UTC*
  - ec4f4f49ca Merge pull request #36478 from rallytime/fix-36475
  - 7be7d5832f Add the "bash" option to the "code-block"directive.
- **PR** #36484: (terminalmage) Fix for temp files being left over by salt-cloud execution @ *2016-09-22 04:11:58 UTC*
  - **PR** #36482: (clarkperkins) Have salt-cloud clean up tmp files (refs: #36484)

  - 4c6e7bf873 Merge pull request #36484 from terminalmage/salt-cloud-tmp-files
  - 0bf520e089 Ensure temp file is actually removed
  - 072fd823f7 Use os.write() on file descriptor instead of opening a filehandle
  - f61e8d6366 Fix for temp files being left over by salt-cloud execution
- **PR** #36486: (terminalmage) Improve the rebase docs in contributing guidelines @ *2016-09-21 19:21:10 UTC*
  - 9005a87635 Merge pull request #36486 from terminalmage/rebase-docs
  - 4839c325ae Improve the rebase docs in contributing guidelines
- **PR** #36455: (twangboy) Update docs for Windows @ *2016-09-21 14:28:28 UTC*
  - bc5ac9adae Merge pull request #36455 from twangboy/windows_installation_docs
  - ec67a9bb2f Add cachedout's recommendations
  - 26a40dadbe Update docs for Windows
- **PR** #36459: (cachedout) Pr 36426 @ *2016-09-21 06:34:29 UTC*
  - 3d23371ca2 Merge pull request #36459 from cachedout/pr-36426
  - bb5c01ae9d Lint
  - 85d2068326 Refactor for testing and adding related engine tests
  - 266adae2fd Make sqs_events engine support owner_acct_id
- **PR** #36442: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-09-20 23:16:30 UTC*
  - c8e15dcdca Merge pull request #36442 from rallytime/merge-2016.3
  - 2740fb7bfd Merge branch '2015.8' into '2016.3'
  - 266dd7c00a Merge pull request #36379 from twangboy/windows_grains
    - 6138390da7 Fix typo
    - cf045e5c03 Remove comment
    - ddb6e11bcb Remove refactoring
    - 45dc920db0 Clarify comments
    - 211fd3b47e Improve version checking
    - 88be5a3761 Check for Python 2.7.12 and 3.5.2
    - 6f80f0062a Add osservicepack grain
    - 04c4ec4f81 Fix lint
    - 5789ea99cf Force string
    - 6c5bd7664b Fix join syntax
    - ac8610d523 Add ServicePack to osrelease
    - 92034936c1 Fix windows grains for os
  - 5625827ee2 Merge pull request #36378 from terminalmage/issue36321
    - 7b1f621206 Fix git.latest test with local changes to reflect changes in state
    - 0364fedb76 Use a single conditional
    - 0dd1e7b53e git.latest Treat an up-to-date checkout with local changes as up-to-date

- **PR** #36310: (thatch45) Fix bug where the client will destroy the loop @ *2016-09-20 13:14:23 UTC*
    - d0a495f08b Merge pull request #36310 from thatch45/keep_loop
    - a3c0d4a0ab Add docstring
    - 083f1d998a Fix bug where the client will destroy the loop
- **PR** #36394: (oba11) fix accound_id in boto_iam and get_region in boto_sns @ *2016-09-20 13:11:28 UTC*
    - 6e16ca46ed Merge pull request #36394 from oba11/module-fixes
    - 966685020c fix accound_id in boto_iam and get_region in boto_sns
- **PR** #36424: (jfindlay) skip some mac_timezone tests @ *2016-09-20 06:43:47 UTC*
    - **PR** #36194: (jfindlay) skip some mac_timezone tests (refs: #36424)
    - ae1fc430c2 Merge pull request #36424 from jfindlay/bp-36194
    - a20a2148bf skip some mac_timezone tests
- **ISSUE** #36388: (qurczak) pkg.list_upgrades return debug information rather than packages list (refs: #36428)
- **PR** #36428: (terminalmage) A couple fixes for Antergos Linux @ *2016-09-20 06:42:16 UTC*
    - 6319e3419a Merge pull request #36428 from terminalmage/issue36388
    - b0069ad0d8 pacman.py: use os_family grain to assign as pkg virtual module
    - 5d632dbfca Properly set os grain for Antergos
    - 0ae8dca2d0 pkg.list_upgrades: Ignore ”downloading” lines in pacman output
- **ISSUE** #36373: (frioux) Salt-API does not validate input properly (refs: #36425)
- **PR** #36425: (whiteinge) Check for dictionary explicitly since we’re accessing it as one @ *2016-09-20 06:41:40 UTC*
    - 155bd14b5e Merge pull request #36425 from whiteinge/salt-api-dict-payload
    - 0b63ed258f Check for dictionary explicitly since we’re accessing it as one
- **ISSUE** saltstack/salt#18341: (falzm) Dry-running state.highstate only returns the first change (refs: #36199)
- **PR** #36199: (thatch45) skip all failhards if test=True @ *2016-09-20 05:38:32 UTC*
    - 420be364ee Merge pull request #36199 from thatch45/fix_18341
    - e13d61f06a skip all failhards if test=True
- **PR** #36418: (rallytime) Back-port #36246 to 2016.3 (refs: #37120) @ *2016-09-19 21:56:52 UTC*
    - **PR** #36246: (twangboy) Fix test_issue_6833_pip_upgrade_pip test on OS X (refs: #36418, #37120)
    - b2365f553e Merge pull request #36418 from rallytime/bp-36246
    - aab02f28b4 Ensure we have a test venv created using virtualenv < 13.0
- **PR** #36419: (rallytime) Back-port #36329 to 2016.3 @ *2016-09-19 21:56:33 UTC*
    - **PR** #36329: (oz123) Fix a minor typo in docs (refs: #36419)
    - bc703e2062 Merge pull request #36419 from rallytime/bp-36329
    - ffdebf7a25 Fix a minor typo in docs
- **PR** #36420: (rallytime) Back-port #36365 to 2016.3 @ *2016-09-19 21:56:17 UTC*
    - **PR** #36365: (Kimamisa) Fix a minor typo in docs (refs: #36420)

  - fbfa0657fc Merge pull request #36420 from rallytime/bp-36365
  - 864e513fca Fix a minor typo in docs
- **PR** #36413: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-09-19 18:59:31 UTC*
  - 3dd2590e28 Merge pull request #36413 from rallytime/merge-2016.3
  - c64e489f6f Merge branch '2015.8' into '2016.3'
    - 9bc4eeb71e Fix typo (#36409)
    - ac5c812e4b Fix OS identification for CloudLinux (#36408)
    - bb4d69f58a git.latest: fail gracefully for misconfigured remote repo (#36391)
    - ad7045ad3b Merge pull request #36315 from puneetk/patch-6
      - 3ac308ac76 Update aptpkg.py
      - 892cc4cd48 Update aptpkg.py
      - cbe98d97a3 Fix pylint whitespace errors
      - e5371ac720 No force_yes parameter to pkg.upgrade #21248
    - 2aa6df859a Merge pull request #36381 from twangboy/fix_win_service
      - 04edea5c59 Add '/y' switch to the net stop and start commands
    - 373c5db180 Merge pull request #36384 from twangboy/update_setup_req
      - a817aef1c2 Add windows requirements file
- **ISSUE** #36371: (nasenbaer13) _extern_path in fileclient is broken (refs: #36305)
- **PR** #36305: (gtmanfred) cache query args with url as well @ *2016-09-19 18:30:51 UTC*
  - a8a3a9f021 Merge pull request #36305 from gtmanfred/2016.3
  - 70e7f6d58b cache query args with url as well
- **PR** #36389: (cachedout) Pr 36386 @ *2016-09-17 11:54:37 UTC*
  - **PR** #36386: (xiaoanyunfei) fix salt-api's default opts were covered by salt-master #35734 (refs: #36389, #36650, #saltstack/salt`#36389`_)
  - **PR** #35734: (xiaoanyunfei) fix salt-api's default opts were covered by salt-master (refs: #36386)
  - 602bd2d1ef Merge pull request #36389 from cachedout/pr-36386
  - f5d63d93cc Lint
  - 93269cfb65 fix salt-api log and pid
  - **PR** #36352: (pass-by-value) Update versionadded and release notes
- **PR** #36369: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-09-16 16:31:14 UTC*
  - 495d365e54 Merge pull request #36369 from rallytime/merge-2016.3
  - 37aea4188a Merge branch '2015.8' into '2016.3'
  - 40b2e3d189 Merge pull request #36353 from rallytime/refresh-db-cleanup
    - 275319193a Check for Ign/Hit membership instead of == in aptpkg.refresh_db
  - df9d9b3624 Merge pull request #36355 from rallytime/bp-36288
    - 70ffdafbf0 Schema test requires jsonschema 2.5.0 or above

  - 3f308d7694 postgres_extension: report changes when an extension was installed (#36335)
  - d2a583bc22 Merge pull request #36337 from cachedout/conduct
    - 2fb61b9c9f SaltStack's code of conduct
  - ef128ad0b0 Return None when find_file identifies the path as a directory (#36342)
  - **PR** #36249: (alertedsnake) Quote postgres privilege target names (refs: #36280)
  - **PR** #36330: (silenius) set __virtualname__ to 'service'
- **ISSUE** #36338: (jbonachera) infoblox.present state does not use "infoblox_server", "infoblox_user" or "infoblox_password" arguments (refs: #36339)
  - **PR** #36339: (jbonachera) Use infoblox_* values if present in arguments
  - **PR** #36345: (gtmanfred) remove help message from glance module
  - **PR** #36346: (rallytime) Add resize2fs unit test from blockdev_test to disk_test
  - **PR** #36344: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 (refs: #36346)
- **ISSUE** #36292: (lorengordon) pkg.check_db is not available in salt 2016.3? (refs: #36350)
- **PR** #36350: (terminalmage) Add note about yumpkg.check_db removal in Boron @ *2016-09-15 20:32:32 UTC*
  - f09c3e499f Merge pull request #36350 from terminalmage/docs
  - b815c98577 Add note about yumpkg.check_db removal in Boron
- **PR** #36344: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 (refs: #36346) @ *2016-09-15 17:38:57 UTC*
  - a33da842c0 Merge pull request #36344 from rallytime/merge-2016.3
  - d1f560147d Merge branch '2015.8' into '2016.3'
  - dc518c5340 Skip test_resize2fs if resize2fs does not exists (#36325)
- **ISSUE** #36308: (ahammond) salt-cloud defaults to IPv6 rather than IPv6 (refs: #36312)
  - **PR** #36312: (ahammond) merge error overwrites correct ssh_host with stale data in ip_address
- **ISSUE** #35819: (cable2999) pkg.group_installed doesn't handle missing package group (refs: #`saltstack/salt`#35907`_`_, #35907)
  - **PR** saltstack/salt#35907: (rallytime) Catch CommandExecutionError when the group in group_installed doesn't exist (refs: #36299)
- **PR** #36299: (rallytime) Gate the pkg.group_installed state test: not all pkg modules have group_install @ *2016-09-14 19:04:26 UTC*
  - 6a3019bbf1 Merge pull request #36299 from rallytime/gate-pkg-group-installed-test
  - 9e15df9b23 Switch the order of the decorator
  - ee997be6d8 Fix pkg group test by passing a list instead of str
  - c7d8867096 Gate the pkg.group_installed state test: not all pkg modules have group_install
- **ISSUE** #33686: (BretFisher) blockreplace marker_end isn't applied with newline (refs: #`saltstack/salt`#36273`_`_, #36273)
  - **PR** saltstack/salt#36273: (techhat) Add append_newline flag for #33686 (refs: #36295)
  - **PR** #36295: (rallytime) Back-port #36273 to 2016.3
  - **PR** #36273: (techhat) Add append_newline flag for #33686 (refs: #36295)
  - **PR** #36296: (rallytime) Back-port #36124 to 2016.3

  - **PR** #36124: (twangboy) Skip test on all OS's but linux (refs: #36296)
- **PR** #36297: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-09-14 16:07:30 UTC*
  - a8a72c985f Merge pull request #36297 from rallytime/merge-2016.3
  - e2f1cf6025 Merge branch '2015.8' into '2016.3'
    - b9b8e45362 Merge pull request #36272 from terminalmage/improved-gitfs-logging
      - 223a20e987 Improved gitfs/git_pillar error logging
    - abb6aacb4b Merge pull request #36277 from terminalmage/gitfs-check-key-path
      - 4fee18c820 salt.utils.gitfs: Check for existence of ssh keys
    - ed2d2bd331 Integration tests fixes for 2015.8 (#36262)
    - 297a12c387 Fix misspelling of "occurred" in log messages/exceptions (#36270)
  - **PR** #36178: (cachedout) Filter out pub kwargs from cloud runner
- **PR** #36238: (pass-by-value) Add ability to clone from a snapshot to salt-cloud vmware driver @ *2016-09-14 05:31:51 UTC*
  - fc7a1d536f Merge pull request #36238 from pass-by-value/vmware_clone_from_snapshot
  - dd670bd18f Fix lint error and add try except
  - d96981639b Add ability to clone from a snapshot to salt-cloud vmware driver
  - **PR** #36263: (meaksh) Integration tests fixes for 2016.3
- **PR** #36264: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-09-13 18:25:41 UTC*
  - d634fd8628 Merge pull request #36264 from rallytime/merge-2016.3
  - f603757b55 Merge branch '2015.8' into '2016.3'
  - 931486ba35 Merge pull request #36096 from twangboy/update_setup
    - dc1988add5 fix download when requests not present
    - b4479bff5f Add additional required dll's
  - b0dd6ff5c8 Merge pull request #36244 from terminalmage/gen-back-bug
    - 363b21fd9b salt.fileserver.Fileserver: Don't try to split a list in _gen_back
  - dcc9380996 Merge pull request #36245 from terminalmage/roots-bug
    - 75d4997b70 roots backend: Don't include '.' or '..' in empty_dirs
  - fdf40907b7 Some unit tests fixes (#36227)
- **ISSUE** #33525: (anlutro) file.serialize no longer indents/pretty-prints in 2016.3 (refs: #35688)
- **PR** #35688: (cachedout) Splat serializer default configs into the serializer kwargs @ *2016-09-13 09:21:46 UTC*
  - de06116075 Merge pull request #35688 from cachedout/issue_33525
  - 4910e8191c Provide fallback for serializers without opts
  - a238666aba Add serializer test
  - 345fd2a9e5 Splat serializer default configs into the serializer kwargs
- **ISSUE** #36021: (mirceaulinic) Scheduled runners not executed (for proxy minions, at least) (refs: #36025)
- **PR** #36025: (mirceaulinic) Potential fix for #36021 @ *2016-09-13 07:46:41 UTC*

  - d9d477ed45 Merge pull request #36025 from cloudflare/CF-FIX-36021
  - 03007be6b1 Potential fix for #36021
  - **PR** #36183: (opdude) Fix timezones states on OS X
- **PR** #36235: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-09-12 17:40:42 UTC*
  - fcbebb40c3 Merge pull request #36235 from rallytime/merge-2016.3
  - 99dcf84b62 Merge branch '2015.8' into '2016.3'
    - 4e9490eebe Merge pull request #36214 from vutny/postgres-extension-doc
      - 5fe548f043 *postgres_extension* state module: fix docstrings
    - 5b7b96c0b7 Merge pull request #36205 from vutny/postgres-tablespace-doc
      - 78296b90d8 Add missing *maintenance_db* kwarg to *postgres_tablespace.present* docstring
    - 6a5f7cb346 Ignore states that do not have a numeric jid, i.e. 'req' (#36185)
- **ISSUE** #35423: (Ch3LL) Stacktrace when running state.sls against an sls does not exist (refs: #36137)
- **ISSUE** #33915: (mattglv) Orchestration runner output on Success vs Failures in 2016.3.0 (refs: #36137)
- **ISSUE** #25664: (sdm24) 2015.5.2 MySQL Returner: salt-run jobs.lookup_jid doesn't return full result for highstate output (refs: #35559)
- **PR** #36137: (cachedout) Allow highstate outputter to show all results @ *2016-09-12 16:37:49 UTC*
  - **PR** #35559: (Jlin317) Fix highstate outputter when it's given multiple results (refs: #36137)
  - 7b96197c5e Merge pull request #36137 from cachedout/issue_35423
  - 1e8431f2b8 Allow highstate outputter to show all results
- **ISSUE** #35340: (dqminh) Custom modules are only resynced to minions at highstate (refs: #36217)
  - **PR** #36217: (cachedout) Docs clarification for module sync and state.apply
- **ISSUE** #35480: (jelenak) 200 processes of salt-master (2016.3.2) (refs: #36184, #36555, #37254)
- **PR** #36184: (DmitryKuzmenko) Disable signal handling while handling signal @ *2016-09-11 22:59:08 UTC*
  - 6ebe655e17 Merge pull request #36184 from DSRCorporation/bugs/35480_master_shutdown
  - 229504efef Removed unused import.
  - ca8eb7e076 Don't run the same signal handler twice. Catch os.kill errors.
- **PR** #36203: (xiaoanyunfei) fix owner of MultiprocessingLoggingQueue (refs: #37119) @ *2016-09-11 09:15:15 UTC*
  - f11f093f8c Merge pull request #36203 from xiaoanyunfei/logowner
  - 74dc90c7bb cancle pr last
  - 90e4a25dd0 Merge branch 'logowner' of https://github.com/xiaoanyunfei/salt into logowner
    - bd61b88fc8 fix log owner
    - 58160ed6c0 Merge branch '2016.3' of github.com:saltstack/salt into 2016.3
    - f2de71782b move back
    - b8214824fd add simplify code
    - aec9385c6b Merge branch '2016.3' of github.com:saltstack/salt into 2016.3
    - 1074b3355d Merge branch '2016.3' of github.com:saltstack/salt into 2016.3

        - ea0d74cd27 fix salt-api opts
    - ffd87b2f2f fix logqueue owner
    - **PR** #36193: (thatch45) Fix stack trace in salt-ssh gitfs
- **PR** #36188: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-09-09 18:59:20 UTC*
    - f035121291 Merge pull request #36188 from rallytime/merge-2016.3
    - 521a7b2470 Merge branch '2015.8' into '2016.3'
    - d4628f3c6b Allow additional kwargs in states.dockerng.image_present (#36156)
    - 24b0387b92 Back-port #36070 to 2015.8 (#36169)
    - 116d7ac3e5 If windows pkg db hasn't been created yet, refresh the db instead of stacktracing (#36008)
- **ISSUE** #35819: (cable2999) pkg.group_installed doesn't handle missing package group (refs: #'salt-stack/salt'#35907'_'_, #35907)
- **PR** #35907: (rallytime) Catch CommandExecutionError when the group in group_installed doesn't exist @ *2016-09-09 10:14:16 UTC*
    - 1d5f97d36b Merge pull request #35907 from rallytime/fix-35819
    - d7380d83be requires_system_grains decorator needs a grains=None kwarg
    - b20f6b9384 Catch CommandExecutionError when group_installed doesn't exist
- **ISSUE** saltstack/salt#35972: (tjyang) DeprecationWarning: The "osmajorrelease" will be a type of an integer. (refs: #36068)
    - **PR** saltstack/salt#35637: (cachedout) Add Nitrogen release notes (refs: #36068)
- **PR** #36068: (rallytime) Remove grains type deprecation warning from 2016.3 @ *2016-09-09 10:00:50 UTC*
    - 40127b6bf3 Merge pull request #36068 from rallytime/fix-35972
    - 2b7679c9f6 Remove grains type deprecation warning from 2016.3
- **ISSUE** #36094: (UtahDave) Windows stacktraces on msgpack on Carbon (refs: #36152)
- **PR** #36152: (cachedout) Remove unnecessary unpack @ *2016-09-09 09:13:47 UTC*
    - 24bd03734d Merge pull request #36152 from cachedout/issue_36094
    - 95eb95a0f8 Remove unnecessary unpack
- **PR** #36158: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-09-08 21:53:50 UTC*
    - dc3a68ed8c Merge pull request #36158 from rallytime/merge-2016.3
    - 7f955bda0a Merge branch '2015.8' into '2016.3'
        - 6242702288 Fix issue with cp.push (#36136)
        - 0e13118f6e Document *owner* kwarg for *postgres_schema.present* state function (#36147)
        - 4cc8ea9577 Merge pull request #36146 from meaksh/tests-fixes-for-2015.8
            - 9f9aa4779c rename darwin_sysctl.py to mac_sysctl.py
            - 2cf6f36d89 modules.darwin_sysctl: __virtual__ return err msg.
            - f74ca15f50 Remove test for file dir behavior
            - c65aefee20 Fix tests that assert CommandExecutionError (#32485)
            - f8c0b439b8 Fixed more lint

      - 63ff731009 Fixed tests
      - 04b1a4a9ca Fixed use of assert_has_calls in tests.
      - 46e4bb58e5 Fixed LoadAuthTestCase
      - 4e9733ad6d Rename dockerio.py unit tests to dockerio_test.py
      - ec0cc943e0 Make sure spm tests are picked up by runtests.
      - 2605f34849 Fix missing first data in stream when subscribing stream using a function 'read_async'.
      - 305bab8be0 Fixed _interfaces_ifconfig output for SunOS test
      - b5ca02c867 Fix tests that assert CommandExecutionError (#32485)
      - 1fb6340fef Fix tests (#35693)
      - 5977f1f54c Skip utils_test if timelib is not installed (#32699)
      - d1b9a4061e Fixing skipped boto tests to prevent errors if boto3 does not exists.
    - c4ddfe3887 Merge pull request #35954 from morganwillcock/upgrade-on-batteries
      - 108f9470f2 win_pkg: report failure for failed launch of Scheduled Task
      - e0978220f7 win_pkg: allow minion upgrade when using batteries
    - 94b7659304 Merge pull request #36129 from terminalmage/pygit2-ssl_verify
      - 640f0c17c6 pygit2: Prevent traceback on initial gitfs setup
    - 7cdbc546f1 Back-port #36062 to 2015.8 (#36118)
  - **PR** #36170: (rallytime) Back-port #36154 to 2016.3
  - **PR** #36154: (DavidWittman) Remove unclosed backticks in walkthrough doc (refs: #36170)
- **ISSUE** #36055: (gladiatr72) 2016.3.3 – missing salt-cloud events on the master event bus (refs: #36161)
- **PR** #36161: (jacobhammons) Adds #36055 to release notes @ *2016-09-08 17:11:36 UTC*
  - 4ccf8a841f Merge pull request #36161 from jacobhammons/relnotes
  - ecb0979be7 Adds #36055 to release notes
- **PR** #36139: (meaksh) Fixing unit tests for 2016.3 @ *2016-09-08 13:20:21 UTC*
  - 1f909038f0 Merge pull request #36139 from meaksh/tests-fixes-for-2016.3
  - 52a7ed605e Fixed _interfaces_ifconfig output for SunOS test
  - 158bcbff65 Fix tests that assert CommandExecutionError (#32485)
  - 8b480167e1 Fix tests (#35693)
  - 29814f9d43 Skip utils_test if timelib is not installed (#32699)
  - d1d806f893 Fix PortageConfigTestCase in case of portage is not present
  - 1c260e4bd0 Fix tests to prevent errors when libcloud is not present
  - 71ebf2c8cd Fixing skipped boto tests to prevent errors if boto3 does not exists.
- **PR** #36143: (multani) doc: fix doc formatting for salt.states.mount @ *2016-09-08 13:11:03 UTC*
  - 3eb3df55ad Merge pull request #36143 from multani/fix-doc-state-mount
  - 035a212a9b doc: fix doc formatting for salt.states.mount

- **ISSUE** saltstack/salt#18419: (jasonrm) salt-cloud fails to run as non-root user (refs: #35483)
- **ISSUE** #36057: (Inveracity) Regression in opennebula cloud provider (refs: #36070)
- **ISSUE** #34806: (jerrykan) salt-cloud ignores sock_dir when firing event (refs: #35483)
- **PR** #36070: (rallytime) Use __utils__ instead of salt.utils.cloud in opennebula driver (refs: #36169) @ *2016-09-08 01:18:45 UTC*
  - **PR** #35483: (gtmanfred) use __utils__ in salt.cloud (refs: #35855, #37057, #36070)
  - 70da628018 Merge pull request #36070 from rallytime/fix-36057
  - de4f77cb68 Fixup failing test: need to mock __utils__ instead of salt.utils.cloud call
  - 25e3f2b4b8 Use __utils__ instead of salt.utils.cloud in opennebula driver
- **PR** #36089: (terminalmage) Support running git states / remote exec funcs as a different user in Windows @ *2016-09-08 01:17:23 UTC*
  - b7556a2aeb Merge pull request #36089 from terminalmage/issue35565
  - 796156c5f5 Add attribution
  - 2e56527ead Move command logging to before win_runas
  - 91eafddda6 Pass the ”password” param to git module functions
  - 7871065d32 Use ”user” instead of ”runas” in _git_run() helper
  - 5943b4662c Add ”password” param to funcs which support the user parameter
  - 5c7b9f0341 Make ”password” an explicit argument, not a kwarg
- **PR** #35923: (kstreee) Fixes a bug that Ctrl-c not working on Salt CLI. @ *2016-09-07 11:47:50 UTC*
  - 45ba2e806b Merge pull request #35923 from kstreee/fix-cli-stalling
  - 6569267afc Fixes a bug that Ctrl-c not working on Salt CLI.
- **ISSUE** #18341: (falzm) Dry-running state.highstate only returns the first change (refs: #36078)
- **PR** #36078: (thatch45) Failhard test=True fix @ *2016-09-07 05:10:35 UTC*
  - 48dc5ad4ee Merge pull request #36078 from thatch45/failhard_test
  - 9b36904149 Fix failhard causing test=True to failhard too soon
- **ISSUE** #34515: (vernondcole) Please actually implement skip_verify for archive.extracted (refs: #34529)
- **PR** #34529: (Ch3LL) Add skip_verify for archive.extracted @ *2016-09-06 21:05:31 UTC*
  - 40081176af Merge pull request #34529 from Ch3LL/add_skip_verify_archive
  - 38203e3d2c add tornado web app to serve up static file for test
  - 617f5680e4 add windows path and add custom tar
  - c5035118bf add skip_verify option to archive.extracted
- **PR** #36073: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-09-06 19:45:48 UTC*
  - fc41c744a0 Merge pull request #36073 from rallytime/merge-2016.3
  - e9c634685b Merge branch ’2015.8’ into ’2016.3’
  - fa09050150 consul: fix formatting of consul.agent_join (#36061)
  - **PR** saltstack/salt#36030: (whiteinge) Add include_* kwargs to the *_dict key functions (refs: #36040)
  - **PR** #36040: (rallytime) Add docs for new kwargs added to the wheel key module

  - **PR** #36047: (whiteinge) Doc cherrypy deemphasize urlencoded
- **PR** #36039: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-09-02 21:50:47 UTC*
  - 74143999d3 Merge pull request #36039 from rallytime/merge-2016.3
  - 5f499cfd41 Merge branch '2015.8' into '2016.3'
  - 1b39c7ed48 Merge pull request #35978 from DSRCorporation/bugs/28462_update_auth_data_on_reauth
    - 778ae9a9ff Update auth data on reauth.
  - b652271ddc Fix type error in networkfbsd osmajorrelease compare (#36016)
  - bc81818075 Merge pull request #36018 from meaksh/bp-36000-to-2015.8
    - 8c05d2aac5 Lint for #35916
    - b5fe6100ee Check for single quote before splitting on single quote
- **ISSUE** saltstack/salt#35683: (JensRantil) Salt wheel key documentation improvements (refs: #35824, #saltstack/salt'#35824'_)
  - **PR** saltstack/salt#35824: (rallytime) Add more documentation to the wheel key module (refs: #36038)
  - **PR** #36038: (rallytime) Back-port #35824 to 2016.3
  - **PR** #35824: (rallytime) Add more documentation to the wheel key module (refs: #36038)
  - **PR** #36033: (gtmanfred) catch unicode encoding errors in json outputter
  - **PR** #36010: (eliasp) modules.service: Do not default to OpenRC on Gentoo, also allow systemd
- **ISSUE** #33969: (Inveracity) Redis returner stacktrace in clean_old_jobs 2016.3.0 (refs: #33998)
  - **PR** #36014: (rallytime) Back-port #33998 to 2016.3
  - **PR** #33998: (jizhilong) fix redis_return's clean_old_jobs. (refs: #36014)
- **ISSUE** #35618: (komljen) [salt-cloud] With 'make_master: True' minions are configured with the masters public IP address on AWS (refs: #35919, #saltstack/salt'#35919'_)
  - **PR** saltstack/salt#35919: (rallytime) Add documentation about salt_interface to EC2 docs (refs: #36015)
  - **PR** #36015: (rallytime) Back-port #35919 to 2016.3
  - **PR** #35919: (rallytime) Add documentation about salt_interface to EC2 docs (refs: #36015)
  - **PR** saltstack/salt#36000: (rallytime) Lint #35916 (refs: #36019, #36018)
  - **PR** saltstack/salt#35916: (swiftgist) Check for single quote before splitting on single quote (refs: #36019, #36018)
- **PR** #36019: (meaksh) Back-port #36000 to 2016.3 @ *2016-09-02 20:34:30 UTC*
  - **PR** #36000: (rallytime) Lint #35916 (refs: #36019, #36018)
  - **PR** #35916: (swiftgist) Check for single quote before splitting on single quote (refs: #'saltstack/salt'#36000'_'_, #36000)
  - e88df5845d Merge pull request #36019 from meaksh/bp-36000-to-2016.3
  - 1b2abeabd1 Lint for #35916
  - 8b4f46fbd0 Check for single quote before splitting on single quote
  - **PR** #36028: (thatch45) Fix error when profiling is turned on and minions don't return (refs: #36730)
  - **PR** #36030: (whiteinge) Add include_* kwargs to the *_dict key functions

- **ISSUE** saltstack/salt#31454: (johje349) Salt Mine memory leak (refs: #36024)
- **PR** #36024: (DmitryKuzmenko) Don't subscribe to events if not sure it would read them. (refs: #36720) @ *2016-09-02 15:41:01 UTC*
  - cd60ec5d57 Merge pull request #36024 from DSRCorporation/bugs/31454_local_client_memleak
  - 01911c530e Don't subscribe to events if not sure it would read them.
- **PR** #36023: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-09-02 15:37:59 UTC*
  - 32d5f896d4 Merge pull request #36023 from rallytime/merge-2016.3
  - a63c9dfc6a Merge branch '2015.8' into '2016.3'
  - e6b93c2380 Merge pull request #36022 from saltstack/revert-33770-service_tests
    - 6cf56843d4 Revert "service state integration tests"
- **PR** #36004: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-09-01 21:11:06 UTC*
  - d248ab0120 Merge pull request #36004 from rallytime/merge-2016.3
  - 318bffed1d Merge branch '2015.8' into '2016.3'
  - 678f10cf8b Avoid traceback in mac_user.py when user.chhome is invoked from a user state (#35901)
  - 2da501071e Merge pull request #35967 from twangboy/improve_show_sls_2015.8
    - 2ed9a82ef8 Allow full path to be passed to show_sls
  - d86fba15b3 Merge pull request #35981 from cachedout/cptestcase_license
    - dd562dd200 Update Salt's licensing information to include cptestcase
- **PR** #35952: (twangboy) Load UserProfile when using RunAs (2016.3) @ *2016-09-01 15:18:15 UTC*
  - f7b85cb70b Merge pull request #35952 from twangboy/fix_win_runas_2016.3
  - 3721a09ea3 Load UserProfile on RunAs
- **PR** #35959: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-08-31 22:14:54 UTC*
  - b8ca3f5e4d Merge pull request #35959 from rallytime/merge-2016.3
  - bb4605ffee Merge branch '2015.8' into '2016.3'
  - 0f0f15d048 Merge pull request #35956 from jacobhammons/dot12
    - 3e21e35933 Version docs to 2015.8.12
  - d2db4ea7a2 cachedir should be /cloud not /master (#35897)
  - f4cdcc0d66 Better logging when file_recv_max_size is exceeded (#35914)
- **PR** #35955: (jacobhammons) Version docs to 2016.3.3 @ *2016-08-31 20:35:55 UTC*
  - a87b91a8ea Merge pull request #35955 from jacobhammons/dot3
  - ac8fe6ff9e Version docs to 2016.3.3
- **ISSUE** #875: (dhoffutt) state pkg won't install package nscd (refs: #35865)
  - **PR** #35865: (jacobhammons) Fix incremental doc builds - OS X, postgres returner, tcp transport doc updates
- **ISSUE** #35829: (amontalban) FreeBSD pkg.latest speed improvement (refs: #35904)
  - **PR** #35904: (amontalban) Fixes #35829 for branch 2016.3
  - **PR** #35931: (vutny) Salt Cloud: add *centos* default user for official CentOS AMIs

  - **PR** saltstack/salt#35892: (cachedout) Fixup Docker test (refs: #35926)
  - **PR** saltstack/salt#35581: (pbdeuchler) Correctly check if image is in current tags (refs: #35926)
  - **PR** #35926: (ticosax) [dockerng] Mention that docker image names must be given with repository
  - **PR** #35581: (pbdeuchler) Correctly check if image is in current tags (refs: #35926)
- **ISSUE** #35825: (tjyang) "'drac' __virtual__ returned False" from salt-run drac.version host (refs: #35868)
- **PR** #35868: (rallytime) Add more helpful return messages for drac runner @ *2016-08-31 01:33:27 UTC*
  - ca06c62900 Merge pull request #35868 from rallytime/fix-35825
  - 00ae17248e Update error message to be more helpful and fix doc formatting
  - 30a422bfe0 Add more helpful return messages for drac runner
- **PR** #35903: (rallytime) [2016.3] Merge forward from 2015.8 into 2016.3 @ *2016-08-30 17:15:36 UTC*
  - 95b89dbce9 Merge pull request #35903 from rallytime/merge-2016.3
  - 9e55bee5d5 Merge branch '2015.8' into '2016.3'
  - 08e10f69eb Clarifies how to create aliased functions (#35891)
  - 6dd5f68a08 Merge pull request #35856 from vutny/2015.8
    - eceedadfa5 salt-cloud: fix path to Salt Master socket dir
  - 336d1a700d Merge pull request #35880 from terminalmage/issue35747
    - 123a611066 pacman.py: Fix incorrect return in pkg.latest_version
  - 6383451c99 Merge pull request #35884 from terminalmage/clarify-pkg-latest-logic
    - b0b419d1d8 Fix condition for Gentoo USE flag update
    - 1542fd4716 Add clarifying comments to the pkg.latest state
- **ISSUE** saltstack/salt#18419: (jasonrm) salt-cloud fails to run as non-root user (refs: #35483)
- **ISSUE** #34806: (jerrykan) salt-cloud ignores sock_dir when firing event (refs: #35483)
- **PR** #35855: (vutny) [REGRESSION] salt-cloud: fix path to Salt Master socket dir (refs: #35856) @ *2016-08-30 07:09:04 UTC*
  - **PR** #35483: (gtmanfred) use __utils__ in salt.cloud (refs: #35855, #37057, #36070)
  - cf8f081401 Merge pull request #35855 from vutny/salt-cloud-fix-sock_dir
  - a662ea5337 salt-cloud: fix path to Salt Master socket dir
- **PR** #35881: (whiteinge) Add fail-safe in case Salt gives us data we can't serialize @ *2016-08-30 06:43:11 UTC*
  - f0987cf27a Merge pull request #35881 from whiteinge/salt-api-catch-serializer-error
  - 6e27fad21f Add fail-safe in case Salt gives us data we can't serialize
- **ISSUE** #35837: (JensRantil) Doc improvement: Mention engine under extension modules (refs: #35864)
  - **PR** #35864: (rallytime) Add engines to list of extension module options in master config docs
- **ISSUE** #35835: (JensRantil) Incorrect SQS config documentation statement (refs: #35861)
  - **PR** #35861: (rallytime) Fix IAM roles statement to be boto version specific in sqs_events
- **ISSUE** #35834: (JensRantil) Incorrect SQS engine config (refs: #35860)
  - **PR** #35860: (rallytime) Fix doc formatting for sqs_events engine example config

- **PR** #35859: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-08-29 18:21:04 UTC*
  - 96747bc3bb Merge pull request #35859 from rallytime/merge-2016.3
  - 5f93d682aa Merge branch '2015.8' into '2016.3'
  - eda2ae0add Merge pull request #35781 from thatch45/ssh_deploy_more
    - 2558dcc100 follow up on the re-deploy if there is a checksum missmatch
  - 165237412c Merge pull request #35815 from gtmanfred/2015.8
    - 805d43598e list_nodes_min should return a minimum dictionary
  - b12c6577d2 Merge pull request #35833 from terminalmage/2015.8-top-file-merging-docs
    - c534d88280 More clarification/correction in minion docs
    - e9e6ea8485 One more tweak to top file merging docs
- **ISSUE** #34478: (hujunya) makedir bug in the file module (refs: #35849)
- **PR** #35849: (theredcat) Fix potential infinite loop with no error when using recursive makedirs @ *2016-08-29 11:37:19 UTC*
  - dc705ff675 Merge pull request #35849 from theredcat/fix_file_makedirs_infinite_loop
  - 86d5398b28 Fix potential infinite loop with no error when using recursive makedirs
- **PR** #35682: (vutny) [BACKPORT] Fix empty *fun_agrs* field in Reactor generated events @ *2016-08-29 04:11:06 UTC*
  - **PR** #35659: (vutny) Fix empty *fun_agrs* field in Reactor generated events (refs: #35682)
  - **PR** #35059: (vutny) Add *fun_args* field to events generated by execution of Master modules (refs: #35659, #35682)
  - 433743f609 Merge pull request #35682 from vutny/backport-35659
  - 78d16a8057 [BACKPORT] Fix empty *fun_agrs* field in Reactor generated events
- **ISSUE** #34973: (szjur) Syndic stops forwarding job results if the local salt-master is restarted (refs: #35792)
- **PR** #35792: (DmitryKuzmenko) Reconnect syndic to event bus if master disappeared. @ *2016-08-29 02:13:19 UTC*
  - 30c2db7b09 Merge pull request #35792 from DSRCorporation/bugs/34973_syndic_reconnect_master_2016.3
  - 9afdbb0e97 Reconnect syndic to master event bus if master disappears.
  - ab1afd002e Fixed syndic event bus connection.
  - ea8e1385c1 Fixed syndic unhandled future exception if master is stopped.
- **PR** #35817: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-08-26 20:44:18 UTC*
  - 43c08ae431 Merge pull request #35817 from rallytime/merge-2016.3
  - e8e73b55ac Merge branch '2015.8' into '2016.3'
  - d285fe64b7 Merge pull request #35811 from rallytime/bp-35576
    - 04c063b315 Updated user.py to redact password when test=true
  - e212c55b7a Schedule documentation update (#35745)
  - eb4d2f299b Better unicode handling in gitfs (#35802)
  - 0ee237a9cb Remove extra "to" in top.rst docs (#35808)

  - 2fc61763d8 Correct the top_file_merging_strategy documentation (#35774)
  - **PR** #35788: (hu-dabao) fix 34241, webutil.useradd_all is deprecated
- **ISSUE** saltstack/salt#33536: (murzick) pkgrepo.managed does not disable a yum repo with "disabled: True" (refs: #35055)
- **ISSUE** #33536: (murzick) pkgrepo.managed does not disable a yum repo with "disabled: True" (refs: #35055, #35806)
  - **PR** #35810: (rallytime) Back-port #35806 to 2016.3
  - **PR** #35806: (rallytime) Bump the deprecation warning in pkgrepo state to Nitrogen (refs: #35810)
  - **PR** #35055: (galet) #33536 pkgrepo.managed does not disable a yum repo with "disabled: True" (refs: #35806)
- **ISSUE** #35741: (fix7) modjk: use of auth credentials to access jk-status broken (refs: #35796)
  - **PR** #35796: (fix7) Fix #35741
  - **PR** #35807: (jacobhammons) Adds mock for tornado.locks
  - **PR** #35800: (alexander-bauer) Trivial documentation spelling fix
- **PR** #35763: (isbm) Sphinx crash: documentation config fix @ *2016-08-25 21:12:39 UTC*
  - 9b5ee2155e Merge pull request #35763 from isbm/isbm-doc-conf-sphinx-crashfix
  - a56ae4e8f5 Configure importing Mock to handle 'total' method from psutils properly
  - 9c057d0266 Return psutil back to the list of mocked imports
  - 3d7758461e Improve Mock to be flexible and able to mock methods from the mocked modules
- **ISSUE** #35771: (bdrung) Spelling errors in salt 2016.3.2 (refs: #35773)
  - **PR** #35773: (rallytime) Documentation spelling fixes
- **PR** #35767: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-08-25 16:09:37 UTC*
  - e355c1cf90 Merge pull request #35767 from rallytime/merge-2016.3
  - 8ad6a12c80 Merge branch '2015.8' into '2016.3'
  - 2a12795bac Fixes Windows download paths (#35742)
- **ISSUE** #20575: (starchy) "salt –subset=n" appears to always choose the same nodes (refs: #35753)
- **PR** #35753: (rallytime) Fixup the unit.client_test.LocalClientTestCase.test_cmd_subset from #35720 @ *2016-08-25 15:55:23 UTC*
  - **PR** #35720: (hu-dabao) fix 20575, make subset really return random subset (refs: #35753)
  - b3f6367621 Merge pull request #35753 from rallytime/fix-client-unit-test
  - 92f8c836e8 Add cmd_mock back in to function spec
  - a671f0a092 Fixup the unit.client_test.LocalClientTestCase.test_cmd_subset from #35720
- **ISSUE** #35458: (iggy) SALT.STATES.APACHE_MODULE needs version annotations (refs: #35732)
  - **PR** #35732: (rallytime) Add versionadded for enabled function in apache_module state
- **PR** #35737: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-08-24 19:48:45 UTC*
  - bab0e3d449 Merge pull request #35737 from rallytime/merge-2016.3
  - 61e37d5956 Merge branch '2015.8' into '2106.3'

  - 06a75be8bd Merge pull request #35701 from gtmanfred/2015.8
    - 2d2bc1ffea use aws.get_location in s3 modules
  - 79bc01b88c Make test runs behave better (#35708)
- **PR** #35729: (cachedout) Remove docs mocks for msgpack and psutils @ *2016-08-24 14:42:06 UTC*
  - 7877ff1d5e Merge pull request #35729 from cachedout/fix_docs_build
  - fdbf01d5ad Remove docs mocks for msgpack and psutils
- **PR** #35628: (jf) Fix user.present state reporting for groups when remove_groups=false @ *2016-08-24 08:15:31 UTC*
  - 962e493304 Merge pull request #35628 from jf/fix_user.present_reporting_when_remove_groups=false
  - 1f818c832e Fix user.present state reporting for groups when remove_groups=false
- **PR** #35696: (xiaoanyunfei) fix maximum recursion depth bug @ *2016-08-24 08:01:16 UTC*
  - 02d86c6550 Merge pull request #35696 from xiaoanyunfei/2016.3
  - 5db9255926 fix maximum recursion depth
- **PR** #35720: (hu-dabao) fix 20575, make subset really return random subset (refs: #35753) @ *2016-08-24 07:03:58 UTC*
  - 79d10aea2d Merge pull request #35720 from hu-dabao/fix-20575
  - 70af980c01 fix 20575, make subset really return random subset
- **PR** #35700: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-08-23 17:23:40 UTC*
  - 5d0b9a248e Merge pull request #35700 from rallytime/merge-2016.3
  - 9e9923c3f4 Merge branch '2015.8' into '2016.3'
  - aee5b62542 Merge pull request #35680 from terminalmage/issue35630
    - d76659a63a Don't use six.text_type() in salt.utils.gitfs
  - 74678923b8 Fixup doc formatting for the sqs_events engine (#35663)
- **PR** #35634: (hu-dabao) fix 34922, StopIteration should not throw exception out @ *2016-08-23 08:13:08 UTC*
  - f305389172 Merge pull request #35634 from hu-dabao/fix-34922
  - fe338ff41f fix 34922, StopIteration should not throw exception out
- **PR** #35679: (twangboy) Revert to vcredist 12 (2013) @ *2016-08-23 08:05:40 UTC*
  - e45aa55d79 Merge pull request #35679 from twangboy/change.vcredist.version.2016.3
  - 3d6d473d48 Revert to vcredist 12 (2013)
- **PR** #35662: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-08-22 19:03:43 UTC*
  - 9fe0972761 Merge pull request #35662 from rallytime/merge-2016.3
  - 1d819d7cc2 Merge branch '2015.8' into '2016.3'
    - 399e9f57cc Update release notes for 2015.8.12 (#35614)
    - f7f8221169 Everything in the sample master config file should be commented out (#35611)
    - c9070c212f Merge pull request #35569 from rallytime/test-for-35384
      - 30f42d5352 Write test for multiple unless commands where 1st cmd passes and 2nd fails
  - **PR** #35661: (justinta) Backport #35627 to 2016.3

  - **PR** #35627: (cachedout) Comment boto lambda test (refs: #35661)
- **PR** #35615: (hu-dabao) fix 35591, verify the acl file exist before proceed @ *2016-08-21 04:41:32 UTC*
  - 67692f868c Merge pull request #35615 from hu-dabao/fix-35591
  - 402b83e4d3 change file verification to exist
  - 7355eb4ecd move python lib import after absolute_import
  - 69a2427670 fix 35591, verify the acl file exist before proceed
- **PR** #35485: (cro) Cassandra returner bugfixes and documentation. @ *2016-08-20 02:42:28 UTC*
  - de6fca3909 Merge pull request #35485 from cro/jpmc_cass_return
  - 0b01a7a266 Six import for range.
  - 7e87d4170d Fix Py3 lint?
  - d4336d011c [1,2,3] -> range(1,4)
  - cec7f6a7ec remove unneeded import
  - e31555345f Add timeout documentation.
  - 901ab8b74c Remove unnecessary log statements
  - 1954c1a3f3 Update cassandra returner for JPMC
- **ISSUE** #35519: (morganwillcock) win_dism state doesn't handle all success return codes (refs: #35520)
- **PR** #35520: (morganwillcock) Check for all success return codes in win_dism state @ *2016-08-20 02:35:01 UTC*
  - edefff51d4 Merge pull request #35520 from morganwillcock/dism-return-codes
  - 0b95b85e69 Check for all success return codes in dism state
- **PR** #35616: (xbglowx) Remove duplicate auth_tries in minion docs @ *2016-08-20 02:32:50 UTC*
  - 27211dbd64 Merge pull request #35616 from xbglowx/2016.3
  - 2801f0fdcc Remove duplicate auth_tries in minion docs
- **ISSUE** #34992: (szjur) Syndic strips vital parts of events (such as 'retcode' and 'success') (refs: #35552)
- **PR** #35552: (DmitryKuzmenko) Syndic fix: don't strip 'retcode' and 'success' from events. @ *2016-08-20 02:00:40 UTC*
  - 25ac9bacc6 Merge pull request #35552 from DSRCorporation/bugs/34992_syndic_strip_retcode
  - d036299f6f Syndic fix: don't strip 'retcode' and 'success' from events.
- **ISSUE** #25664: (sdm24) 2015.5.2 MySQL Returner: salt-run jobs.lookup_jid doesn't return full result for highstate output (refs: #35559)
- **PR** #35559: (Jlin317) Fix highstate outputter when it's given multiple results (refs: #36137) @ *2016-08-20 01:56:25 UTC*
  - bec8322e13 Merge pull request #35559 from Jlin317/fix_highstate_outputter
  - 27aa038cc6 Fix highstate outputter when it's given multiple results
- **ISSUE** #32478: (oliver-dungey) rsync.synchronized - user/group options required (refs: #32739)
- **PR** #35605: (rallytime) Back-port #32739 to 2016.3 @ *2016-08-20 01:39:38 UTC*
  - **PR** #32739: (abednarik) Rsync synchronized updates. (refs: #35605)
  - 4153aeba29 Merge pull request #35605 from rallytime/bp-32739

  - 36d8b4a409 Rsync synchronized updates.
- **PR** #35606: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-08-19 22:19:05 UTC*
  - 6eabe6356f Merge pull request #35606 from rallytime/merge-2016.3
  - f2eb625778 Merge branch '2015.8' into '2016.3'
  - 0c7aa802f5 Update release notes for 2015.8.12 (#35600)
  - dd12b48239 Update release notes for 2015.8.12 (#35599)
  - beb6ca8ef9 Update linux_sysctl tests to reflect new context key (#35584)
  - fd08d33597 Add warning about AWS flagging of nmap usage (#35575)

28.2.46 Salt 2016.3.5 Release Notes

Version 2016.3.5 is a bugfix release for *2016.3.0*.

Statistics

- Total Merges: **190**
- Total Issue References: **112**
- Total PR References: **281**
- Contributors: **74** (Ch3LL, DmitryKuzmenko, Firewire2002, Mrten, Talkless, TronPaul, UtahDave, aaronm-cloudtek, alex-zel, alexandr-orlov, alexbleotu, attiasr, basepi, bdrung, bshelton229, cachedout, calve, clan, clinta, cro, dere, dereckson, dhaines, dincamihai, do3meli, dragon788, edgan, fedusia, fj40crawler, genuss, gtmanfred, haeac, heewa, hu-dabao, jeanpralo, jfindlay, jinm, kevinquinnyo, kontrolld, laleocen, lorengordon, m03, mcalmer, mchugh19, meaksh, mikejford, moio, multani, nevins-b, pass-by-value, rallytime, rbjorklin, siccrusher, silenius, sjmh, sjorge, skizunov, slinn0, sofixa, techhat, tedski, terminalmage, thatch45, thusoy, toanju, tobithiel, twangboy, tyhunt99, vutny, wanparo, whiteinge, xiaoanyunfei, yhekma, zwo-bot)

Security Fixes

CVE-2017-5192 local_batch client external authentication not respected

The `LocalClient.cmd_batch()` method client does not accept `external_auth` credentials and so access to it from salt-api has been removed for now. This vulnerability allows code execution for already-authenticated users and is only in effect when running salt-api as the `root` user.

CVE-2017-5200 Salt-api allows arbitrary command execution on a salt-master via Salt's ssh_client

Users of Salt-API and salt-ssh could execute a command on the salt master via a hole when both systems were enabled.

We recommend everyone on the 2016.3 branch upgrade to a patched release as soon as possible.

Improved Checksum Handling in `file.managed`, `archive.extracted` States

When the `source_hash` argument for these states refers to a file containing checksums, Salt now looks for checksums matching the name of the source URI, as well as the file being managed. Prior releases only looked for checksums matching the filename being managed. Additionally, a new argument (`source_hash_name`) has been added, which allows the user to disambiguate ambiguous matches when more than one matching checksum is found in the `source_hash` file.

A more detailed explanation of this functionality can be found in the *file.managed* documentation, in the section for the new `source_hash_name` argument.

Changelog for v2016.3.4..v2016.3.5

Generated at: 2018-05-27 05:09:33 UTC

- **PR** #38833: (Ch3LL) add 2016.3.5 changelog to release notes @ *2017-01-19 23:27:26 UTC*
  - a04ab86da1 Merge pull request #38833 from Ch3LL/add_release_notes_2016.3.5
  - 374dc1ab88 skip 2016.3.5 due to :doc: references
  - 31f324c4ff add 2016.3.5 changelog to release notes
- **PR** #38812: (rallytime) Update pyobjects test to be a list @ *2017-01-18 21:06:01 UTC*
  - d14f0c64eb Merge pull request #38812 from rallytime/pyobjects-test
  - f3e84c1ab7 Update pyobjects test to be a list
- **ISSUE** #36598: (ikkaro) CloudClient vmware driver reusing SI bug (refs: #38813)
- **PR** #38813: (gtmanfred) catch SIGPIPE in vmware connection @ *2017-01-18 21:05:42 UTC*
  - 50f03f8057 Merge pull request #38813 from gtmanfred/2016.3
  - ce3472cec2 catch SIGPIPE in vmware connection
- **PR** #38809: (twangboy) Fix get_hostname to handle longer computer names @ *2017-01-18 19:32:00 UTC*
  - 23b8b47258 Merge pull request #38809 from twangboy/fix_hostname_2016.3
  - d57a51f9f9 Fix tests for get_hostname
  - 7ca3fd7484 Fix get_hostname to handle longer computer names
- **ISSUE** #38388: (johje349) No INFO logs in minion log file (refs: #38808)
- **PR** #38808: (vutny) Fix #38388 @ *2017-01-18 18:19:36 UTC*
  - 1033bbdde8 Merge pull request #38808 from vutny/fix-38388
  - 9bd203ffcc Fix #38388
- **ISSUE** #38604: (jsandas) Using "batch" with saltmod errors with "ValueError: need more than 2 values to unpack" (refs: #38668)
- **PR** #38668: (terminalmage) Fix proposal for #38604 @ *2017-01-18 17:53:09 UTC*
  - f3ae3cd5c8 Merge pull request #38668 from terminalmage/issue38604
  - 0ea97cdad9 Merge pull request #10 from cachedout/pr-38668
    - db81afc035 Munge retcode into return data for batching
  - a642a995dc Return the ret data from batch execution instead of raw data
- **ISSUE** #38622: (mikejford) Incorrect saltenv argument documentation in salt.modules.state (refs: #38789)
- **PR** #38789: (rallytime) Update some saltenv refs to environment in salt.modules.state docs @ *2017-01-18 15:39:22 UTC*
  - c6a19a9e5a Merge pull request #38789 from rallytime/fix-38622
  - af41fe0c6e Update some saltenv refs to environment in salt.modules.state docs
- **PR** #38790: (cachedout) Fix typo in pyobjects test @ *2017-01-18 15:38:57 UTC*

  - e0bf700020 Merge pull request #38790 from cachedout/fix_pyobjects_test_typo
  - a66afb5f0f Fix typo in pyobjects test
- **ISSUE** #38629: (Arabus) Conflicting documentation about default value of pillar_opts (refs: #38792)
- **PR** #38792: (rallytime) Update pillar tutorial lanuage regarding pillar_opts settings @ *2017-01-18 15:38:19 UTC*
  - 6e9785edea Merge pull request #38792 from rallytime/fix-38629
  - 1e125e2844 Update pillar tutorial lanuage regarding pillar_opts settings
  - **PR** saltstack/salt#38707: (alexbleotu) Fixed prepending of root_dir override to the other paths (refs: #38796)
- **PR** #38796: (cachedout) Revert "Fixed prepending of root_dir override to the other paths" @ *2017-01-17 23:18:18 UTC*
  - 3417adc617 Merge pull request #38796 from saltstack/revert-38707-root_dir_fix-gh
  - cb080f3bbe Revert "Fixed prepending of root_dir override to the other paths"
- **ISSUE** #38524: (rbjorklin) salt-api seems to ignore rest_timeout since 2016.11.0 (refs: #38585, #38527)
- **ISSUE** #38479: (tyeapple) api_logfile setting takes no effect (refs: #38585)
- **PR** #38585: (rallytime) Follow up to PR #38527 @ *2017-01-17 18:40:01 UTC*
  - **PR** #38570: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 (refs: #38585)
  - **PR** #38560: (Ch3LL) fix api logfile (refs: #38585)
  - **PR** #38527: (rbjorklin) salt-api no longer forces the default timeout (refs: #38585)
  - bab3479a3c Merge pull request #38585 from rallytime/follow-up-38527
  - 05587201b6 Pylint fix: add line at end of file
  - fa01367599 Keep a copy of the DEFAULT_API_OPTS and restore them after the test run
  - 2ad07634d9 Test clean up
  - fd2ee7db30 Add some simple unit tests for salt.config.api_config function
  - 3d2fefc83b Make sure the pidfile and log_file values are overriden by api opts
  - 1f6b540e46 Make sure the pidfile and log_file values are overriden by api opts
  - 04d307f917 salt-api no longer forces the default timeout
- **PR** #38707: (alexbleotu) Fixed prepending of root_dir override to the other paths @ *2017-01-17 15:40:13 UTC*
  - 0fb6bb7b77 Merge pull request #38707 from alexbleotu/root_dir_fix-gh
  - 0bac8c8be3 Fixed prepending of root_dir override to the other paths
- **PR** #38774: (vutny) DOCS: add C++ compiler installation on RHEL required for bundled 0mq @ *2017-01-17 15:21:00 UTC*
  - 96c9dc10f7 Merge pull request #38774 from vutny/dev-test-docs
  - 4620dc4afa DOCS: add C++ compiler installation on RHEL required for bundled 0mq
- **PR** #38749: (vutny) pkg build modules throw better exception message if keyid wasn't found @ *2017-01-17 02:13:08 UTC*
  - aedfbb7a43 Merge pull request #38749 from vutny/pkg-build-better-exception-msg
  - 53f2be5b21 pkg build modules throw better exception message if keyid wasn't found

- **PR** #38743: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2017-01-17 01:46:01 UTC*
  - 8466b34e82 Merge pull request #38743 from rallytime/merge-2016.3
  - d24776f5e9 Merge branch '2015.8' into '2016.3'
  - 6869621ed1 Merge pull request #38731 from rallytime/merge-2015.8
    - 9eb191b6ac Pylint fix
    - b910499dbe Various follow up fixes
    - e8309a6bbf Add release notes for 2015.8.13
    - f881f366b7 Merge pull request #20 from rallytime/2015.8.12_follow_up-batch-tests
      - 34282322c0 Clean up tests and docs for batch execution
    - c80b20b957 Merge pull request #19 from whiteinge/batchclient
      - 3d8f3d18f6 Remove batch execution from NetapiClient and Saltnado
    - 97b0f64923 Lintfix
    - d1516664f7 Add explanation comment
    - 62f2c87080 Add docstring
    - 9b0a786aeb Explain what it is about and how to configure that
    - 5ea3579e10 Pick up a specified roster file from the configured locations
    - 3a8614c5df Disable custom rosters in API
    - c0e5a1171d Add roster disable flag
  - e9c59e9b8f Merge pull request #38602 from terminalmage/fix-boto-test
    - 3424a108ac Fix failing unit.states.boto_vpc_test.BotoVpcRouteTableTestCase.test_present_with_routes
- **ISSUE** #38674: (jackywu) There is no code to use parameter 'event_publisher_pub_hwm' in saltstack-2016.3 (refs: #38723)
- **PR** #38723: (rallytime) Remove "event_publisher_pub_hwm" and "salt_event_pub_hwm" from config/__init__.py @ *2017-01-15 18:36:14 UTC*
  - **PR** #29294: (skizunov) ZeroMQ no longer required when transport is TCP (refs: #38723)
  - a642cdef79 Merge pull request #38723 from rallytime/fix-38674
  - 706c885f55 Remove "event_publisher_pub_hwm" and "salt_event_pub_hwm" from config/__init__.py
- **PR** #38669: (rallytime) Update bootstrap script verstion to latest release @ *2017-01-15 18:03:27 UTC*
  - fc545af10b Merge pull request #38669 from rallytime/update-bootstrap-script
  - 78ba76e34c Update bootstrap script verstion to latest release
- **PR** #38693: (twangboy) Update jinja2 to 2.9.4 @ *2017-01-15 14:40:46 UTC*
  - 50d417f267 Merge pull request #38693 from twangboy/update_jinja
  - e0c7e5549b Update jinja2 to 2.9.4
- **PR** #38739: (vutny) DOCS: correct examples of running test suite @ *2017-01-15 14:35:47 UTC*
  - f4233bb18d Merge pull request #38739 from vutny/fix-runtests-doc
  - b872bb63f6 DOCS: correct examples of running test suite
  - **PR** #38735: (vutny) DOCS: add links to File State Backups page where necessary

  - **PR** #38720: (dereckson) Proofread jinja_to_execution_module tutorial
- **ISSUE** #36548: (abonillasuse) openstack auth with nova driver (refs: #38647)
- **PR** #38647: (gtmanfred) Allow novaclient to use keystoneauth1 sessions for authentication @ *2017-01-10 17:48:26 UTC*
  - 7b850d472d Merge pull request #38647 from gtmanfred/nova
  - 5be9b60851 add documentation about using keystoneauth for v3
  - 7b657ca4ae add the ability to use keystone v2 and v3
  - 5646ae1b34 add ability to use keystoneauth to authenitcate in nova driver
- **ISSUE** #38648: (ericuldall) No release file error from PPA on Ubuntu (refs: #38650)
- **ISSUE** #38572: (COLABORATI) ppa:saltstack/salt failure (refs: #38650)
- **ISSUE** #34504: (AvinashDeluxeVR) Installation documentation for Ubuntu server and Windows minion leads the user to use different salt versions. (refs: #38650)
- **PR** #38650: (rallytime) Remove the installation instructions for out-of-date community ppa @ *2017-01-10 17:47:45 UTC*
  - 383768d838 Merge pull request #38650 from rallytime/remove-ubuntu-ppa-docs
  - 30429b2e44 Remove the installation instructions for out-of-date community ppa
- **ISSUE** #38087: (UtahDave) The 'data' field in the return from a minion below a syndic is wrapped in an extra 'data' field. (refs: #38657)
- **PR** #38657: (DmitryKuzmenko) Publish the 'data' field content for Syndic evets @ *2017-01-10 16:59:33 UTC*
  - 7d9f56e3b5 Merge pull request #38657 from DSRCorporation/bugs/38087_syndic_event_format_fix
  - 594c33f396 Publish the 'data' field content for Syndic evets
- **PR** #38649: (Ch3LL) fix unit.modules.file_test @ *2017-01-10 16:44:45 UTC*
  - 83987511fd Merge pull request #38649 from Ch3LL/test_apply_template
  - 47f8b68e0b fix unit.modules.file_test
- **ISSUE** #37355: (Firewire2002) salt-ssh - ImportError: No module named backports.ssl_match_hostname (refs: #38626, #'saltstack/salt'#37358'_'_, #37358)
- **ISSUE** #34600: (davidpsv17) Error trying a salt-ssh test.ping (refs: #'saltstack/salt'#37358'_'_, #37358)
- **ISSUE** #27355: (jerob) salt ssh error with debian 7 on target (refs: #'saltstack/salt'#37358'_'_, #37358)
  - **PR** saltstack/salt#37358: (Firewire2002) Fix/workaround for issue #37355 (refs: #38626)
- **PR** #38626: (cachedout) Revert "Fix/workaround for issue #37355" @ *2017-01-06 21:28:09 UTC*
  - 74ddc71be3 Merge pull request #38626 from saltstack/revert-37358-2016.3.3_issue37355
  - e912ac99c2 Revert "Fix/workaround for issue #37355"
- **ISSUE** #37355: (Firewire2002) salt-ssh - ImportError: No module named backports.ssl_match_hostname (refs: #38626, #'saltstack/salt'#37358'_'_, #37358)
- **ISSUE** #34600: (davidpsv17) Error trying a salt-ssh test.ping (refs: #'saltstack/salt'#37358'_'_, #37358)
- **ISSUE** #27355: (jerob) salt ssh error with debian 7 on target (refs: #'saltstack/salt'#37358'_'_, #37358)
- **PR** #37358: (Firewire2002) Fix/workaround for issue #37355 @ *2017-01-06 18:58:47 UTC*
  - 5e58b32934 Merge pull request #37358 from Firewire2002/2016.3.3_issue37355

  - 910da18bfd fixed typo
  - 4fbc5ddd06 fixed wrong renamed variable and spaces
  - 92366e646c issue #37355
  - 7dc87ab7b8 issue #37355
  - 2878180405 issue #37355
- **PR** #35390: (alexandr-orlov) Returns back missed proper grains dictionary for file module @ *2017-01-06 18:02:13 UTC*
  - 6c2fe615aa Merge pull request #35390 from alexandr-orlov/2016.3
  - cd5ae17e8d fxd missed proper grains dictionary
- **ISSUE** #38558: (multani) pillar.get("…", default=var, merge=true) updates default value (refs: #38579)
- **PR** #38618: (rallytime) Back-port #38579 to 2016.3 @ *2017-01-06 17:37:56 UTC*
  - **PR** #38579: (zwo-bot) Fix #38558 - pillar.get with default= …,merge=true influence subsequent calls of pillar.get (refs: #38618)
  - 2579cfa42d Merge pull request #38618 from rallytime/bp-38579
  - 2052ecee2c Add copy import
  - 2c8845aaa0 add test for pillar.get() + default value
  - c2f98d2f04 ticket 38558: add unit test, deepcopy() only if necessary
  - 30ae0a1958 added deepcopy of default if merge=True
- **PR** #38601: (terminalmage) pillar.get: Raise exception when merge=True and default is not a dict @ *2017-01-05 23:15:51 UTC*
  - da676cebd6 Merge pull request #38601 from terminalmage/pillar-get
  - 8613d7254d pillar.get: Raise exception when merge=True and default is not a dict
- **PR** #38600: (terminalmage) Avoid errors when sudo_user is set (2016.3 branch) @ *2017-01-05 20:57:09 UTC*
  - **PR** #38598: (terminalmage) Avoid errors when sudo_user is set (refs: #38600)
  - 224fc7712a Merge pull request #38600 from terminalmage/issue38459-2016.3
  - 8a45b13e76 Avoid errors when sudo_user is set
- **PR** #38589: (tobithiel) State Gem: fix incorrect warning about missing rvm/rbenv @ *2017-01-05 20:12:15 UTC*
  - a376970f88 Merge pull request #38589 from tobithiel/fix_rvm_rbenv_warning
  - 9ec470b4a5 State Gem: fix incorrect warning about missing rvm/rbenv
- **PR** #38567: (pass-by-value) Create queue if one doesn't exist @ *2017-01-05 18:46:11 UTC*
  - 02e6a78254 Merge pull request #38567 from pass-by-value/pgjsonb_queue_changes_2016.3
  - 67879ebe65 Create queue if one doesn't exist
- **ISSUE** #37498: (githubcdr) service.restart salt-minion fails on Ubuntu 14.04.5 LTS (refs: #37748, #38587)
- **PR** #38587: (rallytime) Change daemontools __virtualname__ from service to daemontools @ *2017-01-05 18:06:01 UTC*
  - 0889cbdb31 Merge pull request #38587 from rallytime/fix-37498
  - 2a5880966f Change daemontools __virtualname__ from service to daemontools

- **PR** #38562: (rallytime) Update arch installation docs with correct package name @ *2017-01-04 20:04:28 UTC*
  - 7b74436d13 Merge pull request #38562 from rallytime/arch-install-docs
  - 8b1897ace9 Update arch installation docs with correct package name
- **PR** #38560: (Ch3LL) fix api logfile (refs: #38585) @ *2017-01-04 19:03:17 UTC*
  - 01860702cb Merge pull request #38560 from Ch3LL/fix_api_log
  - 1b45e9670b fix api logfile
- **PR** #38531: (rallytime) Back-port #33601 to 2016.3 @ *2017-01-04 16:56:53 UTC*
  - **PR** #33601: (mchugh19) Fix slack engine to run on python2.6 (refs: #38531)
  - 0056620a53 Merge pull request #38531 from rallytime/bp-33601
  - c36cb39825 remove the unnecessary double trigger
  - 38414493bf fix spacing lint error
  - 8c1defc710 Remove uncessary type from alias commands. Deduplicate alias handling to autodetect function selection. Add error reporting to slack connectivty problems. Cleanup slack's unicode conversion
  - c2f23bc45e Fix slack engine to run on python2.6
- **ISSUE** #38187: (curiositycasualty) username/password saved as cleartext when using URIs with user:pass@ format (refs: #38541)
- **PR** #38541: (techhat) Strip user:pass from cached URLs @ *2017-01-04 15:39:57 UTC*
  - 50242c7f17 Merge pull request #38541 from techhat/issue38187
  - eae3a435dd Strip user:pass from cached URLs
- **ISSUE** #30454: (favoretti) Using yaml serializer inside jinja template results in unicode being prepended by '!!python/unicode' (refs: #30481, #38554)
- **PR** #38554: (multani) Fix YAML deserialization of unicode @ *2017-01-04 15:31:16 UTC*
  - **PR** #30481: (basepi) Add yaml_safe jinja filter (refs: #38554)
  - 325dc56e59 Merge pull request #38554 from multani/fix/30454
  - 2e7f743371 yaml: support unicode serialization/deserialization
  - df76113c5c jinja: test the "yaml" filter with ordered dicts
  - f7712d417f Revert "Add yaml_safe filter"
  - **PR** #38536: (UtahDave) add note about pyVmomi locale workaround
- **ISSUE** #38353: (Ch3LL) salt-cloud gce specifying (refs: #38542)
  - **PR** #38542: (Ch3LL) fix gce image bug
- **ISSUE** #38449: (swalladge) Parsing issues in *list_tab* (salt/modules/cron.py) (refs: #38487)
- **PR** #38487: (gtmanfred) Fix crontab issues with spaces @ *2017-01-01 20:33:29 UTC*
  - ec60f9c721 Merge pull request #38487 from gtmanfred/2016.3
  - 048b9f6b9d add test
  - c480c11528 allow spaces in cron env
  - c529ec8c34 allow crons to have multiple spaces
- **ISSUE** #37684: (thusoy) State execution duration is timezone-dependent (refs: #38491)

- **PR** #38491: (gtmanfred) Use UTC for timing in case timezone changes @ *2017-01-01 20:30:57 UTC*
    - c5ba11b5e0 Merge pull request #38491 from gtmanfred/timing
    - 79368c7528 Use UTC for timing in case timezone changes
- **ISSUE** #38472: (jinm) file.managed Unable to manage file: 'hash_type' (2016.3.4) (refs: #38503)
- **PR** #38503: (jinm) Hash type fallback for file management @ *2017-01-01 17:36:51 UTC*
    - 86f0aa0bb3 Merge pull request #38503 from jinm/issue_38472_jinm
    - 0cd9df299f Hash type fallback for file management
- **PR** #38457: (bshelton229) Stops git.latest checking for local changes in a bare repo @ *2016-12-30 14:28:47 UTC*
    - ed2ba4bd1b Merge pull request #38457 from bshelton229/git-latest-head-bug
    - 558e7a771a Stops git.latest checking for local changes in a bare repo
- **PR** #38385: (dragon788) Use unambigous long names with double dashes @ *2016-12-29 17:10:48 UTC*
    - 36e21b22cb Merge pull request #38385 from dragon788/2016.3-double-dash
    - 86c4b56f47 Newline for lint compat
    - 9d9b686057 Address review comments, consistency of quotes
    - df9bd5e7f9 Use unambigous long names with double dashes
- **ISSUE** #38209: (limited) Accepting a minion causes tornado to exit (refs: #38474)
- **PR** #38474: (cachedout) Allow an existing ioloop to be passed to salt-key @ *2016-12-29 16:28:51 UTC*
    - 59f2560d88 Merge pull request #38474 from cachedout/key_loop
    - de504538e1 Allow an existing ioloop to be passed to salt-key
- **ISSUE** #38438: (jf) file.line with mode=delete breaks on empty file (refs: #38467)
- **PR** #38467: (gtmanfred) file.line fail with mode=delete @ *2016-12-28 20:00:33 UTC*
    - 3d0c752acd Merge pull request #38467 from gtmanfred/2016.3
    - 7b7c6b3878 file.line fail with mode=delete
- **PR** #38434: (slinn0) Make sysctl.persist fail when failing to set a value into the running kernel @ *2016-12-27 15:37:53 UTC*
    - 940025d5c4 Merge pull request #38434 from slinn0/issue_38433_fixes
    - 22af87a3fc Fixes for https://github.com/saltstack/salt/issues/38433
    - **PR** #38421: (rallytime) Update deprecation notices to the correct version
    - **PR** #38420: (rallytime) Removed various deprecation notices from salt/modules/* files (refs: #38421)
- **ISSUE** #38282: (sash-kan) file.managed fails when file (which contains utf-characters in the name) exists (refs: #38415)
    - **PR** #38415: (terminalmage) file.managed: Fix failure when filename contains unicode chars
- **PR** #38419: (Ch3LL) fix scsci docs example @ *2016-12-22 18:57:51 UTC*
    - 2cdb59d055 Merge pull request #38419 from Ch3LL/fix_doc_scsi
    - 234043b8bb fix scsci docs example
    - **PR** #38407: (terminalmage) Improve pillar documentation
- **ISSUE** #38372: (fanirama) Issue with cron.file. Source: salt://path/to/crontab_file not found (refs: #38398)

- **PR** #38398: (terminalmage) Fix call to file.get_managed in cron.file state @ *2016-12-22 16:46:14 UTC*
  - 423b1fdddf Merge pull request #38398 from terminalmage/issue38372
  - c80dbaa914 Fix call to file.get_managed in cron.file state
  - **PR** #38382: (heewa) Fix http.query when result has no text
- **PR** #38390: (meaksh) Add "try-restart" to fix autorestarting on SUSE systems @ *2016-12-21 16:06:24 UTC*
  - b74b5c7d38 Merge pull request #38390 from meaksh/2016.3-fix-try-restart-for-autorestarting-on-SUSE-systems
  - de6ec05ec0 add try-restart to fix autorestarting on SUSE systems
- **PR** #38221: (UtahDave) Fix default returner @ *2016-12-20 20:34:36 UTC*
  - 2c3a39760a Merge pull request #38221 from UtahDave/fix_default_returner
  - 385640765b remove a blank line to satisfy linter
  - 9c248aa14c validate return opt, remove default.
  - 8bb37f9fe7 specify allowed types and default for "returner"
  - 11863a4bfe add examples of default minion returners
  - e7c6012655 add support for default returners using *return*
- **PR** #38288: (terminalmage) archive.extracted: don't try to cache local sources (2016.3 branch) @ *2016-12-18 13:07:11 UTC*
  - 09d9cff992 Merge pull request #38288 from terminalmage/archive-extracted-local-source-2016.3
  - 845e3d0e75 Update tests to reflect change in cache behavior
  - 5a08d7c70a archive.extracted: don't try to cache local sources (2016.3 branch)
- **PR** #38312: (cro) Backport feature allowing proxy config to live in pillar OR /etc/salt/proxy @ *2016-12-18 12:39:01 UTC*
  - bf37667f8a Merge pull request #38312 from cro/proxy_config_in_cfg
  - 2006c4000e Typo
  - 689d95b10f Backport feature allowing proxy config to live in pillar OR /etc/salt/proxy.
- **ISSUE** #12788: (whiteinge) Comb through docs to replace :doc: roles with :ref: (refs: #38320)
- **PR** #38320: (rallytime) Cleanup doc internal markup references @ *2016-12-18 12:31:28 UTC*
  - c83db5a785 Merge pull request #38320 from rallytime/cleanup-doc-refs
  - 62978cb7a0 Don't check the doc/conf.py file for doc markup refs
  - 770e732d76 Add a unit test to search for new doc markup refs
  - 5c42a361a0 Remove ":doc:" references from all doc/topics/installation/* files
  - 23bce1c929 Remove ":doc:" references from all doc/topics/releases/* files
  - 4aafa41d22 Remove ":doc:" references from a bunch of doc/* files
  - 02bfe7912c Remove more ":doc:" references from doc/* files
  - 6e32267d0c Remove ":doc:" references in salt/* files
  - **PR** #38281: (mikejford) Add nick to args for create_multi
- **ISSUE** #38290: (dragon788) Need to use machine automation friendly output (refs: #38313)

- **PR** #38313: (dragon788) 2016.3 chocolatey fix @ *2016-12-16 17:20:39 UTC*
  - 235682b1e6 Merge pull request #38313 from dragon788/2016.3-chocolatey-fix
  - 1f5fc17551 Use machine readable output for list
  - cdbd2fbe3c Added limit-output to eliminate false packages
- **ISSUE** #38174: (NickDubelman) [syndic] Why can't a syndic node signal when all of it's minions have returned? (refs: #38279)
- **ISSUE** #32400: (rallytime) Document Default Config Values (refs: #38279)
- **PR** #38279: (rallytime) Add docs for syndic_wait setting @ *2016-12-15 18:30:31 UTC*
  - 9e78ddc80e Merge pull request #38279 from rallytime/fix-38174
  - 4a62d01577 Add docs for syndic_wait setting
- **PR** #38248: (meaksh) Successfully exit of salt-api child processes when SIGTERM is received @ *2016-12-15 09:16:27 UTC*
  - fc9e1dff35 Merge pull request #38248 from meaksh/salt-api-successfully-close-child-processes
  - ee6eae9855 Successfully exit of salt-api child processes when SIGTERM.
- **PR** #38254: (terminalmage) Also check if pillarenv is in opts @ *2016-12-15 09:10:24 UTC*
  - 3c718ed35e Merge pull request #38254 from terminalmage/check-pillarenv
  - fa9ad311c6 Also check if pillarenv is in opts
  - **PR** #38256: (rallytime) [2016.3] Bump latest release version to 2016.11.1
- **PR** #38198: (vutny) Add missing requirements for running unit tests: libcloud and boto3 @ *2016-12-13 14:12:20 UTC*
  - 004e46afe7 Merge pull request #38198 from vutny/unit-tests-require-libcloud-boto3
  - a6098bac1a Remove note about SaltTesting installation, now it is in the requirements
  - 004bff113e Add missing requirements for running unit tests: libcloud and boto3
- **PR** #38213: (rallytime) Skip test_cert_info tls unit test on pyOpenSSL upstream errors @ *2016-12-13 12:05:01 UTC*
  - 9d497bc74c Merge pull request #38213 from rallytime/skip-tls-test
  - bdb807fc7c Skip test_cert_info tls unit test on pyOpenSSL upstream errors
- **PR** #38224: (whiteinge) Allow CORS OPTIONS requests to be unauthenticated @ *2016-12-13 12:02:30 UTC*
  - 203109dd17 Merge pull request #38224 from whiteinge/cors-options-unauthed
  - de4d3227ab Allow CORS OPTIONS requests to be unauthenticated
- **PR** #38223: (whiteinge) Add root_dir to salt-api file paths @ *2016-12-13 07:44:19 UTC*
  - **PR** #37272: (vutny) Get default logging level and log file from default opts dict (refs: #38223)
  - 721a5feccd Merge pull request #38223 from whiteinge/salt-api-root_dirs
  - bfbf390c0e Add root_dir to salt-api file paths
- **ISSUE** #38162: (747project) git_pillar does not detect changes to remote repository when told to update (refs: #38191)
- **PR** #38191: (terminalmage) Clarify the fact that git_pillar.update does not fast-forward @ *2016-12-12 09:45:48 UTC*

  - 70f7d22ad6 Merge pull request #38191 from terminalmage/issue38162
  - 1ae543a98a Clarify the fact that git_pillar.update does not fast-forward
- **PR** #38194: (vutny) Document the requirements for running ZeroMQ-based integration tests @ *2016-12-12 09:42:11 UTC*
  - 28171cbfc5 Merge pull request #38194 from vutny/integration-test-requirements-doc
  - e9f419ff64 Document the requirements for running ZeroMQ-based integration tests
- **PR** #38185: (rallytime) Back-port #38181 to 2016.3 @ *2016-12-09 22:27:44 UTC*
  - **PR** #38181: (rallytime) Reset socket default timeout to None (fixes daemons_tests failures) (refs: #38185)
  - a4ef037ab1 Merge pull request #38185 from rallytime/bp-38181
  - 609f814454 Reset socket default timeout to None (fixes daemons_tests failures)
- **PR** #38163: (Ch3LL) enabled ec2 cloud tests @ *2016-12-09 18:01:57 UTC*
  - 65b2ad7b14 Merge pull request #38163 from Ch3LL/enabled_ec2_cloud
  - be74c45463 enabled ec2 cloud tests
- **PR** #38177: (vutny) Correct *cp.get_file_str* docstring and add integration tests @ *2016-12-09 16:55:35 UTC*
  - b63f74e034 Merge pull request #38177 from vutny/fix-cp-get-file-str
  - a449980672 Correct *cp.get_file_str* docstring and add integration tests
- **PR** #38153: (vutny) Master config includes may contain errors and be safely skipped @ *2016-12-08 17:43:34 UTC*
  - 7596313be0 Merge pull request #38153 from vutny/master-includes-error-tolerance
  - cd0154ee93 Master config includes may contain errors and be safely skipped
  - **PR** #38134: (rallytime) Skip daemon unit tests when running on Python 2.6
- **ISSUE** #38091: (tjyang) [WARNING] salt.loaded.int.module.zenoss.__virtual__() is wrongly returning *None.* (refs: #38102)
- **PR** #38102: (rallytime) Add False + msg tuple return if requests is missing for zenoss module @ *2016-12-07 13:24:37 UTC*
  - d3d98fd4eb Merge pull request #38102 from rallytime/fix-38091
  - 4f79d5a0d1 Add False + msg tuple return if requests is missing for zenoss module
- **ISSUE** #36707: (do3meli) slow FreeBSD sysctl module with test=true (refs: #36794)
- **PR** #38104: (rallytime) Back-port #36794 to 2016.3 @ *2016-12-07 13:23:48 UTC*
  - **PR** #36794: (do3meli) FreeBSD sysctl module now handels config_file parameter in show method (refs: #38104)
  - 8c8cbc2734 Merge pull request #38104 from rallytime/bp-36794
  - c906c8a0d5 Pylint fixes
  - da3ebf83e6 FreeBSD sysctl module now handels config_file parameter in show method
- **ISSUE** #35342: (morganwillcock) win_pkg: refresh_db doesn't remove cached items which have been renamed or removed (refs: #38083)
- **PR** #38083: (twangboy) Only delete .sls files from winrepo-ng [DO NOT MERGE FORWARD] @ *2016-12-06 14:13:35 UTC*

  - fbc87769b9 Merge pull request #38083 from twangboy/fix_refresh_db
  - 978af6d83c Remove only .sls files from the cached winrepo-ng
- **PR** #38059: (rallytime) Call exec_test for the Syndic daemon in tests.unit.daemons_test.py @ *2016-12-04 04:18:41 UTC*
  - **PR** #38057: (rallytime) [2016.11] Merge forward from 2016.3 to 2016.11 (refs: #38059)
  - **PR** #38034: (cachedout) Modify daemons test to use multiprocessing (refs: #38059)
  - 9dcfdeef6b Merge pull request #38059 from rallytime/daemons-test-fix
  - eb372b27d8 Add missing "not" statement: The last syndic test should assertFalse()
  - 4e10f8e018 Call exec_test for the Syndic daemon in tests.unit.daemons_test.py
- **ISSUE** #37939: (Talkless) file.comment always report changes in test=True mode (refs: #38039)
- **PR** #38039: (rallytime) Check to see if a line is already commented before moving on @ *2016-12-02 20:08:35 UTC*
  - 9cd42b9b3f Merge pull request #38039 from rallytime/fix-37939
  - 1da7aacfbe Update unit tests to account for additional file.search call
  - 8a685b1820 Check to see if a line is already commented before moving on
  - f2c045520d Write an integration test demonstrating the issue
- **ISSUE** #38037: (dmurphy18) pkg.latest and yumpkg.latest_version return incorrect package versions 2016.3 and 2016.11 (refs: #38045)
- **PR** #38045: (terminalmage) yumpkg.py: don't include non-upgrade versions found by "yum list available" @ *2016-12-02 20:07:38 UTC*
  - a34a763984 Merge pull request #38045 from terminalmage/issue38037
  - 65289503d9 Simplify logic for matching desired pkg arch with actual pkg arch
  - 3babbcda94 yumpkg.py: don't include non-upgrade versions found by "yum list available"
  - **PR** #38034: (cachedout) Modify daemons test to use multiprocessing (refs: #38059)
- **PR** #37995: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-11-30 20:12:55 UTC*
  - 6942d5d95b Merge pull request #37995 from rallytime/merge-2016.3
  - b44e17921c Merge branch '2015.8' into '2016.3'
    - 7a7e36728f Merge pull request #37978 from terminalmage/ext_pillar_first-docs
      - 61ed9a8657 Add clarifying language to ext_pillar_first docs
  - **PR** #38002: (laleocen) fix broken yaml code block
- **ISSUE** #35088: (Modulus) salt/cloud/ec2.py encoding problems. (refs: #37912)
- **PR** #37912: (attiasr) fix encoding problem aws responses @ *2016-11-30 18:10:30 UTC*
  - 3dd45fbedf Merge pull request #37912 from attiasr/fix_aws_response_encoding
  - ba4ec4e7f1 use Requests result encoding to encode the text
  - abe4eb3b98 fix encoding problem aws responses
- **PR** #37950: (vutny) Set default Salt Master address for a Syndic (like for a Minion) @ *2016-11-30 18:09:04 UTC*
  - 69a74a4d2d Merge pull request #37950 from vutny/fix-starting-up-syndic

  - 7d9bc9abce syndic_master: correct default value, documentation and example config
  - 92a7c7ed1b Set default Salt Master address for a Syndic (like for a Minion)
  - **PR** #37964: (terminalmage) Add clarification on expr_form usage and future deprecation
- **ISSUE** #37867: (tobiasBora) Bug into lsb_release that crash salt (refs: #37962)
  - **PR** #37962: (cachedout) Catch possible exception from lsb_release
- **ISSUE** #37945: (gstachowiak) Missing exception handling in salt.master.Maintenance. Process never completes. (refs: #37961)
  - **PR** #37961: (cachedout) Handle empty tokens safely
- **PR** #37272: (vutny) Get default logging level and log file from default opts dict (refs: #38223) @ *2016-11-28 23:04:20 UTC*
  - ea46639ce7 Merge pull request #37272 from vutny/fix-getting-default-logging-opts
  - e5ce52388a Fix description in the Salt Syndic usage info
  - 518a3dd7ee Add unit tests for Salt parsers processing logging options
  - 83d6a44254 Add *ssh_log_file* option to master config and documentation
  - c8a0915460 Fix configuration example and documentation for *syndic_log_file* option
  - e64dd3ed6b Correct default attributes for various parser classes
  - 82a2e216b3 Fix default usage string for Salt command line programs
  - 45dffa292f Fix readding and updating logfile and pidfile config options for Salt API
  - f47253c21b Fix reading and applying Salt Cloud default configuration
  - fad5bec936 Work with a copy of default opts dictionaries
  - b7c24811e5 Fix *log_level_logfile* config value type
  - 1bd76a1d96 Fix setting temporary log level if CLI option omitted
  - 121848cc77 Fix obtaining *log_granular_levels* config setting
  - 44cf07fec2 Make CLI options take precedence for setting up logfile_logger
  - 61afaf1792 Fix setting option attributes when processing *log_level* and *log_file*
  - 3c60e2388e Fix processing of *log_level_logfile* config setting
  - 55a0af5bbd Use attribute functions for getting/setting options and config values
  - c25f2d091e Fix getting Salt API default logfile option
  - f2422373c1 Remove processing of unused and undocumented *cli_*_log_** config options
  - 2065e8311c Get default logging level and file from default opts dict
- **PR** #37925: (kontrolld) Fix missing ipv6 options centos network @ *2016-11-28 22:38:43 UTC*
  - f2f957da6c Merge pull request #37925 from kontrolld/add-ipv6-centos-network
  - ac2b477412 Adding IPv6 functionality for CentOS /etc/sysconfig/network
- **ISSUE** #37059: (basepi) Beacon fileserver operations cause scheduled jobs with fileserver operations to hang (refs: #37899)
- **PR** #37899: (DmitryKuzmenko) Clear functions context in schedule tasks for ZeroMQ. @ *2016-11-28 22:23:45 UTC*

  - c07ad11279 Merge pull request #37899 from DSRCorporation/bugs/37059_schedule_task_hang
  - 9497748546 Clear functions context in schedule tasks for ZeroMQ.
- **ISSUE** #37737: (b-harper) python client api CloudClient multiple calls needed (refs: #37928)
- **PR** #37928: (techhat) Don't modify self.opts directly @ *2016-11-28 21:07:40 UTC*
  - a55519db40 Merge pull request #37928 from techhat/issue37737
  - a09a60e89b Don't modify self.opts directly
- **PR** #37929: (gtmanfred) add list_nodes_min to nova driver @ *2016-11-28 21:05:40 UTC*
  - 9d17f1ce90 Merge pull request #37929 from gtmanfred/2016.3
  - c7d2c73503 add list_nodes_min to nova driver
- **PR** #37926: (kontrolld) Fixes no IPv6 functionality in /etc/sysconfig/network @ *2016-11-28 20:40:00 UTC*
  - 3bb743b59f Merge pull request #37926 from kontrolld/fix-ipv6-centos-network
  - 3ed42e5b44 updated
  - 3b3bc4f239 Fixes no IPv6 functionality in /etc/sysconfig/network
- **PR** #37921: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-11-28 19:54:40 UTC*
  - 271170a9f3 Merge pull request #37921 from rallytime/merge-2016.3
  - 523a67c422 Merge branch '2015.8' into '2016.3'
    - 4cdc6cf5ec Update earlier release channels' docs with Carbon release notes (#37914)
    - d31491a7fe [2015.8] Update version numbers in doc config for 2016.11.0 release (#37918)
- **PR** #37924: (cachedout) Update test for new gem ver @ *2016-11-28 18:17:53 UTC*
  - 6cd6429ac0 Merge pull request #37924 from cachedout/fix_gem_states
  - 894cca3427 Update test for new gem ver
- **PR** #37916: (rallytime) [2016.3] Update version numbers in doc config for 2016.11.0 release @ *2016-11-28 17:09:08 UTC*
  - c35ba1f390 Merge pull request #37916 from rallytime/doc-update-2016.3
  - bd40592289 [2016.3] Update version numbers in doc config for 2016.11.0 release
- **ISSUE** #37287: (aaronm-cloudtek) salt.states.ddns.present: 'NS' record type always returns as changed (refs: #37785)
- **PR** #37785: (aaronm-cloudtek) respect trailing dot in ddns name parameter @ *2016-11-28 14:02:10 UTC*
  - e13a2488c8 Merge pull request #37785 from Cloudtek/ddns-respect-trailing-dot
  - 262e3b3697 respect trailing dot in ddns name parameter
- **ISSUE** #37870: (fj40crawler) salt.states.augeas.change returns None when test=True (refs: #37895)
- **PR** #37895: (fj40crawler) Change return value for salt/states/augeas.py to be True instead of N… @ *2016-11-28 13:49:27 UTC*
  - c03b389422 Merge pull request #37895 from fj40crawler/fix-augeas-return-for-test
  - ddc238df36 Fixed augeas_test.py to match True v.s. None for test_change_in_test_mode
  - ef75c459c0 Merge branch '2016.3' of github.com:saltstack/salt into fix-augeas-return-for-test

  - b0fe0cd256 Change return value for salt/states/augeas.py to be True instead of None for cases where salt is run with test=True. Fixes #37870
- **PR** #37907: (Talkless) Fix server trust in test run of svn.latest @ *2016-11-28 13:47:39 UTC*
  - fdbc31e8d8 Merge pull request #37907 from Talkless/patch-2
  - 072a319490 Fix server trust in test run of svn.latest
- **PR** #37896: (toanju) rh networking: add missing values @ *2016-11-27 10:30:35 UTC*
  - f39fdf443f Merge pull request #37896 from toanju/2016.3
  - c95304188e rh networking: add missing values
- **PR** #37886: (bdrung) Fix various spelling mistakes @ *2016-11-25 02:59:36 UTC*
  - ea935c5a91 Merge pull request #37886 from bdrung/fix-typos
  - 9a51ba5c5b Fix various spelling mistakes
- **ISSUE** #37732: (dhaines) list_semod() (from modules/selinux.py) incompatible with policycoreutils-2.5 (RHEL 7.3) (refs: #37736)
- **PR** #37736: (dhaines) handle semodule version >=2.4 (#37732) and fix typo @ *2016-11-24 01:44:20 UTC*
  - 371b0a86d9 Merge pull request #37736 from dhaines/issue-37732
  - 7ef590a505 Update selinux.py
  - 516a67e6a3 fix indexing error
  - 4e49c1e991 fix typo
  - b16f2d8400 handle semodule version >=2.4 (#37732) and fix typo
- **PR** #37797: (clan) check count of columns after split @ *2016-11-24 01:28:59 UTC*
  - 87aeb66fbf Merge pull request #37797 from clan/extfs
  - acf0f960ef check count of columns after split
- **PR** #37762: (twangboy) Add pre_versions to chocolatey.installed @ *2016-11-24 01:27:29 UTC*
  - f7c7109152 Merge pull request #37762 from twangboy/fix_chocolatey_state
  - 9696b6dfa5 Use keyword args instead of relying on ordering
  - 398eaa074d Add pre_versions to the available arguments
- **PR** #37866: (meaksh) Backport #37149 #36938 and #36784 to 2016.3 @ *2016-11-23 21:54:17 UTC*
  - **PR** #37857: (meaksh) Backport #37149 and #36938 to 2015.8 (refs: #37866)
  - **PR** #37856: (meaksh) Backport #36784 to 2015.8 (refs: #37866)
  - **PR** #37149: (dincamihai) Fix pkg.latest_version when latest already installed (refs: #37857, #37866)
  - **PR** #36938: (wanparo) acl.delfacl: fix position of -X option to setfacl (refs: #37857, #37866)
  - **PR** #36784: (moio) OS grains for SLES Expanded Support (refs: #37856, #37866)
  - 56baa92d55 Merge pull request #37866 from meaksh/2016.3-bp-37149-36938-36784
  - 9d8d578109 Fix pkg.latest_version when latest already installed
  - ffca0d491c - acl.delfacl: fix position of -X option to setfacl
  - 3dfed6b841 Adjust linux_acl unit test argument ordering
  - f185ecdde1 core.py: quote style fixed

  - 8404d13424 Setting up OS grains for SLES Expanded Support (SUSE's Red Hat compatible platform)
- **ISSUE** #32829: (tyhunt99) Dockerng appears to not be using docker registries pillar data (refs: #36893)
- **PR** #37863: (rallytime) Back-port #36893 to 2016.3 @ *2016-11-23 17:09:09 UTC*
  - **PR** #36893: (tyhunt99) add option to force a reauth for a docker registry (refs: #37863)
  - d0cc7f0d56 Merge pull request #37863 from rallytime/bp-36893
  - 4c70534991 Add versionadded to reauth option in dockerng module
  - 5ca2c388c2 added documentation for the new reuth option in docker registry configuration
  - 5b0c11ab47 add option to force a reauth for a docker registry
  - **PR** #37847: (laleocen) add multiline encryption documentation to nacl
- **ISSUE** #37787: (elyulka) user.present state fails to change loginclass on FreeBSD (refs: #37827)
  - **PR** #37827: (silenius) add missing chloginclass
  - **PR** #37826: (rallytime) Update branch refs to more relevant branch
  - **PR** #37822: (laleocen) add documenation for multiline encryption using nacl (refs: #37826)
- **ISSUE** #19269: (markuskramerIgitt) Undocumented feature *names:* of *file.directory* (refs: #37823)
  - **PR** #37823: (rallytime) Add "names" option to file state docs: point users to highstate doc examples
- **ISSUE** #15697: (arthurlogilab) keystone.user_present should not re-set the password when user exists (refs: #37821)
  - **PR** #37821: (rallytime) Clarify keystone.user_present password state docs with default behavior
- **ISSUE** #5999: (pille) libvirt.keys does not work (refs: #37820)
  - **PR** #37820: (rallytime) Add some dependency documentation to libvirt docs
- **PR** #37772: (bdrung) Support initializing OpenSSL 1.1 @ *2016-11-21 20:28:51 UTC*
  - 485270f74e Merge pull request #37772 from bdrung/openssl1.1
  - 819c9658ed Support initializing OpenSSL 1.1
- **ISSUE** #37383: (edwardsdanielj) Orchestration arguments (kwarg) not being interperted / How I learned to stop worrying about documentation and love experimenting (refs: #37817)
  - **PR** #37817: (rallytime) Update orchestrate runner file.copy doc example
- **ISSUE** #37653: (gravyboat) Salt.cron docs don't wrap @hourly and @daily correctly in quotes for the examples (refs: #37816)
- **ISSUE** #31953: (sjorge) Documentation for salt.states.cron is incorrect (refs: #32157)
- **PR** #37816: (rallytime) Back-port #32157 to 2016.3 @ *2016-11-21 20:22:27 UTC*
  - **PR** #32157: (cachedout) Add quotes to cron doc (refs: #37816)
  - c5d3d8b66a Merge pull request #37816 from rallytime/bp-32157
  - d9c297119e Add quotes to cron doc
- **PR** #37812: (rallytime) Back-port #37790 to 2016.3 @ *2016-11-21 18:46:40 UTC*
  - **PR** #37790: (sofixa) Update cloud/proxmox.rst with more options and LXC (refs: #37812)
  - 97e6b6aabe Merge pull request #37812 from rallytime/bp-37790
  - ca3b6e7874 Update proxmox.rst with more options and LXC

- **ISSUE** #37751: (freach) Documentation salt.states.dockerng.running: "privileged" property undocumented (refs: #37789)
- **PR** #37811: (rallytime) Back-port #37789 to 2016.3 @ *2016-11-21 18:46:21 UTC*
  - **PR** #37789: (fedusia) issue: 37751 (refs: #37811)
  - 27703c54bc Merge pull request #37811 from rallytime/bp-37789
  - ba3fef48e1 fix comment
  - a021f76a9b issue: 37751 Add documentation for option privileged
- **PR** #37810: (rallytime) Back-port #37775 to 2016.3 @ *2016-11-21 18:45:53 UTC*
  - **PR** #37775: (calve) Document *python* argument in *salt.states.virtualenv_mod* (refs: #37810)
  - adac9d7c0c Merge pull request #37810 from rallytime/bp-37775
  - 2bed91437b Document *python* argument in *salt.states.virtualenv_mod*
- **ISSUE** #37742: (blaketmiller) Cannot match on nodegroup when checking minions (refs: #37763)
  - **PR** #37763: (cachedout) Add nodegroup check to ckminions
- **ISSUE** #37725: (secumod) salt-call incorrectly parses master hostname:port from minion config (refs: #37766)
  - **PR** #37766: (cachedout) Fix ip/port issue with salt-call
- **ISSUE** #33709: (msummers42) Any/All Salt-SSH invocations in 2016.3.0 Fails with AttributeError: 'module' object has no attribute 'BASE_THORIUM_ROOTS_DIR' (refs: #37767)
  - **PR** #37767: (cachedout) Add thorium path to syspaths
- **PR** #37760: (hu-dabao) Fix couchbase returner and add couple of more features @ *2016-11-18 00:28:23 UTC*
  - bff949f4e9 Merge pull request #37760 from hu-dabao/fix_cb_returner
  - de372f277e 1. returner no need to check whether the jid exists for external job cache setup 2. add full_ret to return doc so that the document will be informative 3. make ttl as a config attribute because salt-minion does not have keep_jobs attribute 4. add password into config attribute 5. update the documents accordingly
- **ISSUE** #36629: (yhekma) The pillar run module does not honor saltenv (refs: #37738)
- **PR** #37738: (terminalmage) Allow pillar.get to retrieve fresh pillar data when saltenv passed @ *2016-11-17 23:13:04 UTC*
  - 1f976ac212 Merge pull request #37738 from terminalmage/issue36629
  - da46678c51 Allow pillar.get to retrieve fresh pillar data when saltenv passed
  - **PR** #37745: (cro) Switch default filter tag for ONE resources from user only to all resources
- **ISSUE** #37498: (githubcdr) service.restart salt-minion fails on Ubuntu 14.04.5 LTS (refs: #37748, #38587)
  - **PR** #37748: (silenius) check for SERVICE_DIR in __virtual__ in salt.modules.daemontools
- **ISSUE** #37734: (Ch3LL) Joyent Cloud Size Issue (refs: #37735)
- **PR** #37735: (Ch3LL) change size and image of joyent profile @ *2016-11-16 21:07:52 UTC*
  - fa7883115e Merge pull request #37735 from Ch3LL/fix_joyent_profile
  - 9ef41dcdfc change size and image of joyent profile
- **PR** #37731: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-11-16 17:13:02 UTC*
  - 98e25c68aa Merge pull request #37731 from rallytime/merge-2016.3

  - ec1389711f Merge branch '2015.8' into '2016.3'
    - f417dbbe99 Merge pull request #37718 from terminalmage/docs
      - 23b8b2a3f4 Fix incorrectly-formatted RST
  - **PR** #37724: (cachedout) Warn on AES test for systems with > 1 core
- **PR** #37721: (terminalmage) Fix for pillar setting 'environment' key in __gen_opts() @ *2016-11-16 16:04:53 UTC*
  - 35655d521f Merge pull request #37721 from terminalmage/zd909
  - acdd5513da Update git_pillar docs to reflect info from bugfix
  - 433737d2dc Fix for pillar setting 'environment' key in __gen_opts()
- **PR** #37719: (terminalmage) Fix incorrectly-formatted RST (2016.3 branch) @ *2016-11-16 08:20:53 UTC*
  - 99cda7c003 Merge pull request #37719 from terminalmage/docs-2016.3
  - f163b4c724 Fix incorrectly-formatted RST
- **PR** #37694: (cachedout) Catch differences in git URLs in npm state @ *2016-11-16 01:56:18 UTC*
  - 8dea695c7c Merge pull request #37694 from cachedout/npm_git
  - 0e3bc2366a Catch differences in git URLs in npm state
- **ISSUE** #37665: (kluoto) boto_elb state fails as key is overwritten by the code (refs: #37705)
- **PR** #37705: (rallytime) Don't overwrite the "key" variable passed in to _listeners_present func @ *2016-11-15 21:26:37 UTC*
  - 329448ccd7 Merge pull request #37705 from rallytime/fix-37665
  - 3b7e9c5e3b Don't overwrite the "key" variable passed in to _listeners_present func
- **PR** #37707: (Ch3LL) add timeout increase on azure tests @ *2016-11-15 21:24:25 UTC*
  - **PR** #37239: (Ch3LL) Fix cloud tests timeout (refs: #37707)
  - ac9a316b50 Merge pull request #37707 from Ch3LL/fix_timeout_azure
  - 363122c675 add timeout increase on azure tests
- **PR** #37704: (twangboy) Fix test disabled 2016.3 [DO NOT MERGE FORWARD] @ *2016-11-15 16:48:52 UTC*
  - 1ece265354 Merge pull request #37704 from twangboy/fix_test_disabled_2016.3
  - a0429cf839 Use nfsd instead of apsd for test_disabled
- **PR** #37690: (twangboy) Update pyzmq to 15.3.0 for 2016.3 [DO NOT MERGE FORWARD] @ *2016-11-15 03:10:36 UTC*
  - 44f05acbff Merge pull request #37690 from twangboy/update_pyzmq_2016.3
  - cf55342150 Update pyzmq to version 15.3.0
- **PR** #37680: (rallytime) Back-port #32965 to 2016.3 @ *2016-11-15 02:56:46 UTC*
  - **PR** #32965: (kevinquinnyo) Fix 'present' option when used without 'key_type' (refs: #37680)
  - a743d8b5e6 Merge pull request #37680 from rallytime/bp-32965
  - 1865b13645 Fix 'present' option when used without 'key_type'
- **ISSUE** #35964: (edgan) salt-ssh doesn't set the return code to non-zero on highstate rendering error (refs: #35965)
- **PR** #37681: (rallytime) Back-port #35965 to 2016.3 @ *2016-11-14 21:19:22 UTC*

  - **PR** #35965: (edgan) Set the return code to 1 on salt-ssh highstate errors (refs: #37681)
  - 1c2d6ff293 Merge pull request #37681 from rallytime/bp-35965
  - 700f3fa57f Set the return code to 1 on salt-ssh highstate errors
- **PR** #37668: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-11-14 15:43:25 UTC*
  - 1b456b55dc Merge pull request #37668 from rallytime/merge-2016.3
  - ef684c6b02 Merge branch '2015.8' into '2016.3'
  - a01b66556f Add docs for rotate_aes_key (#37641)
- **ISSUE** #37492: (JensRantil) Failing *salt -b 1 minion state.highstate* has wrong exit code (refs: #37625)
- **PR** #37625: (cachedout) Return with proper retcodes in batch mode @ *2016-11-12 20:29:09 UTC*
  - 305e51d1c0 Merge pull request #37625 from cachedout/issue_37492
  - b6031524e5 Return with proper retcodes in batch mode
- **ISSUE** #34547: (sebw) salt-cloud deployment fails when deploy: True (refs: #37607)
- **PR** #37639: (rallytime) Back-port #37607 to 2016.3 @ *2016-11-11 20:29:20 UTC*
  - **PR** #37607: (techhat) Try the connection again, in case it's been reset (refs: #37639)
  - **PR** #35673: (cro) Proxies don't handle reusing the SmartConnect instances very well. D… (refs: #37607)
  - **PR** #34059: (alexbleotu) Vmware common gh (refs: #37607)
  - 7510cd4da9 Merge pull request #37639 from rallytime/bp-37607
  - 9914c93bc4 Pylint: Remove kwargs that are not in the 2016.3 branch
  - d941e9354d Disable pylint warning
  - 940ee49a0b Lint fix
  - 69893f0c38 Try the connection again, in case it's been reset
- **ISSUE** saltstack/salt#37118: (gtmanfred) group in file.find module unable to be a list (refs: #37349)
- **ISSUE** #37118: (gtmanfred) group in file.find module unable to be a list (refs: #37349)
- **PR** #37638: (rallytime) Back-port #37349 to 2016.3 @ *2016-11-11 20:29:01 UTC*
  - **PR** #37349: (haeac) Pull request for Bug #37118 (refs: #37638)
  - 24ca96010d Merge pull request #37638 from rallytime/bp-37349
  - ba2105bc39 Fix for Bug #37118, the wrong parameter was being used to convert the group name to group id.
- **ISSUE** #37643: (Ch3LL) digital ocean list_keypairs limits to 20 keys (refs: #37644)
- **PR** #37644: (Ch3LL) digital ocean list_keypairs: increase limit for ssh keys parsed @ *2016-11-11 20:28:46 UTC*
  - e1e8b81d16 Merge pull request #37644 from Ch3LL/fix_37643
  - c02961a2f5 list_keypairs: increase limit for ssh keys parsed
- **ISSUE** #37541: (yhekma) salt-minion does not clean up temp files for templates (refs: #37540, #37640)
- **PR** #37640: (rallytime) Add known issue #37541 to 2016.3.4 release notes @ *2016-11-11 20:28:12 UTC*
  - a97c2ad34b Merge pull request #37640 from rallytime/update-release-notes
  - 6d6de12aff Grammatical fix

  - 24d7f20e16 Add known issue #37541 to 2016.3.4 release notes
- **PR** #37642: (cro) Forward-port change from 2015.8 adding release note for rotate_aes_key @ *2016-11-11 20:27:07 UTC*
  - fab3eaa237 Merge pull request #37642 from cro/rotate_aes_doc
  - 1ca5b958c6 Forward-port change from 2015.8 adding release note for rotate_aes_key
- **ISSUE** #37628: (TronPaul) [git 2016.3] Refreshing of an s3 file server results in an exception (refs: #37629)
- **PR** #37629: (TronPaul) fix __opts__ and provider being None in salt.utils.aws:get_location @ *2016-11-11 09:49:47 UTC*
  - 4c07b3534a Merge pull request #37629 from TronPaul/fix-s3fs-opts
  - a452cded20 fix __opts__ and provider being None issue
- **PR** #37481: (thatch45) Raet internal client reference fix @ *2016-11-11 04:39:41 UTC*
  - 200d9fcb6e Merge pull request #37481 from thatch45/raet_client
  - 50d911160b Attempted fix, needs user verification
- **PR** #37611: (jeanpralo) Fix cmd batch raw @ *2016-11-11 02:53:58 UTC*
  - b14faf1c68 Merge pull request #37611 from jeanpralo/fix-cmd-batch-raw
  - 4f16840ef1 add integration test for salt.client.LocalClient.cmd_batch
  - ead47e4bba update ret dict to avoid hanging
  - 0a2f153b6e fix dict key for raw support to avoid exception
- **PR** #37614: (gtmanfred) remove redundant code @ *2016-11-11 02:49:13 UTC*
  - 35c8333d04 Merge pull request #37614 from gtmanfred/2016.3
  - 71c2df89a9 remove redundent code
- **PR** #37627: (cachedout) Exempt pip.iteritems from test_valid_docs test @ *2016-11-11 02:48:37 UTC*
  - 4fab707bdd Merge pull request #37627 from cachedout/pr-36706
  - 94df2f8e6f Exempt pip.iteritems from test_valid_docs test
- **ISSUE** #36644: (b1naryth1ef) env_vars not properly validated/casted to strings w/ virtualenv.manage/pip.install (refs: #36706)
- **PR** #36706: (siccrusher) Add basic sanity checks for env_vars in pip.install function @ *2016-11-11 02:47:16 UTC*
  - ee74f3116e Merge pull request #36706 from siccrusher/fix_env_var_validation
  - fb27f8b69e Revert change
  - 79f3e83f8d Use fully-qualified path for six
  - 0ca1222833 Update pip.py
  - b15de371c1 * Ensure src is python3 compatible
  - 0976a2d1ae * Before passing on the env_vars dictionary ensure all values are strings. Fixes #36644
- **ISSUE** #37491: (JensRantil) "Failed to authenticate! …" error should exit non-zero (refs: #37626)
- **PR** #37626: (cachedout) Exit with proper retcode on hard client failures @ *2016-11-11 02:38:47 UTC*
  - 902a97575e Merge pull request #37626 from cachedout/issue_37491
  - bab9a729b1 Exit with proper retcode on hard client failures

- **PR** #37617: (terminalmage) Clarify docs for git_pillar dynamic env feature @ *2016-11-11 01:52:52 UTC*
    - 845f835177 Merge pull request #37617 from terminalmage/git_pillar-docs
    - 8cdf5dbb51 Clarify docs for git_pillar dynamic env feature
- **PR** #36627: (sjmh) Skip rest of loop on failed func match @ *2016-11-10 23:47:12 UTC*
    - 3079d78332 Merge pull request #36627 from sjmh/fix/auth_skip_nomatch
    - b3baaf30d0 Skip rest of loop on failed func match
- **PR** #37600: (mcalmer) change TIMEZONE on SUSE systems (bsc#1008933) @ *2016-11-10 21:54:04 UTC*
    - a71e7c77b3 Merge pull request #37600 from mcalmer/fix-timezone-on-SUSE
    - 3530b542f0 change TIMEZONE on SUSE systems (bsc#1008933)
- **ISSUE** #37238: (cmclaughlin) Restarting master causes minion to hang (refs: #37438, #37602)
- **ISSUE** #37018: (tsaridas) get events from python (refs: #37438, #37602)
- **PR** #37602: (DmitryKuzmenko) Handle master restart in appropriate places using *salt.event* listener. @ *2016-11-10 21:53:20 UTC*
    - **PR** #37438: (DmitryKuzmenko) Fix for #37238 salt hang on master restart (refs: #37602)
    - 39b75878cf Merge pull request #37602 from DSRCorporation/bugs/37238_salt_hang_on_master_restart
    - d3d987b19c Handle master restart in appropriate places using *salt.event* listener.
- **PR** #37608: (gtmanfred) allow multiline returns from docker for mac @ *2016-11-10 21:48:59 UTC*
    - 019e1a721b Merge pull request #37608 from gtmanfred/2016.3
    - 74aee1e372 allow multiline returns from docker for mac
- **ISSUE** #37592: (craigafinch) State git.latest does not work with SSH (refs: #37604)
- **ISSUE** #37551: (viict) git.latest "Not a valid commit name" (refs: #37604, #37571)
    - **PR** #37604: (terminalmage) Documentation improvements and corrections
    - **PR** #37579: (pass-by-value) Use existing VM's VDD size if not specified in the cloud profile
- **ISSUE** #37541: (yhekma) salt-minion does not clean up temp files for templates (refs: #37540, #37640)
- **PR** #37540: (yhekma) Added prefix to tempfile for template @ *2016-11-10 00:37:18 UTC*
    - fdd13b4145 Merge pull request #37540 from yhekma/2016.3
    - 93a59f8034 Added prefix to tempfile for template
- **ISSUE** #37084: (aaronm-cloudtek) x509.certificate_managed does not work with m2crypto >=0.25 (refs: #37578)
    - **PR** #37578: (clinta) Update for m2crypto changes removing lhash
    - **PR** #37584: (clinta) Fix eauth example for limiting args
- **ISSUE** #37551: (viict) git.latest "Not a valid commit name" (refs: #37604, #37571)
    - **PR** #37571: (terminalmage) Add a test to ensure we don't check for fast-forward before fetching
- **ISSUE** #33645: (ketzacoatl) saltutil.sync_all does not sync custom pillar modules to masterless minions (refs: #33833)
- **ISSUE** #25297: (Akilesh1597) perform 'refresh_pillar' as a part of 'sync_all' (refs: #25361, #37521)
- **PR** #37553: (rallytime) Back-port #37521 to 2016.3 @ *2016-11-08 23:11:07 UTC*

  - **PR** #37521: (genuss) refresh_pillar() should be called always with refresh=True during saltutil.sync_all (refs: #37553)
  - **PR** #33833: (terminalmage) Support syncing pillar modules to masterless minions (refs: #37521)
  - **PR** #25361: (tedski) perform *refresh_pillar* as part of *sync_all* when *refresh=True* (refs: #37521)
  - b01c247ea9 Merge pull request #37553 from rallytime/bp-37521
  - 30f92b05f4 refresh_pillar() should be called always
  - **PR** saltstack/salt#37549: (Mrten) sqlite is not found in 2015.8 (refs: #37565)
- **PR** #37565: (rallytime) Back-port #37549 to 2016.3 @ *2016-11-08 23:10:25 UTC*
  - **PR** #37549: (Mrten) sqlite is not found in 2015.8 (refs: #37565)
  - 694df30d40 Merge pull request #37565 from rallytime/bp-37549
  - c92a90b8e5 Update sqlite3.py
  - fb76557a2a sqlite is not found in 2015.8
- **ISSUE** #37511: (jdelic) service.dead now only operates if the service file exists (refs: #37562)
  - **PR** #37562: (terminalmage) Fix regression in service.dead state
- **ISSUE** #37554: (sjmh) salt-api doesn't dynamically re-read nodegroups configuration (refs: #37560)
  - **PR** #37560: (whiteinge) Skip config type checking for sdb values
  - **PR** #37556: (rallytime) Don't pass the vpc id to boto.vpc.create_internet_gateway func
  - **PR** #37543: (multani) Documentation rendering fixes
- **ISSUE** saltstack/salt#31081: (JensRantil) salt.modules.file.line documentation unclarities (refs: #37457)
- **PR** #37457: (rallytime) Fixup file.line docs to be more clear and consistent @ *2016-11-08 00:29:20 UTC*
  - 96b8b9a849 Merge pull request #37457 from rallytime/fix-31081
  - 25821bb8db Clarify which modes use "before", "after", and "indent" options
  - 8b2d2b9e7b Clarify file.line state docs as well
  - b2615892eb Move note about using mode=insert with location options to mode section
  - db0b0cefb8 Fixup file.line docs to be more clear and consistent
- **ISSUE** #35799: (davegiles) dsc.apply_config hangs (no error) on empty directory on target (refs: #37526)
- **PR** #37526: (twangboy) Remove loop from dsc.apply_config @ *2016-11-08 00:23:11 UTC*
  - 7de790ffed Merge pull request #37526 from twangboy/fix_35799
  - fc4260911c Remove unnecessary format
  - c934a2bfa7 Remove the loop from apply_config
  - **PR** saltstack/salt#37515: (rallytime) [carbon] Merge forward from 2016.3 to carbon (refs: #37534)
- **PR** #37534: (rallytime) Back-port fix needed from #37515 @ *2016-11-08 00:14:46 UTC*
  - **PR** #37515: (rallytime) [carbon] Merge forward from 2016.3 to carbon (refs: #37534)
  - 94811df2ea Merge pull request #37534 from rallytime/bp-merge-foward-fix
  - d1b2af1d69 Add missing source_hash_name args to a couple funcs
- **PR** #37533: (whiteinge) Return a 504 response instead of 500 for Salt timeouts @ *2016-11-08 00:14:15 UTC*

  - 17adbb0c9f Merge pull request #37533 from whiteinge/salt-api-504-timeouts
  - 63226aeda6 Return a 504 response instead of 500 for Salt timeouts
- **ISSUE** saltstack/salt#36679: (lorengordon) Command 'Import-Module ServerManager' failed with return code: 1 (refs: #'saltstack/salt'#36736'_'_, #36736)
  - **PR** saltstack/salt#36736: (m03) Fix issue 36679 win_servermanager error (refs: #37529)
- **PR** #37529: (lorengordon) Backport: PR 36736 to 2016.3 @ *2016-11-08 00:04:10 UTC*
  - **PR** #36736: (m03) Fix issue 36679 win_servermanager error
  - a9f03eee6f Merge pull request #37529 from lorengordon/bp-36736
  - 21c2664b6a Fix issue 36679 win_servermanager failure
- **ISSUE** #37444: (Tanoti) Returning False from __virtual__ in a returner does not return expected error (refs: #'saltstack/salt'#37502'_'_, #37519, #37502)
  - **PR** saltstack/salt#37502: (cachedout) Log proper message on returners that cannot be loaded (refs: #37519)
- **PR** #37519: (rallytime) Update returner __virtual__() return messages for loader @ *2016-11-07 23:06:23 UTC*
  - 19475aada6 Merge pull request #37519 from rallytime/returner-load-errors
  - fb261a31f3 Update returner __virtual__() return messages for loader
- **ISSUE** #35016: (pingangit) TypeError: save_minions() got an unexpected keyword argument 'syndic_id' (refs: #37527)
- **PR** #37527: (rallytime) Add syndic_id=None kwarg to save_minions funcs in returners @ *2016-11-07 23:04:03 UTC*
  - fefdfab850 Merge pull request #37527 from rallytime/fix-35016
  - 2944b244aa Add syndic_id=None kwarg to save_minions funcs in returners
  - **PR** #37530: (gtmanfred) fix Lithium to 2015.5.0
- **PR** #37514: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-11-07 16:51:06 UTC*
  - 743164844d Merge pull request #37514 from rallytime/merge-2016.3
  - 41166aede4 Merge branch '2015.8' into '2016.3'
  - c505a059ef [2015.8] Doc version updated to 2016.3.4 (#37482)
- **ISSUE** #36713: (Tanoti) ExtraData: unpack(b) received extra data after upgrading to 2016.3.3 (refs: #37503)
- **PR** #37503: (cachedout) Catch loader error on returners without save_load @ *2016-11-07 09:33:57 UTC*
  - 2d924d0820 Merge pull request #37503 from cachedout/issue_36713
  - 5f7f971b2c Catch loader error on returners without save_load
- **ISSUE** #37448: (alisson276) In 'salt/key' events there are acts that never happen (refs: #37499)
- **PR** #37499: (cachedout) Clarify docs on salt-key events @ *2016-11-07 09:33:20 UTC*
  - d95bf59f97 Merge pull request #37499 from cachedout/key_docs_clarify
  - 2758e74785 Clarify docs on salt-key events
- **PR** #37500: (cachedout) Remove unused flag @ *2016-11-07 09:33:04 UTC*
  - 1dd1408ae6 Merge pull request #37500 from cachedout/remove_include_errors
  - 6c705b11e0 Remove unused flag

- **ISSUE** #37444: (Tanoti) Returning False from __virtual__ in a returner does not return expected error (refs: #‘saltstack/salt‘#37502‘_‘_, #37519, #37502)
- **PR** #37502: (cachedout) Log proper message on returners that cannot be loaded @ *2016-11-07 09:32:45 UTC*
  - 4b6f1ab1c4 Merge pull request #37502 from cachedout/issue_37444
  - 4c5ab057ce Remove debugging
  - 17d01e4f4c Log proper message on returners that cannot be loaded
- **ISSUE** #37389: (d101nelson) Some core grains are inaccurate or incomplete for Solaris (refs: #37472)
- **PR** #37494: (sjorge) Forgot to update os_family map in #37472 @ *2016-11-06 22:18:54 UTC*
  - **PR** #37472: (sjorge) 2016.3 solaris grains improvements (refs: #37494)
  - 2422dafd52 Merge pull request #37494 from sjorge/2016.3-osfam_map
  - 96ba545492 Forgot to update os_family map in #37472
- **PR** #37496: (mcalmer) fix status handling in sysv init scripts @ *2016-11-06 22:18:00 UTC*
  - 41bd8e3f52 Merge pull request #37496 from mcalmer/fix-status-handling-in-sysv-init-scripts
  - 1fb2c4dfcf fix status handling in sysv init scripts
- **PR** #37497: (terminalmage) Update 2016.3.5 release notes with source_hash_name explanation @ *2016-11-06 22:17:40 UTC*
  - e741a773a5 Merge pull request #37497 from terminalmage/release_notes
  - c08038d9ea Update 2016.3.5 release notes with source_hash_name explanation
- **PR** #37486: (twangboy) Add requirement for PowerShell 3 on Windows @ *2016-11-06 06:01:07 UTC*
  - f4426c2233 Merge pull request #37486 from twangboy/fix_win_docs
  - 9e0631a1ae Add docs denoting the requirement for at least PowerShell 3
- **PR** #37493: (cachedout) Add sdb support to minion and master configs @ *2016-11-06 06:00:18 UTC*
  - a1f355a569 Merge pull request #37493 from cachedout/minion_master_sdb
  - 9761a462c2 Add sdb support to minion and master configs
- **ISSUE** #31135: (jeffreyctang) file.line mode=replace breaks on empty file. (refs: #37452)
- **PR** #37452: (rallytime) file.line with mode=replace on an empty file should return False, not stacktrace @ *2016-11-06 01:55:11 UTC*
  - be93710fee Merge pull request #37452 from rallytime/fix-31135
  - c792f76d2f Bump log level from debug to warning on empty file
  - 5f181cf00d file.line with mode=replace on an empty file should return False
  - 94a00c66eb Write a unit test demonstrating stack trace in #31135
- **ISSUE** #37001: (phil123456) URGENT : archive.extracted does not work anymore (refs: #37081, #saltstack/salt‘#37081‘_)
- **ISSUE** #29010: (The-Loeki) file.managed download failing checksum testing for Ubuntu initrd w/source_hash (refs: #37469)
  - **PR** saltstack/salt#37081: (terminalmage) Fix archive.extracted remote source_hash verification (refs: #37469)

- **PR** #37469: (terminalmage) Rewrite file.extract_hash to improve its matching ability @ *2016-11-06 01:50:01 UTC*
  - **PR** #37081: (terminalmage) Fix archive.extracted remote source_hash verification (refs: #37469)
  - 129b0387e6 Merge pull request #37469 from terminalmage/issue29010
  - a3f38e5a9f Update file.extract_hash unit tests
  - b26b528f79 Add the source_hash_name param to file.managed states
  - 52fe72d402 Rewrite file.extract_hash
- **ISSUE** #37389: (d101nelson) Some core grains are inaccurate or incomplete for Solaris (refs: #37472)
- **PR** #37472: (sjorge) 2016.3 solaris grains improvements (refs: #37494) @ *2016-11-06 01:46:10 UTC*
  - 9426b9d5c4 Merge pull request #37472 from sjorge/2016.3-solaris-grains
  - 2958f5ce52 detect and properly handle OmniOS
  - 37c3a7f5ab handle Oracle Solaris better
  - 69706d32be parse minorrelease if it has a / in it
  - d1cf4a0e56 improve regex for parsing /etc/release using files from Solaris 8 SPARC and Solaris 10
  - 88eddef765 some more cleanup for smartos
  - d3ff39f09c improve smartos os version grains
- **PR** #37478: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-11-04 20:30:08 UTC*
  - 4ba63aba48 Merge pull request #37478 from rallytime/merge-2016.3
  - 3483a445f2 Merge branch '2015.8' into '2016.3'
    - 35888c2e30 Merge pull request #37408 from terminalmage/issue37286
      - 4e4a05731e Strip slashes from gitfs mountpoints
    - b6c57c6c8d Merge pull request #37418 from terminalmage/issue36849
      - 740bc54239 Do not use compression in tornado httpclient requests
    - 7fba8aaa7e Merge pull request #37441 from rallytime/bp-37428
      - 6fe3ef49de Fix incorrect reference of __utils__ in salt.utils
  - **PR** #37485: (rallytime) Get release notes started for 2016.3.5
  - **PR** #37483: (rallytime) [2016.3] Doc version updated to 2016.3.4
- **ISSUE** #37123: (nevins-b) file.recurse state doesn't support pulling from other environments (refs: #37121)
- **PR** #37121: (nevins-b) allow the file.recurse state to support saltenv @ *2016-11-04 05:59:28 UTC*
  - 580eca709b Merge pull request #37121 from nevins-b/2016.3
  - 99d2c360ed making messaging in tests match new return
  - bc4b0e7cda adding test for saltenv in file.recurse source url
  - 3315b67075 fixing saltenv if not set in url
  - a9683cbbd8 allow the file.recurse state to support saltenv (salt://example/dir?saltenv=dev)
- **PR** #37426: (jfindlay) Wait for macOS to change system settings @ *2016-11-04 04:35:52 UTC*
  - **PR** #37351: (jfindlay) modules.mac_power: give macOS time to change setting (refs: #37426)

  - 766b1437c2 Merge pull request #37426 from jfindlay/mac_sleep
  - 43a8e199bf modules.mac_power: wait for system to make change
  - feabca6e0b modules.mac_system: wait for system to make change
  - 0213eb9a07 utils.mac_utils: add confirm_updated
- **ISSUE** #37238: (cmclaughlin) Restarting master causes minion to hang (refs: #37438, #37602)
- **ISSUE** #37018: (tsaridas) get events from python (refs: #37438, #37602)
- **PR** #37438: (DmitryKuzmenko) Fix for #37238 salt hang on master restart (refs: #37602) @ *2016-11-04 04:10:51 UTC*
  - 9eab5c8f71 Merge pull request #37438 from DSRCorporation/bugs/37238_salt_hang_on_master_restart
  - f253d3ce4a Auto reconnect *salt* to master if the connection was lost.
  - **PR** saltstack/salt#31207: (thusoy) Remove error logging of missing boto libraries (refs: #37440)
- **PR** #37440: (rallytime) Back-port #31207 to 2016.3 @ *2016-11-04 04:09:33 UTC*
  - **PR** #31207: (thusoy) Remove error logging of missing boto libraries (refs: #37440)
  - 9aa7073f70 Merge pull request #37440 from rallytime/bp-31207
  - c71ae61271 Remove error logging of missing boto libraries
- **PR** #37442: (twangboy) Create paths.d directory @ *2016-11-04 04:07:19 UTC*
  - edbfadca21 Merge pull request #37442 from twangboy/fix_osx_postinstall
  - 8091a3065e Create paths.d directory
- **PR** #37445: (twangboy) Check for Server os before checking [DO NOT MERGE FORWARD] @ *2016-11-04 04:04:49 UTC*
  - afb1b3cee5 Merge pull request #37445 from twangboy/fix_import_error_2016.3
  - c0d5ebdd8a Check for Server os before checking
- **PR** #37446: (twangboy) Detect VC++ for Python on Win32 @ *2016-11-04 04:04:02 UTC*
  - 7a9f95ab3b Merge pull request #37446 from twangboy/fix_build_32
  - 2de69f48f8 Detect VC for Python correctly on 32bit Windows
- **ISSUE** saltstack/salt#36961: (nullify005) boto_secgroup assumes a string when checking ip_protocol validity when not tcp|udp|all|-1 (refs: #37447)
- **PR** #37447: (rallytime) Cast ip_protocol rule as a str() in boto_secgroup.present @ *2016-11-04 04:03:45 UTC*
  - 651e0f728f Merge pull request #37447 from rallytime/fix-36961
  - 6b930ac7aa Cast ip_protocol rule as a str() in boto_secgroup.present
- **ISSUE** #36446: (whiteinge) Custom salt-api config problem (refs: #37455)
  - **PR** saltstack/salt#36386: (xiaoanyunfei) fix salt-api's default opts were covered by salt-master #35734 (refs: #37455)
- **PR** #37455: (techhat) Make api opts respect correct root_dir @ *2016-11-04 03:25:40 UTC*
  - **PR** #35734: (xiaoanyunfei) fix salt-api's default opts were covered by salt-master (refs: #'saltstack/salt#36386'_)
  - a51d944c7c Merge pull request #37455 from techhat/issue36446
  - 7eff90d61d Make api opts respect correct root_dir

- **PR** #37459: (twangboy) Fix error message when ConvertTo-Json not supported [DO NOT MERGE FORWARD] @ *2016-11-04 03:22:31 UTC*
  - 3591bf0f58 Merge pull request #37459 from twangboy/fix_dsc_json_msg_2016.3
  - 949b70913d Use cmd.run_all instead of cmd.shell
- **PR** #37430: (meaksh) Including resolution parameters in the Zypper debug-solver call during a dry-run dist-upgrade (2016.3) @ *2016-11-03 14:35:46 UTC*
  - **PR** #37353: (meaksh) Including resolution parameters in the Zypper debug-solver call during a dry-run dist-upgrade (refs: #37430)
  - 80a99c4cc5 Merge pull request #37430 from meaksh/zypper-dist-upgrade-debug-solver-fix-2016.3
  - ffc596f215 Including resolver params for Zypper debug-solver
- **ISSUE** #37388: (tyhunt99) [2016.3.4] Refreshing of an s3 file server results in an exception. (refs: #37428)
  - **PR** #37428: (cachedout) Fix incorrect reference of __utils__ in salt.utils (refs: #37441)
- **PR** #37419: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2016-11-02 21:40:04 UTC*
  - 7864f9b79d Merge pull request #37419 from rallytime/merge-2016.3
  - bce47c9175 Merge branch '2015.8' into '2016.3'
    - 7b1d3b5562 Merge pull request #37392 from rallytime/bp-33190
    - 4063bae5de catch None cases for comments in jboss7 state module
  - **PR** #37416: (terminalmage) Fix regression in output for Ctrl-c'ed CLI jobs
  - **PR** #37414: (pass-by-value) Add unit tests for cloning from snapshot
  - **PR** #37350: (pass-by-value) Add handling for full and linked clone (refs: #37414)
  - **PR** saltstack/salt#37401: (cachedout) Bootstrap delay option for salt-cloud (refs: #37404)
- **PR** #37404: (cachedout) Revert "Bootstrap delay option for salt-cloud" @ *2016-11-02 09:48:53 UTC*
  - ecd794a233 Merge pull request #37404 from saltstack/revert-37401-bootstrap_delay
  - e864de8f03 Revert "Bootstrap delay option for salt-cloud"
- **PR** #37401: (cachedout) Bootstrap delay option for salt-cloud @ *2016-11-02 09:02:13 UTC*
  - 2eb44fbd11 Merge pull request #37401 from cachedout/bootstrap_delay
  - 6e42b0e157 Bootstrap delay option for salt-cloud
- **PR** #37350: (pass-by-value) Add handling for full and linked clone (refs: #37414) @ *2016-11-02 08:02:29 UTC*
  - 9446e48da0 Merge pull request #37350 from pass-by-value/full_and_linked_clone_v1
  - d8b1c9c777 Add handling for full and linked clone and commit disk mode additions
- **ISSUE** #34841: (Ch3LL) Wrong return when using *user.chgroups* on windows (refs: #37386)
- **PR** #37386: (rallytime) Fix win_useradd.chgroups return when cmd.run_all retcode != 0 @ *2016-11-02 06:34:12 UTC*
  - c7f4d7f76a Merge pull request #37386 from rallytime/fix-34841
  - c70492a1fe Fix win_useradd.chgroups return when cmd.run_all retcode != 0
- **ISSUE** #34263: (vernondcole) Use of dnsmasq.set_config injects unintentional text into the configuration file. (refs: #37390)

- **PR** #37390: (rallytime) Don't insert __pub* keys into dnsmasq config file with set_config function @ *2016-11-02 06:31:53 UTC*
    - 34b6c6459a Merge pull request #37390 from rallytime/fix-34263
    - e082ff538b Fix failing test now that we're raising a CommandExecutionError
    - c6a3476abb Filter out the __pub keys passed via **kwargs for dnsmasq.set_config
    - fd380c79b9 Add test case to reproduce dnsmasq.set_config failure in #34263
- **ISSUE** #35163: (SolarisYan) salt file.mkdir (refs: #35287, #35189)
- **PR** #37391: (rallytime) Back-port #35287 to 2016.3 @ *2016-11-02 06:18:26 UTC*
    - **PR** #35287: (dere) 2016.3 (refs: #37391)
    - **PR** #35189: (dere) return value for file.mkdir instead of None (refs: #35287)
    - 798b2acbe3 Merge pull request #37391 from rallytime/bp-35287
    - 0e1ebea5a4 Simplify return value to "True".
    - 13022c5cc4 return value for mkdir instead of None
- **ISSUE** #37264: (junster1) Parsing __grains__ with json.dumps in a module is returning an empty dict in 2016.3.3 (refs: #37279)
- **PR** #37279: (gtmanfred) initialize super class of NamespacedDictWrapper @ *2016-11-01 15:12:49 UTC*
    - 1a4833b3a1 Merge pull request #37279 from gtmanfred/2016.3
    - 597f346d57 initialize super class of NamespacedDictWrapper
- **PR** #37351: (jfindlay) modules.mac_power: give macOS time to change setting (refs: #37426) @ *2016-10-31 19:15:40 UTC*
    - 351175931c Merge pull request #37351 from jfindlay/mac_set
    - 0c58056d84 modules.mac_power: give macOS time to change setting
- **PR** #37340: (cachedout) SIGILL -> SIGKILL in process test @ *2016-10-31 08:50:10 UTC*
    - 25c987e33a Merge pull request #37340 from cachedout/ill_kill_3
    - a6b7417fe9 SIGILL -> SIGKILL in process test
- **ISSUE** #35480: (jelenak) 200 processes of salt-master (2016.3.2) (refs: #37306)
- **PR** #37306: (DmitryKuzmenko) Don't use os.wait() on subprocesses managed by *multiprocessing*. @ *2016-10-31 06:55:30 UTC*
    - 7f1654894d Merge pull request #37306 from DSRCorporation/bugs/35480_master_shutdown_no_process_error
    - b6937ebaa8 Don't use os.wait() on subprocesses managed by *multiprocessing*.
- **ISSUE** #34998: (exowaucka) placementgroup parameter for salt-cloud is undocumented (refs: #37314)
- **PR** #37314: (rallytime) Document the existence of placementgroup option in ec2 driver @ *2016-10-31 06:42:33 UTC*
    - bf8ba97d54 Merge pull request #37314 from rallytime/fix-34998
    - 39459ed30b Document the existence of placementgroup option in ec2 driver
- **ISSUE** #36148: (alex-zel) Eauth error with openLDAP groups (refs: #37219)
- **PR** #37219: (alex-zel) Fix freeipa ldap groups @ *2016-10-28 04:33:37 UTC*
    - e0baf4b193 Merge pull request #37219 from alex-zel/fix-freeipa-ldap-groups

  - b5b2e7e097 Remove trailing whitespaces
  - 32f906b020 Add support for FreeIPA

28.2.47 Salt 2016.3.6 Release Notes

Version 2016.3.6 is a bugfix release for *2016.3.0*.

Statistics

- Total Merges: **119**
- Total Issue References: **52**
- Total PR References: **163**
- Contributors: **43** (Adaephon-GH, Ch3LL, DmitryKuzmenko, Foxlik, GideonRed-zz, The-Loeki, UtahDave, alexbleotu, anlutro, bobrik, cachedout, cro, dincamihai, drawsmcgraw, fboismenu, galet, garethgreenaway, grep4linux, gtmanfred, jacobhammons, jfindlay, joe-niland, lvg01, mbom2004, mcalmer, mchugh19, meaksh, mirceaulinic, morganwillcock, narendraingale2, nasenbaer13, ni3mm4nd, rallytime, s0undt3ch, sergeizv, smarsching, techhat, terminalmage, thatch45, twangboy, velom, vutny, yue9944882)

Security Fix

CVE-2017-7893 Compromised salt-minions can impersonate the salt-master. (Discovery credit: Frank Spierings)

Changelog for v2016.3.5..v2016.3.6

Generated at: 2018-05-27 13:45:07 UTC

- **PR** #40232: (rallytime) Update release notes for 2016.3.6 @ *2017-03-22 21:09:35 UTC*
  - 24c4ae9c21 Merge pull request #40232 from rallytime/update-release-notes
  - 2ead188b4f Update release notes for 2016.3.6
- **ISSUE** #39854: (Foxlik) quoted space in authorized_keys confuses ssh.py (refs: #39855)
- **PR** #39855: (Foxlik) Use regular expression instead of split when replacing authorized_keys @ *2017-03-22 18:28:32 UTC*
  - c59ae9a82c Merge pull request #39855 from Foxlik/use_regex_to_compare_authorized_keys
  - d46845a5b6 Add newline at end of file
  - d4a3c8a66a Use regular expression instead of split when replacing authorized_keys
- **PR** #40221: (rallytime) Back-port #39179 to 2016.3 @ *2017-03-22 17:40:34 UTC*
  - **PR** #39179: (mcalmer) fix error parsing (refs: #40221)
  - fd10430018 Merge pull request #40221 from rallytime/bp-39179
  - 07dc2de084 fix error parsing
- **ISSUE** #40203: (frogunder) 2016.3.6. Minion don't connect to older master. (refs: #40206)
- **PR** #40206: (cro) Leave sign_pub_messages off by default. @ *2017-03-22 16:43:03 UTC*
  - a27a2cc3bb Merge pull request #40206 from cro/sign_pub_take2

  - 01048de83f leave sign_pub_messages off on minion by default.
  - a82b005507 Leave sign_pub_messages off by default.
- **PR** #40193: (rallytime) Back-port #40117 to 2016.3 @ *2017-03-22 16:42:21 UTC*
  - **PR** #40117: (narendraingale2) Fix force remove (refs: #40193)
  - d1abb4cbaa Merge pull request #40193 from rallytime/bp-40117
  - cf1857904b More optimization.
  - 5a08266814 Removed debug statemnt
  - f557f7c6bb Added fix for issue 39393
  - bb62278b73 Reverting changes.
  - a9107cde44 Added if condition for broken link.
- **PR** #40196: (twangboy) Update dependencies for PyOpenSSL @ *2017-03-22 16:40:46 UTC*
  - 0f1ff4d4a8 Merge pull request #40196 from twangboy/win_fix_deps
  - 6761527793 Update dependencies for PyOpenSSL
- **PR** #40184: (terminalmage) Link to minion start reactor example from FAQ. @ *2017-03-21 17:33:09 UTC*
  - b0501515cb Merge pull request #40184 from terminalmage/link-reactor-example
  - a42be82993 Link to minion start reactor example from FAQ.
- **PR** #40182: (terminalmage) Add support for "stopped" state to dockerng's mod_watch @ *2017-03-21 15:40:29 UTC*
  - d4e6c58526 Merge pull request #40182 from terminalmage/dockerng-mod_watch-stopped
  - 4629a26fb7 Add support for "stopped" state to dockerng's mod_watch
- **PR** #40171: (Ch3LL) additional PRs/issues for 2016.3.6 release notes @ *2017-03-20 22:14:17 UTC*
  - a0b4082484 Merge pull request #40171 from Ch3LL/2016.3.6_release
  - 9c6d8d892f additional PRs/issues for 2016.3.6 release notes
- **PR** #40120: (sergeizv) gce: Exclude GCENodeDriver objects from _expand_node result @ *2017-03-20 21:44:42 UTC*
  - 33ba7821f7 Merge pull request #40120 from sergeizv/gce-expand-node-fix
  - 9d0fbe7e01 gce: Exclude GCENodeDriver objects from _expand_node result
- **PR** #40122: (meaksh) Adding "pkg.install downloadonly=True" support to yum/dnf execution module @ *2017-03-20 21:44:15 UTC*
  - 48843977c3 Merge pull request #40122 from meaksh/2016.3-yum-downloadonly-support
  - 067f3f77c2 Adding downloadonly support to yum/dnf module
- **PR** #40159: (cro) Turn on sign_pub_messages by default. @ *2017-03-20 21:00:49 UTC*
  - 60e1d4e2f3 Merge pull request #40159 from cro/sign_pub
  - e663b761fb Fix small syntax error
  - 0a0f46fb14 Turn on sign_pub_messages by default. Make sure messages with no 'sig' are dropped with error when sign_pub_messages is True.
- **PR** #40123: (twangboy) Adds support for inet_pton in Windows to network util @ *2017-03-20 16:25:47 UTC*

  - 28e4fc17b6 Merge pull request #40123 from twangboy/win_fix_network
  - 06dfd55ef9 Adds support for inet_pton in Windows to network util
- **ISSUE** #39995: (frogunder) Head of Develop - Multimaster error (refs: #40141)
- **ISSUE** #39118: (bobrik) Minion ipv6 option is not documented (refs: #39289, #39131)
- **PR** #40141: (bobrik) Use the first address if cannot connect to any @ *2017-03-20 15:06:57 UTC*
  - **PR** #39289: (bobrik) Autodetect IPv6 connectivity from minion to master (refs: #39766, #40141)
  - 35ddb79f59 Merge pull request #40141 from bobrik/fallback-resolve
  - af1545deed Use the first address if cannot connect to any
- **PR** #40059: (terminalmage) Fix traceback when virtualenv.managed is invoked with nonexistant user @ *2017-03-16 20:46:43 UTC*
  - 116201f345 Merge pull request #40059 from terminalmage/fix-virtualenv-traceback
  - e3cfd29d6b Fix traceback when virtualenv.managed is invoked with nonexistant user
- **PR** #40090: (rallytime) Back-port #40056 to 2016.3 @ *2017-03-16 19:42:58 UTC*
  - **PR** #40056: (thatch45) update mention bot blacklist (refs: #40090)
  - a01b52b9a3 Merge pull request #40090 from rallytime/bp-40056
  - ae012db87a update mention bot blacklist
- **PR** #40057: (cachedout) More mentionbot blacklists @ *2017-03-16 18:10:11 UTC*
  - d1570bba4c Merge pull request #40057 from cachedout/ollie_blacklist
  - 0ac2e83d37 Merge branch '2016.3' into ollie_blacklist
- **PR** #40070: (Ch3LL) update 2016.3.6 release notes with additional PR's @ *2017-03-16 15:43:22 UTC*
  - d36bdb1a6e Merge pull request #40070 from Ch3LL/2016.3.6_release
  - a1f8b49bd1 update 2016.3.6 release notes with additional PR's
- **PR** #40018: (meaksh) Allows overriding 'timeout' and 'gather_job_timeout' to 'manage.up' runner call @ *2017-03-15 19:43:01 UTC*
  - 8dcffc7751 Merge pull request #40018 from meaksh/2016.3-handling-timeouts-for-manage.up-runner
  - 9f5c3b7dcd Allows to set custom timeouts for 'manage.up' and 'manage.status'
  - 2102d9c75c Allows to set 'timeout' and 'gather_job_timeout' via kwargs
- **PR** #40038: (velom) correctly parse "pkg_name===version" from pip freeze @ *2017-03-15 19:30:03 UTC*
  - 22fc5299a2 Merge pull request #40038 from velom/fix-pip-freeze-parsing
  - 3fae91d879 correctly parse "pkg_name===version" from pip freeze
- **ISSUE** #40036: (oogali) UnboundLocalError: local variable 'ifcfg' referenced before assignment (refs: #40053)
- **PR** #40053: (gtmanfred) Update rh_ip.py @ *2017-03-15 18:57:32 UTC*
  - 3584f935fa Merge pull request #40053 from saltstack/rh_ip_patch
  - 219947acdb Update rh_ip.py
- **ISSUE** #40011: (tsaridas) salt-minion does not shutdown properly 2016.11.3 rh6 (refs: #40041)
- **PR** #40041: (terminalmage) Fix transposed lines in salt.utils.process @ *2017-03-15 17:58:24 UTC*
  - 837432d3d2 Merge pull request #40041 from terminalmage/issue40011

  - 5b5d1b375c Fix transposed lines in salt.utils.process
- **PR** #40021: (Ch3LL) 2016.3.6 release notes with change log @ *2017-03-14 21:06:18 UTC*
  - ee7f3b1200 Merge pull request #40021 from Ch3LL/2016.3.6_release
  - f3e7e4fb2a Add 2016.3.6 Release Notes
- **PR** #40016: (terminalmage) Attempt to fix failing grains tests in 2016.3 @ *2017-03-14 18:34:32 UTC*
  - 26895b7be2 Merge pull request #40016 from terminalmage/fix-grains-test
  - 0ec81a4cde Fixup a syntax error
  - 5d84b40bfd Attempt to fix failing grains tests in 2016.3
- **PR** #39980: (vutny) [2016.3] Allow to use *bg* kwarg for *cmd.run* state function @ *2017-03-14 17:16:14 UTC*
  - 0c61d064ad Merge pull request #39980 from vutny/cmd-run-state-bg
  - a81dc9dfc1 [2016.3] Allow to use *bg* kwarg for *cmd.run* state function
- **ISSUE** #39942: (Foxlik) Web Documentation not in sync with release 2016.11.3 (refs: #39994)
- **PR** #39994: (rallytime) Add a versionadded tag for dockerng ulimits addition @ *2017-03-13 20:58:02 UTC*
  - b042484455 Merge pull request #39994 from rallytime/ulimits-dockerng-version
  - 37bd800fac Add a versionadded tag for dockerng ulimits addition
- **PR** #39988: (terminalmage) Add comment explaining change from #39973 @ *2017-03-13 18:37:29 UTC*
  - **PR** #39973: (terminalmage) Don't use docker.Client instance from context if missing attributes (refs: #39988)
  - e125c94ba5 Merge pull request #39988 from terminalmage/dockerng-timeout
  - bd2519ed1b Add comment explaining change from #39973
- **PR** #39973: (terminalmage) Don't use docker.Client instance from context if missing attributes (refs: #39988) @ *2017-03-11 14:57:50 UTC*
  - cd0336e868 Merge pull request #39973 from terminalmage/dockerng-timeout
  - 869416e7db Don't use docker.Client instance from context if missing attributes
- **PR** #39962: (cachedout) Disable mention bot delay on 2016.3 @ *2017-03-10 20:24:08 UTC*
  - 282c607d26 Merge pull request #39962 from cachedout/disable_mentionbot_delay_3
  - 7a638f204b Disable mention bot delay on 2016.3
    - 5592c680b5 More mentionbot blacklists
- **PR** #39937: (cachedout) Fix –non-gpg-checks in zypper module @ *2017-03-10 18:02:51 UTC*
  - 1e0c88ae08 Merge pull request #39937 from cachedout/gpg_zypper
  - 13ed0d1209 Fix –non-gpg-checks in zypper module
- **PR** #39929: (terminalmage) Scrap event-based approach for refreshing grains (2016.3 branch) @ *2017-03-09 22:03:16 UTC*
  - 4526fc6e08 Merge pull request #39929 from terminalmage/pr-39770-2016.3
  - cf0100dabe Scrap event-based approach for refreshing grains
- **ISSUE** #22080: (The-Loeki) CIDR matching for IPv6 / improve IPv6 support in utils.network (refs: #39919)
- **PR** #39919: (The-Loeki) CIDR matching supports IPv6, update docs @ *2017-03-09 16:03:00 UTC*

  - 111110caf8 Merge pull request #39919 from The-Loeki/patch-1
  - 170cbadc54 CIDR matching supports IPv6, update docs
- **PR** #39899: (techhat) Update cleanup function for azure @ *2017-03-08 23:28:33 UTC*
  - caf10e9988 Merge pull request #39899 from techhat/cleanupdisks
  - baf4579e63 Update cleanup function for azure
- **PR** #39871: (terminalmage) Squelch warning for pygit2 import @ *2017-03-07 20:40:18 UTC*
  - fcf95f3654 Merge pull request #39871 from terminalmage/squelch-import-warning
  - 2b2ec69d04 Squelch warning for pygit2 import
- **PR** #39794: (cachedout) Clean up errors which might be thrown when the monitor socket shuts down @ *2017-03-04 16:12:37 UTC*
  - f223fa8906 Merge pull request #39794 from cachedout/clean_monitor_socket_shutdown
  - 2e683e788b Clean up errors which might be thrown when the monitor socket shuts down
- **PR** #39819: (terminalmage) Improve the Top File matching docs @ *2017-03-04 16:06:40 UTC*
  - 4002dc1947 Merge pull request #39819 from terminalmage/top-file-matching-docs
  - 7178e77eee Improve the Top File matching docs
- **PR** #39820: (ni3mm4nd) Add missing apostrophe in Beacons topic documentation @ *2017-03-04 16:05:29 UTC*
  - c08aaeb7fd Merge pull request #39820 from ni3mm4nd/beacons_topic_doc_typo
  - 804b12048c Add missing apostrophe
- **PR** #39826: (cachedout) Add group func to yubikey auth @ *2017-03-04 16:02:14 UTC*
  - cbd2a4e3cc Merge pull request #39826 from cachedout/yubikey_fix
  - 6125eff02d Add group func to yubikey auth
- **ISSUE** #39622: (drawsmcgraw) boto_vpc.create_subnet does not properly assign tags (refs: #39624)
- **PR** #39624: (drawsmcgraw) Address issue 39622 @ *2017-03-03 15:59:04 UTC*
  - f575ef459f Merge pull request #39624 from drawsmcgraw/39622
  - 13da50be33 Fix indention lint errors
  - 545026352f Address issue 39622
- **ISSUE** #39119: (frogunder) Head of 2016.3 - Salt-Master uses 90 seconds to restart (refs: #39796)
- **PR** #39796: (cachedout) Stop the process manager when it no longer has processes to manage @ *2017-03-02 23:03:13 UTC*
  - 1f3619c1e5 Merge pull request #39796 from cachedout/master_shutdown
  - e31d46c1b8 Stop the process manager when it no longer has processes to manage
- **ISSUE** #39333: (jagguli) Not Available error - Scheduling custom runner functions (refs: #39791)
- **ISSUE** #38514: (githubcdr) Unable to schedule runners (refs: #39791)
- **PR** #39791: (gtmanfred) load runners if role is master @ *2017-03-02 19:43:41 UTC*
  - 53341cf152 Merge pull request #39791 from gtmanfred/2016.3
  - 3ab4f843bf load runners if role is master
- **ISSUE** #39782: (sergeizv) salt-cloud show_instance action fails on EC2 instances (refs: #39784)

- **ISSUE** #33162: (jfindlay) Key error with salt.utils.cloud.cache_node and EC2 (refs: #39784, #33164)
- **PR** #39784: (sergeizv) Fix 39782 @ *2017-03-02 16:08:51 UTC*
  - **PR** #33164: (jfindlay) cloud.clouds.ec2: cache each named node (refs: #39784)
  - c234c25092 Merge pull request #39784 from sergeizv/fix-39782
  - b71c3fe13c Revert "cloud.clouds.ec2: cache each named node (#33164)"
- **ISSUE** #39336: (GevatterGaul) salt-minion fails with IPv6 (refs: #39766)
- **ISSUE** #39118: (bobrik) Minion ipv6 option is not documented (refs: #39289, #39131)
- **PR** #39766: (rallytime) Restore ipv6 connectivity and "master: <ip>:<port>" support @ *2017-03-02 02:55:55 UTC*
  - **PR** #39289: (bobrik) Autodetect IPv6 connectivity from minion to master (refs: #39766, #40141)
  - **PR** #25021: (GideonRed-zz) Introduce ip:port minion config (refs: #39766)
  - 4ee59be22c Merge pull request #39766 from rallytime/fix-ipv6-connection
  - 65b239664e Restore ipv6 connectivity and "master: <ip>:<port>" support
- **ISSUE** #33187: (usbportnoy) Deploy to jboss TypeError at boss7.py:469 (refs: #39761, #39170)
- **PR** #39761: (cachedout) Properly display error in jboss7 state @ *2017-03-01 18:43:23 UTC*
  - a24da31131 Merge pull request #39761 from cachedout/issue_33187
  - c2df29edb2 Properly display error in jboss7 state
- **PR** #39728: (rallytime) [2016.3] Bump latest release version to 2016.11.3 @ *2017-02-28 18:07:44 UTC*
  - 0888bc32ef Merge pull request #39728 from rallytime/update-release-ver-2016.3
  - c9bc8af8f2 [2016.3] Bump latest release version to 2016.11.3
- **PR** #39619: (terminalmage) Add a function to simply refresh the grains @ *2017-02-28 00:20:27 UTC*
  - b52dbeec68 Merge pull request #39619 from terminalmage/zd1207
  - c7dfb494a6 Fix mocking for grains refresh
  - 7e0ced3b45 Properly hand proxy minions
  - 692c456da3 Add a function to simply refresh the grains
- **ISSUE** #39482: (bobrik) file.managed and file mode don't mention default mode (refs: #39487)
- **PR** #39487: (bobrik) Document default permission modes for file module @ *2017-02-24 23:49:00 UTC*
  - 3f8b5e6733 Merge pull request #39487 from bobrik/mode-docs
  - 41ef69b3ca Document default permission modes for file module
- **ISSUE** #39169: (blueyed) Using batch-mode with *salt.state* in orchestration runner considers all minions to have failed (refs: #39641)
- **PR** #39641: (smarsching) Return runner return code in a way compatible with check_state_result @ *2017-02-24 23:07:11 UTC*
  - f7389bf1f5 Merge pull request #39641 from smarsching/issue-39169-2016.3
  - 88c2d9a540 Fix return data structure for runner (issue #39169).
- **PR** #39633: (terminalmage) Fix misspelled argument in salt.modules.systemd.disable() @ *2017-02-24 18:21:36 UTC*

  - fc970b6a16 Merge pull request #39633 from terminalmage/fix-systemd-typo
  - ca54541abe Add missing unit test for disable func
  - 17109e1522 Fix misspelled argument in salt.modules.systemd.disable()
- **PR** #39613: (terminalmage) Fix inaccurate documentation @ *2017-02-24 06:07:35 UTC*
  - 53e78d67f6 Merge pull request #39613 from terminalmage/fix-docs
  - 9342eda377 Fix inaccurate documentation
- **PR** #39600: (vutny) state.file: drop non-relevant examples for *source_hash* parameter @ *2017-02-23 16:55:27 UTC*
  - 4e2b852f83 Merge pull request #39600 from vutny/state-file-docs
  - 9b0427c27a state.file: drop non-relevant examples for *source_hash* parameter
- **PR** #39584: (cachedout) A note in the docs about mentionbot @ *2017-02-23 15:12:13 UTC*
  - ed83420417 Merge pull request #39584 from cachedout/mentionbot_docs
  - 652044b18f A note in the docs about mentionbot
- **PR** #39583: (cachedout) Add empty blacklist to mention bot @ *2017-02-23 02:22:57 UTC*
  - d3e50b4f2f Merge pull request #39583 from cachedout/mentionbot_blacklist
  - 62491c900d Add empty blacklist to mention bot
- **PR** #39579: (rallytime) [2016.3] Pylint: Remove unused import @ *2017-02-22 23:46:33 UTC*
  - 8352e6b44b Merge pull request #39579 from rallytime/fix-lint
  - 65889e1f30 [2016.3] Pylint: Remove unused import
- **PR** #39578: (cachedout) Add mention-bot configuration @ *2017-02-22 23:39:24 UTC*
  - 43dba3254c Merge pull request #39578 from cachedout/2016.3
  - 344499eef7 Add mention-bot configuration
- **PR** #39542: (twangboy) Gate ssh_known_hosts state against Windows @ *2017-02-22 20:16:41 UTC*
  - 8f7a0f9d96 Merge pull request #39542 from twangboy/gate_ssh_known_hosts
  - c90a52ef27 Remove expensive check
  - 6d645cae0e Add __virtual__ function
- **ISSUE** #39118: (bobrik) Minion ipv6 option is not documented (refs: #39289, #39131)
- **PR** #39289: (bobrik) Autodetect IPv6 connectivity from minion to master (refs: #39766, #40141) @ *2017-02-22 19:05:32 UTC*
  - c10965833a Merge pull request #39289 from bobrik/autodetect-ipv6
  - 2761a1b244 Move new kwargs to the end of argument list
  - 0df6b922e7 Narrow down connection exception to socket.error
  - e8a2cc0488 Do no try to connect to salt master in syndic config test
  - af9578631e Properly log address that failed to resolve or pass connection check
  - 9a34fbeba9 Actually connect to master instead of checking route availability
  - c494839c65 Avoid bare exceptions in dns_check
  - 29f376676d Rewrite dns_check to try to connect to address

  - 55965ce505 Autodetect IPv6 connectivity from minion to master
- **PR** #39569: (s0undt3ch) Don't use our own six dictionary fixes in this branch @ *2017-02-22 18:59:49 UTC*
  - 3fb928b63a Merge pull request #39569 from s0undt3ch/2016.3
  - 49da135abd Don't use our own six dictionary fixes in this branch
- **PR** #39508: (dincamihai) Openscap @ *2017-02-22 18:36:36 UTC*
  - 91e3319df8 Merge pull request #39508 from dincamihai/openscap
  - 9fedb84607 Always return oscap's stderr
  - 0ecde2cd02 Include oscap returncode in response
- **ISSUE** #30802: (kjelle) Missing ulimits on docker.running / dockerng.running (refs: #39562)
- **PR** #39562: (terminalmage) Add ulimits to dockerng state/exec module @ *2017-02-22 16:31:49 UTC*
  - fbe2194a93 Merge pull request #39562 from terminalmage/issue30802
  - c50374041d Add ulimits to dockerng state/exec module
  - da42040c1a Try the docker-py 2.0 client name first
  - **PR** #39544: (terminalmage) dockerng.get_client_args: Fix path for endpoint config for some versions of docker-py
- **ISSUE** #39447: (Foxlik) dockerng keeps restarting privileged container (refs: #39483)
- **PR** #39498: (terminalmage) Resubmit PR #39483 against 2016.3 branch @ *2017-02-20 19:35:33 UTC*
  - **PR** #39483: (Foxlik) dockerng: compare sets instead of lists of security_opt (refs: #39498)
  - dff35b58f8 Merge pull request #39498 from terminalmage/pr-39483
  - 20b097a745 dockerng: compare sets instead of lists of security_opt
- **PR** #39497: (terminalmage) Two dockerng compatibility fixes @ *2017-02-19 17:43:36 UTC*
  - 6418e725ed Merge pull request #39497 from terminalmage/docker-compat-fixes
  - cbd0270bac docker: make docker-exec the default execution driver
  - a6a17d58aa Handle docker-py 2.0's new host_config path
- **PR** #39423: (dincamihai) Openscap module @ *2017-02-17 18:31:04 UTC*
  - 9c4292fb4e Merge pull request #39423 from dincamihai/openscap
  - 9d13422ac1 OpenSCAP module
- **ISSUE** #39444: (clem-compilatio) salt-cloud - IPv6 and IPv4 private_ips - preferred_ip sends False to is_public_ip (refs: #39464)
- **PR** #39464: (gtmanfred) skip false values from preferred_ip @ *2017-02-16 22:48:32 UTC*
  - 7dd2502360 Merge pull request #39464 from gtmanfred/2016.3
  - f829d6f9fc skip false values from preferred_ip
- **PR** #39460: (cachedout) Fix mocks in win_disim tests @ *2017-02-16 19:27:48 UTC*
  - db359ff2c3 Merge pull request #39460 from cachedout/win_dism_test_fix
  - e652a45592 Fix mocks in win_disim tests
- **PR** #39426: (morganwillcock) win_dism: Return failure when package path does not exist @ *2017-02-16 00:09:22 UTC*

  - 9dbfba9b57 Merge pull request #39426 from morganwillcock/dism
  - a7d5118262 Return failure when package path does not exist
- **PR** #39431: (UtahDave) Fix grains.setval performance @ *2017-02-15 23:56:30 UTC*
  - 56162706e3 Merge pull request #39431 from UtahDave/fix_grains.setval_performance
  - 391bbecd90 add docs
  - 709c197f84 allow sync_grains to be disabled on grains.setval
- **ISSUE** #39304: (Auha) boto_s3_bucket documentation dependency clarification (refs: #39405)
- **PR** #39405: (rallytime) Update :depends: docs for boto states and modules @ *2017-02-15 17:32:08 UTC*
  - 239e16e612 Merge pull request #39405 from rallytime/fix-39304
  - bd1fe03ce7 Update :depends: docs for boto states and modules
- **ISSUE** #38762: (oz123) Configuration information for custom returners (refs: #39411)
- **PR** #39411: (rallytime) Update external_cache docs with other configuration options @ *2017-02-15 17:30:40 UTC*
  - 415102f346 Merge pull request #39411 from rallytime/fix-38762
  - e13febe58d Update external_cache docs with other configuration options
  - **PR** #39421: (terminalmage) Update docs on upstream EPEL7 pygit2/libgit2 issues
  - **PR** #39409: (terminalmage) salt.fileserver.roots: Fix regression in symlink_list
  - **PR** #39337: (terminalmage) Don't re-walk the roots fileserver in symlink_list() (refs: #39409)
- **PR** #39362: (dincamihai) Add cp.push test @ *2017-02-14 18:42:11 UTC*
  - 8b8ab8ef8e Merge pull request #39362 from dincamihai/cp-push-test-2016.3
  - 91383c5a19 Add cp.push test
- **PR** #39380: (joe-niland) Quote numeric user names so pwd.getpwnam handles them properly @ *2017-02-14 18:33:33 UTC*
  - 4b726f955b Merge pull request #39380 from joe-niland/quote-numeric-usernames
  - c2edfdd464 Quote numeric user names so pwd.getpwnam handles them properly
- **PR** #39400: (meaksh) Prevents 'OSError' exception in case certain job cache path doesn't exist @ *2017-02-14 18:27:04 UTC*
  - 1116d32df9 Merge pull request #39400 from meaksh/2016.3-fix-local-cache-issue
  - e7e559ef5c Prevents 'OSError' exception in case path doesn't exist
- **PR** #39300: (terminalmage) Replace more usage of str.format in the loader @ *2017-02-13 19:01:19 UTC*
  - **PR** #39227: (terminalmage) Loader optimzation (refs: #39300)
  - 6c854da1d4 Merge pull request #39300 from terminalmage/loader-optimization
  - d3e5d1525e Replace more usage of str.format in the loader
- **PR** #39337: (terminalmage) Don't re-walk the roots fileserver in symlink_list() (refs: #39409) @ *2017-02-13 18:41:17 UTC*
  - 5286b5ff1b Merge pull request #39337 from terminalmage/issue34428
  - a7d2135dc2 Don't re-walk the roots fileserver in symlink_list()

- **PR** #39339: (cro) Add link to external pillar documentation for clarification. @ *2017-02-13 18:40:13 UTC*
  - ce781deeb5 Merge pull request #39339 from cro/pillar_filetree_doc
  - 410810cea2 Clarification on external pillar usage.
  - **PR** #39316: (terminalmage) Document the upstream RedHat bug with their pygit2 package
- **PR** #39313: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2017-02-10 16:23:23 UTC*
  - 9de559ff4e Merge pull request #39313 from rallytime/merge-2016.3
  - 0b8dddf12b Merge branch '2015.8' into '2016.3'
  - fc551bcf5d Merge pull request #39293 from sergeizv/grammar-fix
    - 70f2b586d3 Rewrap paragraph
    - e6ab5178ea Grammar fix
  - 8a1b45632a Merge pull request #39295 from sergeizv/typo-fix
    - 5d9f36d58d Fix typo
  - cfaafece34 Merge pull request #39296 from sergeizv/whitespace-fix
    - 1d4c1dc140 Whitespace fix in docs Makefile
  - 0b4dcf4a47 Merge pull request #39294 from sergeizv/fix-link
    - 04bde6eed2 Fix link in proxyminion guide
- **ISSUE** #38595: (yue9944882) Redis ext job cache occurred error (refs: #38610)
  - **PR** #39299: (rallytime) Back-port #38610 to 2016.3
  - **PR** #38610: (yue9944882) Fix #38595 - Unexpected error log from redis retuner in master's log (refs: #39299)
- **PR** #39297: (cro) Add doc to recommend pgjsonb for master job caches @ *2017-02-09 22:49:59 UTC*
  - f16027d30e Merge pull request #39297 from cro/pg_returner_docs
  - 28bac649ae Typo
  - 19fedcdd23 Add doc to recommend pgjsonb for master job caches
- **PR** #39286: (terminalmage) Allow minion/CLI saltenv/pillarenv to override master when compiling pillar @ *2017-02-09 21:22:46 UTC*
  - 77e50ed8b7 Merge pull request #39286 from terminalmage/fix-pillarenv-precedence
  - 3cb9833e57 Allow minion/CLI saltenv/pillarenv to override master when compiling pillar
- **ISSUE** #39220: (lvg01) state file.line skips leading spaces in content with mode:ensure and indent:False (refs: #39221)
- **PR** #39221: (lvg01) Fix bug 39220 @ *2017-02-09 18:12:29 UTC*
  - 52440416ca Merge pull request #39221 from lvg01/fix-bug-39220
  - e8a41d6341 Removes to early content stripping (stripping is allready done when needed with ident:true), fixes #39220
  - a4b169e0bd Fixed wrong logic, fixes #39220
- **ISSUE** #36913: (terminalmage) Support custom refspecs in GitFS (refs: #39210)
  - **PR** #39280: (terminalmage) Add warning for Dulwich removal

  - **PR** #39210: (terminalmage) salt.utils.gitfs: remove dulwich support, make refspecs configurable (refs: #39280)
  - **PR** #39260: (terminalmage) Update jsonschema tests to reflect change in jsonschema 2.6.0
- **ISSUE** saltstack/salt#33536: (murzick) pkgrepo.managed does not disable a yum repo with "disabled: True" (refs: #35055)
- **ISSUE** #33536: (murzick) pkgrepo.managed does not disable a yum repo with "disabled: True" (refs: #35055)
  - **PR** #39251: (terminalmage) Better handling of enabled/disabled arguments in pkgrepo.managed
  - **PR** #35055: (galet) #33536 pkgrepo.managed does not disable a yum repo with "disabled: True" (refs: #39251)
- **PR** #39227: (terminalmage) Loader optimzation (refs: #39300) @ *2017-02-08 19:38:21 UTC*
  - 8e88f71dd9 Merge pull request #39227 from terminalmage/loader-optimization
  - c750662946 Loader optimzation
- **ISSUE** #38856: (fhaynes) salt-cloud throws an exception when ec2 does not return encoding (refs: #39228)
- **PR** #39228: (gtmanfred) default to utf8 encoding if not specified @ *2017-02-08 19:36:57 UTC*
  - bc89b297f8 Merge pull request #39228 from gtmanfred/2016.3
  - afee047b08 default to utf8 encoding if not specified
- **PR** #39231: (terminalmage) Add clarification for jenkins execution module @ *2017-02-08 19:34:45 UTC*
  - d9b0671dbd Merge pull request #39231 from terminalmage/clarify-jenkins-depends
  - ad1b1255f2 Add clarification for jenkins execution module
- **PR** #39232: (terminalmage) Avoid recursion in s3/svn ext_pillars @ *2017-02-08 19:33:28 UTC*
  - ddcff89a84 Merge pull request #39232 from terminalmage/issue21342
  - c88896c277 Avoid recursion in s3/svn ext_pillars
- **ISSUE** #38697: (fboismenu) On Windows, ip.get_all_interfaces returns at most 2 DNS/WINS Servers (refs: #38793)
  - **PR** #39230: (rallytime) Fix the win_ip_test failures
  - **PR** #38793: (fboismenu) Fix for #38697 (refs: #39197, #39230)
- **ISSUE** #33187: (usbportnoy) Deploy to jboss TypeError at boss7.py:469 (refs: #39761, #39170)
- **PR** #39199: (rallytime) Back-port #39170 to 2016.3 @ *2017-02-07 16:19:32 UTC*
  - **PR** #39170: (grep4linux) Added missing source_hash_name argument in get_managed function (refs: #39199)
  - df5f934c34 Merge pull request #39199 from rallytime/bp-39170
  - c129905310 Added missing source_hash_name argument in get_managed function Additional fix to #33187 Customer was still seeing errors, this should now work. Tested with 2015.8.13 and 2016.11.2
- **ISSUE** #37174: (mikeadamz) The State execution failed to record the order in which all states were executed spam while running pkg.upgrade from orchestration runner (refs: #39206)
- **PR** #39206: (cachedout) Ignore empty dicts in highstate outputter @ *2017-02-07 16:11:36 UTC*
  - 2621c119fd Merge pull request #39206 from cachedout/issue_issue_37174
  - be31e0559c Ignore empty dicts in highstate outputter

- **PR** #39209: (terminalmage) Sort the return list from the fileserver.envs runner @ *2017-02-07 16:07:08 UTC*
  - dd440452ea Merge pull request #39209 from terminalmage/sorted-envs
  - e6dda4a625 Sort the return list from the fileserver.envs runner
  - **PR** #39202: (rallytime) [2016.3] Pylint fix
- **ISSUE** #38697: (fboismenu) On Windows, ip.get_all_interfaces returns at most 2 DNS/WINS Servers (refs: #38793)
- **PR** #39197: (cachedout) Pr 38793 @ *2017-02-06 19:23:12 UTC*
  - **PR** #38793: (fboismenu) Fix for #38697 (refs: #39197, #39230)
  - ab76054127 Merge pull request #39197 from cachedout/pr-38793
  - f3d35fb5c6 Lint fixes
  - 624f25b78d Fix for #38697
- **PR** #39166: (Ch3LL) fix boto ec2 module create_image doc @ *2017-02-06 18:27:17 UTC*
  - fa45cbc359 Merge pull request #39166 from Ch3LL/fix_boto_ec2_docs
  - 90af696331 fix boto ec2 module create_image doc
- **PR** #39173: (rallytime) Restore "Salt Community" doc section @ *2017-02-06 18:19:11 UTC*
  - **PR** #30770: (jacobhammons) Doc restructuring, organization, and cleanup (refs: #39173)
  - **PR** #10792: (cachedout) Documentation overhaul (refs: #39173)
  - a40cb46249 Merge pull request #39173 from rallytime/restore-community-docs
  - 5aeddf42a0 Restore "Salt Community" doc section
- **ISSUE** #38704: (nasenbaer13) Archive extracted fails when another state run is queued (refs: #38705)
  - **PR** #39077: (terminalmage) Apply fix from #38705 to 2016.3 branch
  - **PR** #38705: (nasenbaer13) Fix for #38704 archive extracted and dockerio states (refs: #39077)
  - **PR** #39146: (gtmanfred) update vmware getting started doc
  - **PR** #39145: (garethgreenaway) [2016.3] Fix when targeting via pillar with Salt syndic
- **PR** #38804: (alexbleotu) Second attempt to fix prepending of root_dir to paths @ *2017-02-02 16:10:37 UTC*
  - cd8077ab81 Merge pull request #38804 from alexbleotu/root_dir_fix-2016.3-gh
  - b3bdd3b04a Add missing whiteline
  - c7715acd53 Merge pull request #3 from cro/ab_rootdirfix
    - e8cbafaaf1 When running testsuite, salt.syspaths.ROOT_DIR is often empty.
  - b12dd44a26 Merge pull request #1 from cro/ab_rootdirfix
    - bffc537aca Remove extra if statements (rstrip will check for the presence anyway).
  - 97521b3468 Second attempt to fix prepending of root_dir to paths
- **ISSUE** #39118: (bobrik) Minion ipv6 option is not documented (refs: #39289, #39131)
  - **PR** #39131: (bobrik) Clarify ipv6 option for minion and inteface for master, closes #39118
  - **PR** #39116: (terminalmage) Don't abort pillar.get with merge=True if default is None
- **PR** #39091: (terminalmage) Run test_valid_docs in batches @ *2017-02-01 19:09:05 UTC*

  - cc9b69b6bc Merge pull request #39091 from terminalmage/update-test-valid-docs
  - d76f0380d0 add debug logging for batch vars
  - b4afea2a25 Don't fail test if data is empty
  - b3a5d549c1 Account for trimmed value in 'salt -d' output
  - 909916c78e Run test_valid_docs in batches
  - **PR** #39081: (terminalmage) Move fileclient tests to tests/integration/fileserver/fileclient_test.py
  - **PR** #39067: (rallytime) Bump openstack deprecation notice to Oxygen
- **PR** #39047: (rallytime) [2016.3] Merge forward from 2015.8 to 2016.3 @ *2017-01-30 23:48:14 UTC*
  - a24af5ac46 Merge pull request #39047 from rallytime/merge-2016.3
  - b732a1f646 Merge branch '2015.8' into '2016.3'
  - 56ccae6ff7 Add 2015.8.14 release notes file (#39046)
  - 5943fe65d3 Update 2015.8.13 release notes (#39037)
  - **PR** #39045: (rallytime) Add 2016.3.6 release notes file
  - **PR** #39042: (rallytime) [2016.3] Update release numbers for doc build
  - **PR** #39038: (rallytime) Update 2016.3.5 release notes
- **PR** #39028: (terminalmage) Clarify delimiter argument @ *2017-01-30 18:20:26 UTC*
  - 5b09dc4198 Merge pull request #39028 from terminalmage/clarify-delimiter-argument
  - f29ef071f3 Clarify delimiter argument
  - **PR** #39030: (rallytime) Back-port #38972 to 2016.3
  - **PR** #38972: (rallytime) Add CLI Example for rest_sample_utils.get_test_string function (refs: #39030)
- **ISSUE** #38753: (alexbleotu) *__proxy__* dunder is not injected when invoking the *salt* variable in sls files (refs: #38899, #38829)
- **ISSUE** #38557: (alexbleotu) Proxy not working on develop (refs: #38829)
- **ISSUE** #38265: (mirceaulinic) *__utils__* object not available in proxy module (refs: #38899, #38829)
- **ISSUE** #32918: (mirceaulinic) Proxy minions reconnection (refs: #38829)
  - **PR** #38899: (cro) Enable __proxy__ availability in states, highstate, and utils. Enable __utils__ for proxies.
  - **PR** #38829: (cro) MANY dunder variable fixes for proxies + proxy keepalive from @mirceaulinic (refs: #38899)
  - **PR** #37864: (mirceaulinic) Proxy keepalive feature (refs: #38829)
- **ISSUE** #37938: (johje349) Memory leak in Reactor (refs: #38951)
- **ISSUE** #33890: (hvnsweeting) salt memleak when running state.sls (refs: #38951)
- **PR** #38951: (DmitryKuzmenko) Keep the only one record per module-function in depends decorator. @ *2017-01-27 17:05:42 UTC*
  - da96221741 Merge pull request #38951 from DSRCorporation/bugs/37938_fix_depends_decorator_memleak
  - 0b18f34678 Keep the only one record per module-function in depends decorator.
- **ISSUE** #34780: (joehoyle) S3fs broken in 2016.3.1 (refs: #38982)

- **PR** #38982: (rallytime) Set response when using "GET" method in s3 utils @ *2017-01-27 17:04:48 UTC*
  - 85165edb70 Merge pull request #38982 from rallytime/fix-34780
  - 1583c5579a Set response when using "GET" method in s3 utils
- **PR** #38989: (anlutro) Documentation: fix SLS in environment variable examples @ *2017-01-27 17:00:08 UTC*
  - cfdbc99e12 Merge pull request #38989 from alprs/docfix-state_pt3_environ
  - 52a9ad1c60 fix SLS in environment variable examples
- **PR** #39000: (rallytime) Skip the test_badload test until Jenkins move is complete @ *2017-01-27 16:58:21 UTC*
  - 55e4d2572e Merge pull request #39000 from rallytime/skip-badload-test
  - 4b3ff0fe0f Skip the test_badload test until Jenkins move is complete
- **PR** #38995: (terminalmage) Fix pillar.item docstring @ *2017-01-27 16:58:00 UTC*
  - fe054eb772 Merge pull request #38995 from terminalmage/fix-pillar.item-docstring
  - 06d094dd8f Fix pillar.item docstring
- **ISSUE** #34551: (mbom2004) salt.engines.logstash not loading (refs: #38950)
- **PR** #38950: (mbom2004) Fixed Logstash Engine in file logstash.py @ *2017-01-26 19:10:07 UTC*
  - b66b6f6423 Merge pull request #38950 from mbom2004/2016.3
  - c09f39d6c9 Remove unused json import
  - 249efa3068 Fixed Logstash Engine in file logstash.py
  - **PR** #38973: (rallytime) Handle changing "is_default" value in moto package for boto test mock
- **PR** #38952: (terminalmage) Make the ext_pillars available to pillar.ext tunable @ *2017-01-26 19:01:56 UTC*
  - b965b5dcc2 Merge pull request #38952 from terminalmage/zd1168
  - 6b014e53fc Rename on_demand_pillar to on_demand_ext_pillar
  - d216f90c63 Document new on_demand_pillar option and add to config template
  - 426b20f02f Add documentation for on-demand pillar to pillar.ext docstring
  - 7b10274b6b Make on-demand ext_pillars tunable
  - d54723ccae Add on_demand_pillar config option
- **ISSUE** #35777: (rallytime) Properly deprecate template context data in Fluorine (refs: #38948)
- **PR** #38948: (rallytime) Bump the template context deprecation version to Oxygen @ *2017-01-25 19:45:59 UTC*
  - 2c4ad85a78 Merge pull request #38948 from rallytime/bump-template-context-deprecation
  - 749e0031d7 Bump the template context deprecation version to Oxygen
- **PR** #38946: (rallytime) Back-port #37632 to 2016.3 @ *2017-01-25 19:40:40 UTC*
  - **PR** #37632: (twangboy) Fix versions report for Windows Server platforms (refs: #38946)
  - e4514ca7d8 Merge pull request #38946 from rallytime/bp-37632
  - ee37cdace9 Fix some lint
  - c08071e182 Fix versions report for server OSs
- **PR** #38913: (Adaephon-GH) Ignore plist files without Label key @ *2017-01-25 19:07:27 UTC*
  - 953a20350a Merge pull request #38913 from Adaephon-GH/patch-1

  - e2f4a16fdd Removing trailing whitespace
  - 616292c6b1 Ignore plist files without Label key
- **PR** #38917: (twangboy) Update Jinja2 to 2.9.4 @ *2017-01-25 19:05:38 UTC*
  - 826dce1059 Merge pull request #38917 from twangboy/update_jinja_mac
  - 62e608b627 Update Jinja2 to 2.9.4
- **ISSUE** #38540: (amendlik) API wheel client throws exception and success=true (refs: #38925)
- **ISSUE** #38537: (amendlik) API client wheel_async always returns status 500 (refs: #38925)
- **PR** #38925: (terminalmage) Fix two wheel issues in netapi @ *2017-01-25 18:28:52 UTC*
  - b27733cc33 Merge pull request #38925 from terminalmage/issue38540
  - 76392fc6ad Fix traceback when a netapi module uses wheel_async
  - bd4474fa62 Fix 'success' value for wheel commands
- **PR** #38926: (gtmanfred) add note about pysss for pam eauth @ *2017-01-25 18:12:20 UTC*
  - 618596f0cc Merge pull request #38926 from gtmanfred/2016.3
  - 9cae953c93 add note about pysss for pam eauth
- **ISSUE** #38825: (IshMalik) file.managed multiple sources for redundency failure (refs: #38847)
- **PR** #38847: (terminalmage) Catch MinionError in file.source_list @ *2017-01-24 16:03:10 UTC*
  - 405d86a2ca Merge pull request #38847 from terminalmage/issue38825
  - 11a47803ce Use log.exception() instead
  - e40fac589a Catch MinionError in file.source_list
- **ISSUE** #36121: (Ashald) TemplateNotFound/Unable to cache file (refs: #38875)
- **PR** #38875: (terminalmage) Reactor: fix traceback when salt:// path is nonexistant @ *2017-01-24 15:23:39 UTC*
  - b5df104fc2 Merge pull request #38875 from terminalmage/issue36121
  - fbc4d2a2c4 reactor: ensure glob_ref is a string
  - 2e443d79a3 cp.cache_file: add note re: return for nonexistant salt:// path
- **ISSUE** #37413: (Snarfingcode666) Salt-cloud vmware missing reboot command (refs: #38887, #38890)
- **PR** #38890: (cro) Backport #38887 to 2016.3: Enable resetting a VM via salt-cloud & VMware driver @ *2017-01-24 15:15:35 UTC*
  - **PR** #38887: (cro) Enable resetting a VM via salt-cloud & VMware driver (refs: #38890)
  - e9ebec4d80 Merge pull request #38890 from cro/vmware_reset_vm_20163
  - 0146562fb4 Call correct function for resetting a VM
- **PR** #38883: (techhat) Don't require text_out path to exist @ *2017-01-23 18:20:42 UTC*
  - **PR** #38867: (mchugh19) Touch deploy.sh before use (refs: #38883)
  - **PR** #32026: (techhat) Don't require the decode_out file to already exist (refs: #38883)
  - c3fbfcd231 Merge pull request #38883 from techhat/dontrequire
  - 67bc4d6687 Don't require text_out path to exist
- **PR** #38851: (terminalmage) Support docker-py 2.0 in dockerng @ *2017-01-23 16:48:12 UTC*

  - 6430a45196 Merge pull request #38851 from terminalmage/docker-py-2.0
  - 3c061b21fe Support docker-py 2.0 in dockerng
- **PR** #38844: (cachedout) Fix memory leak in HTTP client @ *2017-01-20 20:59:14 UTC*
  - ac8008d843 Merge pull request #38844 from cachedout/http_memory_leak
  - c46bf85518 Fix memory leak in HTTP client
- **ISSUE** #38798: (ripta) *match.compound* fails to match when pillar data is used (refs: #38823)
- **PR** #38823: (gtmanfred) pass pillar to compound matcher in match module @ *2017-01-20 19:19:09 UTC*
  - dfe6dfe963 Merge pull request #38823 from gtmanfred/2016.3
  - f0a71e8707 pass pillar to compound matcher in match module

28.2.48 Salt 2016.3.7 Release Notes

Version 2016.3.7 is a bugfix release for *2016.3.0*.

Security Fix

CVE-2017-12791 Maliciously crafted minion IDs can cause unwanted directory traversals on the Salt-master

This release corrects a flaw in minion ID validation which could allow certain minions to authenticate to a master despite not having the correct credentials. To exploit the vulnerability, an attacker must create a salt-minion with an ID containing characters that will cause a directory traversal. Credit for discovering the security flaw goes to: Vernhk@qq.com

Changelog for v2016.3.6..v2016.3.7

Generated at: 2018-05-27 14:09:17 UTC

- 11d176ff1b Add release notes for 2016.3.7 release
- dc649ded51 Add clean_id function to salt.utils.verify.py

28.2.49 Salt 2016.3.8 Release Notes

Version 2016.3.8 is a bugfix release for *2016.3.0*.

Security Fix

CVE-2017-14695 Directory traversal vulnerability in minion id validation in SaltStack. Allows remote minions with incorrect credentials to authenticate to a master via a crafted minion ID. Credit for discovering the security flaw goes to: Julian Brost (julian@0x4a42.net)

CVE-2017-14696 Remote Denial of Service with a specially crafted authentication request. Credit for discovering the security flaw goes to: Julian Brost (julian@0x4a42.net)

Changelog for v2016.3.7..v2016.3.8

Generated at: 2018-05-27 14:11:36 UTC

- 8cf08bd7be Update 2016.3.7 Release Notes
- 0425defe84 Do not allow IDs with null bytes in decoded payloads
- 31b38f50eb Don't allow path separators in minion ID

28.2.50 Salt 2016.3.9 Release Notes

Version 2016.3.9 is a bugfix release for *2016.3.0*.

Master Changes

The following options have been added to the master config file:

- *allow_minion_key_revoke* - This option controls whether a minion can request that the master revoke its key. When `True`, a minion can request a key revocation and the master will comply. If it is `False`, the key will not be revoked by the msater.
- `require_minion_sign_messages` - This requires that minions cryptographically sign the messages they publish to the master. If minions are not signing, then log this information at loglevel `INFO` and drop the message without acting on it.
- `drop_messages_signature_fail` - Drop messages from minions when their signatures do not validate. Note that when this option is `False` but *require_minion_sign_messages* is `True`, minions *MUST* sign their messages, but the validity of their signatures is ignored.
- `minion_sign_messages` - Causes the minion to cryptographically sign the payload of messages it places on the event bus for the master. The payloads are signed with the minion's private key so the master can verify the signature with its public key.

28.2.51 Salt 2015.8.0 Release Notes - Codename Beryllium

2015.8.0 Detailed Change List

Extended changelog courtesy of Todd Stansell (https://github.com/tjstansell/salt-changelogs)

Generated at: 2015-09-09T18:15:43Z

This list includes all pull requests merged into the 2015.8 branch between the forking of the branch from develop and the release of 2015.8.0.

Statistics:

- Total Merges: **682**
- Total Issue references: **342**
- Total PR references: **866**

Pull Requests:

- #26993: (*whiteinge*) Backport #26975
- #26970: (*cachedout*) Revert "better path query parsing in fileserver"
- #26980: (*terminalmage*) Use human-readable cachedirs for gitfs-backed winrepo

- #26969: (*TheBigBear*) URL of salt windows downloads has changed
- #26968: (*TheBigBear*) URL of salt windows downloads has changed
- #26958: (*s0undt3ch*) Bradthurber bootstrap command line help doc update
- #26949: (*rallytime*) Back-port #25148 to 2015.8
- #26914: (*cro*) Add salt-proxy script and manpage to setup.py so they will get installed.
- #26909: (*terminalmage*) Don't try to git clone from /tmp on Windows
- #26910: (*s0undt3ch*) Sometimes the event system is just too fast
- #26905: (*s0undt3ch*) Exit the loop if run_once is true
- #26897: (*msteed*) spm file hash part deux
- #26900: (*s0undt3ch*) If no tag is passed, don't actually subscribe to anything.
- #26880: (*s0undt3ch*) Restore backwards compatibility to *salt.utils.event*
- #26896: (*msteed*) spm remove: use pkgfiles to calculate file hashes
- #26891: (*jtand*) Fixed an unboundlocalerror
- #26892: (*cachedout*) Make the testing ioloop the current one
- #26886: (*jtand*) Gets the azure version correctly on python-azure 1.0.0
- #26870: (*rallytime*) Back-port #26834 to 2015.8
- #26865: (*dmurphy18*) Fix apt preferences for apts, repos for pbuilder building for Debian
- #26873: (*terminalmage*) Properly handle getting local config values in older git versions
- #26869: (*rallytime*) Fix provider –> driver change for salt-cloud lxc
- #26858: (*terminalmage*) Fix a couple version checks for git state and execution module
- #26853: (*UtahDave*) Fix salt-cloud on windows
- #26852: (*basepi*) [2015.8] Only reference msgpack if it imported successfully
- #26835: (*terminalmage*) Backport #26572 to 2015.8
- #26836: (*jacobhammons*) Added rst source for salt-proxy man page, added build and copy lines …
- #26818: (*terminalmage*) Support empty repositories in git.latest
- #26819: (*rallytime*) Make sure we're calling _validate_name in the correct place in 2015.8 Linode driver
- #26841: (*l2ol33rt*) Fix reference before assignment in sqs engine
- #26822: (*terminalmage*) Add some missing imports for masterless winrepo
- #26831: (*basepi*) [2015.8] Merge forward from 2015.5 to 2015.8
- #26826: (*techhat*) Pass a package name to unregister_file()
- #26757: (*cachedout*) Fix various filehandle leaks
- #26816: (*gtmanfred*) rev defaults to HEAD
- #26801: (*jacobhammons*) Added doc for dockerng minion configuration options
- #26808: (*anlutro*) Fix git init argument formatting
- #26807: (*terminalmage*) Move salt.utils.itersplit() to salt.utils.itertools.split()
- #26796: (*jacobhammons*) Add doc for __states__

- #26764: (*sjorge*) salt.utils.is_proxy() is no longer always true on SunOS/Illumos/SmartOS
- #26772: (*sjorge*) pull in smartos 'virt' module from develop
- #26726: (*terminalmage*) Redact HTTPS Basic Auth in states/funcs which deal with git remotes
- #26769: (*terminalmage*) Use –track to set tracking branch on older git versions
- #26765: (*basepi*) [2015.8] Merge forward from 2015.5 to 2015.8
- #26761: (*sjorge*) fix SPM paths on smartos/illumos esky
- #26751: (*terminalmage*) Fixes for masterless winrepo
- #26745: (*rallytime*) Make sure pyrax configs are in place before checking for deps
- #26746: (*rallytime*) Make sure nova configs are set before checking for dependencies
- #26750: (*basepi*) [2015.8] Add __utils__ to state modules
- #26752: (*cro*) Fix typo in some diagram labels
- #26747: (*basepi*) [2015.8] Add __states__ to state modules, for cross-calling states
- #26744: (*basepi*) [2015.8] Fix issue from #26717
- #26737: (*dmurphy18*) Fix to allow for package naming other than just salt
- #26742: (*rallytime*) Only warn about vsphere deprecation if vsphere is configured
- #26733: (*sjorge*) Refactor of smartos_vmadm module
- #26735: (*s0undt3ch*) Add *.hg* and *.cvs* to spm_build_exclude
- #26720: (*UtahDave*) Updates for winrepo in 2015.8 to support jinja, while maintaining backwards compat
- #26719: (*jodv*) Backport 26532 to 2015.8
- #26721: (*rallytime*) Linode Driver Cleanup
- #26707: (*techhat*) Add top_level_dir to FORMULAs
- #26723: (*s0undt3ch*) Handle SPM paths in the setup script
- #26717: (*basepi*) [2015.8] Revert loader changes from #26645
- #26712: (*techhat*) Move SPM paths around
- #26680: (*TheBigBear*) add more python libs info in '–versions-report'
- #26716: (*terminalmage*) Allow git identity to be a list
- #26691: (*garethgreenaway*) Fixes to ipset module for 2015.8
- #26701: (*kev009*) Ignore the first element of kern.disks split, which is the sysctl name (new disks grain)
- #26678: (*terminalmage*) Restructure git.latest rewrite to work better when following HEAD
- #26679: (*rallytime*) Back-port #26661 to 2015.8
- #26684: (*techhat*) Add reactor formulas to spm
- #26682: (*basepi*) [2015.8] Merge forward from 2015.5 to 2015.8
- #26671: (*rallytime*) Warn users if cloud driver dependencies are missing.
- #26674: (*rallytime*) Back-port #26583 to 2015.8
- #26670: (*techhat*) Set up SPM to install -conf packages
- #26657: (*jfindlay*) top file compilation fixes

- #26659: (*TheBigBear*) minor doc edits - spelling
- #26654: (*jfindlay*) merge #26650
- #26567: (*jtand*) Added git version check to git module
- #26649: (*twangboy*) Fixed Lint for real in win_repo.py
- #26608: (*jacobhammons*) 2015.8.0 release notes and doc/conf.py updates
- #26646: (*basepi*) [2015.8] Merge forward from 2015.5 to 2015.8
- #26645: (*rallytime*) Back-port #26390 to 2015.8
- #26642: (*twangboy*) Added function to render winrepo Jinja
- #26625: (*twangboy*) Correctly detect packages with no version, docs
- #26575: (*msteed*) Update spm for integration into raas
- #26635: (*cro*) Don't report windows as a proxy.
- #26622: (*rallytime*) [2015.8] Also add -Z to script args for cloud tests
- #26619: (*rallytime*) Apply cloud test fixes from 2015.5 to 2015.8
- #26603: (*terminalmage*) Fixes for git.latest, git module integration tests, etc.
- #26577: (*basepi*) [2015.8] Merge forward from 2015.5 to 2015.8
- #26534: (*cachedout*) Bump required Tornado version to 4.2.1
- #26566: (*cachedout*) Don't stacktrace trying to publish without a master
- #26541: (*terminalmage*) Make winrepo execution module use the same code as the runner
- #26530: (*basepi*) [2015.8] Merge forward from 2015.5 to 2015.8
- #26570: (*cachedout*) Fix haproxy docs to be valid
- #26562: (*cachedout*) Fix suprious error message with systemd-detect
- #26557: (*jfindlay*) add docs to #26550
- #26544: (*nmadhok*) Do not raise KeyError when calling avail_images if VM/template is in disconnected state
- #26501: (*terminalmage*) Update git_pillar docs, add git.list_worktrees function
- #26521: (*terminalmage*) Work around upstream git bug when cloning repo as root
- #26518: (*krak3n*) Fix for #25492
- #26514: (*evverx*) Unmask a runtime masked services too
- #26529: (*mnalt*) bugfix: fix service.enable for missing rc.conf
- #26516: (*techhat*) Move more path operations into SPM loader
- #26533: (*cachedout*) Fix too aggressive even init check
- #26522: (*cro*) Do not load package provider if its not a proxy
- #26531: (*cachedout*) Fix failing event tests and modify event init
- #26433: (*cro*) Add support for default proxy config options, change default location of proxy config and log to /etc/salt/proxy and /var/log/proxy
- #26504: (*nmadhok*) [Backport] Adding ability to specify the virtual hardware version when creating VM
- #26517: (*cachedout*) Better fix for opensuse tornado httpclient

- #26479: (*rallytime*) Don't allow VMs with duplicate names to be created in EC2/AWS
- #26488: (*cachedout*) Don't pass unsupported kwarg to tornado
- #26451: (*terminalmage*) Use 'rpm -qa' instead of repoquery to list installed packages
- #26491: (*jacobhammons*) doc site css fix for tiny fonts that appeared in code or pre tags in …
- #26442: (*rallytime*) Hide API Key from debug logs for Linode Driver
- #26441: (*rallytime*) Refactor a few linode functions to be useful with salt-cloud command
- #26485: (*s0undt3ch*) One more missed typo
- #26495: (*basepi*) [2015.8] Merge forward from 2015.5 to 2015.8
- #26492: (*cachedout*) Fix schedule test error on py26
- #26489: (*cachedout*) Fixing more tarfile tests on py2.6
- #26475: (*cachedout*) Better object checking on asyncreq cleanup
- #26477: (*cachedout*) Fix integration.modules.git.GitModuleTest.test_archive on py26
- #26469: (*jtand*) –annotate and –message aren't valid options in older versions of git.
- #26439: (*basepi*) [2015.8] Merge forward from 2015.5 to 2015.8
- #26464: (*rallytime*) Back-port #26456 to 2015.8
- #26463: (*rallytime*) Back-port #26455 to 2015.8
- #26449: (*s0undt3ch*) The CLI options are not meant to include underscores.
- #26270: (*sjorge*) salt.modules.network now supports SmartOS and SunOS < Solaris 11
- #26436: (*TheBigBear*) minor edits
- #26410: (*basepi*) [2015.8] Merge forward from 2015.5 to 2015.8
- #26427: (*anlutro*) git.latest with no rev: fix concatenation error (NoneType and str)
- #26307: (*cachedout*) Fix bug in top file ordering
- #26428: (*cro*) Update docs to reflect new pillar structure
- #26429: (*cachedout*) Add release note regarding tcp transport on freebsd
- #26418: (*driskell*) Fix forward-merged caching from 2015.5 into 2015.8 to be compatible with the new match_func
- #26252: (*DmitryKuzmenko*) Issues/24048 http client 2015.8
- #26413: (*evverx*) Fix service.{start,restart,reload,force-reload} for masked services
- #26393: (*dmurphy18*) Added option parameters to make_repo to allow for configuration settings
- #26422: (*TheBigBear*) no dots in SLS filename __AND__ any directories (incl git repos)
- #26323: (*0xf10e*) Fix Credentials used in *glance* Exec Module
- #26341: (*terminalmage*) Rewrite git state and execution modules
- #26419: (*terminalmage*) Only use pygit2.errors if it exists
- #26423: (*eliasp*) doc - Correct function name for peer configuration
- #26401: (*cachedout*) Adapt proxy minion to tornado (w/lint)
- #26400: (*rallytime*) Back-port #26318 to 2015.8

- #26397: (*s0undt3ch*) A single *isinstance()* check for all types is enough
- #26385: (*gtmanfred*) don't require volume endpoint in nova driver
- #26287: (*techhat*) Break out SPM components into loaders
- #26384: (*TheBigBear*) Fix shell quoting for cmd.run
- #26391: (*rallytime*) Back-port #26367 to 2015.8
- #26383: (*rallytime*) Allow the creation of a VM without a profile
- #26375: (*s0undt3ch*) [2015.8] Schema DictItem required attribute fixes
- #26363: (*garethgreenaway*) Fixes to mount state 2015.8
- #26347: (*0xf10e*) Load 'pkgng' as 'pkg' on FreeBSD 9 when *providers:pkg* == 'pkgng'
- #26361: (*TronPaul*) sign security token
- #26346: (*TronPaul*) Fix s3 using IAM credentials
- #26331: (*mnalt*) fix bug in sysrc to allow for empty rc variables
- #26334: (*rallytime*) Call salt.utils.cloud.bootstrap in GCE Driver provisioning
- #26308: (*dmurphy18*) Support for environment overrides building packages
- #26279: (*TheScriptSage*) Merge changes for pull'#26083'_ and pull'#25632'_ into 2015.8
- #26224: (*cachedout*) Cleanup of a few cases to move to salt.utils.fopen
- #26260: (*nmadhok*) Correct spelling of integration in docs
- #26226: (*rallytime*) Fix #25463
- #26248: (*nmadhok*) Initial commit of unit tests for vmware cloud driver
- #26228: (*basepi*) [2015.8] Merge forward from 2015.5 to 2015.8
- #26244: (*nmadhok*) Backport additions to VMware cloud driver from develop to 2015.8 branch
- #26235: (*sjorge*) salt.utils.is_smartos_zone, inverse of is_smartos_globalzone
- #26221: (*sjorge*) SmartOS grain fixes
- #26218: (*terminalmage*) Add warning about file.recurse unicode errors with vim swap files.
- #26214: (*rallytime*) Back-port #24878 to 2015.8
- #26211: (*techhat*) Move SPM to its own directory
- #26197: (*TronPaul*) Fix GitFS when whitelisting base
- #26200: (*anlutro*) Make it possible to run salt-cloud as current user
- #26201: (*kev009*) Avoid VBOX storage emulation bugs in FreeBSD disks grain
- #26188: (*basepi*) [2015.8] Merge forward from 2015.5 to 2015.8
- #26194: (*basepi*) Allow virtual grains to be generated even if virt-what is not available
- #26176: (*rallytime*) Back-port #26165 to 2015.8
- #26169: (*terminalmage*) Fix attribute error in gitfs' find_file functions
- #26170: (*nmadhok*) [Backport] Make sure variable is a dictionary before popping something from it.
- #26143: (*nmadhok*) VMware cloud driver fixes [forward port from 2015.5 into 2015.8]
- #26173: (*jacobhammons*) Updates to cloud docs for the provider > driver change

- #26125: (*evverx*) Use timedatectl set-timezone to tzsetting if available
- #26145: (*sjorge*) smartos_imgadm cleanup
- #26148: (*terminalmage*) Refactor winrepo support
- #26128: (*sjorge*) imgadm.avail should return multiple results
- #26109: (*jfindlay*) fix quote indent
- #26089: (*anlutro*) User state/module: fix coercing of None into string "None" in GECOS
- #26081: (*cachedout*) Move invocation routine up
- #26086: (*rallytime*) Back-port #26019 to 2015.8
- #26087: (*rallytime*) Back-port #26059 to 2015.8
- #26052: (*jtand*) Rh_ip fix
- #26078: (*cachedout*) Fix missing key in error return
- #26074: (*basepi*) [2015.8] Re-apply #25358 in 2015.8
- #26069: (*jfindlay*) fix win_firewall.delete_rule
- #26066: (*s0undt3ch*) [2015.8] Update to latest bootstrap stable release v2015.06.08
- #26049: (*basepi*) [2015.8] Merge forward from 2015.5 to 2015.8
- #26026: (*anlutro*) Fix httpasswd result false positive in test mode
- #26037: (*rallytime*) Back-port #25489 to 2015.8
- #26004: (*techhat*) Allow updating a single SPM repo at a time
- #26012: (*cachedout*) Merge kwargs into opts for tcp client
- #26007: (*anlutro*) file.managed: wrap os.remove in if isfile, don't remove on success
- #26009: (*terminalmage*) Add winrepo and dockerng information to 2015.8.0 release notes
- #26006: (*basepi*) Revert #25727 in favor of #25645
- #26001: (*cachedout*) Fix failing tests
- #25978: (*anlutro*) Correct service state changes in test mode
- #25982: (*sjorge*) salt.modules.smartos_* limit to global zone only
- #25989: (*rallytime*) Back-port #25832 to 2015.8
- #25988: (*cachedout*) Move #25642 to 2015.8
- #25999: (*s0undt3ch*) Include subschema defaults
- #25997: (*s0undt3ch*) Allow getting a defaults dictionary from schema defaults
- #25979: (*basepi*) [2015.8] Merge forward from 2015.5 to 2015.8
- #25902: (*basepi*) [2015.8] Merge forward from 2015.5 to 2015.8
- #25956: (*anlutro*) Fix user argument to cron functions
- #25946: (*sjorge*) Fix for salt.utils.decorators under esky
- #25957: (*anlutro*) Remove temporary file after file.managed with checkcmd
- #25874: (*rallytime*) Back-port #25668 to 2015.8
- #25929: (*sjorge*) salt.module.pkgin's __virtual__() should not return None if pkg_info is not present

- #25952: (*garethgreenaway*) Log when event.fire and event.fire_master fail 2015.8
- #25944: (*sjorge*) Smartos libcrypto nonesky fix
- #25906: (*dmurphy18*) Cherry-pick of pkgbuild changes from develop branch
- #25925: (*sjorge*) Create default log location in smartos esky buildscript
- #25928: (*cachedout*) Fix stacktrace for non-existant states
- #25922: (*jacksontj*) Correct max_wait -> max_auth_wait in MultiMinion
- #25907: (*rallytime*) Back-port #25892 to 2015.8
- #25910: (*terminalmage*) Pass osarch to check_32()
- #25849: (*basepi*) Repress template error for GPG renderer (can't seek an OrderedDict)
- #25868: (*rallytime*) Back-port #25404 to 2015.8
- #25896: (*cachedout*) Lint
- #25876: (*jacksontj*) Fixes for 2015.8
- #25867: (*rallytime*) Back-port #25370 to 2015.8
- #25845: (*jacobhammons*) updated versionadded
- #25836: (*jacksontj*) Keep track of SyncWrapper's IOLoop usage
- #25859: (*0xf10e*) warn_until(Carbon,…) instead of Boron
- #25505: (*0xf10e*) Glance state module for 2015.8 "Beryllium"
- #25843: (*jtand*) Fixed a lint error in parsers.py
- #25835: (*techhat*) spm update_repo doesn't always require arguments
- #25837: (*jacobhammons*) regenerated man pages
- #25830: (*sjorge*) Loading of libcrypto on smartos esky fixed
- #25808: (*jfindlay*) add highstate opts to config/__init__.py, update docs
- #25820: (*sjorge*) Prerequisite to fix the smartos libcrypto loading
- #25781: (*anlutro*) Fix iptables.build_rule
- #25764: (*gtmanfred*) allow use of cloudnetworks in ssh_interface
- #25736: (*jfindlay*) insert explicit formatter number
- #25742: (*rallytime*) Back-port #25731 to 2015.8
- #25741: (*rallytime*) Back-port #25727 to 2015.8
- #25712: (*cachedout*) Fix outputter for state.apply
- #25698: (*rallytime*) Back-port #25659 to 2015.8
- #25690: (*anlutro*) Fix highstate duration alignment (again)
- #25684: (*davidjb*) Fix doc around Include/Exclude for states
- #25549: (*techhat*) Switch Scaleway to salt.utils.cloud.bootstrap()
- #25667: (*jfindlay*) add 2015.8.0rc2 autogenerated changelog
- #25653: (*anlutro*) Properly align highstate duration sum
- #25663: (*rallytime*) Back-port #25638 to 2015.8

- #25639: (*terminalmage*) Don't do pre-flight check on git_pillar if it is not configured
- #25587: (*cachedout*) Fix prereq in salt.state
- #25628: (*anlutro*) Highstate output: show duration in seconds instead of milliseconds when appropriate
- #25631: (*basepi*) Remove trailing whitespace
- #25627: (*basepi*) [2015.8] Merge forward from 2015.5 to 2015.8
- #25626: (*basepi*) Fix the highstate outputter if 'duration' is not present
- #25601: (*terminalmage*) Fix error message when local bin pkg path is not absolute
- #25595: (*terminalmage*) Bring git_pillar up to feature parity with gitfs
- #25619: (*cachedout*) Lint stateconf changes
- #25578: (*davidjb*) Allow parent relative includes in state files
- #25610: (*s0undt3ch*) [2015.8] Update the bootstrap script to latest release v2015.07.22
- #25599: (*jfindlay*) fix transport settings in #25596
- #25596: (*jfindlay*) Tcp test
- #25591: (*garethgreenaway*) Return data for scheduled jobs in 2015.8 default to True.
- #25588: (*basepi*) Fix some of the retcode work from #23105
- #25583: (*jtand*) Fixed lint error where pprint wasn't imported.
- #25572: (*rallytime*) Back-port #25570 to 2015.8
- #25575: (*rallytime*) Make Sure Scaleway driver works with deprecation paths
- #25564: (*basepi*) [2015.8] Merge forward from 2015.5 to 2015.8
- #25566: (*techhat*) Fix download process for SPM repo updates
- #25553: (*techhat*) Switch SoftLayer to salt.utils.cloud.bootstrap()
- #25552: (*techhat*) Update pricing for SoftlayerHW
- #25547: (*techhat*) Switch Parallels to salt.utils.cloud.bootstrap()
- #25548: (*techhat*) Switch Proxmox to salt.utils.cloud.bootstrap()
- #25543: (*techhat*) Switch GCE to salt.utils.cloud.bootstrap()
- #25546: (*techhat*) Switch CloudStack to salt.utils.cloud.bootstrap()
- #25558: (*cachedout*) Lint config_test
- #25515: (*s0undt3ch*) salt.utils.schema fixes
- #25514: (*garethgreenaway*) fixes to schedule.add documentation in 2015.8
- #25508: (*s0undt3ch*) [2015.8] Update bootstrap script to latest stable release, v2015.07.17
- #25501: (*basepi*) Add optional job end time to the local_cache returner
- #25491: (*s0undt3ch*) Let's call it for what it is!
- #25462: (*rallytime*) Wrap is_profile_configrured calls in try/except block
- #25439: (*rallytime*) Reduce digital_ocean API call frequency
- #25451: (*s0undt3ch*) Salt-SSH Scan roster bugfixes (And Py3 support)
- #25449: (*ruzarowski*) Exclude dotfiles and directories from minion key lists (Fixes #25448)

- #25421: (*basepi*) [2015.8] Merge forward from 2015.5 to 2015.8
- #25412: (*basepi*) [2015.8] Merge forward from 2015.5 to 2015.8
- #25415: (*bechtoldt*) [docs] declare YAML as code block
- #25407: (*rallytime*) Back-port #23236 to 2015.8
- #25409: (*rallytime*) Back-port #24422 to 2015.8
- #25394: (*rallytime*) Back-port #25355 to 2015.8
- #25393: (*rallytime*) Back-port #25289 to 2015.8
- #25387: (*cachedout*) Lint #25319
- #25319: (*ruzarowski*) [cloud:EC2] Move SourceDest logic to _update_enis and add alias for delete_interface_on_terminate
- #25310: (*anlutro*) Add an "is list" test to the jinja environment
- #25264: (*ruzarowski*) Fix AttributeError in fileserver update_opts
- #25372: (*rallytime*) Don't stacktrace when provisioning instances with softlayer* drivers
- #25315: (*ruzarowski*) [cloud:EC2] Move handling of AssociatePublicIpAddress to associate_eip/allocate_new_eip logic depending on value type
- #25312: (*ruzarowski*) [cloud:EC2] Introduce eni Name property to set name tag value after its creation
- #25311: (*ruzarowski*) [cloud:EC2] Add ability to attach an existing eni
- #25280: (*rallytime*) Remove deprecation warnings for Beryllium
- #25329: (*twangboy*) Fixed some documentation errors
- #25300: (*s0undt3ch*) Fix ordering issue & Added requirements support
- #25283: (*jfindlay*) ensure ret is always defined
- #25252: (*jfindlay*) make args optional with default values in win_firewall.delete_rule
- #25257: (*notpeter*) Document SourceDestCheck added in #25242.
- #25298: (*twangboy*) Continue if profile not found
- #25296: (*twangboy*) Fixed file.comment for windows
- #25254: (*rallytime*) Change versionadded/changed references from Beryllium to 2015.8.0
- #25285: (*thusoy*) Remove error logging of missing victorops keys
- #25266: (*ruzarowski*) cloud: EC2 eni property SourceDestCheck is a AttributeBooleanValue
- #25216: (*jfindlay*) replace shell code with native python code
- #25278: (*rallytime*) Don't require size for all cloud drivers when checking profile configs
- #25271: (*basepi*) [2015.8] Merge forward from 2015.5 to 2015.8
- #25263: (*techhat*) Allow non-standard HTTP requests on tornado
- #25253: (*s0undt3ch*) Remove the deprecation warning. The driver has been renamed.
- #25248: (*techhat*) Do not resize while iterating
- #25244: (*rallytime*) Remove parted deprecations and fix failing tests
- #25242: (*ruzarowski*) Make SourceDestCheck flag available to network interface definition
- #25226: (*nmadhok*) Backporting fix for issue #25223 on 2015.8 branch

- #25234: (*krak3n*) Fix: Bug in boto_asg state argument passing to boto_asg module
- #25222: (*rallytime*) Back-port #25219 to 2015.8
- #25188: (*rallytime*) Use linode status descriptions instead of ints when logging status to CLI
- #25203: (*s0undt3ch*) Added DictConfig with tests & More tests
- #25189: (*basepi*) [2015.8] Merge forward from 2015.5 to 2015.8
- #25184: (*rallytime*) Back-port #25126 to 2015.8
- #25172: (*s0undt3ch*) Comment out imports while the YAML and RST rendering is not in-place.
- #25158: (*s0undt3ch*) Comment out not implemented code
- #25145: (*s0undt3ch*) Implement *oneOf*, *anyOf*, *allOf* and *not* with unit tests
- #25140: (*s0undt3ch*) Make the detection code work under Python 3.4
- #25131: (*s0undt3ch*) Array support in salt.utils.config
- #25130: (*basepi*) [2015.8] Merge forward from 2015.5 to 2015.8

The 2015.8.0 feature release of Salt contains several major new features. As usual the release notes are not exhaustive and primarily include the most notable additions and improvements. Hundreds of bugs have been fixed and many modules have been substantially updated and added.

New SaltStack Installation Repositories

SaltStack now provides installation repositories for several platforms, with more to come. See the following links for instructions:

- *Red Hat / CentOS 5, 6, 7*
- *Debian 8*
- *Windows*

Send Event on State Completion

A `fire_event` global state keyword argument was added that allows any state to send an event upon completion. Useful for custom progress bars and checking in on long state runs. See *fire_event*.

ZeroMQ socket monitoring

If `zmq_monitor` is enabled, log all ZMQ events for socket monitoring purposes. Verbose, but useful.

SPM (Salt Package Manager)

Allows Salt formulas to be packaged for ease of deployment. See *spm*.

Note: The spm executable was not included in the Debian or Ubuntu packages for the 2015.8.0 or the 2015.8.1 releases. This executable will be included in an upcoming release. As a workaround, copy the SPM script from the salt library installation into `/usr/local/bin` or your local equivalent.

Specify a Single Environment for Top Files

A new `default_top` option was added to load the state top file from a single, specific environment, rather than merging top data across all environments. Additionally, new `top_file_merge_strategy` and `env_order` options were added for more control over top file merging. See *The Top File*.

Tornado TCP Transport

Implemented a pure-TCP transport, in addition to ZeroMQ and RAET. The new transport uses Tornado, which allows Salt to use a standardized set of libraries for asynchronous behavior, which should greatly improve reliability and performance.

Note: Tornado is considered expiremental in this release. The following known issues were being investigated at the time of release:

- TCP tests show performance degredation over time (issue #26051)
- TCP transport stacktrace on windows minion: Future exception was never retrieved (issue #25718)
- [freebsd] TCP transport not working in 2015.8.0rc3 (issue #26364)

Proxy Minion Enhancements

Proxy Minions have undergone a significant overhaul in 2015.8, see Proxy Minion Enhancements.

Engines

Salt engines are long-running, external processes that leverage Salt. See *Salt Engines*.

Core Changes

- Add system version info to `versions_report`, which appears in both `salt --versions-report` and `salt '*' test.versions_report`. Also added is an alias `test.versions` to `test.versions_report`. (issue #21906)
- Add colorized console logging support. This is activated by using `%(colorlevel)s`, `%(colorname)s`, `%(colorprocess)s`, `%(colormsg)s` in `log_fmt_console` in the config file for any of `salt-master`, `salt-minion`, and `salt-cloud`.

Git Pillar

The git external pillar has been rewritten to bring it up to feature parity with `gitfs`. Support for pygit2 has been added, bringing with it the ability to access authenticated repositories.

Using the new features will require updates to the git ext_pillar configuration, further details can be found in the *pillar.git_pillar* docs.

Salt Cloud Improvements

- Pricing data from several cloud providers (GCE, DigitalOcean, SoftLayer_HW, EC2)
- All cloud providers now use standardized bootstrapping code.
- Modified the Linode Salt Cloud driver to use Linode's native API instead of depending on apache-libcloud or linode-python.

Salt Cloud Changes

- Changed the default behavior of `rename_on_destroy` to be set to `True` in the EC2 and AWS drivers.
- Changed the default behavior of the EC2 and AWS drivers to always check for duplicate names of VMs before trying to create a new VM. Will now throw an error similarly to other salt-cloud drivers when trying to create a VM of the same name, even if the VM is in the `terminated` state.
- When querying for VMs in `digital_ocean.py`, the number of VMs to include in a page was changed from 20 (default) to 200 to reduce the number of API calls to Digital Ocean.Ocean.

State and Execution Module Improvements

- New and improved Docker state and execution modules (`state` and `execution module`).

Git State and Execution Modules Rewritten

The git state and execution modules have gone through an extensive overhaul.

Changes in the `git.latest` State

- The `branch` argument has been added, allowing for a custom branch name to be used in the local checkout maintained by the *`git.latest`* state. This can be helpful in avoiding ambiguous refs in the local checkout when a tag is used as the `rev` argument. If no `branch` is specified, then the state uses the value of `rev` as the branch name.
- The `always_fetch` argument no longer has any effect, and will be removed in a future release. The state now detects whether or not a fetch is needed based on comparisons made between the local and remote repositories.
- The `force_fetch` argument has been added to force a fetch if the fetch is not a fast-forward (for instance, if someone has done a reset and force-pushed to the remote repository).
- The `remote_name` argument has been deprecated and renamed to `remote`.
- The `force` argument has been deprecated and renamed to `force_clone` to reduce ambiguity with the other "force" arguments.
- Using SHA1 hashes (full or shortened) in the `rev` argument is now properly supported.
- Non-fast-forward merges are now detected before the repository is updated, and the state will not update the repository if the change is not a fast-forward. Non-fast-forward updates must be overridden with the `force_reset` argument. If `force_reset` is set to `True`, the state will only reset the repository if it cannot be fast-forwarded. This is in contrast to the earlier behavior, in which a hard-reset would be performed every time the state was run if `force_reset` was set to `True`.
- A `git pull` is no longer performed by this state, dropped in favor of a fetch-and-merge (or fetch-and-reset) workflow.

`git.config_unset` state added

This state allows for configuration values (or entire keys) to be unset. See *here* for more information and example SLS.

git.config State Renamed to `git.config_set`

To reduce confusion after the addition of *git.config_unset*, the git.config state has been renamed to *git.config_set*. The old config.get name will still work for a couple releases, allowing time for SLS files to be updated.

In addition, this state now supports managing multivar git configuration values. See *here* for more information and example SLS.

Initial Support for Git Worktrees in Execution Module

Several functions have been added to the execution module to manage worktrees (a feature new to Git 2.5.0). State support does not exist yet, but will follow soon.

New Functions in Git Execution Module

- `git.config_get_regexp`
- *git.config_unset*
- *git.is_worktree*
- *git.list_branches*
- *git.list_tags*
- *git.list_worktrees*
- *git.merge_base*
- *git.merge_tree*
- *git.rev_parse*
- *git.version*
- *git.worktree_rm*
- *git.worktree_add*
- *git.worktree_prune*

Changes to Functions in Git Execution Module

`git.add`

- `--verbose` is now implied when running the `git add` command, to provide a list of the files added in the return data.

`git.archive`

- Now returns `True` when the `git archive` command was successful, and otherwise raises an error.
- The `overwrite` argument has been added to prevent an existing archive from being overwritten by this function.
- The `fmt` argument has been deprecated and renamed to `format`.
- Trailing slash no longer implied in `prefix` argument, must be included if this argument is passed.

`git.checkout`

- The `rev` argument is now optional when using `-b` or `-B` in `opts`, allowing for a branch to be created (or reset) using `HEAD` as the starting point.

`git.clone`

- The `name` argument has been added to specify the name of the directory in which to clone the repository. If this option is specified, then the clone will be made within the directory specified by the `cwd`, instead of at that location.
- The `repository` argument has been deprecated and renamed to `url`.

`git.config_get`

- The `setting_name` argument has been deprecated and renamed to `key`.
- The `global` argument has been added, to query the global git configuration
- The `all` argument has been added to return a list of all values for the specified key, allowing for all values in a multivar to be returned.
- The `cwd` argument is now optional if `global` is set to `True`

`git.config_set`

- The value(s) of the key being set are now returned
- The `setting_name` argument has been deprecated and renamed to `key`.
- The `setting_value` argument has been deprecated and renamed to `value`.
- The `is_global` argument has been deprecated and renamed to `global`.
- The `multivar` argument has been added to specify a list of values to set for the specified key. The `value` argument is not compatible with `multivar`.
- The `add` argument has been added to add a value to a key (this essentially just adds an `--add` to the `git config` command that is run to set the value).

`git.fetch`

- The `force` argument has been added to force the fetch when it is not a fast-forward. This could have been achieved in previous Salt versions by including `--force` in the `opts` argument, this argument is just for convenience and to match the usage of other functions with `force` arguments.
- The `refspecs` argument has been added to allow for one or more refspecs to be provided which override the one(s) specified by the **remote.remote_name.fetch** git configuration option.

`git.ls_remote`

- The `repository` argument has been deprecated and renamed to `remote`.
- The `branch` argument has been deprecated and renamed to `ref`.
- The `opts` argument has been added to allow for additional CLI options to be passed to the `git ls-remote` command.

`git.merge`

- The `branch` argument has been deprecated and renamed to `rev`.

`git.status`

- Return data has been changed from a list of lists to a dictionary containing lists of files in the modified, added, deleted, and untracked states.

`git.submodule`

- Added the `command` argument to allow for operations other than `update` to be run on submodules, and deprecated the `init` argument. To do a submodule update with `init=True` moving forward, use `command=update opts='--init'`.
- OpenStack Glance API V2 execution module
- Amazon VPC state module
- RallyDev execution module
- BambooHR execution module
- Stormpath execution, state modules
- Remove unused argument `timeout` in jboss7.status.
- Deprecate `enabled` argument in `pkgrepo.managed` in favor of `disabled`.
- Archive module changes: In the `archive.tar` and `archive.cmd_unzip` module functions, remove the arbitrary prefixing of the options string with `-`. An options string beginning with a `--long-option`, would have uncharacteristically needed its first `-` removed under the former scheme. Also, tar will parse its options differently if short options are used with or without a preceding `-`, so it is better to not confuse the user into thinking they're using the non- `-` format, when really they are using the with- `-` format.
- Added `__states__` to state modules, for cross-calling states. This enables using existing states when writing custom states. See *cross calling states*.

Windows Improvements

- Enhanced the windows minion silent installation with command line parameters to configure the salt master and minion name. See *Silent Installer Options*.
- Improved user management with additional capabilities in the user module for Windows.
- Improved patch management with a new module for managing windows updates (`win_wua`).
- Turned on multi-processing by default for windows in minion configuration.

Windows Software Repo Changes

A next-generation (ng) windows software repo is available for 2015.8.0 and later minions. When using this new repository, the repo cache is compiled on the Salt Minion, which enables pillar, grains and other things to be available during compilation time.

See the *Windows Software Repository* documentation for more information.

Changes to legacy Windows repository

If you have pre 2015.8 Windows minions connecting to your 2015.8 Salt master, you can continue to use the legacy Windows repository for these Salt minions.

If you were previously using this repository and have customized settings, be aware that several config options have been renamed to make their naming more consistent.

See the *Windows Software Repository* documentation for more information.

Win System Module

The unit of the `timeout` parameter in the `system.halt`, `system.poweroff`, `system.reboot`, and `system.shutdown` functions has been changed from seconds to minutes in order to be consistent with the linux timeout setting. (issue #24411) Optionally, the unit can be reverted to seconds by specifying `in_seconds=True`.

Other Improvements

- Sanitize sensitive fields in http.query
- Allow authorization to be read from Django and eauth
- Add templating to SMTP returner
- New REST module for SDB
- Added rest_timeout config option and timeout argument to jobs api call
- Provide config options for Raet lane and road buffer count. (Useful for BSD kernels)
- Implemented ZeroMQ socket monitor for master and minion
- Add end time to master job cache for jobs (optional, off by default)
- Tornado is now the default backend for http.request
- Support pillarenv selection as it's done for saltenv
- salt was updated to use python-crypto version 2.6.1, which removes the dependency on python-m2crypto.

Deprecations

- The `digital_ocean.py` Salt Cloud driver was removed in favor of the `digital_ocean_v2.py` driver as DigitalOcean has removed support for APIv1. The `digital_ocean_v2.py` was renamed to `digital_ocean.py` and supports DigitalOcean's APIv2.
- The `vsphere.py` Salt Cloud driver has been deprecated in favor of the `vmware.py` driver.
- The `openstack.py` Salt Cloud driver has been deprecated in favor of the `nova.py` driver.
- The use of `provider` in Salt Cloud provider files to define cloud drivers has been deprecated in favor of using `driver`. Both terms will work until the 2017.7.0 release of Salt. Example provider file:

```
my-ec2-cloud-config:
  id: 'HJGRYCILJLKJYG'
  key: 'kdjgfsgm;woormgl/aserigjksjdhasdfgn'
  private_key: /etc/salt/my_test_key.pem
  keyname: my_test_key
  securitygroup: default
  driver: ec2
```

- The use of `lock` has been deprecated and from `salt.utils.fopen`. `salt.utils.flopen` should be used instead.
- The following args have been deprecated from the `rabbitmq_vhost.present` state: `user`, `owner`, `conf`, `write`, `read`, and `runas`.
- The use of `runas` has been deprecated from the `rabbitmq_vhost.absent` state.
- Support for `output` in `mine.get` was removed. `--out` should be used instead.
- The use of `delim` was removed from the following functions in the `match` execution module: `pillar_pcre`, `pillar`, `grain_pcre`,

Security Fixes

CVE-2015-6918 - Git modules leaking HTTPS auth credentials to debug log

Updated the Git state and execution modules to no longer display HTTPS basic authentication credentials in loglevel debug output on the Salt master. These credentials are now replaced with `REDACTED` in the debug output. Thanks to Andreas Stieger <asteiger@suse.com> for bringing this to our attention.

Major Bug Fixes

- Fixed minion failover to next master on DNS errors (issue #21082)
- Fixed memory consumption in SaltEvents (issue #25557)
- Don't lookup outside system path in which() util (issue #24085)
- Fixed broken jobs rest api call (issue #23408)
- Fixed stale grains data using in modules (issue #24073)
- Added ssh_identities_only config flag for ssh-agent configured environments (issue #24096)
- Fixed "object has no attribute" errors for Raet transport (issue #21640)
- Flush event returners before master exit (issue #22814)
- Fix CommandExecutionError in grains generation with lspci missing (issue #23342)

- Fix salt-ssh against CentOS 7 when python-zmq not installed (issue #23503)
- Fix salt-ssh issues related to out-of-date six module (issue #20949)
- Fix salt-ssh thin generation after previous run was interrupted (issue #24376)
- Use proper line endings on Windows with "file.managed" w/contents (issue #25675)
- Fixed broken comment/uncomment functions in file.py (issue #24620)
- Fixed problem with unicode when changing computer description (issue #12255)
- Fixed problem with chocolatey module not loading (issue #25717)
- Fixed problem adding users to groups with spaces in the name (issue #25144)
- Fixed problem adding full name to user account (issue #25206)
- Fixed gem module stack trace (issue #21041)
- Fixed problem with file.managed when test=True (issue #20441)
- Fixed problem with powershell hanging while waiting for user input (issue #13943)
- Fixed problem where the salt-minion service would not consistently start (issue #25272)
- Fixed problem where pkg.refresh_db would return True even when winrepo.p was not found (issue #18919)
- Could someone please provide end to end example for Proxy Minion with REST (issue #25500)
- Proxy minions stopped working between 2014.7 and 2015.5 (issue #25053)
- Proxy minion documentation includes outdated code sample (issue #24018)
- Proxy Minion documentation missing grains example (issue #18273)
- Improve process management in proxy minion (issue #12024)
- Proxy minion never comes up with message ' I am XXX and I am not supposed to start any proxies.' (issue #25908)
- Fixed an issue that caused an exception when using Salt mine from pillar. (issue #11509)

28.2.52 Salt 2015.8.1 Release Notes

Version 2015.8.1 is a bugfix release for *2015.8.0*.

Statistics

- Total Merges: **201**
- Total Issue References: **39**
- Total PR References: **135**
- Contributors: **40** (DmitryKuzmenko, The-Loeki, TheBigBear, basepi, bechtoldt, bernieke, blueyed, cachedout, cedwards, clinta, cro, deuscapturus, dmurphy18, dsumsky, eliasp, flowhamster, isbm, jacksontj, jacobhammons, jfindlay, justinta, l2ol33rt, macgyver13, meggiebot, msteed, multani, nasenbaer13, perfinion, pprkut, rallytime, rhealitycheck, ruzarowski, ryan-lane, s0undt3ch, systembell, techhat, terminalmage, ticosax, twangboy, whiteinge)

Security Fixes

CVE-2015-6941 The Windows `user` module and `salt-cloud` display passwords in log when log level is set to `debug` or more verbose.

For the Windows `user` module, the password is now replaced with the string `XXX-REDACTED-XXX`.

For salt-cloud, debug logging no longer displays `win_password` and `sudo_password` authentication credentials.

CVE-2015-6918 Git state/execution modules log HTTPS auth credentials when log level is set to `debug` or more verbose.

These credentials are now replaced with `REDACTED` in the debug output. Thanks to Andreas Stieger <asteiger@suse.com> for bringing this to our attention.

Major Bug Fixes

- Add support for `spm.d/*.conf` configuration of SPM (issue #27010)
- Fix `proxy` grains breakage for non-proxy minions (issue #27039)
- Fix global key management for git state
- Fix passing http auth to `util.http` from `state.file` (issue #21917)
- Fix `multiprocessing: True` in windows (on by default`)
- Add `pkg.info` to pkg modules
- Fix name of `serial` grain (this was accidentally renamed in 2015.8.0`)
- Merge config values from `master.d`/`minion.d` conf files (rather than flat update`)
- Clean grains cache on grains sync (issue #19853)
- Remove streamed response for fileclient to avoid HTTP redirection problems (issue #27093)
- Fixed incorrect warning about `osrelease` grain (issue #27065)
- Fix authentication via Salt-API with tokens (issue #27270)
- Fix winrepo downloads from https locations (issue #27081)
- Fix potential error with salt-call as non-root user (issue #26889)
- Fix global minion provider overrides (issue #27209)
- Fix backward compatibility issues for pecl modules
- Fix Windows uninstaller to only remove `./bin`, `salt*`, `nssm.exe`, `uninst.exe` (issue #27383)
- Fix misc issues with mongo returner.
- Add sudo option to cloud config files (issue #27398)
- Fix regression in RunnerClient argument handling (issue #25107)
- Fix `dockerng.running` replacing creation hostconfig with runtime hostconfig (issue #27265)
- Fix dockerng.running replacing creation hostconfig with runtime hostconfig (issue #27265)
- Increased performance on boto asg/elb states due to `__states__` integration
- Windows minion no longer requires powershell to restart (issue #26629)
- Fix x509 module to support recent versions of OpenSSL (issue #27326)

- Some issues with proxy minions were corrected.

Known Issues

- Proxy minions currently cannot execute a highstate because of the way the proxymodule is being loaded internally. This will be fixed in a future release.

Changelog for v2015.8.0..v2015.8.1

Generated at: 2018-05-27 22:48:32 UTC

- **PR** #27588: (jfindlay) add autogenerated 2015.8.1 release notes @ *2015-10-01 04:52:32 UTC*
  - 87d86e4b3e Merge pull request #27588 from jfindlay/2015.8
  - f2eb20f26b add autogenerated 2015.8.1 release notes
- **PR** #27584: (jacobhammons) added changes list to 2015.8.1 release notes @ *2015-10-01 04:32:47 UTC*
  - f7510baf33 Merge pull request #27584 from jacobhammons/release-notes
  - ee4a3b3549 added changes list for 2015.8.1
- **ISSUE** #27532: (centromere) salt-cloud does not recognize terminated instances (refs: #27575)
- **PR** #27575: (rallytime) Don't report existing instances as running only if they're actually terminated in EC2 @ *2015-09-30 22:17:24 UTC*
  - 1a31b19f15 Merge pull request #27575 from rallytime/fix-27532
  - 57c6535fc2 Make sure message is the most accurate. Instance may be stopped or shutting down.
  - da6b4b3604 Don't report existing instances as running only if they're actually terminated
- **ISSUE** #27290: (pirogoeth) Grains set in minion_opts do not appear in a call to *grains.items*. (refs: #27573)
- **PR** #27573: (basepi) [2015.8] Use the custom yaml serializer for minion_opts for salt-ssh @ *2015-09-30 21:16:22 UTC*
  - bee78a4e5c Merge pull request #27573 from basepi/salt-ssh.grains.minion_opts.27290
  - 0785438b3f Use the custom yaml serializer for minion_opts for salt-ssh
- **ISSUE** #27326: (ralphvanetten) Signing the X509 CA certificate does not work on Debian 8 (refs: #27514)
- **PR** #27514: (clinta) Recent Versions of OpenSSL don't allow importing incomplete PEMs @ *2015-09-30 19:33:12 UTC*
  - a4a53ecff5 Merge pull request #27514 from clinta/2015.8-27326
  - 515e62bfa7 change "None" to empty string
  - 2989f24169 fix 27326 and fix minor errors in docs.
- **PR** #27564: (jacobhammons) Man pages @ *2015-09-30 19:29:37 UTC*
  - 6cf0228adc Merge pull request #27564 from jacobhammons/man-pages
  - cc37dc1087 updated version in salt.7
  - a9dcb23a13 regenerated man pages for 2015.8.1
- **ISSUE** #26629: (efficks) Windows minion: Remove powershell dependencies (refs: #27522)
- **PR** #27522: (twangboy) Removed dependency on powershell to restart salt-minion @ *2015-09-30 16:19:29 UTC*

  - fd11e0cd95 Merge pull request #27522 from twangboy/fix_26629
  - 163c54505d Fixed tests… hopefully
  - dc8c01ed07 Fixed some lint
  - 2cb0f12696 Removed dependency on powershell to restart salt-minion
- **PR** #27550: (rallytime) [2015.8] Clean up salt-cloud logging and make it more useful @ *2015-09-30 15:48:53 UTC*
  - eb76531e96 Merge pull request #27550 from rallytime/cloud-logging
  - 9e0fccd543 Don't commit private-ip changes from testing another bug…
  - 78c85fbb31 Add unit tests for new recursive function
  - d9a2dc6bc5 [2015.8] Clean up salt-cloud logging and make it more useful
- **ISSUE** #27281: (lrhazi) Wrong path for yum repo in installation-rhel-repo (refs: #27517)
- **ISSUE** #27179: (samhamilton) Debian Install Instructions Shows Two Different Repos (refs: #27517)
- **PR** #27517: (jacobhammons) Updated install docs @ *2015-09-30 15:19:51 UTC*
  - 1f7ea7c764 Merge pull request #27517 from jacobhammons/install-docs
  - 167fd2304e Fixed a duplicated link ID
  - c05fa71f91 Updated install docs Refs #27281 Refs #27179
- **PR** #27526: (eliasp) Add missing newlines before param listing to fix doc rendering @ *2015-09-30 15:19:04 UTC*
  - 2a4c11ae24 Merge pull request #27526 from eliasp/2015.8-modules.slack_notify-doc-params
  - 204e66943f Add missing newlines before param listing to fix doc rendering
- **PR** #27525: (basepi) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2015-09-30 03:38:22 UTC*
  - e5de9409c2 Merge pull request #27525 from basepi/merge-forward-2015.8
  - 1f3eb1c526 Remove useless mocked unit test
  - 73b90f155e Merge remote-tracking branch 'upstream/2015.5' into merge-forward-2015.8
    - 6d773f66c3 Merge pull request #27516 from basepi/merge-forward-2015.5
      - a08951f0fa Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.5
      - 5262f01325 Merge pull request #27335 from rallytime/cloud-logging-7
      - adeb1dcad4 Pylint Fix
      - 588c13783c Salt-cloud logging clean up for windows functions
      - 9b6000135c [2014.7] Fixup salt-cloud logging
    - 68d784c3dd Merge pull request #27472 from cachedout/fix_27447
      - 5e745ad6da Change recommeded schema for data field in mysql event table
    - ee6e0ed057 Merge pull request #27468 from cachedout/fix_27351
      - 0bc37c0d41 Fix test
      - f9a19720de fix sysctl truncating newline on os x
    - a214c7f84e Merge pull request #27479 from aboe76/fix_locale_suse
      - a8f2dad1be fix locale on opensuse and suse #27438
    - 931f593b51 Merge pull request #27483 from rallytime/fix-17103

    - 441241eb90 Change sync_outputters to sync_output for consistency, but alias sync_outputters
    - 105528720b Outputters should sync to output, not outputters, on the minion.
- 9c2c028953 Merge pull request #27484 from rallytime/bp-27434-and-27470
    - 5de2ee35ab Minor doc fixup.
    - af656c7e87 Doc: copy key to server via ssh-copy-id
- 927874d316 Merge pull request #27469 from twangboy/fix_27433
    - a996ea46e2 Added quotes to version numbers example
- 382a53403f Merge pull request #27467 from cachedout/lint_27375
    - 4e54a98f5e Lint #27375
    - 278ade52d2 file.managed: check contents_{pillar|grain} result
- ed6207a438 Merge pull request #27419 from rallytime/fix-9856
    - 551396564a Ammend error log to include multiple tips for troubleshooting.
- 73fa89edf7 Merge pull request #27426 from rallytime/fix-16753
    - f6cbd81e66 Don't stacktrace if there are conflicting id errors in highstate
- 5dd1b70475 Merge pull request #27408 from rallytime/fix-27406-for-2015.5
    - 39a4ae5a6c Remove hdd: 19 refs from SL docs - no longer available from SoftLayer.
    - de2f9234d3 Use correct default for bandwith
    - 42d8127f79 Don't set the optional_products default to a boolean, and then try to loop.
    - 9d8a3d8303 Fix avail_locations function for the softlayer_hw driver in 2015.5
- 8f9a3cfbaf Merge pull request #27410 from jacobhammons/doc-updates
    - a9fdecada1 Fix css layout Refs #27389 sample typo fix in linux_acl additional module folders listed in dynamic-modules
- 3746085587 Merge pull request #27336 from rallytime/cloud-logging-five
    - 7956b36076 [2015.5] Fixup salt-cloud logging
- 5a3be10a3e Merge pull request #27358 from lorengordon/escape-search-replacement-text
    - 88bb1fbfff Escape search replacement text, fixes #27356
- 6759f79d6d Merge pull request #27345 from rallytime/docs-for-19236
    - 1d3925bbfb Added version tag for ex_disk_type option
    - f23369300c Allow use of rst header links by separating options out from yaml example
- c2efb291e2 Merge pull request #26903 from bersace/fix-defaults-modules
    - 474d7afc95 fixup! Review defaults loading
    - 36141d226e fixup! Review defaults loading
    - 62b6495358 fixup! Review defaults loading
    - cf0624e8b8 fixup! Review defaults loading
    - 2c58bab977 fixup! Review defaults loading
    - 82c5b1d8fd Review defaults loading

    - a372466922 Merge pull request #27317 from efficks/fix27316
      - bf216c101e State unzip should use unzip command instead of unzip_cmd. Issue #27316
    - bd3771e80f Merge pull request #27309 from rallytime/fix-15514
      - 9383d91ff8 Change a value list to a comma-separated string in boto_route53.present
    - b5fe944875 Merge pull request #27311 from jfindlay/maxoc
      - 8ec2e921bd discuss replacement occurrences in file doc

- **PR** #27513: (terminalmage) Fix integration tests for worktree addition in git >= 2.6 @ *2015-09-29 18:39:19 UTC*
  - 0e37fb3bd3 Merge pull request #27513 from terminalmage/fix-worktree-tests
  - 519bdd6438 Fix integration tests for worktree addition in git >= 2.6
- **PR** #27510: (rallytime) Merge #27475 with test fixes @ *2015-09-29 18:34:32 UTC*
  - **PR** #27475: (ryan-lane) Use __states__ for calls to other boto states (refs: #27510)
  - e974a3c8aa Merge pull request #27510 from rallytime/ryan-lane-test-fix
  - cae2c4e715 Syntax fix
  - 458547ba03 Fix test failures for boto __state__ changes
  - 5e25454fc1 Followups for using __states__
  - a01f8ac62c Use __states__ for calls to other boto states
- **ISSUE** #27265: (Arabus) State: dockerng.running; creation hostconfig replaced with runtime hostconfig when using runtime options (refs: #27451)
- **PR** #27451: (ticosax) [dockerng] Enforce usage of host_config and require docker-py>=1.4.0 @ *2015-09-29 15:51:28 UTC*
  - d85b0cbd69 Merge pull request #27451 from ticosax/dockerng-host-config-support
  - b184faa55b Enforce usage of host_config and require docker-py>=1.4.0
- **PR** #27461: (cachedout) Only clean context if it exists @ *2015-09-29 15:49:52 UTC*
  - e8f58a6a3f Merge pull request #27461 from cachedout/clean_context_ioloop
  - 7367a4e32b Only clean context if it exists
- **ISSUE** #27220: (TheBigBear) [ERROR] Exception 'close_fds is not supported on Windows platforms if you redirect stdin/stdout/stderr' (refs: #27473)
- **PR** #27473: (terminalmage) salt.utils.gitfs: Don't use close_fds=True on Windows @ *2015-09-29 15:34:03 UTC*
  - 25a30a5621 Merge pull request #27473 from terminalmage/issue27220
  - fa70ef2e31 salt.utils.gitfs: Don't use close_fds=True on Windows
- **PR** #27496: (blueyed) Fix version reporting of gitpython @ *2015-09-29 15:31:48 UTC*
  - 3807cd5c4e Merge pull request #27496 from blueyed/fix-gitpython-version
  - d8969363c8 Fix version reporting of gitpython
- **PR** #27502: (ticosax) Add test to check we don't call inspect_image on absent images. @ *2015-09-29 15:15:09 UTC*
  - **PR** #25162: (ticosax) [dockerng] Do not call inspect_image if we know the image is not downloaded (refs: #27502)
  - 057fd0729d Merge pull request #27502 from ticosax/backport-test-from-develop

  - fadd9bd43e Add test to check we don't call inspect_image on absent images.
- **PR** #27497: (blueyed) dockerng: fix image_present for forced, non-existent image @ *2015-09-29 13:49:46 UTC*
  - f3da6e4bb3 Merge pull request #27497 from blueyed/dockerng-fix-404-private-forced
  - e3c66cea3a dockerng: fix image_present for forced, non-existent image
- **ISSUE** #27205: (msummers42) In git.config_set state CommandExecutionError occurs when global=True when using salt 2015.8.0 (refs: #27411)
- **PR** #27411: (terminalmage) Fix invocation of git.config_get and git.config_set @ *2015-09-28 22:53:01 UTC*
  - 284984e6ba Merge pull request #27411 from terminalmage/issue27205
  - c3a17ae992 add missing commas
  - f2751ef7c4 Fix shadowed outer-scope attributes
  - 81a6c27010 Fix invocation of git.config_get and git.config_set
- **ISSUE** #27217: (nasenbaer13) Gitfs cleans up wrong directories (refs: #27218, #27477, #27276, #27382)
- **PR** #27477: (terminalmage) Don't append role to hash_cachedir @ *2015-09-28 22:26:34 UTC*
  - cbcb5475b6 Merge pull request #27477 from terminalmage/issue27217
  - c185e99970 Second attempt to fix #27217
- **PR** #27474: (whiteinge) Add fake pymongo version attribute for the docs @ *2015-09-28 21:49:25 UTC*
  - 2f71833260 Merge pull request #27474 from whiteinge/docs-pymongo-fix
  - 64b54e668a Add fake pymongo version attribute for the docs
- **PR** #27466: (blueyed) Fix version reporting of python-gnupg and mysql-python @ *2015-09-28 20:25:01 UTC*
  - 9202f956f3 Merge pull request #27466 from blueyed/fix-gnupg-version
  - 9c1454fe59 Fix version reporting of mysql-python
  - 437fb4407e Fix version reporting of python-gnupg
- **PR** #27465: (ticosax) Fix usage of dockerng "cmd" was #27459 @ *2015-09-28 19:27:41 UTC*
  - **PR** #27459: (terminalmage) Fix usage of dockerng "cmd" (refs: #27465)
  - **PR** #27444: (ticosax) docker-py expect only *command* argument not *cmd* (refs: #27459)
  - **PR** #27331: (terminalmage) dockerng: Allow both cmd and command to be used to specify command (refs: #27459, #27444)
  - 6d8e9af297 Merge pull request #27465 from ticosax/fix-dockerng-cmd
  - a1ed6cda56 Skip test if docker-py is not installed
  - 6f7769aa94 Correct log messages/docstrings
  - cc8471bd1b dockerpy expect only *command* argument not *cmd*
- **ISSUE** #27409: (pcn) 2015.8.0 API (cherrypy) fails to lookup job id via pepper (refs: #27417)
- **ISSUE** #25107: (whiteinge) Regression in RunnerClient argument handling (refs: #25243)
- **PR** #27417: (whiteinge) Backport #25243 into 2015.8 @ *2015-09-28 19:15:53 UTC*
  - **PR** #25243: (DmitryKuzmenko) Runnerclient regression fix (refs: #27417)
  - aefe6d794a Merge pull request #27417 from whiteinge/bp-25243
  - 53e7a6b7c5 RunnerClient support old style commands with kwargs on top level.

  - 10b522b86c Revert "Fixed GET /jobs/<id> requests"

- **PR** #27423: (dmurphy18) Changes to support configurable repository for Debian / Ubuntu @ *2015-09-28 17:34:22 UTC*
  - a07411a4d9 Merge pull request #27423 from dmurphy18/dgm_envfix
  - 63407fd2a9 Changes to support configurable repository for Debian / Ubuntu
- **ISSUE** #26689: (double-yaya) Salt - SSH using machine IP to execute commands, without having to write a roster file (refs: #27398)
- **PR** #27428: (rallytime) Back-port #27398 to 2015.8 @ *2015-09-28 15:03:16 UTC*
  - **PR** #27398: (flowhamster) Allow cloud roster to use sudo (refs: #27428)
  - d4d96bb3fc Merge pull request #27428 from rallytime/bp-27398
  - 6969326ae2 doc: added documentation to cloud roster and fixed whitespace
  - b4334649d5 Allow cloud roster to use sudo
- **PR** #27429: (rallytime) Back-port #27344 to 2015.8 @ *2015-09-28 15:01:20 UTC*
  - **PR** #27344: (rhealitycheck) Mongo returners patch 1 (refs: #27429)
  - 668c69bd7e Merge pull request #27429 from rallytime/bp-27344
  - e39a57afe1 Update mongo_return.py
  - f796c9a44b Update mongo_return.py
  - 30d07cbb27 Update mongo_return.py
  - 44ef4b48fb Update mongo_future_return.py
  - 34b160b841 Update mongo_return.py
  - b2b5623da3 Update mongo_future_return.py
  - 07f9a8b95b Update mongo_return.py
  - b7ddc83b4d Update mongo_future_return.py
  - 540b3f2690 Update mongo_return.py
  - 405edd0718 Update mongo_future_return.py
  - 5c753a54ff Update mongo_return.py
  - 06e05befa7 Update mongo_future_return.py
- **PR** #27450: (ticosax) [dockerng] Fix typo in docstring @ *2015-09-28 14:27:35 UTC*
  - c639931340 Merge pull request #27450 from ticosax/fix-typo
  - 9cea62de67 Fix typo in docstring
- **PR** #27430: (jacksontj) Fix bug introduced in eee0291ff8b65ff1e22f4dc2447a74aa28a3ce7f @ *2015-09-26 01:09:40 UTC*
  - 333c305ba0 Merge pull request #27430 from jacksontj/2015.8
  - d2aff12f8f Fix bug introduced in eee0291ff8b65ff1e22f4dc2447a74aa28a3ce7f
- **PR** #27418: (terminalmage) Don't always remove dest path in salt.utils.files.rename() @ *2015-09-25 23:09:59 UTC*
  - 1f4ca089a2 Merge pull request #27418 from terminalmage/file-rename

  - 7bc0949d48 Don't always remove dest path in salt.utils.files.rename()
- **ISSUE** #27032: (lorengordon) Windows Installer: Please be more kind to existing configurations (refs: #27383)
- **PR** #27383: (twangboy) Uninstaller only removes specific files and dirs @ *2015-09-25 22:47:24 UTC*
  - ec5faf1829 Merge pull request #27383 from twangboy/fix_27032
  - 63a7305ae9 Uninstaller only removes specific files and dirs
- **PR** #27416: (rallytime) Back-port #27399 to 2015.8 @ *2015-09-25 22:39:07 UTC*
  - **PR** #27399: (multani) Various documentation fixes (refs: #27416)
  - 9ab3c6dc5d Merge pull request #27416 from rallytime/bp-27399
  - 1d848118c9 doc: fixed indentation in salt.renderers.jinja's documentation
  - f5d053a033 doc: fixed indentation in salt.modules.consul's documentation
  - 06beea6b2f doc: fix etcd state documentation typos
  - 97e69ebb97 doc: fix state's top documentation typo
  - b411730d60 doc: fix documentation formatting for state blockdev
  - ce91bb9446 doc: fix formatting in state boto_elb
  - c69229875e doc: fix links in Docker state documentation
  - 15b751d6e2 doc: Docker state use ports and not port_bindings anymore
  - 880b6e0944 doc: fix link to docker-py documentation
  - 33db0c27f8 doc: fix RAET links
  - e69ba2f943 doc: fix rendering of salt.states.hipchat
- **ISSUE** #27093: (TheBigBear) 2015.8.0 winrepo downloader corrupts some installers (refs: #27394, #27163)
- **PR** #27394: (jacksontj) Remove streamed response for fileclient to avoid HTTP redirection problems @ *2015-09-25 21:55:31 UTC*
  - **PR** #27163: (terminalmage) Workaround upstream tornado bug affecting redirects (refs: #27394)
  - 9842d9728b Merge pull request #27394 from jacksontj/2015.8
  - 01132c305c Re-add files.rename call instead of os.rename
  - acf2d51440 Remove streamed response for fileclient to avoid HTTP redirection problems
  - a6ecf35f25 Revert "Remove unused import"
  - 66c73a3996 Revert "Workaround upstream tornado bug affecting redirects"
- **PR** #27415: (ryan-lane) Backwards compat fixes for pecl module @ *2015-09-25 19:40:55 UTC*
  - 44b246bf93 Merge pull request #27415 from lyft/fix-pecl
  - 8be8ef585c Backwards compat fixes for pecl module
- **PR** #27407: (meggiebot) Adding stretch label definition @ *2015-09-25 18:10:46 UTC*
  - d76a77c911 Merge pull request #27407 from saltstack/meggiebot-patch-1
  - 1c779700f6 Adding stretch label definition
- **ISSUE** #27209: (justinta) Provider overrides appear to be broken (refs: #27388)
- **PR** #27388: (basepi) [2015.8] Fix global provider overrides @ *2015-09-25 16:49:03 UTC*

  - db6acfd832 Merge pull request #27388 from basepi/provider.overrides.27209
  - d87147e14b Don't use ret.items(), forces load of all modules
  - a5ee33a9ad pack __salt__ before loading provider overrides
- **ISSUE** #27354: (gravyboat) salt-ssh roster docs should note the requiretty option (refs: #27386)
- **PR** #27386: (rallytime) Document tty: True usage in salt-ssh roster file @ *2015-09-25 15:44:12 UTC*
  - b72e0b1133 Merge pull request #27386 from rallytime/fix-27354
  - 08c04da48b Document tty: True usage in salt-ssh roster file
- **PR** #27380: (justinta) Skipping Async tests @ *2015-09-25 15:13:04 UTC*
  - 51e765078a Merge pull request #27380 from jtand/async_tests
  - fd0dedeb99 Skipping Async tests
- **ISSUE** #27217: (nasenbaer13) Gitfs cleans up wrong directories (refs: #27218, #27477, #27276, #27382)
- **PR** #27382: (terminalmage) Revert "fixes #27217 clear_old_remotes clears wrong directory (gitfs)" @ *2015-09-24 22:54:23 UTC*
  - 633af56517 Merge pull request #27382 from terminalmage/revert-27218
  - 2379748f9e Revert "fixes #27217 clear_old_remotes clears wrong directory (gitfs)"
- **PR** #27361: (cro) Correct some issues with proxy minions @ *2015-09-24 16:03:38 UTC*
  - 12a021da11 Merge pull request #27361 from cro/pxm_doc
  - 1a2c41c9e3 Add versionadded.
  - 93a6397598 func_alias should be list_ and should have a corresponding list_ fn.
  - 0221f7ee4e Pylint
  - 3a297d8036 Add release notes for proxy fixes.
  - 39df44b841 Pylint
  - e3ebff9bce Fix some problems with the rest_sample, remove unnecessary file and make sure that rest_service has the right contents.
  - f4944fe68a Fix typo in docs
- **PR** #27364: (ruzarowski) SaltCloud[EC2] Fix missing credentials in modify_eni_properties api call @ *2015-09-24 13:55:39 UTC*
  - cff74510de Merge pull request #27364 from ruzarowski/2015.8-modify-eni-properties-api-call
  - 100eea46d5 Issue #27121 - Remove leftover code comment
  - c58e7a00f3 Issue #27121 - Attempt to fix missing credentials when modifying eni properties
  - 5d292a221e Merge remote-tracking branch 'upstream/2015.8' into 2015.8
  - 4dbd9ebb30 Merge remote-tracking branch 'upstream/2015.8' into 2015.8
- **PR** #27349: (jfindlay) add freebsd install docs to release notes @ *2015-09-24 13:51:02 UTC*
  - 928ef59a8a Merge pull request #27349 from jfindlay/doc_typos
  - e509cfca17 fix typo in 2015.8.0 pull list
  - 7137e731d3 add FreeBSD documentation to 2015.8.0 notes
- **ISSUE** #26889: (UtahDave) salt-call w/non root user outputs repeating error (refs: #27343)

- **PR** #27343: (cachedout) Close io loop before deleting attribute @ *2015-09-24 13:49:55 UTC*
    - 331230ea4f Merge pull request #27343 from cachedout/issue_26889
    - 2b648e51af Close io loop before deleting attribute
- **PR** #27337: (rallytime) [2015.8] Fixup salt-cloud logging @ *2015-09-24 13:49:17 UTC*
    - cd82ead005 Merge pull request #27337 from rallytime/cloud-logging-eight
    - ed18384108 Merge pull request #7 from jtand/cloud-logging-eight
        - a6c1d0b408 Fixed a bug where logging_command wasnt set as a key in a couple spots
    - 8bb7cb7ff4 Use correct indexes
    - c3483002b0 [2015.8] Fixup salt-cloud logging
- **PR** #27332: (terminalmage) Adjust dockerng/dockerio docstrings @ *2015-09-24 13:45:34 UTC*
    - b2f8418ffc Merge pull request #27332 from terminalmage/adjust-dockerng-docstring
    - bdbf4d8e5c Add deprecation notice to dockerio state module
    - 17829ab38d Fix name of dockerng module in dockerio docstring
    - ed5ae75180 Adjust dockerng docstrings
- **PR** #27353: (cachedout) Fix case where var not set in config @ *2015-09-23 21:45:32 UTC*
    - ac9e6c2532 Merge pull request #27353 from cachedout/fix_retry_get
    - ea286e1874 Fix case where var not set in config
- **ISSUE** #21390: (fyatzeck) Having trouble with GCE cloud profile assigning static IP and enabling IP forward (refs: #27350)
- **PR** #27350: (rallytime) Allow IP-forwarding in GCE driver @ *2015-09-23 21:36:41 UTC*
    - 3f6b06116f Merge pull request #27350 from rallytime/fix-21390
    - 2bf566d934 Allow IP-forwarding in GCE driver
    - 484015a7a3 Added version tag for ex_disk_type option
    - a71ebc97b2 Allow use of rst header links by separating options out from yaml example
- **ISSUE** #27103: (twangboy) Salt-Minion doesn't display logs for new processes with multiprocessing on (refs: #27305)
- **PR** #27305: (cachedout) Re-init logging system on Windows when using multiprocessing @ *2015-09-23 15:32:32 UTC*
    - 6f3da863fc Merge pull request #27305 from cachedout/issue_27103
    - 7a7492d186 Fix typo
    - 22c653482c Re-init logging system on Windows when using multiprocessing
- **PR** #27331: (terminalmage) dockerng: Allow both cmd and command to be used to specify command (refs: #27459, #27444) @ *2015-09-23 15:27:43 UTC*
    - 684e33aeb2 Merge pull request #27331 from terminalmage/dockerng-cmd
    - 7d4eaac8ae dockerng: Allow both cmd and command to be used to specify command
- **PR** #27327: (isbm) Fix a typo in the RPM output @ *2015-09-23 14:27:42 UTC*
    - a3f4fa1106 Merge pull request #27327 from isbm/isbm-pkg-info-typofix

  - 7912f8c13b Fix typo
- **PR** #27312: (basepi) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2015-09-22 22:52:14 UTC*
  - a789303d75 Merge pull request #27312 from basepi/merge-forward-2015.8
  - 647080d064 Add missing import
  - 95e70f0bef Merge remote-tracking branch 'upstream/2015.5' into merge-forward-2015.8
  - ca4597b93a Merge pull request #27310 from basepi/merge-forward-2015.5
    - 7b75e4aed1 Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.5
    - e90412d3b8 Merge pull request #27252 from jfindlay/version.2014.7
      - 3d28307a00 2014.7 -> 2014.7.0
  - 982c21c79f Merge pull request #27308 from terminalmage/fix-refresh_db-regression
    - 77686fb7ce Fix refresh_db regression in yumpkg.py
  - 775a4f9ad0 Merge pull request #27286 from terminalmage/return_retry_timer
    - 540a7dfcf1 Add default values for new minion config options
    - 453b883820 Add a configurable timer for minion return retries
  - 02482c0572 Merge pull request #27278 from rallytime/bp-27256
    - 1beddf6311 Fix error handling in salt.modules.file.statvfs
  - e36c019c37 Merge pull request #27277 from rallytime/bp-27230
    - 3ce77db1bc Fix typo in AWS doc config
  - b22286476e Merge pull request #27253 from jfindlay/version.2015.5
    - 967e3bb72a 2015.5 -> 2015.5.0
  - 51a0193b54 Merge pull request #27244 from garethgreenaway/ec2_create_snapshot_no_return_data_exception
    - 820fd576b9 Fixing the cause when the r_data from aws.query is empty and an exception happens when looking for the snapshotID
  - 26540f15bc Merge pull request #27231 from jfindlay/cronchange
    - 1e335297e2 only write cron file if it is changed
- **PR** #27303: (jacobhammons) Updated module doc index using https://github.com/saltstack/salt/pull… @ *2015-09-22 19:29:04 UTC*
  - c3b690273b Merge pull request #27303 from jacobhammons/ref-updates
  - 7ac98a03b6 Updated module doc index using https://github.com/saltstack/salt/pull/27203
- **ISSUE** #27081: (TheBigBear) winrepo - SSLError: [Errno 1] _ssl.c:510: error:14090086:SSL routines:SSL3_GET_SERVER_CERTIFICATE:certificate verify failed (refs: #27301)
- **PR** #27301: (twangboy) Pass ca_bundle for windows (fixes SSL Error) @ *2015-09-22 19:00:45 UTC*
  - aaa2db9943 Merge pull request #27301 from twangboy/fix_27081
  - 5c4f5f8944 Changed windows gate to check for verify_ssl option
  - e2fe5a60b5 Pass ca_bundle for windows (fixes SSL Error)
- **PR** #27300: (rallytime) Back-port #27287 to 2015.8 @ *2015-09-22 16:59:07 UTC*
  - **PR** #27287: (rhealitycheck) Mongo returners patch 1 (refs: #27300)

  - 55f4050146 Merge pull request #27300 from rallytime/bp-27287
  - e49a6dc449 Update mongo_return.py
  - 63153322b9 Update mongo_future_return.py
- **PR** #27288: (rallytime) Filter on 'name', not 'id', when listing images @ *2015-09-21 22:37:26 UTC*
  - d96462af48 Merge pull request #27288 from rallytime/do-cleanup
  - 6e16fad760 Use name in all places, not id.
  - 9b34542cb0 Filter on 'name', not 'id', when listing images
- **PR** #27283: (justinta) __grains__['osrelease'] returns a string @ *2015-09-21 19:18:44 UTC*
  - 688f24e9e4 Merge pull request #27283 from jtand/yumpkg_yum_fix
  - b73f5289b4 __grains__['osrelease'] returns a string. Cast to int for correct comparison
- **ISSUE** #27217: (nasenbaer13) Gitfs cleans up wrong directories (refs: #27218, #27477, #27276, #27382)
- **PR** #27276: (rallytime) Back-port #27218 to 2015.8 @ *2015-09-21 19:05:54 UTC*
  - **PR** #27218: (nasenbaer13) fixes #27217 clear_old_remotes clears wrong directory (gitfs) (refs: #27276)
  - 78d44a5c74 Merge pull request #27276 from rallytime/bp-27218
  - 8c0991d527 fixes #27217 clear_old_remotes clears wrong directory (gitfs)
- **PR** #27275: (rallytime) Back-port #27213 to 2015.8 @ *2015-09-21 19:05:18 UTC*
  - **PR** #27213: (macgyver13) Make get_event compatible with salt/client (refs: #27275)
  - d5ce81e8e7 Merge pull request #27275 from rallytime/bp-27213
  - 5d4c90c479 Make get_event compatible with salt/client
- **PR** #27274: (rallytime) Back-port #27272 to 2015.8 @ *2015-09-21 18:54:48 UTC*
  - **PR** #27272: (techhat) Make sure list_nodes_full contains a name attribute (refs: #27274)
  - 2be21d6451 Merge pull request #27274 from rallytime/bp-27272
  - f3ea3259a5 Make sure list_nodes_full contains a name attribute
- **PR** #27271: (isbm) Bugfix: crash on token authentication via API @ *2015-09-21 15:53:09 UTC*
  - c0943dd4d1 Merge pull request #27271 from isbm/isbm-bufix-27270
  - fc524c17b9 Reduce the criteria that would match empty iterables as well as None or False values
  - 3152af78b5 Fix the crash on token auth via API (http://git.io/vn4tx)
- **ISSUE** #19947: (gczuczy) Unable to supply provisioning script to softlayer create() (refs: #27251)
- **PR** #27251: (rallytime) Add support for post_uri in SoftLayer cloud drivers @ *2015-09-21 15:43:16 UTC*
  - b11ce6ac2a Merge pull request #27251 from rallytime/fix-19947
  - aafb776808 Add support for post_uri in SoftLayer cloud drivers
- **ISSUE** #21879: (bechtoldt) Reference pages in documentation are outdated again (refs: #27260, #25019, #21880)
- **ISSUE** #19262: (bechtoldt) salt.pillar.file_tree doesn't appear in the documentation (refs: #27260, #25019)
- **PR** #27260: (bechtoldt) add missing module doc references @ *2015-09-21 05:48:38 UTC*
  - **PR** #25019: (bechtoldt) add missing module documentation to references (refs: #27260)
  - **PR** #24421: (bechtoldt) add missing module documentation (refs: #27260, #25019)

  - **PR** #21880: (bechtoldt) update references, fixes #21879 (refs: #27260, #25019)
  - **PR** #20039: (bechtoldt) completing some doc references (refs: #27260, #25019)
  - de6e5abe6c Merge pull request #27260 from bechtoldt/missing_refs
  - 3a7d31a91c add missing module references
- **PR** #27254: (jfindlay) 2015.2,2015.8,Beryllium -> 2015.8.0 @ *2015-09-18 23:44:46 UTC*
  - 1a32b9f778 Merge pull request #27254 from jfindlay/version.2015.8
  - 8ea15f498e 2015.2,2015.8,Beryllium -> 2015.8.0
- **ISSUE** #25079: (jondonas) Salt-cloud does not check for duplicate ssh keys when using provider such as DigitalOcean (refs: #27245)
- **PR** #27245: (rallytime) If two ssh keynames are found in DigitalOcean, abort and warn the user. @ *2015-09-18 21:42:36 UTC*
  - f3a847823b Merge pull request #27245 from rallytime/fix-25079
  - 4b0f7cce1d If two ssh keynames are found in DigitalOcean, abort.
- **ISSUE** #27065: (lorengordon) 2015.8.0: yumpkg reporting ”Unexpected osrelease grain ’6.7’” (refs: #27241)
- **PR** #27241: (jfindlay) osrelease is only an integer for fedora @ *2015-09-18 21:40:50 UTC*
  - e4a5b004ae Merge pull request #27241 from jfindlay/yumwarn
  - 1f7570250f osrelease is only an integer for fedora
- **PR** #27234: (basepi) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2015-09-18 20:41:38 UTC*
  - f8e71f6d7d Merge pull request #27234 from basepi/merge-forward-2015.8
  - be2b0fc497 Merge remote-tracking branch ’upstream/2015.5’ into merge-forward-2015.8
    - 579f375f74 Merge pull request #27233 from basepi/release.notes.stubs
      - f4563ea9b7 Add stub release notes for 2015.5.6
    - f5a322e3f2 Merge pull request #27208 from basepi/nop.state.25423
      - 9414b05b2c Add test.nop example
      - a84ce67b8f Add test.nop state
    - 59a07cae68 Merge pull request #27201 from jfindlay/sshhash
      - 1b620b77cd rename hash_host arg to hash_known_hosts
      - 12f14ae37c update hash_known_hosts docs in ssh module
    - 560545c4c5 Merge pull request #27214 from jacksontj/2015.5
      - e7526bdb44 Correctly support https, port 443 is not a requirement
    - 7a34c7742d Merge pull request #27172 from rallytime/bp-27150
      - 0d7ee4b209 Merge config values from master.d/minion.d conf files
- **PR** #27240: (isbm) Backport of the fix of ’pkg.info*’ for Beryllium @ *2015-09-18 20:02:15 UTC*
  - 2d6c75cbd7 Merge pull request #27240 from isbm/isbm-pkg.info-tz-bugfix-backport-2015.8
  - 19a361851a Return install date only if possible.
  - ff857bc8aa Return RPM package time in UTC timezone
  - eaa0f370bf Remove time fraction and return ISO in UTC

  - ce9570fce6 Return UTC timestamp for modification of path.
- **ISSUE** #27222: (pprkut) Support firewalld zone configuration in network.managed state for rh7 systems (refs: #27223)
- **PR** #27223: (pprkut) Support firewalld per interface zone config on rh7 systems @ *2015-09-18 19:44:45 UTC*
  - 80a45b74ed Merge pull request #27223 from M2Mobi/zone
  - 48023669e7 Support permanent per interface firewalld zone configuration on rh7 systems.
  - **PR** #27239: (bechtoldt) test #27238 prevent keyerror when partition doesn't exist (refs: #27238)
- **PR** #27238: (bechtoldt) salt.modules.disk.percent() throws KeyError when partition doesn't exist (refs: #27239) @ *2015-09-18 19:37:00 UTC*
  - 652b2998af Merge pull request #27238 from bechtoldt/fix_disk_percent_keyerror
  - 0511f611bb prevent KeyError by checking whether partition even exists
- **PR** #27232: (basepi) [2015.8] Add stub release notes for 2015.8.1 @ *2015-09-18 16:53:01 UTC*
  - 253ac5e0c3 Merge pull request #27232 from basepi/release.notes.stubs
  - 25410706ee Add stub release notes for 2015.8.1
- **ISSUE** #24573: (bailsman) cloud.profile RuntimeError: dictionary changed size during iteration (refs: #27199)
- **PR** #27199: (rallytime) Avoid RunTimeError (dictionary changed size during iteration) with keys() @ *2015-09-18 15:44:27 UTC*
  - c542cd49d0 Merge pull request #27199 from rallytime/fix-24573
  - 6b2a00e947 Avoid RunTimeError (dictionary changed size during iteration) with keys()
- **PR** #27206: (rallytime) Don't repeat GCE setup instructions, and make the use of .json files clearer @ *2015-09-18 14:38:40 UTC*
  - 6b79ad69a9 Merge pull request #27206 from rallytime/gce-doc-cleanup
  - cced6e9031 Don't repeat GCE setup instructions, and make the use of .json files clearer
- **PR** #27210: (rallytime) Refactor some digital ocean functions @ *2015-09-18 14:38:01 UTC*
  - 1d022eb5de Merge pull request #27210 from rallytime/do-clean-up
  - 808a5b3b81 Make sure we set the full data to the ret variable
  - 9b635004e2 Refactor some digital_ocean functions to help simplify the driver
- **PR** #27197: (basepi) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2015-09-17 19:53:22 UTC*
  - 8c204a45ab Merge pull request #27197 from basepi/merge-forward-2015.8
  - 2c2a5f85ac Merge remote-tracking branch 'upstream/2015.5' into merge-forward-2015.8
    - e956d88f5f Merge pull request #27194 from rallytime/bp-27180
      - 327d343fef file copy ret result True if no change in test mode
    - a02d043309 Merge pull request #27176 from basepi/merge-forward-2015.5
      - 66f4641be3 Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.5
      - c186e51764 Merge pull request #27117 from jacobhammons/release-docs-2014.7
      - b69e11e0a4 made 2014.7 an archived release minor doc site updates
      - 69d758ee2b Merge pull request #27114 from cachedout/warn_on_insecure_log

    - 507fb04683 Issue warning that some log levels may contain sensitive data
    - aa71bae8aa Merge pull request #27075 from twangboy/fix_password_2014.7
    - c0689e3215 Replaced password with redacted when displayed
- de2027426e Merge pull request #27170 from rallytime/gce-docs
    - a07db909bd Update Getting Started with GCE docs to use cloud.profiles or cloud.profiles.d examples
- 28cfdfd067 Merge pull request #27167 from rallytime/bp-27148
    - d12be52355 Pass filepointers to the serialize load functions.
- 4495f4f4d0 Merge pull request #27168 from techhat/gateimpacket
    - cc448bfdc1 Add further gating of impacket library
- 3e5ef0dc30 Merge pull request #27166 from rallytime/fix-27100
    - 50fb3a489a Allow a full-query for EC2, even if there are no profiles defined
- f1c9de7ed9 Merge pull request #27162 from rallytime/softlayer-service
    - d281068c70 Be explicit in using ”SoftLayer” for service queries in SoftLayer drivers
- 59e9dfd8de Merge pull request #27149 from twangboy/fix_27133
    - 7992b7e20a Fixed some tests… hopefully…
    - d4c8e30f5d Fixed problem with add/remove path
- 097fcd1017 Merge pull request #27147 from rallytime/fix-11669
    - 55312ea03f Provide a more friendly error message.
    - 36555856c7 Enforce bounds in the GCE Regex
- f5c3f157dd Merge pull request #27128 from eguven/2015.5-fix-test-diff
    - ec2d68a84a don’t show diff for test run if show_diff=False
- 088b1dbb3e Merge pull request #27116 from jacobhammons/release-docs-2015.5
    - 6e323b6dd3 Update latest to 2015.8, 2015.5 is now previous Assorted style and minor updates
- 440855b182 Merge pull request #27033 from jfindlay/n0ne
    - 3334b9d548 fix comment and unit test for reg state
    - 391a09d5ac update reg state unit tests
    - ebbf2b05ca Fixed reg state module for None, 0, and ” values
- 35fc74132a Merge pull request #26942 from Arabus/fix-docker.run
    - e61e1de1f5 Fixes value typo for dockerio.loaded state
    - 39fa11b696 further linting
    - 4aec37397c Further Linting to quiet the linter
    - 7eff8ad070 Code Linting and cmd call fix
    - a51676e0eb Fixes #17088 olyif and unless should run on the host
    - d0c6128b8f Fixes #17088 retcode now returns True or False based on return status
    - 8b2e7cc4f5 Syntax clarification

- **PR** #27195: (jacobhammons) Fixed sphinx / latex build warnings and errors @ *2015-09-17 17:28:37 UTC*
    - 430c48c5ea Merge pull request #27195 from jacobhammons/doc-build
    - fad87e34a2 Fixed lint errors
    - e56f02b025 re-add cheatsheet do-over
    - 60a8330561 re-added cheatsheet.tex
    - f7a9e25d52 Fixed sphinx / latex build warnings and errors Added missing modules to contents
- **PR** #27182: (bernieke) fix restart_on_error @ *2015-09-17 17:24:01 UTC*
    - 8f8e75c5ff Merge pull request #27182 from Awingu/2015.8
    - 693b81f7e4 fix restart_on_error #27127
- **ISSUE** #27093: (TheBigBear) 2015.8.0 winrepo downloader corrupts some installers (refs: #27394, #27163)
- **PR** #27163: (terminalmage) Workaround upstream tornado bug affecting redirects (refs: #27394) @ *2015-09-17 16:09:01 UTC*
    - 97d2a5fddc Merge pull request #27163 from terminalmage/issue27093
    - 80b396db73 Handle potential ValueError when checking content length
    - a89c987943 Remove unused import
    - 469e18f74c Workaround upstream tornado bug affecting redirects
    - f2a562ac60 Add salt.utils.files.rename() for cross-platform renaming
- **ISSUE** #19954: (gczuczy) Multiple disks on softlayer (refs: #27173)
- **PR** #27177: (rallytime) Remove note - incorrect info @ *2015-09-17 01:34:04 UTC*
    - **PR** #27173: (rallytime) Add the ability to specify multiple disks on the SoftLayer driver (refs: #27177)
    - 65c59ec2ea Merge pull request #27177 from rallytime/fix-19954
    - 531b44243d Remove note - incorrect info
- **ISSUE** #19954: (gczuczy) Multiple disks on softlayer (refs: #27173)
- **PR** #27173: (rallytime) Add the ability to specify multiple disks on the SoftLayer driver (refs: #27177) @ *2015-09-17 00:32:57 UTC*
    - cbb7e7f1a5 Merge pull request #27173 from rallytime/fix-19954
    - 45c6aabde9 DeviceID '1' is reserved for the SWAP disk; let's skip it.
    - 54e104cf5b Don't stacktrace if local_disk isn't set
    - fe74d203f5 Add the ability to specify multiple disks on the SoftLayer driver
- **ISSUE** #22724: (ty2u) digital_ocean_v2.py doesn't restore snapshot (refs: #26824)
- **PR** #27164: (rallytime) Make sure changes from #26824 to digital_ocean_v2.py driver make it to digital_ocean.py in 2015.8 @ *2015-09-16 18:55:17 UTC*
    - **PR** #26824: (systembell) [salt-cloud] Fix creating droplet from snapshot in digital_ocean provider (refs: #27164)
    - 0e04588d58 Merge pull request #27164 from rallytime/add-26824-changes-to-2015.8
    - a44bd763dd Make sure changes from #26824 to digital_ocean_v2.py driver make it to digital_ocean.py in 2015.8
- **ISSUE** #19853: (ksalman) master needs a way to invalidate grains on the minion (refs: #27143)

- **PR** #27143: (cachedout) Clean grains cache on grains sync @ *2015-09-16 16:27:06 UTC*
  - 38d93a96fe Merge pull request #27143 from cachedout/clean_grains_cache_on_sync
  - 0a660a9f80 Break apart long line
  - 6de2c2a50c Better error checking
  - 252f7c7ea9 Clean grains cache on grains sync
- **ISSUE** #18582: (mainframe) Allow merging file_roots and pillar_roots from different config files included from master.d (refs: #27150)
- **PR** #27150: (cachedout) Merge config values from master.d/minion.d conf files (refs: #27172) @ *2015-09-16 15:36:41 UTC*
  - 626cbe61ce Merge pull request #27150 from cachedout/issue_18582
  - 6351a94d08 Merge config values from master.d/minion.d conf files
- **ISSUE** #27135: (SEJeff) Regression in core grains in the latest version of salt (refs: #27137)
- **PR** #27137: (jfindlay) revert serial grain regression @ *2015-09-15 21:52:25 UTC*
  - **PR** #22267: (The-Loeki) modify _hw core grains to use the new smbios module, add system uuid (refs: #27137)
  - 72fad569b0 Merge pull request #27137 from jfindlay/serial
  - 78c9687f0e revert serial grain regression
- **PR** #27144: (rallytime) Don't stacktrace on softlayer_hw.show_all_prices if a code isn't supplied @ *2015-09-15 21:52:09 UTC*
  - 58b56b9d78 Merge pull request #27144 from rallytime/softlayer-fixes
  - 3963a5cf0f Don't stacktrace on softlayer_hw.show_all_prices if a code isn't supplied
- **PR** #27139: (jacobhammons) Updated key instruction on rhel7 @ *2015-09-15 16:06:14 UTC*
  - b71de75c1c Merge pull request #27139 from jacobhammons/rhel-doc
  - 7ed9f6260f Updated key instruction on rhel7
- **PR** #27134: (isbm) Backport to 2015.8: "pkg.info" @ *2015-09-15 15:57:46 UTC*
  - 0d8248930e Merge pull request #27134 from isbm/isbm-pkg.info-backport-2015.8
  - b60e6a37a7 Lintfix: E7801, C0321
  - cb4706c7e8 Add license extraction for Dpkg.
  - 38753fe8b2 Enhance filter for the "technical" fields that are not generally needed as a package information for the CMDB
  - ffe8f14dae Implement additional package information merger
  - 2aafc469d0 Fix the size and installed-size keys
  - 3fc389435b Add homepage translator key
  - 25040c9c71 Docfix
  - 911bae1baf Add alias for 'info' of deprecation in v. Boron
  - 306958dad0 Fix renamed method
  - 6ba269fbc6 Remove 'N/A' when no data.
  - 137eb75ca2 Rename existing 'info' to 'info_available'

  - 7b376fd5c3 Implement compatible 'info_installed'. Returned keys are common to other systems with other package managers
  - ca7d0d5025 Implement compatible 'info_installed'. Returned keys are common to other systems with other package managers
  - c1faebf0b5 Implement compatible 'info_installed'. Returned keys are common to other systems with other package managers
  - f14f4036df Lint: regexp as a string
  - cabe863b81 Implement package info function
  - 0668f1da53 Implement getting package installation time
  - e03716e5b5 Implement getting general packages information
  - 8737d690fe Extract package description
  - a283d53737 Lintfix the regexp string
  - fc9c959678 Convert time to ISO 8601
  - 9fb9296276 Return a detailed information about package(s)
- **PR** #27119: (l2ol33rt) Boto dynamodb module should be using layer 2 abstractions @ *2015-09-15 14:09:57 UTC*
  - 7f512852ef Merge pull request #27119 from l2ol33rt/boto_dynamo_module_fix
  - 46c7aee367 Boto dynamodb util should be using layer 2 abstractions
- **PR** #27092: (perfinion) salt/master: chdir to root not homedir @ *2015-09-15 14:09:24 UTC*
  - 100e340111 Merge pull request #27092 from perfinion/chdir-fix-2015.8
  - 284d268855 salt/master: chdir to root not homedir
- **PR** #27131: (jacobhammons) Install docs @ *2015-09-15 12:34:38 UTC*
  - 7483556b5f Merge pull request #27131 from jacobhammons/install-docs
  - d1e8af9be6 added command to remove key from rhel6
  - 69d64f177d moved rhel5 commands to separate lines
  - 90431278ea Install instruction updates for rhel6 and debian
- **PR** #27124: (jfindlay) Backport #27123 @ *2015-09-15 08:37:43 UTC*
  - **PR** #27123: (cedwards) update for freebsd installation documentation (refs: #27124)
  - fc8afcc9f9 Merge pull request #27124 from jfindlay/bp-27123
  - 016fb5fafe Update freebsd.rst
  - 026fc9a884 update for freebsd installation documentation
- **PR** #27111: (basepi) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2015-09-15 07:29:30 UTC*
  - 0d62d3470c Merge pull request #27111 from basepi/merge-forward-2015.8
  - ab519fb5ff Remove heavily-mocked unit tests
  - 274464a85b Merge remote-tracking branch 'upstream/2015.5' into merge-forward-2015.8
    - 59f2a0c7ae Merge pull request #26977 from abh/2015.5-ntppeer
      - df3d6e817f Add support for PEERNTP network interface configuration on RH derived systems

- e05b1f3951 Merge pull request #27023 from jfindlay/htwebutilpass
    - 9f3d7890a6 add test support for htpasswd state mod
- 9f999c0027 Merge pull request #27074 from twangboy/fix_password_2015.5
    - fdd3537456 Replaced password with redacted when displayed
- 46b44f85ed Merge pull request #27073 from rallytime/remove-lxc-warning
    - 76c056d02b Remove "use develop branch" warning from LXC tutorial now that 2015.5.0 has been released
- caab21d99c Merge pull request #27054 from rallytime/bp-27029
    - 0be393be22 Removed check for no package name
- 0227e1cb57 Merge pull request #27053 from rallytime/bp-26992
    - 83798aff3c Do not use full return for documentation.
    - d9d5bbaa68 Summary requires full return information.
- b72a0ef86d Merge pull request #27052 from rallytime/bp-26930
    - d9787aa318 aptpkg.mod_repo: Raise when key_url doesn't exist
- 8b554dd16f Merge pull request #27049 from johanek/repoquery-dedupe
    - c113916a23 When running repoquery to check for available versions of packages, run once for all packages rather than once per package
- cc2cbf9869 Merge pull request #27070 from stanislavb/2015.5
    - 1e6e5ddc9c Deprecate salt.utils.iam in Carbon
- e23caa8ccf Merge pull request #27030 from jfindlay/winreg
    - 120fbe78e0 remove trailing line in win_path exec module
    - b36a7107b2 update win_path exec module unit tests
    - a2dc6f2dd7 Fixes win_path module, migrates from reg.(set|get)_key to reg.(set|get)_value
- 843c28b435 Merge pull request #27025 from cachedout/issue_25581
    - ecc09d9b93 Lint
    - bfcaab9ef4 Better try and error handling for prep_jid
- b9baa0b39a Merge pull request #27035 from terminalmage/useradd-contextmanager
    - e430e97f6c Update user states to reflect changes to login class handling
    - f24b979c7c useradd.py: Use contextmanager to prevent leaked filehandles
- 1cdfdf7a92 Merge pull request #27034 from rallytime/softlayer-doc-fix
    - cb641f8145 Update softlayer docs for where to find apikey
- 9e06d3f01a Merge pull request #27024 from rallytime/bp-27004
    - 54d6fcf4c7 Fix 'dict' object has no attribute split
    - bb29d73c71 Fix 'dict' object has no attribute split
    - 5f1a9c46aa Fix 'dict' object has no attribute split
    - 2bfdd9724e Fix 'dict' object has no attribute split
- 9ab2cae1e4 Merge pull request #27027 from rallytime/bp-27013

    - 19a6e9cb1c Remove unwanted debug statement.
- 2c8beb238f Merge pull request #27026 from rallytime/bp-27011
    - f8518d545f Move giant eventlisten.sh example out of the state.event docstring
- e8cdcc62f7 Merge pull request #26972 from twangboy/fix_20522
    - 0110786fa9 Catch the 404 error from fileclient
- fbc95f4685 Merge pull request #26951 from terminalmage/fix-timezone
    - 30a4915762 Update tests to reflect changes to timezone module
    - b6f926919f Fix timezone module for CentOS
- f2ad3c333c Merge pull request #26875 from marccardinal/patch-2
    - 36d5a62262 LXC gateway provisioned only when IP is provided
- 7b2e7b1b37 Merge pull request #26997 from twangboy/fix_symlink_windows
    - 89cc02d4e0 Added *versionadded*
    - 835177b0c8 Fixed symlinks for windows (don't use user root)
- 5389a85894 Merge pull request #27001 from twangboy/fix_reg_docs
    - 2980bbda17 Minor clarification
    - 4684b2ddd1 Added CLI example for reg.delete_key_recursive
- 37814f5dff Merge pull request #26996 from jacobhammons/beacon-doc
    - e475ea688e Fixed typo
    - 2401533d9e New content added to beacon docs.
- 4ba7eed711 Merge pull request #26868 from joejulian/2015.5_lvm_vg_symlink_fix
    - 3dfb33849a Use the actual device name when checking vgdisplay
- 1537e945be Merge pull request #26955 from dsumsky/s3-pillar-module-cache-fix-2015.5
    - 8219acffe7 - fixed pylint warnings
    - a3b10e8ab1 - fixed broken caching in S3 ext_pillar module (file_md5 was a list) - added debugging messages - static parameters are available as module parameters now
- 3e902e86b1 Merge pull request #26987 from rallytime/bp-26966
    - 6a29eac003 URL has changed
- eddb532713 Merge pull request #26915 from rallytime/joyent-tests
    - d4ad42d697 Update Joyent Cloud Tests
- f86814b2a4 Merge pull request #26971 from rallytime/reactor-doc-fix
    - 0214daad19 Fix a couple of typos in reactor docs
- 57b1080f94 Merge pull request #26976 from saltstack/revert-26899-fix_26730
    - 6dd54e6bec Revert "file.symlink gets windows account instead of root"
- 67be01f5fe Merge pull request #26975 from whiteinge/rest_cherrypy-integration
    - 9a0989585b Add additional 'groups' check to rest_cherrypy if groups are not used
    - d68aefcfde Remove mocks from rest_cherrypy integration tests

      - 2aa3da8911 Rename the rest_cherrypy tests to conform to our convention
    - 20a48f7f2e Merge pull request #26899 from twangboy/fix_26730
      - 9d9b3bb47a file.symlink gets windows account instead of root
    - dbc6b862f4 Merge pull request #26960 from rallytime/cherrypy-docs
      - c1420711db Fix bash code block formatting
    - f733e048c9 Merge pull request #26940 from rallytime/api-doc-fix
      - 00fe6a225c Fix minor doc typo in client api
    - de9350466e Merge pull request #26871 from rallytime/bp-26852
      - 5a4c8dd2f5 Only reference msgpack if it imported successfully
    - a563af29d3 Merge pull request #26851 from jacobhammons/doc-bugs
      - ac3bd47440 states/pkgrepo examples, suse installation updates Refs #26644 Refs #26638
    - 5b1b934192 Merge pull request #26817 from jfindlay/grouparg
      - 82d33939f3 modify groupadd for rhel 5
    - cdc0ea2fe3 Merge pull request #26824 from pravka/fix-droplet-creation-from-snapshot-in-dov2
      - 00e3192536 removing log
      - e4a82d78d9 removing stringification of every value in the image dict
      - cdc2b4584a fixing condition for slug check
    - 4af6951a4c Merge pull request #26823 from joejulian/ctlfix
      - a9928cb143 pep8 fixes
      - 6108ec4280 Gated dbus for os families that use it
      - e154c7b16f remove trailing spaces
      - c1c1266cc3 fix indent change
      - 0a35320aa7 Use dbus directly
    - a1749b76b8 Merge pull request #26820 from jfindlay/ctlfix
      - 3a2c0d5fbb add default param in _parse_localectl in locale mod
    - ff733547c4 Merge pull request #26821 from twangboy/fix_26788
      - cf979e4877 Fixed user.rename function in windows
    - c892be3255 Merge pull request #26803 from twangboy/fix_26754
      - 23576c65eb Added check for PyMySQL if MySQLdb import fails
    - 6edfa36083 Merge pull request #26815 from jfindlay/linstr
    - 2ff5823944 stringify linode id before performing str actions
- **PR** #27122: (terminalmage) Fix broken link to git-config(1) docs @ *2015-09-15 07:25:05 UTC*
  - 886e7bc234 Merge pull request #27122 from terminalmage/fix-broken-link
  - 0b212ea5b3 Fix broken link to git-config(1) docs
- **PR** #27115: (jacobhammons) Release docs @ *2015-09-14 22:19:18 UTC*
  - 551bbe70af Merge pull request #27115 from jacobhammons/release-docs

  - 42eaa80997 Restored missing css
  - 9ab642295e Fixed a release notes typo and bad file rename
  - daa3f4eee0 Updated release notes, change 2015.8 to latest release for doc site
  - d939a38c8c release notes updates
- **ISSUE** #11993: (UtahDave) salt-cloud -Q output not consistent across providers (refs: #27110)
- **PR** #27110: (rallytime) Make sure -Q output is consistent across salt-cloud drivers @ *2015-09-14 21:48:40 UTC*
  - 89c90df909 Merge pull request #27110 from rallytime/fix-11993
  - c1abc5a19f Remove implied Nones
  - 5d7d357cdd digital_ocean list_nodes function should list public and private ips like other drivers
  - 4b27aef406 Add 'name' to the output of salt-cloud -Q commands, where needed, for consistency.
- **PR** #27050: (twangboy) Turned multiprocessing on @ *2015-09-14 17:34:18 UTC*
  - 860de8d877 Merge pull request #27050 from twangboy/fix_minion_conf
  - 7e35b13022 Turned multiprocessing on
- **PR** #27086: (techhat) Document develoment of SPM loader modules @ *2015-09-13 04:52:55 UTC*
  - c78d833540 Merge pull request #27086 from techhat/spmdevdocs
  - ee0c8955dd Document develoment of SPM loader modules
- **ISSUE** #23125: (bemeyert) Elasticsearch as master_job_cache throws critical (refs: #26941)
- **PR** #26941: (msteed) Make elasticsearch work as master job cache @ *2015-09-12 17:13:44 UTC*
  - 25b11759f9 Merge pull request #26941 from msteed/issue-23125
  - ff88fe402c add versionadded info to save_load() & get_load()
  - 5d2fae8a89 make master job cache index configurable
  - bc041fa4a7 Merge branch 'issue-23125' of github.com:msteed/salt into issue-23125
    - 9aedc2662e issue-23125
  - 593c4d6b2f issue-23125
- **PR** #27080: (bechtoldt) [Proposal] Add Github SPM label for issues @ *2015-09-12 14:32:58 UTC*
  - b763d0ba52 Merge pull request #27080 from bechtoldt/spm_doc
  - b9e5095bf5 add GH issue label SPM to docs
- **PR** #27064: (twangboy) Fixed user docs @ *2015-09-11 22:37:19 UTC*
  - cf59a03432 Merge pull request #27064 from twangboy/user_docs
  - db03ca198e Fixed user docs
- **PR** #27072: (rallytime) Back-port #26840 to 2015.8 @ *2015-09-11 22:35:52 UTC*
  - **PR** #26840: (deuscapturus) Update http.py (refs: #27072)
  - 71c12cbf46 Merge pull request #27072 from rallytime/bp-26840
  - d0b9ececa4 Update http.py
- **PR** #27060: (cro) Fix grains breakage when hosts are not Linux, Windows, or SunOS @ *2015-09-11 17:28:49 UTC*

  - 0e7555089f Merge pull request #27060 from cro/proxy_grains_breakage
  - e697326f1b Don't check for proxy in the individual is_linux/is_windows/etc functions. This breaks too many things.
- **PR** #27051: (rallytime) Back-port #26953 to 2015.8 @ *2015-09-11 16:28:20 UTC*
  - **PR** #26953: (dsumsky) S3 ext_pillar module has broken caching mechanism (refs: #27051)
  - 8ee87b9f61 Merge pull request #27051 from rallytime/bp-26953
  - eac9d9aba9 Pylint Fix
  - 453440753c - fixed pylint warnings
  - b40dfa459e - fixed broken caching in S3 ext_pillar module (file_md5 was a list) - added debugging messages - static parameters are available as module parameters now
- **PR** #26864: (terminalmage) Only do git_pillar preflight checks on new-style git_pillar configs @ *2015-09-11 07:47:12 UTC*
  - 249f55cd8c Merge pull request #26864 from terminalmage/fix-git_pillar-tests
  - 0b5a653f7c Only do git_pillar preflight checks on new-style git_pillar configs
- **PR** #26967: (TheBigBear) new URL for windows salt downloads @ *2015-09-10 20:51:33 UTC*
  - efaedb8aea Merge pull request #26967 from TheBigBear/patch-4
  - 8d2c042cf7 new URL for windows salt downloads
- **PR** #26921: (terminalmage) Get rid of error in legacy git pillar when using branch mapping notation @ *2015-09-10 20:06:29 UTC*
  - 757d3c4eab Merge pull request #26921 from terminalmage/legacy_git_pillar_tests
  - 28e07d5d06 Get rid of error in legacy git pillar when using branch mapping notation
- **PR** #26923: (rallytime) Code clean up of cloud drivers and files @ *2015-09-10 16:37:26 UTC*
  - 68eb508e6c Merge pull request #26923 from rallytime/cloud-cleanup
  - bf33c99b08 Remove redundant parentheses
  - 5045989be7 Make sure function names comply
  - e327d9a8a4 Remove redundant parens
  - eee0291ff8 Code clean up of cloud drivers and files
- **PR** #27010: (rallytime) Back-port #26988 to 2015.8 @ *2015-09-10 16:30:30 UTC*
  - **PR** #26988: (s0undt3ch) Process *spm.d/*.conf* and add prefix root dir support to SPM directories (refs: #27010)
  - 590c46f4e3 Merge pull request #27010 from rallytime/bp-26988
  - 93b30b5ba8 Whitespace
  - 685fa911e7 Version Added for new apply_spm_config function
  - 9612a6c7ad Process *spm.d/*.conf* and add prefix root dir support to SPM directories
- **PR** #26985: (rallytime) Fix versionadded tag @ *2015-09-10 16:29:38 UTC*
  - ec185d77fa Merge pull request #26985 from rallytime/versionadded-fix
  - 79eb606cb7 Fix versionadded tag

28.2.53 Salt 2015.8.10 Release Notes

Version 2015.8.10 is a bugfix release for *2015.8.0*.

This release includes fixes for two issues discovered in *2015.8.9*:

- Pip state broken in 2015.8.9 with pip <6.0 (issue #33376)
- Fix traceback in logging for config validation (PR #33386)

Final Release of Debian 7 Packages

Regular security support for Debian 7 ended on April 25th 2016. As a result, 2016.3.1 and 2015.8.10 will be the last Salt releases for which Debian 7 packages are created.

Important Post-Upgrade Instructions for Linux Mint

As a result of some upstream changes, the `os` grain on Mint Linux is now being detected as `LinuxMint` (issue #33295). Run the following command **after you upgrade to 2015.8.10** to reset the `os` grain to `Mint` and the `os_family` grain to `Debian`:

```
salt -G 'os:LinuxMint' grains.setvals "{'os': 'Mint', 'os_family': 'Debian'}"
```

Changelog for v2015.8.9..v2015.8.10

Generated at: 2018-05-28 00:51:57 UTC

- c3d2c4eaae Fix traceback in logging for config validation (#33386)
- 2a060ea1e8 restore whitespace
- aa1f45d664 blast, put the try/except int he right place
- be1a7659a3 maintain the fallabck because I am totally sick of this crap

28.2.54 Salt 2015.8.11 Release Notes

Version 2015.8.11 is a bugfix release for *2015.8.0*.

Statistics

- Total Merges: **122**
- Total Issue References: **70**
- Total PR References: **221**
- Contributors: **48** (AAbouZaid, BlaineAtAffirm, DmitryKuzmenko, The-Loeki, abednarik, babilen, bebehei, cachedout, clinta, complexsplit, cro, danslimmon, dcolish, dincamihai, edgan, gerhardqux, ghedo, isbm, jacobhammons, jfindlay, jodv, justinta, l13t, lomeroe, lorengordon, lvg01, mcalmer, meaksh, morganwillcock, oeuftete, opdude, phistrom, rallytime, rmarcinik, ryan-lane, sacren, steverweber, techhat, tegbert, terminalmage, thatch45, the-glu, thegoodduke, ticosax, tveastman, twangboy, vutny, zer0def)

Ubuntu 16.04 Packages

SaltStack is now providing official Salt 2015.8 packages for Ubuntu 16.04.

Returner Changes

- Any returner which implements a `save_load` function is now required to accept a `minions` keyword argument. All returners which ship with Salt have been modified to do so.

New Master Configuration Parameter

- *`rotate_aes_key`* - if `True`, causes Salt to generate a new AES key whenever a minion key is deleted. This eliminates the chance that a deleted minion could continue to eavesdrop on communications with the master if it continues to run after its key is deleted.

Changelog for v2015.8.10..v2015.8.11

Generated at: 2018-05-28 01:16:12 UTC

- **PR** #34682: (jfindlay) update 2015.8.11 release notes
- **PR** #34676: (cachedout) Revert "Modify lodaer global test to use populated dunders" @ *2016-07-14 18:12:55 UTC*
  - 3192e1674b Merge pull request #34676 from cachedout/partial_revert_34644
  - 64a154826a Revert "Modify lodaer global test to use populated dunders"
- **PR** #34601: (lorengordon) Clarifies the proper way to reference states @ *2016-07-14 14:20:41 UTC*
  - 3b6f1089b2 Merge pull request #34601 from lorengordon/clarify-doc
  - bfe0dd0b8a Clarifies the proper way to reference states
  - **PR** saltstack/salt#34644: (cachedout) Cleanup loader errors (refs: #34651)
  - **PR** #34651: (rallytime) Lint 34644
  - **PR** #34647: (cachedout) Adjust the mine test a little bit to give it a better chance of success
- **PR** #34642: (justinta) Check that mysqladmin exists before running mysql integration tests @ *2016-07-13 18:12:44 UTC*
  - 8a0209101e Merge pull request #34642 from jtand/mysql_integration_cleanup
  - dd1559a599 Check that mysqladmin exists before running mysql integration tests.
- **PR** #34618: (justinta) Network state integration test test=True @ *2016-07-13 16:30:15 UTC*
  - 3e612c3794 Merge pull request #34618 from jtand/network_integration_fix
  - 34bcf9ccfc Changed network state test to use test=True
  - b2616833b0 Some small changes
  - ed59113e94 Change network state integration test to use test=True
- **PR** #34617: (rallytime) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2016-07-12 20:11:40 UTC*
  - 9f123543e5 Merge pull request #34617 from rallytime/merge-2015.8
  - 3026df346f Merge branch '2015.5' into '2015.8'

  - 57df38e685 Update github IP for ssh state integration tests (#34592)
  - 2e1007254b Avoid circular imports when calling salt.utils functions (#34584)
- **ISSUE** #33649: (tyhunt99) 2016.3.0 dockerng state fails comparing cmd configuration (refs: #salt-stack/salt`#33851`_, #33851)
  - **PR** saltstack/salt#33851: (ticosax) [dockerng] Add support for edge case when *Cmd* and *Entrypoint* can't be blanked (refs: #34593)
  - **PR** #34593: (rallytime) Back-port #33851 to 2015.8
  - **PR** #33851: (ticosax) [dockerng] Add support for edge case when *Cmd* and *Entrypoint* can't be blanked (refs: #34593)
  - **PR** #34590: (oeuftete) [2015.8] dockerng: When sorting list actual_data, make it a list
  - **PR** #34591: (justinta) Gate docker unit test to check for docker
  - **PR** #34560: (terminalmage) Add a bunch of documentation on getting files from other environments
- **ISSUE** #34397: (jaredhanson11) ignore_epoch needs to be passed through to version_cmp functions (refs: #34531)
- **PR** #34531: (terminalmage) Support ignore_epoch argument in version comparisons @ *2016-07-08 16:43:36 UTC*
  - 91e0656d44 Merge pull request #34531 from terminalmage/issue34397
  - d0fec1b8f6 salt/modules/zypper.py: accept ignore_epoch argument
  - 5ae9463c1f salt/modules/yumpkg.py: accept ignore_epoch argument
  - c2791117af salt/modules/rpm.py: accept ignore_epoch argument
  - c5de8b880d salt/modules/ebuild.py: accept ignore_epoch argument
  - 4ee8e8f037 salt/modules/aptpkg.py: accept ignore_epoch argument
  - 5b123b403c Pass ignore_epoch to salt.utils.compare_versions()
  - 07368fac40 Accept ignore_epoch argument for salt.utils.compare_versions()
- **PR** #34545: (terminalmage) Handle cases where Docker Remote API returns an empty ExecutionDriver @ *2016-07-08 16:34:30 UTC*
  - e99befad47 Merge pull request #34545 from terminalmage/docker-exec-driver
  - dd5838e242 Handle cases where Docker Remote API returns an empty ExecutionDriver
- **PR** #34546: (rallytime) Rename unit.states.boto_secgroup to unit.states.boto_secgroup_test @ *2016-07-08 16:16:42 UTC*
  - 7120d43df0 Merge pull request #34546 from rallytime/rename-boto-secgroup-test
  - f8a3622be7 Rename unit.states.boto_secgroup to unit.states.boto_secgroup_test
- **PR** #34537: (rallytime) Rename tests.unit.simple to tests.unit.simple_test @ *2016-07-08 00:08:36 UTC*
  - ca92061821 Merge pull request #34537 from rallytime/rename-simple-test
  - ceefb6e34c Rename tests.unit.simple to tests.unit.simple_test
  - **PR** #34527: (rallytime) [2015.8] Update bootstrap script to latest stable
  - **PR** #34521: (cachedout) Prevent many errors in the test suite in loader tests
  - **PR** #34507: (AAbouZaid) Fix wrong order of retention_policy_exists.

- **PR** #34518: (terminalmage) Fix pkg.latest integration test for non-LTS ubuntu @ *2016-07-07 19:29:13 UTC*
  - 685df80929 Merge pull request #34518 from terminalmage/fix-pkg.latest-test
  - 4aef44ecdf Fix pkg.latest integration test for non-LTS ubuntu
- **PR** #34513: (cachedout) Lower the log level for modules which cannot be loaded to trace @ *2016-07-07 17:00:48 UTC*
  - a516f116d1 Merge pull request #34513 from cachedout/lower_loader_log
  - 733c5d00c0 Lower the log level for modules which cannot be loaded to trace
- **PR** #34498: (rallytime) Use -O in the wget example in the bootstrap tutorial for the develop branch @ *2016-07-07 16:30:46 UTC*
  - 63f0451041 Merge pull request #34498 from rallytime/bootstrap-tutorial-doc-fix
  - 23c5739c3b Use -O in wget develop example in bootstrap tutorial
  - **PR** #34503: (rallytime) Rename some unit test files by adding _test
- **ISSUE** #34302: (ghost) Salt gitfs loads top files from all branches and tags (refs: #34505)
  - **PR** #34505: (terminalmage) Improve top file merging documentation
  - **PR** #34492: (zer0def) Gracefully handle non-XML output in GlusterFS execution module.
  - **PR** #34489: (justinta) Use skipTest for network state integration test
- **ISSUE** #34261: (vernondcole) salt.modules.dnsmasq documentation errors (refs: #34488, #34323)
  - **PR** #34488: (rallytime) Update dnsmasq.get_config docs to use correct config_file param.
- **PR** #34462: (terminalmage) Use –always when available to git describe @ *2016-07-06 03:59:33 UTC*
  - e2f576e847 Merge pull request #34462 from terminalmage/git-describe-always
  - 6ef7ee198e Restrict use of –always to git 1.5.6 and newer
  - c554b22fc8 modules/git: added –always parameter for git.describe().
- **PR** #34467: (rallytime) Back-port #34457 to 2015.8 @ *2016-07-06 03:56:58 UTC*
  - **PR** #34457: (ryan-lane) Only access key metadata if we found key metadata (refs: #34467)
  - 85f1f18239 Merge pull request #34467 from rallytime/bp-34457
  - 746883741f Only access key metadata if we found key metadata
- **PR** #34432: (twangboy) Fix file.append @ *2016-07-05 23:14:22 UTC*
  - 9e15337b74 Merge pull request #34432 from twangboy/fix_file.append
  - 13f11fddce Remove refactoring code
  - 78f7c530bb Remove unit tests, integration tests written
  - b83392edea Remove len() in favor of boolean test
  - 4373408163 Fix line error
  - 2479b53e2f Fix erroneous report on newline code
  - 75b6ed1fd5 Change back to binary read
  - 65753cff6d Use os.linesep instead of n
  - a55d63f086 Fix object names
  - 3e2fe12e5e Add new line if missing

  - 0b7821c8db Fix file.append state
- **PR** #34429: (terminalmage) Skip version checking for targeted packages in pkg.latest state @ *2016-07-05 17:50:41 UTC*
  - 91e095bb41 Merge pull request #34429 from terminalmage/pkg-latest-versioncheck
  - 667f31a72a Skip version checking for targeted packages in pkg.latest state
  - **PR** #34455: (cro) Forgot reference to inotify
- **PR** #34451: (rallytime) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2016-07-05 15:57:54 UTC*
  - 7bb0868c66 Merge pull request #34451 from rallytime/merge-2015.8
  - 55a91e22be Merge branch '2015.5' into '2015.8'
  - 8c72ee56e4 Merge pull request #34435 from cachedout/backport_config_dir_integration
    - 0e2c71a537 Backport change to integraiton test suite
- **ISSUE** #34390: (mgresser) Use rpmdev-vercmp to determine correct version of rpms in CentOS5 (refs: #34401)
- **PR** #34401: (terminalmage) Use rpmdev-vercmp as a fallback for version comparison on RHEL5 @ *2016-07-01 17:42:24 UTC*
  - e65d1ae374 Merge pull request #34401 from terminalmage/rpm-version_cmp
  - 7cefd4182d Use rpmdev-vercmp as a fallback for version comparison on RHEL5
- **PR** #34366: (steverweber) Update service.py @ *2016-07-01 17:40:31 UTC*
  - 5ddf417432 Merge pull request #34366 from steverweber/fix_servicerestart
  - 7847c39024 Update service.py
- **PR** #34426: (cro) Document that inotify is Linux only @ *2016-07-01 17:04:38 UTC*
  - 485454febb Merge pull request #34426 from cro/inotify-linux-only
  - 54a02f25ba Document that inotify is Linux only
- **PR** #34392: (cro) Clarify that salt-cloud doesn't get installed by bootstrap @ *2016-06-30 18:16:23 UTC*
  - fe18bbb527 Merge pull request #34392 from cro/salt-cloud-doc-clarify
  - 6cce575d40 Clarify that salt-cloud doesn't get installed by bootstrap
- **PR** #34373: (justinta) Network state integration test @ *2016-06-30 15:05:44 UTC*
  - 45b8fb10d7 Merge pull request #34373 from jtand/network_state_integration_test
  - 1d24053e36 network.system sls file
  - 4a9e6af542 network.routes sls file
  - 76c90b2ef6 network.managed sls file
  - 84a36369fa Added network state integration test
  - **PR** #34377: (terminalmage) Optimize pkg integration tests and add a couple new tests
- **PR** #34368: (rallytime) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2016-06-29 17:54:49 UTC*
  - af8ef1e461 Merge pull request #34368 from rallytime/merge-2015.8
  - 3bce0cb510 Merge branch '2015.5' into '2015.8'
  - 970aaa46d4 Merge pull request #34252 from gtmanfred/2015.5
    - 82183f1572 return list of nodes for lxc driver when called directly

  - **PR** #34344: (rallytime) Back-port #34324 to 2015.8
  - **PR** #34324: (cachedout) Test custom grains matcher (refs: #34344)
- **ISSUE** #33674: (edgan) salt-ssh returns a zero code on jinja template failure. (refs: #34316)
- **ISSUE** #28300: (srkunze) [salt-ssh] Does not return non-zero exit code (refs: #34316)
  - **PR** #34342: (rallytime) Back-port #34316 to 2015.8
  - **PR** #34316: (edgan) Making salt-ssh pass proper return codes for jinja rendering errors (refs: #34342)
  - **PR** #34339: (terminalmage) Revert py3modernize lint changes
- **PR** #34306: (ghedo) Fix iptables.flush state: Do not force 'filter' table when flushing @ *2016-06-28 19:03:14 UTC*
  - 046bdaa9f2 Merge pull request #34306 from ghedo/iptables_flush_table
  - 882c6c9c86 Do not force 'filter' table when flushing
- **ISSUE** #34261: (vernondcole) salt.modules.dnsmasq documentation errors (refs: #34488, #34323)
- **ISSUE** #34249: (ssgward) Clarify doc on file.copy (refs: #34323)
- **ISSUE** #34247: (gravyboat) Update logging docs to mention profile level (refs: #34323)
- **ISSUE** #33694: (hjc) Document That Local Files Can Be Used as a Source for File States (refs: #34323)
  - **PR** #34323: (jacobhammons) Doc clarifications to file modules, addition of new *profile* log lev…
  - **PR** #34325: (terminalmage) Remove unnecessarily-disabled sanity check
- **PR** #34335: (rallytime) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2016-06-28 15:07:15 UTC*
  - c5890a0eca Merge pull request #34335 from rallytime/merge-2015.8
  - 2296587536 Merge branch '2015.5' into '2015.8'
  - 6cce545d92 Merge pull request #34313 from rallytime/bootstrap-2015.5
    - c7db73be92 [2015.5] Update to latest bootstrap script v2016.06.27
  - **PR** #34319: (rallytime) Back-port #34244 to 2015.8
  - **PR** #34244: (the-glu) Typo in dockerio doc (refs: #34319)
- **PR** #34312: (rallytime) [2015.8] Update to latest bootstrap script v2016.06.27 @ *2016-06-27 18:59:59 UTC*
  - dd4c937009 Merge pull request #34312 from rallytime/bootstrap-2015.8
  - 944a393f89 [2015.8] Update to latest bootstrap script v2016.06.27
- **PR** #34307: (rallytime) Fix test example in integration testing docs @ *2016-06-27 17:41:24 UTC*
  - 91703d2dc4 Merge pull request #34307 from rallytime/fix-test-example
  - f44a0543fe Fix test example in integration testing docs
- **PR** #34233: (thegoodduke) ipset: fix the comment containing blank @ *2016-06-24 19:28:34 UTC*
  - d235b1245b Merge pull request #34233 from thegoodduke/for_2015.8_ipset
  - 4da5e35bf4 ipset: fix the comment containing blank
- **ISSUE** #34037: (bobrik) salt-call ignores –config-dir resulting in failing gpg renderer (refs: #34257)
- **PR** #34257: (rallytime) Use 'config_dir' setting instead of CONFIG_DIR in gpg renderer @ *2016-06-24 17:25:04 UTC*
  - 65c5675a3f Merge pull request #34257 from rallytime/fix-34037

  - d7a5e9b10e Remove test that doesn't actually test anything
  - c4c037d600 Use 'config_dir' setting instead of CONFIG_DIR in gpg renderer
- **ISSUE** #34273: (clinta) file.recurse does not properly cache files, adds a pipe to path (refs: #34274)
- **PR** #34274: (clinta) Don't escape source before calling managed @ *2016-06-24 17:23:35 UTC*
  - 203870f147 Merge pull request #34274 from clinta/2015.8
  - 6572454918 Don't escape source before calling managed
- **PR** #34258: (rallytime) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2016-06-24 14:27:06 UTC*
  - a59dc85a15 Merge pull request #34258 from rallytime/merge-2015.8
  - ea914b67cd Merge branch '2015.5' into '2015.8'
  - 8d5ed91980 Merge pull request #34225 from richardscollin/fix-win-set-datetime
    - 6286771ef7 Fix win_system.set_system_date_time
  - cb1e8bf082 Merge pull request #34232 from thegoodduke/for_2015.5_ipset
    - 344eb60762 ipset: fix commont containing blank
- **ISSUE** #33873: (hrumph) refresh: True not working with pkg.installed state (refs: #34093)
- **PR** #34093: (terminalmage) Catch CommandExecutionError in pkg states @ *2016-06-23 21:00:13 UTC*
  - 92962957c8 Merge pull request #34093 from terminalmage/issue33873
  - 5edb45d746 win_pkg: refresh pkg database if refresh=True passed to version() or list_pkgs()
  - 0078adee35 Catch CommandExecutionError in pkg states
- **PR** #34136: (meaksh) Fixed behavior for SUSE OS grains in 2015.8 @ *2016-06-23 20:24:58 UTC*
  - **PR** #34134: (meaksh) Fixed behavior for SUSE OS grains in 2016.3 (refs: #34136)
  - **PR** #33903: (meaksh) Fetching grains['os'] from /etc/os-release on SUSE systems if it is possible (refs: #34134)
  - cb5399787c Merge pull request #34136 from meaksh/salt-suse-os-detection-2015.8
  - 97f1958863 some cleanup and renaming
  - 72c8e5d78f better way to check for openSUSE Leap
  - 548971bdc9 Fix for SUSE OS grains in 2015.8
- **ISSUE** #34074: (fooka03) Unable to use S3 file backend with 2016.3.1 on Ubuntu 14.04 or 16.04 (refs: #34208)
- **ISSUE** #32916: (giannello) file.managed memory usage with s3 sources (refs: #33599)
  - **PR** #34208: (lomeroe) fix regression from #33681 which causes pulling a list of s3 objects …
  - **PR** #33681: (rallytime) Back-port #33599 to 2015.8 (refs: #34208)
  - **PR** #33599: (lomeroe) Fix s3 large file download (refs: #33681)
- **ISSUE** #34213: (terminalmage) gitfs w/pygit2 - corner case, traceback with short hexidecimal environment names (refs: #34218)
- **ISSUE** #34212: (terminalmage) gitfs: commit SHAs no longer available as fileserver environments (refs: #34218)
  - **PR** #34218: (terminalmage) Fix a pair of gitfs bugs
- **ISSUE** #34043: (rallytime) state execution stacktraces when psutil isn't installed (refs: #34182)

- **PR** #34182: (rallytime) Handle child PIDs differently depending on the availability of psutils @ *2016-06-22 19:22:06 UTC*
  - **PR** #33942: (cachedout) ZD 762 (refs: #34182)
  - 6d643cd528 Merge pull request #34182 from rallytime/fix-34043
  - b7d49c5052 Handle child PIDs differently depending on the availability of psutils
  - **PR** #34188: (terminalmage) Clarify pkg.list_repo_pkgs docstring for held packages
  - **PR** #34206: (terminalmage) Change target for dockerng assuming default status to Nitrogen release
- **PR** #34184: (rallytime) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2016-06-21 21:43:46 UTC*
  - 1c4369d093 Merge pull request #34184 from rallytime/merge-2015.8
  - 8e36e90966 Merge branch '2015.5' into '2015.8'
  - 5411ebb3b4 Merge pull request #34141 from jtand/boto_vpc_test_fix
    - b7ac6c735a Moved imports to top, out of _get_moto_version function
    - 02f9ba99ba Updated version check. Moved check into it's own function
    - d445026c56 Updated test to work with new moto version. Changed strings to unicode
- **ISSUE** #33972: (morganwillcock) 2016.3.1 breaks diskusage beacon (refs: #34176, #34103)
- **PR** #34176: (rallytime) Back-port #34103 to 2015.8 @ *2016-06-21 20:01:46 UTC*
  - **PR** #34103: (morganwillcock) Fix diskusage beacon (refs: #34176)
  - **PR** #33474: (cachedout) Fix diskusage beacon (refs: #34103)
  - c059d6c08c Merge pull request #34176 from rallytime/bp-34103
  - 2e5e7ed03c Fix diskusage beacon
- **ISSUE** #34114: (onorua) can't read PID from lock file due to exception if gitfs_global_lock is enabled (refs: #34179)
- **PR** #34179: (terminalmage) Raise the correct exception when gitfs lockfile is empty @ *2016-06-21 20:00:59 UTC*
  - 5cbaaed167 Merge pull request #34179 from terminalmage/issue34114
  - 86d1b8e864 Raise the correct exception when gitfs lockfile is empty
- **PR** #34178: (terminalmage) Remove unnecesssary comment @ *2016-06-21 19:15:37 UTC*
  - 67deded119 Merge pull request #34178 from terminalmage/remove-comment
  - 4965be72b1 Remove unnecesssary comment
  - **PR** #34165: (mcalmer) fix salt –summary to count not responding minions correctly
  - **PR** #34175: (rallytime) Back-port #34128 to 2015.8
  - **PR** #34128: (bebehei) doc: add missing dot (refs: #34175)
  - **PR** #34174: (rallytime) Back-port #34066 to 2015.8
  - **PR** #34066: (complexsplit) Typo fix (refs: #34174)
- **PR** #34077: (rallytime) Add some grains targeting tests @ *2016-06-21 16:06:30 UTC*
  - 3669048654 Merge pull request #34077 from rallytime/grains-tests
  - 2199bb8a78 Add integration tests for grains.append
  - 37cfe70724 Add some grains targeting tests

- **PR** #34142: (isbm) Move log message from INFO to DEBUG. @ *2016-06-20 18:57:34 UTC*
  - 65fba5b4d7 Merge pull request #34142 from isbm/isbm-getid-loglevel-shift
  - 236a67b702 Move log message from INFO to DEBUG.
  - **PR** #34100: (terminalmage) Update documentation on "refresh" behavior in pkg states
  - **PR** #34072: (jfindlay) modules.pkg int tests: skip refresh_db upon error
- **PR** #34069: (rallytime) Add a test to check for disconnected minion messaging @ *2016-06-16 21:18:38 UTC*
  - 1b76de1557 Merge pull request #34069 from rallytime/test-minion-return-message
  - 60561ac6fc Add a test to check for disconnected minion messaging
- **ISSUE** #30100: (armooo) Masterless gitfs performance (refs: #34048)
- **PR** #34048: (terminalmage) RFC: proposed fix for multiple fileserver updates in masterless runs @ *2016-06-16 21:10:59 UTC*
  - 3119693dac Merge pull request #34048 from terminalmage/issue30100
  - 715e7af8a4 Ensure only one fileserver update in a masterless run
- **PR** #34011: (rallytime) Back-port #33948 and #34009 to 2015.8 @ *2016-06-16 15:41:02 UTC*
  - **PR** #34009: (rallytime) Back-port #33948 to 2016.3 + add log message (refs: #34011)
  - **PR** #33948: (cachedout) Save an entire minion cache traversal on each master pub (refs: #34011, #34009)
  - dd03024931 Merge pull request #34011 from rallytime/bp-33948-2015.8
  - a4660d1ff7 Warn when custom returners don't have minions kwarg in save_load
  - 78befde62f Add note to release notes about returner minions kwarg change
  - 4e7f35fa36 Fix loop over cache in auth checking!
  - 06963e0505 Save an entire minion cache traversal on each master pub
  - **PR** #34051: (tegbert) Fixed a bug in the consul.py module that was preventing services
- **PR** #34045: (jacobhammons) Updated latest release version @ *2016-06-15 19:22:43 UTC*
  - 8ba117c7f6 Merge pull request #34045 from jacobhammons/release-prev
  - 43b4a12aa2 Updated latest release version
  - **PR** #34020: (twangboy) Always make changes to minion config if set (2015.8)
  - **PR** #34030: (vutny) More YAML indentation fixes in state module examples
  - **PR** #34003: (vutny) states.file: fix indentation in YAML examples (refs: #34030)
- **PR** #34018: (rallytime) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2016-06-14 22:53:19 UTC*
  - 5b5eae4ca9 Merge pull request #34018 from rallytime/merge-2015.8
  - 77f44f3087 Merge branch '2015.5' into '2015.8'
  - 871f7966ce Lint fix for #34000 (#34005)
  - f758e42172 Fix incorrectly written test (#34000)
  - cf6281b4cf Add loader.utils() example to calling minion_mods (#33953)
  - 6b98e8a9ea Merge pull request #33880 from terminalmage/zh744
    - ea726d11c8 pkg.uptodate: Pass kwargs to pkg.list_upgrades

        - de90b35d2b salt/modules/zypper.py: add fromrepo support to list_upgrades
        - 35fbb06df5 salt/modules/win_pkg.py: add kwargs to list_upgrades
        - bf5505f425 salt/modules/solarisips.py: add kwargs to list_upgrades
        - 6e89a8be98 salt/modules/pkgutil.py: add kwargs to list_upgrades
        - 5179dbcec4 salt/modules/pacman.py: add kwargs to list_upgrades
        - 46e5a52784 salt/modules/macports.py: add kwargs to list_upgrades
        - 76143b76ca salt/modules/ebuild.py: add kwargs to list_upgrades
        - b40fc9bc62 salt/modules/brew.py: add kwargs to list_upgrades
        - 4f11c16d86 salt/modules/aptpkg.py: add fromrepo support to list_upgrades
    - cb88960ed1 Merge pull request #33904 from rallytime/bp-33806
        - 638ccf501d Work around upstream cherrypy bug
    - **PR** #34003: (vutny) states.file: fix indentation in YAML examples (refs: #34030)
- **ISSUE** #20809: (lorengordon) Function pam.read_file is not available? (refs: #34002)
    - **PR** #34002: (lorengordon) Remove loader test for pam module
- **PR** #33990: (jacobhammons) Adds links to several current Salt-related projects @ *2016-06-14 01:15:20 UTC*
    - c4dab6a074 Merge pull request #33990 from jacobhammons/community-projects
    - b20213fd79 Adds links to several current Salt-related projects Removes the salt_projects.rst file which hasn't been updated in a long time, this is replaced by the updated topics/projects/index.rst file Adds a note about Salt Pack to the installation doc
- **PR** #33983: (twangboy) Clarify the *account_exists* parameter @ *2016-06-14 01:11:48 UTC*
    - 444c15792c Merge pull request #33983 from twangboy/fix_docs_join_domain
    - b057be04b4 Fix typo, more documentation
    - d8c2f3e57a Clarify the *account_exists* parameter
- **PR** #33951: (jfindlay) modules.gem int tests: more fixes @ *2016-06-14 00:46:43 UTC*
    - 9bd2317992 Merge pull request #33951 from jfindlay/gem_tests
    - 2eb633ccad modules.gem int tests: only check known installed gems
    - 9f3e18b037 modules.gem int tests: (un)install a non-core gem
- **PR** #33984: (jfindlay) Add docs and tests to disk state @ *2016-06-14 00:43:38 UTC*
    - 53baae6eb1 Merge pull request #33984 from jfindlay/disk_capacity
    - 6cbe31e6c2 states.disk: rewrite unit tests
    - 82c77b533f states.disk.status: validate percent values
    - aedc4e15e5 states.disk: add documentation
- **PR** #33985: (rallytime) Write some more simple batch command tests @ *2016-06-14 00:38:05 UTC*
    - fa5efb6a69 Merge pull request #33985 from rallytime/more-batch-tests
    - 3e7ab8c7b3 Write some more simple batch command tests
    - **PR** #33684: (jfindlay) add acl unit tests
    - **PR** #33942: (cachedout) ZD 762 (refs: #34182)

- **PR** #33946: (rallytime) Back-port #33698 to 2015.8 @ *2016-06-13 15:55:22 UTC*
  - **PR** #33698: (opdude) Vsphere fixes (refs: #33946)
  - 0281d491c6 Merge pull request #33946 from rallytime/bp-33698
  - 5fdfed1cb9 Make sure we only use GetConnection if we are using a proxy salt minion
  - 1505c5724b Fix a bug with self signed certificates and creating a new VM
- **ISSUE** #33911: (xlotlu) salt-ssh + grains.filter_by Type error: filter_by() got an unexpected keyword argument 'base' (refs: #33952)
- **PR** #33952: (rallytime) Add base argument to salt-ssh grains wrapper for filter_by func @ *2016-06-13 15:51:33 UTC*
  - dff3f51955 Merge pull request #33952 from rallytime/fix-33911
  - 03b7cbbd2c Add base argument to salt-ssh grains wrapper for filter_by func
  - **PR** #33962: (jacobhammons) Adds a "Generated on <timestamp>" line to the html footer
- **ISSUE** #29525: (apergos) master config setting ping_on_rotate is broken if minion_data_cache is disabled (refs: #33765)
  - **PR** #33765: (cachedout) Correct issue with ping on rotate with minion cache
- **PR** #33888: (jfindlay) random.org checks @ *2016-06-10 15:45:07 UTC*
  - 378dd7ca06 Merge pull request #33888 from jfindlay/random_check
  - 6acee3cc30 modules.random_org._query: only return text if present
  - 82f95429db modules.random_org unit tests: skip if random.org down
  - 1f9422e0cd utils.http.query: also except gaierror with tornado
- **ISSUE** #31499: (Reiner030) FeatureRequest: boto_elb misses connection_settings - idle_timeout (refs: #33936)
  - **PR** #33936: (rallytime) Add connecting_settings to boto_elb state attributes list
- **ISSUE** #29249: (timcharper) *salt-cloud sync_after_install: all* does not seem to sync anything at all (refs: #33917)
  - **PR** #33917: (techhat) Wait for up to a minute for sync_after_install
- **PR** #33877: (rallytime) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2016-06-09 14:50:42 UTC*
  - ef6da0be5d Merge pull request #33877 from rallytime/merge-2015.8
  - 398534a9e7 Fix ret return from merge-conflict resolution
  - b8e4706074 Merge branch '2015.5' into '2015.8'
  - cdda593c50 Merge pull request #33829 from terminalmage/update-versionchanged
    - f7028eb1c6 Update versionchanged directive
  - b8e6c144d8 Merge pull request #33833 from terminalmage/issue33645
    - 91745c2a67 Support syncing pillar modules to masterless minions
  - e061788e81 Merge pull request #33814 from terminalmage/archive-extracted-xz
    - 897a716df2 Support extraction of XZ archives in archive.extracted state
  - fa983e91cf Merge pull request #33778 from sodium-chloride/2015.5-2016-0604-1938
    - a5fb6d7a69 Fix minor docstring issues
  - b9133326c8 Merge pull request #33726 from jtand/sysmod_skip_valid_docs_glance

    - ebee8a89af glance.warn_until shouldn't be checked for a doc string
  - 137f0b19f3 Merge pull request #33611 from TargetHolding/2015.5
    - 1dd15a603b solve' TypeError: expected string or buffer' in json/decoder.py
    - eaf42ca892 solve AttributeError: 'module' object has no attribute 'exception'
- **ISSUE** #33810: (chiro79) locate.locate fails always (refs: #33827)
- **PR** #33827: (cachedout) Fix broken locate.locate function @ *2016-06-08 13:49:57 UTC*
  - ec09095c45 Merge pull request #33827 from cachedout/issue_33810
  - 9d36f1e474 Fix broken locate.locate function
- **PR** #33839: (cachedout) Fix another unit test stacktrace in pkg_resource @ *2016-06-08 13:32:55 UTC*
  - f7b3d0eda0 Merge pull request #33839 from cachedout/fix_pkgresource_test_stacktrace
  - 435547a747 Fix another unit test stacktrace in pkg_resource
- **PR** #33840: (cachedout) Remove matcher tests @ *2016-06-08 13:31:41 UTC*
  - 5f081ef31c Merge pull request #33840 from cachedout/remove_matcher_unit_tests
  - 6297448377 Remove matcher tests
- **PR** #33836: (cachedout) Fixing more stupid unit tests @ *2016-06-07 21:34:04 UTC*
  - cda032dab2 Merge pull request #33836 from cachedout/fix_winserver_manager_test
  - 453fb1ac91 Fixing more stupid unit tests
- **PR** #33805: (jfindlay) states.pkg int tests: skip if pkg mgr unavailable @ *2016-06-07 14:40:47 UTC*
  - 1db559afe9 Merge pull request #33805 from jfindlay/pkg_tests
  - 0c069ddc95 states.pkg int tests: skip if pkg mgr unavailable
- **PR** #33808: (jfindlay) fix some problems with the gem module integration tests @ *2016-06-07 14:40:25 UTC*
  - 3984b65486 Merge pull request #33808 from jfindlay/gem_tests
  - f7c19a1a58 modules.gem int tests: relax version checks
  - 6af47d2ba7 modules.gem int tests: remove pkgs before testing install
- **PR** #33770: (jfindlay) service state integration tests @ *2016-06-07 14:37:54 UTC*
  - c30d8a8c61 Merge pull request #33770 from jfindlay/service_tests
  - f13f914755 states.service: add integration tests
  - 90aee79c39 states.service.mod_watch: update unit test
  - d210a92f09 states.service.mod_watch: update sfun and force docs
- **PR** #33691: (justinta) Gem integration test @ *2016-06-06 11:13:23 UTC*
  - 7fdfbe9a28 Merge pull request #33691 from jtand/gem_integration_test
  - ff2dae103d ubuntu doesn't install default gems when ruby is installed
  - 504df9a65a Fixed lint error
  - 0cb1bfa0d3 Removed extra :
  - 86f59b3e80 Made more pythonic
  - 2f36f34981 Fixed salt.util import. Added status check to make sure external resource is available

  - 400a71ec33 Removed redundancies
  - 91db411bea A couple lint fixes
  - c97f3319b9 Add check for gem binary
  - 210aceb402 Refactored tests to not use return messages
  - 9d437bd45d Removed artifact from testing
  - 134e1fa888 Fixed typos, and added destructiveTest decorator
  - 37bc3ad8fd Fixed typo, uninstalled to uninstall
  - 5b23b91ac6 Integration test for gem module
- **PR** #33777: (sacren) Fix minor docstring issue of arg being missing @ *2016-06-06 10:44:59 UTC*
  - bb4194bb79 Merge pull request #33777 from sodium-chloride/2015.8-2016-0604-1939
  - c1fd830a1a Fix minor docstring issue of arg being missing
- **ISSUE** #31219: (gladiatr72) when the minions have all been destroyed… (refs: #33759)
- **PR** #33759: (cachedout) Catch no minions exception in batch mode @ *2016-06-03 21:22:49 UTC*
  - c749aea409 Merge pull request #33759 from cachedout/issue_31219
  - 15a39f8646 Catch no minions exception in batch mode
- **ISSUE** #33554: (jfindlay) local cache missing directories while running test suite (refs: #33653)
- **PR** #33719: (cachedout) Catch oserror for race condition @ *2016-06-03 17:25:26 UTC*
  - **PR** #33653: (cachedout) Create missing jid dir if it doesn't exist (refs: #33719)
  - 47d668e071 Merge pull request #33719 from cachedout/fixup_33653
  - 635efa248b Change to just surround the mkdir
  - 21b7123a60 Catch oserror for race condition
- **PR** #33712: (meaksh) Fix for groupadd execution module failures in SLES11 systems @ *2016-06-03 16:13:06 UTC*
  - 11e39e7203 Merge pull request #33712 from meaksh/fix-for-groupadd-module-failures-in-SLE11-2015.8
  - ab738416ba pylint fix
  - bf27e5d36e test_members cleanup
  - ba815dbf76 improvements on groupadd unit tests
  - 3bbc5ae0d9 one line is better
  - a53dc192c9 fix groupadd module for sles11 systems
- **PR** #33718: (rallytime) Back-port #33700 to 2015.8 @ *2016-06-03 16:10:44 UTC*
  - **PR** #33700: (sacren) Fix incorrect args passed to timezone.set_hwclock (refs: #33718)
  - 2c450a7494 Merge pull request #33718 from rallytime/bp-33700
  - a6a446121a Fix speed issue
  - a41146730a Fix incorrect args passed to timezone.set_hwclock
- **ISSUE** #33725: (terminalmage) git_pillar w/pygit2 fails to checkout a non-master branch when remote repo has no master branch (refs: #33727)
- **PR** #33727: (terminalmage) Fix git_pillar edge case for remote repos without a master branch @ *2016-06-03 16:03:59 UTC*

  - b07701f0a0 Merge pull request #33727 from terminalmage/issue33725
  - d8ba7ed5a5 Fix git_pillar edge case for remote repos without a master branch
- **PR** #33728: (jfindlay) Make *configurable_test_state* configurable in test mode @ *2016-06-03 16:02:57 UTC*
  - 015e50cec8 Merge pull request #33728 from jfindlay/test_state_test
  - 87e018af2a states.test.configurable_test_state: add unit tests
  - c2d0679c4b states.test.configurable_test_state: refactor change_data
  - f06ff1af1f states.test.configurable_test_state test mode
- **PR** #33729: (twangboy) Add exclude option to win_servermanager @ *2016-06-03 15:53:13 UTC*
  - 1cf8fe3f1d Merge pull request #33729 from twangboy/fix_win_servermanager
  - 2de91d166f Fix docstring
  - 9870479d99 Add exclude option to state
  - 50bd76e206 Add exclude option
- **ISSUE** #31816: (vutny) Deprecate or update the http://debian.saltstack.com/ (refs: #33743)
- **PR** #33743: (vutny) Debian installation docs: drop section about community-maintained repo @ *2016-06-03 15:29:45 UTC*
  - 6c150d840d Merge pull request #33743 from vutny/drop-debian-community-repo-doc
  - 8621f5be54 Debian installation docs: drop section about community-maintained repository
- **ISSUE** #33554: (jfindlay) local cache missing directories while running test suite (refs: #33653)
  - **PR** #33653: (cachedout) Create missing jid dir if it doesn't exist (refs: #33719)
- **PR** #33654: (twangboy) Fix win servermanager @ *2016-06-02 17:55:45 UTC*
  - 8a566ff4b9 Merge pull request #33654 from twangboy/fix_win_servermanager
  - 6c7b21676a Fix lint and tests
  - 4775e6bdf0 Add additional params to state
  - b0af32346d Add additional params to install and remove
- **ISSUE** #33424: (thusoy) Error logging with non-environment branches in gitfs (refs: #33679)
- **PR** #33679: (terminalmage) Only compile the template contents if they evaluate to True @ *2016-06-02 17:20:00 UTC*
  - 996ff56dd4 Merge pull request #33679 from terminalmage/issue33424
  - 9da40c4437 Append empty dictionaries for saltenvs with no top file
  - 5eb1b3ca62 Only compile the template contents if they evaluate to True
- **PR** #33685: (jfindlay) modules.cp.get_url: add test for https:// @ *2016-06-01 22:25:41 UTC*
  - c8dc70b96a Merge pull request #33685 from jfindlay/get_url_test
  - 2b5035fdc0 modules.cp.get_url: add test for https://
- **PR** #33581: (dincamihai) Call zypper refresh after adding/modifying a repository @ *2016-06-01 22:25:11 UTC*
  - 5e022ff29c Merge pull request #33581 from dincamihai/2015.8
  - 788730ea72 DRY test
  - 1d3769ccfa Improve zypper_patcher_config looks

  - 42d8d4195c Assert only gpgautoimport: True works
  - ced75e8e62 Reverse if conditions and rename variable
  - 80bfbe5c52 Reduce dicts and lists to one line where possible
  - 1d5d6d7d60 Update test method names to pass pylint
  - c7ae5907ee Call zypper refresh after adding/modifying a repository
- **ISSUE** #32916: (giannello) file.managed memory usage with s3 sources (refs: #33599)
- **PR** #33681: (rallytime) Back-port #33599 to 2015.8 (refs: #34208) @ *2016-06-01 21:14:29 UTC*
  - **PR** #33599: (lomeroe) Fix s3 large file download (refs: #33681)
  - 069ee15b7c Merge pull request #33681 from rallytime/bp-33599
  - 45143a599b use requests streaming for uploads/downloads to file (return_bin unchanged) allows downloading files larger than amount of memory (non-stream reads into memory before writing to disk or uploading)
  - 4a9b23f03f first go at having requests use streaming for get/put requests
- **ISSUE** #33393: (babilen) pip.installed does not work with ancient pip versions (refs: #33396)
- **PR** #33396: (babilen) Issue 33393 @ *2016-06-01 21:12:03 UTC*
  - 13537c4891 Merge pull request #33396 from babilen/issue-33393
  - 57e0475cd4 Make pip InstallationError import more robust
  - 291a3e21fa Remove duplicated code.
- **PR** #33652: (terminalmage) Lower the log level for failed auths @ *2016-06-01 16:37:09 UTC*
  - 7bce4ece1a Merge pull request #33652 from terminalmage/zh723
  - 411841603a Lower the log level for failed auths
- **ISSUE** #33582: (waxie) mysql module gives traceback if no working authentication (refs: #33615)
- **PR** #33615: (danslimmon) Fix crash on unconnectable MySQL server (resolves #33582) @ *2016-05-31 16:03:51 UTC*
  - 504989388a Merge pull request #33615 from danslimmon/mysql-traceback-33582
  - 180099ae9f Wrote test for broken server connection
  - c6c3ff02e3 Added some error checking to resolve #33582.
- **PR** #33558: (twangboy) Fix win servermanager @ *2016-05-27 22:05:43 UTC*
  - b47182e47c Merge pull request #33558 from twangboy/fix_win_servermanager
  - 62a6bde0ea Fix comment when already installed
  - 79bc7195dc Fix unit tests
  - 56a6f6bb83 Fix changes
  - 8ebe99ec5e Fix restart_needed
  - 6e478cbda0 Add restart needed
  - 72ebf26616 Add missing import
  - 193583be96 Use dictionary compare for changes in remove
  - 1ae7dd76c1 Use dictionary compare for changes

- **ISSUE** #33544: (tjuup) Salt 2016.3.0 (Boron) clean_old_jobs fails (refs: #33555)
- **PR** #33555: (cachedout) Fix crashing Maintenence process @ *2016-05-26 19:25:39 UTC*
  - 58d89d66e3 Merge pull request #33555 from cachedout/issue_33544
  - fe7ee7a470 Fix crashing Maintenence process
- **PR** #33501: (meaksh) unit tests for rpm.checksum() and zypper.download() @ *2016-05-26 14:34:27 UTC*
  - d052908729 Merge pull request #33501 from meaksh/zypper-download-check-signature-2015.8
  - eaaef25c79 lint issue fixed
  - 6b6febb211 unit tests for rpm.checksum() and zypper.download()
- **ISSUE** #33319: (ghost) Salt interpets jinja syntax in contents pillar (refs: #33513)
- **PR** #33513: (rallytime) Add a section to the jinja docs about escaping jinja @ *2016-05-26 14:24:58 UTC*
  - e2d0c4abb1 Merge pull request #33513 from rallytime/fix-33319
  - 81c1471209 Add a section to the jinja docs about escaping jinja
- **PR** #33520: (jacobhammons) Updated version numbers in the docs for the 2016.3.0 release @ *2016-05-26 14:15:00 UTC*
  - fabc15e616 Merge pull request #33520 from jacobhammons/release-notes.8
  - 42e358af7d Updated version numbers in the docs for the 2016.3.0 release
- **PR** #33507: (rallytime) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2016-05-25 19:14:41 UTC*
  - 5a6b037cbd Merge pull request #33507 from rallytime/merge-2015.8
  - 03b0c97520 Merge branch '2015.5' into '2015.8'
  - 6f7fda0354 Merge pull request #33486 from jtand/2015.5
    - d1e210fff8 Merge branch '2015.5' of https://github.com/saltstack/salt into 2015.5
    - ee2ae0ea8a Added docstring examples to glance.image_schema and schema_get
    - 59e90064e6 modules.swift.head does not have a body. Should not be checked for a docstring right now.
  - f72ec1479b Merge pull request #33482 from rallytime/pillar-opts-docs
    - 087564528d Add pillar_opts docs to master.rst
  - dc644b145d Merge pull request #33488 from rallytime/fix-18752
    - b0a9f4181f Add docs for the syndic_finger config
  - a4e84aa7d2 Merge pull request #33454 from scubahub/2015.5
    - df3c0b8e78 Correct (and make consistent) determination of the test flag.
  - 3a52ace673 manage account information for pam (#33473)
- **ISSUE** #15252: (gravyboat) Standalone minion docs don't explain what file is being modified. (refs: #33503)
- **PR** #33503: (rallytime) Add docs about minion config file in standalone minion docs @ *2016-05-25 17:23:08 UTC*
  - ee76be3b0b Merge pull request #33503 from rallytime/fix-15252
  - cfc07f7641 Add docs about minion config file in standalone minion docs
- **PR** #33474: (cachedout) Fix diskusage beacon (refs: #34103) @ *2016-05-25 17:10:54 UTC*
  - e9b648e461 Merge pull request #33474 from cachedout/issue_29451

  - aa2bac3a0d Remove debugging
  - 68d8050cb8 Fix diskusage beacon
- **PR** #33465: (meaksh) jobs.exit_success allow to check if a job has executed and exit successfully @ *2016-05-25 16:52:53 UTC*
  - 3bfb6bf719 Merge pull request #33465 from meaksh/check-if-job-returns-successfully-2015.8
  - 9deb70fd8e jobs.exit_success() now works parsing the results of jobs.lookup_id()
  - 7ba40c4f31 jobs.exit_success allow to check if a job has executed and exit successfully
  - **PR** saltstack/salt-jenkins#175: (justinta) Adding back shade to setup states (refs: #33487)
- **PR** #33487: (justinta) Add docstring examples to glance.py and nova.py [2015.8] @ *2016-05-25 16:47:25 UTC*
  - 70eb7b66f3 Merge pull request #33487 from jtand/glance_doc_fixes
  - 0b1cae05d9 Added docstring examples to glance methods and nova.list
  - ebf1256545 Don't need to check swift.head due to it having no body
- **ISSUE** #33423: (warden) etcd profile doesn't work when used in master conf file (refs: #33481)
- **PR** #33481: (rallytime) Fix docs about etcd config options and add pillar_opts doc (refs: #33482) @ *2016-05-25 16:41:56 UTC*
  - 56ea979916 Merge pull request #33481 from rallytime/fix-33423
  - 7fd3e8f361 Fix docs about etcd config options and add pillar_opts doc
- **ISSUE** #16319: (lsh-0) create a postgresql *query* function (refs: #33490)
- **PR** #33490: (rallytime) Document the postgres.psql_query function @ *2016-05-25 16:41:22 UTC*
  - 2394cdc4bf Merge pull request #33490 from rallytime/fix-16319
  - 0c5548f9d1 Document the postgres.psql_query function
- **PR** #33480: (jfindlay) states.service: minor doc updates @ *2016-05-25 16:38:14 UTC*
  - ede232f0f1 Merge pull request #33480 from jfindlay/service_doc
  - 29c00a1b1b states.service: clarify function description language
  - 6a9ae09e79 states.service.__virtual__: add load fail reason
  - **PR** #33483: (twangboy) Return full pending computer name (2015.8)
- **ISSUE** #32444: (justindesilets) Feature Request - jobs runner list by target (refs: #33491)
  - **PR** #33499: (cachedout) Use six.string_types in jobs runner
  - **PR** #33491: (BlaineAtAffirm) fix jobs.list_jobs failing with search_target (refs: #33499)
- **ISSUE** #32444: (justindesilets) Feature Request - jobs runner list by target (refs: #33491)
- **PR** #33491: (BlaineAtAffirm) fix jobs.list_jobs failing with search_target (refs: #33499) @ *2016-05-25 15:11:22 UTC*
  - 2e24a04565 Merge pull request #33491 from BlaineAtAffirm/2015.8
  - 7599b18995 fix jobs.list_jobs failing with search_target
- **ISSUE** #33467: (beelit94) Orchestration gives exception when a target does not exist (refs: #33478)
- **ISSUE** #32479: (ssgward) Orchestration gives exception when a target does not exist (refs: #32484, #33478)
- **PR** #33478: (rallytime) Back-port #32484 to 2015.8 @ *2016-05-24 19:14:23 UTC*

  - **PR** #32484: (cachedout) Only unsub if we have a jid (refs: #33478)
  - 1861af427e Merge pull request #33478 from rallytime/bp-32484
  - 042f17efa4 Only unsub if we have a jid
- **PR** #33457: (rallytime) Make doc formatting consistent and use correct versionadded @ *2016-05-24 17:52:34 UTC*
  - b8154b678e Merge pull request #33457 from rallytime/doc-formatting
  - 82f8f3efff Make doc formatting consistent and use correct versionadded
  - **PR** #33477: (terminalmage) Don't allow a "repo" kwarg for pkgrepo.managed
- **ISSUE** #29451: (githubcdr) 2015.8.3 pillar beacons bugged? (refs: #33476)
  - **PR** #33476: (cachedout) Allow for config entry to be a list in a dict for beacons
- **PR** #33469: (meaksh) check the RPM signature of zypper pkg.download packages and report errors @ *2016-05-24 16:09:05 UTC*
  - 9f56ab4c45 Merge pull request #33469 from meaksh/zypper-download-check-signature-2015.8
  - a65071a6d1 simpler rpm.checksum function
  - 80fe303e38 Renamed check_sig to checksum and some refactoring
  - d56e3f4258 bugfix: showing errors when a package download fails using zypper pkg.download
  - 8a21b9149e check the signature of downloaded RPM files
- **ISSUE** #33389: (DaveQB) Too many hostnames in pillar? (refs: #33459)
  - **PR** #33459: (rallytime) Add docs about PyYAML's 1024 character limitations for simple keys
  - **PR** #33464: (isbm) Prevent several minion processes on the same machine
  - **PR** #33432: (dincamihai) Make –gpg-auto-import-keys a global param when calling zypper
- **ISSUE** #32446: (sel-fish) " salt '*' saltutil.wheel minions.connected " not work (refs: #33414)
  - **PR** #33414: (rallytime) Fix the saltutil.wheel function and add integration tests
- **ISSUE** #29286: (harlanbarnes) Can't disable Job Cache? (refs: #33328)
  - **PR** #33440: (rallytime) Make sure the path we're removing is present first - avoid an OSError
  - **PR** #33328: (rallytime) Update job_cache and keep_jobs docs to be more specific to their behavior (refs: #33440)
- **ISSUE** #26913: (imchairmanm) manage.bootstrap runner quotation escape bug (refs: #33443)
  - **PR** #33443: (rallytime) Avoid a syntax error by using " instead of escaped '
  - **PR** #33436: (rmarcinik) Fix virtual function
- **PR** #33438: (rallytime) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2016-05-23 17:50:51 UTC*
  - 6e94a4a03b Merge pull request #33438 from rallytime/merge-2015.8
  - 7c41c34528 Merge branch '2015.5' into '2015.8'
    - 2cc650965a update 2015.5.11 release notes (#33412)
    - dc8ce2d8b1 Fix traceback in logging for config validation (#33386) (#33405)
- **ISSUE** #33395: (fmnisme) salt doc error (refs: #33421)
  - **PR** #33421: (abednarik) Documentation update in file.serialize.

  - **PR** #33398: (lvg01) Fix LVM parameter devices as a pure list. Comma seperated lists are c...
  - **PR** #33406: (rallytime) Back-port #33387 to 2015.8
  - **PR** #33387: (tveastman) Spelling correction. (refs: #33406)
- **ISSUE** #33298: (lorengordon) Windows: pkg.install returns failed for msiexec/instmsi exit code 3010 (ERROR_SUCCESS_REBOOT_REQUIRED) (refs: #33321)
  - **PR** #33321: (lorengordon) Update windows pkg.[install|remove] error logic
- **ISSUE** #29252: (mitar) reload_modules is not documented for the pkg state (refs: #33374)
  - **PR** #33374: (rallytime) Add note about reload_modules functionality for pkg.installed
- **ISSUE** #31430: (The-Loeki) Salt Coding Style regarding absolute_imports (refs: #33377)
  - **PR** #33377: (rallytime) Add note to absolute_imports practice about __future__ import
- **ISSUE** #21720: (kaithar) Revisiting aliases.file option. (refs: #33380)
  - **PR** #33380: (rallytime) Document how to set the alias file location for alias state
- **PR** #33403: (jacobhammons) 2015.8.10 release notes @ *2016-05-20 16:02:50 UTC*
  - 3c9def310c Merge pull request #33403 from jacobhammons/dot10
  - e850c298a9 2015.8.10 release notes
- **PR** #33381: (rallytime) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2016-05-20 15:58:11 UTC*
  - 91059224f6 Merge pull request #33381 from rallytime/merge-2015.8
  - 5aec32b20f Merge branch '2015.5' into '2015.8'
    - d15f5e2cef Merge pull request #33383 from thatch45/2015.5
      - f5ebcba21c restore whitespace
      - 1d8b289db1 blast, put the try/except int he right place
      - 081e6c5b83 maintain the fallabck because I am totally sick of this crap
    - 755acfb97e Improve doc clarity for disable_modules documentation (#33379)
  - 8ef7697806 Merge branch '2015.5' into '2015.8'
    - 2b5ad128bf Better YAML syntax error handling (#33375)
    - bb3e98cad2 Merge pull request #33372 from jacobhammons/release-update
      - 5ce502160b revved 2015.8 branch to .9 in version selector
  - **PR** #33386: (terminalmage) Fix traceback in logging for config validation (refs: #33405)
- **ISSUE** #27737: (mpaolini) name param never mentioned in pillar_ext git documentation (refs: #33369)
  - **PR** #33369: (rallytime) Add note about name parameter in git_pillar docs
- **ISSUE** #32913: (hrumph) Possible problem with salt.states.pkg.installed documentation (refs: #33362)
  - **PR** #33362: (rallytime) Add win_pkg to list of modules that support "version" in pkg.installed
- **ISSUE** #27779: (jbouse) [Doc] Hipchat returner documentation update (refs: #33365)
  - **PR** #33365: (rallytime) Add note to docs about api settings for Hipchat API v2
  - **PR** saltstack/salt-bootstrap#828: (vutny) Fix bootstrapping from git on Debian 8 by installing latest *tornado* via pip (refs: #32857)

  - **PR** #820: (dcolish) Refactor of cli parsers, normalize around conf_file (refs: #`saltstack/salt-bootstrap#828`_)
  - **PR** #32857: (vutny) Add initscripts, SystemD service units and environment files for Debian
- **PR** #33370: (jacobhammons) Update docs version to 2015.8.9 @ *2016-05-19 19:59:15 UTC*
  - 80f52a658e Merge pull request #33370 from jacobhammons/2015.8.9
  - 146b4df6be Updates docs version to 2015.8.9 Adds note regarding the os grain on Mint Linux Adds an FAQ regarding grains that change due to upstream changes
- **PR** #33366: (rallytime) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2016-05-19 19:41:40 UTC*
  - 3e5689abbf Merge pull request #33366 from rallytime/merge-2015.8
  - 52b3128678 Merge branch '2015.5' into '2015.8'
  - 55be0abf4d Expanded documentation for boto_elb state and module (#33341)
- **ISSUE** #33313: (morganwillcock) pkg.py: pkgs parameter documented as not supported on Windows (refs: #33361)
- **ISSUE** #3313: (mou) If no fileserver backend initialized there should be warning or error message on performing various file operations (refs: #33361)
  - **PR** #33361: (rallytime) Remove mentions of windows not supporting pkgs param
- **ISSUE** #29286: (harlanbarnes) Can't disable Job Cache? (refs: #33328)
  - **PR** #33328: (rallytime) Update job_cache and keep_jobs docs to be more specific to their behavior (refs: #33440)
- **ISSUE** #33295: (andrew-vant) Linux Mint service module not correctly detected. (refs: #33359)
  - **PR** #33359: (terminalmage) Properly detect newer Linux Mint distros
- **ISSUE** #32260: (jagguli) git.latest UnboundLocalError: local variable 'desired_upstream' referenced before assignmen (refs: #33340)
  - **PR** #33340: (terminalmage) Fix UnboundLocalError in git.latest
  - **PR** #33339: (phistrom) states.boto_elb Describe parameters in register_instances function
  - **PR** #33347: (rallytime) Fix some link errors in the test writing tutorial
  - **PR** #33312: (twangboy) Fix network.managed for windows
  - **PR** #33327: (cro) Bp 28467 calm mine
  - **PR** #28467: (jodv) Make mine.update more manageable for large environments (refs: #33327)
  - **PR** #33334: (jfindlay) import ps from psutil_compat in beacons
- **ISSUE** #21520: (jfindlay) sudo.salt_call is broken (refs: #25089)
  - **PR** #33318: (jfindlay) remove redundant, incorrect sudo_runas config documentation
  - **PR** #25089: (jfindlay) fix minion sudo (refs: #33318)
  - **PR** #22480: (thatch45) Add sudo user docs into salt (refs: #33318)
  - **PR** #20226: (thatch45) Allow sudo priv escalation (refs: #25089, #33318)
- **ISSUE** #33323: (terminalmage) Overeager globbing in systemd.py for sysv service detection (refs: #33324)
  - **PR** #33324: (terminalmage) Disambiguate non-exact matches when checking if sysv service is enabled
- **ISSUE** #30130: (dreampuf) Non-root minion not work with state.sls module (refs: #33325)

  - **PR** #33325: (cachedout) Allow concurrency mode in state runs if using sudo
- **ISSUE** #29674: (jakehilton) Salt Master Hang (refs: #33333)
  - **PR** #33333: (DmitryKuzmenko) Fix master hanging after a request from minion with removed key.
- **ISSUE** #33266: (Timandes) Method *grains.items* returns unexpected *manufacturer* information (refs: #33302)
  - **PR** #33306: (rallytime) Back-port #33302 to 2015.8
  - **PR** #33302: (The-Loeki) Cleanup comments in smbios.get output (fixes #33266) (refs: #33306)
- **ISSUE** #23643: (falzm) Error in iptables module: argument –match-set: expected 2 argument(s) (refs: #33314, #33301, #28325)
  - **PR** #33314: (gerhardqux) Fix iptables –match-set (#23643)
  - **PR** #33301: (gerhardqux) Fix iptables –match-set (#23643) (refs: #33314)
  - **PR** #28325: (l13t) Fix issue wiith –match-set option. #23643 (refs: #33314)
- **PR** #33308: (rallytime) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2016-05-17 19:26:05 UTC*
  - d0ed1616b0 Merge pull request #33308 from rallytime/merge-2015.8
  - 1c43a62f85 Merge branch '2015.5' into '2015.8'
    - 9b42a05519 Added some more docs for master and minion config settings (#33292)
  - 5004d2fa61 Merge branch '2015.5' into '2015.8'
  - 8acee5e06c Fix iptables –match-set (#23643) (#33301)
  - 757ef20a31 fix "loose" typo (#33290)
  - b7d98da64d Add auth_tries config option to minion.rst docs (#33287)
  - 061851bcbf Document minion_id_caching config value (#33282)

28.2.55 Salt 2015.8.12 Release Notes

Version 2015.8.12 is a bugfix release for *2015.8.0*.

Statistics

- Total Merges: **58**
- Total Issue References: **43**
- Total PR References: **117**
- Contributors: **29** (Azidburn, Ch3LL, UtahDave, bobrik, cachedout, cedwards, deepakhj, dere, gongled, gtmanfred, hrumph, hu-dabao, isbm, jacobhammons, jfindlay, jmesquita, junovitch, justinta, kev009, martinhoefling, multani, rallytime, randomed, sjorge, terminalmage, thatch45, theothergraham, twangboy, whiteinge)

Changelog for v2015.8.11..v2015.8.12

Generated at: 2018-05-28 01:19:12 UTC

- **PR** #35614: (rallytime) Update release notes for 2015.8.12
- **PR** #35611: (rallytime) Everything in the sample master config file should be commented out
- **ISSUE** #35384: (ghost) The unless requisite stops at first successful command (refs: #35569)

  - **PR** saltstack/salt#35545: (hu-dabao) fix-35384, fix cmd.run unless (refs: #35566)
- **PR** #35569: (rallytime) Write test for multiple unless commands where 1st cmd passes and 2nd fails @ *2016-08-19 19:28:01 UTC*
  - **PR** #35566: (rallytime) Back-port #35545 to 2015.8 (refs: #35569)
  - **PR** #35545: (hu-dabao) fix-35384, fix cmd.run unless (refs: #35569, #35566)
  - c9070c212f Merge pull request #35569 from rallytime/test-for-35384
  - 30f42d5352 Write test for multiple unless commands where 1st cmd passes and 2nd fails
  - **PR** #35600: (rallytime) Update release notes for 2015.8.12
  - **PR** #35599: (rallytime) Update release notes for 2015.8.12
  - **PR** #35584: (terminalmage) Update linux_sysctl tests to reflect new context key
  - **PR** #35575: (terminalmage) Add warning about AWS flagging of nmap usage
- **PR** #35577: (terminalmage) Unit file changes for 2015.8.12, 2016.3.3 @ *2016-08-18 20:36:25 UTC*
  - 26a7f7d9f7 Merge pull request #35577 from terminalmage/unit-file-changes
  - 6cb0fb47f3 pkg/salt-syndic.service: change Type to notify
  - 175ba99e0e pkg/salt-minion.service: remove KillMode, change Type to notify
  - 540ec28954 pkg/salt-master.service: remove KillMode
  - 69fad464ab pkg/salt-api.service: change Type to notify
  - **PR** saltstack/salt#35545: (hu-dabao) fix-35384, fix cmd.run unless (refs: #35566)
  - **PR** #35566: (rallytime) Back-port #35545 to 2015.8 (refs: #35569)
  - **PR** #35545: (hu-dabao) fix-35384, fix cmd.run unless (refs: #35569, #35566)
  - **PR** #35492: (terminalmage) Clarify config.get docstring
- **ISSUE** saltstack/salt#18419: (jasonrm) salt-cloud fails to run as non-root user (refs: #35483)
- **ISSUE** #34806: (jerrykan) salt-cloud ignores sock_dir when firing event (refs: #35483)
- **PR** #35483: (gtmanfred) use __utils__ in salt.cloud @ *2016-08-18 13:32:22 UTC*
  - 205d8e2e7b Merge pull request #35483 from gtmanfred/2015.8
  - 2d8ec1e9db use __opts__ in salt.utils.cloud for cache functions
- **PR** #35546: (whiteinge) Salt api eauth fail gracefully @ *2016-08-18 07:21:55 UTC*
  - 70fa2d0901 Merge pull request #35546 from whiteinge/salt-api-eauth-fail-gracefully
  - eb3574adae Don't fail hard if the user's permissions cannot be found
  - ec597bd54c Change groups check in token to look for truthy values
  - **PR** #35525: (UtahDave) add missing glob import
  - **PR** #35540: (rallytime) Whitespace fix for 2015.8
- **ISSUE** #33803: (dmurphy18) systemd notification is not fully supported by Salt (refs: #35510)
- **ISSUE** #33516: (Ch3LL) When upgrading from 2015.8.10 to 2016.3.0 on centos7/redhat7 I have to restart the salt-minion twice (refs: #35510)
- **PR** #35510: (terminalmage) Better systemd integration @ *2016-08-17 18:54:11 UTC*
  - fd3274c800 Merge pull request #35510 from terminalmage/issue33516

  - 5b5f19d269 Update zypper unit test to reflect call to config.get
  - 2730edb516 Add note about systemd-run usage in package states
  - e2d9e87e10 salt/modules/systemd.py: Use systemd-run –scope where needed
  - 22919a25bc Notify systemd on salt-api start
  - a40b3f8a08 Notify systemd on syndic start
  - e847d3af30 Notify systemd on minion start
  - d648887afc salt/modules/zypper.py: Use systemd-run –scope where needed
  - 2e17976722 salt/modules/yumpkg.py: Use systemd-run –scope where needed
  - 86b59c1e74 salt/modules/pacman.py: Use systemd-run –scope where needed
  - e32d92c6d5 salt/modules/ebuild.py: Use systemd-run –scope where needed
  - c7d21d3ae3 salt/modules/aptpkg.py: Use systemd-run –scope where needed
  - f83e0ef242 Add unit tests for salt.utils.systemd
  - 5b12f030c6 Add func to salt.utils.systemd to tell if scopes are available
  - **PR** #35513: (cachedout) Might be a good idea to be able to download the software we make
- **PR** #35302: (Ch3LL) Add job cache test @ *2016-08-17 10:45:28 UTC*
  - 9f87081cef Merge pull request #35302 from Ch3LL/add_job_cache_test
  - ccb2a5cadf remove unused imports
  - 512ae81dfd remove TMP and add integration.TMP
  - c9b7c3cf80 need to add returners option in other places
  - 7316df7a02 fix pylint
  - 50a4f0fe6a fix comment
  - 6837acf742 add job cache integration tests
- **PR** #35512: (cachedout) Fixup 35419 @ *2016-08-17 10:11:17 UTC*
  - 1c82c6bee5 Merge pull request #35512 from cachedout/fixup_35419
  - 253662541a Fix import
  - f16a30786b Fixes consul.agent_service_register which was broken for registering service checks.
- **PR** #35497: (deepakhj) Fixes spacing in requirements files @ *2016-08-17 09:34:15 UTC*
  - e1a373fa4c Merge pull request #35497 from deepakhj/2015.8
  - 685db4ab88 Fix spacing
- **PR** #35508: (terminalmage) Add Carbon to versionadded for git.diff @ *2016-08-17 06:17:12 UTC*
  - 4048255ed6 Merge pull request #35508 from terminalmage/update-docstring
  - 67c945fce0 Add Carbon to versionadded for git.diff
  - **PR** #35486: (rallytime) Update bootstrap script to latest stable (2016.08.16)
- **ISSUE** #35296: (szjur) cp.push_dir gets confused when using upload_path and is probably insecure too (refs: #35413)
- **PR** #35413: (cachedout) Resolve path issues with cp.push @ *2016-08-16 16:40:39 UTC*

  - 240fc12863 Merge pull request #35413 from cachedout/issue_35296
  - fb8a12d677 Fix silly error
  - 3646cf1afa Additional checks on master and integration test
  - 09efde7634 Splat the list into os.path.join
  - fc0d5878bc Set file_recv on test master
  - 81c4d136c5 Transition file push paths to lists
- **ISSUE** saltstack/salt#35380: (anlutro) salt-ssh with sudo stopped working (refs: #35476)
- **PR** #35476: (cachedout) Fixup SSH bug where sudo without sudo user would break @ *2016-08-16 15:41:25 UTC*
  - c3319b2a8b Merge pull request #35476 from cachedout/issue_35380
  - c05fcf33d1 Fixup SSH bug where sudo without sudo user would break
- **PR** #35471: (terminalmage) win_pkg: Fix traceback when package is not installed @ *2016-08-16 02:02:00 UTC*
  - 004778c966 Merge pull request #35471 from terminalmage/issue34479
  - e243c63e43 win_pkg: Fix traceback when package is not installed
- **PR** #35448: (isbm) Add ignore_repo_failure option to suppress zypper's exit code 106 on … @ *2016-08-16 01:39:43 UTC*
  - 5c9428c32d Merge pull request #35448 from isbm/isbm-zypper-106-fix
  - dd82e6a848 Add ignore_repo_failure option to suppress zypper's exit code 106 on unavailable repos
- **PR** #35451: (isbm) Bugfix: zypper mod repo unchanged @ *2016-08-16 01:38:25 UTC*
  - 1473474b04 Merge pull request #35451 from isbm/isbm-zypper-mod_repo-unchanged
  - 8790197d86 Fix Unit test for suppressing the exception removal on non-modified repos
  - 3f00c6997a Remove zypper's raise exception if mod_repo has no arguments and/or no changes
- **ISSUE** saltstack/salt#34279: (vmadura) Salt 2016.3.1 - Master Side Pillar Cache (backend: Disk) never Expires. (refs: #35453)
- **ISSUE** #34279: (vmadura) Salt 2016.3.1 - Master Side Pillar Cache (backend: Disk) never Expires. (refs: #35453)
- **PR** #35453: (theothergraham) fixes #34279 - disk cache ttl expiry @ *2016-08-16 01:34:33 UTC*
  - a8c4f17f50 Merge pull request #35453 from theothergraham/fix_CacheDisk
  - ae5b233d51 fixes #34279
- **PR** #35459: (thatch45) Ensure that output for salt-ssh gets back @ *2016-08-16 01:29:16 UTC*
  - d8c35b5260 Merge pull request #35459 from thatch45/shim_fix
  - 10037b00cb Some environments refuse to return the command output
  - **PR** #35460: (rallytime) [2015.8] Update bootstrap script to latest stable (2016.08.15)
- **ISSUE** saltstack/salt#35010: (vchav73) cp.push_dir returns incorrect result for non-existent directories (refs: #35442)
  - **PR** #35442: (cachedout) Fix cp.push_dir pushing empty dirs
- **ISSUE** saltstack/salt#35387: (mzealey) Document reload_grains and reload_pillar (refs: #35436)
  - **PR** #35436: (cachedout) Minor doc fixup
- **ISSUE** saltstack/salt#35121: (sjorge) file.append always results in change (refs: #35132)

- **PR** #35132: (sjorge) fixes , causing lots of mayham (onchange) with 2016.3.2 for me @ *2016-08-15 07:11:22 UTC*
    - a0b128a85a Merge pull request #35132 from sjorge/2015.8-35121
    - 5cb38c8ae0 switch to fpread().splitlines(), as per @lorengordon suggestion
    - 634f1dded5 fixes #35121, causing lots of mayham (onchange) with 2016.3.2 for me
    - **PR** saltstack/salt#34573: (cedwards) Update freebsd.rst (refs: #35394)
    - **PR** #35394: (rallytime) Back-port #34573 to 2015.8
    - **PR** #34573: (cedwards) Update freebsd.rst (refs: #35394)
    - **PR** #35359: (terminalmage) Clean up open filehandles
- **PR** #35339: (isbm) Bugfix: Prevent continuous restart, if a dependency wasn't installed @ *2016-08-11 16:15:17 UTC*
    - 9ea7a34c30 Merge pull request #35339 from isbm/isbm-2015.8-minion-importerror-fix
    - 12af60b7be Fix continuous minion restart if a dependency wasn't installed
- **PR** #35357: (twangboy) Fix file.recurse with clean: True on Windows (2015.8) @ *2016-08-11 00:44:14 UTC*
    - fd9b05ace4 Merge pull request #35357 from twangboy/file.recurse.clean.2015.8
    - d328ec0157 Fix file.recurse with clean: True
- **PR** #35323: (thatch45) Fix issue with bad error check in salt-vt @ *2016-08-10 11:33:49 UTC*
    - 4618b433e9 Merge pull request #35323 from thatch45/ssh_crazy
    - 8a5b47b5d7 Collect all error data from the wfuncs call
    - 11864c31b7 supress a stack trace to show clean ssh error
    - 9fbfa282fa wow this solves an issue!
- **PR** #35325: (kev009) Fix freebsd netstat route on fbsd 10+ @ *2016-08-10 11:33:12 UTC*
    - cfae862972 Merge pull request #35325 from kev009/fbsd-netstat-route
    - 0d49dd3c29 Fix fbsd netstat route on fbsd 10+
- **ISSUE** #35264: (bobrik) ssh_known_hosts.present is not idempotent in test=true with port (refs: #35301)
    - **PR** #35301: (bobrik) Pass port to ssh.check_known_host, closes #35264
- **ISSUE** #34945: (babilen) file.recurse breaks directory permissions (refs: #35309)
    - **PR** #35309: (terminalmage) file.recurse: Do not convert octal mode string to int
- **ISSUE** #35051: (terminalmage) Runner/Wheel funcs still print return data to console when invoked from orchestration (refs: #35290)
- **PR** #35290: (terminalmage) Resolve a couple bugs in orchestration output @ *2016-08-09 15:27:00 UTC*
    - 2efc1b333b Merge pull request #35290 from terminalmage/issue35051
    - d621aa7b61 Update runner/wheel unit tests to reflect new key in ret dict
    - 90c12a9c7b Add __orchestration__ key to orch returns for runner/wheel funcs
    - 7b8c3b86e7 Suppress error about invalid changes data for orchestration jobs
    - 54a1704d6c Suppress event for wheel/runner funcs executed from orchestration
    - f409f62bf2 Accept print_event option in WheelClient.cmd()
    - b42b25ccce Add cmd func for RunnerClient

  - 480065fe00 Add print_event option to client mixins
- **ISSUE** #31074: (turtletraction) salt-ssh sudo_user execution not running as sudo_user (refs: #35211)
- **PR** #35211: (cachedout) Alternative sudo users for salt-ssh @ *2016-08-08 15:40:55 UTC*
  - f8158124d5 Merge pull request #35211 from cachedout/issue_31074
  - 6f53232e6d Better error handling and a workaround for group mismatch.
  - 5b56a4acf7 Docs
  - ae04e7aaeb Initial POC
- **ISSUE** #35166: (bobrik) state_output_profile defaults are confusing (refs: #35271)
- **PR** #35271: (bobrik) Default state_output_profile to True everywhere, closes #35166 @ *2016-08-08 14:36:24 UTC*
  - 3e4eb13daa Merge pull request #35271 from bobrik/default-output-profile
  - 6cdee21036 Default state_output_profile to True everywhere, closes #35166
- **ISSUE** #32719: (azweb76) Salt-Call Hangs when IPv6 is disabled on System (refs: #35233)
- **PR** #35233: (terminalmage) Do not attempt to get fqdn_ip{4,6} grains when ipv{4,6} grains are empty @ *2016-08-06 22:58:32 UTC*
  - 673e1aa1aa Merge pull request #35233 from terminalmage/issue32719
  - 730a077041 Do not attempt to get fqdn_ip{4,6} grains when ipv{4,6} grains are empty
- **PR** #35202: (multani) doc: fix broken links in the test documentation page @ *2016-08-06 08:29:41 UTC*
  - cdf3c0fe73 Merge pull request #35202 from multani/fix/test-doc
  - 1642dba5d1 doc: fix broken links in the test documentation page
- **ISSUE** saltstack/salt#34861: (dere) minion incorrectly reports package cannot be installed (refs: #35119)
- **PR** #35236: (rallytime) Back-port #35119 to 2015.8 @ *2016-08-06 08:10:54 UTC*
  - **PR** #35119: (dere) Assume two EVRs are equal if E and V are equal but one R is missing. (refs: #35236)
  - e1331cd2a3 Merge pull request #35236 from rallytime/bp-35119
  - 9ade78de7b Revise unnecessary code duplication
  - 7c15f5b20a Fix formatting
  - 64f93f8938 Assume two EVRs are equal if E and V are equal but one R is missing.
- **ISSUE** saltstack/salt#29785: (paul-mulvihill) pkg.installed to accept 'latest' as a version keyword (refs: #35225)
- **ISSUE** #29785: (paul-mulvihill) pkg.installed to accept 'latest' as a version keyword (refs: #35240)
- **PR** #35240: (dere) Backport #35225 to 2015.8 @ *2016-08-06 07:54:19 UTC*
  - **PR** #35225: (dere) Add missing documentation for pkg.installed (refs: #35240)
  - 4f2b8aa5b6 Merge pull request #35240 from derekmaciel/bp-35225
  - 9ed47f713a Add missing documentation for pkg.installed
- **PR** #35241: (terminalmage) Ensure max recursion in gitfs results in no blob object being returned. @ *2016-08-06 07:53:49 UTC*
  - 4bcfaa97d0 Merge pull request #35241 from terminalmage/gitfs-fixes
  - e05648cc2d Break from loop when file is found
  - 6764a88601 Ensure that failed recursion results in no blob object being returned

  - **PR** saltstack/salt#35039: (whiteinge) Add saltenv support to module.run (refs: #35245)
- **PR** #35245: (rallytime) Back-port #35039 to 2015.8 @ *2016-08-06 07:52:44 UTC*
  - **PR** #35039: (whiteinge) Add saltenv support to module.run (refs: #35245)
  - f6d7360e0b Merge pull request #35245 from rallytime/bp-35039
  - 51ab9cd6d4 Add saltenv support to module.run
- **ISSUE** #35214: (tdenny) git.latest fails on non-fast-forward when a fast-forward is possible (refs: #35249)
- **PR** #35249: (terminalmage) Fix regression in git.latest @ *2016-08-06 07:52:15 UTC*
  - d65a5c7134 Merge pull request #35249 from terminalmage/issue35214
  - bcd5129e9f Fix regression in git.latest when update is fast-forward
  - e2e8bbbfde Add integration test for #35214
- **ISSUE** saltstack/salt#34691: (dmacvicar) beacons.list does not include beacons configured from the pillar/ext_pillar (refs: #saltstack/salt`#34827`_, #34827)
  - **PR** saltstack/salt#35146: (cachedout) Don't discard running beacons config when listing becaons (refs: #35174)
  - **PR** saltstack/salt#34827: (thatch45) fix beacon list to include all beacons being processed (refs: #35146, #`saltstack/salt`#35146`_`_)
  - **PR** #35174: (rallytime) Back-port #35146 to 2015.8
  - **PR** #35146: (cachedout) Don't discard running beacons config when listing becaons (refs: #35174)
  - **PR** saltstack/salt#35135: (rallytime) Add missing CLI Examples to aws_sqs module funcs (refs: #35173)
  - **PR** #35173: (rallytime) Back-port #35135 to 2015.8
  - **PR** #35135: (rallytime) Add missing CLI Examples to aws_sqs module funcs (refs: #35173)
  - **PR** #35145: (jacobhammons) doc version update to 2015.8.11, updates to release notes
- **PR** #35114: (terminalmage) Add clarification docs on a common git_pillar misconfiguration @ *2016-08-02 00:30:48 UTC*
  - 81845ee31d Merge pull request #35114 from terminalmage/git_pillar-env-remap-docs
  - 5951554e9f Add clarification docs on a common git_pillar misconfiguration
- **ISSUE** saltstack/salt#34767: (hrumph) Ensure that pkg.installed function refreshes properly with windows. (refs: #34768)
- **ISSUE** #34767: (hrumph) Ensure that pkg.installed function refreshes properly with windows. (refs: #34768)
- **PR** #34768: (hrumph) Fixes #34767 @ *2016-08-01 21:46:16 UTC*
  - 88a9fb1b31 Merge pull request #34768 from hrumph/bad-installed-state
  - e1fcb8311d Put pkg.latest_version in try/except structure Move refreshed or refresh to different spot (just for code tidyness)
  - e0b6261659 changed name of varibale 'refreshed' to 'was_refreshed'
  - 340110b4b4 Move check for rtag to outermost-nesting in function
  - ac67c6b493 Lint fix
  - 0435a1375e Get rid of repetition in code by using new "refreshed" variable instead
  - 3b1dc978e2 Lint fix

  - a9bd1b92b9 lint fixes
  - 71d69343ef Fixes #34767
- **PR** #35043: (rallytime) Start release notes file for 2015.8.12 @ *2016-08-01 17:22:04 UTC*
  - 343576408f Merge pull request #35043 from rallytime/new-release-notes
  - bdcc81a384 Start release notes file for 2015.8.12
- **PR** #35050: (terminalmage) [orchestration] Properly handle runner/wheel funcs which accept a 'saltdev' argument @ *2016-08-01 15:48:08 UTC*
  - 848bf0272f Merge pull request #35050 from terminalmage/fix-saltdev-arg
  - 40cfa7cf17 Avoid needlessly running 2 argspecs in salt.utils.format_call()
  - fd186b7e4c Pass environment as 'saltdev' if runner/wheel func accepts a saltdev argument
  - 951b52ab93 Pass __env__ from saltmod orch states to to saltutil.{runner,wheel}
- **PR** #35066: (jfindlay) returners.postgres_local_cache: do not log in __virtual__ @ *2016-07-30 01:32:17 UTC*
  - 2144178ae0 Merge pull request #35066 from jfindlay/postgres_log
  - c2c442234f returners.postgres_local_cache: do not log in __virtual__
- **ISSUE** #34927: (bobrik) Salt does not run "systemd daemon-reload" on unit override (refs: #35024)
- **PR** #35024: (bobrik) Cache systemd unit update check per unit, closes #34927 @ *2016-07-28 17:56:29 UTC*
  - 7121618142 Merge pull request #35024 from bobrik/daemon-reload-fix
  - c300615e9d Cache systemd unit update check per unit, closes #34927
  - **PR** #35026: (cachedout) Expressly deny a minion if a key cannot be found
  - **PR** saltstack/salt#33875: (jmesquita) Fix naive fileserver map diff algorithm (refs: #35000)
- **PR** #35000: (rallytime) Back-port #33875 and #34999 to 2015.8 @ *2016-07-27 21:55:58 UTC*
  - **PR** #34999: (cachedout) Fixup #33875 (refs: #35000)
  - **PR** #33875: (jmesquita) Fix naive fileserver map diff algorithm (refs: #35000, #34999)
  - 2b511f3013 Merge pull request #35000 from rallytime/bp-33875
  - 35696ad637 Pylint fix
  - f9fd6ddd8a Fixup #33875
  - 56b1f6c651 Fix naive fileserver map diff algorithm
- **ISSUE** saltstack/salt#34526: (danielmotaleite) salt-ssh + mine = weird error (refs: #34835, #'saltstack/salt'#34835'_'_)
  - **PR** saltstack/salt#34835: (thatch45) Make the mine and publish combine minion and master opts in salt-ssh (refs: #34994)
- **PR** #34994: (rallytime) Back-port #34835 to 2015.8 @ *2016-07-27 18:21:10 UTC*
  - **PR** #34835: (thatch45) Make the mine and publish combine minion and master opts in salt-ssh (refs: #34994)
  - 837bc6ba7d Merge pull request #34994 from rallytime/bp-34835
  - 9268a793de same thing for the mine in salt-ssh
  - 3e11e19714 Fix the mine in salt ssh

- **PR** #34991: (cachedout) SSH timeout @ *2016-07-27 17:24:38 UTC*
  - b58c663d8d Merge pull request #34991 from cachedout/ssh_timeout
  - 39cd8da399 Lint diff against salt-testing
  - 443e5cdde2 Add timeout to ssh tests
  - **PR** #34976: (cachedout) Refine errors in client
- **ISSUE** #34509: (srkunze) No atomic thin.tgz deploy (refs: #34831)
- **PR** #34831: (thatch45) If the thin does not match, then redeploy, don't error @ *2016-07-26 22:27:01 UTC*
  - a83cdf9339 Merge pull request #34831 from thatch45/recoverssh
  - fa73041a49 If the thin does not match, then redeploy, don't error
  - **PR** #34916: (cachedout) Master performance improvement
- **PR** #34911: (cachedout) Backport #34906 @ *2016-07-22 23:23:24 UTC*
  - **PR** #34906: (cachedout) Set timeout for run_salt in test suite (refs: #34911)
  - 34dc2fd792 Merge pull request #34911 from cachedout/backport_34906
  - 8becec2f4f Backport #34906
- **ISSUE** saltstack/salt#33620: (TheBigBear) [2016.3.0] win_pkg: pkg.list_upgrades loops (almost) endlessly - cmds take VERY long (refs: #34898)
- **PR** #34898: (hrumph) Stop multiple refreshes during call to pkg.list_upgrades @ *2016-07-22 22:28:42 UTC*
  - 6ccc27f697 Merge pull request #34898 from hrumph/list_upgrades_refresh
  - acd4b1a23b Fixes #33620
- **PR** #34606: (isbm) Bugfix: Exit on configuration read (backport) (refs: #34751) @ *2016-07-22 17:35:18 UTC*
  - 5c13ee0e72 Merge pull request #34606 from isbm/isbm-config-reading-exit-2015.8
  - 5f5b802c0c Add option to master config reader on ignoring system exit for wrong configuration
  - 6fc677f177 Ignore minion config errors everywhere but the minion itself
  - 8699194647 Remove deprecation: BaseException.message deprecated as of 2.6
  - 0e65cfec91 Fix lint: E8302
  - 67faa56bf1 Use Salt default exit codes instead of hard-coded values
  - a84556e596 Exit immediately on configuration error
  - 43d965907c Raise an exception on any found wrong configuration file
  - 30ed728d05 Cover exception handling in the utils.parsers
  - 5e8c0c6bdb Introduce configuration error exception
- **ISSUE** saltstack/salt#27783: (anlutro) salt-ssh not properly updating file_lists, causing file.recurse to fail (refs: #34862)
  - **PR** #34862: (thatch45) Fix salt-ssh cacheing issue
- **ISSUE** #34725: (akoumjian) *git.latest* with *force_reset* set to *True* does not reset local changes, causing it to fail. (refs: #34869)
  - **PR** #34869: (terminalmage) Fail git.latest states with uncommitted changes when force_reset=False
- **PR** #34859: (cachedout) Fix wheel test @ *2016-07-21 19:55:25 UTC*

  - 4f4381e5b9 Merge pull request #34859 from cachedout/fix_wheel_test
  - b4be66dedf Fix wheel test
- **ISSUE** saltstack/salt#34798: (Ch3LL) exception when running state.low over salt-ssh (refs: #34822)
- **ISSUE** saltstack/salt#34796: (Ch3LL) exception when running state.high over salt-ssh (refs: #34822)
- **PR** #34822: (thatch45) Fix salt-ssh state.high and state.low @ *2016-07-21 19:16:19 UTC*
  - acc9e31c02 Merge pull request #34822 from thatch45/ssh_fixes
  - b5de492143 fix #34798
  - 5ad6bd7307 fix #34796
- **PR** #34847: (cachedout) Add an option to skip the verification of client_acl users @ *2016-07-21 17:55:55 UTC*
  - 5d91139bc9 Merge pull request #34847 from cachedout/pwall
  - 2c8298dc6e Profile logging
  - 3affafa2e9 Add an option to skip the verification of client_acl users
- **ISSUE** saltstack/salt#34691: (dmacvicar) beacons.list does not include beacons configured from the pillar/ext_pillar (refs: #saltstack/salt‘#34827‘_, #34827)
- **PR** #34827: (thatch45) fix beacon list to include all beacons being processed @ *2016-07-21 14:49:56 UTC*
  - 07d1d36653 Merge pull request #34827 from thatch45/34691
  - 1ccf35eca4 fix beacon list to include all beacons being processed
  - **PR** saltstack/salt#28521: (gongled) SPM: packaging doesn’t work in Python 2.6. Fixed. (refs: #34833)
- **PR** #34833: (rallytime) Back-port #28521 to 2015.8 @ *2016-07-21 14:37:24 UTC*
  - **PR** #28521: (gongled) SPM: packaging doesn’t work in Python 2.6. Fixed. (refs: #34833)
  - b375720251 Merge pull request #34833 from rallytime/bp-28521
  - e50a6783ce SPM: packaging doesn’t work in Python 2.6. Fixed.
- **ISSUE** #25213: (aboe76) Add spm man page to setup.py (refs: #saltstack/salt‘#25276‘_, #25276)
  - **PR** saltstack/salt#25276: (jacobhammons) copy spm.1 man page during setup (refs: #34823)
- **PR** #34823: (rallytime) Back-port #25276 to 2015.8 @ *2016-07-20 20:56:04 UTC*
  - **PR** #25276: (jacobhammons) copy spm.1 man page during setup (refs: #34823)
  - 042646582f Merge pull request #34823 from rallytime/bp-25276
  - a028796eff copy spm.1 man page during setup Refs #25213
- **ISSUE** saltstack/salt#34648: (bortels) Error that % cannot start token (refs: #34828)
- **ISSUE** #34648: (bortels) Error that % cannot start token (refs: #34828)
  - **PR** #34828: (thatch45) Fix #34648
  - **PR** saltstack/salt#34642: (justinta) Check that mysqladmin exists before running mysql integration tests (refs: #34818)
- **PR** #34818: (justinta) Skip mysql state test if mysqladmin is not available @ *2016-07-20 16:10:35 UTC*
  - 98fa4a404e Merge pull request #34818 from jtand/mysql_state_integration_test_cleanup
  - 9abb6f91bb Skip mysql state test if mysqladmin is not available
- **ISSUE** saltstack/salt#26278: (jiahua-h) ”order: first” doesn’t work? (refs: #34803)

- **ISSUE** saltstack/salt#24744: (anlutro) Allow states to define order: first (refs: #34803)
- **ISSUE** #24744: (anlutro) Allow states to define order: first (refs: #34803)
- **PR** #34803: (junovitch) salt/state.py: set 'chunk['order'] = 0' with 'order: first'; fixes #24744 @ *2016-07-20 13:56:20 UTC*
  - 6636f2b449 Merge pull request #34803 from junovitch/issue_24744
  - 64c850410f salt/state.py: set 'chunk['order'] = 0' with 'order: first'; fixes #24744
- **PR** #34773: (randomed) Bugfix: Startup states on minions are not being written to mysql returner @ *2016-07-19 12:39:53 UTC*
  - 58021035a9 Merge pull request #34773 from randomed/mysql-returner-startup/2015.8
  - 0cd55eb7d7 Add jid=req handling for mysql returner. It should also store the return jid into the jid list table.
  - **PR** #34751: (cachedout) Remove unnedeed config test
  - **PR** #34606: (isbm) Bugfix: Exit on configuration read (backport) (refs: #34751)
- **PR** #34754: (cachedout) Disable test @ *2016-07-18 18:40:50 UTC*
  - f19caac8e4 Merge pull request #34754 from cachedout/disable_mid_test
  - 46901c6e65 Disable test
- **ISSUE** saltstack/salt#34678: (martinhoefling) config.get module is broken due to bug in dictupdate.py (refs: #34726, #'saltstack/salt'#34726'_'_, #34741)
  - **PR** saltstack/salt#34726: (martinhoefling) Always loop over updated keys in non recursive update (refs: #34741)
- **PR** #34741: (rallytime) Back-port #34726 to 2015.8 @ *2016-07-18 18:00:23 UTC*
  - **PR** #34726: (martinhoefling) Always loop over updated keys in non recursive update (refs: #34741)
  - 81f29006f2 Merge pull request #34741 from rallytime/bp-34726
  - d949110993 Loop over updated keys in non recursive update
- **ISSUE** saltstack/salt#34703: (Cashwini) Is it possible to return output from python execution module to a file on salt master? (refs: #34721)
- **PR** #34721: (rallytime) Add output_file option to master config docs @ *2016-07-16 20:04:03 UTC*
  - e9e5bbe38b Merge pull request #34721 from rallytime/fix-34703
  - 9c803d05a5 Add output_file option to master config docs
- **ISSUE** saltstack/salt#32276: (javicacheiro) pkg.installed using sources from master fails with file not found after first succesful run (refs: #34689)
- **PR** #34689: (Azidburn) fix second run problems with pkg.installed using sources @ *2016-07-15 21:19:39 UTC*
  - 08d00f3a61 Merge pull request #34689 from Azidburn/fix_pkg_sources
  - 2c0fc919b3 fix second run problems with pkg.installed using sources
- **PR** #34695: (isbm) Bugfix: Zypper *pkg.list_products* returns False on some empty values (2015.8) @ *2016-07-15 21:08:00 UTC*
  - 4cb1ded520 Merge pull request #34695 from isbm/isbm-zypper-product-boolean-values
  - 5ed5142fbc Update test data for 'registerrelease' and 'productline' fields
  - 21444ee240 Bugfix: return boolean only for 'isbase' and 'installed' attributes

28.2.56 Salt 2015.8.13 Release Notes

Version 2015.8.13 is a bugfix release for *2015.8.0*.

Security Fixes

CVE-2017-5192 local_batch client external authentication not respected

The `LocalClient.cmd_batch()` method client does not accept `external_auth` credentials and so access to it from salt-api has been removed for now. This vulnerability allows code execution for already-authenticated users and is only in effect when running salt-api as the `root` user.

CVE-2017-5200 Salt-api allows arbitrary command execution on a salt-master via Salt's ssh_client

Users of Salt-API and salt-ssh could execute a command on the salt master via a hole when both systems were enabled.

We recommend everyone on the 2015.8 branch upgrade to a patched release as soon as possible.

28.2.57 Salt 2015.8.2 Release Notes

Version 2015.8.2 is a bugfix release for *2015.8.0*.

Statistics

- Total Merges: **379**
- Total Issue References: **138**
- Total PR References: **351**
- Contributors: **83** (DmitryKuzmenko, JaseFace, LoveIsGrief, MasterNayru, Oro, SmithSamuelM, The-Loeki, TheBigBear, aboe76, ajacoutot, anlutro, avinassh, basepi, bdrung, bechtoldt, bernieke, blueyed, cachedout, cbuechler, cedwards, clarkperkins, cro, dkiser, douglas-vaz, dr4Ke, eguven, eliasp, erchn, eyj, favadi, flavio, garethgreenaway, gravyboat, gtmanfred, hedinfaok, hexedpackets, hyn-salt, isbm, itsamenathan, jacksontj, jacobhammons, jeffreyctang, jejenone, jfindlay, johnsocp, justinta, keesbos, lathama, ldobson, lomeroe, martinhoefling, mbarrien, mbologna, merll, mrosedale, msteed, multani, nasenbaer13, nmadhok, notpeter, opdude, papertigers, pass-by-value, plastikos, quantonganh, rallytime, redmcg, rowillia, ruzarowski, ryan-lane, s0undt3ch, sdm24, sjansen, skizunov, srkunze, techhat, terminalmage, ticosax, tkwilliams, toddtomkinson, twangboy, twellspring, whiteinge)

Important: A significant orchestrate issue (issue ##29110) was discovered during the release process of 2015.8.2, so it has not been officially released. Please use *2015.8.3* instead.

Changelog for v2015.8.1..v2015.8.2

Generated at: 2018-05-27 23:17:44 UTC

- **PR** #28865: (jfindlay) add 2015.8.2 release notes @ *2015-11-13 17:30:18 UTC*
  - af297bb0ae Merge pull request #28865 from jfindlay/2015.8
  - 1f847fc9ba add 2015.8.2 release notes
- **ISSUE** #27392: (ahammond) schedule running state.orchestrate fails (refs: #28730)

- **PR** #28730: (garethgreenaway) Fixes to how return_job is handled in the scheduler for the salt master. @ *2015-11-13 16:58:20 UTC*
  - 15672a3faa Merge pull request #28730 from garethgreenaway/27392_2015_8_scheduler_return_job_master
  - 882350a543 Fixing the salt scheduler so that it only attempts to return the job data to the master if the scheduled job is running from a minion's scheduler.
- **PR** #28848: (cro) Lint @ *2015-11-13 13:46:36 UTC*
  - 5560cb662b Merge pull request #28848 from cro/fx2_multi_creds
  - f032bffd7c Lint
  - 6bb6703c3e Merge branch 'fx2_multi_creds' of git://github.com/cro/salt into cro
  - 3b7d22248c Fix fallback credentials, add grains based on dracr.server_info and dracr.inventory, fix short-circuited for loop that was preventing retrieval of most data from CMC and DRAC devices, format responses from racadm more clearly.
  - b86c614564 Better logic around fallback credentials.
  - 2701826a99 Update fx2.py, fix typos in new fallback parameters.
  - 8ce5348808 Better variable name.
  - 92038b8718 Default configuration file for proxy minions.
- **PR** #28842: (cachedout) Add transport setting to shell test @ *2015-11-12 21:43:11 UTC*
  - 778ace3ca5 Merge pull request #28842 from cachedout/tcp_shell_test
  - 785bf94f55 Add transport setting to shell test
- **PR** #28837: (basepi) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2015-11-12 21:17:14 UTC*
  - 5639971744 Merge pull request #28837 from basepi/merge-forward-2015.8
  - 1c91ad6765 fix lint
  - 4b706ac76a Merge remote-tracking branch 'upstream/2015.5' into merge-forward-2015.8
    - eb904665dc Merge pull request #28832 from basepi/backport.28826
      - 57be72eb91 Add backports_abc and singledispatch_helpers to thin as well
      - 897cad627b Add singledispatch to the thin
    - eff811a0ad Merge pull request #28833 from basepi/increase.gather_job_timeout.8647
      - c09243dd01 Increase the default gather_job_timeout
    - e4a036365d Merge pull request #28829 from basepi/merge-forward-2015.5
      - f8b8441485 Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.5
      - 76e69b4bff Merge pull request #28777 from rallytime/bp-28740-2014.7
      - da5fac2b36 Back-port #28740 to 2014.7
      - 45c73ebf2f Merge pull request #28716 from rallytime/bp-28705
      - 32e7bd3ea0 Account for new headers class in tornado 4.3
      - f4fe921965 Merge pull request #28717 from cachedout/umask_note
      - 1874300e08 Add note about recommended umask
  - 5aeab71f76 Merge remote-tracking branch 'upstream/2015.5' into merge-forward-2015.8

        - 93562631aa Merge pull request #28756 from MrCitron/fix-25775
            - 82075c809c Add logs and correct pylint error
            - e31e22d96a Fix 25775
        - 30cc48e37f Merge pull request #28786 from chrigl/fix-28783
            - ba6d814553 closes #28783
        - 8f1d0b636e Merge pull request #28776 from rallytime/bp-28740-2015.5
            - 49256b7d90 Back-port #28740 to 2015.5
        - 77d4b980f1 Merge pull request #28760 from dmyerscough/28732-Fix-cherrypi-api-keys-endpoint
            - 206d1684b2 Fixing CherryPy key bug
        - 6f8f04975f Merge pull request #28746 from rallytime/bp-28718
            - 092f441cad Account for no POST data
- **ISSUE** #28549: (ldelossa) dockerng module issue (refs: #28827)
- **PR** #28827: (jacksontj) Cleanup virtual_timer in loader @ *2015-11-12 19:39:29 UTC*
    - c4fb185147 Merge pull request #28827 from jacksontj/2015.8
    - f49502fd48 __*modules*__ isn't a global, although __*salt*__ is
    - c734cb8876 Fix virtual_timer branch such that it will catch exceptions.
- **PR** #28836: (cachedout) Cast to dict to fix wheel tests in tcp @ *2015-11-12 19:22:44 UTC*
    - 21520c6c1d Merge pull request #28836 from cachedout/fix_tcp_wheel_tests
    - 8d3244166b Cast to dict to fix wheel tests in tcp
- **PR** #28834: (cachedout) Fix breakage in tcp server @ *2015-11-12 18:57:18 UTC*
    - 560671a170 Merge pull request #28834 from cachedout/tcp_revert_master_uri
    - 755d493bed Fix breakage in tcp server
- **PR** #28804: (cachedout) TCP test fixes @ *2015-11-12 18:39:25 UTC*
    - 224602437a Merge pull request #28804 from cachedout/tcp_test_fixes
    - f799971280 Change logic
    - 52ed06500a Fix typo
    - 9b18f372e6 Normalize IPC check among transports
    - e8ead2bfed Allow for tcp transport in publish
    - e33b903e7b Allow for tcp transport in mine
    - 3d80e67a2d Allow for tcp transport in auth
- **ISSUE** #28828: (basepi) salt-ssh doesn't package tornado's new deps in the thin (refs: #28826)
- **PR** #28826: (basepi) [2015.8] Add new tornado deps to salt-ssh thin (refs: #28832) @ *2015-11-12 18:14:43 UTC*
    - 49992070db Merge pull request #28826 from basepi/salt-ssh.singledispatch.thin
    - 1e1a74fd61 Add backports_abc and singledispatch_helpers to thin as well
    - da1a2773dd Add singledispatch to the thin
- **PR** #28759: (jfindlay) simplify stdin use of stdin in at.present state @ *2015-11-12 18:11:55 UTC*

  - **PR** #28187: (sjansen) fix at.present (refs: #28759)
  - af52c3272f Merge pull request #28759 from jfindlay/at
  - 987d1fee7c simplify stdin use of stdin in at.present state
- **PR** #28824: (rallytime) Back-port #28778 and #28820 to 2015.8 @ *2015-11-12 18:06:31 UTC*
  - **PR** #28820: (cro) Add versionadded directives for chronos and marathon proxy grains. (refs: #28824)
  - **PR** #28778: (toddtomkinson) marathon and chronos proxy minions (refs: #28824)
  - 08891cb210 Merge pull request #28824 from rallytime/bp-28778-and-28820
  - ab5943995b Change versionaddeds to 2015.8.2 from Boron
  - da7ad0df99 Add versionadded directives.
  - 4bdd10fdf5 documentation updates
  - 675bc2acce more pylint fixes
  - 8e19b5c518 pylint fixes
  - ba94878f45 marathon and chronos proxy minions
- **ISSUE** #23271: (twisty7867) Unicode paths break file states with masterless minion on Vagrant/Ubuntu 14.04 (refs: #28803)
- **PR** #28803: (jfindlay) decode strings to utf-8 @ *2015-11-12 04:59:38 UTC*
  - 30ea94439c Merge pull request #28803 from jfindlay/sdecodes
  - 11163380cf sdecode chunk name in state compiler
  - 7f95c483e1 sdecode strings in file state
  - fe4d08526d sdecode strings in highstate outputter
- **ISSUE** #25363: (syphernl) rabbitmq_{user|vhost}.present in test=True reports unnecessary changes (refs: #28269)
- **ISSUE** #24856: (pruiz) rabbitmq_user state incorrectly reports result=True when using test=true (refs: #28269)
- **PR** #28782: (rallytime) Fixes to rabbitmq user state @ *2015-11-12 00:59:57 UTC*
  - **PR** #28269: (rallytime) Refactor rabbitmq_user state to use test=True correctly (refs: #28782, #28772)
  - 59b505ff7c Merge pull request #28782 from rallytime/rabbitmq-user-state
  - e2b0fee57e Don't change perms list, only existing perms should be a dictionary.
  - 7601647d69 Revert "rabbitmq.list_user_permissions returns a dict, not a list. Don't expect a list."
- **ISSUE** #28429: (cbuechler) salt-cloud VMware driver fails with uncustomizable guest when not customizing guest (refs: #28789)
- **PR** #28789: (nmadhok) Provide ability to enable/disable customization for newly create VMs using VMware salt-cloud driver @ *2015-11-11 22:48:57 UTC*
  - 098d48ad26 Merge pull request #28789 from nmadhok/2015.8-customization-fix
  - 9294ebd984 Provide ability to enable/disable customization for new VMs. Fixes #28429
- **ISSUE** #28692: (mrosedale) puppet.run fails with arguments (refs: #28768)
- **PR** #28768: (mrosedale) 2015.8 @ *2015-11-11 19:29:11 UTC*
  - 1e510be55b Merge pull request #28768 from mrosedale/2015.8

  - fbbbdcc02e Update puppet.py
  - 1c1a4b4410 Update puppet.py
  - 59bd6aef5c Merge pull request #1 from mrosedale/mrosedale-patch-1
    - c26ea916aa Update puppet.py

- **ISSUE** #25363: (symphernl) rabbitmq_{user|vhost}.present in test=True reports unnecessary changes (refs: #28269)
- **ISSUE** #24856: (pruiz) rabbitmq_user state incorrectly reports result=True when using test=true (refs: #28269)
- **PR** #28772: (rallytime) rabbitmq.list_user_permissions returns a dict, not a list. Don't expect a list. @ *2015-11-11 18:17:09 UTC*
  - **PR** #28269: (rallytime) Refactor rabbitmq_user state to use test=True correctly (refs: #28782, #28772)
  - a6cad46301 Merge pull request #28772 from rallytime/rabbitmq-user-state
  - 07482211eb rabbitmq.list_user_permissions returns a dict, not a list. Don't expect a list.
- **ISSUE** #28724: (quantonganh) Exception occurred when calling boto_vpc.route_table_present with test=True (refs: #28725)
- **PR** #28774: (rallytime) Back-port #28725 to 2015.8 @ *2015-11-11 18:16:27 UTC*
  - **PR** #28725: (quantonganh) boto_vpc: return an empty dict in case cannot get the route tables (refs: #28774)
  - d570ac48f4 Merge pull request #28774 from rallytime/bp-28725
  - c3420461c3 boto_vpc: return an empty dict in case cannot get the route tables
- **PR** #28775: (rallytime) Back-port #28740 to 2015.8 @ *2015-11-11 17:57:24 UTC*
  - **PR** #28740: (MasterNayru) Add missing S3 module import (refs: #28777, #28775, #28776)
  - 806d1b3669 Merge pull request #28775 from rallytime/bp-28740
  - 8a2780da18 Add missing S3 module import
- **PR** #28755: (rallytime) Move most vmware driver list_* functions to use salt.utils.vmware functions @ *2015-11-11 17:49:16 UTC*
  - f273c46f07 Merge pull request #28755 from rallytime/vmware-utils
  - 5abe010023 Move most vmware driver list_* functions to use salt.utils.vmware functions
- **ISSUE** #28655: (sjorge) possible issue with state module boto_cfn/docker/… (refs: #28744)
- **PR** #28744: (jfindlay) import gate elementtree @ *2015-11-11 16:29:12 UTC*
  - 0d912bf0d4 Merge pull request #28744 from jfindlay/elementttree
  - e321d60002 import gate elementtree in artifactory module
  - f20f3f697b import gate elementtree in boto_iam state
  - 9845d2f2c6 import gate elementtree in boto_cfn state
- **ISSUE** #28726: (feigenblatt) user.present ignores "createhome: False" (refs: #28758)
- **PR** #28758: (jfindlay) remove redundant logic in useradd execution module @ *2015-11-11 16:22:21 UTC*
  - b65e786351 Merge pull request #28758 from jfindlay/user
  - dbd582cd8d fix doc formatting in user.present state
  - 3824d2e9fc only change/report new home when createhome is True

  - 3fbf81611f remove redundant logic in useradd execution module
- **PR** #28757: (mbarrien) Bug fix: pip command to not quote spaces in cmd line args @ *2015-11-11 16:08:46 UTC*
  - 6eced26013 Merge pull request #28757 from mbarrien/fix-pip-cmd
  - 6df6cb82a6 Fix pip command to not quote spaces in cmd line args
- **PR** #28764: (multani) Various documentation fixes @ *2015-11-11 16:06:10 UTC*
  - 356bf2987d Merge pull request #28764 from multani/fix/docs
  - 1a31b69763 doc: fix documentation formatting in salt.utils.jinja
  - 59c105b4b9 doc: fix documentation formatting in salt.states.boto_iam*
  - cbb167c8ee doc: fix documentation formatting in in salt.modules.lxc
  - cb03a89e52 doc: fix documentation formatting in salt.modules.aptpkg
- **PR** #28752: (aboe76) Update openSUSE grain for tumbleweed @ *2015-11-11 03:54:37 UTC*
  - d77c24e70d Merge pull request #28752 from aboe76/suse_tumbleweed_grain
  - 764cb16ef0 Update openSUSE grain for tumbleweed
- **ISSUE** #28712: (hexedpackets) Service registration in the Consul module is broken (refs: #28713)
- **PR** #28713: (hexedpackets) Rename consul.list to consul.list_keys. @ *2015-11-11 00:57:23 UTC*
  - a620bc5596 Merge pull request #28713 from hexedpackets/fix-consul-module
  - 0889907b3c Make consul.list a function alias.
- **PR** #28719: (jacobhammons) removed dependencies info from docs @ *2015-11-10 00:04:53 UTC*
  - decc31a766 Merge pull request #28719 from jacobhammons/spm
  - d7017be031 removed dependencies info from docs
- **PR** #28709: (basepi) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2015-11-09 23:38:27 UTC*
  - 989069f44a Merge pull request #28709 from basepi/merge-forward-2015.8
  - 2d04ddc108 Merge remote-tracking branch 'upstream/2015.5' into merge-forward-2015.8
    - f40c617bad Merge pull request #28705 from cachedout/tornado_http_headers
      - 7ac6cde1ee Account for new headers class in tornado 4.3
  - c90431eddc Rip out unit test that doesn't apply anymore
  - aeeaa7c90d Merge remote-tracking branch 'upstream/2015.5' into merge-forward-2015.8
    - 604a7b4199 Merge pull request #28699 from rallytime/bp-28670
      - e436b23296 psutil can fail to look-up a uid and raise a KeyError
    - 7bd3eb8370 Merge pull request #28703 from rallytime/bp-28690
      - a0988dab58 Fix 28689 : Check s3 ext pillar cache file before calculating expiration
    - 2a40f57b93 Merge pull request #28694 from s0undt3ch/2015.5
      - 0910c6ffe4 Update to latest bootstrap script v2015.11.09
    - 3249b322e8 Merge pull request #28669 from rallytime/fix-26592
      - 098fb815af Use the -q argument to strip extraneous messages from rabbitmq
    - 29e8250d0c Merge pull request #28645 from jacksontj/2015.5

      - f63c2d70a7 Rework minion return_retry_timer
    - 1bbaea8aad Merge pull request #28668 from twangboy/fix_15177
      - 745b8f75f6 Fixed some lint
      - a43eb53f28 Added version added notes in docs
      - 6b537c8640 Fixed join_domain and unjoin_domain for Windows
    - 4ad5056066 Merge pull request #28666 from jfindlay/r_data
      - 29228f445f define r_data before using it in file module
    - e129e889ad Merge pull request #28662 from cachedout/issue_24758
      - 78f4894333 Add note about disabling master_alive_interval
    - df121d0cec Merge pull request #28627 from twangboy/backport_win_useradd
      - 87282b6354 Backport win_useradd

- **ISSUE** #28469: (mlalpho) state boto_secgroup.present fails to find vpc_name (refs: #28710, #28534)
- **PR** #28710: (rallytime) Pass kwargs correctly to _get_group from get_group_id @ *2015-11-09 22:29:09 UTC*
  - 8d5ab15c16 Merge pull request #28710 from rallytime/fix-28469
  - 0571608f5d Pass kwargs correctly to _get_group from get_group_id
- **PR** #28698: (rallytime) Back-port #28530 to 2015.8 @ *2015-11-09 18:11:51 UTC*
  - **PR** #28530: (skizunov) AsyncTCPReqChannel will fail after 10 uses (refs: #28614, #28698)
  - cfa0cec19c Merge pull request #28698 from rallytime/bp-28530
  - d94d0db805 AsyncTCPReqChannel will fail after 10 uses
- **ISSUE** #28678: (johnsocp) Error in netapi/rest_tornado preventing it from starting (refs: #28679)
- **PR** #28700: (rallytime) Back-port #28679 to 2015.8 @ *2015-11-09 18:07:44 UTC*
  - **PR** #28679: (johnsocp) Adding err variable definition to fix error that is preventing rest_tornado from initializing (refs: #28700)
  - 2fe9e2e7c5 Merge pull request #28700 from rallytime/bp-28679
  - 4e0870e636 Adding variable definition for issue #28678
  - **PR** saltstack/salt-bootstrap#868: (cachedout) Always refresh the Arch Linux keyring if needed (refs: #28695, #28694)
- **PR** #28695: (s0undt3ch) [2015.8] Update to latest bootstrap script v2015.11.09 @ *2015-11-09 17:50:15 UTC*
  - 8ccea2a855 Merge pull request #28695 from s0undt3ch/2015.8
  - bb6c60a330 Update to latest bootstrap script v2015.11.09
- **ISSUE** #28526: (clarkperkins) yumpkg.installed broken in salt v2015.8.1 on CentOS 6 minions (refs: #28656)
- **PR** #28656: (clarkperkins) #28526 fixed yumpkg module issue with pkg.installed @ *2015-11-09 05:16:00 UTC*
  - 61ba00b1c3 Merge pull request #28656 from clarkperkins/bugfix/fix-yumpkg-module
  - e11f87be93 #28526 fixed yumpkg module
- **ISSUE** #28588: (aboe76) openSUSE Leap not recognized as 'Suse' os grain and os_family grain (2015.8.1) (refs: #28672)
- **PR** #28672: (jfindlay) add OS grain support for SuSE Leap @ *2015-11-08 01:05:51 UTC*

  - 54484e4e29 Merge pull request #28672 from jfindlay/suse_grain
  - b44ba6fa9c add OS grain support for SuSE Leap
- **ISSUE** #28603: (alexharrington) MooseFS/LizardFS mount options force remount (refs: #28673)
- **PR** #28673: (jfindlay) add hidden_opts to mount.mounted @ *2015-11-08 00:51:19 UTC*
  - 476f55ebc0 Merge pull request #28673 from jfindlay/mount_hide
  - 1dcaa8e1d7 add hidden_opts to mount.mounted
  - d3aff8f6b8 minor refactor of mount state
- **PR** #28667: (cro) saltutil.sync_all should sync proxymodules as well as the rest. @ *2015-11-07 01:09:28 UTC*
  - 24d75709fa Merge pull request #28667 from cro/proxy_sync_all
  - 08e53b317f Sync proxymodules with sync_all
- **PR** #28665: (jfindlay) fixes to windows execution and state modules @ *2015-11-07 00:47:38 UTC*
  - 019c13948a Merge pull request #28665 from jfindlay/win_fixorz
  - e8c7371b56 fix minor doc issues in win_system module
  - 5828f391b9 handle error on nonexistent net dev in win_network
  - d1560f9ea9 check for wua time setting as a str
- **ISSUE** #28542: (Ch3LL) s3.get execution module returns error (refs: #28660)
- **PR** #28660: (techhat) Don't sign empty regions @ *2015-11-06 20:49:25 UTC*
  - ce3ce7ddf2 Merge pull request #28660 from techhat/emptyregion
  - a52518494a Don't sign empty regions
- **PR** #28632: (terminalmage) Fixes/improvements to pkgbuild state/modules @ *2015-11-06 20:48:07 UTC*
  - 0583575f82 Merge pull request #28632 from terminalmage/pkgbuild-fixes
  - 59f31b4dca Initialize logging in pkgbuild state
  - af0b2c4a33 Fix false-positives for pkgbuild.built state
  - d83e779eac rpmbuild: Change return data to include a list of packages built
  - 03d9321379 debbuild: Change return data to include a list of packages built
- **ISSUE** #28591: (ssgward) SPM package install error (refs: #28658)
- **PR** #28658: (techhat) Remove _pkgdb_fun() references @ *2015-11-06 20:25:59 UTC*
  - b82abadd9b Merge pull request #28658 from techhat/issue28591
  - 4f2b175467 Remove _pkgdb_fun() references
- **ISSUE** #28470: (mlalpho) salt boto_rds.present fails to execute, too many arguments (refs: #28612, #28653)
- **PR** #28653: (rallytime) Provide possible parameters for boto_rds.present engine values @ *2015-11-06 18:58:35 UTC*
  - e59d160120 Merge pull request #28653 from rallytime/boto_rds_engine_docs
  - 7b30d7e002 Provide possible parameters for boto_rds.present engine values
- **PR** #28649: (bdrung) Fix OS related grains on Debian @ *2015-11-06 18:25:46 UTC*
  - 911761d8bc Merge pull request #28649 from bdrung/2015.8

  - 92a17d4cae Fix OS related grains on Debian
- **ISSUE** #26889: (UtahDave) salt-call w/non root user outputs repeating error (refs: #28113, #27343)
- **PR** #28646: (rallytime) Back-port #28614 to 2015.8 @ *2015-11-06 18:19:08 UTC*
  - **PR** #28614: (skizunov) Fixed memory leak in AsyncTCPReqChannel (refs: #28646)
  - **PR** #28530: (skizunov) AsyncTCPReqChannel will fail after 10 uses (refs: #28614, #28698)
  - **PR** #28113: (skizunov) 'RuntimeError: IOLoop is closing' thrown in Minion on TCP transport (refs: #28614)
  - **PR** #27343: (cachedout) Close io loop before deleting attribute (refs: #28614)
  - 7531bc7334 Merge pull request #28646 from rallytime/bp-28614
  - 034cf28e57 Fixed memory leak in AsyncTCPReqChannel
- **PR** #28647: (rallytime) Back-port #28624 to 2015.8 @ *2015-11-06 18:18:32 UTC*
  - **PR** #28624: (hyn-salt) Added reasoning why boto_cloudwatch.py cannot be loaded. (refs: #28647)
  - a829120746 Merge pull request #28647 from rallytime/bp-28624
  - 3b59cfae5f Added reasoning why boto_cloudwatch.py cannot be loaded.
- **PR** #28648: (rallytime) Merge branch '2015.5' into '2015.8' @ *2015-11-06 17:46:59 UTC*
  - 52d70c986d Merge pull request #28648 from rallytime/merge-2015.8
  - 81c4974fde Merge branch '2015.5' into '2015.8'
    - 64a20228c6 Merge pull request #28617 from cachedout/umask_module_sync
      - 227792e158 Set restrictive umask on module sync
    - 065f8c7fb3 Merge pull request #28622 from gravyboat/update_puppet_module_docs
      - 4ea28bed30 Update puppet module wording
- **PR** #28638: (anlutro) Salt-SSH: Return more concise error when SSH command fails @ *2015-11-06 16:54:46 UTC*
  - 4722e41787 Merge pull request #28638 from alprs/saltssh-handle_ssh_errors
  - 5419b98363 return concise error when ssh fails
- **PR** #28644: (pass-by-value) Make sure versionchanged is correct @ *2015-11-06 16:53:31 UTC*
  - e72e60d4b4 Merge pull request #28644 from pass-by-value/update_versionchanged
  - f4c297e794 Make sure versionchanged is correct
- **ISSUE** #8: (thatch45) Network persistence (refs: #28615)
- **ISSUE** #64: (thatch45) State file rendering system (refs: #28615)
- **ISSUE** #54: (thatch45) Release items (refs: #28615)
- **PR** #28615: (The-Loeki) Fixes to FreeBSD pkg @ *2015-11-05 23:43:33 UTC*
  - **PR** #198: (techhat) Basic salt support for Tomcat (refs: #28615)
  - cf79722260 Merge pull request #28615 from The-Loeki/patch-1
  - a9ee178e0d rehash is a shell builtin, needs cmd.shell to work
  - 17f3852bdd environ.get has no output_loglevel
- **PR** #28613: (cachedout) Add facility to deepcopy bound methods in Py2.6 and apply to grains @ *2015-11-05 23:28:50 UTC*

  - **PR** #28587: (cachedout) Reset yaml rendering hooks to avoid leaks (refs: #28613)
  - 9196c57e3f Merge pull request #28613 from cachedout/py26_method_deepcopy
  - 0935fcf4fc Spelling is hard
  - 2435b45195 Move to compat module to avoid namespace collisions in salt.utils
  - f519661875 Add facility to deepcopy bound methods in Py2.6 and apply to grains
- **ISSUE** #28527: (Oro) boto_rds.create needs storage_type, which does not exist in boto.rds2 create_db_instance (refs: #28561)
- **ISSUE** #28470: (mlalpho) salt boto_rds.present fails to execute, too many arguments (refs: #28612, #28653)
- **PR** #28612: (rallytime) Remove unsupported storage_type argument for parity with boto_rds module @ *2015-11-05 19:07:42 UTC*
  - **PR** #28561: (Oro) Issue #28527 boto_rds.create does not work (refs: #28612)
  - 2032d61e68 Merge pull request #28612 from rallytime/fix-28470
  - 8fd26a5488 Remove unsupported storage_type argument for parity with boto_rds module
- **PR** #28611: (rallytime) [2015.8] Be explicit about salt.utils.vmware function calls @ *2015-11-05 18:43:36 UTC*
  - d81330ac7f Merge pull request #28611 from rallytime/vmware-utils-fix
  - f46547eb56 [2015.8] Be explicit about salt.utils.vmware function calls and avoid namespacing
- **PR** #28610: (pass-by-value) Lxc config additions @ *2015-11-05 18:43:05 UTC*
  - 35dbca24e7 Merge pull request #28610 from pass-by-value/lxc_config_additions
  - 83193641ca Add doc about cloud lxc options
  - 8977ddad59 Add argument to init
  - 2be3f8b5bb Add bootstrap delay and systemd check options
- **ISSUE** #28601: (nasenbaer13) boto_asg.present overwrites custom dimensions in alarms (refs: #28602)
- **PR** #28602: (nasenbaer13) Allow setting of custom dimensions in asg alarm specification @ *2015-11-05 15:00:24 UTC*
  - 464aa6b062 Merge pull request #28602 from eyj/fix-28601
  - 963ad4250a Allow setting of custom dimensions in asg alarm specification
- **PR** #28596: (rallytime) Merge branch '2015.5' into '2015.8' @ *2015-11-05 14:25:09 UTC*
  - 572d95b3e1 Merge pull request #28596 from rallytime/merge-2015.8
  - eec9d69387 Merge branch '2015.5' into '2015.8'
    - 08295de5a5 Merge pull request #28563 from s0undt3ch/2015.5
      - 16f4db79a0 Update to latest bootstrap script v2015.11.04
    - 1e09f186ce Merge pull request #28541 from twangboy/fix_28173
      - 7edf5ce370 Fixed problem with system.set_computer_name
    - f44ed780b5 Merge pull request #28537 from jfindlay/decode_state_2015.5
      - 06e514940c decode filename to utf-8 in file.recurse state
    - 6acf87593f Merge pull request #28529 from rallytime/fix-28272
      - a959681858 Add link to Sending a GH PR to documentation docs

      - 1c612e2772 Update contributing and documentation pages to recommend submitting against branches
    - 025bff2bf0 Merge pull request #28548 from nmadhok/2015.5-task-error
      - 804a0a6537 Tasks can be in queued state instead of running. Fixes #28511
    - 63bd3e52b3 Merge pull request #28531 from rallytime/fix-24585
      - bc577b2531 Add versionadded directives to virtualenv_mod state/module
    - ea3bf972c4 Merge pull request #28508 from twangboy/fix_unit_tests_windows
      - 0da6ff7c50 Fixed some logic
      - cf1e059be5 Fixed windows tests
    - 73c5735fc1 Merge pull request #28525 from rallytime/route53_spacing
      - 6ab2ce615c Fix spacing in doc examples for boto_route53 state and module
    - 2d7f934f67 Merge pull request #28517 from rallytime/fix-28243
      - be8f650901 Punctuation.
      - fd846822c1 Add state_auto_order defaults to True note to ordering docs
- **PR** #28593: (blueyed) doc: fix typo with salt.states.file: s/preseve/preserve/ @ *2015-11-04 22:33:25 UTC*
  - 73c33e0b4a Merge pull request #28593 from blueyed/fix-typo-preserve
  - eaf27d6ee7 doc: fix typo with salt.states.file: s/preseve/preserve/
- **PR** #28578: (twangboy) Fixed the script… something got broke… @ *2015-11-04 22:00:18 UTC*
  - 8b483ee354 Merge pull request #28578 from twangboy/fix_windows_installer_script
  - 90b19a3279 Fixed the script… something got broke…
- **PR** #28579: (jfindlay) fix __virtual__ returns: tls,uptime mods @ *2015-11-04 22:00:02 UTC*
  - 7ca7ed4b37 Merge pull request #28579 from jfindlay/virt_ret
  - 333c132378 fix __virtual__ returns: tls,uptime mods
- **ISSUE** #27574: (jgill) salt-cloud: Could not associate elastic ip address <None> with network interface <eni-xxxxxxxx> (refs: #28584)
- **PR** #28584: (rallytime) If AssociatePublicIpAddress is set to True, don't auto-assign eip. @ *2015-11-04 21:59:38 UTC*
  - **PR** #25315: (ruzarowski) [cloud:EC2] Move handling of AssociatePublicIpAddress to associate_eip/allocate_new_eip logic depending on value type (refs: #28584)
  - ae764c6b5c Merge pull request #28584 from rallytime/fix-27574
  - 490e1bd5bb If AssociatePublicIpAddress is set to True, don't auto-assign eip.
- **ISSUE** #28392: (jacksontj) AsyncZeroMQReqChannel does not implement *tries* (2015.8) (refs: #28410)
- **PR** #28576: (jacksontj) Only encode the zmq message once @ *2015-11-04 21:59:20 UTC*
  - **PR** #28410: (jacksontj) Add retries to the zeromq.AsyncReqMessageClient (refs: #28576)
  - 231cdd4316 Merge pull request #28576 from jacksontj/transport
  - b29fc676a3 Only encode the zmq message once
- **PR** #28587: (cachedout) Reset yaml rendering hooks to avoid leaks (refs: #28613) @ *2015-11-04 21:37:11 UTC*

  - ab62f5cd12 Merge pull request #28587 from cachedout/fix_yaml_render_leak
  - 2da64bd736 Reset yaml rendering hooks to avoid leaks
- **ISSUE** #3436: (madduck) Pillar does not handle Unicode data (refs: #28134, #saltstack/salt‘#28134‘_)
  - **PR** saltstack/salt#28134: (bernieke) fix unicode pillar values #3436 (refs: #28581)
- **PR** #28581: (basepi) Revert b4875e585a165482c4c1ddc8987d76b0a71ef1b0 @ *2015-11-04 19:28:20 UTC*
  - 69081d00e0 Merge pull request #28581 from saltstack/revert-28134-2015.8
  - 0a07c90d5e Revert b4875e585a165482c4c1ddc8987d76b0a71ef1b0
- **ISSUE** #28477: (anlutro) KeyError with file.managed HTTPS source (refs: #28573)
- **PR** #28573: (jacksontj) Add *body* to salt.utils.http.query returns @ *2015-11-04 17:18:19 UTC*
  - ea3658eac8 Merge pull request #28573 from jacksontj/2015.8
  - d55ea7550b Add *body* to salt.utils.http.query returns
- **ISSUE** #655: (thatch45) Add general command management to service (refs: #‘saltstack/salt-bootstrap#656‘_)
  - **PR** saltstack/salt-bootstrap#674: (jfindlay) add support for repo.saltstack.com (refs: #28564, #28563)
  - **PR** saltstack/salt-bootstrap#665: (mbologna) Change to ’dnf’ as package manager for Fedora 22-> (refs: #28564, #28563)
  - **PR** saltstack/salt-bootstrap#656: (eyj) Add bootstrap -b flag (don’t install dependencies) (refs: #28564, #28563)
  - **PR** saltstack/salt-bootstrap#654: (hedinfaok) Fixes error finding python-jinja2 in RHEL 7 (refs: #28564, #28563)
  - **PR** saltstack/salt-bootstrap#653: (cbuechler) Make bootstrap work with FreeBSD 11-CURRENT. (refs: #28564, #28563)
- **PR** #28564: (s0undt3ch) [2015.8] Update to latest bootstrap script v2015.11.04 @ *2015-11-04 15:29:46 UTC*
  - 3a729c2b40 Merge pull request #28564 from s0undt3ch/2015.8
  - b6a53a6bfb Update to latest bootstrap script v2015.11.04
- **ISSUE** #28527: (Oro) boto_rds.create needs storage_type, which does not exist in boto.rds2 create_db_instance (refs: #28561)
- **PR** #28561: (Oro) Issue #28527 boto_rds.create does not work (refs: #28612) @ *2015-11-04 15:13:09 UTC*
  - fed4c6f482 Merge pull request #28561 from Oro/fix-boto-rds-create
  - 54782b6fd9 Removed exception message where there is no exception
  - e08f45c824 Issue #28527 boto_rds.create does not work
- **PR** #28560: (bdrung) Fix various typos @ *2015-11-04 15:06:36 UTC*
  - ec924e8410 Merge pull request #28560 from bdrung/2015.8
  - 89dcb66310 Fix the wrong ”allow to do” phrase
  - 859b6b46a6 Fix typo an nonexistant -> nonexistent
  - 66921cc61e Fix typo an succesfully -> successfully
  - c1e3ef7c8d Fix typo an explicitely -> explicitly
  - 029a95398c Fix typo an superflous -> superfluous
  - 026c215933 Fix typo an unecessary -> unnecessary

  - 5f7fc5f94b Fix typo an edditable -> editable
  - 0b768944c2 Fix typo an deamon -> daemon
  - 5af49881d7 Fix typo an completly -> completely
  - 14d2a16f74 Fix typos of compatibility
  - 46a5a9b073 Fix typo an suppored -> supported
  - abc490a78e Fix typo an usefull -> useful
  - ddd412180c Fix typo an targetting -> targeting
  - 610a6a77ae Fix typo an verison -> version
  - e0a5d46a1e Fix typo an seperated -> separated
  - 7f11cfd5e1 Fix typo an helpfull -> helpful
  - 2e9b520d84 Fix typos of omitted
  - 3029f64481 Fix typo an compatbility -> compatibility
  - 470e82f17f Fix typo an dictionnary -> dictionary
  - 5843c7aa24 Fix typo an optionnal -> optional
  - 730d0f95e7 Fix typo an transfered -> transferred
  - c7e7884de2 Fix typo an recieved -> received
  - 50eea287f3 Fix typo an managment -> management
  - cb01da81c6 Fix typos of parameter
  - 45fcc7d339 Fix typo an dont -> don't
  - 3624935d32 Fix typo an other -> another
  - d16afe2607 Fix typo sofwares -> software
  - b9b7cbe525 Fix typo sofware -> software
  - 8edd2c1add Fix typos of dependency
  - 3a5e2e3437 Fix typo documention -> documentation

- **ISSUE** #28528: (schlagify) timezone.system error: CommandExecutionError: Failed to parse timedatectl output, this is likely a bug (refs: #28550)
- **PR** #28550: (jfindlay) check timedatectl errno and return stdout on failure @ *2015-11-04 15:00:24 UTC*
  - bd0b291b63 Merge pull request #28550 from jfindlay/ctl_err
  - 11a9a5868f simplify timezone module unit test mocks
  - 476b651c94 update timezone module unit tests for timedatectl
  - 5c0e5dacc0 check timedatectl errno and return stdout on failure
- **ISSUE** #19249: (ahetmanski) Cannot create cache_dir salt master exception. (refs: #28545)
- **PR** #28545: (jfindlay) pass on concurrent create of jid_dir in local_cache @ *2015-11-04 14:54:11 UTC*
  - e048667c91 Merge pull request #28545 from jfindlay/concurrent_dir
  - 58ad699331 pass on concurrent create of cache_dir in roots fs
  - e456184b04 pass on concurrent create of jid_dir in local_cache

- **PR** #28544: (rallytime) Start moving some vmware.py cloud funcs to utils/vmware.py @ *2015-11-04 14:52:59 UTC*
    - 082ffd5734 Merge pull request #28544 from rallytime/vmware-utils
    - 403fe37704 Pylint.
    - d9301eea95 Don't move _set_cd_or_dvd_backing_type yet
    - 8d69639230 Start moving some vmware.py cloud funcs to utils/vmware.py
- **PR** #28543: (gtmanfred) clean up changes for pkg.uptodate and supervisord.dead @ *2015-11-04 14:49:46 UTC*
    - bf4f7cdc4b Merge pull request #28543 from gtmanfred/2015.8
    - 3d57b392cb return changes if supervisord stopped process
    - 5547a34acc return empty changes if server is uptodate
- **ISSUE** #28524: (bmcorser) UnicodeDecodeError in states.file (refs: #28537, #28538)
- **PR** #28538: (jfindlay) decode path and url to utf-8 in url.create (refs: #28537) @ *2015-11-04 14:48:34 UTC*
    - d345768b81 Merge pull request #28538 from jfindlay/decode_state
    - b05dfc5c58 decode path and url to utf-8 in url.create
- **ISSUE** #28476: (ColorFuzzy) state.sls UnicodeDecodeError (refs: #28533)
- **PR** #28533: (jfindlay) decode highstate error messages to utf-8 @ *2015-11-04 14:47:55 UTC*
    - 2e0c8264db Merge pull request #28533 from jfindlay/decode_err
    - 9c9bb75c37 decode highstate error messages to utf-8
- **PR** #28547: (nmadhok) [Backport] [2015.8] Tasks can be in queued state instead of running @ *2015-11-04 04:13:30 UTC*
    - cfc3146b2d Merge pull request #28547 from nmadhok/2015.8-task-error
    - 3fb1f9ee6b Tasks can be in queued state instead of running. Fixes #28511
- **PR** #28535: (techhat) Fail gracefully if 169.254* isn't available @ *2015-11-03 22:39:38 UTC*
    - 7e22e7cf24 Merge pull request #28535 from techhat/fixcreds
    - 8d9224bd09 Catch timeouts too
    - fa46dbb2a3 Lint
    - f05a5e0936 Fail gracefully if 169.254* isn't available
- **PR** #28536: (cro) Default configuration file for proxy minions. @ *2015-11-03 21:26:27 UTC*
    - 9a5208e8aa Merge pull request #28536 from cro/proxyconf
    - 1e031c4940 Default configuration file for proxy minions.
- **ISSUE** #28469: (mlalpho) state boto_secgroup.present fails to find vpc_name (refs: #28710, #28534)
- **PR** #28534: (rallytime) Add versionadded directive for vpc_name arg in boto_secgroup.present @ *2015-11-03 19:30:04 UTC*
    - 2bc78a32ef Merge pull request #28534 from rallytime/fix-28469
    - ebe3b34ae7 Add versionadded directive for vpc_name arg in boto_secgroup.present
- **PR** #28516: (rallytime) Back-port #28489 to 2015.8 @ *2015-11-03 14:05:54 UTC*

  - **PR** #28489: (TheBigBear) Update windows-package-manager.rst (minor edit) adding missing single quote pairs. (refs: #28516)
  - c6a6fe0089 Merge pull request #28516 from rallytime/bp-28489
  - 2e5684a1e4 Update windows-package-manager.rst
- **PR** #28506: (basepi) [2015.8] Log minion list for all rosters, at debug level @ *2015-11-03 14:05:22 UTC*
  - 36a217acbd Merge pull request #28506 from basepi/salt-ssh.minions.log.debug
  - 06cdb50494 Log minion list for all rosters, at debug level
- **PR** #28514: (basepi) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2015-11-03 01:19:33 UTC*
  - 8cbea63e40 Merge pull request #28514 from basepi/merge-forward-2015.8
  - 463a03b2a9 Merge remote-tracking branch 'upstream/2015.5' into merge-forward-2015.8
  - 63ce8f78d5 Merge pull request #28512 from basepi/merge-forward-2015.5
    - 61c382133a Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.5
    - 4bf56cad3f Merge pull request #28461 from cachedout/issue_28455
      - 097838ec0c Wrap all cache calls in state.sls in correct umask
    - f3e61db045 Merge pull request #28407 from DSRCompany/issues/24910_token_auth_fix_2014
      - b7b5bec309 Don't request creds if auth with key.
- **PR** #28502: (cachedout) Lint #28427 @ *2015-11-02 21:09:20 UTC*
  - **PR** #28427: (cro) More updates (refs: #28502)
  - b919f55f8d Merge pull request #28502 from cachedout/lint_28427
  - 459a342102 Lint #28427
  - d354885c3d Lint
  - dbb1f0899e Lint
  - 749383c413 Lint
  - 0fa067ea30 Add datacenter getter/setter, change 'dell_switch' to just 'switch', trap call to change_password.
  - 4bcb5a508b Add datacenter getter/setter, change 'dell_switch' to just 'switch', trap call to change_password.
- **PR** #28464: (basepi) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2015-11-02 20:18:21 UTC*
  - 238411c8ce Merge pull request #28464 from basepi/merge-forward-2015.8
  - 6f6e687cb4 Mock master_uri for even tests
  - 3286a5250f Merge remote-tracking branch 'upstream/2015.5' into merge-forward-2015.8
    - 37ceae1e88 Merge pull request #28448 from gwaters/add-redhat-notes
      - e70990704a added a note to the tutorial for those that redhat so they can use the state file too.
    - 5ef50d60cd Merge pull request #28406 from rallytime/bp-28381
      - e5322d2c44 Add FreeBSD detection for VirtualBox
    - 30d5f7bbae Merge pull request #28413 from rallytime/bp-28400

  - ae1921b922 Do not execute _preflight_check if not_installed list is empty in _find_install_targets. Calling with empty list on rhel/centos cause execution of repoquery –whatprovides without pkg list which is memory consumptive task for host and also for red hat satellite server.
- 045d540aff Merge pull request #28366 from erchn/fix_yumpkg_mod_repo_disabled
  - 8187a4ce20 re-arrange things a bit to have less overall changes
  - f1d570ff18 move todelete above disabled check, add comment
  - 64feec413f also remove disabled key from repo_opts
  - 2f2ebb7bb6 mark repo not enabled when pkgrepo state passes in disable: True
- 3923f4a569 Merge pull request #28373 from beverlcl/fix-use_carrier-28372
  - 32cffeceb6 Fixing bug #28372 for use_carrier option on bonding network interfaces.
- e07e3f257b Merge pull request #28359 from rallytime/bp-28358
  - 9cacbf582b docstring typo fix - list returners not runners
- 282be7ba5a Merge pull request #28346 from twangboy/fix_installer
  - f65e3e5275 Updated documentation to reflect the new parameter
  - a0c5223554 Fixes #27923 and #28000
- 7858f04ebc Merge pull request #28315 from gwaters/update-pillar-doc
  - b15285c0b4 adding a working example of setting pillar data on the cli
- 45305ccf29 Merge pull request #28211 from terminalmage/legacy_git_pillar-2015.5
  - 0d6a4ac115 Remove non-functional test
  - ab991d61d9 Fix for ext_pillar being compiled twice in legacy git_pillar code (2015.5 branch)
- a6cc84c407 Merge pull request #28263 from cachedout/issue_26411-1
  - 3b880a5f07 New channel for event.fire_master
  - 29e9533aab Stand up a new channel if using salt-call
- 788e1463d8 Merge pull request #28293 from cachedout/fix_28271
  - 499ed8519b Minor grammar changes to #28271
- e178af0b90 Merge pull request #28271 from gwaters/update-tutorial-documentation
  - f96d39483d updated the tutorial with gravyboat's suggestions
  - b1f4a2bdf4 i think i changed the wrong header, updated to fix
  - 846b3aece1 I found you can not run the cp.push commands until after enabling the feature in the conf, so I wanted to update the docs so others who try these commands wont bump into the same issue I had.
- e3eff9b909 Merge pull request #28280 from 0xf10e/patch-1
  - 6d4316b0ac Correct Jinja function load_* to import_*
- 909fa3dc97 Merge pull request #28255 from cachedout/cli_opt
  - a2408157de Add __cli opt
- 0fa094ae11 Merge pull request #28213 from rallytime/boto_route53_state
  - 237d64ff11 If record returned None, don't continue with the state. Something went wrong.

        - 1768014705 Merge pull request #28238 from basepi/fix.schedule.present.28217
            - 087a8dc3c2 Only insert enabled if it's a dict
            - 5b49f41fab Fix schedule comparison to adjust for 'enabled' being added in schedule.list
            - 2dc1226ab8 Build new item with 'enabled' if available
        - bdd48c92de Merge pull request #28174 from lorengordon/file-replace-multiline
            - acdef2da60 Update docstrings with new guidance
            - 0835b005b7 Use a test that makes the extra file read unnecessary
            - 6d6121a6e5 Use *flags* when checking whether content was added previously
            - b25e609e9e Set *flags=8* since now the file is read as a MULTILINE string by default
            - 89e8dcdffd Use *finally* block to ensure mmap object is closed
            - 5aea6647c9 Add support for multiline regex in file.replace
        - 2225925fb5 Merge pull request #28175 from twangboy/fix_19673
            - ae8fbb208f Fixes #19673

- **ISSUE** #15583: (dr4Ke) state grain.present should accept dict values (isn't it?) (refs: #26945)
- **ISSUE** #11870: (gpkvt) Nested Grain-Support for grains.present / grains.absent (refs: #26945)
- **PR** #28486: (rallytime) Back-port #26945 to 2015.8 @ *2015-11-02 18:43:35 UTC*
    - **PR** #26945: (dr4Ke) Feature state grains support nested and dict (refs: #28486)
    - a25ce38fda Merge pull request #28486 from rallytime/bp-26945
    - 8d26bbd777 grains module and state: documentation fixes
    - df7e936910 grains module and state: use a unique object…
    - df8ec1184c grains module documentation fixes
    - 25e9a5c9ad grains state and module: fix version strings
    - eee2318873 grains state: allow deleting grain with 'False' value
    - c92326f5ea grains module: yaml representer for OrderedDict
    - 2c9c8d4073 grains state doc update
    - 576252da05 grains state: list_present, list_absent support nested grain
    - 62a1f37d86 grains state: nested support for grains.append
    - 3019a055c9 grains state: rewrite doc + example
    - c19cff517a grains state: more tests
    - cc844e4a2c grains state tests: test the grain file content as well
    - 1c5cd4c82d grains state: changes comment more accurate
    - 563fd2b56c grains state: use DEFAULT_TARGET_DELIM
    - c63913e602 grains module: simpler comment for already set key
    - 2000180791 grains.present uses grains.set
    - a03c79b13b module grains.set default comment is a string
    - 64e9e2c3b3 grains.absent uses set(None)

  - 6b8c245b87 grains state: new tests for nested grains
- **PR** #28472: (gtmanfred) overwrite more than one value with names @ *2015-11-02 17:56:53 UTC*
  - f3640b3ad6 Merge pull request #28472 from gtmanfred/2015.8
  - 8b90ccedf5 overwrite more than one value with names
- **PR** #28493: (rallytime) Back-port #28492 to 2015.8 @ *2015-11-02 17:54:09 UTC*
  - **PR** #28492: (cedwards) Updated FreeBSD installation docs (refs: #28493)
  - e31ef51053 Merge pull request #28493 from rallytime/bp-28492
  - ffc77259c9 Updated FreeBSD installation docs:
- **PR** #28494: (whiteinge) Fix filter_by passing incorrect parameters to match functions @ *2015-11-02 17:53:55 UTC*
  - 38c77206db Merge pull request #28494 from whiteinge/match-filter_by-argfix
  - e61ac75d6f Fix filter_by passing incorrect parameters to match functions
- **ISSUE** #23685: (Snergster) inotify beacon on file. 'change' event to reactor to reset file to known state will cause loop (refs: #28388)
- **PR** #28491: (rallytime) Back-port #28388 to 2015.8 @ *2015-11-02 17:13:23 UTC*
  - **PR** #28388: (cachedout) Beacon state disable (refs: #28491)
  - d19affd44d Merge pull request #28491 from rallytime/bp-28388
  - f740a19477 Working right now
  - 700eaebad0 Disable starting to come to life
  - f8b17748ef More fixing
  - 04585a2878 Documentation for disable_during_state_run
  - dbbd53689d Add documentation note in inotify beacon
  - 40217fe813 More refactoring and add new option to disable during state run
  - 19af5e5ed3 Starting on refactor of beacon config parsing
- **ISSUE** #12363: (joehealy) unable to manage password expiry of windows users (refs: #28465)
- **PR** #28465: (twangboy) Fix #12363: Password Expiration in Windows @ *2015-11-02 17:01:18 UTC*
  - f7042ba967 Merge pull request #28465 from twangboy/fix_12363
  - bcf7d58dbb Fixed array if there's a problem with user.info
  - 4b36cb8b6e Added documentation to win_shadow
  - fc8f197f69 Fix #12363
- **ISSUE** #28484: (nasenbaer13) Elasticcache subnet group creation raises TypeError (refs: #28485)
- **PR** #28485: (nasenbaer13) Fix invalid usage of _get_conn causing #28484 @ *2015-11-02 16:47:52 UTC*
  - ec0cbec00b Merge pull request #28485 from eyj/fix_28484
  - 9d80fb6070 Fix invalid usage of _get_conn causing #28484
- **ISSUE** #28453: (sdm24) Fix Formatting for Nodegroup Targetting Docs (refs: #28454)
- **ISSUE** #28268: (gravyboat) Update nodegroup docs to explain how to target via nodegroups (refs: #28306)

- **PR** #28454: (sdm24) Fixed nodegroup doc formatting to correctly link to pillar_opts in the master config @ *2015-11-02 15:14:40 UTC*
    - **PR** #28306: (sdm24) Updated the Nodegroup docs to include how to target nodegroups in SLS Jinja (refs: #28454)
    - 1116798f21 Merge pull request #28454 from sdm24/fix-formatting-in-nodegroup-docs
    - b968581eb1 Fixed nodegroup doc formatting to correctly link to pillar_opts in the master config
- **PR** #28487: (cachedout) Lint 28456 @ *2015-11-02 14:52:27 UTC*
    - fac7803a59 Merge pull request #28487 from cachedout/lint_28456
    - 58fe15437a Lint #28456
    - 322a28bb06 updated states.virtualenv_mod comments to reflect that some kwargs need 'distribute: True'
- **ISSUE** #24775: (ymote) jinja returned host ip address with square bracket (refs: #28457)
- **PR** #28457: (sdm24) Clarified comments for grains/core.py for ip_interfaces, ip4_interfac… @ *2015-11-02 14:47:59 UTC*
    - 22a4f14625 Merge pull request #28457 from sdm24/update-grain-ip-interfaces-comments
    - eb92afe238 Clarified comments for grains/core.py for ip_interfaces, ip4_interfaces, and ip6_interfaces, to explicitly state that the ips for each interface are passed as a list
- **PR** #28473: (anlutro) Show check_cmd output on failure @ *2015-11-02 14:15:30 UTC*
    - 5818b28c85 Merge pull request #28473 from alprs/feature-cmd_check_output
    - a772ce330a fix tests
    - 90b01e9e0d show check_cmd output on failure
- **PR** #28460: (justinta) Skipped wipefs test if wipefs does not exist on OS @ *2015-10-31 04:09:32 UTC*
    - cfe39df7ac Merge pull request #28460 from jtand/wipe_fs_fix
    - 7ca79f1f7b Skipped wipefs test if wipefs does not exist on OS
- **PR** #28426: (terminalmage) pkgbuild.built: make template engine optional @ *2015-10-30 17:13:36 UTC*
    - 9b44b5e347 Merge pull request #28426 from terminalmage/pkgbuild-template
    - 6d32497848 pkgbuild.built: make template engine optional
- **ISSUE** #28123: (hrumph) local.cmd not working for windows minions (refs: #28422)
- **PR** #28422: (cachedout) Handle windows logging on thread_multi [WIP] @ *2015-10-30 17:12:26 UTC*
    - 31777cb4e9 Merge pull request #28422 from cachedout/issue_28123
    - fd3b2a9e20 Handle windows logging on thread_multi
- **ISSUE** #13513: (ironwilliamcash) Windows Registry Key Problem on 64bit Machine (refs: #28425)
- **PR** #28425: (twangboy) Fix #13513 - Reflection @ *2015-10-30 17:07:23 UTC*
    - f9992fc948 Merge pull request #28425 from twangboy/fix_13513
    - beb141df69 Fixed some lint
    - 0d747355c4 Fix #13513
- **ISSUE** #27980: (rayba) salt-cloud 2015.5.0 azure provider could not be loaded (refs: #28417)
- **PR** #28417: (rallytime) Add note about azure sdk version to getting started docs @ *2015-10-29 19:47:05 UTC*

  - 4c8cd064a4 Merge pull request #28417 from rallytime/azure-version-warning
  - 8e3a2ba7e7 Add note about azure sdk version to getting started docs
- **ISSUE** #28392: (jacksontj) AsyncZeroMQReqChannel does not implement *tries* (2015.8) (refs: #28410)
- **PR** #28410: (jacksontj) Add retries to the zeromq.AsyncReqMessageClient (refs: #28576) @ *2015-10-29 18:05:50 UTC*
  - 7ead823731 Merge pull request #28410 from jacksontj/2015.8
  - 70b5ae9b1d Add retries to the zeromq.AsyncReqMessageClient
- **ISSUE** #28382: (cedwards) [FreeBSD] user state option *empty_password: True* fails with Traceback (refs: #28395)
- **PR** #28404: (rallytime) Back-port #28395 to 2015.8 @ *2015-10-29 16:09:20 UTC*
  - **PR** #28395: (cedwards) Updating bsd_shadow to match mainline shadow (refs: #28404)
  - 50845a1e91 Merge pull request #28404 from rallytime/bp-28395
  - badcb677e9 Use correct version release number
  - c5c66b8bab Updating bsd_shadow to match mainline shadow
- **PR** #28405: (opdude) Detect legacy versions of chocolatey correctly @ *2015-10-29 15:57:30 UTC*
  - e746b564b4 Merge pull request #28405 from Unity-Technologies/hotfix/choco-version-detect
  - 0076d73872 Make sure we exit out correctly when checking for choco version
  - 157e0f446d Detect legacy versions of chocolatey correctly
- **PR** #28187: (sjansen) fix at.present (refs: #28759) @ *2015-10-29 15:49:18 UTC*
  - 4304001a8f Merge pull request #28187 from sjansen/patch-1
  - 52c915e29d fix at.present
- **PR** #28375: (merll) Merge pillar includes correctly @ *2015-10-29 15:12:48 UTC*
  - 5efac26c10 Merge pull request #28375 from Precis/fix-pillar-include-loop
  - f8e2c26473 Variable err is from previous loop, too.
  - 042314246f Unit test for merging included pillars.
  - a42c51f9bf Do not merge previous values in pillar include loop.
- **PR** #28376: (ryan-lane) Support update of route53 records with multiple values @ *2015-10-29 14:54:47 UTC*
  - **PR** #28374: (ryan-lane) Support update of route53 records with multiple values (refs: #28376)
  - a69b124aaa Merge pull request #28376 from lyft/multivalue-route53-values-2015.8
  - cd221515a1 Support update of route53 records with multiple values
- **PR** #28377: (terminalmage) Deprecate 'always' in favor of 'force' in pkgbuild.built @ *2015-10-29 14:42:22 UTC*
  - 9e5a510e73 Merge pull request #28377 from terminalmage/force-pkgbuild
  - f18305e19e Add versionadded directive
  - 7046d0d896 Deprecate 'always' in favor of 'force' in pkgbuild.built
- **PR** #28380: (cro) Add missing call for service provider @ *2015-10-29 14:26:55 UTC*
  - cd632f798d Merge pull request #28380 from cro/sshprox_fix
  - 7bcc275dce Lint + logic error.

  - 92d712a54b Add a missing call for the service provider
- **ISSUE** #28202: (guettli) Docs: Difference between modules.cron.rm_job and modules.cron.rm (refs: #28348)
- **PR** #28348: (jfindlay) salt.utils.alias informs user they are using a renamed function @ *2015-10-28 20:46:36 UTC*
  - e7571e6d61 Merge pull request #28348 from jfindlay/alias
  - 7915d7e5e8 use alias util to formally alias module functions
  - 6a8b61bd12 create function alias to improve api documentation
- **PR** #28364: (justinta) In CentOS 5 the .split() causes a stacktrace. @ *2015-10-28 20:46:02 UTC*
  - 072eb98a26 Merge pull request #28364 from jtand/blockdev_test_fix
  - 3b4d03ff1a In CentOS 5 the .split() causes a stacktrace. Confirmed.split() appears to be unneeded in other OSs.
- **ISSUE** #26415: (CaesarC) salt.wheel.WheelClient doesn't work follow the python api(AttributeError: 'NoneType' object has no attribute 'get') (refs: #28087)
- **PR** #28361: (rallytime) Back-port #28087 to 2015.8 @ *2015-10-28 20:44:32 UTC*
  - **PR** #28087: (DmitryKuzmenko) Revert "Update __init__.py" (refs: #28361)
  - 06b928cfdb Merge pull request #28361 from rallytime/bp-28087
  - 41536e55b9 Revert "Update __init__.py"
- **PR** #28360: (multani) Various documentation fixes @ *2015-10-28 20:43:20 UTC*
  - d9e5fba9b5 Merge pull request #28360 from multani/fix/docs
  - ed4a54f839 doc: fix warnings in clouds.linode
  - 5a9c4c2d60 doc: simplified states.postgres_tablespace introduction
  - cf38ff1384 doc: fix rendering of titles in the /ref/states/all/ index page
- **PR** #28370: (rallytime) Back-port #28276 to 2015.8 @ *2015-10-28 20:37:49 UTC*
  - **PR** #28276: (plastikos) Correct state pkg.updtodate to succeed when packages are up-to-date (refs: #28370)
  - 4157c8331b Merge pull request #28370 from rallytime/bp-28276
  - 227ddbcb24 Simplify setting success when there are no pkg updates.
  - cd58165138 Correct state pkg.updtodate to succeed when packages are up-to-date
- **ISSUE** #27890: (dkiser) pillar recurse list strategy (refs: #27891)
- **ISSUE** #25954: (tbaker57) [2015.8.0rc2] pillar merge strategy default behaviour change (refs: #28353)
- **PR** #28353: (merll) Consider each pillar match only once. @ *2015-10-28 15:05:21 UTC*
  - **PR** #27891: (dkiser) introduce recurse_list pillar_source_merging_strategy (refs: #28353, #28013)
  - 3942b4d0e6 Merge pull request #28353 from Precis/fix-pillar-sls-matches
  - 2f3f2d6f29 Consider each pillar match only once.
- **PR** #28334: (anlutro) iptables needs -m comment for –comment to work @ *2015-10-28 14:24:52 UTC*
  - 0d8bea6c43 Merge pull request #28334 from alprs/fix-iptables_comment
  - 170ea7c50d iptables needs -m comment for –comment to work

- **ISSUE** #27789: (eduherraiz) UnicodeDecodeError: 'ascii' codec can't decode byte in 2015.8.0 (refs: #28340, #27833)
- **PR** #28340: (jfindlay) sdecode file and dir lists in fileclient @ *2015-10-28 14:23:10 UTC*
  - 7000b6ee8f Merge pull request #28340 from jfindlay/decode_client
  - bd9151b5e3 sdecode file and dir lists in fileclient
- **PR** #28344: (ryan-lane) Fix iptables state for non-filter tables @ *2015-10-28 14:21:54 UTC*
  - 48448c9a48 Merge pull request #28344 from lyft/fix-iptables-non-filter
  - 21ba070b3d Fix iptables state for non-filter tables
- **PR** #28343: (rallytime) Back-port #28342 to 2015.8 @ *2015-10-28 13:58:28 UTC*
  - **PR** #28342: (gravyboat) Fix up a dup doc entry for the file state. (refs: #28343)
  - 72f0c106cf Merge pull request #28343 from rallytime/bp-28342
  - 03d15dd090 Fix up a dup doc entry.
- **PR** #28330: (rallytime) Back-port #28305 to 2015.8 @ *2015-10-27 17:20:35 UTC*
  - **PR** #28305: (rowillia) Fix Cabal states. (refs: #28330)
  - 64d5c2362a Merge pull request #28330 from rallytime/bp-28305
  - a46dbcb62b Fix Cabal states.
- **ISSUE** #21216: (syphernl) State rabbitmq_plugin missing proper error handling (refs: #28270)
- **PR** #28270: (rallytime) Refactor RabbitMQ Plugin State to correctly use test=true and format errors @ *2015-10-27 17:18:35 UTC*
  - a44c8d8dab Merge pull request #28270 from rallytime/refactor_rabbitmq_plugin_state
  - 9e40c3a6a6 Fine tuning and fix tests
  - d50916ccdd Pylint fix
  - 196b18146d Refactor RabbitMQ Plugin State to correctly use test=true and format errors
- **ISSUE** #25363: (syphernl) rabbitmq_{user|vhost}.present in test=True reports unnecessary changes (refs: #28269)
- **ISSUE** #24856: (pruiz) rabbitmq_user state incorrectly reports result=True when using test=true (refs: #28269)
- **PR** #28269: (rallytime) Refactor rabbitmq_user state to use test=True correctly (refs: #28782, #28772) @ *2015-10-27 17:17:42 UTC*
  - 4efd07eba5 Merge pull request #28269 from rallytime/refactor_rabbitmq_user_state
  - aebbcb88ea Pylint fix
  - 19b8b868a3 Clean-up/fixes to rabbitmq_user state and test adjustments
  - 3e0e8fc8c6 Refactor rabbitmq_user state to use test=True correctly
- **ISSUE** #27855: (dverbeek84) boto_vpc is not reading availability_zone (refs: #28299, #28168)
- **PR** #28299: (rallytime) Add test for availability_zone check to boto_vpc_tests @ *2015-10-27 14:17:11 UTC*
  - **PR** #28168: (rallytime) Make sure availability zone gets passed in boto_vpc module when creating subnet (refs: #28299)
  - 93a930615e Merge pull request #28299 from rallytime/tests-for-28168
  - 65fdb50246 Get the list indice to compart before looking at keys

  - 95defb87c5 Add test for availability_zone check to boto_vpc_tests
- **ISSUE** #28268: (gravyboat) Update nodegroup docs to explain how to target via nodegroups (refs: #28306)
- **PR** #28306: (sdm24) Updated the Nodegroup docs to include how to target nodegroups in SLS Jinja (refs: #28454) @ *2015-10-27 14:07:12 UTC*
  - 0ab7c0053d Merge pull request #28306 from sdm24/update-nodegroup-docs-with-state-targeting
  - 02cac9d8c0 Update nodegroups.rst
  - b2c3307c2e Update nodegroups.rst
  - e79a930f57 updated nodegroups.rst
  - f2a6bc94df Updated the Nodegroup docs to include how to target nodegroups in SLS Jinja
- **ISSUE** #27435: (LukeCarrier) firewalld state: firewalld.prepare calls new_service, not add_service (refs: #28308)
- **PR** #28308: (rallytime) Firewalld state services should use –add-service, not –new-service @ *2015-10-27 14:02:45 UTC*
  - bba26ffeca Merge pull request #28308 from rallytime/fix-27435
  - d37298f973 Don't forget to pass the zone!
  - fcafe6f355 Firewalld state services should use –add-service, not –new-service
- **ISSUE** #21744: (rallytime) [2015.5] Multi-Master Minions Block on Authentication (refs: #28302)
- **PR** #28302: (DmitryKuzmenko) Always close socket even if there is no stream. @ *2015-10-27 01:08:41 UTC*
  - 044737ba6e Merge pull request #28302 from DSRCompany/issues/21744_fix_context_term
  - b0fc66fa68 Always close socket even if there is no stream.
- **PR** #28282: (keesbos) Fix for __env__ in legacy git_pillar @ *2015-10-26 21:20:25 UTC*
  - 2f2f51906d Merge pull request #28282 from keesbos/git-pillar-env-fix
  - d46e09afc6 Fix for __env__ in legacy git_pillar
- **PR** #28258: (pass-by-value) Add service module for ssh proxy example @ *2015-10-26 14:57:47 UTC*
  - 6a92bfbd42 Merge pull request #28258 from pass-by-value/ssh_service
  - 04bc1c64ad Add versionadded information
  - 76d8d859f1 Add service module for ssh proxy example
- **PR** #28294: (bechtoldt) correct a bad default value in http utility @ *2015-10-26 14:45:27 UTC*
  - **PR** #25668: (techhat) Sanitize sensitive fields in http.query() (refs: #28294)
  - 25778cf1ba Merge pull request #28294 from bechtoldt/fix_bad_param_default_val
  - 4852c03d08 don't iterate over var that is NoneType
- **PR** #28185: (justinta) Added single package return for latest_version, fixed other bug. @ *2015-10-26 14:09:40 UTC*
  - 0245820b73 Merge pull request #28185 from jtand/zypper_pkg
  - 457ff5d085 Added back nfo.get lines after finding the problem in them
  - 5cdb15c9e3 Added single package return for latest_version, fixed other bug.
- **PR** #28297: (cachedout) Lint fix proxy junos @ *2015-10-26 13:59:44 UTC*
  - **PR** #28116: (jejenone) converted junos proxy minion to new __proxy__ global (refs: #28297)

  - 443b486c22 Merge pull request #28297 from cachedout/lint_fix_proxy_junos
  - 5194d9a2ef Lint
  - 28eff3caf2 converted junos proxy minion to new __proxy__ global added cli() in junos.py module to execute arbitrary command
- **ISSUE** #28209: (basepi) Legacy git_pillar configs cause duplicate ext_pillar calls (refs: #28210)
- **PR** #28210: (terminalmage) Fix for ext_pillar being compiled twice in legacy git_pillar code (refs: #28211) @ *2015-10-26 12:36:58 UTC*
  - c8dd79d683 Merge pull request #28210 from terminalmage/legacy_git_pillar
  - 86f00e71bf Remove non-functional test
  - b80da6e23a Fix for ext_pillar being compiled twice in legacy git_pillar code
- **ISSUE** #28203: (edhgoose) blockdev.formatted failing on 2nd+ run, despite disk already being formatted (refs: #28265)
- **PR** #28265: (jfindlay) fix blockdev execution and state modules @ *2015-10-26 12:27:36 UTC*
  - 62485e567f Merge pull request #28265 from jfindlay/blockdev
  - 0dc72135de update blockdev exec and state module unit tests
  - 07253cb5fb move fstype checks to blockdev execution module
  - 20ec4a1dc6 move fs create logic from blockdev state to module
  - 613671a85c safer examples in blockdev exec module docs
  - 359df1bcf7 refactor dump in blockdev exec module
  - 88acc9356d check, notify for deps in blockdev exec/state mods
- **PR** #28266: (rallytime) Back-port #28260 to 2015.8 @ *2015-10-26 12:20:56 UTC*
  - **PR** #28260: (justinta) Ioflo lint (refs: #28266)
  - 556d7d583e Merge pull request #28266 from rallytime/bp-28260
  - 03509e60b2 Removed unnecessary blank line
  - 2d06c97879 Moved lint disable to end of offending line
  - d13fe0cf53 Disabled lint check for ioflo
- **PR** #28253: (rallytime) Back-port #28063 to 2015.8 @ *2015-10-23 18:10:56 UTC*
  - **PR** #28063: (SmithSamuelM) Fixes broken Salt Raet. master.flo file path broken (refs: #28253)
  - acd2214c9d Merge pull request #28253 from rallytime/bp-28063
  - db4aa58f7b Changed reference to reflect refactor of ioflo package locations as of ioflo 1.2.3 Deprecated package locations still supported in ioflo for now
  - 87abf84b54 Changed reference to reflect refactor of ioflo package locations as of ioflo 1.2.3 Deprecated package locations still supported in ioflo for now
  - 19a81dcb77 Fixed exception in loader when no file extension
  - 2afbe6803c Raet Salt broken when config moved to package directory The path to the master.flo file no longer worked This fixes
  - a177bf8f47 fixed unittests missing close of roadstack caused error on other tests
- **ISSUE** #28227: (jfindlay) DigitalOcean FreeBSD profile fails with *image: 10.2* (refs: #28231)

- **PR** #28231: (rallytime) Make sure we're compairing strings when getting images in the DO driver @ *2015-10-23 13:49:37 UTC*
  - 570e7faa3b Merge pull request #28231 from rallytime/fix-28227
  - 0985780f12 Make sure we're compairing strings when getting images in the DO driver
- **PR** #28224: (techhat) Optimize create_repo for large packages @ *2015-10-23 13:40:06 UTC*
  - 1c55513ce3 Merge pull request #28224 from techhat/spmoptimize
  - faeef55d2f Optimize create_repo for large packages
- **ISSUE** #27374: (mool) boto_route53 state doesn't create a record (refs: #28214, #28213)
- **PR** #28214: (rallytime) Don't stacktrace if invalid credentials are passed to boto_route53 state @ *2015-10-23 13:37:30 UTC*
  - **PR** #28213: (rallytime) If record returned None, don't continue with the state. Something went wrong (refs: #28214)
  - f269f40905 Merge pull request #28214 from rallytime/fix_boto_route53_stacktrace
  - cdeb8caabe Pylint Fix
  - 11c475b0ad Don't stacktrace if invalid credentials are passed to boto_route53 state
- **PR** #28228: (rallytime) Back-port #27562 to 2015.8 @ *2015-10-23 13:34:42 UTC*
  - **PR** #27562: (techhat) Add dependency resolution to SPM (refs: #28228)
  - 0775d159f8 Merge pull request #28228 from rallytime/bp-27562
  - 847809541e Updates as per @s0undt3ch
  - cf5fefdf5f Add dependency resolution to SPM
- **ISSUE** #28230: (jfindlay) DigitalOcean FreeBSD fails to bootstrap: *Please use the freebsd@ user to access this droplet.* (refs: #28232)
- **PR** #28232: (rallytime) Add documentation to supply the ssh_username: freebsd config to DO docs @ *2015-10-23 13:31:52 UTC*
  - af241dc054 Merge pull request #28232 from rallytime/fix-28230
  - 8b06ab4335 Add documentation to supply the ssh_username: freebsd config to DO docs
- **PR** #28198: (jacobhammons) Added note regarding missing spm exe on Debian/Ubuntu @ *2015-10-22 04:40:18 UTC*
  - 36dc12c62c Merge pull request #28198 from jacobhammons/docs
  - cfadda0c0c Added note regarding missing spm exe on Debian/Ubuntu Minor fixes to spm docs
- **PR** #28182: (erchn) Some fixes for nova driver for Rackspace @ *2015-10-21 21:26:18 UTC*
  - fbad88fb99 Merge pull request #28182 from erchn/fix_nova_rackspace
  - 7b54f04ba2 wrap server_list[_detailed] in try/except block for TypeError
  - b7f8487615 rackconnectv3 default to False, not 'False' get private_ips in rackconnectv2 environment and populate data object get public_ips and put in data object before returning "result" structure
- **ISSUE** #27454: (MrFishFinger) firewalld returns a dictionary rather than a string in the ret['comment'] (refs: #28181)
- **PR** #28181: (rallytime) Revamp firewalld state to be more stateful. @ *2015-10-21 21:19:18 UTC*
  - a1a924f170 Merge pull request #28181 from rallytime/fix-27454

  - 3e13880af8 Make sure we catch all potential exceptions
  - cb4efa87e8 Make sure state returns False when execution module calls fail
  - 232b2825e4 Revamp firewalld state to be more stateful.
- **PR** #28176: (cro) Add ping function @ *2015-10-21 20:49:54 UTC*
  - d93ad103c7 Merge pull request #28176 from cro/ssh_no_ping
  - 3e05437f15 Add ping function
- **PR** #28167: (The-Loeki) file.serialize needs to add a final newline to serialized files @ *2015-10-21 17:12:33 UTC*
  - 8e08f39381 Merge pull request #28167 from The-Loeki/patch-1
  - 46bf6d4fa3 Update file.serialize test for Python serialized
  - 66831fd087 file.serialize needs to add a final newline to serialized files
- **ISSUE** #27855: (dverbeek84) boto_vpc is not reading availability_zone (refs: #28299, #28168)
- **PR** #28168: (rallytime) Make sure availability zone gets passed in boto_vpc module when creating subnet (refs: #28299) @ *2015-10-21 14:48:03 UTC*
  - 559a517ad6 Merge pull request #28168 from rallytime/fix-27855
  - 50fb77dc50 Make sure availability zone gets passed in boto_vpc module when creating subnet
- **ISSUE** #26107: (thecosmicfrog) Issue targeting nodegroups - Invalid compound target: (L@ …) (refs: #28148)
- **ISSUE** #24660: (Mrten) nodegroups not backwards compatible (refs: #28148)
- **PR** #28148: (basepi) [2015.8] Only expand nodegroups to lists if there is a nested nodegroup @ *2015-10-21 13:20:06 UTC*
  - dcd90363fe Merge pull request #28148 from basepi/fix.nodegroup.backwards.compat.24660
  - 11d6a2b6ac Add some docs
  - 036d767a98 Keep track of recursive nodegroup_comp calls, keep list format if it's recursing
  - 155634a0aa Finish thought
  - 528b16756b Only expand nodegroups to lists if there is a nested nodegroup
- **PR** #28155: (basepi) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2015-10-20 23:48:41 UTC*
  - 053ad408c7 Merge pull request #28155 from basepi/merge-forward-2015.8
  - c4c889f97b Merge remote-tracking branch 'upstream/2015.5' into merge-forward-2015.8
  - ab18dcf637 Merge pull request #28140 from rallytime/bsd-installation-doc
    - 458a544d83 Add OpenBSD installation documentation to 2015.5 branch
  - fad38eb3c3 Merge pull request #28138 from rallytime/bp-28130-sizes-only
    - 6ab31e1886 Pylint
    - 37e4ed58a9 Added missing comma
    - 667f5e669f Added a bunch of instance sizes and updated some outdated ones
  - ce8f858536 Merge pull request #28097 from jacksontj/2015.5
    - 75e04bcbbc For all multi-part messages, check the headers. If the header is not your minion_id, skip the message
  - 9cdb970289 Merge pull request #28117 from rallytime/fix-23655

    - dfb908e405 Clean up stacktrace when master can't be reached in lxc cloud driver
- bf7ed0a397 Merge pull request #28110 from terminalmage/masterless-mode
    - ed90103124 Add explanation of file_client: local setting masterless mode
- a569ef4980 Merge pull request #28109 from rallytime/fix-27940
    - 18b2245611 Add created reactor event to lxc cloud driver
- d4604fdb26 Merge pull request #27996 from rallytime/fix-21845
    - f8380d751e Provide empty string as default stdout instead of None
    - f9406b5828 Don't fail if pip package is already present and pip1 is installed
- 28b97c514f Merge pull request #28056 from rallytime/bp-28033
    - af2c5ab759 Fixed win_useradd.py
- dfc3aaec74 Merge pull request #28059 from rallytime/bp-28040
    - 76a0d4937b Revert "Allow passing in auth_version, defaulting to 2."
    - 63d5675d34 default auth_version = 2
    - 8072716888 remove extra spaces
    - 9770f56f04 cleanup whitespace, default to None to be consistent with profile
    - f4adfe98c0 Allow passing in auth_version, defaulting to 2.
    - fab1ad39af Rackspace support for switft module.
- d1fa036b55 Merge pull request #28047 from cachedout/issue_27534
    - 6ea37ddbca Context manager
    - 4d6f6bb371 Lint
    - 59018289dc Restore FTP functionality to file client
- fd2ca2df1b Merge pull request #28032 from twangboy/fix_win_path
    - 2bcac93314 Fixed win_path.py
- 88c1770be4 Merge pull request #28037 from rallytime/bp-28003
    - 4fcf51fb1e Fix PR #26336
- de727d8bd2 Merge pull request #28031 from jacobhammons/relnotes6
    - 05927bb6f0 Updated release notes with additional CVE information
- 16c0272849 Merge pull request #28008 from jfindlay/host_path
    - 9f7047dd3c platform independent line endings in hosts mod
- d41018fa8e Merge pull request #28012 from rallytime/fix-28010
    - 0d7059e0c2 Clean up stack trace when something goes wrong with minion output
- f728307001 Merge pull request #27995 from jacobhammons/pillar-doc
    - 2870af2ba3 added link to grains security FAQ to targeting and pillar topics.
- efede904a7 Merge pull request #27986 from jacobhammons/dot6
    - bb61c68c11 Changed current release to 5.6 and added CVE to release notes
- 831ec680d9 Merge pull request #27913 from pass-by-value/proxmox_verify_ssl

        - 0b721efe37 Set default
    - 41cccb3a30 Merge pull request #27876 from terminalmage/git_pillar-AttributeError-2015.5
        - 07794c837a 2015.5 branch: Fix traceback when 2015.8 git ext_pillar config schema used
- **PR** #28149: (pass-by-value) Add clarification to cloud profile doc about host @ *2015-10-20 19:46:05 UTC*
    - 53dd01fc24 Merge pull request #28149 from pass-by-value/proxmox_profile_doc_change
    - bc371c55cd Add clarification to cloud profile doc about host
- **PR** #28146: (cachedout) Lint dracr.py @ *2015-10-20 17:55:07 UTC*
    - 7badd634ae Merge pull request #28146 from cachedout/lint_dracr
    - 8b057f39e8 Lint dracr.py
- **ISSUE** #28118: (basepi) Salt-cloud Linode driver using RAM number for disk size (refs: #28141)
- **PR** #28141: (rallytime) Don't use RAM for root disk size in linode.py @ *2015-10-20 17:32:29 UTC*
    - 5f99bd4dc6 Merge pull request #28141 from rallytime/fix-28118
    - 59f8e41554 Don't use RAM for root disk size in linode.py
- **PR** #28143: (justinta) Removed blank line at end of chassis.py @ *2015-10-20 16:39:35 UTC*
    - 7cd0440c33 Merge pull request #28143 from jtand/lint_fix
    - 427df95515 removed extraneous file
    - 1a58283f23 Removed blank line at end of chassis.py
- **PR** #28021: (blueyed) Handle includes in *include_config* recursively @ *2015-10-20 16:19:37 UTC*
    - 858875e9fd Merge pull request #28021 from blueyed/recursive-include
    - 1d80520958 Handle includes in *include_config* recursively
- **ISSUE** #27998: (papertigers) pkgin install broken (refs: #28001)
- **PR** #28095: (rallytime) Back-port #28001 to 2015.8 @ *2015-10-20 16:18:11 UTC*
    - **PR** #28001: (papertigers) #27998 Cleanup pkgin isatty mess (refs: #28095)
    - 4dbaec6b0c Merge pull request #28095 from rallytime/bp-28001
    - ddf8a8d2bb Cleanup pkgin isatty mess
- **ISSUE** #28060: (LoveIsGrief) Default paths for test environment (refs: #28061)
- **PR** #28096: (rallytime) Back-port #28061 to 2015.8 @ *2015-10-20 16:15:34 UTC*
    - **PR** #28061: (LoveIsGrief) Fix #28060 - Default paths for test environment (refs: #28096)
    - 572487073c Merge pull request #28096 from rallytime/bp-28061
    - cb8a72d580 Fix #28060
- **PR** #28139: (rallytime) Back-port #28103 to 2015.8 @ *2015-10-20 16:15:05 UTC*
    - **PR** #28103: (ajacoutot) OpenBSD salt package: update list of dependencies. (refs: #28140, #28139)
    - 9ce526260b Merge pull request #28139 from rallytime/bp-28103
    - bc9159a126 OpenBSD salt package: update list of dependencies.
- **ISSUE** #26844: (double-yaya) The function "state.sls" is running as PID XXXX and was started at …. with jid XXXX always shows the current jid (refs: #28098, #28097)

- **PR** #28098: (jacksontj) For all multi-part messages, check the headers. If the header is not … @ *2015-10-20 15:00:08 UTC*
  - 97dfb00a68 Merge pull request #28098 from jacksontj/2015.8
  - 6d26842925 For all multi-part messages, check the headers. If the header is not your minion-id or a broadcast, drop the message.
- **ISSUE** #3436: (madduck) Pillar does not handle Unicode data (refs: #28134, #saltstack/salt'#28134'_)
- **PR** #28134: (bernieke) fix unicode pillar values #3436 @ *2015-10-20 14:51:10 UTC*
  - b4875e585a Merge pull request #28134 from Awingu/2015.8
  - 53285f7781 fix unicode pillar values #3436
- **PR** #28076: (redmcg) Replace option 'i' with an explicit queryformat @ *2015-10-20 13:59:57 UTC*
  - f990a21029 Merge pull request #28076 from redmcg/2015.8
  - 07413ec162 Remove unnecessary padding from rpm.info
  - 4987530986 Replace option 'i' with an explicit queryformat
- **PR** #28119: (jacksontj) Check if the remote exists before casting to a string. @ *2015-10-20 12:34:10 UTC*
  - 3fdb52d1bf Merge pull request #28119 from jacksontj/fetch_issue
  - c012dcc2f6 Check if the remote exists before casting to a string.
- **ISSUE** #28080: (githubcdr) Salt minion locale module missing on Archlinux (refs: #28105)
- **PR** #28105: (jfindlay) add reason for not loading localemod @ *2015-10-20 12:25:40 UTC*
  - 69ab1d30e2 Merge pull request #28105 from jfindlay/locale_msg
  - 1e75665a9a add reason for not loading localemod
- **ISSUE** #28074: (eliasp) Salt logfiles are created world-readable (refs: #28108)
- **PR** #28108: (cachedout) Set logfile permsissions correctly @ *2015-10-20 12:25:22 UTC*
  - 8db7e016ec Merge pull request #28108 from cachedout/issue_28074
  - b416dcc07b Set logfile permsissions correctly
- **PR** #27922: (cro) WIP States/Modules for managing Dell FX2 chassis via salt-proxy @ *2015-10-19 23:29:21 UTC*
  - 1085eeab2b Merge pull request #27922 from cro/fx2
  - 6ccafa2ae5 Lint
  - 104c3cbe7f Lint
  - fe75594737 Lint
  - 479137cef8 Lint
  - 3712066fc9 More docs.
  - 2a3ebf5688 More Documentation.
  - 4ce2f8bb11 Documentation.
  - 18663306fb Cleanup, add blade_idrac stub
  - 0957beea46 Lint fixes and some changes by @rallytime
  - cca310eee0 WIP modules and states for managing Dell FX2 chassis via salt-proxy
- **PR** #28104: (pass-by-value) Add documentation for proxy minion ssh @ *2015-10-19 19:30:20 UTC*

  - a715803c92 Merge pull request #28104 from pass-by-value/proxy_ssh_docs
  - 7c8f236115 Add documentation for proxy minion ssh
- **ISSUE** #27130: (githubcdr) salt-run broken in 2015.8? (refs: #28020)
- **PR** #28020: (DmitryKuzmenko) LazyLoader deepcopy fix. @ *2015-10-19 13:17:57 UTC*
  - 07cac0b434 Merge pull request #28020 from DSRCompany/issues/27130_loader_deepcopy_fix
  - 5353518623 Fix lint errors
  - 8c256c94f4 LazyLoader deepcopy fix.
- **ISSUE** #27932: (eliasp) Can't include Pillar SLS across GitPillar repositories (refs: #27933)
- **PR** #27933: (eliasp) Provide all git pillar dirs in *opts[pillar_roots]* @ *2015-10-19 13:05:54 UTC*
  - f884df5d78 Merge pull request #27933 from eliasp/fix-27932
  - 05782aa78f Provide all git pillar dirs in *opts[pillar_roots]*
- **ISSUE** #27890: (dkiser) pillar recurse list strategy (refs: #27891)
- **PR** #28013: (rallytime) Back-port #27891 to 2015.8 @ *2015-10-19 12:57:51 UTC*
  - **PR** #27891: (dkiser) introduce recurse_list pillar_source_merging_strategy (refs: #28353, #28013)
  - 1db6406bef Merge pull request #28013 from rallytime/bp-27891
  - 9ea33bf0e4 Pylint fixes
  - 4af5b5c33f introduce recurse_list pillar_source_merging_strategy
- **ISSUE** #27938: (mostafahussein) Grains are not rendering correctly (refs: #28018)
- **PR** #28018: (rallytime) Add example to Writing Grains of how grains can be loaded twice @ *2015-10-19 12:47:10 UTC*
  - 26b3e01dda Merge pull request #28018 from rallytime/fix-27938
  - c23af0d8e2 Clarify loading vs rendering the final grains data structure
  - a4d7fb7e60 Add example to Writing Grains of how grains can be loaded twice
- **PR** #28084: (cachedout) #28069 with lint @ *2015-10-19 12:18:38 UTC*
  - **PR** #28069: (blueyed) dockerng: use error from modules.dockerng in states' __virtual__ (refs: #28084)
  - c6e7dd4812 Merge pull request #28084 from cachedout/lint_28069
  - 8026212733 Lint
  - 7a2c80cf6f dockerng: use error from modules.dockerng in states' __virtual__
- **PR** #28079: (The-Loeki) Fix for trace dump on failing imports for win32com & pythoncom 4 win_task @ *2015-10-19 12:12:11 UTC*
  - 428e64e24d Merge pull request #28079 from The-Loeki/fix-trace-on-windows-tasks
  - 869e212e81 Fix for trace dump on failing imports for win32com & pythoncom 4 win_task
- **PR** #28081: (The-Loeki) fix for glance state trace error on import failure @ *2015-10-19 12:08:47 UTC*
  - 2ac8fd793d Merge pull request #28081 from The-Loeki/fix-trace-on-keystone-state
  - 258e11f754 fix for glance state trace error on import failure
- **ISSUE** #27794: (The-Loeki) Requests backend for HTTP fetches is broken after removing streamed response handlers (refs: #28066)

- **PR** #28066: (jacksontj) Use the generic *text* attribute, not .body of the handler @ *2015-10-18 16:17:12 UTC*
  - a2128c8f80 Merge pull request #28066 from jacksontj/issue_27794
  - b1bf79821d Use the generic *text* attribute, not .body of the handler
- **ISSUE** #27828: (cubranic) Note the version when 'user' and 'group' became available in docs for archive.extracted (refs: #28019)
- **PR** #28019: (rallytime) Clean up version added and deprecated msgs to be accurate @ *2015-10-17 17:31:50 UTC*
  - 9c974c9a41 Merge pull request #28019 from rallytime/fix-27828
  - aca864643f Clean up version added and deprecated msgs to be accurate
- **PR** #28058: (rallytime) Back-port #28041 to 2015.8 @ *2015-10-17 17:27:19 UTC*
  - **PR** #28041: (gtmanfred) use the correct discover_extensions (refs: #28058)
  - 9adcd3b90d Merge pull request #28058 from rallytime/bp-28041
  - 04ad8dc521 use the correct discover_extensions
- **PR** #28055: (rallytime) Back-port #28043 to 2015.8 @ *2015-10-17 17:26:37 UTC*
  - **PR** #28043: (gtmanfred) the nova driver does not require libcloud (refs: #28055)
  - 6db970c93a Merge pull request #28055 from rallytime/bp-28043
  - 744e556be7 the nova driver does not require libcloud
- **PR** #28046: (pass-by-value) Add pkg install and remove functions @ *2015-10-17 14:56:24 UTC*
  - d7263d2a8e Merge pull request #28046 from pass-by-value/proxy_minion_ssh_example_additions
  - 3435d28fc9 Add pkg install and remove functions
- **PR** #28050: (ryan-lane) Use a better method for checking dynamodb table existence @ *2015-10-17 14:55:52 UTC*
  - dd0fdd827e Merge pull request #28050 from lyft/better-dynamo-exists-check-2015.8
  - 24fff4ea12 Use a better method for checking dynamodb table existence
- **ISSUE** #28038: (gtmanfred) [Docs] the ubuntu repo documentation needs to be fixed (refs: #28042)
- **PR** #28042: (jfindlay) fix repo path in ubuntu installation documentation @ *2015-10-16 19:30:52 UTC*
  - 027092e2fb Merge pull request #28042 from jfindlay/ubuntu_docs
  - ae92a8a1dc fix repo path in ubuntu installation documentation
- **PR** #28033: (twangboy) Fixed win_useradd.py (refs: #28056) @ *2015-10-16 19:19:44 UTC*
  - a3390cfbe6 Merge pull request #28033 from twangboy/fix_win_useradd
  - 2137b5f79a Fixed win_useradd.py
- **PR** #28027: (cro) Make ssh conn persistent. @ *2015-10-16 18:50:51 UTC*
  - 4f81358e9a Merge pull request #28027 from cro/persistent_ssh
  - 8b4067b6db Spelling, lint.
  - 76a93d5922 Spelling.
  - c800f60338 Default multiprocessing to False since anything that needs salt.vt will have trouble with our forking model.
  - cc0ad81b3d Lint, remove debug.

  - e41b677450 Make SSH connection 'persistent'. Note that right now this requires 'multiprocessing: False' in /etc/salt/proxy.

- **PR** #28029: (jacobhammons) Updated release notes with additional CVE information @ *2015-10-16 16:19:33 UTC*
  - 4dec2f9307 Merge pull request #28029 from jacobhammons/relnotes8
  - 0d1b691549 Updated release notes with additional CVE information
- **PR** #28022: (jacobhammons) Updated Debian and Ubuntu repo paths with new structure for 2015.8.1 @ *2015-10-16 15:31:36 UTC*
  - 5286c01f39 Merge pull request #28022 from jacobhammons/install
  - e4d7df8695 Updated Debian and Ubuntu repo paths with new structure for 2015.8.1
- **ISSUE** #27971: (srkunze) pip.installed returned Result: None (refs: #27983)
- **PR** #27983: (rallytime) Pip state run result should be False, not None, if installation error occurs. @ *2015-10-16 13:37:42 UTC*
  - 340229355c Merge pull request #27983 from rallytime/fix-27971
  - 9855290b99 Maintain stateful output if something went wrong running the pip command
  - 5bcc89bb8e Pip state run result should be False, not None, if installation error occurs.
- **ISSUE** #20678: (damon-atkins) Windows Installer (Separation/Downloader/Contains VC++) (refs: #27991)
- **PR** #27991: (twangboy) Fix for #20678 @ *2015-10-16 13:33:48 UTC*
  - 97d473af0d Merge pull request #27991 from twangboy/fix_20678
  - 5254ba18b3 Fix for #20678
- **ISSUE** #21845: (kitsemets) pip.install: fails in v2015.2.0rc1 when the package is already installed (pip v1.0) (refs: #27996)
- **PR** #27997: (rallytime) Remove note about pip bug with pip v1 vs pip v2 return codes @ *2015-10-16 13:23:58 UTC*
  - **PR** #27996: (rallytime) Don't fail if pip package is already present and pip1 is installed (refs: #27997)
  - bd7b39bc18 Merge pull request #27997 from rallytime/remove-pip-bug-note
  - f08d488313 Remove note about pip bug with pip v1 vs pip v2 return codes
- **PR** #27994: (justinta) Fix schedule_test failure @ *2015-10-16 13:20:56 UTC*
  - 3256e38932 Merge pull request #27994 from jtand/schedule_test-fix
  - cd67843bd0 Fix schedule_test failure
- **ISSUE** #27949: (itsamenathan) Error enabling or disabling a beacon on a minion (refs: #27992)
- **PR** #27992: (cachedout) Make load beacon config into list @ *2015-10-16 12:43:53 UTC*
  - 4a7a25eef7 Merge pull request #27992 from cachedout/issue_27949
  - 8944e1395a Make load beacon config into list
- **ISSUE** #26336: (jfindlay) windows user.present broken (refs: #28003)
- **PR** #28003: (twangboy) Fix #26336 (refs: #28037) @ *2015-10-16 12:43:07 UTC*
  - bae81d3a8d Merge pull request #28003 from twangboy/fix_26336
  - 6c94146d86 Fix PR #26336

- **PR** #27984: (rallytime) Versionadded for clean_file option for pkgrepo @ *2015-10-15 18:57:54 UTC*
    - **PR** #19561: (favadi) add pkgrepo.managed clean_file option (refs: #27984)
    - e15eeee2d3 Merge pull request #27984 from rallytime/version-clean-file
    - b094c8843e Versionadded for clean_file option for pkgrepo
- **PR** #27989: (ryan-lane) Do not try to remove the main route table association @ *2015-10-15 18:57:42 UTC*
    - 6efa71a482 Merge pull request #27989 from lyft/boto_vpc-main-route-association2-2015.8
    - 296931d29f Do not try to remove the main route table association
- **PR** #27982: (pass-by-value) Add example for salt-proxy over SSH @ *2015-10-15 17:27:57 UTC*
    - 7169fad02d Merge pull request #27982 from pass-by-value/proxy_ssh_sample
    - b85f6ab339 Add example for salt-proxy over SSH
- **PR** #27985: (jacobhammons) Changed current release to 8.1 and added CVEs to release notes @ *2015-10-15 17:27:05 UTC*
    - d0be1ab98e Merge pull request #27985 from jacobhammons/dot1
    - 236992b2be Changed current release to 8.1 and added CVEs to release notes
- **ISSUE** #27750: (justyns) Salt-master too sensitive to whitespace in public keys (again) (refs: #27979)
- **ISSUE** #21910: (justyns) Salt-master too whitespace-sensitive when dealing with minion pub keys (refs: #22115)
- **PR** #27979: (cachedout) Fix regression with key whitespace @ *2015-10-15 15:26:08 UTC*
    - **PR** #22115: (douglas-vaz) Strip whitespace characters using strip() for pub key check (refs: #27979)
    - 7e4058605d Merge pull request #27979 from cachedout/issue_27750
    - 12c6bf4358 Fix regression with key whitespace
- **ISSUE** #27712: (eduherraiz) saltutil.sync_all can't sync with the minion (refs: #27977)
- **PR** #27977: (cachedout) Decode unicode names in fileclient/server @ *2015-10-15 15:17:01 UTC*
    - 6f8925ee84 Merge pull request #27977 from cachedout/issue_27712
    - 5173ef43c8 Decode unicode names in fileclient/server
- **PR** #27981: (justinta) Fixed trailing whitespace lint @ *2015-10-15 15:10:15 UTC*
    - fc1375fc39 Merge pull request #27981 from jtand/cloudstack-lint
    - 5dfad190c2 Fixed trailing whitespace lint
- **PR** #27969: (jeffreyctang) fix parse of { on next line @ *2015-10-15 15:04:33 UTC*
    - 1ae302b202 Merge pull request #27969 from jeffreyctang/logrotate_parse
    - 2c9b2bc367 lint fixes
    - 8c6197d42e fix parse of { on next line
- **PR** #27978: (terminalmage) Add note about dockerng.inspect_image usage @ *2015-10-15 14:54:10 UTC*
    - a4ba982b1d Merge pull request #27978 from terminalmage/dockerng-inspect_image-docstring
    - 595f4a6939 Add note about dockerng.inspect_image usage
- **PR** #27955: (pass-by-value) Bp 27868 @ *2015-10-15 12:43:37 UTC*
    - **PR** #27868: (pass-by-value) Add SSHConnection object

  - bd9d1ed8b5 Merge pull request #27955 from pass-by-value/bp-27868
  - c02ec8b943 Fix pylint errors
  - 6553d135d0 Add SSHConnection object
- **PR** #27953: (The-Loeki) Fix CloudStack cloud for new 'driver' syntax @ *2015-10-15 12:38:58 UTC*
  - c50802a80f Merge pull request #27953 from The-Loeki/patch-1
  - f0d5c9f375 Pop deprecated 'provider' into new 'driver' key
  - 4e6b09edd1 Fix CloudStack cloud for new 'driver' syntax
- **PR** #27965: (ryan-lane) Fail in boto_asg.present if alarms fail @ *2015-10-15 12:32:53 UTC*
  - 7006c37627 Merge pull request #27965 from lyft/HOTFIX-boto-asg-fix
  - b8f4079c33 Fail in boto_asg.present if alarms fail
- **PR** #27958: (twangboy) Added new functionality to win_task.py @ *2015-10-15 12:30:31 UTC*
  - 6624ec1f48 Merge pull request #27958 from twangboy/update_win_task
  - 6ecbdba246 Added run_wait function
  - 5731bdcadb Clarified an error
  - 23b9c1c199 Added new functionality
- **ISSUE** #27956: (The-Loeki) Salt-cloud CLI 2015.8 borks out with global name '__opts__' is not defined (refs: #27959)
- **PR** #27959: (techhat) Change __opts__ to self.opts @ *2015-10-14 22:29:13 UTC*
  - 1efa87a964 Merge pull request #27959 from techhat/issue27956
  - bc01c48122 Change __opts__ to self.opts
- **PR** #27943: (rallytime) Back-port #27910 to 2015.8 @ *2015-10-14 20:27:20 UTC*
  - **PR** #27910: (twellspring) htpasswd state add comment about dependency on apache2-utils (refs: #27943)
  - 877e217388 Merge pull request #27943 from rallytime/bp-27910
  - 33b3d8f5b3 Clarify that apache2-utils is for Debian-based distros
  - 8ca0bc823c Add dependency on apache2-utils
- **PR** #27944: (rallytime) Back-port #27909 to 2015.8 @ *2015-10-14 20:26:52 UTC*
  - **PR** #27909: (twellspring) htpasswd module add comment about dependency on apache2-utils (refs: #27944)
  - 5f6edc8ac2 Merge pull request #27944 from rallytime/bp-27909
  - a3401c11b1 Clarify that apache2-utils is for Debian-based distros
  - 08b7bdeb97 Add dependency on apache2-utils
- **PR** #27946: (justinta) Changed grain to look at osmajorrelease instead of osrelease @ *2015-10-14 19:54:08 UTC*
  - f29ca5f87b Merge pull request #27946 from jtand/pkgrepo-fix
  - d88ac2589f Changed grain to look at osmajorrelease instead of osrelease
- **ISSUE** #27815: (tbaker57) Documentation regarding associate_eip for EC2 profiles (refs: #27914)
- **PR** #27914: (rallytime) Use eipalloc instead of eni in EC2 interface properties example @ *2015-10-14 14:37:52 UTC*

  - bb900d428b Merge pull request #27914 from rallytime/fix-27815
  - 13a9bc9053 Use eipalloc instead of eni in EC2 interface properties example
- **PR** #27926: (rallytime) Back-port #27905 to 2015.8 @ *2015-10-14 14:35:37 UTC*
  - **PR** #27905: (itsamenathan) Small documentation error for beacon disable (refs: #27926)
  - 679e603905 Merge pull request #27926 from rallytime/bp-27905
  - 30e6b055ec Small documentation error fixed
- **ISSUE** #27911: (ryan-lane) rules_egress in boto_secgroup should not manage egress rules, if set to None (refs: #27927)
- **PR** #27927: (ryan-lane) Do not manage ingress or egress rules if set to None @ *2015-10-14 14:03:17 UTC*
  - 3b4d86467b Merge pull request #27927 from lyft/boto_secgroup-fixes-2015.8
  - 0fedcc9a0b Update docs
  - 9cc65bba76 Do not manage ingress or egress rules if set to None
- **PR** #27928: (rallytime) Back-port #27908 to 2015.8 @ *2015-10-14 14:00:50 UTC*
  - **PR** #27908: (lathama) Documentation note kwargs for mdadm state already mentioned in module (refs: #27928)
  - b0f9db409d Merge pull request #27928 from rallytime/bp-27908
  - 7febb06223 Sneaky white space
  - 31d54bbe63 Note kwags for mdadm in state
- **ISSUE** #27661: (alf) The dockerng module uses deprecated API in docker-py (refs: #27676)
- **PR** #27676: (ticosax) [dockerng] WIP No more runtime args passed to docker.start() @ *2015-10-14 13:38:41 UTC*
  - 2d0b16559e Merge pull request #27676 from ticosax/no-more-arg-to-docker-start
  - a1d0ba392f fixup! Do not pass any argument to docker.start
  - 8cddbb15c4 prevent potential error while reporting mismatch versions to user.
  - 65c8762e1f Do not pass any argument to docker.start
  - d8cca2a009 docker.version_info is now provided.
- **PR** #27885: (basepi) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2015-10-13 22:44:20 UTC*
  - 722327ee5f Merge pull request #27885 from basepi/merge-forward-2015.8
  - 5ecd5615f2 Remove failing heavily-mocked test
  - 3b5e16db67 Merge remote-tracking branch 'upstream/2015.5' into merge-forward-2015.8
    - c9c3b7760e Merge pull request #27726 from jfindlay/hashhosts
      - ebce47de7c add docs to ssh.recv_known_host exec module fcn
      - b6ee16b1e5 deprecate hash_hostname in favor of hash_known_hosts
    - 18e31584b0 Merge pull request #27776 from jfindlay/local_msg
      - 03afa3cffa return message when local jobs_cache not found
    - 86cc7b5537 Merge pull request #27766 from jfindlay/debmail
      - ee78da2c27 better check for debian userdel error
    - c224386c9a Merge pull request #27758 from iggy/patch-1

        - 0994fb6a8c Remove redundant text from syslog returner
    - 34a005041f Merge pull request #27841 from terminalmage/issue27832
        - 8e09fbd6a3 Detect Manjaro Linux as Arch derivative
    - 3944a498ad Merge pull request #27852 from rallytime/bp-27806
        - a84bf18bc4 Empty string is falsy
    - 7508a1c474 Merge pull request #27838 from basepi/fix.runner.highstate.outputter.27831
        - 8ae9b66fd9 Don't pop 'outputter', we expect it further down
    - d178315f93 Merge pull request #27791 from eguven/2015.5-postgres-user-groups-backport
        - 2caf1d21d6 fix test
        - bc90c5bffe improve change reporting for postgres_user groups
        - 8712bce91a backport postgres_user groups
- **ISSUE** #26908: (twangboy) Fix *service.restart salt-minion* for other locales (refs: #27882)
- **ISSUE** #26906: (mblixter) Bug fix #22020 causes a new bug due to the expected date format for the /SD parameter in schtask.exe (refs: #27882)
- **PR** #27882: (twangboy) Created win_task.py module @ *2015-10-13 16:54:13 UTC*
    - 36f05fb526 Merge pull request #27882 from twangboy/win_task_module
    - 56c3f3ebb2 Fixed an egregious error with an import
    - 07939ea29c More lint
    - 14e060ed9c Fixed some tests
    - 1e1bd29426 Fixed some lint
    - 082277a727 Win_service.py to use the new task module
    - 2212b52620 Created win_task.py module
- **ISSUE** #27738: (fphhotchips) Git Pillar locks not managed by fileserver runner (refs: #27802)
- **PR** #27802: (terminalmage) Correct warning logging when update lock is present for git_pillar/winrepo, add runner function for clearing git_pillar/winrepo locks @ *2015-10-13 15:09:11 UTC*
    - 577191696d Merge pull request #27802 from terminalmage/issue27738
    - 1dbc3b5489 Fix comment in docstring that trailed off mid-sentence
    - 94b5fc572f Process both old and ng winrepo configs when clearing git locks
    - 7f4366d42e Add CLI example
    - 3952c66888 Change log message to reflect new runner function
    - c00ef718bf Add cache.clear_git_lock runner function
    - d7ca297f7b Add salt.fileserver.clear_lock()
    - 947ed5f739 Clarify docstring
- **ISSUE** #26632: (ryanwalder) postgres_user crashes when trying to add groups formed in a list (refs: #27886)
- **PR** #27886: (rallytime) Handle group lists as well as comma-separated group strings. @ *2015-10-13 15:00:10 UTC*
    - d655bb3616 Merge pull request #27886 from rallytime/fix-26632

  - d235abf907 Handle group lists as well as comma-separated group strings.
- **ISSUE** #26313: (anlutro) Timezone module error when timedatectl fails to query server (refs: #27746)
- **PR** #27746: (anlutro) timezone module: handle timedatectl errors @ *2015-10-13 14:55:27 UTC*
  - a158cd50e6 Merge pull request #27746 from alprs/fix-timedatectl_failure
  - f616b550b2 lint - use indexed curly brace formatting
  - bc0f167850 update timezone mod unit tests for errors
  - ef26f067b2 timezone module: handle timedatectl errors
- **ISSUE** #27710: (anlutro) salt-ssh and system.reboot/shutdown (refs: #27816)
- **PR** #27816: (anlutro) Make system.reboot use *shutdown -r* when available @ *2015-10-13 14:52:06 UTC*
  - 9dc19caa79 Merge pull request #27816 from alprs/fix-reboot_delay
  - 04ef51e524 make system.reboot use *shutdown -r* when available
- **PR** #27874: (rallytime) Add mention of Periodic Table naming scheme to deprecation docs @ *2015-10-13 14:51:45 UTC*
  - dd92b8a2e3 Merge pull request #27874 from rallytime/deprecation-docs
  - 8c056ba501 Add mention of Periodic Table naming scheme to deprecation docs
- **PR** #27883: (terminalmage) Work around –is-ancestor not being present in git-merge-base before git 1.8.0 @ *2015-10-13 14:51:27 UTC*
  - 7f96ebd69e Merge pull request #27883 from terminalmage/git-merge_base-is_ancestor
  - 45c666e8dd Work around –is-ancestor not being present in git-merge-base before git 1.8.0
  - 38d715ec0a Remove redundant SaltInvocationError raises
- **ISSUE** #24111: (yermulnik) cli option '–summary' got broken after upgrade to 2015.5.1 (refs: #24732)
- **PR** #27877: (rallytime) Back-port #27774 to 2015.8 @ *2015-10-13 14:50:45 UTC*
  - **PR** #27774: (plastikos) Summary is not correctly inspecting return data to identify not responding|connected minions (refs: #27877)
  - **PR** #27099: (plastikos) Fix access to ret parameter of _print_returns_summary() (reverts 54b33dd35948 #24732) (refs: #27774)
  - **PR** #24732: (msteed) Fix stacktrace when *–summary* is used (refs: #27099)
  - 4fb20d9b4f Merge pull request #27877 from rallytime/bp-27774
  - d940d87306 Summary is not correctly inspecting return data to identify not responding|connected minions.
- **ISSUE** #26284: (storner) apache_module.enable fails on SUSE (SLES 11 SP3) (refs: #27878)
- **PR** #27878: (rallytime) Use apache2ctl binary on SUSE in apache module @ *2015-10-13 14:45:56 UTC*
  - 97da0a87e3 Merge pull request #27878 from rallytime/fix-26284
  - 87f0d987a3 Use apache2ctl binary on SUSE in apache module
- **PR** #27879: (cro) Add docs for 2015.8.2+ changes to proxies @ *2015-10-13 14:45:30 UTC*
  - 067968c0e4 Merge pull request #27879 from cro/proxydoc
  - 5b33df9d19 Add docs for 2015.8.2+ changes
- **PR** #27731: (cro) Add __proxy__ to replace opts['proxymodule'] @ *2015-10-12 20:41:22 UTC*

- 922e2018ef Merge pull request #27731 from cro/dunder_proxy
- ba3e423b87 Missing object item throws an AttributeError not a NameError.
- 4cf2b56d5f Lint.
- dc07245df2 @rallytime is awesome. Moved proxy=None to end of def minion_mods
- 3152d8ee3f Minor loader fix
- b15083d719 Flip sense of test for grains load at end of regular minion startup
- 37c145bcd5 More places where salt.state.State needs a proxy param, sysmod had wrong __proxyenabled__, core grains were checking for proxy the wrong way.
- ed23f36279 One more check for presence of __proxy__
- 62d9f5092e what was I thinking?
- ccf366e1a5 Lint
- 8aef6e8aa9 Fix comment
- 48f9755103 Oops, forgot temp var.
- f0360ca00e More cleanup, found another spot where proxy needed to be passed to a load_modules.
- 81a4abfe5a __proxy__ is getting nuked somewhere
- f9461ff298 Add config option so old-style proxymodules will keep loading
- 3d6ed5b7ff Remove debug statement.
- b5a19a9740 Enable syncing proxymodules from the master. Proxymodules can go in /srv/salt/_proxy.
- f878011543 Lint, and some parameter fixes to add proxy= to some overridden load_modules fns.
- 22f035d8eb Remove debug statement
- 4432499b45 More progress toward __proxy__
- 1a229c17b2 Further work on __proxy__
- 85fd6a41c7 One more check for presence of __proxy__
- 15e1d3e3df Forgot absolute_import.
- c5d9d54f19 Fix py3 lint
- dd50c33543 This module was accidentally overwriting core grains during tests.
- 525256fa68 Some calls to highstate won't have __proxy__ in scope
- a615e5a876 what was I thinking?
- fae3f3ca83 Lint
- b049377cbe Remove rest_sample_test, it wasn't testing anything
- 42188480d4 Fix comment
- 4112c583e4 Oops, forgot temp var.
- e9b281041c More cleanup, found another spot where proxy needed to be passed to a load_modules.
- 64f967d731 __proxy__ is getting nuked somewhere
- bdffb9f57b Add config option so old-style proxymodules will keep loading
- b79b6a39dd Remove debug statement.

  - 02fc2d9323 Enable syncing proxymodules from the master. Proxymodules can go in /srv/salt/_proxy.
  - 72032650b8 Add __proxy__ to the list of builtins.
  - db4c034596 Lint, and some parameter fixes to add proxy= to some overridden load_modules fns.
  - 1032ad28fc Remove debug statement
  - c41e49d8e5 Make sure that the __proxy__ gets passed all the way into the state system.
  - 4a20d48b35 More progress toward __proxy__
  - d337f4329e Further work on __proxy__
- **ISSUE** #26904: (anlutro) pip install –upgrade with virtualenv.managed? (refs: #27745)
- **PR** #27745: (anlutro) Add pip_upgrade arg to virtualenv.managed state @ *2015-10-12 16:11:02 UTC*
  - 644f003fb2 Merge pull request #27745 from alprs/fix-virtualenv_pip_upgrade
  - 4bd219f8d4 add pip_upgrade arg to virtualenv.managed state, clean up docstring
- **PR** #27809: (ticosax) [dockerng] Remove dockerng.ps caching @ *2015-10-12 16:07:48 UTC*
  - 698f477336 Merge pull request #27809 from ticosax/remove-dockerng.ps-caching
  - 0eb1145856 Remove caching to prevent returning stale data from dockerng.ps
- **PR** #27859: (ticosax) [dockerng] Clarify doc port bindings @ *2015-10-12 16:06:27 UTC*
  - e96d06d71a Merge pull request #27859 from ticosax/clarify-doc-port-bindings
  - 75f7a3ec55 Must be a string
- **ISSUE** #8646: (micahhausler) Make the clean parameter in the file.directory state respect foreign require_in (refs: #27748)
- **PR** #27748: (multani) Fix #8646 @ *2015-10-12 15:55:57 UTC*
  - ba2a39d4b7 Merge pull request #27748 from multani/fix-8646
  - 6d95cbc998 Fix lint errors
  - 4ff9f4be2a Fix file.directory with clean=true and require_in with states ID
  - 0d391275de Test cases to demonstrate bug #8646
- **ISSUE** #27721: (ldobson) boto_cloudwatch_alarm.present returns diff on no change (refs: #27722)
- **PR** #27850: (rallytime) Back-port #27722 to 2015.8 @ *2015-10-12 15:31:58 UTC*
  - **PR** #27722: (ldobson) Sorted compare for alarm actions (refs: #27850)
  - ce1493e06b Merge pull request #27850 from rallytime/bp-27722
  - 33936605a0 Sorted compare for alarm actions
- **PR** #27851: (rallytime) Back-port #27771 to 2015.8 @ *2015-10-12 15:31:06 UTC*
  - **PR** #27771: (srkunze) [VIRTUALENV_MOD] added docs strings to explain parameters (refs: #27851)
  - c95437a710 Merge pull request #27851 from rallytime/bp-27771
  - 144a743503 added docs strings to explain parameters
- **ISSUE** #27789: (eduherraiz) UnicodeDecodeError: 'ascii' codec can't decode byte in 2015.8.0 (refs: #28340, #27833)
- **PR** #27833: (jfindlay) decode path before string ops in fileclient @ *2015-10-12 15:26:39 UTC*
  - a41b59bf6e Merge pull request #27833 from jfindlay/path_decode

  - 66c74e591e decode path before string ops in fileclient
- **ISSUE** #27804: (chrismcmacken) cmd.run/cmd.run_all documentation contradictory for python_shell argument (refs: #27837)
- **PR** #27837: (jfindlay) reverse truth in python_shell documentation @ *2015-10-12 15:25:13 UTC*
  - e264db7702 Merge pull request #27837 from jfindlay/true_shell
  - 1c9708a457 reverse truth in python_shell documentation
- **PR** #27860: (flavio) Fix OS related grains on openSUSE and SUSE Linux Enterprise @ *2015-10-12 15:22:59 UTC*
  - faec838744 Merge pull request #27860 from flavio/fix-os-grains-on-suse-and-opensuse
  - fc8d296d72 Fix OS related grains on openSUSE and SUSE Linux Enterprise
- **PR** #27768: (rallytime) Clean up bootstrap function to be slightly cleaner @ *2015-10-12 15:06:54 UTC*
  - 4ac5344c31 Merge pull request #27768 from rallytime/cleanup_bootstrap
  - 9df6e106c3 Clean up bootstrap function to be slightly cleaner
- **PR** #27797: (isbm) Zypper module clusterfix @ *2015-10-12 15:06:02 UTC*
  - e1bd91e392 Merge pull request #27797 from isbm/isbm-zypper-fixes
  - 36281f6b06 Bugfix: crash if no package specified on adding a lock
  - 29806a1af9 Bugfix: crash if no package specified on removing lock
  - 453a18ea15 Return an actual amount of removed locks.
  - eaa6af9898 Bugfix: sometimes error goes to the STDOUT instead of STDERR in the RPM
  - 350340dafa Bugfix: use boolean type instead of string "Yes" or "No" (NOTE: this was forgotten)
  - decb989eb4 Bugfix and refactor due to the crash on unknown package and incorrect return value
  - a6c285bd12 Initialization fix
  - 510dedd29f Bugfix: newer Zypper includes also a version of installed package
  - f9bef516de Bugfix: broken "upgrade_available" and should always return dict.
- **ISSUE** #27821: (leodus) Deploy VM on Proxmox requires 'size' configuration setting? Not according the docs! (refs: #27849)
- **PR** #27849: (rallytime) Don't require a size parameter for proxmox profiles @ *2015-10-11 01:33:28 UTC*
  - 286b08a0f5 Merge pull request #27849 from rallytime/fix-27821
  - 1bf17c7d48 Don't require a size parameter for proxmox profiles
- **PR** #27827: (techhat) Add additional error checking to SPM @ *2015-10-09 18:23:09 UTC*
  - 4a69db27cd Merge pull request #27827 from techhat/spmfixes
  - ffc8df223b Add additional error checking to SPM
- **ISSUE** #27825: (martinhoefling) Salt-api is not adding cors headers if auth fails (refs: #27826)
- **PR** #27826: (martinhoefling) Fixes #27825 @ *2015-10-09 16:08:05 UTC*
  - 9bc19ba7d2 Merge pull request #27826 from martinhoefling/fix-27825
  - 401e7de33d Fixes #27825
- **PR** #27824: (techhat) Update Azure errors @ *2015-10-09 15:25:14 UTC*
  - 1e2dede122 Merge pull request #27824 from techhat/azureerrors

  - 5b23ac7099 Update Azure errors
- **PR** #27795: (eguven) better change reporting for postgres_user groups @ *2015-10-08 23:56:53 UTC*
  - ec35666ff2 Merge pull request #27795 from eguven/2015.8-postgres_user-group-change
  - ffffede412 better change reporting for postgres_user groups
- **ISSUE** #27703: (ryan-lane) git.latest seems to ignore the user argument in 2015.8 (refs: #27799)
- **PR** #27799: (terminalmage) Fix usage of identity file in git.latest @ *2015-10-08 23:36:19 UTC*
  - 5420006209 Merge pull request #27799 from terminalmage/issue27703
  - 75d2b07b0c Pass user in calls to git.rev_parse
  - 786786a245 Fix wrong argument name for _git_run()
- **PR** #27717: (pass-by-value) Proxy beacon example @ *2015-10-08 22:58:49 UTC*
  - 0533a2b1dd Merge pull request #27717 from pass-by-value/proxy_beacon_example
  - cac3da1ffa Fix pylint error
  - 7fef5ea08c Make a call to beacon end point
  - 497f965c33 Comment
  - 8ad7082913 Add example beacon that works with salt-proxy
- **PR** #27793: (anlutro) update code that changes log level of salt-ssh shim command @ *2015-10-08 19:20:12 UTC*
  - dd9dba8f59 Merge pull request #27793 from alprs/fix-salt_ssh_b64_log
  - 2597d13fc8 update code that changes log level of salt-ssh shim command
- **ISSUE** #27714: (The-Loeki) 2015.8 git_pillar merge inconsistency/bug (refs: #27761)
- **PR** #27761: (terminalmage) Merge git pillar data instead of using dict.update() @ *2015-10-08 15:00:18 UTC*
  - bccb74ffc5 Merge pull request #27761 from terminalmage/issue27714
  - d149095bb0 Merge git pillar data instead of using dict.update()
- **PR** #27741: (ticosax) [dockerng] pass filters argument to dockerng.ps @ *2015-10-08 03:40:14 UTC*
  - 2ae7ada3c9 Merge pull request #27741 from ticosax/docker.containers-filters
  - 821ed72f37 pass filters argument to dockerng.ps
- **PR** #27760: (basepi) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2015-10-07 19:11:17 UTC*
  - 82a51cebde Merge pull request #27760 from basepi/merge-forward-2015.8
  - 35425b14ad Merge remote-tracking branch 'upstream/2015.5' into merge-forward-2015.8
  - b2937b6a16 Merge pull request #27759 from basepi/merge-forward-2015.5
    - 792ee084bb Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.5
    - d284eb165b Merge pull request #27390 from JaseFace/schedule-missing-enabled
    - 563db71bfd Ensure we pass on the enable setting if present, or use the default of True if not in build_schedule_item() Prior to this, when schedule.present compares the existing schedule to the one crafted by this function, enabled will actually be removed at each run. schedule.present sees a modification needs to be made, and invokes schedule.modify, which does so with enabled: True, creating and endless loop of an 'enabled' removal and addition.
  - 4b9128b491 Merge pull request #27732 from jacobhammons/26673

    - 75cc07cf10 noted that __virtual__ can return False and an error string
    - b928e1afa8 update docs for __virtual__ and __virtualname__ Refs #26673
  - a130896d1c Merge pull request #27747 from Sacro/fix-chocolatey-version
    - 8f1fa9e78e Chocolatey doesn't have a help command.
  - 4e48651de0 Merge pull request #27733 from jacobhammons/bug-fixes
    - cbecd4f553 Updated saltstack2 theme to add SaltConf16 banner
    - 117e0c2bcc Added hardening topic based on the information in Refs #27088
  - c58da846bf Merge pull request #27706 from jacobhammons/bug-fixes
    - 76dc8de71b Assorted doc bugs Refs #9051 Refs #13407 Refs #21475 Refs #14876 Refs #27005
  - 43fba89865 Merge pull request #27695 from rallytime/bp-27671
    - 2a88028595 Added skip test_ext_pillar_env_mapping if git module does not exist.
  - cb3d92676e Merge pull request #27524 from jfindlay/pkgng_quiet
    - 5e9107b970 parse pkgng output in quiet mode for >= 1.6.0
  - 5b88c55cc3 Merge pull request #27686 from rallytime/bp-27476
    - 3e08d3de8a fix for: https://github.com/saltstack/salt/issues/27373
  - f9ddd4647f Merge pull request #27684 from rallytime/bp-27656
    - d3780cba00 Fix #27655: handling of success in postgres_local_cache
  - 7ca6f854ff Merge pull request #27683 from rallytime/bp-27659
    - 84b6ee0c58 .pub as public key is what we should send to remote
  - a0f3e34656 Merge pull request #27682 from rallytime/bp-27566
    - 2a44255748 minor: fix/format doc for returners.local_cache.prep_jid
    - fd485e2396 returners.local_cache: fix endless loop on OSError
  - 0b9ba911c4 Merge pull request #27681 from rallytime/bp-25928
    - 17e1ddf137 Fix stacktrace for non-existant states
  - 23da0d316a Merge pull request #27680 from rallytime/bp-27535
    - 04aed5e105 Versionadded change since 2015.5.6 has already been tagged
    - 579f2646ba .. versionadded:: 2015.5.6
    - cbaf46e066 python <2.7 compability (pylint issue)
    - ecde499478 s/bin/b to avoid confusion with bin()
    - 4237c5db80 add a __virtual__ to check that daemontools is installed properly
    - 623935a1bc fix doc
    - 573de3abd6 fix pylint issue
    - 5eb6a30d40 fix pep8 issues
    - 298cf4f5c0 import missing logging module
    - fe0ad36609 log was missing
    - e457083465 s/systemd/FreeBSD

    - 3512712e89 forgot service name..
    - 8f193a7bcc fixes #27505
  - 7d7b97eab6 Merge pull request #27442 from JaseFace/fix-27391-for-2015.5
    - bfbf63e1cc Ensure we pass on the enable setting if present, or use the default of True if not in build_schedule_item() Prior to this, when schedule.present compares the existing schedule to the one crafted by this function, enabled will actually be removed at each run. schedule.present sees a modification needs to be made, and invokes schedule.modify, which does so with enabled: True, creating and endless loop of an 'enabled' removal and addition.
  - ccbba8656b Merge pull request #27641 from rallytime/gate-psutil-diskusage
    - da2d93a3dd Gate the psutil import and add depends doc for diskusage beacon
  - 09183994f9 Merge pull request #27644 from rallytime/bp-27640
    - a9063a9745 fix typo in default pillar path
  - 27fcecccbe Merge pull request #27612 from rallytime/fix-27609
    - 8dc047dc18 If external_up is set to None, don't stacktrace, just use the private ip.
    - 2ebf790f9f [salt-cloud] gce: don't stacktrace if Ephemeral is given instead of ephemeral
  - c84a1edc1b Merge pull request #27568 from jacobhammons/man-pages-five
    - b59c03d20d regenerated man pages
  - 304dc68f7f Merge pull request #27582 from jfindlay/2015.5
    - 4f0d55cda6 add 2015.5.6 release notes
  - 7201ce71e4 Merge pull request #27557 from jfindlay/mine_doc
    - 3727d79bad edit mine doc for style and markup
    - 7e037a4666 add doc motivating mine vs grains
  - 59c3d5f93e Merge pull request #27515 from jfindlay/suse_fire
    - 4460ad2785 save iptables rules on SuSE
  - 9b26357b19 Merge pull request #27509 from jfindlay/gluster_reason
    - 1ccda538d2 tell the user why the gluster module does not work
  - 989733ea86 Merge pull request #27379 from jfindlay/pip_vars
  - aee51ffdef document and check dict type for pip env_vars
- **ISSUE** #27643: (blueyed) Please document extended return values of __virtual__ (refs: #27724)
- **ISSUE** #26755: (lorengordon) Associate package dependencies to modules/states? (refs: #27724)
- **PR** #27757: (jfindlay) fix virtual fcn return doc indentation @ *2015-10-07 17:50:18 UTC*
  - **PR** #27724: (jfindlay) update __virtual__ return documentation (refs: #27757)
  - **PR** #27116: (jacobhammons) Update latest to 2015.8, 2015.5 is now previous (refs: #27724)
  - aced4229cb Merge pull request #27757 from jfindlay/virtret
  - 03400ef45b fix virtual fcn return doc indentation
- **ISSUE** #27636: (brian-bk) Salt-ssh cannot do simple state 'test.nop': "'test.nop' is not available." (refs: #27754)
- **PR** #27754: (rallytime) Change test.nop version directive to 2015.8.1 @ *2015-10-07 15:59:55 UTC*

  - 57b5b594bd Merge pull request #27754 from rallytime/fix-27636
  - 31b9852d9a Change test.nop version directive to 2015.8.1
- **PR** #27734: (jacobhammons) Updated saltstack2 theme to add SaltConf16 banner @ *2015-10-07 01:43:53 UTC*
  - 9a0171089d Merge pull request #27734 from jacobhammons/theme-updates
  - 3a52d3606b Updated saltstack2 theme to add SaltConf16 banner
- **ISSUE** #27595: (ralphvanetten) Debian package does not depend on python-m2crypto which is required by the x509 state/module (refs: #27719)
- **PR** #27727: (rallytime) Merge #27719 w/pylint fix @ *2015-10-06 21:13:37 UTC*
  - **PR** #27719: (jfindlay) tell user when x509 exec/state module can't load (refs: #27727)
  - d3f2dfe835 Merge pull request #27727 from rallytime/merge-27719
  - a7fd156162 Pylint
  - 6bf2ee2751 tell user when x509 exec/state module can't load
- **ISSUE** #27643: (blueyed) Please document extended return values of __virtual__ (refs: #27724)
- **ISSUE** #26755: (lorengordon) Associate package dependencies to modules/states? (refs: #27724)
- **PR** #27724: (jfindlay) update __virtual__ return documentation (refs: #27757) @ *2015-10-06 21:06:47 UTC*
  - **PR** #27116: (jacobhammons) Update latest to 2015.8, 2015.5 is now previous (refs: #27724)
  - f26bcd2d21 Merge pull request #27724 from jfindlay/virtret
  - 6bddf80546 update __virtual__ return documentation
- **ISSUE** #27481: (basepi) Fix issues with cross-calling states (refs: #27725)
- **PR** #27725: (basepi) Fix global injection for state cross calls @ *2015-10-06 21:02:15 UTC*
  - d67e8c5c2c Merge pull request #27725 from basepi/states.cross.call.27481
  - e12269d871 Remove unused import
  - 4e6505b2e7 Return the wrapper (whoops)
  - fadb954676 Use new method for injecting globals into state functions
  - 17b267470a Add decorator for injecting globals into functions in the loader
- **PR** #27628: (ticosax) [dockerng] Add support of *labels* parameter for dockerng @ *2015-10-06 13:58:40 UTC*
  - 06e67d25f8 Merge pull request #27628 from ticosax/dockerng-container-label
  - edf625c8b4 Add support of *labels* parameter for dockerng
- **ISSUE** #26604: (ari) Poor compound matcher documentation (2015.8 docs) (refs: #27704)
- **PR** #27704: (jacobhammons) Update compound matcher docs to clarify the usage of alternate delimi… @ *2015-10-06 05:36:55 UTC*
  - e47d849af6 Merge pull request #27704 from jacobhammons/26604
  - 1c51ce28a9 Update compound matcher docs to clarify the usage of alternate delimiters Refs #26604
- **PR** #27705: (rallytime) Merge #27602 with final pylint fix @ *2015-10-05 23:36:50 UTC*
  - **PR** #27602: (blueyed) dockerng: fix/enhance version warning in __virtual__ (refs: #27705)
  - 2491ce40f1 Merge pull request #27705 from rallytime/merge-27602
  - 81aad83386 Ignore import error

  - 561dc4cf94 dockerng: fix/enhance version warning in __virtual__
- **ISSUE** #13850: (ryan-lane) s3:// urls in file.managed (and likely elsewhere) require s3.key and s3.keyid to be in minion config (refs: #27691)
- **PR** #27691: (notpeter) Faster timeout (3s vs 2min) for instance metadata lookups. #13850. @ *2015-10-05 22:55:52 UTC*
  - b76eb08c68 Merge pull request #27691 from notpeter/iam_fail_faster
  - 3d9483b4e2 Faster timeout (3s vs 2min) for instance metadata lookups. #13850.
- **PR** #27696: (blueyed) loader.proxy: call _*modules_dirs* only once @ *2015-10-05 22:42:32 UTC*
  - fc78f49dc5 Merge pull request #27696 from blueyed/load-proxy-call-_module_dirs-only-once
  - 55a76be6c1 loader.proxy: call _*modules_dirs* only once
- **PR** #27630: (ticosax) Expose container_id in mine.get_docker @ *2015-10-05 21:56:53 UTC*
  - 77516912fa Merge pull request #27630 from ticosax/include-container-id-docker-mine
  - 7293ded2f6 fixup! Expose container_id in mine.get_docker
  - 9e56a7e9db Expose container_id in mine.get_docker
- **PR** #27600: (blueyed) dockerng: use docker.version=auto by default @ *2015-10-05 21:29:14 UTC*
  - 8453cb3eb1 Merge pull request #27600 from blueyed/dockerng-auto-version
  - 53c6e3b3de dockerng: use docker.version=auto by default
- **PR** #27689: (rallytime) Merge #27448 with test fixes @ *2015-10-05 21:17:41 UTC*
  - **PR** #27448: (JaseFace) Ensure we pass on the enable setting if present, or use the default of True if not in build_schedule_item() (refs: #27689)
  - 7a4291008e Merge pull request #27689 from rallytime/fix-tests-27448
  - 05a506ec9f Use correct comment in test
  - 8296fefb31 Merge #27448 with test fixes
  - d9f5e9fd2f Ensure we pass on the enable setting if present, or use the default of True if not in build_schedule_item() Prior to this, when schedule.present compares the existing schedule to the one crafted by this function, enabled will actually be removed at each run. schedule.present sees a modification needs to be made, and invokes schedule.modify, which does so with enabled: True, creating and endless loop of an 'enabled' removal and addition.
- **ISSUE** #27520: (rmarcinik) winrepo is unavailable in 2015.8 (refs: #27616, #27693)
- **ISSUE** #23239: (cachedout) [RFC] Deprecate ext_processses (refs: #27693)
- **PR** #27693: (jacobhammons) initial engines topic, updates to windows repo docs @ *2015-10-05 21:05:26 UTC*
  - 4ddc87157a Merge pull request #27693 from jacobhammons/doc-updates
  - 5a3e72fc8e __ops__ to __opts__
  - 5a9867aad1 initial engines topic, updates to windows repo docs Refs #23239 Refs #27520
- **PR** #27601: (blueyed) dockerng: handle None in container.Names @ *2015-10-05 20:32:19 UTC*
  - f7f48d1eef Merge pull request #27601 from blueyed/dockerng-none-names
  - b1442ac904 dockerng: handle None in container.Names
- **PR** #27596: (blueyed) gitfs: fix UnboundLocalError for 'msg' @ *2015-10-05 20:18:00 UTC*

  - 3ffb5a3369 Merge pull request #27596 from blueyed/fix-gitfs-UnboundLocalError
  - e70cbda490 gitfs: fix UnboundLocalError for 'msg'
- **PR** #27651: (eliasp) Check for existence of 'subnetId' key in subnet dict @ *2015-10-05 17:01:34 UTC*
  - 4d7be3f972 Merge pull request #27651 from eliasp/2015.8-cloud.clouds.ec2-check-for-subnetId-before-using-it
  - f21a763809 Check for existence of 'subnetId' key in subnet dict
- **ISSUE** #23370: (lisa2lisa) salt artifactory.downloaded module ignore classifier (refs: #27639)
- **PR** #27639: (rallytime) Docement version added for new artifactory options @ *2015-10-05 17:01:21 UTC*
  - d9266505a7 Merge pull request #27639 from rallytime/fix-23370
  - 6de99bd5b7 Docement version added for new artifactory options
- **PR** #27677: (rallytime) Back-port #27675 to 2015.8 @ *2015-10-05 15:47:34 UTC*
  - **PR** #27675: (avinassh) Fix a typo (refs: #27677)
  - 771e5136f1 Merge pull request #27677 from rallytime/bp-27675
  - bfa0acfbfe Fix a typo
- **PR** #27637: (rallytime) Back-port #27604 to 2015.8 @ *2015-10-05 14:54:59 UTC*
  - **PR** #27604: (plastikos) Fix module path to SaltCacheError (refs: #27637)
  - 6bc5ddc561 Merge pull request #27637 from rallytime/bp-27604
  - 3d2ee4297d Fix module path to SaltCacheError
- **ISSUE** #19291: (gfa) pkg module could accept version: latest (refs: #27657)
- **PR** #27657: (garethgreenaway) Fix to pkg state module @ *2015-10-03 23:56:02 UTC*
  - 905acc6229 Merge pull request #27657 from garethgreenaway/19291_pkg_state_latest_fix
  - c950527b24 When latest is passed in the state as the version to install, once the package is installed the state runs will fail. pkg.latest_version returned an empty string once the package is installed so we need to grab the installed version in that case to avoid passing an empty string to the pkg module in question.
- **ISSUE** #27538: (lomeroe) boto_iam is not passing parameters properly on a handful of function calls (refs: #27539)
- **PR** #27632: (rallytime) Back-port #27539 to 2015.8 @ *2015-10-02 19:28:39 UTC*
  - **PR** #27539: (lomeroe) boto_iam updates to function calls that were not passing arguments properly (refs: #27632)
  - 83ae6a1432 Merge pull request #27632 from rallytime/bp-27539
  - 2b0afd0230 Add versionadded to new path option
  - e54afed73a moving path kwarg to end of function definition
- **ISSUE** #27545: (lomeroe) boto_asg allow removing launch configuration with 'absent' state (refs: #27546)
- **ISSUE** #27544: (lomeroe) boto_asg state incorrectly processes return from boto_vpc.get_subnet_association (refs: #27559, #27546)
- **PR** #27633: (rallytime) Back-port #27559 to 2015.8 @ *2015-10-02 19:22:07 UTC*
  - **PR** #27559: (lomeroe) vpc_id fix for boto_vpc.get_subnet_association (refs: #27633)
  - **PR** #27546: (lomeroe) boto_asg state updates (refs: #27559)

  - 888e9bdf5d Merge pull request #27633 from rallytime/bp-27559
  - 3f03815ada rebasing
- **ISSUE** #27463: (ryan-lane) boto_route53 module should default to region universal, rather then None (refs: #27579)
- **PR** #27579: (rallytime) Change boto_route53 region default to 'universal' to avoid problems with boto library @ *2015-10-02 18:56:17 UTC*
  - 8b7da5e469 Merge pull request #27579 from rallytime/fix-27463
  - d5956132ef Change boto_route53 region default to 'universal' to avoid problems with boto library
- **PR** #27581: (tkwilliams) Add support for 'vpc_name' tag in boto_secgroup module and state @ *2015-10-02 15:40:40 UTC*
  - ce4c64a2e3 Merge pull request #27581 from tkwilliams/boto_secgroup_add_vpc_name
  - 159cccf43f Faulty check logic around optional params
  - 84ab0bbd74 One last bug to squash. Seriously. It's the last one. Ever! - fixed param vpc_id being passed where vpc_name was intended.
  - 002cbf5cde Grrr. Add back the import of SaltInvocationError that pylint wanted me to remove :)
  - 0671c0d8d9 Consolidate some redundant code - thanks @ryan-lane !
  - fae1199276 Followed @ryan-lane's suggestion to remove duplicated code from boto_vpc and instead call into that module
  - 3a38a440b7 Merge remote-tracking branch 'upstream/2015.8' into boto_secgroup_add_vpc_name
  - f7ef0bcd4c Fixups for picayune pylint pedantery :)
  - 35b66e28a3 Merge remote-tracking branch 'upstream/2015.8' into boto_secgroup_add_vpc_name
  - 6770f721f8 Add support for 'vpc_name' tag in boto_secgroup module and state
- **PR** #27624: (nasenbaer13) Wait for sync is not passed to boto_route53 state @ *2015-10-02 15:37:44 UTC*
  - fb6f6b9ce4 Merge pull request #27624 from eyj/fix_wait_for_sync
  - ed6a8c0aa6 Wait for sync is not passed to boto_route53 state
- **PR** #27614: (blueyed) doc: minor fixes to doc and comments @ *2015-10-02 15:34:02 UTC*
  - eb59cb8d1c Merge pull request #27614 from blueyed/doc-minor
  - 98a8c0f055 doc: minor fixes to doc and comments
- **PR** #27627: (eyj) Fix crash in boto_asg.get_instances if the requested attribute is None @ *2015-10-02 15:33:32 UTC*
  - 61f8a6f39f Merge pull request #27627 from eyj/pr-instance-attribute
  - 03d7c6af3d Fix crash in boto_asg.get_instances if the requested attribute may be None
- **ISSUE** #27549: (carlpett) Document winrepo_remotes_ng (refs: #27616)
- **ISSUE** #27520: (rmarcinik) winrepo is unavailable in 2015.8 (refs: #27616, #27693)
- **PR** #27616: (jacobhammons) Updated windows software repository docs @ *2015-10-02 05:04:37 UTC*
  - 764d70af79 Merge pull request #27616 from jacobhammons/win-repo-docs
  - 1c8b32ce26 Updated windows software repository docs

- **ISSUE** #27543: (lomeroe) boto_elb incorrectly processes return from boto_vpc.get_subnet_assocaition (refs: #27569)
- **PR** #27569: (lomeroe) boto_vpc.get_subnet_association now returns a dict w/key of vpc_id, a… @ *2015-10-01 16:03:06 UTC*
  - db963b7864 Merge pull request #27569 from lomeroe/fix_boto_elb
  - ae09a0fb61 boto_vpc.get_subnet_association now returns a dict w/key of vpc_id, adding code to handle the dict now
- **ISSUE** #25441: (ahammond) modules.ps documentation missing (refs: #27567)
- **PR** #27567: (whiteinge) Use getattr to fetch psutil.version_info @ *2015-10-01 15:39:59 UTC*
  - b269cd4754 Merge pull request #27567 from whiteinge/psutil-version-fix
  - 7ebe9acc44 Use getattr to fetch psutil.version_info
- **PR** #27583: (tkwilliams) Fixup zypper module @ *2015-10-01 15:38:53 UTC*
  - 9cc69e2440 Merge pull request #27583 from tkwilliams/fix_zypper
  - cdd44e4128 Fixup zypper module - expected return type of pkg.latest was changed without updating zypper module - unchecked list deref fixed - "zypper info -t" out-of-date status field format has changed
- **PR** #27597: (blueyed) gitfs: remove unused variable "bad_per_remote_conf" @ *2015-10-01 15:38:15 UTC*
  - 5ca7e72c70 Merge pull request #27597 from blueyed/gitfs-remove-unused-bad_per_remote_conf
  - 58af4d68f0 gitfs: remove unused variable "bad_per_remote_conf"
- **PR** #27585: (ryan-lane) Fix undefined variable in cron state module @ *2015-10-01 05:05:58 UTC*
  - 9805bdeddf Merge pull request #27585 from lyft/cron-fix
  - 5474666b61 Fix undefined variable in cron state module

28.2.58 Salt 2015.8.3 Release Notes

Version 2015.8.3 is a bugfix release for *2015.8.0*.

Statistics

- Total Merges: **74**
- Total Issue References: **26**
- Total PR References: **64**
- Contributors: **30** (DmitryKuzmenko, RealKelsar, alexproca, anlutro, basepi, bogdanr, cachedout, cedwards, chrigl, cro, fcrozat, gtmanfred, isbm, jfindlay, kiorky, kt97679, lomeroe, lorengordon, mhoogendoorn, nmadhok, optix2000, paulnivin, quantonganh, rallytime, s0undt3ch, schwing, sjorge, tampakrap, terminalmage, ticosax)

Security Fix

CVE-2015-8034 Saving `state.sls` cache data to disk with insecure permissions

This affects users of the `state.sls` function. The state run cache on the minion was being created with incorrect permissions. This file could potentially contain sensitive data that was inserted via jinja into the state SLS files. The permissions for this file are now being set correctly. Thanks to zmalone for bringing this issue to our attention.

Changelog for v2015.8.2..v2015.8.3

Generated at: 2018-05-27 23:24:21 UTC

- **PR** #29173: (jfindlay) add 2015.8.3 release notes @ *2015-11-25 00:07:51 UTC*
  - 345206b68e Merge pull request #29173 from jfindlay/2015.8
  - 212f7dd281 add 2015.8.3 release notes
  - cafbb49cb6 add note on 2015.8.2 release notes
- **PR** #29172: (basepi) [2015.8] Backport new philips_hue proxy features from develop @ *2015-11-24 23:52:55 UTC*
  - 5e88e9e9c0 Merge pull request #29172 from basepi/philips_backport
  - 1df6c3083b Backport new philips_hue proxy features from develop
- **PR** #29167: (basepi) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2015-11-24 21:40:34 UTC*
  - 2fb1ca0eac Merge pull request #29167 from basepi/merge-forward-2015.8
  - 525f9fbbbb Merge remote-tracking branch 'upstream/2015.5' into merge-forward-2015.8
  - a26c10a811 Merge pull request #29164 from jfindlay/bp-29113
    - 50fab35188 kill unneeded import
  - 4f03196e7d Merge pull request #29138 from jfindlay/2015.5
    - be045f5cb1 add 2015.5.8 release notes
- **PR** #29141: (optix2000) Add test case for require: sls with only import statements @ *2015-11-24 16:17:57 UTC*
  - 68d6c454b8 Merge pull request #29141 from optix2000/full_sls_import
  - 596843e8d6 Add test case for sls with only import Tests https://github.com/saltstack/salt/issues/10852
- **ISSUE** #29015: (jakehilton) git_pillar not honoring git_pillar_base (refs: #29072)
- **ISSUE** #28311: (strocknar) git_pillar conflicts (refs: #29072)
- **ISSUE** #27432: (mafrosis) Using specific tag as GitFS remote (refs: #29072)
- **PR** #29072: (terminalmage) Several gitfs/git_pillar fixes @ *2015-11-24 16:04:39 UTC*
  - 732f5364a2 Merge pull request #29072 from terminalmage/issue28311
  - dae738fda3 Use common code to detect envs
  - a9c0cacb77 Don't add head ref if head red matches desired ref
  - e7540e956b pygit2: Don't clean local heads along with stale remote refs
  - 1e6c46f554 pygit2: Properly resolve base saltenv from tag ref
  - 0c592ab552 Support string whitelist/blacklist
  - 744487864d Fix base branch detection for git_pillar
  - 1cd9a4d1b4 Add some debug logging for git_pillar
  - fac588c0bb Add HEAD ref in git_pillar/winrepo checkout
- **PR** #29118: (ticosax) [dockerng] Add networking capabilities @ *2015-11-24 15:47:36 UTC*
  - 95689ee1a4 Merge pull request #29118 from ticosax/dockerng-network
  - e98d18ba41 Expose docker networking as state

  - 94135d91c3 cosmetic
  - 17ff5c1ab5 Add expose networking to modules.dockerng
- **ISSUE** #29144: (anlutro) Error in fileclient with file.managed (refs: #29145)
- **PR** #29145: (anlutro) Remove duplicate import of salt.utils.s3 @ *2015-11-24 15:36:05 UTC*
  - 4b4f212d2d Merge pull request #29145 from alprs/fix-duplicate_import
  - e1101bea19 Remove duplicate import of salt.utils.s3
- **ISSUE** #29147: (lomeroe) boto_route53 unexpected keyword arguments in create_zone() (refs: #29148)
- **PR** #29148: (lomeroe) correcting parameter calls to boto get_zone/create_zone functions in … @ *2015-11-24 15:33:53 UTC*
  - 6079569580 Merge pull request #29148 from lomeroe/boto_route53_create_zone_fix-backport
  - 75408ccf99 correcting parameter calls to boto get_zone/create_zone functions in create_zone parameter check on create_zone on private_zone=True add boto version requirement
- **ISSUE** #29107: (lorengordon) Salt hangs when passing a string representation as the *length* parameter to *random.get_str()* (refs: #29108)
- **PR** #29108: (lorengordon) Enforce length as an int, fixes #29107 @ *2015-11-23 19:06:52 UTC*
  - 17638c734b Merge pull request #29108 from lorengordon/type-enforce-length
  - c71825d3b0 Enforce length as an int, fixes #29107
- **PR** #29125: (basepi) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2015-11-23 18:48:46 UTC*
  - 233ab8a474 Merge pull request #29125 from basepi/merge-forward-2015.8
  - 1432cc806d Merge remote-tracking branch 'upstream/2015.5' into merge-forward-2015.8
    - 219367a23d Merge pull request #29128 from cachedout/tweak_29122
      - b08858b040 Missed check
      - 584efe81ee Set a safer default value for ret in saltmod
  - 8d86bc3056 Merge remote-tracking branch 'upstream/2015.5' into merge-forward-2015.8
    - 2250a36647 Merge pull request #29122 from cachedout/issue_29110
      - 4b9302d794 Fix broken state orchestration
    - 200e771efb Merge pull request #29096 from rallytime/bp-29093
      - f5734423a4 Compare gem versions as a string.
    - d8a2018bc8 Merge pull request #29084 from rallytime/bp-29055
      - 52e650aed9 Add section to style guide
    - b5cff1a351 Merge pull request #29083 from rallytime/bp-29053
      - f1884de0e7 Update rabbitmq_user.py
    - b3e3bebef0 Merge pull request #28932 from twangboy/fix_28928
      - 0653a04887 Fixed user.present / user.absent in windows
    - a2e4a227e0 Merge pull request #29011 from rallytime/bp-28630
      - 7baccc1b05 Lint - newline before def
      - 9e5c16d4da Reading S3 credentials from Pillar

      - a3216f813d Fixed requests HTTPError handler, it was still in urllib2 style
    - 1a4cd6002f Merge pull request #28982 from basepi/merge-forward-2015.5
      - bfbb109fbd Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.5
      - 4b8bdd0afb Merge pull request #28839 from cachedout/revert_28740
      - 215b26c06f Revert #28740
- **ISSUE** #29005: (fcrozat) non-standard umask breaks salt-call call in salt-ssh (refs: #29126)
- **ISSUE** #28830: (fcrozat) non-standard umask breaks salt-ssh deployement (refs: #29126)
- **PR** #29126: (fcrozat) Fix deployment when umask is non-standard @ *2015-11-23 17:53:46 UTC*
  - dc0d47fa2e Merge pull request #29126 from fcrozat/2015.8
  - 4da11a5f3c Fix deployment when umask is non-standard. Fixes #29005
  - bbccb752f9 Fix deployment when umask is non-standard. Fixes #28830
- **PR** #29124: (rallytime) Back-port #28130 to 2015.8 @ *2015-11-23 17:31:00 UTC*
  - **PR** #29120: (alexproca) Import keypair (refs: #29124)
  - **PR** #28130: (bogdanr) Ec2 upload public key and updated instances size list (refs: #29124)
  - 994d8bd71a Merge pull request #29124 from rallytime/bp-28130
  - e290ea4a3f Pylint Fix
  - 9d8e5c8b4d Added missing comma
  - 4a7eee08a8 Documented import_keypair for the ec2 driver
  - 715c12014c Added a bunch of instance sizes and updated some outdated ones
  - 506ff01f65 Import public key
- **PR** #29076: (RealKelsar) We can't query installed use flags for a non installed pkg @ *2015-11-23 16:19:40 UTC*
  - d9c32011b4 Merge pull request #29076 from RealKelsar/2015.8
  - f3d1ba1509 We can't query installed use flags for a non installed pkg. Also one if is enough…
  - 96566d3060 We can't query installed use flags for a non installed pkg
- **ISSUE** #29100: (quantonganh) boto_ec2.exists does not use region when checking? (refs: #29070)
- **PR** #29097: (rallytime) Back-port #29070 to 2015.8 @ *2015-11-22 17:03:04 UTC*
  - **PR** #29070: (quantonganh) boto_ec2: missing region when checking existence of an EC2 instance (refs: #29097)
  - 1931870f26 Merge pull request #29097 from rallytime/bp-29070
  - 3b202efadc boto_ec2: missing region when checking existence of an EC2 instance
- **PR** #29090: (gtmanfred) clean up novaclient module @ *2015-11-21 15:43:58 UTC*
  - bb28b9186b Merge pull request #29090 from gtmanfred/2015.8
  - 2aab45f9d2 clean up novaclient module
- **PR** #29095: (terminalmage) Add warning about pygit2 API instability @ *2015-11-21 15:38:59 UTC*
  - 4ff54c6429 Merge pull request #29095 from terminalmage/pygit2-warning
  - 139f5ba4c3 Add warning about pygit2 API instability

- **PR** #28919: (cro) Update Philips Hue proxy minion to support __proxy__ instead of proxymodule stored in __opts__ @ *2015-11-21 15:31:36 UTC*
  - 27160b0454 Merge pull request #28919 from cro/hue_proxy_backport
  - 8823225c81 Add 'versionadded'
  - 6bdf98d2c6 Backport philips_hue proxy module to 2015.8, use __proxy__ instead of opts['proxymodule']
  - 0945d3b5b2 Add the license
  - a8be2d7382 Fix the docstring
  - 13a8973f94 Validate if "requests" are around. NOTE: this will be changed soon!
  - 835e84181b Fix the documentation
  - 68accf6180 Allow view status from all lamps, if not specified
  - 96adc9cca9 Fix lint issues
  - cd00c5d99f Remove dead code
  - 6a08d2b6b5 Implement static grains for the Philips HUE
  - 5d3c3e09fc Bugfix: show all devices, if no specific IDs were passed
  - 76e86d2d7d Implement color temperature
  - a2d87a18cc Fix the documentation
  - adeecb49d4 Implement brightness
  - a2b1a71e01 Fix crash if the controller is down
  - a7d5aafbe3 Update documentation for the color settings
  - 15f83e180d Add more preset colors
  - 44339f3dc1 Impement color setter with transition
  - 0f4d5b9eac Implement effects method
  - f341910174 Implement alert function
  - e0c95b4c7f Separate device (lamps) getter
  - 37ed834a63 Implement lamp rename
  - 66b155c3db Enhance _set method so it can set more than just lights status
  - 8e94aad5c1 Enhance internal ping report on failures (device is not reachable)
  - 3bf79e6920 Implement blink function
  - 334371d660 Use blink on internal ping
  - a8e4c2162c Fix bug: call in a proper order, if all devices
  - a98d5187f8 Remove the debug
  - a1244223bf Enhance switch method
  - e902764e25 Switch all lamps if IDs are not passed
  - 1e508e9155 Fix bug: crash, if only one lamp ID is passed
  - c0e6706d9a Implement status
  - 6d8e6d6e23 Fix lint

  - db053fbd8b Add licence
  - 2abdb19934 Implement proxy minion configuration
  - 1a75be3f71 Cleanup code
  - 1a46a180bc Implement light switch ON/OFF
  - cc5ee382c5 Implement lights method.
  - bfbe4160b2 Add constants class-struct
  - 7a8d72de3f Implement device state change
  - d769bc85a7 Implement available device listing
  - c9e7f4dc18 Cleanup code
  - 5503b6f20e Implement Philips HUE wrapper caller for Minion Proxy
  - 1b11d1ec74 Initial implementation of Philips HUE proxy
- **ISSUE** #28810: (syedaali) test.ping is not available (refs: #29065)
- **ISSUE** #28761: (syedaali) Numerous module import errors in /var/log/salt/minion (test,oracle,archive) (refs: #29065)
- **ISSUE** #25756: (nshalman) Esky builds on SmartOS broken in 2015.5 branch (refs: #25946, #25923)
- **PR** #29065: (cachedout) Handle failures inside python's inspect if a module is reloaded @ *2015-11-20 18:10:42 UTC*
  - **PR** #25946: (sjorge) Fix for salt.utils.decorators under esky (refs: #29065)
  - **PR** #25923: (sjorge) Fix for salt.utils.decorators and module.__name__ under esky (refs: #25946)
  - 88c0354c0c Merge pull request #29065 from cachedout/issue_28810
  - 4767503eb2 Remove trailing whitespace
  - c5b667f048 Handle failures inside python's inspect if a module is reloaded
- **PR** #29057: (paulnivin) Add local file support for file.managed source list @ *2015-11-19 21:57:34 UTC*
  - 714ef8ff27 Merge pull request #29057 from lyft/file-manage-local-source-list
  - 3d7aa19cd8 Support local files in list of sources
  - d175061c5d Add tests for file.source_list with local files
  - 4f8e2a30fe Update documentation to clarify URL support for lists of sources with file.managed
- **ISSUE** #28981: (mimianddaniel) 2015.8.2 import pagerduty error (refs: #29017)
- **PR** #29017: (jfindlay) pagerduty runner: add missing salt.utils import @ *2015-11-19 19:28:35 UTC*
  - f4f43381fc Merge pull request #29017 from jfindlay/pager_util
  - 5cc06207fe pagerduty runner: add missing salt.utils import
- **PR** #29039: (anlutro) Allow passing list of pip packages to virtualenv.managed @ *2015-11-19 19:13:50 UTC*
  - 1c61bce0a6 Merge pull request #29039 from alprs/feature-virtualenv_pip_pkgs
  - f9bff51382 allow passing list of pip packages to virtualenv.managed
- **PR** #29047: (schwing) Fix salt.modules.gpg.import_key exception: 'GPG_1_3_1 referenced before assignment' @ *2015-11-19 19:07:36 UTC*
  - b692ab1cfb Merge pull request #29047 from schwing/fix-gpg-exception

  - 813f6e6808 Fix 'GPG_1_3_1 referenced before assignment'
- **PR** #29050: (terminalmage) Make git_pillar global config option docs more prominent @ *2015-11-19 19:06:38 UTC*
  - b4fc2f28a4 Merge pull request #29050 from terminalmage/issue29015
  - 20da057a94 Make git_pillar global config option docs more prominent
- **PR** #29048: (nmadhok) Fix incorrect debug log statement @ *2015-11-19 19:04:10 UTC*
  - 4b3b2fe1e7 Merge pull request #29048 from nmadhok/patch-1
  - 9489d6c3b6 Update vmware.py
- **PR** #29024: (jfindlay) cache runner test: add new unit tests @ *2015-11-19 19:02:54 UTC*
  - e52c117368 Merge pull request #29024 from jfindlay/run_test
  - 0c0bce3ea6 cache runner test: add new unit tests
- **PR** #28967: (cro) Fix some issues with password changes @ *2015-11-19 18:57:39 UTC*
  - bcec8d8608 Merge pull request #28967 from cro/fx2_switch
  - 67b5b9b8d2 Add docs on automatic lockout on failed auth attempts.
  - 8a3cea4d95 Lint.
  - 04095e3b74 Prevent stacktrace if something goes wrong retrieving inventory
  - e7cbce15a5 Don't need to get grains at init time here now that we are confirming username and password differently.
  - e42100cf8a Switch from admin_password and fallback_admin_password to a list of passwords to try.
  - 4b382e977d Add 'versionadded'
- **ISSUE** #8516: (xoJIog) salt-ssh not working with nodegroups and lists (refs: #29020)
- **PR** #29020: (basepi) [2015.8] Add special list-only nodegroup support to salt-ssh @ *2015-11-18 21:15:50 UTC*
  - 14b5d0ed0f Merge pull request #29020 from basepi/salt-ssh.nodegroups.8516
  - 6433abf36f Rename ssh_nodegroups to ssh_list_nodegroups
  - bd8487b3b9 Properly save minion list in local_cache for ssh jobs
  - 4b1bf7d5e2 Add support for comma separated list matching in salt-ssh
  - 65c6528cbc Add "nodegroup" matching to salt-ssh
  - 688a78c08c Add new ssh_nodegroups config
- **ISSUE** #28911: (ccmills) GitFS numeric tags cause errors with environments (refs: #28970)
- **PR** #28970: (terminalmage) Properly handle non-string saltenvs @ *2015-11-18 20:38:41 UTC*
  - 89801b172a Merge pull request #28970 from terminalmage/issue28911
  - ec64ec85d6 Force file_roots environments to be strings
  - b2690140c7 Properly handle non-string saltenvs
- **ISSUE** #28945: (rallytime) Dell Chassis State Example Improvements (refs: #28959)
- **PR** #28959: (rallytime) Add blade password example and make note of timeout @ *2015-11-18 19:39:04 UTC*
  - 83c54351c9 Merge pull request #28959 from rallytime/fix-28945
  - 2f326b57bf Clarify chassis password functionality

  - 3614a88811 Add blade password example and make note of timeout
- **PR** #29000: (kiorky) [Mergeable] Fix up LXC @ *2015-11-18 18:02:47 UTC*
  - d8dc81bb2c Merge pull request #29000 from kiorky/2015.8_lxc
  - a4d197821a LXC: doc
  - 43fb0eff02 lxc: remove useless and error prone uses_systemd knob
  - 7ec08cd41c Fix bootstrap delay kwarg exchange
- **ISSUE** #28995: (timcharper) systemd.get_all broken on non-bsd systems / salt-bootstrap failure (refs: #29014)
- **PR** #29014: (jfindlay) systemd module: remove unneeded col command @ *2015-11-18 17:58:59 UTC*
  - eedd50e7c3 Merge pull request #29014 from jfindlay/sysctl_col
  - d75e4d5d21 systemd module: line wrap function comment
  - 960d2b936d systemd module: remove unneeded col command
- **PR** #28983: (basepi) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2015-11-18 00:49:36 UTC*
  - ac85cfdbd0 Merge pull request #28983 from basepi/merge-forward-2015.8
  - f1c80ab943 Merge remote-tracking branch 'upstream/2015.5' into merge-forward-2015.8
  - edd26d763a Merge pull request #28949 from whiteinge/sync-sdb
    - b0ec9ab25b Add sync_sdb execution function
  - 43da1bc4ce Merge pull request #28930 from twangboy/fix_28888
    - f5c489eaad Added missing import mmap required by file.py
  - 2488b873b8 Merge pull request #28908 from rallytime/doc-convention-spelling
    - 60e6eddb77 A couple of spelling fixes for doc conventions page.
  - 827a1ae020 Merge pull request #28902 from whiteinge/json-keys
    - 9745903301 Fix missing JSON support for /keys endpoint
  - d23bd49130 Merge pull request #28897 from rallytime/bp-28873
    - 077e671ead Fix salt-cloud help output typo
  - a9dc8b6ca6 Merge pull request #28871 from basepi/mdadm.fix.28870
    - 323bc2d2ac Fix command generation for mdadm.assemble
  - ec7fdc539b Merge pull request #28864 from jfindlay/2015.5
    - 648b697951 add 2015.5.7 release notes
  - bed45f4208 Merge pull request #28731 from garethgreenaway/27392_2015_5_scheduler_return_job_master
    - 771e9f7b6f Fixing the salt scheduler so that it only attempts to return the job data to the master if the scheduled job is running from a minion's scheduler.
  - 06f4932876 Merge pull request #28857 from rallytime/bp-28851
    - aa4b193f87 [states/schedule] docstring: args, kwargs -> job_args, job_kwargs
  - 0934a52b34 Merge pull request #28856 from rallytime/bp-28853
  - 37eeab2683 Typo (with → which)
- **PR** #28969: (rallytime) Back-port #28825 to 2015.8 @ *2015-11-17 20:43:30 UTC*

  - **PR** #28825: (s0undt3ch) Take into account a pygit2 bug (refs: #28969)
  - f172a0ee03 Merge pull request #28969 from rallytime/bp-28825
  - 40f4ac5b21 Add missing import
  - 2c43da1578 Take into account a pygit2 bug
- **ISSUE** #28784: (chrigl) iptables.get_saved_rules tests pretty much useless (refs: #28787)
- **ISSUE** #28783: (chrigl) iptables.get_saved_rules does not handle family=ipv6 (refs: #28787)
- **PR** #28787: (chrigl) closes #28784 @ *2015-11-17 15:54:04 UTC*
  - 1e9214f4e4 Merge pull request #28787 from chrigl/fix-28784
  - 8639e3e9c3 closes #28784
- **PR** #28944: (rallytime) The ret result must contain 'name', not 'chassis_name' for the state compiler. @ *2015-11-17 15:34:21 UTC*
  - d63344575a Merge pull request #28944 from rallytime/dellchassis-state-name-fix
  - f3ea01bbfa Make sure dellchassis.blade_idrac has a name arg and a ret['name']
  - fb718539e9 The ret result must contain 'name', not 'chassis_name' for the state compiler
- **PR** #28957: (terminalmage) Fix version number for new state option @ *2015-11-17 15:33:50 UTC*
  - fcef9f8995 Merge pull request #28957 from terminalmage/fix-docstring
  - f159000de2 Fix version number for new state option
- **PR** #28950: (DmitryKuzmenko) PR 28812 which test fix @ *2015-11-17 15:32:16 UTC*
  - **PR** #28812: (isbm) Enhance 'which' decorator reliability (refs: #28950)
  - 5b680c938a Merge pull request #28950 from DSRCompany/pr/28812_which
  - 18571000c5 Fix which test in PR'#28812'_
- **PR** #28812: (isbm) Enhance 'which' decorator reliability (refs: #28950) @ *2015-11-17 15:32:10 UTC*
  - 73719928f9 Merge pull request #28812 from isbm/isbm-which-decorator-enhancement
  - 20033eeeb7 Save modified environment path
  - 2d43199d20 Preserve 'first found first win' ordering
  - 1c59eedec2 Enhance 'which' decorator reliability for peculiar environments
- **PR** #28934: (terminalmage) git.latest: Add update_head option to prevent local HEAD from being updated @ *2015-11-17 15:15:16 UTC*
  - facc34efed Merge pull request #28934 from terminalmage/issue27883
  - 6a35a39ca5 Add update_head option to git.latest
  - 3787f7ed00 Change return output of git.fetch to a dict
  - 9ca0f8f440 Add redirect_stderr argument to cmd.run_all
- **PR** #28937: (rallytime) Update dellchassis state example to use correct jinja syntax @ *2015-11-17 15:12:28 UTC*
  - 7da93aad5b Merge pull request #28937 from rallytime/chassis-doc-fix
  - d53713ddba We only need one fancy pillar example to match our state.
  - e2926b1996 Update dellchassis state example to use correct jinja syntax

- **ISSUE** #27961: (ahammond) aggregate: False should disable aggregation even when state_aggregate: True enabled (refs: #28889)
- **PR** #28889: (jfindlay) state compiler: relax aggregate conditional check @ *2015-11-16 17:39:24 UTC*
    - 16ebda999e Merge pull request #28889 from jfindlay/aggregate
    - eb9970019a state compiler: relax aggregate conditional check
- **ISSUE** #24803: (cachedout) Rewrite GPG renderer tests (refs: #25470)
- **PR** #28921: (rallytime) Back-port #25470 to 2015.8 @ *2015-11-16 17:38:59 UTC*
    - **PR** #25470: (jfindlay) #24314 with tests (refs: #28921)
    - **PR** #24314: (cedwards) refactor gpg renderer; removing dependency on python-gnupg (refs: #28921, #25470)
    - 91a327bbce Merge pull request #28921 from rallytime/bp-25470
    - a5eee74c20 Change Beryllium to 2015.8.3 release
    - 5ce61abf57 rewrite GPG unit tests
    - 7aa424209e reduce globals in GPG renderer for easier testing
    - de5b6682ef log error and return ciphered txt on decrypt error
    - 6afb344fe3 updated logic to properly detect GPG_KEYDIR path
    - bc9750b85e refactor gpg renderer; removing dependency on python-gnupg
- **PR** #28922: (rallytime) Change 2015.8.2 release note title to reflect proper version @ *2015-11-16 16:47:33 UTC*
    - 3707eb1e7c Merge pull request #28922 from rallytime/release-notes-ver
    - 61029f8db1 Change 2015.8.2 release note title to reflect proper version
- **ISSUE** #23971: (dumol) Problems disabling a service in SLES11 SP3. (refs: #28891)
- **PR** #28891: (jfindlay) rh_service module: fix logic in _chkconfig_is_enabled @ *2015-11-16 02:44:14 UTC*
    - 23eae0d9e0 Merge pull request #28891 from jfindlay/chkconfig_check
    - e32a9aab85 rh_service._chkconfig_is_enabled unit tests
    - 5a93b7e53c rh_service module: fix logic in _chkconfig_is_enabled
- **ISSUE** #24019: (dumol) SUSE Linux Enterprise Server 11 SP3 not detected as SLES. (refs: #28892)
- **PR** #28892: (jfindlay) grains.core: correctly identify SLES 11 distrib_id @ *2015-11-16 02:30:30 UTC*
    - 8e6acd97ae Merge pull request #28892 from jfindlay/sles_grain
    - 1cfdc500c9 grains.core: correctly identify SLES 11 distrib_id
- **PR** #28910: (lorengordon) Fix winrepo command in windows pkg mgmt doc @ *2015-11-16 02:29:12 UTC*
    - cf929c3847 Merge pull request #28910 from lorengordon/patch-1
    - 64655398b3 Fix winrepo command in windows pkg mgmt doc
- **PR** #28896: (rallytime) Back-port #28855 to 2015.8 @ *2015-11-15 00:43:15 UTC*
    - **PR** #28855: (tampakrap) fix the os grain in sle11sp4 to be SUSE instead of SLES (refs: #28896)
    - 7a4fb9a790 Merge pull request #28896 from rallytime/bp-28855
    - baf238f270 fix the os grain in sle11sp4 to be SUSE instead of SLES
- **PR** #28895: (rallytime) Back-port #28823 to 2015.8 @ *2015-11-15 00:43:07 UTC*

  - **PR** #28823: (tampakrap) Add support for priority and humanname in pkrepo zypper backend (refs: #28895)
  - 64dc3c23e0 Merge pull request #28895 from rallytime/bp-28823
  - d167a6b83d Add support for priority and humanname in pkrepo zypper backend
- **ISSUE** #28754: (kt97679) service.enabled fails on xen server (refs: #28885)
- **PR** #28885: (kt97679) fix for: service.enabled fails on xen server #28754 @ *2015-11-14 04:55:38 UTC*
  - a45ce78e20 Merge pull request #28885 from kt97679/2015.8
  - 7d0f1f11cb fix for: service.enabled fails on xen server #28754
- **PR** #28880: (terminalmage) Add "profile" loglevel @ *2015-11-14 02:07:25 UTC*
  - 58b57e77be Merge pull request #28880 from terminalmage/profile-logging
  - a62852d407 Add @wraps decorator
  - cac9f17307 Add profile logging for template rendering
  - c625725f70 Add decorator to do profile-level logging for a function
  - 5a2b94ce39 Add "profile" loglevel
- **ISSUE** #28881: (basepi) salt-ssh stacktraces on first run (refs: #28882)
- **PR** #28882: (basepi) [2015.8] salt-ssh: Check return type to make sure it's an error @ *2015-11-14 00:14:46 UTC*
  - 5dc7fccb07 Merge pull request #28882 from basepi/salt-ssh.stacktrace.28881
  - f1a1cad607 Check return type to make sure it's actually an error
- **PR** #28867: (rallytime) [fx2 grains] Grains functions should return dictionaries @ *2015-11-13 21:14:13 UTC*
  - 430e9376f6 Merge pull request #28867 from rallytime/fx2-grains-patch
  - 022cf5d230 [fx2 grains] Grains functions should return dictionaries
- **ISSUE** #28859: (mhoogendoorn) ebuild.install runs *refresh_db()* when *refresh=False* is given. (refs: #28863)
- **PR** #28863: (mhoogendoorn) Fix ebuild.install causing extra refresh_db calls. @ *2015-11-13 18:46:03 UTC*
  - 304072456e Merge pull request #28863 from mhoogendoorn/fix-issue-28859
  - eca09b89a4 Fix ebuild.install causing extra refresh_db calls.

28.2.59 Salt 2015.8.4 Release Notes

Version 2015.8.4 is a bugfix release for *2015.8.0*.

Statistics

- Total Merges: **322**
- Total Issue References: **120**
- Total PR References: **312**
- Contributors: **78** (AkhterAli, DmitryKuzmenko, MadsRC, Oro, The-Loeki, abednarik, akissa, anlutro, basepi, bastiaanb, bdrung, borgstrom, cachedout, clan, clinta, cournape, cro, ctrlrsf, dmacvicar, dmurphy18, dnd, dr4Ke, eliasp, fcrozat, frioux, galet, garethgreenaway, gqgunhed, gtmanfred, hexedpackets, isbm, jacksontj, jacobhammons, jfindlay, jleimbach, job, joejulian, julianbrost, justinta, kingsquirrel152, kiorky, l2ol33rt, lagesag,

lorengordon, mbarrien, mpreziuso, multani, nmadhok, oeuftete, opdude, optix2000, pass-by-value, paulnivin, plastikos, pritambaral, rallytime, rasathus, rmatulat, ruxandraburtica, ryan-lane, s0undt3ch, seanjnkns, serge-p, sjorge, stanislavb, tbaker57, techhat, terminalmage, thatch45, thegoodduke, thomaso-mirodin, ticosax, tim-charper, tkunicki, trevor-h, twangboy, whiteinge, whytewolf)

Known Issues

- `*_in` requisites (issue #30820)

  This issue affects all users targeting an explicit `- name: <name>` with a `_in` requisite (such as `watch_in` or `require_in`). If you are not using explicit `- name: <name>` arguments, are targeting with the state ID instead of the name, or are not using `_in` requisites, then you should be safe to upgrade to 2015.8.4.

  This issue is resolved in the *2015.8.5* release.

Security Fix

CVE-2016-1866 Improper handling of clear messages on the minion, which could result in executing commands not sent by the master.

This issue affects only the 2015.8.x releases of Salt. In order for an attacker to use this attack vector, they would have to execute a successful attack on an existing TCP connection between minion and master on the pub port. It does not allow an external attacker to obtain the shared secret or decrypt any encrypted traffic between minion and master. Thank you to Sebastian Krahmer <krahmer@suse.com> for bringing this issue to our attention.

We recommend everyone upgrade to 2015.8.4 as soon as possible.

Core Changes

- Support for IAM roles added to S3 module
- Added option `mock=True` for `state.sls` and `state.highstate`. This allows the state compiler to process sls data in a state run without actually calling the state functions, thus providing feedback on the validity of the arguments used for the functions beyond the preprocessing validation provided by `state.show_sls` (issue #30118 and issue #30189).

```
salt '*' state.sls core,edit.vim mock=True
salt '*' state.highstate mock=True
salt '*' state.apply edit.vim mock=True
```

Changelog for v2015.8.3..v2015.8.4

Generated at: 2018-05-27 23:28:18 UTC

- **PR** #30615: (jfindlay) add 2015.8.4 release notes @ *2016-01-25 18:11:02 UTC*
  - 1c6c394d0e Merge pull request #30615 from jfindlay/2015.8
  - e4043403e4 add 2015.8.4 release notes
- **PR** #30612: (rallytime) Back-port #29940 to 2015.8 @ *2016-01-25 17:52:43 UTC*
  - **PR** #29940: (dr4Ke) file.line: better diff (refs: #30612)
  - ec50581aad Merge pull request #30612 from rallytime/bp-29940
  - 3ebb8249d7 file.line: better diff

- **PR** #30613: (basepi) Fix minion/syndic clearfuncs @ *2016-01-25 17:40:59 UTC*
  - 48373e0ea9 Merge pull request #30613 from basepi/minion_clearfuncs_2015.8
  - a3c3182f39 Correctly handle clearfuncs on the syndic
  - 098ce4335d Correct handle clearfuncs on the minion
- **ISSUE** #29601: (seanjnkns) pillars not merging properly with 2015.8.3 (refs: #30062)
- **PR** #30609: (seanjnkns) Fix documentation for pillar_merge_lists which default is False, not … @ *2016-01-25 17:15:45 UTC*
  - **PR** #30062: (seanjnkns) Remove recurse_list from pillar_source_merging_strategy and add pilla… (refs: #30609, #30458)
  - 89b4f3de1b Merge pull request #30609 from seanjnkns/backport_30602
  - 9924acdc43 Fix documentation for pillar_merge_lists which default is False, not True. From PR #30062
- **PR** #30584: (julianbrost) file.line state: add missing colon in docstring @ *2016-01-25 16:37:38 UTC*
  - 24ead62c41 Merge pull request #30584 from julianbrost/fix-doc-file-line-missing-colon
  - 2ab367f95b file.line state: add missing colon in docstring
- **PR** #30589: (terminalmage) Merge 2015.5 into 2015.8 @ *2016-01-25 16:20:41 UTC*
  - a7ba2df5e2 Merge pull request #30589 from terminalmage/2015.5-2015.8
  - d649551fbf Merge branch '2015.5' into 2015.5-2015.8
    - a823e21428 Merge pull request #30582 from terminalmage/dnf-repoquery-multiple-targets
      - 410da789f9 yumpkg.check_db: run separate repoquery commands when multiple names passed
    - 8e56be7f4c Merge pull request #30548 from jacobhammons/doc-fixes
      - 03c51bb54d Added placeholder release notes for 2015.5.10 Changed old doc links from docs.saltstack.org to docs.saltstack.com
    - 1aafd4c5b5 Merge pull request #30530 from terminalmage/yumpkg-dnf-cleanup
      - 2586f71bcf 2015.5 tweaks from #30529
- **PR** #30599: (multani) Documentation formatting fixes @ *2016-01-25 15:37:46 UTC*
  - 3a55d11916 Merge pull request #30599 from multani/fix/docs
  - 038ecc4acd For doc formatting of salt.states.module
  - 4062c63b9f Fix doc formatting for yaml_idiosyncrasies
  - 6efb77bc04 Fix doc formatting of salt.modules.parted
  - a329adfb21 Add missing salt.queues.* documentation
  - 2465cf4ba5 Remove non-existing documentation
  - 814e64c304 Fix documentation markup in salt.modules.osquery
  - d2614d6169 Fix documentation markup for salt.modules.ipmi
  - 276eb3a843 Fix GCE documentation
- **ISSUE** #10157: (martinb3) salt-cloud actions don't have very useful error messages (refs: #30554)
- **PR** #30554: (rallytime) Make the salt-cloud actions output more verbose and helpful @ *2016-01-22 20:23:18 UTC*

  - b1e604add3 Merge pull request #30554 from rallytime/fix-10157
  - 6fa952f16d Make the salt-cloud actions output more verbose and helpful
- **PR** #30549: (techhat) Salt Virt cleanup @ *2016-01-22 18:45:18 UTC*
  - 2eb5a3803d Merge pull request #30549 from techhat/virtcleanup
  - 9baab73cd0 Fix copy pasta
  - 6413c11f29 Salt Virt cleanup
- **PR** #30553: (techhat) AWS: Support 17-character IDs @ *2016-01-22 18:41:46 UTC*
  - f63b183e43 Merge pull request #30553 from techhat/awsid
  - a95fbff4bc Support 17-character IDs
- **PR** #30532: (whiteinge) Add execution module for working in sls files @ *2016-01-22 17:25:16 UTC*
  - 05d05263ab Merge pull request #30532 from whiteinge/slsutil-mod
  - a57d9984e4 Add slsutil to doc index
  - 155966c9d2 Add execution module for working in sls files
- **PR** #30529: (terminalmage) Merge 2015.5 into 2015.8 (refs: #30530) @ *2016-01-22 17:19:39 UTC*
  - 1da1bb9afc Merge pull request #30529 from terminalmage/2015.5-2015.8
  - e85ad690fb Lint fixes
  - 43829ecee6 Docstring tweaks
  - 92d5a2a49c Fix spelling
  - fdc60fc04a Modify pkg.group_installed to reflect changes in yumpkg.py
  - a118eb5d2e Merge branch '2015.5' into 2015.5-2015.8
  - 7798d42272 Merge pull request #30484 from terminalmage/dnf-yumpkg-2015.5
    - 330e26d1da Hide get_locked_packages
    - 5a637420e8 Backport DNF support to 2015.5 branch
- **PR** #30526: (twangboy) Added FlushKey to make sure it's changes are saved to disk @ *2016-01-22 02:33:13 UTC*
  - e366f6a7fd Merge pull request #30526 from twangboy/reg_flushkey
  - 23085ffbbb Added FlushKey to make sure it's changes are saved to disk
- **PR** #30521: (basepi) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2016-01-21 23:05:03 UTC*
  - cdc731b8c5 Merge pull request #30521 from basepi/merge-forward-2015.8
  - f22f5ff851 Fix lint
  - 117fb205de Merge remote-tracking branch 'upstream/2015.5' into merge-forward-2015.8
    - b348f804b1 Merge pull request #30512 from jfindlay/repo_test
      - 66f06f2bd3 disable pkgrepo test for ubuntu 15.10+
    - a9348dfef8 Merge pull request #30478 from jtand/pip_8_update
      - 6227368830 Convert version to int, instead of comparing strings to ints
      - 20384a4810 Added InstallationError to except block
      - baa274bca9 Updated pip_state to work with pip 8.0

    - a30147c64f Merge pull request #30482 from borgstrom/pyobjects_recursive
      - 2c55a7580b Fixup lint errors
      - b46df0e4b5 Allow recursive salt:// imports
      - 51bfa16173 Add test to prove that recursive imports are currently broken
    - 5c7cc51937 Merge pull request #30459 from jfindlay/pkg_tests
      - fb9972f590 modules.pkg: disable repo int test for ubuntu 15.10
    - dd2ceb4c07 Merge pull request #30443 from jtand/boto_vpc_5
      - 2f77152479 Boto uses False for is_default instead of None
    - 62d9dddced Merge pull request #30420 from attiasr/patch-1
      - 4de343c5a1 Backport #26853
- **PR** #30485: (justinta) Updated pip_state to work with pip 8.0 on 2015.8 @ *2016-01-21 22:55:38 UTC*
  - 019af349af Merge pull request #30485 from jtand/pip_8_update_2015.8
  - 9cb17332fa Updated pip_state to work with pip 8.0 on 2015.8
- **PR** #30494: (isbm) Zypper: info_installed — 'errors' flag change to type 'boolean' @ *2016-01-21 22:55:05 UTC*
  - 3259fde362 Merge pull request #30494 from isbm/isbm-zypper-nfoinst-bool-fix
  - 4d7659270e Place the boolean check
  - 58db1c3b16 Fix typo
  - 43254aa993 Update docstring according to the boolean flag
  - a7d3e0d5ad Change 'errors' flag to boolean.
- **PR** #30506: (jacksontj) Properly remove newlines after reading the file @ *2016-01-21 22:53:57 UTC*
  - 596892326d Merge pull request #30506 from jacksontj/2015.8
  - e1dea6f843 Properly remove newlines after reading the file
- **ISSUE** #30444: (dnd) Cloning linode server with salt-cloud fails trying to create disk config (refs: #30508)
- **ISSUE** #30432: (dnd) Cloning linode server with salt-cloud requires payment term (refs: #30508)
- **PR** #30508: (rallytime) Fix Linode driver cloning functionality @ *2016-01-21 22:53:36 UTC*
  - 15c7aedd46 Merge pull request #30508 from rallytime/linode-clone-fixes
  - d26ed74bde Make sure the correct profile parameters are being checked when cloning
  - 1d7e229377 Fix Linode driver cloning functionality.
- **PR** #30522: (terminalmage) Update git.list_worktree tests to reflect new return data @ *2016-01-21 22:34:20 UTC*
  - 79528c59c3 Merge pull request #30522 from terminalmage/fix-worktree-tests
  - ea0ca70187 Add git.list_worktrees unit test
  - 393015edbb Remove git.list_worktrees tests
- **ISSUE** #30465: (alandrees) Nested imports with pyobjects (refs: #30483, #30482)
- **PR** #30483: (borgstrom) Pyobjects recursive import support (for 2015.8) @ *2016-01-21 15:55:27 UTC*
  - 119f025073 Merge pull request #30483 from borgstrom/pyobjects_recursive-2015.8
  - 788b672e3a Fixup lint errors

  - e148ea2d52 Allow recursive salt:// imports
  - 6bbac64d3a Add test to prove that recursive imports are currently broken
- **PR** #30491: (jacksontj) Add multi-IP support to network state @ *2016-01-21 15:51:42 UTC*
  - d8d19cf75d Merge pull request #30491 from jacksontj/2015.8
  - 82213555ca Normalize yaml spacing to 2 space
  - 3d1469b8d9 Add example of multiple addrs/ipv6addrs to docs
  - 91c8a1b4e4 Add support for multiple IP addresses per interface to rh_ip
- **PR** #30496: (anlutro) Fix KeyError when adding ignored pillars @ *2016-01-21 15:51:03 UTC*
  - 56332ca504 Merge pull request #30496 from alprs/fix-ignored_pillars_keyerror
  - bbcb783621 fix KeyError when adding ignored pillars
- **PR** #30359: (kingsquirrel152) Removes suspected copy/paste error for zmq_filtering functionailty @ *2016-01-20 18:42:42 UTC*
  - e425cbd654 Merge pull request #30359 from distil/zmq_filtering_bug_fix
  - 44bfbbf15b Removes suspected copy/paste error.
- **PR** #30448: (cournape) Fix osx scripts location @ *2016-01-20 17:59:29 UTC*
  - 13add7d142 Merge pull request #30448 from cournape/fix-osx-scripts-location
  - 3c27ab5310 BUG: fix osx .pkg script locations to match the .plist files.
  - ed9ab68d3b BUG: fix missing sudo when linking certify cert.
- **ISSUE** #22820: (VynceMontgomery) some docs missing again (cf #22720) (refs: #30457)
- **PR** #30457: (rallytime) Remove fsutils references from modules list @ *2016-01-20 16:43:50 UTC*
  - 2b7d20cee7 Merge pull request #30457 from rallytime/fix-22820
  - 3288ff104d Remove fsutils references from modules list
- **ISSUE** #30442: (ssplatt) salt-cloud linode query only lists private or public IP, not both (refs: #30453)
- **PR** #30453: (rallytime) Make sure private AND public IPs are listed for Linode driver @ *2016-01-20 16:41:51 UTC*
  - e706b71871 Merge pull request #30453 from rallytime/fix-30442
  - a1f882f4fe Make sure private AND public IPs are listed for Linode driver
- **ISSUE** #29601: (seanjnkns) pillars not merging properly with 2015.8.3 (refs: #30062)
- **PR** #30458: (rallytime) Back-port #30062 to 2015.8 @ *2016-01-20 16:40:23 UTC*
  - **PR** #30062: (seanjnkns) Remove recurse_list from pillar_source_merging_strategy and add pilla… (refs: #30609, #30458)
  - 73f372dc98 Merge pull request #30458 from rallytime/bp-30062
  - 9665d9655f Set (pillar_)merge_lists to default for PR #30062
  - 7ea4dbf478 Fix lint for PR30062
  - e44a30620b Remove recurse_list from pillar_source_merging_strategy and add pillar_merge_list (bool) instead
- **PR** #30468: (timcharper) make note of s3 role assumption in upcoming changelog @ *2016-01-20 16:28:04 UTC*

  - c3fb4006b0 Merge pull request #30468 from timcharper/2015.8
  - 721c1c871b make note of s3 role assumption in upcoming changelog
- **PR** #30470: (whiteinge) Add example of the match_dict format to accept_dict wheel function @ *2016-01-20 16:26:16 UTC*
  - c611541916 Merge pull request #30470 from whiteinge/match_dict
  - 5034e13f5d Add example of the match_dict format to accept_dict wheel function
- **ISSUE** #28017: (ThomasZhou) Using salt-cloud nova driver, raise error: SaltNova' object has no attribute '_discover_extensions' (refs: #30450)
- **PR** #30450: (gtmanfred) fix extension loading in novaclient @ *2016-01-19 21:16:32 UTC*
  - d70e6b312a Merge pull request #30450 from gtmanfred/2015.8
  - 4aa6faaf48 fix extension loading in novaclient
- **ISSUE** #30150: (rapenne-s) file.line reset permissions to 600 (refs: #30212, #30168)
- **PR** #30212: (abednarik) Fix incorrect file permissions in file.line @ *2016-01-19 21:15:48 UTC*
  - 0af5e16809 Merge pull request #30212 from abednarik/fix_file_line_permissions
  - dec15d1357 Fix incorrect file permissions in file.line
- **ISSUE** #29918: (WangWenchao) UnicodeDecodeError when saltutil.sync_modules for Windows salt-minion 2015.8.3 (refs: #29947)
- **PR** #29947: (jfindlay) fileclient: decode file list from master @ *2016-01-19 20:36:32 UTC*
  - 3c12b451fe Merge pull request #29947 from jfindlay/remote_decode
  - b9241fb6b0 state: use simple string formatting for messages
  - f6162f168c fileclient: decode file list from master
- **ISSUE** #30203: (terminalmage) Update salt.modules.git.list_worktrees() to use 'git worktree list' for Git >= 2.7.0 (refs: #30363)
- **PR** #30363: (terminalmage) Use native "list" subcommand to list git worktrees @ *2016-01-19 20:35:41 UTC*
  - 6e8b1e89a5 Merge pull request #30363 from terminalmage/issue30203
  - ee40491166 Fix redefined variable
  - 5f95851987 Use native "list" subcommand to list git worktrees
  - 911105f27c Fix incorrect missing gitdir file detection
- **PR** #30445: (justinta) Boto uses False for is_default instead of None @ *2016-01-19 18:28:18 UTC*
  - dfb9dec84f Merge pull request #30445 from jtand/boto_vpc_8
  - 00943ff1e6 Boto uses False for is_default instead of None
- **PR** #30406: (frioux) Add an example of how to use file.managed/check_cmd @ *2016-01-19 18:23:49 UTC*
  - f9b3f3f038 Merge pull request #30406 from ZipRecruiter/check-cmd-example
  - 92e0d77a9a Add an example of how to use file.managed/check_cmd
- **PR** #30424: (isbm) Check if byte strings are properly encoded in UTF-8 @ *2016-01-19 17:52:25 UTC*
  - 05ad3dcc94 Merge pull request #30424 from isbm/isbm-zypper-utf-8-errors
  - a0f263f411 Clarify the error message

  - 12f8e93247 Update documentation accordingly.
  - 1d384b6abd Add error handling to the RPM broken strings
  - cf0dad3a6c Rename keywords arguments variable to a default name.
  - 26aa801342 Check if byte strings are properly encoded in UTF-8
- **ISSUE** #30051: (joejulian) glusterfs.status fails with glusterfs 3.7 (refs: #30075)
- **PR** #30405: (justinta) Updated glusterfs.py for python2.6 compatibility. @ *2016-01-15 22:50:06 UTC*
  - **PR** #30075: (joejulian) Convert glusterfs module to use xml (refs: #30405)
  - 1bace55e45 Merge pull request #30405 from jtand/glusterfs_py26
  - a332e06c4a Fixed lint error
  - 522b4990ef Updated the rest of glusterfs.py for python2.6 compatibility
  - 971ce58cd6 updated list_peers to be python2.6 compatible
- **PR** #30396: (pass-by-value) Remove hardcoded val @ *2016-01-15 22:03:53 UTC*
  - cb1c0958bd Merge pull request #30396 from pass-by-value/remove_hardcoded_val
  - dd90b325e7 Get vm info
  - 9430ad1465 Remove hardcoded value
- **PR** #30391: (justinta) Added else statements @ *2016-01-15 19:17:55 UTC*
  - **PR** #30389: (justinta) Older versions of ipset don't support comments (refs: #30391)
  - 60737c970e Merge pull request #30391 from jtand/ipset
  - 345b056406 Fixed lint error
  - c20f9b6a87 Added else statements
- **ISSUE** #30277: (webtrekker) [salt-cloud] Error actioning machines: 'ascii' codec can't encode character u'\\xa0' in position 20 (refs: #30374)
- **PR** #30375: (rallytime) Wrap formatted log statements with six.u() in cloud/__init__.py @ *2016-01-15 18:41:55 UTC*
  - **PR** #30374: (rallytime) Wrap formatted log statements with six.u() in the VMware module (refs: #30375)
  - 6ac1f6cf54 Merge pull request #30375 from rallytime/fix-cloud-log-formatting
  - 5e7fb0c428 Wrap formatted log statements with six.u() in cloud/__init__.py
- **PR** #30384: (isbm) Bugfix: info_available does not work correctly on SLE 11 series @ *2016-01-15 18:31:57 UTC*
  - c478148b60 Merge pull request #30384 from isbm/isbm-zypper-info-avaiable-fix
  - c7bc20e865 Split information, that is compatible with the Zypper's output on SLE11.
- **PR** #30376: (pritambaral) Fix FLO_DIR path in 2015.8 @ *2016-01-15 18:25:49 UTC*
  - 9fe2df82bd Merge pull request #30376 from pritambaral/fix/flo-dir
  - 534879e79f Revert "Raet Salt broken when config moved to package directory"
- **PR** #30389: (justinta) Older versions of ipset don't support comments (refs: #30391) @ *2016-01-15 17:41:02 UTC*
  - 3ac3804ddc Merge pull request #30389 from jtand/ipset
  - fac6c3f6ae Fixed some typos from testing
  - 67d4997316 Older versions of ipset don't support comments

- **PR** #30373: (basepi) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2016-01-15 16:23:38 UTC*
    - 4cc9422bf8 Merge pull request #30373 from basepi/merge-forward-2015.8
    - 5b53bf2597 Merge remote-tracking branch 'upstream/2015.5' into merge-forward-2015.8
        - 5a923b3aa9 Merge pull request #30364 from rallytime/fix-30341
            - 79bcf151cb Add TLS version imports and add linode driver documentation notices
        - f037fd9c27 Merge pull request #30184 from rallytime/bp-30166
            - fa6b1b3022 adding split_env call to cp.hash_file to pick up saltenv in file query parameter
        - 1d8413fd2f Merge pull request #30291 from thegoodduke/for_fix_ipset
            - 62d6ccf561 ipset: fix test=true & add comment for every entry
        - 92889db638 Merge pull request #30248 from jfindlay/2015.5
            - 741f7aba31 add 2015.5.9 release notes
        - 7a329d89d7 Merge pull request #30237 from jacobhammons/man-pages-prev
            - 2431c4c5c3 Updated man page and doc conf.py copyright year to 2016
            - fe3da1c174 Updated man pages and doc version for 2015.5.9
        - 2c0b725924 Merge pull request #30207 from rallytime/rabbitmq_states_doc_fix
            - 8d48c24182 Use correct spacing in rabbitmq state examples
        - b49cf910f4 Merge pull request #30191 from jacobhammons/banner-prev
            - c3390955b0 Updated doc site banners
- **PR** #30372: (jacobhammons) Updated man pages for 2015.8.4, updated copyright to 2016 @ *2016-01-14 23:18:40 UTC*
    - a9edb194a4 Merge pull request #30372 from jacobhammons/man-pages
    - 891ddafcba Updated man pages for 2015.8.4, updated copyright to 2016
- **PR** #30370: (rallytime) Remove incomplete function @ *2016-01-14 22:49:45 UTC*
    - e77585de17 Merge pull request #30370 from rallytime/remove-incomplete-func
    - e220fa5125 Remove incomplete function
- **ISSUE** #23215: (lichtamberg) Rbenv: gem.installed not using correct ruby version if it's not default on 2015.02 (refs: #28702)
- **PR** #30366: (rallytime) Back-port #28702 to 2015.8 @ *2016-01-14 21:10:03 UTC*
    - **PR** #28702: (dnd) Pass RBENV_VERSION in env dict, and protect shlex.split (refs: #30366)
    - eb1ecd9732 Merge pull request #30366 from rallytime/bp-28702
    - 4f2274a275 Remove extra line
    - 048b13cf73 Pass RBENV_VERSION in env dict, and protect shlex.split
- **PR** #30361: (cro) Flip the sense of the test for proxymodule imports, add more fns for esxi proxy @ *2016-01-14 20:54:08 UTC*
    - 40594efc0b Merge pull request #30361 from cro/esxi-proxy2
    - 8f7490ca98 Missed return statement.
    - 389ede9e3e Lint

  - 9db34d6ffe Lint
  - b5c7a46f7a Lint
  - 1a3b1f2626 Don't use short variables
  - b80577182a Remove stub functions.
  - 58f7fc3285 Don't use single character variables.
  - e712664bcf Better comment.
  - 9e9a37d0d4 Indentation.
  - 6c9bf76e19 Revert earlier vmware change for ssl cert checking.
  - db8a281ab8 Flip the sense of the test for items (modules, etc) loaded by the proxy. Now load everything a regular minion would load, and only check to make sure __proxyenabled__ is present for proxymodules and grains
  - 00c4ef6ec2 Need a list
  - 0da7a6d6d1 Recreate the pr

- **PR** #30267: (isbm) Fix RPM issues with the date/time and add package attributes filtering @ *2016-01-14 18:00:01 UTC*
  - f4118be6e4 Merge pull request #30267 from isbm/isbm-zypper-isotimefix
  - 18281e7e0b Add "*time_t" as a separate attributes
  - 4105157cfd Add *time_t to the docs
  - 78e16a7b00 Construct RPM query dynamically
  - 6992d74806 Update documentation for the valid package attributes
  - 6710e4900d Use renamed variable (filter_attrs to attr)
  - b68e1228e9 Remove unnecessary check for the "name" key
  - e5b3e77186 Remove key transformations
  - 9ac52c9123 Add zone to the ISO from unix time
  - d352c08305 Fix lint: unused import
  - d571381f76 Update the documentation for the Zypper module
  - 5651a043e6 Update documentation for lowpkg
  - 7edb0e8f3f Replace "*_date_iso" with "*_date" and use Unix time as "*_date_time_t"
  - b2b21f877a Add epoch (note: this is empty on SUSE systems)
  - 0eebe10d9a Clarify description
  - a745d9ecdf Fix syntax for the documentation in zypper module
  - c95c2d24db Fix the documentation syntax
  - 1fb84538b1 Update documentation
  - 43ebff4dd7 Return build date in Unix ticks
  - bdaa1e4d6e Add package architecture attribute
  - ba64df4def Update documentation
  - 6e3743dce6 Incorporate lowpkg.info into info_installed

  - b72b8d5323 Fix the documentation
  - 181314b20e Add filtering per attributes feature
  - 39e70ef762 Fix ISO and Unix time of the package for RPM systems on C locale.
- **ISSUE** #30330: (JensRantil) salt.state.file.absent doesn't document recursiveness (refs: #30360)
- **PR** #30360: (jfindlay) file.remove, file.absent: mention recursive dir removal @ *2016-01-14 17:30:26 UTC*
  - b61cb7a238 Merge pull request #30360 from jfindlay/remove_doc
  - a21ccd2700 file.remove, file.absent: mention recursive dir removal
- **ISSUE** #26845: (maio) Postgres module (user_exists) doesn't work with PostgreSQL 9.5 (refs: #30221)
- **PR** #30221: (mbarrien) No rolcatupdate for user_exist in Postgres>=9.5 #26845 @ *2016-01-14 16:52:49 UTC*
  - ba8d128025 Merge pull request #30221 from mbarrien/postgres-9.5
  - a8f2bc7998 No rolcatupdate for user_exist in Postgres>=9.5 #26845
- **PR** #30358: (terminalmage) Add libgit2 version to versions-report @ *2016-01-14 16:37:28 UTC*
  - 4787c2c9ad Merge pull request #30358 from terminalmage/libgit2-version
  - 89fe571791 Add libgit2 version to versions-report
- **PR** #30346: (pass-by-value) Prevent orphaned volumes @ *2016-01-14 16:37:08 UTC*
  - af2ddfd31c Merge pull request #30346 from pass-by-value/aws_vols_attach
  - 19fce03ee2 Prevent orphaned volumes
- **PR** #30349: (rallytime) Back-port #30347 to 2015.8 @ *2016-01-14 16:26:37 UTC*
  - **PR** #30347: (rallytime) Merge #30231 with updates to dependency documentation (refs: #30349)
  - **PR** #30231: (nmadhok) Fix issue where pyVmomi 6.0.0 raises SSL Error for systems using Python2.7+ (refs: #30347)
  - bccb8f3b5b Merge pull request #30349 from rallytime/bp-30347
  - df70afdaa3 Merge #30231 with updates to dependency documentation
  - a7c2ad5505 Fix issue where pyVmomi 6.0.0 raises SSL Error for systems using Python2.7+
- **PR** #30354: (anlutro) Make sure all ignore_missing SLSes are catched @ *2016-01-14 16:24:19 UTC*
  - **PR** #19429: (ryan-lane) Add new ignore_missing option to pillar top (refs: #30354)
  - 7ee61f0d62 Merge pull request #30354 from alprs/fix-pillar_ignore_missing
  - 2f662bbc8d make sure *all* ignore_missing slses are catched
- **PR** #30356: (nmadhok) Adding code author @ *2016-01-14 16:23:08 UTC*
  - 4bdade6010 Merge pull request #30356 from nmadhok/patch-1
  - 581e4f5dc7 Adding code author
- **PR** #30340: (justinta) Updated seed_test.py for changes made to seed module @ *2016-01-13 22:50:34 UTC*
  - d5b8776355 Merge pull request #30340 from jtand/seed_test_fix
  - ee764ee952 Updated seed_test.py for changes made to seed module
- **ISSUE** #26478: (rasathus) nested upstart services are not supported (refs: #26511)
- **PR** #30339: (jfindlay) Backport #26511 @ *2016-01-13 22:35:17 UTC*

  - **PR** #26511: (rasathus) Adds support for nested upstart scripts in the form of subfolder/serv… (refs: #30339)
  - 3bbed62d07 Merge pull request #30339 from jfindlay/bp-26511
  - 89d9cd5e38 Adds support for nested upstart scripts in the form of subfolder/service. This is implemented via an os.walk through the /etc/init folder, rather than the previous glob for *.conf method.
- **ISSUE** #28339: (boboli) salt-call state.highstate fails with ZMQError when minion has no id set in /etc/salt/minion (refs: #28423, #28431)
- **PR** #30343: (rallytime) Fix 2015.8 from incomplete back-port @ *2016-01-13 21:56:26 UTC*
  - **PR** #30187: (rallytime) Back-port #27606 to 2015.8 (refs: #30343)
  - **PR** #28431: (plastikos) Use a broader test for unset "id" (refs: #30343)
  - **PR** #28423: (cachedout) Fix issue with empty str as default minion id (refs: #28431)
  - **PR** #28189: (plastikos) Always get default option settings from salt.config (refs: #30343, #28431)
  - **PR** #28131: (cachedout) Set a fallback HWM (refs: #30343)
  - **PR** #27606: (plastikos) RFC: Add additional ZMQ tuning parameters necessary for 1k+ minions per master [WIP] (refs: #30343, #30187)
  - 6079a96e6e Merge pull request #30343 from rallytime/fix-2015.8
  - 5eef9d5067 Use a broader test for unset "id"
  - 460a3c98cc Additional corrections to use option defaults directly from salt.config
  - 4e53ef0bf6 Always get default option settings from salt.config
  - 94ee6f88af Set a fallback HWM
- **PR** #30342: (eliasp) Correct whitespace placement in error message @ *2016-01-13 21:32:26 UTC*
  - 7276d808ff Merge pull request #30342 from eliasp/2015.8-log-message-format
  - 8e37e36ac7 Correct whitespace placement in error message
- **ISSUE** #30250: (mbarrien) npm.bootstrap state runs even when test=True (refs: #30257)
- **PR** #30308: (rallytime) Back-port #30257 to 2015.8 @ *2016-01-13 19:20:13 UTC*
  - **PR** #30257: (abednarik) Add test in npm state. (refs: #30308)
  - 10b5728f84 Merge pull request #30308 from rallytime/bp-30257
  - 0b0d73756e Fix typos in nmp module.
  - deeeb71dda Add test in npm state.
- **PR** #30187: (rallytime) Back-port #27606 to 2015.8 (refs: #30343) @ *2016-01-13 19:03:11 UTC*
  - **PR** #27606: (plastikos) RFC: Add additional ZMQ tuning parameters necessary for 1k+ minions per master [WIP] (refs: #30343, #30187)
  - afa61c03db Merge pull request #30187 from rallytime/bp-27606
  - 8ef6d6c6fd Add additional ZMQ tuning parameters necessary for 1,000+ minions per server. Start collecting tuning parameters together in the master config file.
- **PR** #30223: (serge-p) adding support for DragonFly BSD @ *2016-01-13 18:24:29 UTC*
  - 7e89a460e4 Merge pull request #30223 from serge-p/patch-11
  - ec798acbcd Update pkgng.py

  - 45206dfe3d adding support for DragonFly BSD
- **ISSUE** #28396: (ymote) salt-cloud parallel provisioning (-P option) failed on 2015.8.1 (refs: #30238)
- **ISSUE** #23824: (kiorky) salt.crypt broken in develop (refs: #23825)
- **PR** #30238: (rallytime) Reinit crypto before calling RSA.generate when generating keys. @ *2016-01-13 18:22:11 UTC*
  - **PR** #23825: (kiorky) Fix crypto (refs: #30238)
  - 5a8da62008 Merge pull request #30238 from rallytime/fix-28396
  - 41d9df45bb Reinit crypto before calling RSA.generate when generating keys.
- **ISSUE** #24237: (Grokzen) Minion schedule return data missing some fields (refs: #30246)
- **PR** #30246: (dmacvicar) Add missing return data to scheduled jobs (#24237) @ *2016-01-13 17:51:49 UTC*
  - 15707e0ac8 Merge pull request #30246 from dmacvicar/dmacvicar-2015.8-24237
  - c462139dbb lint: E8713(test-for-membership-should-be-not-in)
  - 5a1b2ca486 include the 'success' field in scheduled jobs return data (part of #24237)
  - f72a4ca42d add retcode to scheduled jobs return data (part of #24237)
- **PR** #30292: (thegoodduke) ipset: fix test=true & add comment for every entry @ *2016-01-13 17:49:16 UTC*
  - **PR** #30170: (thegoodduke) ipset: fix comment and test (refs: #30291, #30292)
  - 8706720148 Merge pull request #30292 from thegoodduke/fix_ipset
  - 49d70bff16 ipset: fix test=true & add comment for every entry
- **ISSUE** #30240: (snw1968) firewalld inconsistent permanent option used for services but not ports - other options required (refs: #30275)
- **PR** #30275: (abednarik) Add permanent argument in firewalld. @ *2016-01-13 17:44:43 UTC*
  - ea607675f5 Merge pull request #30275 from abednarik/fix_firewalld_ports_permanent
  - e3d4bf51da Add permanent argument in firewalld.
- **PR** #30328: (cachedout) Fix file test @ *2016-01-13 17:42:22 UTC*
  - f02db44757 Merge pull request #30328 from cachedout/fix_file_test
  - dcfba51556 Lint
  - b9921128af Kill pointless tests
  - 63c157d0a3 Fix test_managed
- **PR** #30310: (pass-by-value) Empty bucket fix @ *2016-01-13 17:30:45 UTC*
  - edd94aea2c Merge pull request #30310 from pass-by-value/empty_bucket_fix
  - aef5a8898c Add fix for else code path
  - 9398c44945 Check and report empty S3 bucket
- **PR** #30211: (techhat) Execute choot on the correct path @ *2016-01-13 16:53:40 UTC*
  - f23f0f30d4 Merge pull request #30211 from techhat/tmppath
  - 11ac2ff0bf Revert "We're putting the keys directly in place; -c isn't used"
  - e75b48f5ff We're putting the keys directly in place; -c isn't used
  - 5d7a0f6d81 Execute choot on the correct path

- **ISSUE** #30286: (tkunicki) salt-cloud ec2 spot requests fail with userdata_file in config or profile (refs: #30304)
- **PR** #30309: (rallytime) Back-port #30304 to 2015.8 @ *2016-01-13 16:41:53 UTC*
  - **PR** #30304: (tkunicki) add spot_prefix to UserData param (refs: #30309)
  - 5154c71127 Merge pull request #30309 from rallytime/bp-30304
  - 4a8cc87b47 add spot_prefix to UserData param
- **PR** #30278: (nmadhok) If datacenter is specified in the config, then look for managed objects under it @ *2016-01-13 15:29:36 UTC*
  - 1624d6cebd Merge pull request #30278 from nmadhok/2015.8-samename-objects-fix
  - b0e86afa00 get_mor_by_property needs container_ref to be a positional parameter
  - 56dfc63f91 If datacenter is specified, start all searches under datacenter
  - fcf77b738e If datacenter is specified then look under it instead of looking under inventory root folder
- **PR** #30305: (jacobhammons) Changed examples to use the "example.com" domain instead of "mycompan… @ *2016-01-12 20:42:10 UTC*
  - fc9304f7f8 Merge pull request #30305 from jacobhammons/example-domain
  - 53d17f1f85 Changed examples to use the "example.com" domain instead of "mycompany.com" or "company.com"
- **PR** #30249: (mpreziuso) Fixes performance and timeout issues on win_pkg.install @ *2016-01-12 20:14:54 UTC*
  - 3bd02a898f Merge pull request #30249 from mpreziuso/patch-2
  - d6e6e10534 Fixes lint issues
  - 3251424838 Fixes performance and timeout issues on win_pkg.install
- **PR** #30217: (pass-by-value) Make sure cloud actions can be called via salt run @ *2016-01-12 20:11:13 UTC*
  - 461a741e14 Merge pull request #30217 from pass-by-value/cloud_actions_dispatch
  - 1f68ce05bc Fix pylint error
  - d5b1b60b99 Add CLI Example
  - 5264449fdb Make sure cloud actions can be called via salt run
- **ISSUE** #9569: (clearclaw) How can a binary file, such as a license key, be distributed via Pillar? (refs: #30268)
- **PR** #30268: (terminalmage) Optimize file_tree ext_pillar and update file.managed to allow for binary contents @ *2016-01-12 20:09:19 UTC*
  - 4a6b53f329 Merge pull request #30268 from terminalmage/issue9569
  - 724b2f36ce Add file_tree/file.managed/contents_pillar example to FAQ
  - 854c7d9978 Remove old FAQ item referencing gitfs bug in 0.16.x
  - e9a6d709f9 salt.states.file.managed: Allow for binary contents
  - 1ba448b619 salt.pillar.file_tree: Optimizations, deprecate raw_data
  - 650cc0af5c salt.modules.file: Improve docstrings
- **ISSUE** #29078: (Reiner030) boto_secgroup didn't work as expected in Debian Jessie (refs: #30155)
- **PR** #30245: (rallytime) Boto secgroup/iam_role: Add note stating us-east-1 is default region @ *2016-01-12 20:04:31 UTC*

  - **PR** #30155: (rallytime) Update boto_secgroup and boto_iam_role docs to only use region OR profile (refs: #30245)
  - dbe7bcdc9a Merge pull request #30245 from rallytime/botosecgroup-docs
  - 406a138f76 Boto secgroup/iam_role: Add note stating us-east-1 is default region
- **PR** #30299: (rallytime) ESXi Proxy minions states are located at salt.states.esxi, not vsphere. @ *2016-01-12 20:03:31 UTC*
  - 6b183778f1 Merge pull request #30299 from rallytime/esxi-proxy-doc-fix
  - db68fc48a8 Fix CLI Example syntax
  - 1cb9f29798 ESXi Proxy minions states are located at salt.states.esxi, not vsphere.
- **PR** #30202: (opdude) Fixed the periodic call to beacons @ *2016-01-12 19:58:44 UTC*
  - 903289d3fb Merge pull request #30202 from Unity-Technologies/hotfix/beacon_periodic
  - ea7a86fa7d Fixed the periodic call to beacons
- **PR** #30303: (jacobhammons) Changed notes to indicate that functions are matched using regular ex… @ *2016-01-12 19:15:16 UTC*
  - 48d2bd9e78 Merge pull request #30303 from jacobhammons/pcre-match
  - e5079ab4c9 Changed notes to indicate that functions are matched using regular expressions instead of minions
- **ISSUE** #29684: (snarfmonkey) Upgrade from 2015.8.1 to 2015.8.3 via apt for Ubuntu 14.04 causes Dulwich-backed gitfs to stop working (refs: #30284)
- **PR** #30284: (terminalmage) salt.utils.gitfs: Fix Dulwich env detection and submodule handling @ *2016-01-12 19:11:36 UTC*
  - 675ac4b43f Merge pull request #30284 from terminalmage/issue29684
  - a746014f7e salt.utils.gitfs: Fix Dulwich env detection and submodule handling
- **PR** #30280: (jfindlay) add state mocking to release notes @ *2016-01-12 19:10:40 UTC*
  - 8f65e822d7 Merge pull request #30280 from jfindlay/state_mock_doc
  - 22c1129f02 modules.state.sls,highstate: mock versionadded
  - 934de30939 add state mock to 2015.8.4 release notes
- **ISSUE** #30117: (MadsRC) Service beacons fails with Stacktraces (refs: #30121)
- **PR** #30273: (rallytime) Back-port #30121 to 2015.8 @ *2016-01-12 19:10:16 UTC*
  - **PR** #30121: (MadsRC) Patch for issue #30117 (refs: #30273)
  - c9ade42d10 Merge pull request #30273 from rallytime/bp-30121
  - c8c30f2105 I fail at linting… Fixed my uppercase/lowercase problem
  - 0877b33026 Fixed some linting issues
  - 8ec36497a1 Added note about systemd and uncleanshutdown. Also fixed line lenght of comments to max 80 characters as per PEP0008
  - a50428d02c On an unclean shutdown, if oncleanshutdown is given a path, an keyy:value of shutdown:unclean is added to the returned data. The documentation states that the key should be 'uncleanshutdown' and that the value should either be True or False. This is fixed in the code

  - 51b57f1820 Fixed issue number #30117 - When no parameters are given to a service, the service object is of type None and thus isn't iterable. This is contrary to the documentation which states that there are default values. Default values added as False

- **PR** #30301: (cachedout) Accept whatever comes into hightstate mock for state tests @ *2016-01-12 18:33:14 UTC*
  - 3a5a84a790 Merge pull request #30301 from cachedout/fix_state_tests
  - 2c62b464b1 Accept whatever comes into hightstate mock for state tests
- **ISSUE** #28586: (zmalone) file.append does not differentiate between tabs and spaces (refs: #30156)
- **PR** #30282: (cachedout) Fix file.append logic @ *2016-01-12 18:27:30 UTC*
  - **PR** #30156: (abednarik) Add option in file.append to ignore_whitespace. (refs: #30282)
  - 8438d19815 Merge pull request #30282 from cachedout/fix_30156
  - 3f633ff15e Lint
  - 99dd11dec2 Remove debugging
  - 35ef585c54 Fix logic error in file.append
- **PR** #30289: (cro) Fix problems with targeting proxies by grains @ *2016-01-12 18:16:57 UTC*
  - 530c9ec6ec Merge pull request #30289 from cro/proxy_grains_fix
  - 8362d76440 Add comments.
  - 4e50962642 Merge branch 'proxy_grains_fix' of github.com:cro/salt into proxy_grains_fix
    - 61bb6a9a14 Lint.
    - 7c35333509 Force a grains sync after we load the proxy's grains.
    - 2855ba7da5 Disallow non-proxyenabled modules and grains
  - 8fd8f3beb7 Lint.
  - 144fea02e5 Force a grains sync after we load the proxy's grains.
  - 5ecf85017b Disallow non-proxyenabled modules and grains
- **PR** #30293: (cro) Ensure we don't log stuff we shouldn't @ *2016-01-12 18:04:25 UTC*
  - 75b83453cf Merge pull request #30293 from cro/proxy_log_cleanup
  - b358fe370c Merge remote-tracking branch 'origin/proxy_log_cleanup' into proxy_log_cleanup
    - c9a5680427 Add unused 'output_loglevel' kwarg. This is here for when we alias cmd.run_all directly to _run_all_quiet in certain chicken-and-egg situations where modules need to work both before and after the __salt__ dictionary is populated (cf dracr.py).
    - 8c46de12e4 Ensure we don't log stuff we shouldn't.
  - 3267d92216 Add unused 'output_loglevel' kwarg. This is here for when we alias cmd.run_all directly to _run_all_quiet in certain chicken-and-egg situations where modules need to work both before and after the __salt__ dictionary is populated (cf dracr.py).
  - 6a86bdc6da Ensure we don't log stuff we shouldn't.
- **PR** #30279: (cachedout) Allow modules to be packed into boto utils @ *2016-01-12 16:53:54 UTC*
  - 46681658e0 Merge pull request #30279 from cachedout/boto_pack
  - 11d27ba694 Mock config module in utils test
  - 62a1818287 Lint

  - cf440036dd Remove unused import
  - 36d55ea0ad Allow modules to be packed into boto utils
- **ISSUE** #29951: (Reiner030) boto_ec2 params needed (refs: #30186)
- **PR** #30186: (rallytime) Update CLI Examples in boto_ec2 module to reflect correct arg/kwarg positioning @ *2016-01-08 19:00:45 UTC*
  - 54b9641330 Merge pull request #30186 from rallytime/fix-29951
  - a943b505cc Update CLI Examples in boto_ec2 module to reflect correct arg/kwarg positioning
- **ISSUE** #28586: (zmalone) file.append does not differentiate between tabs and spaces (refs: #30156)
- **PR** #30156: (abednarik) Add option in file.append to ignore_whitespace. (refs: #30282) @ *2016-01-08 16:07:23 UTC*
  - 1256fd11e1 Merge pull request #30156 from abednarik/ignore_whitespace_file_append
  - af68086e5c Add option in file.append to ignore_whitespace.
- **PR** #30189: (rallytime) Back-port #30185 to 2015.8 @ *2016-01-07 23:32:05 UTC*
  - **PR** #30185: (cachedout) Fix #30118 (refs: #30189)
  - **PR** #30118: (thatch45) State mock (refs: #30185, #30189)
  - ad7522c98d Merge pull request #30189 from rallytime/bp-30185
  - 70681bf03b Fix for mock state PR #30118
  - f9480f66d8 change arg to mocked to try test suite collision fix
  - 2fbdcda703 fix some typos
  - 6f757b8c81 Add Mock to state.sls
  - fb0cbd185e fix issue where the name may be in 2 places
  - 5f0326e521 Start on the state mock system
- **ISSUE** #9319: (gravyboat) Update Reactor docs with an example using salt-cloud from the commandline. (refs: #30215)
- **ISSUE** #8146: (basepi) Make implications of extra accepted keys on timeouts more obvious (refs: #30215)
- **ISSUE** #6853: (Psycojoker) Salt formulas should be way more visible in the documentation (refs: #30215)
- **ISSUE** #4381: (mlister2006) peer_run: glob, pcre matching. Better docs (refs: #30215)
- **ISSUE** #2229: (alekibango) how to debug zeromq problem with hanging salt communication? (refs: #30215)
- **ISSUE** #15042: (cvrebert) percent signs not escaped in cron commands (refs: #30215)
- **ISSUE** #14946: (ryan-lane) reload_modules not documented in global state arguments documentation (refs: #30215)
- **ISSUE** #13777: (gravyboat) Update top module docs with more concise examples (refs: #30215)
- **ISSUE** #13036: (tminn) salstack tomcat module (refs: #30215)
- **PR** #30215: (jacobhammons) Assorted doc bug fixes @ *2016-01-07 21:53:27 UTC*
  - 8f30f7045a Merge pull request #30215 from jacobhammons/doc-issues
  - 44ce704206 Updated *zmq_monitor* docs
  - 0d2111d397 Assorted doc bug fixes

- **ISSUE** #30204: (anlutro) salt can't find local cache return file (refs: #30206)
- **PR** #30206: (cachedout) Revert "Fix incorrect file permissions in file.line" @ *2016-01-07 17:55:48 UTC*
  - 2000800915 Merge pull request #30206 from cachedout/revert_30168
  - ee786293e7 Revert "Fix incorrect file permissions in file.line"
- **PR** #30190: (jacobhammons) Updated doc site banners @ *2016-01-06 22:37:34 UTC*
  - 5632ccb796 Merge pull request #30190 from jacobhammons/banners
  - 266023baf1 Updated doc site banners
- **ISSUE** #30171: (jamusj) Python 2.7 dependency in x509.py (refs: #30180)
- **PR** #30180: (jfindlay) modules.x509._dec2hex: add fmt index for 2.6 compat @ *2016-01-06 19:48:50 UTC*
  - 9a83247992 Merge pull request #30180 from jfindlay/2.7x509
  - 907469d04a modules.x509._dec2hex: add fmt index for 2.6 compat
- **PR** #30179: (terminalmage) Backport #26962 to 2015.8 branch @ *2016-01-06 19:48:30 UTC*
  - **PR** #26962: (ctrlrsf) Add –state-verbose command line option to salt cmd (refs: #30179)
  - 6516d5b5d0 Merge pull request #30179 from terminalmage/bp-26962
  - 08f2021f52 Fix pylint warnings: unnecessary parens after if keyword
  - a2ec721661 Add –state-verbose command line option to salt cmd
- **ISSUE** #29654: (schaarsc) ssh_auth should report missing source (refs: #29693)
- **PR** #29693: (abednarik) Handle missing source file in ssh_auth. @ *2016-01-06 17:13:06 UTC*
  - 27df7276bc Merge pull request #29693 from abednarik/handle_missing_source_in_ssh_auth
  - fc024e3cf4 Handle missing source file in ssh_auth.
- **ISSUE** #29078: (Reiner030) boto_secgroup didn't work as expected in Debian Jessie (refs: #30155)
- **PR** #30155: (rallytime) Update boto_secgroup and boto_iam_role docs to only use region OR profile (refs: #30245) @ *2016-01-06 17:09:50 UTC*
  - f9863dd9fb Merge pull request #30155 from rallytime/boto-secgroup-docfix
  - f0381a955f Update boto_secgroup and boto_iam_role docs to only use region OR profile.
- **ISSUE** #29905: (Reiner030) pillar referencing for boto profiles seems not completely working right / docu missing (refs: #30158)
- **PR** #30158: (rallytime) Move _option(value) calls to __salt__['config.option'] in boto utils @ *2016-01-06 16:35:59 UTC*
  - e36e8e2e73 Merge pull request #30158 from rallytime/fix-29905
  - 3321c5d408 Move _option(value) calls to __salt__['config.option'] in boto utils
- **ISSUE** #29770: (Ch3LL) disk.usage does not work on AIX (refs: #30160)
- **PR** #30160: (dmurphy18) Fix parsing disk usage for line with no number and AIX values in Kilos @ *2016-01-06 16:34:45 UTC*
  - ec009a6812 Merge pull request #30160 from saltstack/aix_dskusage
  - 8450df0483 Fix parsing disk usage for line with no number and AIX values in Kilos
- **ISSUE** #29919: (abcfy2) State `grains.append` cannot append to a non-exist grain name. (refs: #30162)

- **PR** #30162: (rallytime) Update list_present and append grains state function docs to be more clear. @ *2016-01-06 16:33:25 UTC*
  - f808ffbbbd Merge pull request #30162 from rallytime/fix-29919
  - 9bbd129c60 Update list_present and append grains state function docs to be more clear
- **ISSUE** #28923: (aabognah) passing argument with '=no' to file.line (refs: #30163)
- **PR** #30163: (rallytime) Add warning about using "=" in file.line function @ *2016-01-06 16:32:39 UTC*
  - 83245930a6 Merge pull request #30163 from rallytime/fix-28923
  - 0e4f91fca2 Add warning about using "=" in file.line function
- **PR** #30164: (basepi) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2016-01-06 16:28:59 UTC*
  - 106efd258a Merge pull request #30164 from basepi/merge-forward-2015.8
  - d73a7d6c4d Merge remote-tracking branch 'upstream/2015.5' into merge-forward-2015.8
    - 9363d6f5b6 Merge pull request #30125 from abednarik/update_user_home
      - 56544a77f6 Update user home event when createhome is set to False
    - 1a5d585d91 Merge pull request #30127 from jsutton/clarify-documenation-for-random_master
      - 01dbf385ef Adding random_master to reference and updating master_shuffle. Adding master_shuffle to the minion example config file as it is needed for multi-master PKI.
    - 28b1bbbe77 Merge pull request #30110 from markckimball/fix-verify_ssl-in-joyent-cloud
      - e1c08cb269 Fixed flag sent to salt.utils.http in order for verify_ssl to work appropriately.
    - 040412b0b1 Merge pull request #30093 from zmalone/pillar-notes
      - cfbfd58afe Noting that file_roots and "state tree" should both be avoided, because in some environments, the actual states show up another level down. Adding notes about why this is undesirable.
    - 25edefc93a Merge pull request #30097 from cachedout/note_on_password_process_list
      - 58aec884ef Note concern about cleartext password in docs for shadow.gen_password
    - 6b1c3a6bf2 Merge pull request #30089 from mpreziuso/patch-1
      - 50533add40 Fixes terminology and adds more accurate details about the algorithms
    - 200d09385d Merge pull request #30086 from cachedout/issue_29921
      - 8c29e2dd6a Document that gitfs needs recent libs
    - 404414bf57 Merge pull request #30070 from cachedout/issue_27835
      - 60431e342a Add documentation on debugging salt-ssh
- **ISSUE** #30150: (rapenne-s) file.line reset permissions to 600 (refs: #30212, #30168)
- **PR** #30168: (abednarik) Fix incorrect file permissions in file.line @ *2016-01-06 16:25:08 UTC*
  - e5d87a02b9 Merge pull request #30168 from abednarik/2015.8
  - 79daa25a15 Fix incorrect file permissions in file.line
- **PR** #30154: (Oro) Fix file serialize on windows @ *2016-01-05 18:08:40 UTC*
  - bed38d1a65 Merge pull request #30154 from Oro/fix-file-serialize-windows
  - 071a675f8a Fix file serialize on windows

- **PR** #30144: (rallytime) Added generic ESXCLI command ability to ESXi Proxy Minion @ *2016-01-05 16:23:38 UTC*
    - 7d51d8bb46 Merge pull request #30144 from rallytime/vsphere-esxcli-cmd
    - 2f9ec5db96 Added generic ESXCLI command ability to ESXi Proxy Minion
- **ISSUE** #29994: (adithep) dockerng.push should not auto tag :latest (refs: #30142)
- **ISSUE** #29993: (adithep) Dockerng as a whole is not compatible with v2 registries. (refs: #30142)
- **PR** #30142: (terminalmage) Fix dockerng.push, and allow for multiple images @ *2016-01-04 22:53:50 UTC*
    - 1a21b3d46b Merge pull request #30142 from terminalmage/issue29994
    - 66698986e4 Fix dockerng.push, and allow for multiple images
- **ISSUE** #30051: (joejulian) glusterfs.status fails with glusterfs 3.7 (refs: #30075)
- **PR** #30075: (joejulian) Convert glusterfs module to use xml (refs: #30405) @ *2016-01-04 20:33:58 UTC*
    - 5419699bd2 Merge pull request #30075 from iodatacenters/2015.8_gluster_usexml
    - 01a8e7ee10 Convert glusterfs module to use xml
- **PR** #30129: (optix2000) Clean up _uptodate() in git state @ *2016-01-04 20:23:18 UTC*
    - a6d94358ed Merge pull request #30129 from optix2000/2015.8
    - c68ea6332a No point to recast comments to a string. _uptodate() should only accept strings for comments.
    - 6c5bac4909 Properly fix concat list issue in git state.
- **ISSUE** #28814: (peter-slovak) The "virtual" grain detection with virt-what on LXC incorrectly yields "physical" (refs: #29589)
- **PR** #30139: (rallytime) Back-port #29589 to 2015.8 @ *2016-01-04 20:22:47 UTC*
    - **PR** #29589: (abednarik) Added lxc in virt-what list. (refs: #30139)
    - 68dfa0f5de Merge pull request #30139 from rallytime/bp-29589
    - 2c73990ff2 Added lxc in virt-what list.
- **ISSUE** #29833: (iMilnb) salt minion won't start: Non valid IP address match on BSD alias format (refs: #30124)
- **PR** #30124: (abednarik) Update regex to detect ip alias in OpenBSD. @ *2016-01-04 19:48:28 UTC*
    - dd8d3e6f6b Merge pull request #30124 from abednarik/fix_openbsd_ip_alias
    - 595a12977d Update regex to detect ip alias in OpenBSD.
- **PR** #30133: (stanislavb) Fix typo in gpgkey URL @ *2016-01-04 19:29:57 UTC*
    - c3014be84b Merge pull request #30133 from stanislavb/fix-gpg-key-url-typo
    - d81f6f7206 Fix typo in gpgkey URL
- **ISSUE** #29912: (rterbush) s3 ext_pillar fails if key and keyid are not provided (refs: #30126)
- **PR** #30126: (stanislavb) Log S3 API error message @ *2016-01-04 19:22:39 UTC*
    - c06671a259 Merge pull request #30126 from stanislavb/2015.8
    - 8c4a101c8f Log S3 API error message
- **PR** #30128: (oeuftete) Log retryable transport errors as warnings @ *2016-01-04 19:15:31 UTC*
    - aeec21ea65 Merge pull request #30128 from oeuftete/fileclient-attempt-error-to-warning

  - a5d99b13e1 Log retryable transport errors as warnings
- **ISSUE** #28171: (srkunze) cron.rm cannot remove @special entries (refs: #30096)
- **PR** #30096: (cachedout) Add rm_special to crontab module @ *2016-01-01 00:56:08 UTC*
  - 941bcaed07 Merge pull request #30096 from cachedout/issue_28171
  - 259a0582ac Add docs
  - ad9424820e Add rm_special to crontab module
- **PR** #30106: (techhat) Ensure last dir @ *2016-01-01 00:52:54 UTC*
  - cb0f80831f Merge pull request #30106 from techhat/seeddirs
  - 01d1a49937 Ensure last dir
- **PR** #30101: (gtmanfred) fix bug where nova driver exits with no adminPass @ *2015-12-31 13:45:16 UTC*
  - 6bc968db9a Merge pull request #30101 from gtmanfred/2015.8
  - 1b98f7af38 fix bug where nova driver exits with no adminPass
- **PR** #30090: (techhat) Add argument to isdir() @ *2015-12-30 22:41:02 UTC*
  - 3652dbae76 Merge pull request #30090 from techhat/seeddirs
  - f7c7d9c7c2 Add lstrip
  - c70257163b Add argument to isdir()
- **PR** #30094: (rallytime) Fix doc formatting for cloud.create example in module.py state @ *2015-12-30 22:40:24 UTC*
  - a12bda4b30 Merge pull request #30094 from rallytime/module_state_doc_fix
  - 8fbee322b9 Fix doc formatting for cloud.create example in module.py state
- **PR** #30095: (rallytime) Add the list_nodes_select function to linode driver @ *2015-12-30 21:06:58 UTC*
  - d7f46b5438 Merge pull request #30095 from rallytime/select_query_linode
  - 4731d9442e Add the list_nodes_select function to linode driver
- **ISSUE** #28763: (cybacolt) grain saltversioninfo not returning values by index (refs: #30082)
- **PR** #30082: (abednarik) Fixed saltversioninfo grain return @ *2015-12-30 18:23:17 UTC*
  - dce64c0868 Merge pull request #30082 from abednarik/fix_grain_saltversion_index
  - 882e9ac9ed Fixed saltversioninfo grain return.
- **PR** #30084: (rallytime) Back-port #29987 to 2015.8 @ *2015-12-30 18:19:09 UTC*
  - **PR** #29987: (pass-by-value) Make sure output file works for salt cloud (refs: #30084)
  - 5602b8833e Merge pull request #30084 from rallytime/bp-29987
  - 16e1df90e9 Make sure output file works for salt cloud
- **PR** #30071: (rallytime) Merge branch '2015.5' into '2015.8' @ *2015-12-29 23:18:00 UTC*
  - 654cab0314 Merge pull request #30071 from rallytime/merge-forward-2015.8
  - 394d7548c5 Additional spelling fixes for boto_vpc module
  - f7e58a241c Merge branch '2015.5' into '2015.8'
    - 84db12212d Merge pull request #30059 from mpreziuso/patch-1

        - 1cb1c2da07 Fixes wrong function scope
      - 1c6c9b1a06 Merge pull request #30025 from jtand/boto_tests
        - e706642152 Skipping some Boto tests until resolved moto issue
      - 0f91021c59 Merge pull request #29949 from aletourneau/2015.5
        - cf855fe262 Fixed trailing white spaces
        - 864801e002 fixed version
        - 041d9346c4 Enhanced netscaler docstring
      - 229d3eb60b Merge pull request #29941 from cachedout/boto_spelling
        - b11bfd07b8 Fix spelling error in boto_vpc
      - 69c5ada636 Merge pull request #29908 from cachedout/issue_29880
        - 4cd77b4118 Allow kwargs to be passed to pacman provide for update func
      - ad0de4d563 Merge pull request #29909 from abednarik/freebsd_pkgng_non_interactive_fix
        - 8ac213001a FreeBSD pkgng fix for non-interactive install.
- **PR** #30067: (ryan-lane) Pass in kwargs to boto_secgroup.convert_to_group_ids explicitly @ *2015-12-29 23:04:33 UTC*
  - 1bf9853808 Merge pull request #30067 from lyft/boto-elb-stable-fix
  - ae22edb1b4 Pass in kwargs to boto_secgroup.convert_to_group_ids explicitly
- **PR** #30069: (techhat) Ensure that pki_dir exists @ *2015-12-29 23:03:23 UTC*
  - 0a37c4de1a Merge pull request #30069 from techhat/seeddirs
  - 0f05d49bde Ensure that pki_dir exists
- **ISSUE** #30045: (AkhterAli) salt-cloud make syndic not possible. (refs: #30064)
- **PR** #30064: (rallytime) Add Syndic documentation to miscellaneous Salt Cloud config options @ *2015-12-29 20:15:45 UTC*
  - 896655602e Merge pull request #30064 from rallytime/fix-30045
  - 6176f383e5 Spelling fixes
  - 83c05729d6 Add Syndic documentation to miscellaneous Salt Cloud config options
- **PR** #30049: (rallytime) Add some more unit tests for the vsphere execution module @ *2015-12-29 17:07:41 UTC*
  - bad6daca93 Merge pull request #30049 from rallytime/esxi-unit-tests
  - 1a83147986 Remove unnecessary import block
  - 695107ae6e Add some more unit tests for the vsphere execution module
- **PR** #30060: (rallytime) Back-port #27104 to 2015.8 @ *2015-12-29 17:06:58 UTC*
  - **PR** #27104: (hexedpackets) Remove only the file extension when checking missing cached nodes. (refs: #30060)
  - cedee772d7 Merge pull request #30060 from rallytime/bp-27104
  - f0566c4b8f Remove only the file extension on cached node files instead of replacing every '.p' substring.
- **ISSUE** #28540: (whiteinge) The rest_cherrypy automodule docs are hard to digest (refs: #30048)
- **PR** #30048: (jacobhammons) Remove internal APIs from rest_cherrypy docs. @ *2015-12-28 23:24:13 UTC*

  - 87667e2de6 Merge pull request #30048 from jacobhammons/28540
  - a04cebd48c Remove internal APIs from rest_cherrypy docs. Refs #28540
- **ISSUE** #29960: (anlutro) Circular import in salt.utils.jinja (refs: #30043)
- **PR** #30043: (rallytime) Be explicit about importing from salt.utils.jinja to avoid circular imports @ *2015-12-28 21:35:18 UTC*
  - 3c63527313 Merge pull request #30043 from rallytime/fix-29960
  - a157c78bc8 Be explicit about importing from salt.utils.jinja to avoid circular imports
- **PR** #30038: (rallytime) Back-port #30017 to 2015.8 @ *2015-12-28 20:41:45 UTC*
  - **PR** #30017: (anlutro) Change how alternatives states check for installed (refs: #30038)
  - 6cdca314c7 Merge pull request #30038 from rallytime/bp-30017
  - aab35b883e Add versionadded directive for new check_exists function.
  - ca290ec3e1 change how alternatives states check for installed
- **PR** #30036: (rallytime) Back-port #29995 to 2015.8 @ *2015-12-28 20:41:04 UTC*
  - **PR** #29995: (ruxandraburtica) Location from profiles not correctly set (refs: #30036)
  - c846e7bc86 Merge pull request #30036 from rallytime/bp-29995
  - 129a6d7b9f Added vm_ to the get_location query.
  - af8d01a367 Updated ec2 file to correctly propagate location.
- **PR** #30035: (rallytime) Back-port #29895 to 2015.8 @ *2015-12-28 20:20:58 UTC*
  - **PR** #29895: (pass-by-value) Do not SSH to the instance if deploy is False (refs: #30035)
  - 27b0bd2c34 Merge pull request #30035 from rallytime/bp-29895
  - 09f208fe63 Do not SSH to the instance if deploy is False
- **PR** #30034: (rallytime) Back-port #29893 to 2015.8 @ *2015-12-28 20:20:51 UTC*
  - **PR** #29893: (pass-by-value) Add info about VolumeType (refs: #30034)
  - 9e385369b7 Merge pull request #30034 from rallytime/bp-29893
  - 2fcf1590b8 Add info about VolumeType
- **PR** #30033: (rallytime) Back-port #29876 to 2015.8 @ *2015-12-28 20:20:42 UTC*
  - **PR** #29876: (abednarik) Updated Cloud msic section. (refs: #30033)
  - 4d4dfd692a Merge pull request #30033 from rallytime/bp-29876
  - a257249789 Add versionadded to SSH Port docs
  - 0bb83e51aa Updated Cloud msic section.
- **PR** #30029: (terminalmage) git.latest: Fix handling of nonexistant branches @ *2015-12-28 19:39:29 UTC*
  - a5f7d9c2fc Merge pull request #30029 from terminalmage/git.latest-nonexistant-branch
  - 0b95894c9f git.latest: Fix handling of nonexistant branches
- **PR** #30016: (anlutro) Properly normalize locales in locale.gen_locale @ *2015-12-28 15:33:48 UTC*
  - e7fe24dc64 Merge pull request #30016 from alprs/fix-gen_locale_normalize
  - 75eb4511d3 properly normalize locales in locale.gen_locale

- **PR** #30015: (anlutro) locale module: don't escape the slash in \n @ *2015-12-28 15:31:20 UTC*
  - 90611e95f4 Merge pull request #30015 from alprs/fix-gen_locale_escaped_newline
  - 5799729aee locale module: don't escape the slash in n
- **PR** #30022: (gqgunhed) Two minor typos fixed @ *2015-12-28 15:22:24 UTC*
  - b871ce5310 Merge pull request #30022 from gqgunhed/winrepo_typo
  - a052ff016e fixed minor typos and a :ref: link
  - e47db1a076 Merge remote-tracking branch 'refs/remotes/saltstack/2015.8' into winrepo_typo
  - 0c4c8b9b5c Merge remote-tracking branch 'refs/remotes/saltstack/2015.8' into 2015.8
- **PR** #30026: (anlutro) states.at: fix wrong variable being used @ *2015-12-28 15:21:23 UTC*
  - 4b8ac20d45 Merge pull request #30026 from alprs/fix-at_without_tag_job
  - c0fe9c09bd states.at: fix wrong variable being used
- **PR** #29966: (multani) Fix bigip state/module documentation + serializers documentation @ *2015-12-23 15:06:46 UTC*
  - a3410fdf41 Merge pull request #29966 from multani/fix/docs
  - e6e36372a4 doc: fix documentation link for salt.serializers
  - 23ef472a07 bigip: fix documentation formatting, remove warnings during doc building
- **PR** #29904: (twangboy) Improvements to osx packaging scripts @ *2015-12-22 21:40:23 UTC*
  - **PR** #29858: (twangboy) Osx build (refs: #29904)
  - 8f8c8cedd0 Merge pull request #29904 from twangboy/osx_build
  - 0be53953af Added function to download and check hashes, added hash files
  - 7f0b87bfb3 Added pre/post flight scripts (not running)
  - 9eeb6da7bd Improvements to osx packaging scripts
- **PR** #29950: (multani) boto_iam: fix deletion of IAM users when using delete_keys=true @ *2015-12-22 18:43:07 UTC*
  - 9522bdf4a5 Merge pull request #29950 from multani/fix/states.boto_iam-delete-user
  - 516c8661f4 boto_iam: fix deletion of IAM users when using delete_keys=true
- **PR** #29937: (multani) Fix states.boto_iam group users @ *2015-12-22 17:33:02 UTC*
  - be95d4d79a Merge pull request #29937 from multani/fix/states.boto_iam-group-users
  - 5c86a78d75 boto_iam: handle group's users empty list by removing all users of the group
  - f3461053df boto_iam: passes connection information down to callees
- **PR** #29934: (multani) Fix state.boto_iam virtual name @ *2015-12-22 17:16:25 UTC*
  - 4f2cc5eba7 Merge pull request #29934 from multani/fix/boto_iam
  - 503ede4178 Fix state.boto_iam virtual name
- **ISSUE** #29933: (Reiner030) boto_rds.absent misses pillar variables for final backup (refs: #29943)
- **PR** #29943: (cachedout) Check args correctly in boto_rds @ *2015-12-22 17:15:48 UTC*
  - b36302291d Merge pull request #29943 from cachedout/issue_29933
  - 8bab5eaeaa Check args correctly in boto_rds

- **PR** #29924: (gqgunhed) fixed: uptime now working on non-US Windows @ *2015-12-22 15:03:17 UTC*
  - 02ed5b8fd1 Merge pull request #29924 from gqgunhed/gqgunhed-2015.8
  - b67c3b45e1 removed duplicate datetime line
  - ed8ee91dcf fixed: uptime now working on non-US Windows
- **PR** #29883: (serge-p) fix for nfs mounts in _active_mounts_openbsd() @ *2015-12-21 18:26:49 UTC*
  - 5e44639334 Merge pull request #29883 from serge-p/patch-6
  - dd94332f24 Update mount.py
  - 9d059a1ea5 fix for nfs mounts in _active_mounts_openbsd()
- **ISSUE** #29866: (tony) spm(1) command should obey Saltfile (refs: #29894)
- **PR** #29894: (techhat) Support Saltfile in SPM @ *2015-12-21 18:03:07 UTC*
  - 08fd81cc3d Merge pull request #29894 from techhat/spmsaltfile
  - 279ec61274 Support Saltfile in SPM
- **PR** #29856: (rallytime) Added some initial unit tests for the salt.modules.vsphere.py file @ *2015-12-21 17:12:20 UTC*
  - 4f46255044 Merge pull request #29856 from rallytime/esxi-unit-tests
  - b908ebd123 Added some initial unit tests for the salt.modules.vsphere.py file
- **PR** #29855: (rallytime) Back-port #29740 to 2015.8 @ *2015-12-21 17:11:28 UTC*
  - **PR** #29740: (kiorky) Type mess in git.latest (refs: #29855)
  - 096fec6182 Merge pull request #29855 from rallytime/bp-29740
  - 4c5e277367 Type mess in git.latest
- **PR** #29890: (multani) Various documentation fixes @ *2015-12-21 16:25:15 UTC*
  - 02ab9b8858 Merge pull request #29890 from multani/fix/docs
  - 5aa0e9b1e0 Fix documentation typo for pillars
  - f2b41d04d7 Fix rendering issues for Cherrypy netapi documentation.
  - 6922da46dc doc: fix warnings + some rendering issues
- **PR** #29850: (basepi) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2015-12-18 21:33:49 UTC*
  - 50f48c4bf3 Merge pull request #29850 from basepi/merge-forward-2015.8
  - 7402599c62 Merge remote-tracking branch 'upstream/2015.5' into merge-forward-2015.8
  - f43f3d166c Merge pull request #29730 from rallytime/fix-24698
    - 120fd5fdf0 Update docker-py version requirement to 0.6.0 for dockerio.py files
  - c393a4175a Merge pull request #29715 from rallytime/fix-23343
    - a0ed857c37 Install correct package version, if provided, for npm state.
  - 1310afbbc2 Merge pull request #29721 from terminalmage/nested-output-multiline-fix
    - 761be9cb93 Fix display of multiline strings when iterating over a list
  - 52cc07cec9 Merge pull request #29646 from rallytime/fix-29488
    - c5fa9e9351 Don't stacktrace on kwargs.get if kwargs=None

- **PR** #29811: (anlutro) influxdb: add retention policy module functions @ *2015-12-18 17:19:02 UTC*
    - 05d2aaaef2 Merge pull request #29811 from alprs/feature-influxdb_retention
    - 51088d938a add tests, rename a function to more closely mirror influxdb
    - 785da17a67 missing comma
    - 7e9e9a1030 influxdb: add retention policy module functions
- **ISSUE** #29396: (Ch3LL) Windows 2012 Multi-Master ZMQError (refs: #29814)
- **PR** #29814: (basepi) [2015.8][Windows] Fix multi-master on windows @ *2015-12-18 17:16:52 UTC*
    - 7eefaac58a Merge pull request #29814 from basepi/multi-master.windows.29396
    - 2405501d75 Add documentation for tcp ipc_mode and multi-master
    - 307e867980 For tcp ipc_mode, give each minion different pub/pull ports
    - 5a21893e82 Fix ipc_mode check in windows
- **PR** #29819: (rallytime) Add esxi module and state to docs build @ *2015-12-18 16:20:27 UTC*
    - fb4eb28645 Merge pull request #29819 from rallytime/esxi-docs
    - e7c5863468 Add esxi module and state to docs build
- **PR** #29832: (jleimbach) Fixed typo in order to use the keyboard module for RHEL without systemd @ *2015-12-18 16:04:57 UTC*
    - e865c787a4 Merge pull request #29832 from jleimbach/fix-keyboard.py-for-rhel-without-systemd
    - 7b72b3c52c Fixed typo in order to use the keyboard module for RHEL without systemd
- **PR** #29803: (rallytime) Add vSphere module to doc ref module tree @ *2015-12-17 18:52:56 UTC*
    - 4044f3bb93 Merge pull request #29803 from rallytime/vsphere-docs
    - 3b7f5540ec Add vSphere module to doc ref module tree
- **ISSUE** #29751: (ether42) mod_hostname behavior is systemd dependant (refs: #29767)
- **PR** #29767: (abednarik) Hosts file update in mod_hostname. @ *2015-12-17 18:31:18 UTC*
    - 9b4c2194f6 Merge pull request #29767 from abednarik/network_mod_hpstname_fix
    - eebd3e3e4a Hosts file update in mod_hostname.
- **ISSUE** #29631: (joshughes) pygit2: git submodules cause traceback in file_list (refs: #29772)
- **PR** #29772: (terminalmage) pygit2: skip submodules when traversing tree @ *2015-12-17 18:23:16 UTC*
    - 0c65eeb82b Merge pull request #29772 from terminalmage/issue29631
    - 8c4ea64b0d pygit2: skip submodules when traversing tree
- **PR** #29765: (gtmanfred) allow nova driver to be boot from volume @ *2015-12-17 18:20:33 UTC*
    - 1b430b251f Merge pull request #29765 from gtmanfred/2015.8
    - e95f7561c5 cloudnetworks should be making public_ips a list
    - ec7e45fbfb add documentation for boot from volume on nova driver
    - eafcc5e3ac Add boot from volume for openstack nova
- **PR** #29773: (l2ol33rt) Append missing wget in debian installation guide @ *2015-12-17 17:29:18 UTC*
    - c4f226f31e Merge pull request #29773 from l2ol33rt/debian_install_docfix

  - 64cb4b0540 Append missing wget in debian installation guide
- **PR** #29800: (rallytime) Back-port #29769 to 2015.8 @ *2015-12-17 17:28:52 UTC*
  - **PR** #29769: (pass-by-value) Add documentation about scopes (GCE) (refs: #29800)
  - aca4da3abc Merge pull request #29800 from rallytime/bp-29769
  - 10bfcb8cb0 Add documentation about scopes (GCE)
- **PR** #29775: (paulnivin) Change listen requisite resolution from name to ID declaration @ *2015-12-16 22:56:03 UTC*
  - ab61f78295 Merge pull request #29775 from lyft/listen-id-declaration-resolution-stable
  - ff3a809c11 Change listen requisite resolution from name to ID declaration
- **PR** #29754: (rallytime) Back-port #29719 to 2015.8 @ *2015-12-16 17:25:51 UTC*
  - **PR** #29719: (gqgunhed) fixed: include all items from kern.disks split (refs: #29754)
  - 5af64b64f2 Merge pull request #29754 from rallytime/bp-29719
  - ed275977e3 fixed: include all items from kern.disks split
- **PR** #29713: (The-Loeki) Pillar-based cloud providers still forcing use of deprecated 'provider' @ *2015-12-16 14:51:31 UTC*
  - **PR** #27953: (The-Loeki) Fix CloudStack cloud for new 'driver' syntax (refs: #29713)
  - b3f17fdaf8 Merge pull request #29713 from The-Loeki/patch-1
  - 35fe2a5c18 lint fix
  - dfab6f8186 Update __init__.py
  - 65e2d9ac1e Pillar-based cloud providers still forcing use of deprecated 'provider'
- **ISSUE** #14634: (Sacro) 'unless' documentation isn't logically plausible (refs: #29729)
- **PR** #29729: (rallytime) Further clarifications on "unless" and "onlyif" requisites. @ *2015-12-16 14:45:06 UTC*
  - 1f4810be0f Merge pull request #29729 from rallytime/fix-14634
  - 45b77fb288 Add note about shell truthiness vs python truthiness
  - 3bfb87c031 Spelling fixes
  - 15c466cc12 Further clarifications on "unless" and "onlyif" requisites.
- **ISSUE** #29736: (akissa) Pillar sqlite3 examples incorrect (refs: #29737)
- **PR** #29737: (akissa) fix pillar sqlite3 documentation examples @ *2015-12-16 14:41:57 UTC*
  - 7084f79199 Merge pull request #29737 from akissa/fix-pillar-sqlite3-examples
  - 1c98f8d609 fix pillar sqlite3 documentation examples
- **ISSUE** #29741: (akissa) Pillar Sqlite3 does not honour database config option when using salt-call (refs: #29743)
- **PR** #29743: (akissa) fix pillar sqlite not honouring config options @ *2015-12-16 14:40:27 UTC*
  - e977096409 Merge pull request #29743 from akissa/fix-pillar-sqlite3-does-not-honour-config
  - 6184fb1ae1 fix pillar sqlite not honouring config options
- **ISSUE** #29152: (guettli) docs for states.postgres_user.present: name and password twice? (refs: #29723)
- **PR** #29723: (rallytime) Clarify db_user and db_password kwargs for postgres_user.present state function @ *2015-12-15 23:58:43 UTC*

  - 2cea0b0a2d Merge pull request #29723 from rallytime/fix-29152
  - 8d8fdd0a27 Clarify db_user and db_password kwargs for postgres_user.present state function
- **ISSUE** #29154: (guettli) [Docs] for cmd.run. Missing link to details for "stateful" (refs: #29722)
- **PR** #29722: (rallytime) Link "stateful" kwargs to definition of what "stateful" means for cmd state. @ *2015-12-15 23:25:48 UTC*
  - 30eab23c43 Merge pull request #29722 from rallytime/fix-29154
  - 5c045a86af Link "stateful" kwargs to definition of what "stateful" means for cmd state.
- **ISSUE** #29091: (gravyboat) Salt pillar best practices should show 2 matchers in base (refs: #29724)
- **PR** #29724: (rallytime) Add examples of using multiple matching levels to Pillar docs @ *2015-12-15 23:02:32 UTC*
  - c9ca1a371e Merge pull request #29724 from rallytime/fix-29091
  - 45080f3629 Add examples of using multiple matching levels to Pillar docs
- **PR** #29726: (cachedout) Disable some boto tests per resolution of moto issue @ *2015-12-15 22:15:35 UTC*
  - 4985cc57f1 Merge pull request #29726 from cachedout/disable_moto_2015_8
  - d19827fd3a Disable some boto tests per resolution of moto issue
- **ISSUE** #25723: (jamesog) file.directory fails in test mode when using recurse ignore_files (refs: #29708)
- **PR** #29708: (lagesag) Fix test=True for file.directory with recurse ignore_files/ignore_dirs. @ *2015-12-15 19:15:14 UTC*
  - aba82abffd Merge pull request #29708 from lagesag/fix-file-directory-test-mode
  - a872b5eecf PyLint fix #25723
  - 3e46cb9213 Fix test=True for file.directory with recurse ignore_files/ignore_dirs.
- **ISSUE** #29199: (hubez) 2015.8.1 and 2015.5.6: salt-minion self-restart doesn't work in daemon mode. Works when not a daemon (refs: #29642)
- **PR** #29642: (cachedout) Correctly restart deamonized minions on failure @ *2015-12-15 19:02:40 UTC*
  - 7c38dec0ad Merge pull request #29642 from cachedout/issue_29199
  - 8b2c6817cf Sleep before restart
  - 4105e2abfb Correctly restart deamonized minions on failure
- **PR** #29599: (cachedout) Clean up minion shutdown @ *2015-12-15 19:01:35 UTC*
  - bd918394c3 Merge pull request #29599 from cachedout/clean_minion_shutdown
  - 0b917971fe Log at debug level instead
  - a04280ceb3 Re-raise error to preserve restart behavior
  - dc480e332a Clean up warning on failed master ping.
  - 049a3dbbbc Additional fixes.
  - 8a4969b730 Clean up minion shutdown
- **PR** #29675: (clinta) allow returning all refs @ *2015-12-15 18:55:36 UTC*
  - 31eb291caf Merge pull request #29675 from clinta/git-ls-remote-noref
  - f8c34b0c76 version updated

  - 73b169e7dd lint, remove trainling whitespace
  - 8400e68426 allow returning all refs
- **PR** #29683: (rallytime) Catch more specific error to pass the error message through elegantly. @ *2015-12-15 18:41:54 UTC*
  - 7c50533d3f Merge pull request #29683 from rallytime/vsan_fixes
  - afc003079e Catch more specifc error to pass the error message through elegantly.
- **PR** #29687: (basepi) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2015-12-15 18:38:46 UTC*
  - 30499e4896 Merge pull request #29687 from basepi/merge-forward-2015.8
  - b51cba59c0 Merge remote-tracking branch 'upstream/2015.5' into merge-forward-2015.8
    - f606c23ea8 Merge pull request #29673 from rallytime/fix-29661
      - e4af7a1157 Default value should be False and not 'False'
    - f77c8e7baf Merge pull request #29527 from jfindlay/2015.5
      - 1a8044f0c9 2015.5.7 notes: add note about not being released
- **ISSUE** #27611: (benburkert) PR #26818 broke git.latest with :mirror/:bare (refs: #29681)
- **PR** #29681: (clinta) fix bare/mirror in git.latest @ *2015-12-15 18:37:16 UTC*
  - 3c427e82bf Merge pull request #29681 from clinta/git-mirror
  - b387072a6f fix bare/mirror in git.latest
- **PR** #29644: (rallytime) Fixed a couple more ESXi proxy minion bugs @ *2015-12-14 18:36:28 UTC*
  - fe0778dad5 Merge pull request #29644 from rallytime/esxi-fixes
  - 577d5487a3 Fixed a couple more ESXi proxy minion bugs
- **PR** #29645: (rallytime) Back-port #29558 to 2015.8 @ *2015-12-14 18:11:38 UTC*
  - **PR** #29558: (ruxandraburtica) Returning security group when no VPC id is given (refs: #29645)
  - ef2c9e3f61 Merge pull request #29645 from rallytime/bp-29558
  - 2cf9374342 Replaced tabs with spaces.
  - 5e7e3fe682 Returning security group when no VPC id is given, even if the group is not in EC2-classic.
- **ISSUE** #29630: (c4t3l) Fresh minion install (2015.8.3) returns service __virtual__ is False errors on salt-calls (refs: #29632)
- **ISSUE** #29581: (zmalone) Complaints about pyOpenSSL version on Saltstack 2015.8.3 (refs: #29632)
- **PR** #29632: (jfindlay) reduce severity of tls module __virtual__ logging @ *2015-12-11 20:11:32 UTC*
  - a2a7f1527b Merge pull request #29632 from jfindlay/tls_virt
  - 3ed6a052fd modules.tls.__virtual__: don't spam everyone's error log
  - 76a200e780 modules.tls.__virtual__: refactor cert path comment
  - 0a0532e598 modules.tls.__virtual__: remove redundant parens
- **ISSUE** #29598: (javicacheiro) Duplicated MTU entry added (refs: #29606)
- **PR** #29606: (abednarik) Fixed duplicate mtu entry in RedHat 7 network configuration. @ *2015-12-11 17:24:45 UTC*
  - f6f3aa6613 Merge pull request #29606 from abednarik/remove_duplicate_mtu_entry_rh7_net_template

  - afb2f887ba Fixed duplicate mtu entry in RedHat 7 network configuration.
- **PR** #29613: (rallytime) Various ESXi Proxy Minion Bug Fixes @ *2015-12-11 17:18:58 UTC*
  - c7e73bc4c8 Merge pull request #29613 from rallytime/esxi-fixes
  - aa5dd88b6f Various ESXi Proxy Minion Bug Fixes
- **ISSUE** #26364: (cedwards) [freebsd] TCP transport not working in 2015.8.0rc3 (refs: #29628)
- **PR** #29628: (DmitryKuzmenko) Don't create io_loop before fork @ *2015-12-11 17:15:11 UTC*
  - a56c763423 Merge pull request #29628 from DSRCompany/bug/26364_freebsd_tcp
  - 729ffcae36 Don't create io_loop before fork
- **PR** #29609: (basepi) [2015.8][salt-ssh] Add ability to set salt-ssh command umask in roster @ *2015-12-10 22:52:27 UTC*
  - 41b8117237 Merge pull request #29609 from basepi/salt-ssh.umask.29574
  - 0afa5b0d5d Add cmd_umask to roster docs
  - 5c03f892bc Allow setting the cmd_umask from within the roster
- **ISSUE** #29586: (basepi) Orchestrate doesn't handle minion error properly (refs: #29603)
- **ISSUE** #29546: (jefferyharrell) Can't seem to get orchestrate to recognize a failed state (refs: #29603)
- **PR** #29603: (basepi) Fix orchestration failure-checking @ *2015-12-10 21:23:57 UTC*
  - 1e394f5ab1 Merge pull request #29603 from basepi/orchestrate.failures.29546
  - 2bdcadaa27 Remove unnecessary and
  - 501f91a388 Fix error in failure checking for salt.state within orchestration
- **ISSUE** #29584: (kwilliams057) dockerng image-present fails when trying to pull from registry (refs: #29597)
- **PR** #29597: (terminalmage) dockerng: Prevent exception when API response contains empty dictionary @ *2015-12-10 19:57:42 UTC*
  - b5b80b9324 Merge pull request #29597 from terminalmage/issue29584
  - d68067b5db dockerng: Prevent exception when API response contains empty dictionary
- **ISSUE** #29585: (job) cidr argument in salt.modules.network.ip_addrs6() is broken (refs: #29587)
- **PR** #29596: (rallytime) Back-port #29587 to 2015.8 @ *2015-12-10 19:57:18 UTC*
  - **PR** #29587: (job) Fix the 'cidr' arg in salt.modules.network.ip_addrs6() (refs: #29596)
  - ffb54cced7 Merge pull request #29596 from rallytime/bp-29587
  - bfb75ce363 Fix the 'cidr' arg in salt.modules.network.ip_addrs6()
- **PR** #29588: (rallytime) Added ESXi Proxy Minion Tutorial @ *2015-12-10 16:17:51 UTC*
  - 08dd663a27 Merge pull request #29588 from rallytime/esxi-proxy-tutorial
  - 5a2bb260d3 Added ESXi Proxy Minion Tutorial
- **ISSUE** #29557: (arthurlogilab) [modules/nova] AttributeError: 'module' object has no attribute 'discover_extensions' when using nova.image_list (refs: #29572)
- **PR** #29572: (gtmanfred) [nova] use old discover_extensions if available @ *2015-12-09 17:35:42 UTC*
  - fe5db23863 Merge pull request #29572 from gtmanfred/2015.8
  - d0ffa520f4 use old discover_extensions if available

- **ISSUE** #29009: (LoveIsGrief) git.latest doesn't checkout submodules (refs: #29545)
- **PR** #29545: (terminalmage) git.latest: init submodules if not yet initialized @ *2015-12-09 16:19:42 UTC*
  - ecbc60ba05 Merge pull request #29545 from terminalmage/issue29009
  - 6619503aec git.latest: init submodules if not yet initialized
- **PR** #29548: (rallytime) Back-port #29449 to 2015.8 @ *2015-12-09 16:19:07 UTC*
  - **PR** #29449: (AkhterAli) Adding message for null public IP (refs: #29548)
  - 3b2c93a2e5 Merge pull request #29548 from rallytime/bp-29449
  - 3715cd7d65 Adding message for null public IP
- **PR** #29547: (rallytime) Refactored ESXCLI-based functions to accept a list of esxi_hosts @ *2015-12-09 16:08:03 UTC*
  - fd67903bf9 Merge pull request #29547 from rallytime/esxi-proxy
  - 469648dd07 Refactored ESXCLI-based functions to accept a list of esxi_hosts
- **PR** #29563: (anlutro) Fix a call to deprecated method in python-influxdb @ *2015-12-09 16:00:24 UTC*
  - 21437f9235 Merge pull request #29563 from alprs/fix-influx_deprecated_method
  - 7c69c177ed update test
  - 46d7d92069 fix a call to deprecated method in python-influxdb
- **PR** #29565: (bdrung) Fix typos and missing release note @ *2015-12-09 15:59:21 UTC*
  - f29e0a7021 Merge pull request #29565 from bdrung/2015.8
  - b96d8ff1d9 Minor update to release notes for missing fix
  - e72354aac4 Fix typo specfic -> specific
  - 5708355762 Fix typo comparsion -> comparison
- **PR** #29540: (basepi) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2015-12-08 21:27:01 UTC*
  - 25d3a75d8c Merge pull request #29540 from basepi/merge-forward-2015.8
  - e59364ad1d Fix failing unit test
  - 9673fd0937 Merge remote-tracking branch 'upstream/2015.5' into merge-forward-2015.8
    - 867d550271 Merge pull request #29539 from basepi/merge-forward-2015.5
      - 2c9c4ba430 Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.5
      - 85aa70a6cb Merge pull request #29392 from jacobhammons/2014.7
      - d7f0db1dd8 updated version number to not reference a specific build from the latest branch
    - de7f3d5a59 Merge pull request #29504 from rallytime/fix-12072
      - 8357c95dc2 Document userdata_file option for EC2 driver
    - 65deba8bb5 Merge pull request #29507 from rallytime/ec2-doc-fix
      - 90b4823bc2 Switch volumes and del_*_on_destroy example ordering
    - 0918c9294f Merge pull request #29469 from abednarik/doc_note_for_saltcloud_connection_timeout
      - 8e5c3e366a Added Documentation note in salt cloud.
    - e43c7c05a6 Merge pull request #29461 from dmyerscough/fix-resource-limits

        - 85a8a3b033 Fix resource limits, systemd sets the default number of open files to 4096 causing te master to complain about limits when you have a large number of keys
    - 730f02fbdf Merge pull request #29439 from rallytime/bp-28656
        - 2f11bb021f #28526 fixed yumpkg module
    - 197210d52e Merge pull request #29418 from jacobhammons/dot8
        - 4f51a737f9 Added CVE 2015-8034 to 2015.5.8 release notes
    - b3452f2a1a Merge pull request #29389 from jacobhammons/2015.5
        - 824721ff36 updated version numbers
    - 6a7a95f28a Merge pull request #28501 from twangboy/jmoney_26898
        - c0cf33332c Fixed some Lint…
        - df17fc59d3 Merge pull request #6 from jfindlay/twang_test
        - bc7e0cfe64 add file.symlink unit tests
        - 9381dc7215 orthogonalize file.symlink unit tests
        - 8f462ba044 Merge pull request #5 from cachedout/fix_twangboy_test
        - 5293150d25 Fix tests
        - 7d39091c91 Fixed some more lint
        - 3dbd62af2c Fixed some tests… hopefully
        - f187db3288 Removed unnecessary logic
        - 89ebd268e6 Added file attributes restore on fail
        - 9ec72ca724 fix file state unit tests for win symlink feature
        - 69c32a663e Fixed some lint
        - 638dec5027 Fixed some tests… let's see if they're really are
        - 5ed7a99792 Replaced instances of shutil.rmtree in file state
        - 2651ce509f Fix file.remove for windows
    - 760a521603 Merge pull request #29348 from jtand/file_search_fix
        - 04f82bd4fd Fixes an file.search on python2.6
    - 51ea88d489 Merge pull request #29336 from rallytime/bp-29276
        - 3a0e19debb Prevent adding port twice when adding entry in known hosts
    - 28255af52a Merge pull request #29333 from rallytime/bp-29280
        - 722d02ff4a Lint
        - 4a0040c1b4 [Doc] Add note for SVN state

- **PR** #29499: (rallytime) Initial commit of ESXi Proxy Minion @ *2015-12-08 21:10:13 UTC*
    - 3ae096b7ac Merge pull request #29499 from rallytime/esxi-proxy
    - d8b1ba3991 Make sure ESXCLI gating is correct in vsphere __virtual__
    - 55589f8021 Provide some more inline comments for longer functions
    - baf2f8ce7a Pylint fix

  - 763ae5d676 VMotion functions, gate ESXCLI requirement, allow protocol/port for ESXCLI function
  - d909df254e Bug fixes for esxi states
  - 7102677679 Bug fixes and move ntp and ssh service start/stop/restart to single funcs
  - 77b37add84 Added syslog_configured state, and some minor bug fixes
  - df49f533f6 More state functions and a couple of bug fixes
  - a50c74cfe2 Merge pull request #13 from cro/esxi-proxy3
    - 87fc980f33 Add syslog config and network firewall rules enable
  - 42be49f481 Merge pull request #11 from cro/esxi-proxy
    - d858642f05 Add documentation.
    - 43879d1dfe Functions for setting network coredumps
  - 7d7d2afa7f Initial commit of ESXi state and refactored vsan_add_disks to include a get function.
  - bc945a48db Add execution module functions to upload ssh key for root and retrieve ssh key for root.
  - 238b0f5bea Update error return policy and add service running/policy functions
  - 9ba9019419 Initial commit of ESXi proxy work.
- **PR** #29526: (jfindlay) 2015.8.2 notes: add note about not being released @ *2015-12-08 21:09:50 UTC*
  - 873f6a9460 Merge pull request #29526 from jfindlay/2015.8
  - 917e6f850c 2015.8.2 notes: add note about not being released
- **ISSUE** #29484: (m7v8) patchlevel detection broken for openSuSE (refs: #29531)
- **PR** #29531: (jfindlay) grains.core: handle undefined variable @ *2015-12-08 21:07:38 UTC*
  - 3de61e3655 Merge pull request #29531 from jfindlay/suse_patch
  - 1ad5a088fc grains.core: handle undefined variable
- **ISSUE** #29486: (m7v8) Pull request breaks our setup (umask) (refs: #29538)
- **ISSUE** #29005: (fcrozat) non-standard umask breaks salt-call call in salt-ssh (refs: #29126)
- **ISSUE** #28830: (fcrozat) non-standard umask breaks salt-ssh deployement (refs: #29126)
- **PR** #29538: (basepi) [2015.8] [salt-ssh] Remove umask around actual execution for salt-ssh @ *2015-12-08 20:45:58 UTC*
  - **PR** #29126: (fcrozat) Fix deployment when umask is non-standard (refs: #29538)
  - 1d8014411a Merge pull request #29538 from basepi/salt-ssh.umask.29486
  - 5edfa014f5 Remove umask around actual execution for salt-ssh
- **ISSUE** #28715: (mlalpho) Tagging Resources with boto_rds (refs: #29505)
- **PR** #29505: (rallytime) Update boto_rds state docs to include funky yaml syntax for "tags" option. @ *2015-12-08 17:05:02 UTC*
  - fb02fc1ef1 Merge pull request #29505 from rallytime/fix-28715
  - f43f851a92 Update boto_rds state docs to include funky yaml syntax for "tags" option.
- **PR** #29513: (bdrung) Drop obsolete syslog.target from systemd services @ *2015-12-08 16:05:01 UTC*
  - 38888add5e Merge pull request #29513 from bdrung/2015.8
  - b1a4ade618 Drop obsolete syslog.target from systemd services

- **PR** #29500: (rallytime) Back-port #29467 to 2015.8 @ *2015-12-07 23:24:00 UTC*
  - **PR** #29467: (serge-p) Update module.py (refs: #29500)
  - 148dad6674 Merge pull request #29500 from rallytime/bp-29467
  - ca0be8bff0 Update module.py
- **ISSUE** #29001: (olfway) debconf.set doesn't support "prereq" in states (refs: #29463)
- **PR** #29463: (abednarik) Add **kwargs to debconf.set. @ *2015-12-07 19:56:05 UTC*
  - 9d11acc7db Merge pull request #29463 from abednarik/debconf_fix_prereq_support
  - b17f1fed43 Add **kwargs to debconf.set.
- **ISSUE** #29311: (Reiner030) Feature Request: System uptime also in seconds (refs: #29399)
- **PR** #29399: (jfindlay) modules.status: add human_readable option to uptime @ *2015-12-07 19:53:52 UTC*
  - 7efd6dd140 Merge pull request #29399 from jfindlay/second_up
  - 1903124814 modules.win_status: add reason to virtual ret
  - 35ba7da470 modules.status: add reason to __virtual__ return
  - 48e7beb0eb modules.status: add in_seconds option to uptime
- **PR** #29433: (cro) Files for building .pkg files for MacOS X @ *2015-12-07 19:47:23 UTC*
  - 042daf91b8 Merge pull request #29433 from cro/mac_native_pkg
  - 8e191ae264 Add web references
  - 5f1459d708 Update mac packaging
  - 092b7ddd0a First crack at build files for Mac OS X Native package
- **ISSUE** #29445: (shawnbutts) salt.loaded.int.module.nova.__virtual__() is wrongly returning *None*. (refs: #29455)
- **PR** #29455: (jfindlay) modules.nova.__init__: do not return `None` @ *2015-12-07 19:44:00 UTC*
  - 5ff3749108 Merge pull request #29455 from jfindlay/nova_none
  - 19da8233c8 modules.nova.__init__: do not return `None`
- **PR** #29454: (jfindlay) rh_service module __virtual__ return error messages @ *2015-12-07 19:32:15 UTC*
  - 289e9d169e Merge pull request #29454 from jfindlay/rh_service
  - 9975508f86 modules.rh_service.__virtual__: handle SUSE osrelease as num
  - d7ab7bf51f modules.rh_service: __virtual__ error messages
- **PR** #29476: (tbaker57) Doc fix - route_table_present needs subnet_names (not subnets) as a key @ *2015-12-07 18:47:22 UTC*
  - cb465927d6 Merge pull request #29476 from tbaker57/boto_vpc_docfix
  - 36946640b8 Fix - don't specify 'name' key inside the list - just the subnet names
  - 5cab4b775a Doc fix - route_table_present needs subnet_names (not subnets) as a key
- **PR** #29487: (rallytime) Back-port #29450 to 2015.8 @ *2015-12-07 17:25:23 UTC*
  - **PR** #29450: (pass-by-value) Raise error if dracr password is above 20 chars (refs: #29487)
  - 6696cf6eb5 Merge pull request #29487 from rallytime/bp-29450
  - 2c55c55ff1 Raise error if dracr password is above 20 chars

- **ISSUE** #29133: (cedwards) FX2 proxy-minion dellchassis idrac state incomplete (refs: #29441)
- **PR** #29441: (rallytime) Make sure docs line up with blade_idrac function specs @ *2015-12-05 16:30:27 UTC*
  - a1ffc5aacb Merge pull request #29441 from rallytime/fix-doc-dellchassis
  - cf62361830 Make sure docs line up with blade_idrac function specs
- **PR** #29440: (rallytime) Back-port #28925 to 2015.8 @ *2015-12-05 00:21:26 UTC*
  - **PR** #28925: (pass-by-value) Fx2 firmware update (refs: #29440)
  - 6cc6f776bc Merge pull request #29440 from rallytime/bp-28925
  - 1b57a57c48 Lint fixes
  - 7cea3afb4f Support multiple hosts
  - 0be3620715 Set kwarg
  - b7324b5102 Add doc for new state
  - 613dd0b7a2 Make sure creds are set before `racadm update`
  - 929e679b25 Add firmware update state to dellchassis
  - 6356af3b99 Raise error
  - 820ad7b3df Validate file existence
  - 94704304ec Add firmware update functions to module
- **ISSUE** #29425: (paclat) services for older OEL releases. (refs: #29435)
- **PR** #29435: (galet) Grains return wrong OS version and other OS related values for Oracle Linux @ *2015-12-05 00:19:11 UTC*
  - 129f45f7c3 Merge pull request #29435 from galet/2015.8
  - fdaa81ccf8 Grains return wrong OS version and other OS related values for Oracle Linux
  - c494ddd5fc Grains return wrong OS version and other OS related values for Oracle Linux
- **ISSUE** saltstack/salt#29313: (rmatulat) state/host.present and alias-assignment to multiple IPs fails (refs: #29430)
- **PR** #29430: (rmatulat) Fix host.present state limitation @ *2015-12-04 23:08:20 UTC*
  - e2b43a3f1e Merge pull request #29430 from rall0r/2015.8
  - d3dacff4a2 Fix host.present state limitation
- **PR** #29417: (jacobhammons) Repo install updates @ *2015-12-04 02:39:41 UTC*
  - ab890b632a Merge pull request #29417 from jacobhammons/repo-install-updates
  - d58182c5fa updated repo path for RHEL installation
  - 5e54359869 Updated Debian, RHEL / Cent, Ubuntu installation instructions with new repo structure for 2015.8.3. Added CVE-2015-8034 to release notes.
- **PR** #29402: (techhat) Add rate limiting to linode @ *2015-12-03 20:27:10 UTC*
  - cb1e2e6e73 Merge pull request #29402 from techhat/ratelimit
  - f0a4d93077 Add rate limiting to linode
- **ISSUE** #19332: (QuinnyPig) Nondeterminism in Pillar (refs: #29400)
- **PR** #29400: (twangboy) Fix #19332 @ *2015-12-03 20:25:16 UTC*

  - 8fe39d0ef8 Merge pull request #29400 from twangboy/fix_19332
  - 7bdddaca53 Fixed grammer
  - d965d00a09 Fix #19332
- **PR** #29398: (cachedout) Lint 29288 @ *2015-12-03 18:03:53 UTC*
  - d2c0fcbc97 Merge pull request #29398 from cachedout/lint_29288
  - 3b0033e529 Lint #29288
  - 386459ca6d Merge pull request #1 from jfindlay/glustest
    - 4d6c71aa80 modules.glusterfs: fix start_volume unit test
  - f336c44630 Bootstrap failed, retrying
  - bd729cb3ea Set default GlusterFS version to 6
  - 443bfc6a81 Fixed volume status for >= 3.7 in glusterfs.py
- **ISSUE** #29116: (johnsocp) Unresolvable masters in the minions masters list cause minion to raise an error (refs: #29331)
- **PR** #29331: (DmitryKuzmenko) Bugfix - #29116 raet dns error @ *2015-12-03 17:10:40 UTC*
  - 5b8e7820ac Merge pull request #29331 from DSRCompany/bug/29116_raet_dns_error_2
  - 8c2b217af5 Make pylint happy
  - e5672ee716 Don't exit if no master found
  - 1c324f5467 Don't fail if can't connect to master
- **PR** #29390: (jacobhammons) updated version numbers in documentation @ *2015-12-03 17:02:05 UTC*
  - 7bc6b1210d Merge pull request #29390 from jacobhammons/2015.8
  - 486935b233 updated version numbers
- **ISSUE** #25446: (DmitryKuzmenko) Stack overflow on LazyLoader deep copying (refs: #29381)
- **PR** #29381: (nmadhok) No need to deepcopy since six.iterkeys() creates a copy @ *2015-12-03 15:54:52 UTC*
  - fd677e1d58 Merge pull request #29381 from nmadhok/2015.8-runtime-fix
  - f109698196 No need to deepcopy since six.iterkeys() creates a copy
- **PR** #29349: (cro) Fix mis-setting chassis names @ *2015-12-03 00:56:54 UTC*
  - 2973025058 Merge pull request #29349 from cro/fx2_name_fix
  - 95d6d72a5d Fix mis-setting the name of the chassis.
- **ISSUE** #29236: (sjorge) network.mod_bufsize has wrong docstring (refs: #29237)
- **ISSUE** #29235: (sjorge) network.get_bufsize has wrong docstring (refs: #29237)
- **ISSUE** #29234: (sjorge) network.dig should only be available if we have the 'dig' binary (refs: #29237)
- **ISSUE** #29233: (sjorge) network.default_route does not seem to honor the family parameter (refs: #29237)
- **ISSUE** #29232: (sjorge) network.active_tcp seems linux specific (refs: #29237)
- **ISSUE** #29231: (sjorge) docstrings in salt/utils/network.py are incorrect (refs: #29237)
- **PR** #29334: (rallytime) Back-port #29237 to 2015.8 @ *2015-12-02 19:37:31 UTC*
  - **PR** #29237: (sjorge) Module network fixes (refs: #29334)

  - 17d80c051a Merge pull request #29334 from rallytime/bp-29237
  - 598226def1 fix unit test (attempt 1)
  - a461d7bf12 changed from Boron to 2015.8.4, so this can be backported
  - 3892b12514 fix up a few remarks from jfindlay
  - 2f940e22aa also we should keep returning {} for other systems
  - 4953f58894 forgot to remove a debug line, how embarasing
  - e96f3c0c3b fix docs in salt/utils/network.py #29231 - looks like this got copied at some point
  - 3888bb403f fixup network.default_route with family set on SunOS #29233
  - c0e6ea98a6 fix network.active_tcp on SunOS (we fake it until we make it) #29232
  - 92f881284e add decorator to network.dig #29234
  - 77950eb55c fix docstring for get_bufsize #29235
  - 52fb80cd18 fix docstring for mod_bufsize #29236
- **ISSUE** #28990: (adithep) Dockerng volume (refs: #29300)
- **PR** #29300: (ticosax) [dockerng] Add support for volume management in dockerng @ *2015-12-02 17:48:53 UTC*
  - 5ec7947595 Merge pull request #29300 from ticosax/dockerng-volumes
  - 80d085ea92 fix typo
  - cb9cb463b0 Provide states for managing docker volumes
  - dff6fa1fb2 Add execution module to manage docker volumes
- **PR** #29218: (clan) check service enable state in test mode @ *2015-12-02 15:31:00 UTC*
  - 99b7d87688 Merge pull request #29218 from clan/service_state
  - a1250a9729 check service enable state in test mode
- **PR** #29315: (jfindlay) dev tutorial doc: fix markup errors @ *2015-12-01 21:42:17 UTC*
  - 08ced73b13 Merge pull request #29315 from jfindlay/docs
  - e8e23dc444 dev tutorial doc: fix markup errors
- **PR** #29317: (basepi) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2015-12-01 21:28:30 UTC*
  - a3a463ff8b Merge pull request #29317 from basepi/merge-forward-2015.8
  - 0d90dd3a19 Merge remote-tracking branch 'upstream/2015.5' into merge-forward-2015.8
  - 14e94b3593 Merge pull request #29316 from basepi/merge-forward-2015.5
    - 33f40b3c47 Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.5
    - d2fb2109a3 Merge pull request #29296 from douardda/patch-3
    - d2885390f4 Use process KillMode on Debian systems also
  - 6a2ffbfb7c Merge pull request #29216 from clan/file_search_on_proc_file
    - 91a20c07a1 try mmap first
    - 8aa4f2053e remove extra space to fix lint failure
    - d34e6b1a9a use read only if has read() method
    - 3209c1cdb5 size is 0 doesn't mean no data, e.g, /proc/version

  - d6aaae8d7b Merge pull request #29261 from attiasr/patch-1
    - 7a99b90596 add log and return if pkg already installed
    - 1843c7ab8e fix incorrect reinstallation of windows pkg
  - 9236188867 Merge pull request #29214 from cro/ssl_verify_ssl
    - e9c13c561b Doc bug–salt.utils.http takes verify_ssl not ssl_verify.
  - df7b35a86b Merge pull request #29204 from lorengordon/fix-29202
  - b1dae5e6fe Use os.path.join to return full path to ca bundle
- **PR** #29240: (clan) handle acl_type [[d]efault:][user|group|mask|other] @ *2015-12-01 17:56:20 UTC*
  - 39667fda12 Merge pull request #29240 from clan/linux_acl
  - 02429aca69 handle acl_type [[d]efault:][user|group|mask|other]
- **PR** #29305: (lorengordon) Add 'file' as a source_hash proto @ *2015-12-01 17:39:37 UTC*
  - 027bed7c90 Merge pull request #29305 from lorengordon/source_hash_protos
  - 53fdf0bf97 Update message for invalid source_hash
  - 2d20d71bd5 Add *file* as a source_hash proto
- **ISSUE** #29251: (adamsewell) status.uptime causes exception on Windows minion 2015.8.1 (refs: #29272)
- **PR** #29272: (jfindlay) win_status module: handle 12 hour time in uptime @ *2015-12-01 16:33:12 UTC*
  - 1129ee1d2e Merge pull request #29272 from jfindlay/win_up_time
  - 6a2315109e win_status module: python timedelta to find uptime
  - b7a535341f win_status module: handle 12 hour time in uptime
- **ISSUE** #26526: (JensRantil) Managing a file:// source fails (refs: #29289)
- **PR** #29289: (terminalmage) file.managed: Allow local file sources to use source_hash @ *2015-12-01 16:19:27 UTC*
  - 0fd3e8b0fb Merge pull request #29289 from terminalmage/issue26526
  - 64ae3f996e file.managed: Allow local file sources to use source_hash
- **ISSUE** #29262: (anlutro) ssh_auth.absent removes keys when test=True (refs: #29264)
- **PR** #29264: (anlutro) Prevent ssh_auth.absent from running when test=True @ *2015-11-30 21:54:15 UTC*
  - 8d32d8d43d Merge pull request #29264 from alprs/fix-ssh_auth_absent_test
  - 9193676f9c fix ssh_auth_test
  - febbfa792f prevent ssh_auth.absent from running when test=True
- **ISSUE** #29071: (eliasp) *git_pillar.update* runner can't handle >=2015.8.0 configuration (refs: #29277)
- **PR** #29277: (terminalmage) Update git_pillar runner to support new git ext_pillar config schema @ *2015-11-30 21:39:51 UTC*
  - 459d30f27f Merge pull request #29277 from terminalmage/issue29071
  - 6981bb3be7 Update git_pillar runner to support new git ext_pillar config schema
  - 293c8e635c Separate repo locking logic into its own function
- **PR** #29283: (cachedout) Single-quotes and use format @ *2015-11-30 21:34:41 UTC*

  - **PR** #29139: (thomaso-mirodin) [salt-ssh] Add a range roster and range targeting options for the flat roster (refs: #29283)
  - df1f0d93c7 Merge pull request #29283 from cachedout/style_29139
  - d764497b17 Single-quotes and use format
- **PR** #29139: (thomaso-mirodin) [salt-ssh] Add a range roster and range targeting options for the flat roster (refs: #29283) @ *2015-11-30 21:25:50 UTC*
  - 3aa84b6763 Merge pull request #29139 from thomaso-mirodin/salt-ssh-flat-roster-range-filter
  - 56b3302fe9 Pylint fixes for PR #29139
  - e010f2d3b5 Add a range roster for salt-ssh
  - c5eeb77ebc Add range support to salt-ssh's flat roster
- **PR** #29282: (*cachedout*) dev docs: add development tutorial (refs: #29282) @ *2015-11-30 21:14:50 UTC*
  - dbf7755aa2 Merge pull request #29282 from cachedout/fix_29279
  - 1efaab2dd5 Fix typo in #29279
  - a5ea39132f dev docs: add development tutorial
- **ISSUE** #28991: (timcharper) allow role-assumption with s3 credentials (refs: #28994)
- **PR** #28994: (timcharper) add support to s3 for aws role assumption @ *2015-11-30 20:52:18 UTC*
  - 87e4aa4fae Merge pull request #28994 from timcharper/2015.8.1-dev
  - e060986828 add support to s3 for aws role assumption
- **ISSUE** #29209: (ssgward) SPM logging level doesn't seem to be functional (refs: #29278)
- **PR** #29278: (techhat) Add verify_log to SPM @ *2015-11-30 20:48:32 UTC*
  - 3d16434f14 Merge pull request #29278 from techhat/issue29209
  - 759e8c4542 Add verify_log to SPM
- **PR** #29067: (jacksontj) Fix infinite recursion in state compiler for prereq of SLSs @ *2015-11-30 20:27:09 UTC*
  - d651d7167e Merge pull request #29067 from jacksontj/2015.8
  - 64e439cda2 Add test for infinite recursion with sls prerequisites
  - d687682016 No reason to continuously resolve the k, v pair here since it doesn't change in the inner loop
  - 6d747df5db Correctly resolve requisite_in for SLS requisites
- **ISSUE** #29161: (jefferyharrell) saltmod.state's ret argument seems to do nothing (refs: #29207)
- **PR** #29207: (jfindlay) do not shadow ret function argument @ *2015-11-30 20:14:06 UTC*
  - d42bcea905 Merge pull request #29207 from jfindlay/ret_non_shadow
  - 5de0b93ac6 saltutil.cmd module: do not shadow ret function argument
  - 7809f2a389 saltmod.state state: do not shadow ret function argument
- **PR** #29215: (rallytime) Back-port #29192 to 2015.8 @ *2015-11-30 20:12:30 UTC*
  - **PR** #29192: (bastiaanb) fix issue 29191: only try partial matches when a wildcard has been sp… (refs: #29215)
  - 8cc1d8de46 Merge pull request #29215 from rallytime/bp-29192
  - 5226cd8f79 remove trailing whitespace fix subdict_match test cases

  - 44713cdb95 fix issue 29191: only try partial matches when a wildcard has been specified
- **PR** #29217: (clan) show duration only if state_output_profile is False @ *2015-11-30 20:11:18 UTC*
  - **PR** #19320: (clan) add 'state_output_profile' option for profile output (refs: #29217)
  - f488d25911 Merge pull request #29217 from clan/highstate_duration
  - 9bdaae8325 show duration only if state_output_profile is False
- **PR** #29221: (ticosax) [dokcerng] Docu network mode @ *2015-11-30 19:22:49 UTC*
  - e5bd1c293d Merge pull request #29221 from ticosax/docu-network_mode
  - a0b674a0ea Extend documentation of network_mode parameter.
- **ISSUE** #29250: (adamsewell) status.cpu_load is not available on Salt 2015.8.1 (refs: #29269)
- **PR** #29269: (jfindlay) win_status module: fix function names in docs @ *2015-11-30 19:14:24 UTC*
  - 7fd02c2145 Merge pull request #29269 from jfindlay/winstatus
  - f2f2dab491 win_status module: fix function names in docs
- **PR** #29213: (rallytime) Move _wait_for_task func from vmware cloud to vmware utils @ *2015-11-30 18:53:24 UTC*
  - 6c2e62f7d4 Merge pull request #29213 from rallytime/vmware_utils_wait_for_task
  - 44e7f83686 Move _wait_for_task func from vmware cloud to vmware utils
- **PR** #29271: (techhat) Pass full path for digest (SPM) @ *2015-11-30 18:35:42 UTC*
  - 69cbc09ca0 Merge pull request #29271 from techhat/issue29212
  - 6cd6a0ace0 Pass full path for digest (SPM)
- **PR** #29244: (isbm) List products consistently across all SLES systems @ *2015-11-30 18:31:42 UTC*
  - 1efe484309 Merge pull request #29244 from isbm/isbm-zypper-products
  - db36a73b16 Remove code duplication
  - d62abedbf7 Remove dead code
  - 302b5d3bc1 List products consistently across all SLES systems
- **ISSUE** #29119: (mo-mughrabi) salt.modules.consul.catalog_register does not accept address as a string (refs: #29255)
- **PR** #29255: (garethgreenaway) fixes to consul module @ *2015-11-30 18:30:02 UTC*
  - 318ad36449 Merge pull request #29255 from garethgreenaway/29119_consul_module_fixes
  - 655b0ec403 various fixes to the consul execution module, in particular a fix to address #29119
- **PR** #29208: (whytewolf) Glance more profile errors @ *2015-11-25 23:50:27 UTC*
  - b225263279 Merge pull request #29208 from whytewolf/glance_more_profile_errors
  - c8fe514ec1 found 3 more spots where the profile was not being passed through.
  - b2e3c1f8de Merge pull request #1 from saltstack/2015.8
- **ISSUE** #29140: (davidballano) mount.unmounted is not behaving as I would expect (refs: #29200)
- **PR** #29200: (jfindlay) mount state: unmount by device is optional @ *2015-11-25 20:03:22 UTC*
  - 6d3c04516f Merge pull request #29200 from jfindlay/singular_umount
  - b54de47b1b mount state: unmount by device is optional

- **ISSUE** #29187: (trevor-h) salt-cloud Windows provisioning on EC2 fails to use winrm (refs: #29205)
- **PR** #29205: (trevor-h) Fixes #29187 - using winrm on EC2 @ *2015-11-25 20:00:01 UTC*
  - fffcf9fef6 Merge pull request #29205 from trevor-h/fix-salt-cloud-winrm-ec2
  - 48e0edd0d2 Fixes #29187 - using winrm on EC2
- **PR** #29170: (cachedout) Migrate pydsl tests to integration test suite @ *2015-11-25 19:56:48 UTC*
  - 1937a47dec Merge pull request #29170 from cachedout/refactor_pydsl_test
  - 2477ff2eab Add __init__ and pydsl test
  - 063f075a99 Add integration renderer tests to the suite
  - 81bf332be4 Migrate pydsl tests to integration test suite
- **ISSUE** #29137: (Dravu) MTU is output twice when used in network.managed (refs: #29198)
- **PR** #29198: (jfindlay) rh_ip module: only set the mtu once @ *2015-11-25 18:11:09 UTC*
  - 11d68f7b1c Merge pull request #29198 from jfindlay/single_mtu
  - 0a8952f6ac rh_ip module: only set the mtu once
- **ISSUE** #29111: (eliasp) Backtrace in state *ssh_known_hosts.present* when *ssh-keygen* is not available (refs: #29135)
- **PR** #29135: (jfindlay) ssh_known_hosts.present state: catch not found exc @ *2015-11-25 18:10:43 UTC*
  - f19355e0bb Merge pull request #29135 from jfindlay/ssh_except
  - 363add7131 ssh_known_hosts.present state: catch not found exc
- **PR** #29196: (s0undt3ch) We need novaclient imported to compare versions @ *2015-11-25 17:16:27 UTC*
  - 6a12197e13 Merge pull request #29196 from s0undt3ch/2015.8
  - 78a7c34f2b We need novaclient imported to compare versions
- **ISSUE** #28072: (jchv) pygit 0.23.2 is not supported in Salt 2015.8.1 (refs: #29059)
- **PR** #29059: (terminalmage) Work around upstream pygit2 bug @ *2015-11-25 16:39:30 UTC*
  - 0c0e15d4e9 Merge pull request #29059 from terminalmage/issue28072
  - 82e223087e Work around upstream pygit2 bug
- **PR** #29112: (eliasp) Prevent backtrace (KeyError) in *ssh_known_hosts.present* state @ *2015-11-25 16:25:57 UTC*
  - cc69c87dd2 Merge pull request #29112 from eliasp/ssh_known_hosts.present-backtrace-test
  - 3f19c311e8 Prevent backtrace (KeyError) in *ssh_known_hosts.present* state
- **PR** #29178: (whytewolf) Profile not being passed to keystone.endpoint_get in _auth. so if a p… @ *2015-11-25 16:09:49 UTC*
  - 7775d65089 Merge pull request #29178 from whytewolf/glance_keystone_profile_fix
  - 807dd426a6 Profile not being passed to keystone.endpoint_get in _auth. so if a profiles are being used, then keystone.endpoint_get will not be able to authenticate causing glance to not be able to get it's endpoint.

28.2.60 Salt 2015.8.5 Release Notes

Version 2015.8.5 is a bugfix release for *2015.8.0*.

Important: About this Release Salt 2015.8.5 is identical to the 2015.8.4 release with the addition of a fix for issue #30820, fixed by PR #30833. See *here* for the 2015.8.4 release notes.

Known Issue in `boto_*` execution modules

This release contains an issue that causes the `boto_*` execution modules to display a `__salt__ not defined` error (issue #30300). This issue will be fixed in an upcoming release, but can be manually resolved by completing the following:

1. Download the `boto_*` execution modules that you would like to update from the 2015.8 branch of Salt. A complete list of affected modules with the specific changes is available in :pull‘30867‘.

   A simple way to get the updated modules is to download a zip file of the 2015.8 branch from GitHub. The updated modules are in the `salt\modules` directory.

2. Place the `boto_*` modules into `salt://_modules`.

3. Run the following command to sync these modules to all Salt minions:

```
salt '*' saltutil.sync_modules
```

Changelog for v2015.8.4..v2015.8.5

Generated at: 2018-05-27 23:47:32 UTC

- c7db4350d5 Fix regression in scanning for state with ’name’ param

28.2.61 Salt 2015.8.7 Release Notes

Version 2015.8.7 is a bugfix release for *2015.8.0*.

Note: Salt 2015.8.4, 2015.8.5, and 2015.8.7 were all released within a short period due to regressions found soon after the releases of 2015.8.4 and 2015.8.5. See *here* for the 2015.8.4 release notes, and *here* for the 2015.8.5 release notes.

Statistics

- Total Merges: **2**
- Total Issue References: **1**
- Total PR References: **5**
- Contributors: **4** (gtmanfred, justinta, pass-by-value, terminalmage)

Change to Epoch Support for YUM/DNF

For *pkg.installed* states, on Linux distributions which use yum/dnf, packages which have a non-zero epoch in the version number now require this epoch to be included when specifying an exact version for a package. For example:

```
vim-enhanced:
  pkg.installed:
    - version: 2:7.4.160-1.el7
```

The *pkg.latest_version* and *pkg.list_repo_pkgs* functions can be used to get the correct version string to use, as they will now contain the epoch when it is non-zero.

Changelog for v2015.8.5..v2015.8.7

Generated at: 2018-05-28 00:17:59 UTC

- **PR** #31111: (justinta) Fixes failing npm test on arch. @ *2016-02-10 21:51:47 UTC*
  - 8d84c636cf Merge pull request #31111 from jtand/8_4_npm_fix
  - b0a48e5ef2 Fixes failing npm test on arch.
  - **PR** #30217: (pass-by-value) Make sure cloud actions can be called via salt run
- **ISSUE** #31014: (gtmanfred) [2015.8] pkg breaks for yum pkgs.latest if the packages has an epoch (refs: #31031, #31015)
- **PR** #31092: (terminalmage) Apply PR #31031 to 2015.8.4.follow_up @ *2016-02-10 20:54:37 UTC*
  - **PR** #31031: (terminalmage) More complete fix for #31014 (refs: #31092)
  - **PR** #31015: (gtmanfred) include possible epoch in version for rpm (refs: #31031)
  - 5a6a93e98b Merge pull request #31092 from terminalmage/issue31014-2015.8.4.follow_up
    - 2767a4e519 Don't handle epoch specially for dnf
    - e5dfcc0ef2 More efficient way to add the epoch before version number
    - ed7462793c include possible epoch in version for rpm
  - 6c6b66aedd Comment multiprocessing line in minion config
  - 1f7dfefc4a Set multiprocessing to true in config.py
  - 433c645c20 Fix remove placeholder files
  - 71037560d4 Remove placeholder files
  - 20b381fdf7 Set overwrite to off
  - ca50f56d6c Fix boto_secgroup
  - fd571d23de Fix boto test failures
  - cfb6588744 Fix regression when contents_pillar/contents_grains is a list.
  - 881d8669e3 utils.aws: use time lib to conver to epoch seconds
  - 31412920fc The call to cp.get_url needs the saltenv, if you're using environments other than base, it will fail.
  - a8694014a9 Fix regression in git_pillar when multiple remotes are configured
  - 2243f25be5 Properly set the default value for pillar_merge_lists

  - c7472ff6aa Lint
  - d868711a83 Fix failing boto_vpc module unit tests
  - ed09516469 Fix failing state module tests
  - fd0e940088 Pylint fix
  - bc780a7c25 Don't use pack=pack. Just pass in pack=__salt__ always.
  - 1ae022dbfe Pass in 'pack' variable to utils.boto.assign_funcs function from ALL boto modules.
  - 1efaff107d Remove bad symlinks in osx pkg dirs

28.2.62 Salt 2015.8.8 Release Notes

Version 2015.8.8 is a bugfix release for *2015.8.0*.

Important: Version *2015.8.8.2* was released shortly after 2015.8.8 to fix several known issues. If you installed 2015.8.8 before 03/30/2016, you likely have installed 2015.8.8 and can optionally upgrade (find out which version you have installed using `salt --version`.

Statistics

- Total Merges: **313**
- Total Issue References: **146**
- Total PR References: **312**
- Contributors: **74** (Ch3LL, DmitryKuzmenko, JohannesEbke, RabidCicada, Talkless, The-Loeki, abednarik, anlutro, basepi, bdrung, cachedout, captaininspiration, clarkperkins, clinta, cro, darix, dmacvicar, dr4Ke, dschaller, edencrane, garethgreenaway, gladiatr72, gtmanfred, iacopo-papalini, isbm, jacksontj, jacobhammons, jakehilton, jespada, jfindlay, joejulian, justinta, kiorky, kraney, llua, mcalmer, mchugh19, mew1033, mlalpho, moltob, multani, myii, opdude, paiou, pass-by-value, peripatetic-sojourner, pprince, rallytime, redmcg, replicant0wnz, rhansen, rmtmckenzie, s0undt3ch, sakateka, sbreidba, seanjnkns, sjmh, sjorge, skizunov, szeestraten, tbaker57, techhat, terminalmage, thusoy, ticosax, twangboy, virtualguy, vutny, whiteinge, xmj, xopher-mc, yannis666, youngnick, zygiss)

Security Fix

CVE-2016-3176 Insecure configuration of PAM external authentication service

This issue affects all Salt versions prior to 2015.8.8/2015.5.10 when PAM *external authentication* is enabled. This issue involves passing an alternative PAM authentication service with a command that is sent to *LocalClient*, enabling the attacker to bypass the configured authentication service. Thank you to Dylan Frese <dmfrese@gmail.com> for bringing this issue to our attention.

This update defines the PAM eAuth `service` that users authenticate against in the Salt Master configuration.

Read Before Upgrading Debian 7 (Wheezy) from 2015.8.7 to 2015.8.8

Before you upgrade from 2015.8.7 on Debian 7, you must run the following commands to remove previous packages:

```
sudo apt-get remove python-pycrypto
sudo apt-get remove python-apache-libcloud
```

Note that `python-pycrypto` will likely remove `python-apache-libcloud`, so the second command might not be necessary. These have been replaced by `python-crypto` and `python-libcloud` with ~bpo70+1 moniker.

Read Before Upgrading Debian 8 (Jessie) from Salt Versions Earlier than 2015.8.4

Salt `systemd` service files are missing the following statement in these versions:

```
[Service]
KillMode=process
```

This statement must be added to successfully upgrade on these earlier versions of Salt.

Changelog for v2015.8.7..v2015.8.8

Generated at: 2018-05-28 00:23:11 UTC

- **PR** #31964: (jfindlay) update 2015.8.8 release notes @ *2016-03-17 21:22:04 UTC*
    - b9d0336cf8 Merge pull request #31964 from jfindlay/2015.8
    - b984659678 update 2015.8.8 release notes
- **ISSUE** #31586: (frogunder) Proxy minion service.modules fails (refs: #31601)
- **ISSUE** #31585: (frogunder) Proxy minion commands causing exceptions (refs: #31601)
- **PR** #31947: (cro) Move proxymodule assignment earlier in proxy minion init @ *2016-03-17 18:14:23 UTC*
    - **PR** #31601: (cro) Proxy fixes for #31585 and #31586
    - fefb694104 Merge pull request #31947 from cro/bp-31601
    - 4eb193edb7 Lint, unrelated but fixed anyway.
    - d661081016 Lint.
    - 59e0a6f923 Dont add this file
    - c68b968403 Old-style proxymodules need to be setup earlier in minion init. Also include more correct comments in config.py
- **PR** #31948: (rallytime) Revert "not not" deletion and add comment as to why that is there @ *2016-03-17 17:00:22 UTC*
    - a86490ee68 Merge pull request #31948 from rallytime/disable-pylint-error
    - 86196cd59d Revert "not not" deletion and add comment as to why that is there
- **PR** #31952: (rallytime) Fix lint for 2015.8 branch @ *2016-03-17 16:59:49 UTC*
    - db3af864ae Merge pull request #31952 from rallytime/lint-2015.8
    - 3e964ec9d4 Fix lint for 2015.8 branch
- **PR** #31933: (rallytime) Fix linking syntax in testing docs @ *2016-03-17 14:44:13 UTC*
    - 9ab4d6164b Merge pull request #31933 from rallytime/fix-test-links
    - 06dd2c0411 Fix linking syntax in testing docs

- **ISSUE** #31586: (frogunder) Proxy minion service.modules fails (refs: #31601)
- **ISSUE** #31585: (frogunder) Proxy minion commands causing exceptions (refs: #31601)
- **PR** #31930: (cro) Backport changes from 2016.3 @ *2016-03-16 22:12:29 UTC*
  - **PR** #31601: (cro) Proxy fixes for #31585 and #31586
  - 723d0ca19f Merge pull request #31930 from cro/bp-31601
  - aa9a288b5a Add these files back in
  - 916ef26957 Remove .orig file mistakenly added, reformat example.
  - 3c8185571d Lint.
  - 9de9b9e86d Missin import
  - d571f3b8fe Backport PR'#31601'_
- **PR** #31924: (jfindlay) update 2015.8.8 release notes @ *2016-03-16 22:10:15 UTC*
  - ce765ad2df Merge pull request #31924 from jfindlay/2015.8
  - 64dd8aebb2 update 2015.8.8 release notes
- **ISSUE** #31890: (damon-atkins) salt/fileclient.py get_url should include the URL in any error message (refs: #31922)
- **PR** #31922: (cachedout) For 2015.8 head @ *2016-03-16 19:07:11 UTC*
  - 390ef9fea7 Merge pull request #31922 from cachedout/issue_31890_1
  - da075d9341 For 2015.8 head
- **PR** #31904: (rallytime) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2016-03-16 17:23:54 UTC*
  - 03e8b72655 Merge pull request #31904 from rallytime/merge-2015.8
  - f8b4b1b211 last pylint!
  - 892591a39c More pylint fixes
  - 35b2076584 Pylint fixes
  - 1a1ce05186 Merge branch '2015.5' into '2015.8'
    - 440e0dcbe0 Merge pull request #31825 from jtand/udpate_pylintrc
      - 9a14e02766 Updated beacons/sh.py to work with enumerate()
      - 0ecec691a0 Adjusted beacons to work with enumerate better
      - f509b4113e Fixed final lint error
      - 5945b3f11f Fix and disable pylint errors
      - 06ae6eaf55 Fixed pylint errors on jboss state and module
      - de96db97c8 Fixed more pylint errors, and disabled some more
      - c07b0a20b5 Merge branch 'lint_fixes' into udpate_pylintrc
      - 2e6a152308 Fixed lint error in lxc.py
      - 908ca1a439 Fixed lint error in ssh_py_shim
      - 404c1b50f7 Changed range(len()) to enumerate()
      - 1e13586546 Changed range(len()) to enumerate()

      - 9ccce7a9a5 Added more disables
      - 9c1aab3b4e Updated .testing.pylintrc to match newer versions of pylint
    - 471c9444a3 Merge pull request #31900 from rallytime/fix-psutil-warning
      - 22403d69ae Add "python module" clarification to ps __virtual__ warning.
    - c44c1b5e59 Merge pull request #31878 from rallytime/fix-psutil-warning
      - 44b29f72a1 Make sure __virtual__ error message is helpful when psutil is missing
    - 5c592b6768 Merge pull request #31852 from rallytime/merge-2015.5
      - 1470de17fa Merge branch '2014.7' into '2015.5'
      - 218c902091 Merge pull request #31834 from jfindlay/2014.7
      - 358fdad0c8 add 2014.7.8 release notes
      - a423c6cd04 Merge pull request #31833 from jfindlay/2014.7
      - 6910fcc584 add 2014.7.9 release notes
      - c5e7c03953 Merge pull request #31826 from gtmanfred/2014.7
      - d73f70ebb2 Remove ability of authenticating user to specify pam service
    - 0cc1d5db03 Merge pull request #31827 from gtmanfred/2015.5
      - 979173b78a Remove ability of authenticating user to specify pam service
    - 8cf0b9eb3d Merge pull request #31810 from whiteinge/saltenv-jinja-var
      - cb72b19240 Fix outdated Jinja 'env' variable reference

- **PR** #31906: (sbreidba) Win_dacl module: fix FULLCONTROL / FILE_ALL_ACCESS definition @ *2016-03-16 15:20:19 UTC*
  - a4b3462346 Merge pull request #31906 from sbreidba/win_dacl_fixes
  - 54d81b9b42 Fix FULLCONTROL / FILE_ALL_ACCESS definition (bugfix and code simplification). Use consistent mechanism fro obtaining user SID. Allow wildcarding (via optional parameters) for a variety of methods (get, rm_ace, check_ace).
- **PR** #31745: (isbm) Fix the always-false behavior on checking state @ *2016-03-15 23:02:20 UTC*
  - b068eaa963 Merge pull request #31745 from isbm/isbm-always-minion-errcode-2-fix
  - 1882e1c960 Adjust test
  - f96c8f9b5e Keep first level away from lists.
  - baaed005b8 Fix PEP8 continuation
  - 1db61ea59a Fix the always-false behavior on checking state (there are always lists at some point!)
- **PR** #31911: (rallytime) Merge #31903 with pylint fix @ *2016-03-15 20:35:35 UTC*
  - **PR** #31903: (terminalmage) Use remote_ref instead of local_ref to see if checkout is necessary (refs: #31911)
  - d05c3eeba9 Merge pull request #31911 from rallytime/merge-31903-with-pylint
  - 85e5acd11a Merge #31903 with pylint fix
- **PR** #31883: (paiou) Fix scaleway cloud provider and manage x86 servers @ *2016-03-15 20:31:18 UTC*
  - 819a4a8b54 Merge pull request #31883 from mvpstars/scaleway-x86

  - 1662a080e1 Update scaleway cloud provider to manage x86 servers
- **PR** #31903: (terminalmage) Use remote_ref instead of local_ref to see if checkout is necessary (refs: #31911) @ *2016-03-15 20:04:56 UTC*
  - 142c47c50d Merge pull request #31903 from terminalmage/fix-git-pillar
  - af29940e1c Use remote_ref instead of local_ref to see if checkout is necessary
- **PR** #31845: (sakateka) Now a check_file_meta deletes temporary files when test=True @ *2016-03-15 19:55:21 UTC*
  - ffd65c36e5 Merge pull request #31845 from sakateka/check_file_meta_clean_tmp
  - 5b30336b89 Now a check_file_meta deletes temporary files when test=True
- **ISSUE** #31791: (alexbleotu) Proxy minion starts spinning after running state.highstate (refs: #31846)
- **ISSUE** #31728: (bgridley) Custom grains syncing problem with proxy minion which causes high CPU utilization (refs: #31846)
- **PR** #31901: (rallytime) Back-port #31846 to 2015.8 @ *2016-03-15 19:12:43 UTC*
  - **PR** #31846: (cro) Proxy infinite loop (refs: #31901)
  - 7428c73724 Merge pull request #31901 from rallytime/bp-31846
  - 1edd6ce302 Extra comment.
  - 6c2ef03b11 Fix event bus flood caused by unexpected recursive call.
- **PR** #31905: (terminalmage) Update versionadded directive @ *2016-03-15 18:43:06 UTC*
  - 37f1ce9be2 Merge pull request #31905 from terminalmage/update-versionadded
  - dcc196c9e1 Update versionadded directive
- **PR** #31902: (rallytime) Update versionadded tag for new funcs @ *2016-03-15 18:41:08 UTC*
  - **PR** #31857: (sjorge) gen_password and del_password missing from solaris_shadow (refs: #31902)
  - 35f6407d11 Merge pull request #31902 from rallytime/update-version-31857
  - 5cd09150cd Update versionadded tag for new funcs
- **PR** #31888: (terminalmage) Fix salt.utils.decorators.Depends @ *2016-03-15 17:09:54 UTC*
  - 1be9c91761 Merge pull request #31888 from terminalmage/fix-depends-decorator
  - 394410e2b0 Add integration test for depends decorator
  - caa3cc1007 Fix salt.utils.decorators.Depends
- **PR** #31857: (sjorge) gen_password and del_password missing from solaris_shadow (refs: #31902) @ *2016-03-14 20:29:51 UTC*
  - d357e4ea44 Merge pull request #31857 from sjorge/solarish_shadow
  - 38231303f3 .9 release as mentioned by rallytime
  - 3e25f70968 fix version added
  - d768ed25b4 develop, 2016.3 and 2015.8 has missing gen_password and del_password for shadow module
- **PR** #31879: (cro) Clarify some comments @ *2016-03-14 19:59:35 UTC*
  - 1b0b2d3f1a Merge pull request #31879 from cro/idrac_fixes_0314
  - 42ef3a7970 Extra comment.

- **ISSUE** #8927: (brutasse) file state: unable to use *contents_pillar* with *template: jinja* (refs: #31815)
- **ISSUE** #26944: (boltronics) file.managed contents and contents_pillar should support a template rendering engine (refs: #31815)
- **ISSUE** #14664: (jacksontj) Unable to have a template with file.managed contents (or contents_pillar) (refs: #31815)
- **PR** #31815: (dr4Ke) Fix template on contents 2015.8 @ *2016-03-14 17:41:46 UTC*
  - fb81bbea23 Merge pull request #31815 from dr4Ke/fix_template_on_contents_2015.8
  - dcd6f5a5a9 test for file.apply_template_on_contents
  - 10d882296d file.managed: templating contents, not just files
- **PR** #31818: (anlutro) Prevent event logs from writing huge amounts of data @ *2016-03-14 17:27:47 UTC*
  - aa120cb716 Merge pull request #31818 from alprs/fix-event_logging_spam
  - 83fa136da7 work on event logging
- **ISSUE** #31293: (deuscapturus) Git Pillars lose HEAD reference over time (refs: #31836)
- **ISSUE** #29239: (timwsuqld) Occasionaly git_pillar pull fails causing incorrect results of highstate (when running highstate for multiple minions) (refs: #31836)
- **PR** #31836: (terminalmage) Fix git_pillar race condition @ *2016-03-14 15:48:28 UTC*
  - f2445bdbdc Merge pull request #31836 from terminalmage/issue31293
  - 5048fa857c Fix duplicate output
  - 155b84b88a salt.fileserver: Add ability to clear checkout locks
  - af410d8dd1 Pass through the lock_type
  - 3d7796d5dd salt.runners.cache: Add ability to clear checkout locks
  - 8e086099f5 salt.utils.gitfs: rewrite locking code
  - 06b212519c Add GitLockError exception class
  - ad04ccfb93 Strip whitespace when splitting
- **PR** #31824: (rallytime) Back-port #31819 to 2015.8 @ *2016-03-13 19:59:32 UTC*
  - **PR** #31819: (mchugh19) raise error on unsupported distro (refs: #31824)
  - 5464be07b1 Merge pull request #31824 from rallytime/bp-31819
  - 4d516adade raise error on unsupported distro
- **ISSUE** #24559: (iacopo-papalini) salt-cloud - Azure - should be possible to specify virtual network & subnet in profile (refs: #31856, #24569)
- **PR** #31856: (szeestraten) Adds missing docs for Virtual Network and Subnet options in salt-cloud Azure cloud profile @ *2016-03-13 19:06:52 UTC*
  - **PR** #24569: (iacopo-papalini) Fix Issue #24559 - salt-cloud - Azure - should be possible to specify… (refs: #31856)
  - 7781b357e0 Merge pull request #31856 from szeestraten/add-missing-docs-for-azure-cloud-profile
  - a1a2229405 Adds missing docs for Azure cloud profile
- **PR** #31839: (jfindlay) add 2015.8.8 release notes @ *2016-03-11 23:23:34 UTC*
  - 3f88f3a8cf Merge pull request #31839 from jfindlay/2015.8

  - 47ac41ba27 add 2015.8.8 release notes
- **PR** #31828: (gtmanfred) Remove ability of authenticating user to specify pam service @ *2016-03-11 20:40:37 UTC*
  - 46bdd10a56 Merge pull request #31828 from gtmanfred/2015.8
  - 7c3134a3d3 Remove ability of authenticating user to specify pam service
- **ISSUE** #30489: (chris-martin) influxdb_user.present fails: "InfluxDBClient' object has no attribute 'get_list_cluster_admins" (refs: #31787, #31770)
- **PR** #31787: (anlutro) Fix user_create and db_create for new versions of influxdb @ *2016-03-11 15:19:22 UTC*
  - 3d370b471c Merge pull request #31787 from alprs/fix-influxdb_user
  - 6a5211c8d8 don't swallow exceptions
  - a7e9c1e381 fix db_create for influxdb 0.9+
  - 5a8a645d4b fix create_user for new versions of influxdb
- **PR** #31800: (rallytime) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2016-03-10 20:49:53 UTC*
  - 7fb2331ebc Merge pull request #31800 from rallytime/merge-2015.8
  - 44c15f0b16 Merge branch '2015.5' into '2015.8'
  - 970ef0e445 Merge pull request #31744 from brejoc/fix-attribute-error-with-older-libcloud/2015.5
    - bb29dc2283 Added version to libcloud depends statement
    - 87f9534fce Added log message with update suggestion for libcloud
    - 72eab406cd Fix for AttributeError with libcloud <0.15
  - df2d23ba5d Merge pull request #31740 from terminalmage/issue31666
    - aeaf5864cd Fall back to False when pillar_opts not set
    - fe19d77eb4 Add default value for pillar_opts on minion
  - e22f5c0a26 Merge pull request #31750 from rallytime/bp-26170
    - 3c11234a05 Make sure variable is a dictionary before popping something from it.
  - 9162925dd0 Merge pull request #31689 from rallytime/bp-29467
    - 1f8f4cb99b Update module.py
- **PR** #31797: (Ch3LL) Change pkg name to less for suse pkg.info_installed test @ *2016-03-10 19:08:16 UTC*
  - 75dfb2ed40 Merge pull request #31797 from Ch3LL/fix_pkginfo_test
  - 910f0d9ffc change pkg name to less for suse
- **ISSUE** #31617: (tampakrap) service.running fails on sle11 sp3 and sp4 (refs: #31629, #31793)
- **PR** #31793: (xopher-mc) fixing init system detection on sles 11, refs #31617 @ *2016-03-10 18:42:27 UTC*
  - 1386b72bbf Merge pull request #31793 from xopher-mc/fix_sles_state_service_module
  - d242cb19b4 fixing init system dectection on sles 11, refs #31617
- **PR** #31786: (isbm) Bugfix: zypper doesn't detect base product on SLE11 series @ *2016-03-10 18:12:46 UTC*
  - 2f28c166dd Merge pull request #31786 from isbm/isbm-zypper-list-products-sles11
  - ee1a002673 Update test case to cover SLE11 and SLE12
  - 4b134fb2ab Add SLE11 product info snapshot, rename previous

  - 3c5fc857b2 Bugfix: on SLE11 series base product reported as additional
- **ISSUE** #31776: (gtmanfred) ProxyMinion does not close connections (at least with esxi proxy) (refs: #31780)
- **PR** #31780: (gtmanfred) use already created vsphere connection @ *2016-03-10 17:41:53 UTC*
  - d6f669623c Merge pull request #31780 from gtmanfred/2015.8
  - 070eaf07f0 use already created vsphere connection
- **ISSUE** #31772: (sbreidba) win_dacl state causes state.apply output to be YAML, not highstate (refs: #31779)
- **PR** #31779: (sbreidba) win_dacl state & module: return comment field as strings, not lists. @ *2016-03-10 17:41:08 UTC*
  - a067de3712 Merge pull request #31779 from sbreidba/win-dacl-highstate-output-2015.8
  - aeb2bfcf46 win_dacl state & module: return comment field as strings, not lists.
- **ISSUE** #31563: (sjorge) regression in 2016.3 from today? (refs: #31723, #31707)
- **PR** #31723: (sjorge) file_ignore_regex is a list, not bool @ *2016-03-09 23:36:10 UTC*
  - **PR** #31707: (sjorge) Fix incorrect default types for master_tops and file_ignore_regex (refs: #31723)
  - baeefac252 Merge pull request #31723 from sjorge/2015.8-file_ignore_regex
  - df1ba94cbb file_ignore_regex is a list, not bool
- **ISSUE** #27960: (The-Loeki) salt-cloud CLI 2015.8 borks out with SaltClientError: 'timeout' (refs: #31747)
- **PR** #31747: (techhat) Use get_local_client with MASTER opts, not MINION @ *2016-03-09 23:14:58 UTC*
  - cd43cf919c Merge pull request #31747 from techhat/issue27960
  - 44c100d610 Use get_local_client with MASTER opts, not MINION
- **PR** #31688: (whiteinge) Various SMTP returner fixes @ *2016-03-09 22:40:37 UTC*
  - 286ea1f61b Merge pull request #31688 from whiteinge/smtp-renderer
  - 76671b6a81 Check if we have a StringIO and grab the string instead
  - 17b8cd755f Add a default for the subject
  - 26479bee24 Clean up the SMTP returner docstring and show an actual config example
  - 74563f17ed Make sure the email subject and body are strings
  - fc69d08e8e Default to just 'jinja' for the SMTP renderer
  - 2af7cd2789 Add missing 'port' to smtp options
- **PR** #31752: (rallytime) Back-port #31686 to 2015.8 @ *2016-03-09 21:23:01 UTC*
  - **PR** #31686: (myii) Fix typo in example for section *winrepo_dir_ng* (refs: #31752)
  - 1d6d982e5c Merge pull request #31752 from rallytime/bp-31686
  - e4df5d9a55 Fix typo in example for section *winrepo_dir_ng*
- **PR** #31733: (jacobhammons) docs to clarify cloud configuration @ *2016-03-09 20:54:10 UTC*
  - ec90294442 Merge pull request #31733 from jacobhammons/cloud-docs
  - 209c641a41 Made udpates as suggested by @rallytime
  - 26d4991cb3 moved previous intro to new quick start topic (topics/cloud/qs.rst) added new intro that explains the salt cloud configuration files added an inheritance and minion startup state example to topics/cloud/config.rst

- **ISSUE** #26498: (rallytime) [salt-cloud] Able to create multiple VMs with the same name across providers (refs: #31754, #31775)
- **PR** #31775: (techhat) Show correct provider/driver name @ *2016-03-09 20:53:10 UTC*
  - 92ba7f3495 Merge pull request #31775 from techhat/correctmsg
  - c1433650b4 Show correct provider/driver name
- **ISSUE** #26498: (rallytime) [salt-cloud] Able to create multiple VMs with the same name across providers (refs: #31754, #31775)
- **PR** #31754: (techhat) Check all providers, not just the current one @ *2016-03-09 18:38:19 UTC*
  - 249a3602eb Merge pull request #31754 from techhat/issue26498
  - 08c61446b7 Check all providers, not just the current one
- **ISSUE** #31639: (mshirley) salt-cloud digital ocean api v2 doesn't implement all available actions (refs: #31735)
- **PR** #31735: (rallytime) Add reboot, start, and stop actions to digital ocean driver @ *2016-03-09 17:57:58 UTC*
  - 7ad521f7a5 Merge pull request #31735 from rallytime/fix-31639
  - 67d1aa6740 Remove experimental/incomplete function
  - b209623ca9 Add reboot, start, and stop actions to digital ocean driver
- **ISSUE** #30489: (chris-martin) influxdb_user.present fails: "InfluxDBClient' object has no attribute 'get_list_cluster_admins" (refs: #31787, #31770)
- **PR** #31770: (anlutro) Fix influxdb user functionality for version 0.9+ @ *2016-03-09 17:09:26 UTC*
  - fd3610c6a4 Merge pull request #31770 from alprs/fix-influxdb_user
  - 1349bdd2e8 fix influxdb user functionality for version 0.9+
- **PR** #31743: (Talkless) Fix parentheses missmatch in documentation @ *2016-03-08 18:01:23 UTC*
  - c0868307df Merge pull request #31743 from Talkless/patch-1
  - 26ff46dbc6 Fix parenthesis missmatch in documentation
- **PR** #31162: (isbm) Remove MD5 digest from everywhere and default to SHA256 @ *2016-03-07 19:11:36 UTC*
  - 826fea6582 Merge pull request #31162 from isbm/isbm-md5-to-sha1
  - 9d64abed0c Fix PyLint
  - 327ea11139 Add daemons unit test to verify hash_type settings
  - f3aecc0b22 Standardize logging
  - 51f556243d Verify if hash_type is using vulnerable algorithms
  - 95ec634f00 Report environment failure, if any
  - 63eedefe54 Use mixin for the daemon classes
  - 82dd383630 Create a mixin class that will be reused in the similar instances (daemons)
  - 36da8f5efa Use MD5 hash algorithm by default (until deprecated)
  - 584325797c Remove SHA1 in favor of SHA256
  - 373493c13f Remove SHA1 for SHA256
  - d5cb4dd424 Remove sha1 to sha265
  - 73b8d35e01 Add note to the Tomcat module for SHA256

  - efb78f1055 Remove SHA1 to SHA265 by default
  - 6198976edb Use SHA1 by default instead of MD5
  - 73f2df76ce Use SHA1 hash by default in Tomcat module, refactor for support different algorithms
  - 0d4e4e31f8 Use SHA1 hash by default
  - 785717703b Use configurable hash_type for general Key fingerprinting
  - f0d931f4d0 Use hash_type configuration for the Cloud
  - 95cb59dec7 Set defalt hash as SHA1 in config and explain why.
  - 8f9543c292 Set config hash_type to SHA1
  - 413eca124d Set default checksum for key fingerprint to SHA1
- **ISSUE** #30528: (UtahDave) Missing Minion notifications missing from job cache (refs: #31670)
- **PR** #31670: (terminalmage) Write lists of minions targeted by syndic masters to job cache @ *2016-03-07 18:51:53 UTC*
  - a1f32b71bd Merge pull request #31670 from terminalmage/issue30528
  - 65e5a3c53e Pass syndic_id to save_minions()
  - cf94c2597a Add argument to save_minions() to pass a syndic ID
  - cb92114377 Add syndic_id param for API compatibility
  - 1d39eec69b Skip events with minion lists but no jid
  - 651e3926f7 lint fixes
  - 0f175a4edf salt.returners.sqlite3_return: add no-op save_minions() func for API compatibility
  - f8664103b1 salt.returners.redis_return: add no-op save_minions() func for API compatibility
  - 0ea1b76c22 salt.returners.postgres_local_cache: add no-op save_minions() func for API compatibility
  - d6d794b484 salt.returners.postgres: add no-op save_minions() func for API compatibility
  - 82750ab699 salt.returners.pgjsonb: add no-op save_minions() func for API compatibility
  - d8f90f6578 salt.returners.odbc: add no-op save_minions() func for API compatibility
  - a1957c3706 salt.returners.mysql: add no-op save_minions() func for API compatibility
  - ef6aa5de1c salt.returners.multi_returner: add no-op save_minions() func for API compatibility
  - 5b4eb58d99 salt.returners.mongo_return: add no-op save_minions() func for API compatibility
  - da1acbb8f2 salt.returners.mongo_future_return: add no-op save_minions() func for API compatibility
  - c13bb6549c salt.returners.memcache_return: add no-op save_minions() func for API compatibility
  - 4322ad9ef3 salt.returners.influxdb_return: add no-op save_minions() func for API compatibility
  - 1dd106183c salt.returners.etcd_return: add no-op save_minions() func for API compatibility
  - 8e80535516 salt.returners.couchdb_return: add no-op save_minions() func for API compatibility
  - 44538dfced salt.returners.cassandra_cql_return: add no-op save_minions() func for API compatibility
  - 084a78407a salt.returners.couchbase_return: move minion list updates to new save_minions() func
  - f731dc5d32 Update a job's minion list to include minion lists forwarded by syndic
  - 504f7df460 Add utils function to invoke a returner's save_minions() func

  - 0b4616a3eb Separate writing of serialized minion list into its own function
  - 214fedc3f6 Simplify jobs.get_jobs logic, generally improve jobs runner docs
  - 3f527be748 Add an exception class for errors encountered while locking files.
  - 1e6b43eef8 Add a contextmanager for file locking
  - 978b6cb32f Add missing RST file for slsutil module
  - 2ad8ceffc2 Add salt.utils.split_input()
- **ISSUE** #31595: (dverbeek84) dockerng ports specified in Dockerfile must be in sls file otherwise salt gives an error (refs: #31711)
- **PR** #31711: (ticosax) [dockerng] Port and Volume comparison should consider Dockerfile @ *2016-03-07 18:25:19 UTC*
  - 24568b1a5d Merge pull request #31711 from ticosax/fix-port-and-volume-discovery
  - cf38691597 Port and Volume comparison should consider Dockerfile
- **ISSUE** #31579: (bradthurber) salt-cloud delete with a map file fails when multiple providers defined (refs: #31719)
- **PR** #31719: (techhat) Don't worry about KeyErrors if the node is already removed @ *2016-03-07 18:16:40 UTC*
  - b936e09fb3 Merge pull request #31719 from techhat/issue31579
  - 88905095c9 Don't worry about KeyErrors if the node is already removed
- **PR** #31713: (ticosax) [dockerng] Fix dockerng.network_present when container is given by name @ *2016-03-07 15:14:41 UTC*
  - 604eb87e82 Merge pull request #31713 from ticosax/fix-dockerng-networking-container_id
  - 3837cf44ca Fix network_present by dealing with containers ID's instead of names.
- **ISSUE** #31704: (peripatetic-sojourner) Foreman external pillar doesn't load (refs: #31705)
- **PR** #31705: (peripatetic-sojourner) Foreman pillar @ *2016-03-07 14:24:58 UTC*
  - 8f28e4510d Merge pull request #31705 from peripatetic-sojourner/foreman_pillar
  - ba33d75949 passing lint test
  - 63e39a8999 refactored parameter population for foreman pillar
  - c3325bc15d add return of virtualname
- **PR** #31702: (rallytime) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2016-03-06 19:24:47 UTC*
  - aa5c13f0b8 Merge pull request #31702 from rallytime/merge-2015.8
  - 6559ea15b0 Merge branch '2015.5' into '2015.8'
  - d7914cdb14 Merge pull request #31687 from cachedout/rm_gpg_test
    - 8b00513ebb Removed useless tests
  - bd4d12a155 Merge pull request #31660 from terminalmage/issue31619
    - da954d7b92 Add integration test for packages with epoch in version
    - 4fa7e4defe Move epoch removal
    - 290192af56 Remove epoch from version string if present when installing with yum
  - e33c1f456a Merge pull request #31683 from rallytime/bp-31578

    - 8fe46789b7 allow queueing of state runs through saltmod
  - 27f443895d Merge pull request #31682 from cachedout/cache_meaning
    - a75e146125 Add definition of job cache to glossary
  - bd04c964d1 Merge pull request #31658 from rallytime/add-style-to-contrib
    - 6b526b5878 Add mentioned of Salt's Coding Style docs to the Contributing docs
  - 10658dffe6 Merge pull request #31655 from rallytime/pylint-docs
    - 6e0377d376 Make note of pylint dependencies in docs
  - 6075774a01 Merge pull request #31440 from cachedout/master_tops_type
    - f49cc75049 Set correct type for master_tops config value
- **PR** #31700: (s0undt3ch) It's a function! @ *2016-03-06 17:33:58 UTC*
  - ace290629e Merge pull request #31700 from s0undt3ch/2015.8
  - 1ca2beea3e It's a function!
- **PR** #31679: (cro) Fix bad link to the sample REST endpoint in salt-contrib. @ *2016-03-04 21:05:50 UTC*
  - cf438aa873 Merge pull request #31679 from cro/proxy_contrib_doc_fix
  - d638971b73 Correct url to salt-contrib
- **ISSUE** #21932: (clinta) Salt Coding Style docs should list requirements for salt pylintrc (refs: #31655)
- **PR** #31668: (rallytime) Some more testing documentation improvements @ *2016-03-04 20:48:57 UTC*
  - **PR** #31658: (rallytime) Add mentioned of Salt's Coding Style docs to the Contributing docs (refs: #31668)
  - **PR** #31655: (rallytime) Make note of pylint dependencies in docs (refs: #31668)
  - **PR** #31641: (rallytime) Improve Salt Testing tutorial to be a more comprehensive intro (refs: #31668)
  - 97127a8b83 Merge pull request #31668 from rallytime/testing-docs
  - beb9d0fe84 Ensure all integration test classes and funcs are documented w/examples
  - 7f8ebf7c97 Found another spelling error
  - c8c188535f Spelling fix
  - f260c51762 Some more testing documentation improvements
- **ISSUE** #29753: (jakehilton) New minion fails to authenticate properly to multi-master setup (refs: #31653)
- **PR** #31653: (DmitryKuzmenko) Don't attempt to verify token if it wasn't sent to master. @ *2016-03-03 17:39:35 UTC*
  - 2ed7286af1 Merge pull request #31653 from DSRCompany/issues/29753_multimaster_auth_fail
  - 2557707cc7 Don't attempt to verify token if it wasn't sent to master.
- **ISSUE** #31617: (tampakrap) service.running fails on sle11 sp3 and sp4 (refs: #31629, #31793)
- **PR** #31629: (darix) Fix services on sles @ *2016-03-03 16:41:27 UTC*
  - 118fcde425 Merge pull request #31629 from darix/fix-services-on-sles
  - 9b8d6cbb72 make the suse check consistent with rh_service.py
  - c0c8a77242 Fix numerical check of osrelease
- **PR** #31641: (rallytime) Improve Salt Testing tutorial to be a more comprehensive intro (refs: #31668) @ *2016-03-03 16:08:47 UTC*

  - 4d1701de60 Merge pull request #31641 from rallytime/testing-tutorial
  - 6ab3961748 Improve Salt Testing tutorial to be a more comprehensive intro
- **ISSUE** #30651: (sjorge) salt.states.grains.list_present should not show changes if none are made! (refs: #31651, #30689)
- **PR** #31651: (dr4Ke) test case: test_list_present_nested_already @ *2016-03-03 16:02:55 UTC*
  - **PR** #30689: (sjorge) fix for #30651 grains.list_present and grains.list_absent (refs: #31651, #31271)
  - 584f8401b8 Merge pull request #31651 from dr4Ke/test_case_for_30689
  - fc9dd356e8 test case: test_list_present_nested_already
- **PR** #31643: (opdude) Make sure we are really updating the mercurial repository @ *2016-03-03 14:30:53 UTC*
  - 5566f1f2a7 Merge pull request #31643 from Unity-Technologies/hotfix/hg-fix-repo-updated
  - ca41c4b8c1 Make sure we are really updating the mercurial repository
- **ISSUE** #30761: (sjmh) Cannot target subsets of minions when using pillar and external_auth (refs: #31598)
- **PR** #31598: (terminalmage) Remove limitations on validation types for eauth targets @ *2016-03-02 22:14:41 UTC*
  - 36c790eede Merge pull request #31598 from terminalmage/issue30761
  - 5dedaa2d9d Remove limitations on validation types for eauth targets
- **PR** #31627: (jakehilton) Handling error from using gevent 1.1. @ *2016-03-02 22:01:22 UTC*
  - cc4c31cf7f Merge pull request #31627 from jakehilton/2015.8
  - 02fb5ed616 Handling error from using gevent 1.1.
- **PR** #31630: (rallytime) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2016-03-02 20:49:52 UTC*
  - 191241e71a Merge pull request #31630 from rallytime/merge-2015.8
  - 75bb692990 Merge branch '2015.5' into '2015.8'
  - 6d31b8918f Merge pull request #31622 from jfindlay/query_doc
    - 4e48fec806 doc/topics/tutorials/http: update query decoding docs
  - dbf6e0786c Merge pull request #31558 from cachedout/ensure_ssh_installed
    - cecc6e0a5f Don't stacktrace if ssh binary is not installed with salt-ssh
- **PR** #31594: (rallytime) Back-port #31589 to 2015.8 @ *2016-03-02 16:33:24 UTC*
  - **PR** #31589: (techhat) Ensure that the latest node data is returned (refs: #31594)
  - 38ddd62aef Merge pull request #31594 from rallytime/bp-31589
  - 6cd89459c7 Ensure that the latest node data is returned
- **ISSUE** #31596: (joejulian) gluster –xml does not always produce xml with legacy versions (refs: #31604)
- **PR** #31604: (joejulian) Workaround for non-xml output from gluster cli when not tty @ *2016-03-02 15:53:44 UTC*
  - 86a0fc46b4 Merge pull request #31604 from joejulian/2015.8_31596_workaround_no_xml_when_not_tty
  - c567a823a9 Workaround for non-xml output from gluster cli when not tty
- **PR** #31583: (vutny) Remove trailing white spaces @ *2016-03-02 15:38:01 UTC*
  - 36ce240596 Merge pull request #31583 from vutny/remove-trailing-white-spaces

  - bbcad93a8d Fix trailing white spaces in Salt PRM spec file
  - 86433f2378 Revert changes in files used by *roots_test.py* integration test
  - e7a8dbf498 Remove trailing white spaces in tests files
  - 776b2ea9a6 Remove trailing white spaces in files under *salt/* dir
  - fbfc3abccf Remove trailing white spaces in files under *pkg/* dir
  - aebc48163d Remove trailing white spaces in documentation files
  - 7eaf778695 Remove trailing white spaces in conf dir file
- **PR** #31592: (rallytime) Back-port #31546 to 2015.8 @ *2016-03-01 23:51:02 UTC*
  - **PR** #31546: (terminalmage) Rework of PR #31529 (refs: #31592)
  - **PR** #31529: (llua) nspawn.py: Fix bad keyword assignment (refs: #31546)
  - c9fe8d87f3 Merge pull request #31592 from rallytime/bp-31546
  - 9a296bd1bf Use clean_kwargs and invalid_kwargs utils funcs to handle invalid kwargs
  - 43099a2b63 nspawn.py: Fix bad keyword assignment
- **ISSUE** #30866: (kevinquinnyo) WheelClient cmd returns None but wheel functions called directly work (refs: #31570)
- **ISSUE** #26415: (CaesarC) salt.wheel.WheelClient doesn't work follow the python api(AttributeError: 'NoneType' object has no attribute 'get') (refs: #28087)
- **PR** #31593: (rallytime) Back-port #31570 to 2015.8 @ *2016-03-01 23:50:05 UTC*
  - **PR** #31570: (cro) Need to return the value (refs: #31593)
  - **PR** #28087: (DmitryKuzmenko) Revert "Update __init__.py" (refs: #31570)
  - c8dbc93ac6 Merge pull request #31593 from rallytime/bp-31570
  - b2294d0a28 Need to return the value
- **ISSUE** #28585: (robthralls) FIPS compliance (2015.8.1-1) (refs: #31567)
- **PR** #31567: (cachedout) Restore FIPS compliance when using master_finger @ *2016-03-01 19:50:03 UTC*
  - 068807558a Merge pull request #31567 from cachedout/issue_28585
  - 7006a1eecf Fix failed unit test
  - 10cd328dda Lint
  - 174337d020 Restore FIPS compliance when using master_finger
- **PR** #31568: (twangboy) Grant permissions using SID instead of name @ *2016-03-01 04:22:53 UTC*
  - 77d9aae8bb Merge pull request #31568 from twangboy/fix_perms
  - 1f6a95694d Grant permissions using SID instead of name
- **ISSUE** #31516: (justinta) beacons.enable_beacon does not write to beacons.conf on some OS's (refs: #31561)
- **PR** #31561: (justinta) Skipped test @ *2016-03-01 04:11:22 UTC*
  - ada5ab344d Merge pull request #31561 from jtand/beacons_test_fix
  - 196dd4db99 Skipped test
- **ISSUE** #31041: (fredrikaverpil) Reading about win_service in 2015.8.5 docs, but it's not available in 2015.8.5 (refs: #31550, #31049)

- **PR** #31550: (rallytime) Correct versionadded tag for win_service.config @ *2016-02-29 21:11:24 UTC*
  - **PR** #31049: (twangboy) Fix versionadded in win_service.config (refs: #31550)
  - 658c1865ab Merge pull request #31550 from rallytime/win_service-docs
  - 51aa26334c Correct versionadded tag for win_service.config
- **PR** #31549: (rallytime) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2016-02-29 18:46:35 UTC*
  - 9bb9a54f8d Merge pull request #31549 from rallytime/merge-2015.8
  - b683df9b82 Pylint fix
  - 24505d2dcf Merge branch '2015.5' into '2015.8'
    - 060a60fd90 Merge pull request #31521 from terminalmage/issue24753
      - 0d352bbc16 Add fileclient tests
      - d9370a8041 Update cp module salt-ssh wrapper to use new cachedir param
      - 0320494b1d Update the SSH state module wrappers to pass an alternate cachedir
      - 65bdcb3afa Accept and pass through the alternate cachedir when prepping the thin tar
      - c3f7a2f2e5 Add ability to specify an alternate base dir for file caching
    - 92f8f89218 Merge pull request #31497 from rallytime/remove-timeout-dup
      - 83e6480d20 Remove duplicate "timeout" definition in Roster docs
    - da001bcb49 Merge pull request #31472 from rallytime/update-contributing-docs
      - 5871e4d1e0 Update contributing docs
    - f35e2dd1d3 Merge pull request #31461 from DSRCompany/issues/30183_fix_multimaster_failover_2015.5
      - 3d09c3b7a3 Set auth retry count to 0 if multimaster mode is failover.
- **ISSUE** #29701: (tonyyang132) Running salt-call on salt master would crash the master node with code level 2015.8.3 (refs: #31544)
- **ISSUE** #27063: (lorengordon) 2015.8.0: Error writing to */var/log/salt/minion*? (refs: #31544)
- **PR** #31544: (DmitryKuzmenko) Protect getattr from recursion @ *2016-02-29 17:48:15 UTC*
  - 5a6aff1791 Merge pull request #31544 from DSRCompany/issues/29701_getattr_recursion_protection
  - b7a45b8fae Protect getattr from recursion
- **ISSUE** #30643: (Ch3LL) multi-master failover stack trace when minion fails over to other master (refs: #31512, #31525)
- **ISSUE** #30181: (jakehilton) Minion failover only works once (refs: #31512)
- **ISSUE** #29567: (freebsdly) multi master failover successful but execute command 'salt * test.ping' on second master return 'Minion did not return. [No response]' (refs: #31512)
- **PR** #31525: (DmitryKuzmenko) Issues/30643 merge forward fixes @ *2016-02-29 16:08:47 UTC*
  - **PR** #31512: (DmitryKuzmenko) Don't fork in try with critical finally logic. (refs: #31525)
  - **PR** #30796: (skizunov) Fix minion failover after disconnect (refs: #31512, #31525)
  - d5a4daa17b Merge pull request #31525 from DSRCompany/issues/30643_merge_forward_fixes
  - a50b33d96a Don't fork in try with critical finally logic.
  - 877bc25381 Fix minion failover after disconnect

- **ISSUE** #24955: (damonnk) Minion fails to start after bootstrap on Raspberry PI (refs: #31536)
- **PR** #31536: (virtualguy) Remove debian repo from raspbian installation @ *2016-02-29 15:32:52 UTC*
  - 95af21325f Merge pull request #31536 from virtrnd/remove-jessie-backports-from-raspbian-install
  - e48900ac55 Use python-tornado from jessie-backports for pure debian
  - 6e338e2601 Remove debian repo from raspbian installation
- **ISSUE** #31193: (gwaters) RHEL7 gpg key problem (refs: #31528)
- **PR** #31528: (vutny) Correct Salt Cloud documentation about updating Salt Bootstrap script @ *2016-02-29 15:30:59 UTC*
  - 5965319600 Merge pull request #31528 from vutny/cloud-bootstrap-doc
  - f7beeb69f2 Correct Salt Cloud documentation about updating Salt Bootstrap script
- **ISSUE** #31365: (cwicklein) osrelease_info broken for CentOS 7 (refs: #31539)
- **PR** #31539: (DmitryKuzmenko) Added temporary workaround for CentOS 7 os-release id bug. @ *2016-02-29 15:30:34 UTC*
  - 96c0926298 Merge pull request #31539 from DSRCompany/issues/31365_centos7_osrelease_fix
  - a3b806d126 Added temporary workaround for CentOS 7 os-release id bug.
- **PR** #31508: (mcalmer) Zypper correct exit code checking @ *2016-02-26 15:21:23 UTC*
  - 95db870325 Merge pull request #31508 from mcalmer/zypper-correct-exit-code-checking
  - 66e8f6aa37 restructure the code a bit
  - f5c125de19 remove new lines between zypper command and check result
  - 1425c6496c use specialized assert functions for tests
  - f266cfdaac test _zypper_check_result()
  - aff6467782 adapt tests to new zypper_check_result() output
  - edad780cdf use _zypper_check_result()
  - 7c5d5a2b7a add _zypper_check_result() to raise and error or return stdout
  - a6785ef7a9 check zypper exit code everywhere
  - 935b0510c9 add function to check zypper exit codes
- **ISSUE** saltstack/salt-bootstrap#695: (mtippett) Install Failures With Raspbian Jessie (refs: #31510, #31477, #31458)
- **PR** #31510: (vutny) Add installation guide for Raspbian (Debian on Raspberry Pi) @ *2016-02-26 15:06:57 UTC*
  - e51126179c Merge pull request #31510 from vutny/debian-raspbian-install-guide
  - 50f3e072b1 Add instruction how to install salt-minion on Debian Jessie from Stretch
  - abcd505178 Update Debian installation guide with information about Raspbian
- **PR** #31498: (Ch3LL) rename methods in pkg states test @ *2016-02-25 23:29:51 UTC*
  - 9d458bb420 Merge pull request #31498 from Ch3LL/rename_test
  - 29a53f4353 rename methods in pkg states test
- **ISSUE** #31427: (githubcdr) salt.states.grains.list_present adds duplicates names (refs: #31471)

- **PR** #31471: (cachedout) Correct issue where duplicate items in grains list during state run will result in duplicate grains @ *2016-02-25 20:15:20 UTC*
  - 625da0d261 Merge pull request #31471 from cachedout/issue_31427
  - 74c3053c91 Remove debugging
  - 30eb5fccf7 Additional tests, but disable the test for distinct lists, because of a problem with context in test suite
  - 3d2aec05e5 Check for duplicate grains during list insertion
- **ISSUE** #29727: (oeuftete) dockerng.running does not pull image as documented (refs: #31352, #31455)
- **PR** #31455: (ticosax) [dockerng] Disable notset check @ *2016-02-25 19:15:09 UTC*
  - e85ae2341a Merge pull request #31455 from ticosax/diable-NOTSET-check
  - e072937243 dockerd returns sometimes *None* or *[]* for ports.
  - 5630401889 _api_mismatch was a good idea
- **PR** #31488: (isbm) Unit Test for Zypper's "remove" and "purge" @ *2016-02-25 17:52:33 UTC*
  - e68a0947b7 Merge pull request #31488 from isbm/isbm-zypper-ut-removepurge
  - d30f2e4627 Implement unit test for remove and purge
  - 4caf201052 Refactor code (a bit)
  - df89da4d15 Fix the docstring
- **PR** #31485: (jacobhammons) Fixed transport description in minion / master config @ *2016-02-25 17:04:15 UTC*
  - cd87760c87 Merge pull request #31485 from jacobhammons/2015.8
  - 748acab8b5 Fixed zeromq casing in transport settings
  - 765a226907 Fixed transport description in minion / master config
- **PR** #31411: (justinta) Added some beacons execution module integration tests @ *2016-02-25 16:16:26 UTC*
  - fb1ef92e2b Merge pull request #31411 from jtand/beacons_tests
  - 7d32b56015 Added some more checks to verify beacon changes were actually happening
  - 2da5285c03 Added codeauthor
  - fd1e2838ea Lint
  - 421a112914 Added config_dir to test minion config
  - bf6a4c0983 Fixed lint error
  - 3566fbbcca More updates to beacons test
  - 37c4bf22d2 Updated beacons integration test
  - 6db628be1a Basic integration tests for beacons execution module
  - 3b238c2e68 Started adding beacons execution module tests
- **ISSUE** #31216: (oliver-dungey) pkg.installed documentation not consistent with implementation (refs: #31475)
- **ISSUE** #30464: (sjmh) pillar_env minion config option needs to be documented (refs: #31475)
- **ISSUE** #30261: (MadsRC) Add ability to define custom beacons (refs: #31475)
- **ISSUE** #29636: (ronnix) Documentation for the refresh_password arg in postgres_user.present is confusing (refs: #31475)

- **ISSUE** #29528: (apergos) nitpick for "Using Salt at scale" tutorial (refs: #31475)
- **ISSUE** #29520: (arthurlogilab) [doc] transport option not in default master configuration nor in the example file of the documentation (refs: #31475)
- **ISSUE** #10330: (jhenry82) exclude keyword not working (refs: #31475)
- **PR** #31475: (jacobhammons) Assorted doc issues @ *2016-02-25 16:03:54 UTC*
  - 2e9a705e75 Merge pull request #31475 from jacobhammons/2015.8
  - a72dc15720 Assorted doc issues Fixes #10330 Fixes #31216 Fixes #30464 Fixes #29520 Fixes #30261 Fixes #29636 Fixes #29528
- **ISSUE** saltstack/salt-bootstrap#695: (mtippett) Install Failures With Raspbian Jessie (refs: #31510, #31477, #31458)
- **PR** #31477: (vutny) Correct installation documentation for Ubuntu @ *2016-02-25 16:01:38 UTC*
  - 3905dd81d3 Merge pull request #31477 from vutny/correct-doc-install-ubuntu
  - 172f34a6ca Correct headers in Debian/Ubuntu/SUSE install instructions
  - 4248f9ea0a Add common packages installation section to Debian install guide
  - 8c6e179870 Add note about *amd64* packages to Ubuntu install guide
  - afaa24723a Update Ubuntu install guide:
- **PR** #31479: (isbm) Zypper unit tests & fixes @ *2016-02-25 15:58:15 UTC*
  - f027dc0cf8 Merge pull request #31479 from isbm/isbm-zypper-unittest
  - 9f64333ccb Do not use Zypper purge (reason: too dangerous)
  - bc05acf7c3 Fix PyLint
  - c0eab8b549 Add space before "assert" keyword
  - 6bcb89a8f6 Implement list packages test
  - 78837d2926 Add mocking data
  - 0b64b8137f Implement test for version compare, where python fall-back algorithm is called
  - 18b30a3274 Implement test for version compare, where RPM algorithm is called
  - 59eca53441 Adjust test case for the third package in the test static data
  - 8034cf0b91 Add third test package static info
  - 90f209569a Implement test for the upgrade_available
  - ad87e719d6 Bugfix: when only one package, no dict is returned. Still upgrade_available should return boolean.
  - 7eb5f19cb4 Implement test for latest_available
  - e372c0b596 Implement test for the info_available
  - 447771c0fc Add Zypper static data for the available packages
  - 6989871d27 Implement test for info_installed
  - 0cc6bce4aa Use strings instead of unicode strings
  - 3342c03987 Implement list upgrades test
  - 8862d7af65 Add list upgrades Zypper static data

  - 4d38d318f4 Implement error handling test for listing upgrades
  - 080b4ee617 Do not strip the output
  - 53338402a5 Use renamed zypper products data file
  - c6135975b0 Rename Zypper products static test data file
  - ab3ff53d89 Reimplement list_upgrades to use XML output from Zypper instead
  - e87864986d Add Zypper unit test: test_list_products and test_refresh_db
  - cd6419fc9c Add Zypper Unit Test installed products sample data
- **ISSUE** #31370: (Ch3LL) pkg.info_installed on ubuntu12 does not output info and stack trace (refs: #31439)
- **ISSUE** #31366: (Ch3LL) pkg.info_installed on centos5 does not output info (refs: #31445)
- **PR** #31445: (rallytime) Only use LONGSIZE in rpm.info if available. Otherwise, use SIZE. @ *2016-02-24 18:35:31 UTC*
  - **PR** #31439: (rallytime) Fix lowpkg.info function for Ubuntu 12 - make sure we have a pkg name (refs: #31445)
  - 987dd89979 Merge pull request #31445 from rallytime/fix-31366
  - 42415a4a7b Make rpm_tags query more concise
  - 9965fe188a Added to pkg.info_installed test for RedHat and Suse systems
  - 47cc7c3466 Add error check when retcode is 0, but stderr is present
  - 294371243d Only use LONGSIZE in rpm.info if available. Otherwise, use SIZE.
- **PR** #31464: (Ch3LL) integartion test: ensure decorator only runs on one method and not class @ *2016-02-24 18:35:00 UTC*
  - 979c8b4faa Merge pull request #31464 from Ch3LL/fix_int_test
  - a387d175d8 integartion test- ensure decorator only runs on one method and not entire class
- **ISSUE** saltstack/salt-bootstrap#695: (mtippett) Install Failures With Raspbian Jessie (refs: #31510, #31477, #31458)
- **PR** #31458: (vutny) Correct installation documentation for Debian @ *2016-02-24 17:01:09 UTC*
  - aa0a9a03dd Merge pull request #31458 from vutny/correct-doc-install-debian
  - 42aa7eeafd Add section about installation from the Debian Main Repository
  - 07dece2f8f Remove duplicate post-installation section for Debian install guide
  - 10c05f6943 Add install section for Debian Stretch (Testing) from community repository
  - b2c78e08dc Add note about supported Debian architectures on SaltStack corp repo
- **PR** #31457: (rallytime) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2016-02-24 16:42:17 UTC*
  - 330c4d8b0f Merge pull request #31457 from rallytime/merge-2015.8
  - 94b3cf08c7 Merge branch '2015.5' into '2015.8'
    - 26733ce988 Merge pull request #31442 from sastorsl/salt-modules-file.py-copy-check-src
      - 0a4132866d removed lint in the exception string
      - f8b5d498c3 Add os.path.exists(src) to file.py, def copy
    - e480727d27 Merge pull request #31441 from cachedout/issue_30739

        - ffcfad1570 Include localhost minions in presence detection for runner
    - 91ff95f093 Merge pull request #31416 from carlwgeorge/selinux_doc_fix
        - 0e6846d72e selinux module documentation fix
    - 7d01979898 Merge pull request #31336 from terminalmage/config-validation-logging
        - 795008bad1 Improve config validation logging
    - fed096a29d Merge pull request #31374 from sjorge/solarish_hwaddr
        - bdf2576dfb missed a .format and messed up the join
        - bbd2fdc96d fix for illumos/solaris hwaddr
    - 6ee17f905b Merge pull request #31339 from jacobhammons/dot7prev
        - 07120a8d48 changed latest release to 2015.8.7

- **ISSUE** #31370: (Ch3LL) pkg.info_installed on ubuntu12 does not output info and stack trace (refs: #31439)
- **PR** #31439: (rallytime) Fix lowpkg.info function for Ubuntu 12 - make sure we have a pkg name (refs: #31445) @ *2016-02-24 16:24:46 UTC*
  - e553f18dc4 Merge pull request #31439 from rallytime/fix-31370
  - 1931c61563 Only run this pkg.info_installed test on distros that have that func
  - 0488668a00 Fix lowpkg.info function for Ubuntu 12 - make sure we have a pkg name
- **PR** #31456: (RabidCicada) Clarified the form of requisite targets/requisite-references @ *2016-02-24 16:24:00 UTC*
  - fcb12dbe96 Merge pull request #31456 from RabidCicada/clarify-requisites-doc
  - 87f4843490 Clarified the form of requisite targets/requisite-references
- **ISSUE** #30431: (nbow) cp.get_url with large files results in an Uncaught Exception (refs: #30704)
- **ISSUE** #27093: (TheBigBear) 2015.8.0 winrepo downloader corrupts some installers (refs: #30704)
- **PR** #31453: (DmitryKuzmenko) Backport cp_geturl fix for large files into 2015.8 @ *2016-02-24 15:38:24 UTC*
  - **PR** #30704: (DmitryKuzmenko) Issues/30431 get url large file (refs: #31453)
  - 7dac1db55d Merge pull request #31453 from DSRCompany/issues/30431_get_url_large_file_2015.8_backbort
  - 664bdec2b3 Backport cp_geturl fix for large files into 2015.8
- **PR** #31444: (jacobhammons) Documentation updates - ddns state, file.line state/exe function, installation dependencies @ *2016-02-23 22:40:05 UTC*
  - 8f6c4be618 Merge pull request #31444 from jacobhammons/ddns-docs
  - 0b8fce1de4 Fixes #31402 Added arguments to state *file.line* to fix issue where exe module uses *line* and state module uses *name*. Reformatted parameters in exe module *file.line* placeholder release notes for 2015.5.8
  - 0b1fdf7e21 Added note clarifying when dnspython is not required
- **PR** #31341: (twangboy) Clarification on Windows Package Manager docs @ *2016-02-23 16:09:18 UTC*
  - 42027e0d72 Merge pull request #31341 from twangboy/package_manager_docs
  - c16cfc6360 Fix typos
  - 8dff065cec Fix some formatting issues
  - dfef24f13b Merge branch '2015.8' of https://github.com/saltstack/salt into 2015.8

  - 807257b138 Clarification for Windows Package Manger
- **PR** #31380: (kiorky) Bring up ext_pillar rendering errors as well @ *2016-02-23 16:08:39 UTC*
  - 30d968c0a7 Merge pull request #31380 from kiorky/p
  - e3e97a43ce Bring up ext_pillar rendering errors as well
- **ISSUE** #31410: (terminalmage) Debian GNU/Linux grains broken in head of 2015.8 branch (refs: #31418)
- **PR** #31418: (terminalmage) Fix core grains when Debian OS detected as 'Debian GNU/Linux' @ *2016-02-23 15:49:49 UTC*
  - 64ed9fcd01 Merge pull request #31418 from terminalmage/fix-debian-grains
  - 5c833efc01 Support running grains tests
  - 0e0cd17160 Rename core.py to core_test.py
  - d3cd1b596d Add unit test for core grains
  - e3d549d376 Fix debian grains setup
- **PR** #31429: (mcalmer) fix argument handling for pkg.download @ *2016-02-23 15:48:23 UTC*
  - ec01b994bd Merge pull request #31429 from mcalmer/fix-refresh-arguments
  - 299c07fa7d fix argument handling for pkg.download
- **PR** #31432: (ticosax) [dockerng] Hotfix docker 1.10.2 @ *2016-02-23 15:39:04 UTC*
  - 05c12b9ba1 Merge pull request #31432 from ticosax/hotfix-docker-1.10.2
  - 1e9f6ff324 handle inconsistencies in dockerd API
  - 8484815f58 pep8
- **PR** #31420: (twangboy) Handle Unversioned Packages @ *2016-02-22 23:46:24 UTC*
  - fb81e905e4 Merge pull request #31420 from twangboy/unversioned_pkgs
  - 816e991e87 Fix version check
  - 85d8b938ad Match unversioned packages to winrepo
- **PR** #31417: (jacobhammons) ddns state docs updated with notes regarding the name, zone, and keyfile. @ *2016-02-22 23:16:48 UTC*
  - 19d7810478 Merge pull request #31417 from jacobhammons/ddns-docs
  - 5c4cbbb572 Added notes regarding the name, zone, and keyfile.
- **PR** #31391: (redmcg) Added sanity check: is 'pillar' in self.opts @ *2016-02-22 20:05:27 UTC*
  - ac6af79abc Merge pull request #31391 from redmcg/master_schedule_fix
  - 91e74feaf3 Added sanity check: is 'pillar' in self.opts
- **PR** #31376: (cro) Some distros don't have a /lib/systemd @ *2016-02-22 18:11:39 UTC*
  - c7bd13c9c9 Merge pull request #31376 from cro/suse_service2
  - f3fec5562e We need one more mocked return from listdir.
  - ab9d9e7008 Can't add a tuple and a string.
  - 8f12bdb1a0 Check to see if a path is a link, because it's likely that if it IS a link, one of the other paths points to it. Ignore so we don't get duplicates.
  - 8f0e866f1b Some distros do not seem to have a /lib/systemd, but do have a /usr/lib/systemd

- **ISSUE** #29727: (oeuftete) dockerng.running does not pull image as documented (refs: #31352, #31455)
- **ISSUE** #27976: (syphernl) Module dockerng.inspect_image always returns 404 (refs: #31352)
- **PR** #31352: (ticosax) [dockerng] Pull missing images when calling dockerng.running @ *2016-02-22 16:54:10 UTC*
  - 105821efc7 Merge pull request #31352 from ticosax/pull-image-on-running
  - 8c86eeb4dc Pull missing images when calling dockerng.running
- **PR** #31378: (mcalmer) Zypper refresh handling @ *2016-02-22 16:50:28 UTC*
  - 83294e4f3a Merge pull request #31378 from mcalmer/zypper-refresh-handling
  - 274e6467be do not change kwargs in refresh while checking a value
  - 644b14c273 simplify checking the refresh paramater
  - db0e0de2fd add refresh option to more functions
  - 5836be3f59 unify behavior of refresh
- **ISSUE** #31229: (eykd) git.latest broken behavior in 2015.8.x on older Git (refs: #31373)
- **PR** #31373: (terminalmage) Use –set-upstream instead of –track to set upstream on older git @ *2016-02-22 16:46:00 UTC*
  - e24685b89a Merge pull request #31373 from terminalmage/issue31229
  - 28f0a75cc1 Use –set-upstream instead of –track to set upstream on older git
- **ISSUE** #31137: (jeffreyctang) logrotate creates .bak files in /etc/logrotate.d which logrotate reads. (refs: #31390)
- **PR** #31390: (abednarik) Fix Logrotate module. @ *2016-02-22 16:09:15 UTC*
  - c5790bc4d6 Merge pull request #31390 from abednarik/remove_deprecated_psed_in_lorgotate
  - c1e0ff7785 Fix Logrotate module.
- **ISSUE** #28004: (warden) dockerng.image_present should allow public repository pulling by default (refs: #31354)
- **PR** #31354: (ticosax) [dockerng] Dont require auth for all registries @ *2016-02-20 05:45:10 UTC*
  - 174ee10fc2 Merge pull request #31354 from ticosax/dont-require-auth-for-all-registries
  - 4a9f661d66 It exists public registries where auth is not required.
- **PR** #31368: (whiteinge) Update list of netapi clients for autoclass @ *2016-02-19 20:57:28 UTC*
  - 8d0498eff4 Merge pull request #31368 from whiteinge/netapi-client-list
  - 0cfe5d89a0 Update list of netapi clients for autoclass
- **PR** #31367: (techhat) Add docs on how to actually use SDB @ *2016-02-19 20:07:17 UTC*
  - 9b0e29107b Merge pull request #31367 from techhat/sdbdocs
  - eea192a545 Add docs on how to actually use SDB
- **PR** #31357: (ticosax) [dockerng] Support docker inconsistencies @ *2016-02-19 20:02:08 UTC*
  - 7e599f0e27 Merge pull request #31357 from ticosax/support-docker-inconsistencies
  - 3672b8e7b1 docker daemon returns sometimes empty list and sometimes None
- **PR** #31353: (ticosax) [dockerng] Fix when ports are integers @ *2016-02-19 19:55:30 UTC*
  - **PR** #31326: (ticosax) [dockerng] Detect settings removal (refs: #31353)

  - 18bd78260d Merge pull request #31353 from ticosax/fix-when-port-are-integers
  - 20fdc43968 Follow up for #31326
- **PR** #31346: (ticosax) Backport #31130 to 2015.8 @ *2016-02-19 19:46:48 UTC*
  - **PR** #31130: (ticosax) Saltnado: provide also get parameters to the context (refs: #31346)
  - dec254a7a2 Merge pull request #31346 from ticosax/backport-31130-to-2015.8
  - a8dc33a5e3 Saltnado provide also get parameters to the context
- **PR** #31332: (terminalmage) Clarify documentation for gitfs/hgfs/svnfs mountpoint and root options @ *2016-02-19 18:31:29 UTC*
  - d639d65381 Merge pull request #31332 from terminalmage/issue31167
  - eebc325040 Clarify documentation for gitfs/hgfs/svnfs mountpoint and root options
- **PR** #31305: (mcalmer) call zypper with option –non-interactive everywhere @ *2016-02-19 18:14:57 UTC*
  - d067e77fee Merge pull request #31305 from mcalmer/zypper-non-interactive-everywhere
  - 75e776761c write a zypper command builder function
  - 3df302fcb7 call zypper with option –non-interactive everywhere
- **PR** #31337: (jacobhammons) Release notes and versioning for 2015.8.7 @ *2016-02-19 00:20:30 UTC*
  - 98a14f8090 Merge pull request #31337 from jacobhammons/dot7
  - d4fb33939e Release notes and versioning for 2015.8.7
- **PR** #31326: (ticosax) [dockerng] Detect settings removal (refs: #31353) @ *2016-02-18 22:02:50 UTC*
  - f0ba9c1eca Merge pull request #31326 from ticosax/2015.8-dockerng-detect-settings-removal
  - 7bedd86ebe Add detection of removed settings.
- **PR** #31292: (twangboy) Fix dunder virtual to check for Remote Administration Tools @ *2016-02-18 18:57:26 UTC*
  - 130f515391 Merge pull request #31292 from twangboy/win_servermanager
  - 89b47ab3c5 Update return documentation for install/remove
  - a0be43120b Fix cmd_quote error
  - 13cd57a890 Remove repeating Import ServerManager command
  - 3270a2859f Add check for server manager module
  - 4bdae47a44 Added checks for Windows 2008 R2
- **ISSUE** #30932: (johje349) Glusterfs peered fails on secondary host in 2015.8.4 (refs: #31287)
- **PR** #31287: (joejulian) Rework tests and fix reverse peering with gluster 3.7 @ *2016-02-18 17:57:23 UTC*
  - 5d31714b44 Merge pull request #31287 from joejulian/2015.8_30932_peer_probe_by_ip
  - 783e9b2e13 Rework tests and fix reverse peering with gluster 3.7
- **PR** #31196: (sakateka) Here are a few fixes utils.network @ *2016-02-18 17:27:00 UTC*
  - a2f6447f8d Merge pull request #31196 from sakateka/utils-network-fix
  - a7b11024dd fix typo
  - 92fd48fcf7 Do not Fallback to use lsof if proc available
- **ISSUE** #29795: (vutny) Unable to override state-output setting in command line (refs: #31299)

- **PR** #31299: (rallytime) Allow state-output and state-verbose default settings to be set from CLI @ *2016-02-18 17:25:23 UTC*
  - d20a30b8be Merge pull request #31299 from rallytime/fix-29795
  - 483f31922b Allow state-output and state-verbose default settings to be set from CLI
- **PR** #31317: (terminalmage) Fix versonadded directive @ *2016-02-18 16:56:32 UTC*
  - 25d8af21c9 Merge pull request #31317 from terminalmage/git-version-audit
  - acc3b54621 Fix versonadded directive
- **ISSUE** #30999: (orymate) git.latest rev=tag fails with old git(1) (refs: #31245, #31301)
- **PR** #31301: (terminalmage) Corrected fix for #30999 @ *2016-02-18 15:59:40 UTC*
  - **PR** #31245: (jespada) fix git state for git version older than 1.9.2 (refs: #31301)
  - f2b662371c Merge pull request #31301 from terminalmage/issue30999
  - 625af70e08 Fix –unset-upstream handling
  - 7940881797 fix git state github issue #30999
- **PR** #31302: (terminalmage) Audit CLI opts used in git states @ *2016-02-18 15:58:48 UTC*
  - 408d89e174 Merge pull request #31302 from terminalmage/git-version-audit
  - ca410c0a94 Audit CLI opts used in git states
- **PR** #31312: (terminalmage) Merge 2015.5 into 2015.8 @ *2016-02-18 15:57:08 UTC*
  - 098f05eb3c Merge pull request #31312 from terminalmage/merge-forward-2015.5-2015.8
  - 808d150fe4 Merge remote-tracking branch 'upstream/2015.5' into merge-forward-2015.5-2015.8
  - cd3400e67e Merge pull request #31288 from notpeter/ssh_known_hosts_docs
    - 3f573d89a2 Improve salt.states.ssh_known_hosts documentation.
  - 875d9925fa Merge pull request #31183 from heyfife/fix-gce-named-static-ip-reservation
    - 26774e2323 Fixed named external_ip reservation/re-use code.
  - e56c402c0c Merge pull request #31032 from terminalmage/issue31001
    - 42daea4509 yumpkg.py: Remove repoquery usage everywhere but check_db
    - 50befbc149 backport salt.utils.pkg.rpm to 2015.5
    - a1ad14994a Move salt.utils.itersplit() to salt.utils.itertools.split()
    - 5b8646ce64 Ignore failure to install new enough dnf-plugins-core
    - defe0859fd Ensure that dnf-plugins-core 0.1.15 is installed
  - cec69b74f0 Merge pull request #31264 from sjorge/if_missing-155-fix
    - 545edbf5e1 fix if_missing gets appended to dirs list, take III
- **ISSUE** #31223: (pprince) file_tree pillar: fails when data files at root end in '\\n' (refs: #31225)
- **PR** #31225: (pprince) Fix in file_tree pillar (Fixes #31223.) @ *2016-02-18 06:06:12 UTC*
  - c58f654bc3 Merge pull request #31225 from pprince/PR/bugfix/file_tree
  - d592d8636b Fix regression in file_tree pillar (Fixes #31223.)
- **PR** #31233: (mcalmer) implement version_cmp for zypper @ *2016-02-17 20:20:19 UTC*

  - fe9e5d27e6 Merge pull request #31233 from mcalmer/2015.8-zypperpy-add-version_cmp
  - 389a4b2548 Check if rpm-python can be imported
  - 6ad6a90955 pylint changes
  - 7beaf26068 implement version_cmp for zypper
- **PR** #31273: (rallytime) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2016-02-17 20:14:05 UTC*
  - 93c03a400b Merge pull request #31273 from rallytime/merge-2015.5
  - 11cfb636fb Pylint fix
  - 023ad4635c Merge branch '2015.5' into '2015.8'
    - fa3f474de9 Merge pull request #31110 from cachedout/fixup_30730
      - 5bf5848e04 Fixup unit test
      - f558f68e0a Fixes pylint warnings
      - 56a975ec43 Attempt to fix pylint warnings
      - 55d71be057 Make documentation and code examples consistent with code
      - 1f04fed6f8 Change parameter name from includes to skips
      - ccf5e13e7d Adding support for skipHidden in SetInclude
      - 4f2d4af2e7 Variable names standardization
      - f5917ac1e8 Fixes typo
      - 26e5236073 Invert RebootRequired logic
      - 8065a7abf6 Add basic documentation and define how the skips parameter works.
      - 389fea7508 Change parameter name from includes to skips
      - 30e1fef906 Adding support for skipHidden in SetInclude
      - 1244eea5be Variable names standardization, consistent if/else logic with states.win_update
- **PR** #31253: (gtmanfred) allow for nova servers to be built with premade volumes @ *2016-02-17 17:55:39 UTC*
  - dc2e7c8956 Merge pull request #31253 from gtmanfred/2015.8
  - 36bf06e539 fix doc for boot_volume
  - 9660c91b57 allow for nova servers to be built with premade volumes
- **ISSUE** #30651: (sjorge) salt.states.grains.list_present should not show changes if none are made! (refs: #31651, #30689)
- **PR** #31271: (rallytime) Back-port #30689 to 2015.8 @ *2016-02-17 16:52:36 UTC*
  - **PR** #30689: (sjorge) fix for #30651 grains.list_present and grains.list_absent (refs: #31651, #31271)
  - 29e3dd091d Merge pull request #31271 from rallytime/bp-30689
  - 3dae79d516 fix nested grains always show update due to __grains__.get() not supporting the ":" seperator
- **ISSUE** #30461: (jfindlay) update documentation on bootstrap-supported platforms (refs: #31255)
- **PR** #31255: (jacobhammons) Fixes #30461 @ *2016-02-17 02:23:46 UTC*
  - fcfc6f4fd3 Merge pull request #31255 from jacobhammons/doc-fixes
  - 3c4f8215c3 Fixes #30461 Credited Sebastian Kramer for finding CVE 2016-1866 in release notes Added note about salt virt not working on KVM in a VM

- **ISSUE** #31106: (rvandegrift) Exception from scheduled runner (refs: #31189)
- **PR** #31189: (dmacvicar) Fix crash with scheduler and runners (#31106) @ *2016-02-16 18:49:36 UTC*
  - 62d76902ce Merge pull request #31189 from dmacvicar/dmacvicar-2015.8-31106
  - 9ad8cb1e6b Fix crash with scheduler and runners (#31106)
- **ISSUE** #30962: (fantasy86) Targeting by matching ip address doesn't work (refs: #31201)
- **ISSUE** #30169: (colinlabs) Can't use Subnet/IP Address Matching (refs: #31201)
- **ISSUE** #29733: (roshan3133) salt -S <ipaddress> test.ping command output getting list of minions which did not not return. (refs: #31201)
- **ISSUE** #29188: (bergemalm) Unable to target minions via ipcidr in 2015.8 (refs: #31201)
- **PR** #31201: (The-Loeki) Utilize prepared grains var in master-side ipcidr matching @ *2016-02-16 18:36:10 UTC*
  - dc78d0a504 Merge pull request #31201 from The-Loeki/patch-1
  - 318689d728 Correct ordering of address/network matching, improve performance of master-side cidr matching
  - 4e4e0926da Utilize prepared grains var in master-side ipcidr matching
- **PR** #31239: (terminalmage) Improve logging when master cannot decode a payload @ *2016-02-16 16:35:46 UTC*
  - 60bbac36fa Merge pull request #31239 from terminalmage/better-bad-load-logging
  - 1fbe3cba1f Improve logging when master cannot decode a payload
- **ISSUE** #31185: (twangboy) pkg.refresh_db leaves old sls files if the name changes (refs: #31190)
- **PR** #31190: (twangboy) Clear minion cache before caching from master @ *2016-02-16 16:11:26 UTC*
  - 80f1c3553b Merge pull request #31190 from twangboy/refresh_db
  - 860437665d Fix some lint
  - 799d938d6a Clear minion cache before caching from master
- **PR** #31226: (pprince) Minor docs fix: file_tree pillar (Fixes #31124) @ *2016-02-16 15:25:33 UTC*
  - **PR** #31124: (zygiss) Make load beacon cross-platform (refs: #31226)
  - 28a2b8097b Merge pull request #31226 from pprince/PR/docfix/file_tree
  - c13852fbbf Minor docs fix: file_tree pillar (Fixes #31124)
- **PR** #31234: (mcalmer) improve doc for list_pkgs @ *2016-02-16 15:25:06 UTC*
  - 9afad13306 Merge pull request #31234 from mcalmer/zypperpy-comment-list_pkgs
  - e3bb862a32 improve doc for list_pkgs
- **PR** #31237: (mcalmer) add handling for OEM products @ *2016-02-16 15:12:21 UTC*
  - e8f3a707ae Merge pull request #31237 from mcalmer/zypper_py-add-OEM-product-handling
  - d773b7317b add handling for OEM products
- **PR** #31182: (rallytime) Back-port #31172 to 2015.8 @ *2016-02-13 21:36:07 UTC*
  - **PR** #31172: (techhat) Use correct deploy directory (refs: #31182)
  - 415654ee9e Merge pull request #31182 from rallytime/bp-31172
  - a743778e98 Use correct deploy directory

- **ISSUE** #27498: (arthurlogilab) [runner] salt-run cache.clear_mine_func broken, can't take clear_mine_func (refs: #31191)
- **PR** #31191: (rallytime) Make sure doc example matches kwarg @ *2016-02-13 21:34:57 UTC*
  - 434e05667a Merge pull request #31191 from rallytime/fix-27498
  - 0bdbaa49d1 Make sure doc example matches kwarg
- **PR** #31171: (Ch3LL) added logic to check for installed package @ *2016-02-12 22:10:21 UTC*
  - c5e5af827c Merge pull request #31171 from Ch3LL/megan-20158
  - a12e2f566b fix lint error
  - a123efd4ef added logic to check for installed package
- **ISSUE** #30934: (marnovdm) contents_pillar no longer works with lists in 2015.8.5 (refs: #31026, #31177)
- **PR** #31177: (Ch3LL) add integration test for issue #30934 @ *2016-02-12 22:09:31 UTC*
  - a024d3536f Merge pull request #31177 from Ch3LL/test_content_pillars
  - 9204e3f562 add integration test for issue 30934w
- **PR** #31181: (cachedout) Lint 2015.8 branch @ *2016-02-12 21:57:02 UTC*
  - 1f22335e28 Merge pull request #31181 from cachedout/lint_20158
  - 4c0be11627 Lint 2015.8 branch
- **ISSUE** #29423: (l13t) iptables and match-set with two parameters (refs: #29718)
- **PR** #31169: (rallytime) Back-port #29718 to 2015.8 @ *2016-02-12 18:28:13 UTC*
  - **PR** #29718: (thusoy) Support match-sets in iptables module (refs: #31169)
  - 4d1b49c1e7 Merge pull request #31169 from rallytime/bp-29718
  - ceae2a16f8 Support match-sets in iptables module
- **PR** #31170: (rallytime) Back-port #31157 to 2015.8 @ *2016-02-12 18:27:49 UTC*
  - **PR** #31157: (captaininspiration) Fix locale generation on Ubuntu (refs: #31170)
  - f2efd3e6c1 Merge pull request #31170 from rallytime/bp-31157
  - 27776b5f4e Fix locale generation on Ubuntu
- **PR** #31147: (cro) Documentation clarifications. @ *2016-02-12 17:16:27 UTC*
  - 7f49fbb70d Merge pull request #31147 from cro/fx2_doc
  - a005e4af55 Documentation clarifications.
- **PR** #31153: (edencrane) Fixed invalid host causing 'reference to variable before assignment' @ *2016-02-12 16:30:19 UTC*
  - 7986b9e033 Merge pull request #31153 from edencrane/fix-network-connect-invalid-hostname
  - a14c4bb5f2 Fixed invalid host causing 'reference to variable before assignment'
- **ISSUE** #30994: (onorua) beacon enable from state is failing (refs: #31152)
- **PR** #31152: (garethgreenaway) fixes to beacon module, state module and friends @ *2016-02-12 16:27:40 UTC*
  - f5ab76801b Merge pull request #31152 from garethgreenaway/30994_beacon_add_failing_and_other_fixes

  - 91b14dca40 fixing the beacon module and state module to handle passing enabled properly. Also reworking how what is returned from the validating functions is handled to ensure when beacon configurations aren't validate the results indicate exactly why.
- **PR** #31149: (jfindlay) add 2015.8.7 release notes @ *2016-02-12 00:06:15 UTC*
  - c8047d979d Merge pull request #31149 from jfindlay/2015.8
  - b58783b895 add 2015.8.7 release notes
- **PR** #31134: (isbm) Fix types in the output data and return just a list of products @ *2016-02-11 20:19:22 UTC*
  - 5c394ac49c Merge pull request #31134 from isbm/isbm-zypper-list-products
  - 670a326e3d Fix types in the output data and return just a list of products
- **ISSUE** #31115: (nfillot) 2015.8.5 salt-cloud nova valid ip address was not found (refs: #31120)
- **ISSUE** #29758: (zaide) 2015.8.3 salt-call runners.cloud : local variable 'access_ip' referenced before assignment (refs: #31120)
- **ISSUE** #29666: (tminn) Nova driver broken for 2015.8.[1-3] (refs: #31120)
- **PR** #31120: (gtmanfred) Clean up some bugs in the nova driver @ *2016-02-11 20:17:41 UTC*
  - 8f2e3a26e5 Merge pull request #31120 from gtmanfred/2015.8
  - 4a411c0817 fix comment
  - 47ecb7a150 include all ips in public_ips or private_ips
  - b2e8202f5d dont exit on a missing server
  - 8ad1ee6db4 clean up references to access_ip extra network
- **ISSUE** #31099: (Ch3LL) Cannot specify size in map file in 2015.8 (refs: #31132)
- **PR** #31132: (rallytime) Make sure required profile configurations passed in a map file work @ *2016-02-11 20:16:46 UTC*
  - 2d592a398e Merge pull request #31132 from rallytime/fix-31099
  - 1da03da9df Pylint fix
  - 337592ec56 Make sure required profile configurations passed in a map file work
- **ISSUE** #31014: (gtmanfred) [2015.8] pkg breaks for yum pkgs.latest if the packages has an epoch (refs: #31131, #31015, #31031)
- **PR** #31131: (Ch3LL) integration test for issue #31014 @ *2016-02-11 17:33:23 UTC*
  - b831e0a865 Merge pull request #31131 from Ch3LL/megan-20158
  - af82b1233a integration test for issue #31014
- **PR** #31133: (cachedout) Fixup 31121 @ *2016-02-11 17:32:24 UTC*
  - e378afd891 Merge pull request #31133 from cachedout/fixup_31121
  - a4040da46d Fix bad unit test
  - 0e68fafb74 Fix alternative module and state.
- **PR** #31125: (isbm) Force-kill websocket's child processes faster than default two minutes. @ *2016-02-11 16:50:57 UTC*
  - a4a40262f8 Merge pull request #31125 from isbm/isbm-salt-api-service
  - f73f70375c Force-kill websocket's child processes faster than default two minutes.

- **PR** #31119: (sakateka) fixes for ipv6-only multi-master faliover @ *2016-02-11 16:21:45 UTC*
    - 79c85859bc Merge pull request #31119 from sakateka/fix-for-ipv6only-failover
    - 2c45d151d1 fix unintentional breaking changes
    - 043a5e6fd7 fixes for ipv6-only multi-master faliover
- **PR** #31107: (techhat) Don't try to add a non-existent IP address @ *2016-02-10 21:52:42 UTC*
    - 825b510030 Merge pull request #31107 from techhat/nebulaprivip
    - 1fa69982c4 Don't try to add a non-existent IP address
- **PR** #31108: (justinta) Changed npm integration test to install request. @ *2016-02-10 21:52:02 UTC*
    - c56a819fd8 Merge pull request #31108 from jtand/npm_test_fix
    - a5eac47b25 Changed npm integration test to install request.
- **PR** #31105: (cachedout) Lint 30975 @ *2016-02-10 21:11:21 UTC*
    - de1abae9d1 Merge pull request #31105 from cachedout/lint_30975
    - 446b4c2aff Lint #30975
    - b4fe9aaa11 fixes issue in which s3.role_arn was defaulting to ''
- **ISSUE** #31069: (symphorien) Wrong filename in documentation for x509 state (refs: #31100)
- **PR** #31100: (jfindlay) states.x509: docs: peer.sls -> peer.conf @ *2016-02-10 20:47:45 UTC*
    - 2e5499748a Merge pull request #31100 from jfindlay/x509_sls
    - 6c303b99c2 states.x509: docs: peer.sls -> peer.conf
- **PR** #31103: (twangboy) Point to reg.delete_key_recursive @ *2016-02-10 20:46:53 UTC*
    - f2bede1c00 Merge pull request #31103 from twangboy/fix_reg_state
    - fe1ca906d2 Point to reg.delete_key_recursive
- **PR** #31093: (techhat) Ensure double directories don't get created @ *2016-02-10 18:53:47 UTC*
    - 94fa76831f Merge pull request #31093 from techhat/spmfix
    - 4f4c8877ad Ensure double directories don't get created
- **ISSUE** #31056: (JensRantil) file.symlink documentation improvement (refs: #31095)
- **PR** #31095: (jfindlay) modules.file, states.file: explain symbolic links @ *2016-02-10 18:53:24 UTC*
    - c015ca865c Merge pull request #31095 from jfindlay/link_doc
    - 7d9df6b26c modules.file, states.file: explain symbolic links
- **ISSUE** #31059: (mf-collinhayden) salt-cloud rename fails in 2015.8.5 (refs: #31061)
- **ISSUE** #30950: (tmaulik) Salt-cloud create_snapshot is not recognizing snapshot_name parameter in salt 2015.8.5 (refs: #31061)
- **PR** #31061: (rallytime) Revert #30217 - was causing salt-cloud -a breakage @ *2016-02-10 18:13:59 UTC*
    - **PR** #30217: (pass-by-value) Make sure cloud actions can be called via salt run (refs: #31061, #30691)
    - 4d6706b3e7 Merge pull request #31061 from rallytime/revert-breakage
    - ced2d9f922 Revert #30217
- **ISSUE** #31088: (gladiatr72) request for color logging fix backport (refs: #31090)

- **PR** #31090: (rallytime) Back-port #30542 to 2015.8 @ *2016-02-10 18:06:38 UTC*
  - **PR** #30542: (gladiatr72) address color log dict lookup exceptions w/ non-posix log level names (refs: #31090)
  - 482eea9883 Merge pull request #31090 from rallytime/bp-30542
  - 67a713f2f6 Some 3rd-party modules (e.g. gnupg) define custom log levels that emit at INFO level and above. This patch sets the color data lookups to default to TextFormat('reset') rather than producing a stack trace every time a log message is generated from an affected module.
- **PR** #31085: (jacksontj) Correctly remove path we added after loader is completed @ *2016-02-10 17:47:22 UTC*
  - 5dcaa8d387 Merge pull request #31085 from jacksontj/2015.8
  - dd5051c9e6 Correctly pop the path we added after loader is completed.
- **ISSUE** #28142: (zmalone) Deprecate or update the copr repo (refs: #31037)
- **PR** #31037: (vutny) Update RHEL installation guide to reflect latest repo changes @ *2016-02-10 17:36:04 UTC*
  - 27bf83fa59 Merge pull request #31037 from vutny/correct-doc-install-on-rhel
  - 6370ddda9f Update RHEL installation guide
  - afdaefbf3d Add *systemctl* examples for RHEL 7 to the installation guide
  - 069a661eb1 Correct ZeroMQ4 repo install guide for RHEL
  - d2a9d67b5b Update installation instruction for community repos on RHEL
  - bbdf2523c8 Add workaround for RHEL 7 systems mentioned in the issue #29094
- **PR** #31050: (basepi) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2016-02-09 20:13:34 UTC*
  - 8704750cf9 Merge pull request #31050 from basepi/merge-forward-2015.8
  - d86e014a39 Merge remote-tracking branch 'upstream/2015.5' into merge-forward-2015.8
    - 1c699a1664 Merge pull request #30974 from rallytime/bp-30949
      - ff6542f593 Replace cfdisk with sfdisk
    - c7f87cc371 Merge pull request #30942 from rallytime/bp-30897
      - 885e00ba54 Only remove the word linux from distroname when its not part of the name
    - 35b7f62669 Merge pull request #30922 from jacobhammons/prev-rel-notes
      - 57c1ec637a Rev latest version to 2015.8.5
    - 2488bb902e Merge pull request #30865 from abednarik/better_boto_elb_error
      - 3561e8c19b Better boto elb error message.
    - 4da04f82c8 Merge pull request #30831 from jacobhammons/readme-update
      - 01a92f5d98 Updated readme
    - 90c1ea9f6c Merge pull request #30829 from jacobhammons/release-2015.5
      - c95bb60148 Version to 2015.8.4
    - 80a36793cb Merge pull request #30784 from rallytime/bp-24952
      - a07908bdea Don't split the string on a single line
    - e978f5392f Merge pull request #30764 from terminalmage/issue30560

      - 39736afcd7 Work around yum versionlock's inability to remove holds by package name alone
    - 6f565c0d76 Merge pull request #30760 from toanju/2015.5
      - dc4256f7df Changed output format of arp_ip_target from list to comma delimited string
    - 1c205b4898 Merge pull request #30757 from yannis666/fix-for-mine-update-merge
      - 61bb23e256 Fix to mine update to merge configuration
    - f9fde8f6a7 Merge pull request #30749 from abednarik/fix_network_system_test
      - 1e9e97df59 Fix Netwotk hostname Module in Debian systems.
- **PR** #31053: (cachedout) Fix boto test failures @ *2016-02-09 20:02:16 UTC*
  - f13ffd4608 Merge pull request #31053 from cachedout/boto_test_fix
  - c73b5a4a66 Fix boto_secgroup
  - 25bcc68357 Fix boto test failures
- **ISSUE** #30938: (lorengordon) Windows: Upgrade overwrites minion config file (refs: #31029, #31028)
- **PR** #31029: (twangboy) Windows defaults to multiprocessing true @ *2016-02-09 18:20:36 UTC*
  - 87f2816ef5 Merge pull request #31029 from twangboy/win_defaults
  - baffbbdb74 Comment multiprocessing line in minion config
  - 933544b8c8 Set multiprocessing to true in config.py
- **ISSUE** #27796: (onsmribah) IOError: [Errno 13] Permission denied: '/var/cache/salt/master/.dfn' when using python salt.wheel module (refs: #30998)
- **PR** #30998: (dmacvicar) add_key/reject_key: do not crash w/Permission denied: '/var/cache/salt/master/.dfn' (#27796) @ *2016-02-09 17:57:36 UTC*
  - 0dcdd0a2a7 Merge pull request #30998 from dmacvicar/dmacvicar-2015.8-27796
  - 9602fe2aeb Do not crash on add_key/reject_key if the previous one set the drop file. (#27796)
- **ISSUE** #31041: (fredrikaverpil) Reading about win_service in 2015.8.5 docs, but it's not available in 2015.8.5 (refs: #31550, #31049)
- **PR** #31049: (twangboy) Fix versionadded in win_service.config (refs: #31550) @ *2016-02-09 17:55:07 UTC*
  - e773fc822a Merge pull request #31049 from twangboy/win_svc_docs
  - 98005255d1 Fix versionadded in win_service.config
- **PR** #30987: (youngnick) Changed glusterfs.peer() module so state can handle localhost peering attempts. @ *2016-02-09 17:51:58 UTC*
  - c3f115724a Merge pull request #30987 from youngnick/add-back-localhost-peer-handling
  - 730b5ef3e2 Update tests to cover new peering return val.
  - b2407305e8 Changed glusterfs.peer() module call return val so state can handle localhost peering attempts.
- **PR** #31042: (moltob) Allow using Windows path in archive.extracted name attribute @ *2016-02-09 17:47:20 UTC*
  - 8518655bfb Merge pull request #31042 from moltob/fix-archive-winpath
  - 9dcc617a53 Allow using Windows path in archive.extracted name attribute, including drive letter colon and backslashes.

- **PR** #31012: (terminalmage) Fix gitfs/git_pillar/winrepo provider to allow lowercase values @ *2016-02-09 17:24:25 UTC*
  - 1950359580 Merge pull request #31012 from terminalmage/fix-gitfs-provider-lc
  - 763581798b Add unit tests to ensure a valid provider
  - 49ec61d58b Fix gitfs/git_pillar/winrepo provider to allow lowercase values
- **ISSUE** #30983: (JensRantil) salt.modules.aptpkg.upgrade does not necessarily do *apt-get dist-upgrade* (refs: #31024)
- **PR** #31024: (jfindlay) modules.aptpkg.upgrade: clarify dist-upgrade usage @ *2016-02-09 17:20:57 UTC*
  - 3d8681b63e Merge pull request #31024 from jfindlay/dist_upgrade
  - 3d1be080ad modules.aptpkg.upgrade: clarify dist-upgrade usage
- **ISSUE** #30938: (lorengordon) Windows: Upgrade overwrites minion config file (refs: #31029, #31028)
- **PR** #31028: (twangboy) Fix config overwrite by windows installer @ *2016-02-09 17:20:24 UTC*
  - a0454ffb00 Merge pull request #31028 from twangboy/fix_installer
  - 8876893b5c Fix remove placeholder files
  - 788855cc94 Remove placeholder files
  - c834a9d5e5 Set overwrite to off
- **ISSUE** #31014: (gtmanfred) [2015.8] pkg breaks for yum pkgs.latest if the packages has an epoch (refs: #31131, #31015, #31031)
- **PR** #31031: (terminalmage) More complete fix for #31014 @ *2016-02-09 17:04:42 UTC*
  - **PR** #31015: (gtmanfred) include possible epoch in version for rpm (refs: #31031)
  - 071b9d4904 Merge pull request #31031 from terminalmage/issue31014
  - 6d15a17d6b Fix yumpkg _get_branch_option()
  - 4b855a85ee Don't handle epoch specially for dnf
  - 5244de2fae More efficient way to add the epoch before version number
  - e1211ed89f include possible epoch in version for rpm
- **ISSUE** #30934: (marnovdm) contents_pillar no longer works with lists in 2015.8.5 (refs: #31026, #31177)
- **PR** #31026: (terminalmage) Fix regression when contents_pillar/contents_grains is a list. @ *2016-02-09 00:03:15 UTC*
  - 2b8f7a12e7 Merge pull request #31026 from terminalmage/issue30934
  - f43aaf4dff Fix regression when contents_pillar/contents_grains is a list.
- **ISSUE** #30472: (sjorge) KeyError with schedule (refs: #30978)
- **PR** #30978: (garethgreenaway) fixes to state.py in 2015.8 @ *2016-02-08 18:49:05 UTC*
  - de215bd0cd Merge pull request #30978 from garethgreenaway/30472_state_functions_no_default_retcode
  - e33b5140f6 removing extra spaces.
  - f668ccf1f7 removing duplicate code, just set the default in the _set_retcode function
  - 5f2f0f60c0 The functions in the state module that return a retcode when something goes wrong, eg. a 1 or a 2, do not return a 0 when things go the way they're supposed to go. With the recent changes to the scheduler to ensure that the retcode is returned this is problematic and results in exceptions when

a state function is run from the schedule. This simple fix ensures a default retcode of 0 exists, it is then override in the _set_retcode function if there is an issue with the run

- **PR** #30893: (bdrung) Make build reproducible @ *2016-02-08 18:44:35 UTC*
  - 65fbf980cf Merge pull request #30893 from bdrung/reproducible
  - 089c869ec3 Make build reproducible
- **PR** #30945: (cachedout) Note that pillar cli args are sent via pub @ *2016-02-08 18:43:59 UTC*
  - 5b0c7649c7 Merge pull request #30945 from cachedout/note_pillar_cli
  - 3ff7d49555 Note that pillar cli args are sent via pub
- **ISSUE** #31000: (rmtmckenzie) Salt-cloud profile state fails to create LXC minion (refs: #31002)
- **PR** #31002: (rmtmckenzie) Fix lxc cloud provided minion reporting present @ *2016-02-08 18:14:50 UTC*
  - 3b7b6f2398 Merge pull request #31002 from rmtmckenzie/cloud-lxc-provide-fix
  - 9b17fdce5e Fix lxc cloud provided minion reporting present
- **PR** #31007: (justinta) Fixed rabbitmq_vhost test failure. @ *2016-02-08 17:48:00 UTC*
  - c48122ae9a Merge pull request #31007 from jtand/rabbitmq_vhost_test_fix
  - 962e0deda5 Fixed rabbitmq_vhost test failure.
- **ISSUE** #30993: (fredrikaverpil) Overstate: "This documentation has been moved here" (dead end) (refs: #31004)
- **PR** #31004: (rallytime) Remove overstate docs and a few references. @ *2016-02-08 17:08:24 UTC*
  - 811461e4b4 Merge pull request #31004 from rallytime/fix-30993
  - 33eb6ba125 Remove overstate docs and a few references.
- **PR** #30965: (anlutro) Fix rabbitmq_vhost.present result when test=True @ *2016-02-08 04:34:45 UTC*
  - 64125de6c7 Merge pull request #30965 from alprs/fix-rabbitmq_vhost_present_test
  - 2313747958 return changes when test=True
  - 95c8e74b72 make the code a bit simpler
  - aba29a73c4 fix rabbitmq_vhost.present result when test=True
- **PR** #30955: (Ch3LL) docs: add clarification when source is not defined @ *2016-02-06 18:29:33 UTC*
  - ef02779391 Merge pull request #30955 from Ch3LL/clarify_file_doc
  - 97b57ed2b1 docs: add clarification when source is not defined
- **PR** #30941: (rallytime) Back-port #30879 to 2015.8 @ *2016-02-05 21:15:20 UTC*
  - **PR** #30879: (rhansen) Don't delete a Docker volume if the volume's driver differs (refs: #30941)
  - d9785451c0 Merge pull request #30941 from rallytime/bp-30879
  - eb6f289fc1 change default for volume_present()'s force parameter to False
  - 34f3057e04 add 'force' to replace (or not) volumes with driver mismatch
  - d6d3b15738 typo fixes
- **PR** #30940: (twangboy) Fix Build Process for OSX @ *2016-02-05 18:44:34 UTC*
  - 21a83065aa Merge pull request #30940 from twangboy/mac_build_3
  - 3654a0e0c2 Change 2015 to 2016 in license file

  - aa7d0602a8 Update instructions in readme.md for shasum
  - 6f1a8f4146 Added code to add /opt/salt/bin to the path
  - 1e7468a08c Disable master, syndic, and api in postinstall
  - d49b3dcf1b Re-added start on load and keep alive
  - 3ff50a2254 Removed keepalive option
  - eb5d04bdf1 Remove autostart for api, master, and syndic
  - 3c0cce34c9 Added minimum requirements for installation
  - 1dcc23c85b Fix error on kickstart command
  - 7a163c46d8 Change to new way of starting and stopping services
  - 23d47722b7 Fix preinstall and postinstall scripts
  - 7ef723d815 Upgrade to latest pip
  - 0f09ad517f Updated pip dependencies
  - d3d4c1d13f Removed GPL Licensed software from build

- **PR** #30944: (jacobhammons) 2015.8.5 release notes linking and clean up @ *2016-02-05 17:40:10 UTC*
  - 183b500055 Merge pull request #30944 from jacobhammons/rel-notes
  - fbb7605366 2015.8.5 release notes linking and clean up
- **ISSUE** #30882: (hoonetorg) state lvm.vg_present broken with pv on devicemapper-dev (centos 7.2) (refs: #30905)
- **ISSUE** #26867: (joejulian) lvm pv's can show as not belonging to their vg if symlink is used (refs: #30905)
- **PR** #30905: (joejulian) Add realpath to lvm.pvdisplay and use it in vg_present @ *2016-02-05 17:05:32 UTC*
  - 91806b03b9 Merge pull request #30905 from joejulian/2015.8_fix_lvm_pv_mapper
  - f96650f3c3 Add realpath to lvm.pvdisplay and use it in vg_present
- **ISSUE** #30923: (youngnick) Starting a glusterfs volume after creation fails with an exception in 2015.8 and after. (refs: #30924)
- **PR** #30924: (youngnick) Fix small bug with starting volumes after creation. @ *2016-02-05 16:58:22 UTC*
  - af2832b69d Merge pull request #30924 from youngnick/glusterfs-start-volume-bug
  - be5295cf7b Fix small bug with starting volumes after creation.
- **PR** #30910: (cro) fix iDRAC state @ *2016-02-05 16:49:06 UTC*
  - 3a6666ad25 Merge pull request #30910 from cro/fx2_idrac
  - 68af2ab185 Lint.
  - c274c7ef6c Lint.
  - 3e38b762bf Add generic command for executing racadm commands on individual blades in a chassis.
  - 05979010f5 Finish the idrac state, fix problem with grains not loading sometimes.
- **PR** #30919: (garethgreenaway) Fixes to ssh_auth state module @ *2016-02-05 16:15:28 UTC*
  - 101fa12479 Merge pull request #30919 from garethgreenaway/ssh_auth_cp_get_url_needs_saltenv
  - c9ba038553 The call to cp.get_url needs the saltenv, if you're using environments other than base, it will fail.

- **ISSUE** #30300: (AkhterAli) boto_route53 __salt__ not defined. (refs: #30867, #30920)
- **PR** #30920: (jacobhammons) Versioned to 2015.8.5, added known issue #30300 to release notes @ *2016-02-05 01:12:17 UTC*
  - 6d4fd11dd0 Merge pull request #30920 from jacobhammons/release-notes
  - 93d47f8615 Versioned to 2015.8.5, added known issue #30300 to release notes
- **PR** #30894: (terminalmage) git module/state: Handle identity files more gracefully @ *2016-02-04 23:55:01 UTC*
  - 3d3321ab92 Merge pull request #30894 from terminalmage/issue30858
  - 08741eb969 Update versionadded/versionchanged
  - 8909d430e1 salt.states.git.latest(): Prevent tracebacks when git ssh auth fails
  - c961cf1c7d git: only use passphrase-protected key if invoked using salt-call
  - 0b286f1bc3 Add global ssh_config path to git ssh wrapper
  - f813cce4ad Add salt.modules.ssh.key_is_encrypted()
  - 1ae7c53e17 Add salt.utils.files.process_read_exception()
- **ISSUE** #30694: (pankajghadge) Tomcat war deployment version issue in new SALT version (refs: #30750)
- **PR** #30750: (jfindlay) extract whole war version @ *2016-02-04 21:41:01 UTC*
  - 2415b3e62e Merge pull request #30750 from jfindlay/war_version
  - 4b01c28ff9 modules,states.tomcat: allow specifying war version
  - 6deecdca0f states.tomcat: _extract_war_version parses path
  - 8dd3b6dfe9 modules.war._extract_war_version: allow non-semver
- **ISSUE** #30817: (bogdanr) If the private_key filespecified in the provider is missing then the driver will be disabled (refs: #30884)
- **PR** #30884: (rallytime) Move checks for private_key file existence and permissions to create function @ *2016-02-04 21:03:23 UTC*
  - 6a6456eaa6 Merge pull request #30884 from rallytime/fix-30817
  - 086ddae476 We need to check for a key_filename before looking for the path
  - e79321b418 Move checks for private_key file existence and permissions to create function
- **PR** #30888: (ticosax) Backport #30797 to 2015.8 @ *2016-02-04 21:02:25 UTC*
  - **PR** #30797: (rhansen) don't delete existing Docker volume if driver unspecified (refs: #30888)
  - 4ae2d829f0 Merge pull request #30888 from ticosax/backport-30797
  - 413c47a45f don't delete existing Docker volume if driver unspecified
  - 68b51be869 add additional states.dockerng.volume_present() unit tests
  - 849b94ed73 document the behavior if the driver is unspecified
- **PR** #30895: (bdrung) Fix various typos @ *2016-02-04 20:55:10 UTC*
  - 4372851ad9 Merge pull request #30895 from bdrung/2015.8
  - 708f2ff8ea Fix typo reponse -> response
  - 72c4eab6d7 Fix typo propogate -> propagate
  - 4912e365cb Fix typo directores -> directories

  - 74c8aba03e Fix typo exeption -> exception
  - 4692d84b07 Fix typos of improvement
  - 213fc2d858 Fix typo occuring -> occurring
  - fe6124003b Fix typo nonexistant -> nonexistent
  - 56ce7479b1 Fix typo catched -> caught
  - 821e690e65 Fix typo develoment -> development
  - b51279e086 Fix typo overide -> override
  - 4f2f04ea7d Fix typo relevent -> relevant
  - fe8be562c5 Fix typo existance -> existence
  - 4a2f4de1a8 Fix typo accross -> across
  - 9ae50c993e Fix typo Lenth -> Length
  - 20e79981e1 Fix typo preferrably -> preferably
  - f8d9f608dd Fix typo addres -> address
  - a7f12a13f0 Fix typo keywork -> keyword
  - bf92c3663b Fix typo formating -> formatting
  - ca4450d881 Fix typo wont -> won't
  - cd72b12161 Fix typo thats -> that's
  - 6db9724ec7 Fix typo doesnt -> doesn't
  - 58d46a7e98 Fix typo certficate -> certificate
- **ISSUE** #30887: (anlutro) salt-ssh fails on import msgpack - 2015.8 (refs: #30889)
- **PR** #30889: (anlutro) Make msgpack an optional dependency in salt.utils.cache @ *2016-02-04 20:53:39 UTC*
  - cdca33021a Merge pull request #30889 from alprs/fix-cache_msgpack_optional
  - ab7aae3221 make msgpack an optional dependency in salt.utils.cache
- **ISSUE** #6602: (corywright) Add ability to match on nodegroups to the compound matcher (refs: #30896)
- **ISSUE** #25292: (lichtamberg) Nodegroup matching in pillars via salt-SSH? (refs: #30896)
- **PR** #30896: (vutny) Update nodegroups parameter examples in master config example and docs @ *2016-02-04 20:52:35 UTC*
  - 0dff45b4ac Merge pull request #30896 from vutny/nodegroups-in-master-config-example
  - 936c1ff6c8 Add explanation about *N@* classifier. Inspired by #25292
  - 8bc2426816 Update example in master config documentation reference
  - ca8c0bdc3f Update nodegroups section example in master config according to docs
- **ISSUE** #30792: (bender-the-greatest) Specifying version in pkgs list returns failure even though it succeeds (on Ubuntu) (refs: #30898)
- **PR** #30898: (abednarik) Fix pkg install with version. @ *2016-02-04 20:52:14 UTC*
  - 33a400e943 Merge pull request #30898 from abednarik/fix_pkg_version_debian_family
  - b15cdfd799 Fix pkg install with version.
- **ISSUE** #30843: (HeathNaylor) SALT.STATES.BOTO_ELB register_instances error (refs: #30867)

- **ISSUE** #30808: (Reiner030) Nice2have: better boto error handling when AWS service isn't available (here: some authentication problems) (refs: #30867)
- **ISSUE** #30300: (AkhterAli) boto_route53 __salt__ not defined. (refs: #30867, #30920)
- **PR** #30867: (rallytime) Pass in 'pack' variable to utils.boto.assign_funcs function from ALL boto modules @ *2016-02-04 18:37:05 UTC*
  - **PR** #30279: (cachedout) Allow modules to be packed into boto utils (refs: #30867)
  - 89bac9076a Merge pull request #30867 from rallytime/boto-utils-fix
  - 6ad7642f6d Lint
  - 58778dfc88 Fix failing boto_vpc module unit tests
  - adb85892de Fix failing state module tests
  - b5ec0991b0 Pylint fix
  - c26c01568f Don't use pack=pack. Just pass in pack=__salt__ always.
  - 6146209c53 Pass in 'pack' variable to utils.boto.assign_funcs function from ALL boto modules.
- **ISSUE** #30798: (tbaker57) salt/utils/aws.py has Python 2.7 dependency (refs: #30849)
- **PR** #30849: (jfindlay) utils.aws: use time lib to conver to epoch seconds @ *2016-02-03 22:47:31 UTC*
  - 276cf626b0 Merge pull request #30849 from jfindlay/aws_seconds
  - 17ae74dab1 utils.aws: use time lib to conver to epoch seconds
- **ISSUE** #30869: (Ch3LL) git pillar: do not see all pillar data with multiple repos in 2015.8.4 (refs: #30874)
- **PR** #30874: (terminalmage) Fix regression in git_pillar when multiple remotes are configured @ *2016-02-03 22:24:02 UTC*
  - 4cbc8a8250 Merge pull request #30874 from terminalmage/issue30869
  - 9cf0c8126d Fix regression in git_pillar when multiple remotes are configured
- **ISSUE** #30814: (gpenin) [2015.8.*][Ubuntu 12.04 LTS][dpkg.py] Invalid "${binary:Package}" field in dpkg-query (refs: #30850)
- **PR** #30850: (jfindlay) modules.dpkg._get_pkg_info: allow for ubuntu 12.04 @ *2016-02-03 16:33:26 UTC*
  - 8410842aea Merge pull request #30850 from jfindlay/dpkg_var
  - d53a88762e modules.dpkg._get_pkg_info: handle older ubuntu
  - d3c6732539 modules.dpkg._get_pkg_info: use pythonic initializers
- **PR** #30852: (replicant0wnz) Added more descriptive error message @ *2016-02-03 16:30:15 UTC*
  - 9a3ec9d028 Merge pull request #30852 from replicant0wnz/error-message-libgit
  - c3649023b5 Added more descriptive error message
- **PR** #30847: (terminalmage) Backport #30844 to 2015.8 branch @ *2016-02-03 16:26:46 UTC*
  - **PR** #30844: (terminalmage) Perform initial gitfs/git_pillar fetch when init'ing remotes on masterless minion (refs: #30847)
  - **PR** #30703: (kraney) Fix for gitfs ext_pillar on standalone minion (refs: #30844)
  - 0338f445d9 Merge pull request #30847 from terminalmage/bp-30844
  - 58c4c01743 Add __role to master opts for gitfs integration tests
  - 17dfec2dd4 Only perform initial fetch when running on a minion

  - 53c4b4aaa4 gitfs: add initial fetch to pygit2 and dulwich
  - 78f92e9ab2 Fix for gitfs ext_pillar on standalone minion
- **PR** #30860: (vutny) Correct installation documentation for RHEL-based distributions @ *2016-02-03 16:13:09 UTC*
  - e51182495c Merge pull request #30860 from vutny/correct-doc-install-on-rhel
  - 6648fd4c62 Correct links to Fedora COPR repositories
  - 083037fccc Remove duplicate post-installation tasks section
- **PR** #30841: (jacobhammons) Release notes for 2015.8.5 @ *2016-02-03 00:04:05 UTC*
  - f1cf027308 Merge pull request #30841 from jacobhammons/release-notes
  - 6d0562ef86 Release notes for 2015.8.5
- **ISSUE** #30820: (Supermathie) State runs involving watch_in or extending break on 2015.8.4 (refs: #30837, #30835, #30833)
- **PR** #30835: (terminalmage) Integration test for #30820 @ *2016-02-02 23:51:53 UTC*
  - f8ac6002d3 Merge pull request #30835 from terminalmage/issue30820
  - ef14956db0 Integration test for #30820
- **ISSUE** #30820: (Supermathie) State runs involving watch_in or extending break on 2015.8.4 (refs: #30837, #30835, #30833)
- **PR** #30837: (jacobhammons) Added known issue #30820 to release notes @ *2016-02-02 22:33:43 UTC*
  - e0901854ce Merge pull request #30837 from jacobhammons/release-notes
  - 29e12a7fef Added known issue #30820 to release notes
- **ISSUE** #28790: (jfindlay) add grains (and others?) to salt modindex (refs: #30832)
- **PR** #30832: (rallytime) Add grains modules to salt modindex @ *2016-02-02 21:47:46 UTC*
  - b512c7757a Merge pull request #30832 from rallytime/fix-28790
  - ca044dd201 Add grains modules to salt modindex
- **ISSUE** #28971: (belt-ascendlearning) if the user exists, but has no permissions, rabbitmq_user.list_user_permissions() blows (refs: #30822)
- **PR** #30822: (rallytime) Make sure setting list_user_permissions to [", ", "] doesn't stacktrace @ *2016-02-02 21:42:26 UTC*
  - 75db37a97d Merge pull request #30822 from rallytime/rabbitmq-user-state-fixes
  - 272cc653ca Make sure setting list_user_permissions to [", ", "] doesn't stacktrace
  - a7afa7a368 Don't return a set() when checking for new tags in rabbitmq_user state
- **ISSUE** #30820: (Supermathie) State runs involving watch_in or extending break on 2015.8.4 (refs: #30837, #30835, #30833)
- **PR** #30833: (terminalmage) Fix regression in scanning for state with 'name' param @ *2016-02-02 21:25:09 UTC*
  - 557766f20b Merge pull request #30833 from terminalmage/issue30820
  - be3b8e2be6 Fix regression in scanning for state with 'name' param
- **ISSUE** #30722: (yannis666) mine config is not merged from minion config and pillar (refs: #30757, #30823)
- **PR** #30823: (yannis666) Fix for mine to merge configuration on update. @ *2016-02-02 20:21:24 UTC*

  - ec4e2bb9bb Merge pull request #30823 from yannis666/fix-for-mine-update-merge2
  - 99c7c12aba Fix for mine to merge configuration on update. This fix was previously applied to 2015.5. It fixes #30722
- **PR** #30827: (jacobhammons) Version to 2015.8.4, added CVE 2016-1866 to release notes @ *2016-02-02 20:03:31 UTC*
  - d24b9f1ea1 Merge pull request #30827 from jacobhammons/release-2015.8
  - dfc1f7a57d Version to 2015.8.4, added CVE 2016-1866 to release notes
- **ISSUE** #30809: (anlutro) Master configuration "pillar_merge_lists" has no effect (refs: #30813)
- **ISSUE** #29601: (seanjnkns) pillars not merging properly with 2015.8.3 (refs: #30062)
- **PR** #30813: (anlutro) Properly set the default value for pillar_merge_lists @ *2016-02-02 19:53:52 UTC*
  - **PR** #30458: (rallytime) Back-port #30062 to 2015.8 (refs: #30813)
  - **PR** #30062: (seanjnkns) Remove recurse_list from pillar_source_merging_strategy and add pilla… (refs: #30813, #30458)
  - f83845d7c3 Merge pull request #30813 from alprs/fix-pillar_merge_list_default
  - ec34cabee8 Properly set the default value for pillar_merge_lists
- **PR** #30826: (cachedout) Fix 30682 @ *2016-02-02 19:40:05 UTC*
  - a3feba4a26 Merge pull request #30826 from cachedout/fix_30682
  - 3b246db0b0 Fix stupid test
  - 12dc677628 Changed list conversion to use correct method and return whole set
  - 97eb4b8bf7 Pop values from new_tags set before loading into dict value
- **PR** #30818: (rallytime) Back-port #30790 to 2015.8 @ *2016-02-02 18:57:55 UTC*
  - **PR** #30790: (xmj) salt/modules/sysrc.py: Fix documentation for set_ (refs: #30818)
  - b25b845d05 Merge pull request #30818 from rallytime/bp-30790
  - c7c66afd0c salt/modules/sysrc.py: Fix documentation for set_
- **ISSUE** #30604: (vutny) Reactor overwrites *user* argument when calling runner or wheel module (refs: #30815)
- **PR** #30815: (vutny) Pick right user argument for updating reactor function's low data @ *2016-02-02 16:50:23 UTC*
  - 3cb7a9ee54 Merge pull request #30815 from vutny/reactor-low-data-fix
  - 4d4d67f9ac Pick right user argument for updating reactor function's low data
- **ISSUE** #30676: (bwillcox) testsystemd.sh tries to use 'which' that does not exist in centos 7 lxc rootfs (refs: #30747)
- **PR** #30747: (jfindlay) modules.lxc.running_systemd: use *command -v* not *which* @ *2016-02-02 14:54:17 UTC*
  - 36752906c4 Merge pull request #30747 from jfindlay/lxc_which
  - f8f867570f modules.lxc.running_systemd: use *command -v* not *which*
- **PR** #30800: (twangboy) Ability to handle special case installations @ *2016-02-02 14:25:44 UTC*
  - 8abb5b30ad Merge pull request #30800 from twangboy/chrome
  - fe0747c14e Fix another typo
  - 2815efc522 Fixes spelling

  - 6027e1ec53 Updates documentation to reflect new features
  - 1444ab1a48 Adds return success/failure for reg.broadcast_change
  - f2a36904d2 Fixes problem with missing key in old
  - 581a4df523 Added logic for dealing with latest in remove
  - c4357a6d80 Adds more logic for detecting latest
  - 40a66a2501 Logic for handling version: latest
  - b7dadd3b9b Fixes message formatting
  - a305c8ceae Added more descriptive failure message
  - fe49dcb57c Added broadcast change to force registry update
- **PR** #30794: (rallytime) A spelling fix and some spacing fixes for the boto_ec2 module docs @ *2016-02-01 21:45:33 UTC*
  - 7b44c0844d Merge pull request #30794 from rallytime/boto_ec2-mod-doc-fix
  - 5188bc4b96 A spelling fix and some spacing fixes for the boto_ec2 module docs
- **ISSUE** #23789: (hoonetorg) log output of salt orchestrate run changed between 2014.7.5 and 2015.5.0 significantly - hard to debug (refs: #30756)
- **PR** #30756: (basepi) [2015.8] Fix two error conditions in the highstate outputter @ *2016-02-01 21:39:23 UTC*
  - 1f87ad0387 Merge pull request #30756 from basepi/highstate.outputter.23789
  - 16ad24d42c Import the logger
  - 1b5c6a240c Handle non-string types in comment
  - 11e34d047b Ensure rdurations are all floats for the highstate outputter
- **PR** #30788: (rallytime) Fix incorrect doc example for dellchassis blade_idrac state @ *2016-02-01 21:20:29 UTC*
  - 46adb2d1af Merge pull request #30788 from rallytime/fix-dellchassis-doc-example
  - bfc16d9f7a Fix incorrect doc example for dellchassis blade_idrac state
- **ISSUE** #29161: (jefferyharrell) saltmod.state's ret argument seems to do nothing (refs: #30791, #29207)
- **PR** #30791: (Ch3LL) do not shadow ret function argument for salt.function @ *2016-02-01 20:07:31 UTC*
  - **PR** #29207: (jfindlay) do not shadow ret function argument (refs: #30791)
  - 333041aeb1 Merge pull request #30791 from Ch3LL/2015.8
  - d54f220c0a do not shadow ret function argument for salt.function
- **ISSUE** #30706: (carsonoid) minion traceback when Log4mongo installed but not configured (refs: #30726)
- **PR** #30726: (sjmh) Fix improper use of yield in generator @ *2016-02-01 18:13:24 UTC*
  - ce3be26e8f Merge pull request #30726 from sjmh/fix/log4mongo
  - d501f1cc03 Fix improper use of yield in generator
- **PR** #30752: (terminalmage) Backport systemd and yum/dnf optimizations from develop into 2015.8 @ *2016-02-01 18:11:42 UTC*
  - a49b75e065 Merge pull request #30752 from terminalmage/zh459
  - 8a836c88f4 Update systemd tests
  - 54ddb92474 Backport yum/dnf optimizations from develop into 2015.8

  - 1ec13699b6 Backport systemd optimizations from develop into 2015.8
- **PR** #30759: (thusoy) Allow managing empty files @ *2016-01-31 19:06:37 UTC*
  - ea15628446 Merge pull request #30759 from thusoy/empty-files
  - c6244b46ac Allow managing empty files
- **PR** #30758: (thusoy) Support mounting labelled volumes with multiple drives @ *2016-01-31 19:04:03 UTC*
  - 120d8344e4 Merge pull request #30758 from thusoy/multi-device-mount
  - 9a6dc4898f Support mounting labelled volumes with multiple drives
- **PR** #30686: (cachedout) Master-side pillar caching @ *2016-01-31 18:52:47 UTC*
  - 9e8af2f994 Merge pull request #30686 from cachedout/pillar_cache_2015_8
  - 02d8ff626a Pillar cache for master
- **ISSUE** #30662: (JoaquinVeira) UnicodeDecodeError on 2015.8 (refs: #30675)
- **PR** #30675: (jfindlay) handle non-ascii minion IDs @ *2016-01-29 23:12:10 UTC*
  - 4008e1719a Merge pull request #30675 from jfindlay/decode_id
  - 8f6737b6c4 output.key: decode minion ids to unicode
  - 7a16f1c941 config: decode id to unicode
- **ISSUE** #29602: (multani) cloud.action start raises ”got an unexpected keyword argument 'kwargs'” (refs: #30691)
- **PR** #30691: (rallytime) Make sure we use the ”instance” kwarg in cloud.action. @ *2016-01-29 23:11:37 UTC*
  - **PR** #30217: (pass-by-value) Make sure cloud actions can be called via salt run (refs: #31061, #30691)
  - 5ca75fbdc9 Merge pull request #30691 from rallytime/cloud-action-instance
  - 0873a41601 Make note of empty dict return in docstring
  - 64a73502ed Make sure we just the ”instance” kwarg in cloud.action.
- **PR** #30713: (rallytime) Fix-up autodoc proxy modules for consistency @ *2016-01-29 23:10:54 UTC*
  - 7c632d61d3 Merge pull request #30713 from rallytime/proxy-module-docs
  - 86c3f2016e Fix-up autodoc proxy modules for consistency
- **ISSUE** #30654: (Horgix) Misleading locale(mod) module behavior (refs: #30741)
- **PR** #30741: (jfindlay) states.locale.__virtual__: return exec mod load err @ *2016-01-29 23:00:41 UTC*
  - 1f5f41cc07 Merge pull request #30741 from jfindlay/locale_state
  - a3a2a44735 states.locale.__virtual__: return exec mod load err
- **PR** #30751: (basepi) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2016-01-29 22:43:41 UTC*
  - 716c2bb7c8 Merge pull request #30751 from basepi/merge-forward-2015.8
  - 84eeab7720 Merge remote-tracking branch 'upstream/2015.5' into merge-forward-2015.8
  - 076268089a Merge pull request #30699 from abednarik/save_load_retry_time
    - 186872cf49 Add Retry to save_load.
  - 8d79d1b9c7 Merge pull request #30659 from sjmh/fix-scsi
    - 3544dd995e Fix lsscsi issues for certain platforms

- **PR** #30720: (clinta) x509.pem_managed does not return changes dict @ *2016-01-29 17:07:26 UTC*
  - 1f0d0f591e Merge pull request #30720 from clinta/fix-pem-managed-changes
  - 5c28efa9d3 return changes on test as well
  - e611f0269c fix typos and no changes returned for pem_managed
- **PR** #30687: (clarkperkins) Setting 'del_root_vol_on_destroy' changes the root volume type to 'standard' @ *2016-01-28 00:02:26 UTC*
  - **PR** #30677: (clarkperkins) Fix EC2 volume creation logic (refs: #30687)
  - 36db0f99ed Merge pull request #30687 from clarkperkins/bugfix/del-root-vol-loses-type
  - a71e181c18 Don't set on a volume when creating from a snapshot
  - 8cef43c68d When setting del_root_vol_on_destroy, preserve the existing volumeType on the AMI
- **ISSUE** #28257: (peterzalewski) git_pillar remote with multiple branches yields conflicting cachedirs or check-out conflict (refs: #30673)
- **PR** #30673: (terminalmage) Properly derive the git_pillar cachedir from the id instead of the URL @ *2016-01-27 23:52:01 UTC*
  - 690b8d26b9 Merge pull request #30673 from terminalmage/issue28257
  - 8b5933fab4 Properly derive the git_pillar cachedir from the id instead of the URL
  - 62654ade1d Add additional reason for pillar env being found
- **PR** #30666: (cachedout) Fix grains cache @ *2016-01-27 22:23:12 UTC*
  - 9f0e97693c Merge pull request #30666 from cachedout/grains_cache_fix
  - 52716694f5 Fix grains cache
- **PR** #30623: (twangboy) Added service.config function @ *2016-01-27 21:08:12 UTC*
  - 8b17c77d72 Merge pull request #30623 from twangboy/add_config
  - c70e182cdf Fixed indenting… got messed up somehow…
  - 246f75f2dd Renamed variables, updated docs, added tag
  - a4534ee94c Fixed documentation
  - 54b50236a6 Fixed another error
  - 76a0cf33e5 Fixed syntax error
  - 3937380b79 Added service.config function
- **PR** #30678: (rallytime) Back-port #30668 to 2015.8 @ *2016-01-27 20:39:25 UTC*
  - **PR** #30668: (multani) Fix salt.modules.mount documentation (refs: #30678)
  - 6af1927bd3 Merge pull request #30678 from rallytime/bp-30668
  - 7c7076e6af Fix salt.modules.mount documentation
- **PR** #30677: (clarkperkins) Fix EC2 volume creation logic (refs: #30687) @ *2016-01-27 18:09:29 UTC*
  - 6c71b29f25 Merge pull request #30677 from clarkperkins/bugfix/ec2-volume-logic
  - bfec052e7d Added some extra documentation
  - ed2eee8e39 Allow volume params to be set even when specifying a snapshot
- **ISSUE** #18980: (lrhazi) salt-cloud: ExtraData: unpack(b) received extra data. (refs: #30671)

- **PR** #30680: (cro) Merge forward from 2015.5, primarily for #30671 @ *2016-01-27 17:56:48 UTC*
  - **PR** #30671: (techhat) Add file locking to cloud index (refs: #30680)
  - 36142390d4 Merge pull request #30680 from cro/mf20155-20158-20160127
  - f8ae3a20ff Merge remote-tracking branch 'upstream/2015.5' into mf20155-20158-20160127 Mergeforward from 2015.5.
  - 516919525a Merge pull request #30671 from techhat/lockcloud
    - 4719f8d4ea Whitespace
    - 8e7eca23e4 Add file locking to cloud index
- **PR** #30663: (isbm) Zypper: latest version bugfix and epoch support feature @ *2016-01-27 17:10:42 UTC*
  - f6feddecb4 Merge pull request #30663 from isbm/isbm-zypper-latest-versionfail
  - 4336487765 Add support for epoch in Zypper
  - 12d515fa0c Fix package status filtering on latest version
- **PR** #30652: (mew1033) Fix sh beacon @ *2016-01-27 17:00:29 UTC*
  - 9d8ddeb525 Merge pull request #30652 from mew1033/fix-sh-beacon
  - 256d037e0f Fix sh beacon
- **ISSUE** #29678: (dschaller) NPM Install Forces Silent (refs: #29650)
- **PR** #30657: (jfindlay) [2015.8] Backport #30378 and #29650 @ *2016-01-27 00:34:00 UTC*
  - **PR** #30378: (dschaller) Adding silent flag to npm.bootstrap (refs: #30657)
  - **PR** #29650: (dschaller) Adding ability to disable npm install silent flag (refs: #30657)
  - 1fa1963895 Merge pull request #30657 from jfindlay/backport_quiet
  - ca4adbf382 Adding ability to disable npm install silent flag
  - afe149eb6d Adding ability to disable npm install silent flag
  - c1101b5f0b Adding ability to disable npm install silent flag
  - d29ad8bbf6 Adding ability to disable npm install silent flag
  - 7a21dbf0d9 Adding silent flag to npm.bootstrap
  - 354c0bdf26 Adding silent flag to npm.bootstrap
- **PR** #30656: (rallytime) [2015.8] Merge 2015.5 into 2015.8 @ *2016-01-27 00:33:30 UTC*
  - 3621651bf8 Merge pull request #30656 from rallytime/merge-forward-2015.8
  - 76ab6981a5 Merge branch '2015.5' into 2015.8
  - 643c9c9616 Merge pull request #30586 from abednarik/fix_comment_line_perms
  - 8b395a42cb Fix comment_line permissions.
- **PR** #30644: (tbaker57) Another go at fixing 30573 @ *2016-01-26 20:18:41 UTC*
  - 30e03a8b0c Merge pull request #30644 from tbaker57/another_go_at_30573
  - 267b8827fd Another go at fixing 30573
- **PR** #30611: (isbm) Bugfix: Zypper *pkg.latest* crash fix @ *2016-01-26 16:35:47 UTC*
  - 7d307e2a04 Merge pull request #30611 from isbm/isbm-zypper-latest

  - a7141be651 Put 'kwargs' on its own line according to the common pattern
  - ee9b3f859b Bugfix: do not treat SLS id as a package name if an empty 'pkgs' list specified.
  - d3cfd8ed41 Cleanup formatting
  - 1bdbaac658 Add error handling
  - 2ec5cec8a4 Add a new line before the last return
  - 424383b8c4 Remove unnecessary complexity and string increment
  - 48e8d90343 Avoid backslashes where they are not needed
  - 6df5d500f0 Use regexp type for the string.
  - c2ca141956 Get version as an explicit parameter
  - 9e944db706 Check the version of the package, instead of the package name
  - 59ea758efb Fix formatting
  - 514f6349d4 Bugfix: crash on "key not found" error
  - ea75f55a1a Fix PEP8: line continuation
  - ece35ebc26 Replace old fashion string memcopy with the list
  - 716445e588 Fix PEP8: line continuation
  - 0f11079ff9 Fix PEP8 for the operator
- **ISSUE** #7811: (kiall) RabbitMQ Cluster/Plugins/Policy etc states do not track changes, preventing "watch" from working (refs: #30631)
- **PR** #30631: (rallytime) Refactor rabbitmq_cluster states to use test=true functionality correctly @ *2016-01-26 16:23:49 UTC*
  - 5bc11d7539 Merge pull request #30631 from rallytime/fix-7811
  - bf9ffded6d Refactor rabbitmq_cluster states to use test=true functionality correctly
- **ISSUE** #25658: (tsaridas) rabbitmq_policy.present state (refs: #30628)
- **PR** #30628: (rallytime) Refactor rabbitmq_policy states to use test=true functionality correctly @ *2016-01-26 00:21:03 UTC*
  - ef6c4e8377 Merge pull request #30628 from rallytime/fix-25658
  - 1e8e86007c Refactor rabbitmq_policy states to use test=true functionality correctly
- **PR** #30624: (cro) Remove bad symlinks from osx pkg dir @ *2016-01-26 00:02:25 UTC*
  - 80d0e428aa Merge pull request #30624 from cro/remove_bad_symlinks
  - f5fd38624e Remove bad symlinks in osx pkg dirs
- **ISSUE** #30621: (zer0def) Current latest (2015.8.3) list of builtin states docu doesn't list 'glance' (refs: #30622)
- **PR** #30622: (rallytime) Add glance state to list of state modules @ *2016-01-25 23:55:54 UTC*
  - 330ea9a292 Merge pull request #30622 from rallytime/fix-30621
  - 57b7e6cc93 Add glance state to list of state modules
- **ISSUE** #19288: (oba11) AssociatePublicIpAddress doesnt work with salt-cloud 2014.7.0 (refs: #20972, #30591)
- **PR** #30618: (rallytime) Back-port #30591 to 2015.8 @ *2016-01-25 23:55:20 UTC*
  - **PR** #30591: (mlalpho) salt-cloud-clouds-ec2 AssociatePublicIpAddress fix (refs: #30618)

  - **PR** #20972: (JohannesEbke) Fix interface cleanup when using AssociatePublicIpAddress in #19288 (refs: #30591)
  - f00d8f398a Merge pull request #30618 from rallytime/bp-30591
  - 2c9d59fa42 looks like a re-merge of PR #20972 which relates to #19288
- **ISSUE** #30587: (sjorge) [docs] docs confusing on client_acl and external_auth usage (refs: #30625)
- **PR** #30625: (jfindlay) doc.topics.eauth: clarify client_acl vs eauth @ *2016-01-25 23:03:24 UTC*
  - 6b940d9655 Merge pull request #30625 from jfindlay/eauth_acl
  - b5e2cff028 doc.topics.eauth: clarify client_acl vs eauth

28.2.63 Salt 2015.8.8.2 Release Notes

Version 2015.8.8.2 is a bugfix release for *2015.8.0*.

Fixes to 2015.8.8

Salt 2015.8.8.2 includes fixes for the following known issues in 2015.8.8:

- Key master with value [...] has an invalid type of list Error (issue #32044)
- Failed to import module win_dacl Error (issue #32004)
- Wrong validation type for file_ignore_glob key (issue #32114)
- Fix file.managed for windows (issue #31969)

Important: issue #32183 prevents Salt Cloud from installing the Salt minion on new systems. To workaround this issue, call `salt-cloud -u` to update the bootstrap script to the latest version.

Changelog for v2015.8.8..v2015.8.8.2

Generated at: 2018-05-28 00:29:12 UTC

- 403563e441 Change type check errors to debug loglevel
- 8323005b3d Support multiple valid option types when performing type checks
- 2f95082a96 Fixed validation type for file_ignore_glob Fixes #32114
- 2685e61d9e Move constant declaration into member variable to avoid issues when modules can't be loaded.
- bc10d7dede Add apply_template_on_contents for windows

28.2.64 Salt 2015.8.9 Release Notes

Version 2015.8.9 is a bugfix release for *2015.8.0*.

Statistics

- Total Merges: **145**
- Total Issue References: **110**
- Total PR References: **264**
- Contributors: **71** (Ch3LL, DmitryKuzmenko, DylanFrese, Ferbla, Kurocon, Lothiraldan, RuriRyan, Talkless, The-Loeki, UtahDave, Xiami2012, abednarik, afletch, ahammond, ahus1, aletourneau, alxf, amontalban, anlutro, arthurlogilab, atengler, basepi, bdrung, bradthurber, cachedout, captaininspiration, cedwards, clarkperkins, clinta, cro, dmurphy18, exowaucka, garethgreenaway, guettli, idonin, isbm, jacobhammons, jbonachera, jfindlay, jfray, junster1, justinta, krak3n, lalmeras, lloydoliver, lomeroe, mcalmer, mitar, mrproper, multani, nmadhok, notpeter, onorua, paclat, papertigers, rallytime, rkgrunt, sakateka, sbreidba, schancel, sjorge, stk0vrfl0w, techhat, terminalmage, thatch45, ticosax, tomlaredo, twangboy, twellspring, vutny, whiteinge)

Important Post-Upgrade Instructions for Linux Mint

As a result of some upstream changes, the `os` grain on Mint Linux is now being detected as `LinuxMint` (issue #33295). Run the following command **after you upgrade to 2015.8.9** to reset the `os` grain to `Mint` and the `os_family` grain to `Debian`:

```
salt -G 'os:LinuxMint' grains.setvals "{'os': 'Mint', 'os_family': 'Debian'}"
```

Changelog for v2015.8.8.2..v2015.8.9

Generated at: 2018-05-28 00:36:04 UTC

- **PR** #33310: (jfindlay) update 2015.8.9 release notes
- **PR** #33293: (twangboy) Fix minion start retry on Windows (2015.8) @ *2016-05-17 17:03:41 UTC*
  - e3eff27c55 Merge pull request #33293 from twangboy/fix_33277_2015_8
  - 652f0079db Fix minion start retry on Windows
- **ISSUE** #31270: (4001982248998) acl.present: TypeError on subsequent runs (refs: #33172)
  - **PR** #33305: (rallytime) Back-port #33172 to 2015.8
  - **PR** #33172: (Kurocon) linux_acl: Allow '-' as a separation character in ACL permissions. Fi… (refs: #33305)
- **ISSUE** #33299: (jbonachera) salt-cloud: scp_file() and sftp_file() don't work with ipv4-only hosts (refs: #33300)
- **ISSUE** #33243: (jbonachera) salt-cloud: wait_for_port() doesn't work with ipv4-only hosts (refs: #33246, #33300)
  - **PR** #33300: (jbonachera) Handle more ipv6 error as an exception #33299
- **ISSUE** #26062: (silenius) service.status is broken under FreeBSD (refs: #33294)
- **ISSUE** #23435: (JaseFace) service.status currently reports an error on FreeBSD if the service isn't running (refs: #33294)
  - **PR** #33294: (terminalmage) Ignore retcode when checking service's status
- **PR** #33274: (rallytime) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2016-05-16 16:41:32 UTC*
  - 06edba448e Merge pull request #33274 from rallytime/merge-2015.8

  - bf641d3a66 Merge branch '2015.5' into '2015.8'
  - 8fa72f6588 Clarify file.replace MULTILINE flag interaction with regex anchors (#33137)
  - 4b1f460256 update 2015.5.11 release notes (#33236)
- **ISSUE** #30258: (rallytime) Changes dictionary return should be mentioned in test state docs (refs: #33254)
  - **PR** #33254: (rallytime) Add comment for test=true w/o changes ret and add changes dict example
- **ISSUE** #30946: (rallytime) Update SaltStack Git Policy Documentation (refs: #33252)
  - **PR** #33252: (rallytime) Update Git Policy docs to match Contribution guide
- **ISSUE** #33238: (clinta) x509 CSR fails if the csr does not contain any extensions (refs: #33239)
  - **PR** #33239: (clinta) Fix #33238
  - **PR** #33245: (terminalmage) Backport #33244 to 2015.8
  - **PR** #33244: (terminalmage) Properly report on invalid gitfs/git_pillar/winrepo repos (refs: #33245)
  - **PR** #32238: (ticosax) [gitfs] only 2 argument are passed to this template when render error message (refs: #33244, #33245)
- **ISSUE** #30605: (eyj) Update development/conventions/release.rst docs - they're out of date with the current process. (refs: #33253)
- **PR** #33253: (rallytime) Update the release process docs @ *2016-05-13 21:28:11 UTC*
  - 94a53da92e Merge pull request #33253 from rallytime/fix-30605
  - a129d05b6d Update the release process docs
  - **PR** #33251: (jfindlay) update 2015.8.9 release notes
- **ISSUE** #33243: (jbonachera) salt-cloud: wait_for_port() doesn't work with ipv4-only hosts (refs: #33246, #33300)
  - **PR** #33246: (techhat) Handle ipv6 error as an exception
- **ISSUE** #33073: (robnagler) TypeError: unhashable type: 'dict' (refs: #33213)
  - **PR** #33213: (terminalmage) Check rendered YAML for invalid keys
- **ISSUE** #21903: (basepi) Document _file.conf pattern for master.d/ and minion.d/ (refs: #33224)
  - **PR** #33224: (rallytime) Make note of files that begin with '_' in master.d or minion.d dirs
- **ISSUE** #31975: (rajvidhimar) Docstrings not reflected in the salt documenation. (refs: #33150)
  - **PR** #33150: (rallytime) Gate jnpr imports in salt.proxy.junos.py
- **ISSUE** #21315: (ryan-lane) No example documentation for http.query state (refs: #33222)
  - **PR** #33222: (rallytime) Add docs for the http state
- **ISSUE** #29796: (vutny) Fail to use 'highstate' outputter explicitly (refs: #33215)
  - **PR** #33215: (rallytime) Don't stacktrace when using –out=highstate at CLI during state run.
  - **PR** #33219: (lalmeras) propagate opts to salt.util.http call
  - **PR** #33154: (lalmeras) propagate opts to salt.util.http call (refs: #33219)
  - **PR** #33237: (jfindlay) update 2015.8.9 release notes
- **PR** #33217: (rallytime) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2016-05-12 22:45:39 UTC*
  - 6dc5d605b1 Merge pull request #33217 from rallytime/merge-forward-2015.8

  - 4655607b58 Merge branch '2015.5' into '2015.8'
  - 698f1eb657 Merge pull request #33211 from cachedout/user_kill
    - d4f2e5baa7 Don't try to kill a parent proc if we can't
  - f86832911e Resolve issue with pkg module on Mint Linux (#33205)
  - a09e1b6335 Add pip installed and removed test (#33178)
  - 96e3586f12 update 2015.5.11 release notes (#33197)
  - 09b072a412 Fix file.managed for Windows (#33181)
  - **PR** #33207: (rallytime) [2015.8] Merge forward from 2015.5 to 2015.8
- **ISSUE** #32917: (bradthurber) standalone minion pygit2 pillar data doesn't refresh without manual git fetch (refs: #33204)
  - **PR** #33204: (terminalmage) Add a fetch when compiling git_pillar for masterless minions
- **ISSUE** #33162: (jfindlay) Key error with salt.utils.cloud.cache_node and EC2 (refs: #33164)
  - **PR** #33164: (jfindlay) cloud.clouds.ec2: cache each named node
- **ISSUE** #32385: (aronneagu) git.latest throws expected string or buffer (refs: #33203)
  - **PR** #33203: (terminalmage) Properly handle failed git commands when redirect_stderr=True
- **ISSUE** #32685: (gidantribal) git state does not take into account ssh config file (refs: #33152)
  - **PR** #33152: (terminalmage) Don't force use of global ssh_config when git identity file is specified
  - **PR** #33198: (jfindlay) update 2015.8.9 release notes
- **PR** #33188: (rallytime) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2016-05-11 22:32:29 UTC*
  - 6177a6a36f Merge pull request #33188 from rallytime/merge-2015.8
  - f12bba6ebc Merge branch '2015.5' into '2015.8'
  - 30868ab06c [2015.5] Update to latest bootstrap script v2016.05.11 (#33185)
  - 264ad34b3b Pip fix (#33180)
  - 43288b268d add 2015.5.11 release notes (#33160)
  - e0da8fda7d [2015.5] Update to latest bootstrap script v2016.05.10 (#33155)
  - **PR** #33161: (jfindlay) add 2015.8.9 release notes
  - **PR** #33156: (rallytime) [2015.8] Update to latest bootstrap script v2016.05.10
- **ISSUE** #25040: (yi9) grains.get can't get minion's /etc/salt/grains value in multi-master set up (refs: #33142)
  - **PR** #33142: (cachedout) Hash fileclients by opts
- **ISSUE** #22142: (multani) State *acl.present* doesn't allow to set "default" ACLs (refs: #31769)
  - **PR** #33139: (rallytime) Back-port #31769 to 2015.8
  - **PR** #31769: (DylanFrese) Fix acl.present and acl.absent when adding default ACLs (refs: #33139)
- **PR** #33144: (rallytime) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2016-05-10 19:40:31 UTC*
  - 2800762b44 Merge pull request #33144 from rallytime/merge-2015.8
  - 449176f06e Merge branch '2015.5' into '2015.8'
  - 6cd1641840 Merge pull request #33141 from jtand/disable_local_pkg_install_test

    - 8b1e34fb17 Skipping salt-call –local test
- **PR** #33140: (rallytime) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2016-05-10 16:57:55 UTC*
  - 72d075e14e Merge pull request #33140 from rallytime/merge-2015.8
  - c732c8104b Merge branch '2015.5' into '2015.8'
  - 878d34a865 Doc mock decorators (#33132)
- **ISSUE** #32834: (beardedeagle) Masterless Minion - Unable to query job cache (refs: #33017, #33100)
  - **PR** #33100: (rallytime) If cache_jobs: True is set, populate the local job cache when running salt-call
  - **PR** #33135: (stk0vrfl0w) Fix broken parsing of usermgmt.conf on OpenBSD
  - **PR** #33129: (rallytime) Back-port #33101 to 2015.8
  - **PR** #33101: (thatch45) Add a check that the cmdline of the found proc matches (refs: #33129)
  - **PR** #33064: (terminalmage) salt.utils.gitfs: fix formatting for warning messages
- **ISSUE** #33058: (aclemetson) Unable to run "win_servermanager.list_available" on minion. (refs: #33099)
  - **PR** #33099: (twangboy) Fix 33058
- **ISSUE** #32999: (basepi) Stacktrace for *master_finger* mismatch on minion (refs: #33106)
- **PR** #33106: (abednarik) Moved _finger_fail method to parent class. @ *2016-05-09 16:31:09 UTC*
  - 8acc3147d6 Merge pull request #33106 from abednarik/abednarik_master_Finger_stacktrace
  - 91a69ba54a Moved _finger_fail method to parent class.
  - **PR** #33102: (Ch3LL) clarify docs that map is designed to be run once. is not stateful
  - **PR** #33098: (rallytime) Back-port #33061 to 2015.8
  - **PR** #33061: (ahammond) cloud.query needs to define mapper.opts (refs: #33098)
- **PR** #33096: (rallytime) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2016-05-06 19:27:57 UTC*
  - c1f7aed8a5 Merge pull request #33096 from rallytime/merge-2015.8
  - 0fd5e9d157 Merge branch '2015.5' into '2015.8'
  - 30edeadafd Lower display of msgpack failure msg to debug (#33078)
  - d4928c5a22 Use saltstack repo in buildpackage.py on CentOS 5 (#33080)
  - 61d126cb98 add test for installing package while using salt-call –local (#33025)
  - 6d3e4e8935 File and User test fixes for 2015.5 on Fedora23 (#33055)
  - d48b2b8b52 test pillar.items output (#33060)
  - 398793bfc0 Fix minor document error of test.assertion (#33067)
  - f8757631b2 Saltfile with pillar tests (#33045)
  - 1d7892421e Backport #33021 manually to 2015.5 (#33044)
  - f00b5f91b3 Add run_on_start docs to schedule.rst (#32958)
  - **PR** #32865: (idonin) salt-cloud: fix ipv6-only virtual machines
  - **PR** #33084: (jfindlay) modules.npm: do not log npm –version at info level
- **ISSUE** #33068: (pythonwood) salt-ssh do not support centos5 because old-version-python ? (refs: #33081)
- **PR** #33081: (jfindlay) ssh docs: install py-2.6 for RHEL 5 @ *2016-05-06 15:18:39 UTC*

  - 3808d05838 Merge pull request #33081 from jfindlay/ssh_doc
  - a2c927b173 ssh docs: install py-2.6 for RHEL 5
- **PR** #33088: (isbm) Bugfix: Restore boolean values from the repo configuration @ *2016-05-06 15:13:27 UTC*
  - 6d604926d3 Merge pull request #33088 from isbm/isbm-zypper-fix-booleans
  - 3ca203eb8e Bugfix (follow-up): setting priority requires non-positive integer
  - 79a46e091c Add repo config test
  - 222b8369ca Add test data for repos
  - b746fa35f0 Bugfix: Restore boolean values from the repo configuration
- **ISSUE** #12422: (creaky) Bug: file.blockreplace inserts additional blank line on multi-line content (refs: #33049)
  - **PR** #33082: (Ch3LL) Fix tests for file.blockplace to remove newline
  - **PR** #33049: (thatch45) Don't append a newline when creating new content with blockreplace (refs: #33082)
- **PR** #32892: (isbm) Resolve Zypper locks on asynchronous calls @ *2016-05-05 14:34:59 UTC*
  - fb89877cf2 Merge pull request #32892 from isbm/isbm-zypper-env-variables
  - 1601a7e07a Prevent the use of "refreshable" together with "nolock" option.
  - 52e1be2fa9 Remove unused variable in a constructor. Adjust the docstring accordingly.
  - 7e00f566ef Move log message down to the point where it actually sleeps. Rephrase the message.
  - 4b7dab83ff Fix PID file path for SLE11
  - 7f37961d4b Rename tags
  - c55b0fab58 Test DOM parsing
  - c54e928e4f Add exception handling test
  - 3d245bbe84 Parse DOM out of the box, when XML mode is called
  - 6a98f523ac Add Zypper caller test suite
  - f189f90124 Bugfix: always trigger __getattr__ to reset and increment the configuration before the call.
  - 7e1712dd80 Fix tests according to the new calling model
  - 3a30b7fbcd Remove an obsolete test case
  - 6e5877a2ee Add Zypper Call mock
  - bb5540cb4a Bugfix: inverted logic on raising (or not) exceptions
  - ce9262fe71 Make Zypper caller module-level reusable
  - 77dc8695af Update docstrings according to the bugfix
  - 46d86b21d5 Bugfix: accept refresh override param
  - cb40618262 Fire an event about released Zypper with its result
  - 0728f0bc00 Replace string values with the constants
  - 6af3f7141b Check if zypper lock exists and add more debug logging
  - 0167b30a75 Add Zypper lock constant
  - 370ff21d36 Fire an event to the Master about blocked Zypper.

  - 1727ca3de2 Use new Zypper call implementation
  - 485164aa5c Remove blocking-prone Zypper call implementation
  - f161f0612c Implement block-proof Zypper call implementation
  - baf35ed708 Remove one-char variables
  - 2c94eb016f Remove an unused variable
  - 6869ebc557 Remove an empty line
  - 7e06489da9 Remove verbose wrapping
  - 2131ff04af Standarize zypper call to "run_all"
  - 046ef44ca3 Bugfix: version_cmp crashes in CLI if there are versions, that looks like integer or float.
  - b869a92eea Change Zypper calls to a single point
- **ISSUE** #24237: (Grokzen) Minion schedule return data missing some fields (refs: #33039)
  - **PR** #33039: (The-Loeki) Add fun_args to scheduled return data (part of #24237)
- **ISSUE** #12422: (creaky) Bug: file.blockreplace inserts additional blank line on multi-line content (refs: #33049)
  - **PR** #33049: (thatch45) Don't append a newline when creating new content with blockreplace (refs: #33082)
- **ISSUE** #24996: (danlsgiga) –failhard option not working as expected (refs: #33048)
  - **PR** #33048: (rallytime) Pass all data to batch.run() call when using –failhard
- **ISSUE** #32452: (nicholascapo) cmd.run_all with –batch and –failhard gives no output on failure (refs: #33050)
  - **PR** #33050: (rallytime) Display command output when command fails with batch + failhard options
- **ISSUE** #33041: (anitakrueger) boto_elb.present security_groups kwarg is a list - needs documentation (refs: #33053)
  - **PR** #33053: (rallytime) Allow security_groups kwarg for boto_elb.present to be string or list
  - **PR** #33054: (rallytime) [2015.8] Merge forward from 2015.5 to 2015.8
  - **PR** #33056: (justinta) File and User test fixes for 2015.8 on Fedora23
- **ISSUE** #32472: (esn89) salt-minion is stuck in a restart loop with not much info: (refs: #33030)
  - **PR** #33040: (rallytime) Back-port #33030 to 2015.8
  - **PR** #33030: (thatch45) When we restart the minion we should show the error that caused it (refs: #33040)
- **ISSUE** #32834: (beardedeagle) Masterless Minion - Unable to query job cache (refs: #33017, #33100)
  - **PR** #33017: (rallytime) Update the docs for saltutil.find_job to be more clear/accurate
  - **PR** #33031: (rallytime) Back-port #33002 to 2015.8
  - **PR** #33002: (whiteinge) Add saltenv to the cmd.script state function (refs: #33031)
  - **PR** #33021: (UtahDave) Fix syndic regression (refs: #33044)
- **ISSUE** #11801: (slai) Salt does not match user names properly under Windows (refs: #32674)
  - **PR** #32674: (twangboy) Compare uid and gid instead of name and group
- **ISSUE** #32856: (DeanScothern) jjid not shown when running the salt command line with –batch-size using either –verbose or –show-jid with certain salt versions (refs: #32996)
- **ISSUE** #31738: (igorwidlinski) salt –show-jid does not show job id when run in batch mode (refs: #32450)

  - **PR** #32996: (rallytime) Allow batch mode to use verbose option, as well as show_jid.
  - **PR** #32450: (cachedout) Pass parser options into batch mode (refs: #32996)
- **ISSUE** #32954: (atengler) glusterfs.peered fails with 'NoneType' object is not iterable (refs: #32955)
  - **PR** #32955: (atengler) Fixed glusterfs.peered output
- **ISSUE** #26011: (rodriguezsergio) states.virtualenv != modules.virtualenv (refs: #32994)
  - **PR** #32994: (rallytime) Clarify some arg docs for virtualenv state
  - **PR** #32986: (justinta) Fix boto_secgroup_test
- **ISSUE** #32777: (sjorge) cron.present broken on Solarish systems if user specified (refs: #32970)
  - **PR** #32970: (sjorge) fix user cron on solarish operating systems
  - **PR** #32796: (jfindlay) salt.log.setup: process user args before format
- **ISSUE** #32891: (guettli) docs: Note " This document represents behavior exhibited by Salt requisites as of version 0.9.7 of Salt." (refs: #32934)
  - **PR** #32934: (jfindlay) doc.ref.states.ordering: clarify requisite change
- **ISSUE** #32882: (papertigers) carbon_return is missing a default value. (refs: #32883)
  - **PR** #32928: (rallytime) Back-port #32883 to 2015.8
  - **PR** #32883: (papertigers) mode should default to 'text' (refs: #32928)
- **ISSUE** #32646: (deamen) FileClient Class (client = salt.minion.FileClient(__opts__)) does not exist (refs: #32925)
  - **PR** #32925: (rallytime) Remove FileClient class references from docs - it doesn't exist.
- **ISSUE** #23683: (gravyboat) contents_grains should have an example (refs: #32922)
  - **PR** #32922: (rallytime) Update contents_grains option with relevant docs
- **PR** #32926: (rallytime) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2016-04-28 19:47:52 UTC*
  - e60c12640d Merge pull request #32926 from rallytime/merge-2015.8
  - 5a184881be Merge branch '2015.5' into '2015.8'
  - edce22a143 backport PR #32732 to 2015.5 fixes #23714 (#32848)
  - **PR** #32908: (Ch3LL) Specify EBS volume tags in profile configuration in aws
- **ISSUE** #23952: (neogenix) iptables state append doesn't honor position -1 (refs: #32906)
  - **PR** #32906: (rallytime) Update docs to warn users that -1 isn't valid for iptables insert state
- **ISSUE** #32510: (Ch3LL) Cannot specify image in provider file when using map file (refs: #32900)
  - **PR** #32900: (rallytime) Allow profile options to be specified in provider file when using maps
- **ISSUE** #30855: (guettli) Docs: does salt.states.service support systemd? (refs: #32880)
  - **PR** #32880: (rallytime) Clarify service state opening docs - uses 'service' virtualname
- **PR** #32884: (terminalmage) Fix incorrect deprecation notice @ *2016-04-27 15:47:35 UTC*
  - e1b40b3b76 Merge pull request #32884 from terminalmage/fix-incorrect-deprecation-notice
  - b307c5452a Fix incorrect deprecation notice
- **PR** #32878: (jacobhammons) added note about updating the bootstrap script in salt-cloud using th… @ *2016-04-26 21:09:51 UTC*

  - a2921b9da0 Merge pull request #32878 from jacobhammons/salt-cloud
  - 3887938727 added note about updating the bootstrap script in salt-cloud using the -u flag, removed the saltconf banner.
- **ISSUE** #32861: (bradthurber) Is it master_syndic or syndic_master? (refs: #32869)
- **PR** #32869: (rallytime) Use correct config setting in cloud syndic docs @ *2016-04-26 19:13:21 UTC*
  - 71db10fd2c Merge pull request #32869 from rallytime/fix-32861
  - 0e73daa126 Use correct config setting in cloud syndic docs
- **PR** #32844: (rallytime) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2016-04-26 17:38:08 UTC*
  - 02c681311f Merge pull request #32844 from rallytime/merge-2015.8
  - 1fc9de1d04 Add 'file.source_list' mock to archive state unit tests
  - 9064d3bbfb Merge branch '2015.5' into '2015.8'
    - 9b5c14c37c *salt-cloud -u* downloads stable version from bootstrap.saltstack.com by default (#32837)
    - 9725804448 update bootstrap to 2016.04.18 release (#32667)
    - c842e1e437 Merge pull request #32776 from rallytime/merge-2015.5
      - 7ecbf9f885 Merge pull request #14 from whiteinge/runner-async-low
      - 211f7b4af1 Format low data correct for runner_async
      - ce72851861 Merge branch '2014.7' into '2015.5'
      - 2775edc176 Saltnado /run fix (#32590)
      - b19c5a5ce7 Verify auth in saltnado run (#32552)
    - 67d0c81184 Support remote sources in a source list (#32691)
    - bd5442d768 Merge pull request #32686 from cachedout/issue_32661
      - f704df90bc Fix stacktrace in batch with dup minion ids
    - 3ec9502a86 Update "Low Hanging Fruit" to "Help Wanted" (#32675)
    - 77bea56b68 Additional documentation on calling exec modules from templates (#32657)
    - c910b8dd51 Fixing critical bug to remove only the specified Host instead of the entire Host cluster (#32639)
    - 4568565d45 Add _syspaths.py to .gitignore (#32638)
- **ISSUE** #32799: (belt) ssh_auth.present creates ~/~${USER}/.ssh (refs: #32868)
  - **PR** #32868: (rallytime) Back-port #31139 to 2015.8
  - **PR** #31139: (exowaucka) Improve %h and %u handling in SSH module (refs: #32868)
- **ISSUE** #23714: (naemono) file.copy force ignored during highstate, but not with 'salt-call state.sls_id' (refs: #32732, #32847, #32848)
  - **PR** #32847: (lomeroe) backport PR #32732 for issue #23714
  - **PR** #32732: (lomeroe) correct use of force flag in file.copy #23714 (refs: #32847, #32848)
- **ISSUE** #32824: (bradthurber) salt-cloud vmware: wrong pyvmomi installed for RHEL/CentOS 6 (refs: #32845)
  - **PR** #32845: (rallytime) Add pyvmomi version warning to Getting Started with VMware docs

- **ISSUE** #25492: (hernanc) "docker-py mem_limit has been moved to host_config in API version 1.19" error (refs: #26518, #32818)
  - **PR** #32841: (rallytime) Back-port #32818 to 2015.8
  - **PR** #32818: (mitar) Pass None as memory limit (refs: #32841)
  - **PR** #26518: (krak3n) Fix for #25492 (refs: #32818)
- **ISSUE** #32605: (Talkless) pkgrepo.managed with apt does not add comments value later (refs: #32813)
  - **PR** #32839: (rallytime) Back-port #32813 to 2015.8
  - **PR** #32813: (abednarik) Add comments as an option for apt in pkgrepo.managed. (refs: #32839)
  - **PR** #32659: (anlutro) Various improvements on cloud deploy script docs
  - **PR** #32668: (jfindlay) [2015.8] update bootstrap to 2016.04.18 release
  - **PR** #32785: (rallytime) Back-port #29322 to 2015.8
  - **PR** #29322: (mrproper) add http proxy support for tornado (refs: #32785)
- **ISSUE** #32710: (bradthurber) conf/master missing many gitfs and git_pillar parameters (refs: #32722)
- **PR** #32787: (rallytime) Back-port #32722 to 2015.8 @ *2016-04-25 15:19:21 UTC*
  - **PR** #32722: (bradthurber) Catch up the conf/master file to include gitfs/git_pillar parms from … (refs: #32787)
  - 96a3d4e556 Merge pull request #32787 from rallytime/bp-32722
  - 8d7148d41b Catch up the conf/master file to include gitfs/git_pillar parms from recent releases
- **PR** #32786: (rallytime) Back-port #32703 to 2015.8 @ *2016-04-25 15:19:13 UTC*
  - **PR** #32703: (schancel) Make example top file match templated version (refs: #32786)
  - 36f70f5847 Merge pull request #32786 from rallytime/bp-32703
  - baa4df25c9 Make example top file match templated version
  - 227ef4aabb Fix unnecessary capitalization
  - 73cd9f26c3 Merge branch 'gitfs_perremote_doc_updates' of https://github.com/l2ol33rt/salt into pr-32775
  - b69d406ada Including name per-remote config option in example
  - **PR** #32779: (terminalmage) Improve documentation on pygit2 versions
- **ISSUE** #32609: (anlutro) Tornado ioloop fails when master disconnects? (refs: #32749)
  - **PR** #32749: (DmitryKuzmenko) Properly handle minion failback failure.
- **ISSUE** #32144: (vutny) Pillar targeting starts to work only after calling *saltutil.refresh_pillar* (refs: #32643)
  - **PR** #32643: (vutny) Document pillar cache options
- **ISSUE** #32705: (joakimkarlsson) win_dacl.present: Specifying propagations for a directory fails (refs: #32720)
  - **PR** #32720: (jfindlay) modules.win_dacl: consistent case of dacl constants
- **ISSUE** #30761: (sjmh) Cannot target subsets of minions when using pillar and external_auth (refs: #31598)
- **ISSUE** #21303: (Lothiraldan) Explicit and document ACL rules format (refs: #32733)
  - **PR** #32733: (Lothiraldan) Update external auth documentation to list supported matcher.
  - **PR** #31598: (terminalmage) Remove limitations on validation types for eauth targets (refs: #32733)

  - **PR** #32693: (techhat) Check dependencies type before appling str operations
  - **PR** #32692: (garethgreenaway) Handle when beacon not configured and we try to enable/disable them
- **PR** #32718: (garethgreenaway) Fixes to schedule.list in 2015.8 @ *2016-04-20 19:51:24 UTC*
  - f52af5a596 Merge pull request #32718 from garethgreenaway/2015_8_schedule_list_fix
  - 7fa5d809d2 backporting a fix from develop where the use of splay would result in seconds=0 in the schedule.list when there was no seconds specified in the origina schedule
- **PR** #32684: (captaininspiration) Fix routes for redhat < 6 @ *2016-04-19 19:18:20 UTC*
  - **PR** #32682: (captaininspiration) Fix routes for redhat < 6 (refs: #32684)
  - f63566e452 Merge pull request #32684 from captaininspiration/2015.8
  - 640c7a90da Fix routes for redhat < 6
  - **PR** #32683: (techhat) Handle a couple of arguments better (Azure)
- **ISSUE** #32523: (junster1) network.py/loader.py failing because cfn variable is not defined before use. (refs: #32672)
  - **PR** #32672: (junster1) Fix for issue 32523
- **ISSUE** #32517: (Ch3LL) Minion restarting and erroring when cannot reach the masters in multi-master failover (refs: #32555, #32556)
  - **PR** #32556: (DmitryKuzmenko) Don't access deprecated Exception.message attribute.
  - **PR** #32655: (cachedout) Lower log level for pillar cache
- **ISSUE** #31542: (duk3luk3) jinja stringifies dict before passing it to execution module (maybe salt-ssh specific?) (refs: #32588)
- **PR** #32588: (anlutro) Fix salt-ssh module function call argument type juggling by JSON encoding them @ *2016-04-18 15:57:14 UTC*
  - a6a427463d Merge pull request #32588 from alprs/fix-salt_ssh_module_types
  - d912f1c3c6 json encode arguments passed to an execution module function call
- **ISSUE** #32229: (seanjnkns) 2015.8.8.2: pkg.installed fails to update packages with epoch (refs: #32563)
  - **PR** #32563: (terminalmage) yumpkg: Ignore epoch in version comparison for explict versions without an epoch
  - **PR** #32640: (nmadhok) [2015.8] - Fixing critical bug to remove only the specified Host instead of the entire Host cluster
  - **PR** #32649: (mcalmer) align OS grains from older SLES with current one
  - **PR** #32652: (isbm) Prevent crash if pygit2 package is requesting re-compilation of the e…
- **PR** #32614: (rallytime) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2016-04-15 19:27:47 UTC*
  - 05a41a13cd Merge pull request #32614 from rallytime/merge-2015.8
  - 046e401dd8 Merge branch '2015.5' into '2015.8'
    - 027b502335 Merge pull request #32561 from gtmanfred/user_passwords
      - 3db5e78d5d redact passwords and hashes from user.present updates
- **PR** #32616: (rallytime) Back-port #32547 to 2015.8 @ *2016-04-15 19:27:36 UTC*
  - **PR** #32547: (cro) Expand on the open-source vs open-core FAQ (refs: #32616)

  - ef17bde054 Merge pull request #32616 from rallytime/bp-32547
  - 4242bc7399 Language clarification.
  - 965e3bc1d1 Expand on the open-source vs open-core FAQ
  - **PR** #32604: (Talkless) Fix comments value in salt.states.pkgrepo example
  - **PR** #32558: (terminalmage) Revert PR #32480 and apply #32314 with fixes / documentation
  - **PR** #32480: (terminalmage) Clear VCS fsbackend and git_pillar locks on master start (refs: #32558)
  - **PR** #32314: (onorua) prevent eternal gitfs lock due to process crash (refs: #32480, #32558)
- **ISSUE** #32519: (Ch3LL) Minion restarting and erroring when cannot reach the master (refs: #32576)
  - **PR** #32576: (DmitryKuzmenko) Better log message on minion restart if master couldn't be reached.
- **ISSUE** #32517: (Ch3LL) Minion restarting and erroring when cannot reach the masters in multi-master failover (refs: #32555, #32556)
  - **PR** #32555: (DmitryKuzmenko) Don't return None from eval_master
- **PR** #32536: (rallytime) Back-port #31898 to 2015.8 @ *2016-04-13 18:49:05 UTC*
  - **PR** #31898: (afletch) Ensure rh_service not used on CloudLinux 7 (refs: #32536)
  - 27e91e40cc Merge pull request #32536 from rallytime/bp-31898
  - 60d80c4dee Ensure rh_service not used on CloudLinux 7
  - **PR** #32542: (twangboy) Fix binary search and replace
- **PR** #32539: (rallytime) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2016-04-13 15:10:08 UTC*
  - cce7de76b0 Merge pull request #32539 from rallytime/merge-2015.8
  - fbaeb165c9 Merge branch '2015.5' into merge-2015.8
  - 7307bcb88e Merge pull request #32538 from rallytime/bp-32528
    - 46a4e8a310 Remove merge conflict line
    - e0d947c707 Document "grains" setting in the minion configuration reference
- **ISSUE** #32493: (bberberov) dockerng.volume_present fails when no volumes already exist on the system (refs: #32531)
- **PR** #32531: (ticosax) [dockerng] Fix support of dockerng.volume_present when no volume is on present. @ *2016-04-13 14:42:13 UTC*
  - 1834bdefe3 Merge pull request #32531 from ticosax/support-no-volumes
  - 958b2ec749 Fix support of dockerng.volume_present when no volume is on present.
  - **PR** #32475: (ticosax) [dockerng] Enhance dockerng.wait() to control success on exit_code and on already stopped containers
  - **PR** #32436: (isbm) Bugfix: salt-key crashes if tries to generate keys to the directory w/o write access
  - **PR** #32515: (terminalmage) Turn on exc_info when logging failed minion startup
  - **PR** #32520: (terminalmage) Add ignore_epoch option to pkg.installed/removed/purged states
  - **PR** #32505: (isbm) Isbm zypper list products sles11 crash
  - **PR** #32480: (terminalmage) Clear VCS fsbackend and git_pillar locks on master start (refs: #32558)
  - **PR** #32314: (onorua) prevent eternal gitfs lock due to process crash (refs: #32480, #32558)

- **ISSUE** #32327: (joakimkarlsson) salt-minion fails to start on Windows (refs: #32491)
    - **PR** #32491: (twangboy) Use win32api to get Total System Memory
- **ISSUE** #31927: (afletch) pkg.installed compares version including package epoch (pkg.version problem?) (refs: #32487)
- **PR** #32487: (terminalmage) Add explanation of nonzero epoch requirement to pkg.installed state documentation @ *2016-04-11 20:48:57 UTC*
    - e335e313fe Merge pull request #32487 from terminalmage/epoch-documentation
    - e04cf879b6 Document new behavior of pkg.installed for yum/dnf packages with non-zero epoch
    - 61e9761224 Add explanation of nonzero epoch requirement to pkg.installed state documentation
- **PR** #32482: (rallytime) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2016-04-11 20:12:26 UTC*
    - e8de50ff37 Merge pull request #32482 from rallytime/merge-2015.8
    - 1b04f0ddec Merge branch '2015.5' into '2015.8'
    - 29333e533e Add documentation for some master/minion configs (#32454)
    - 100c6e1b25 Merge pull request #32458 from terminalmage/clarify-providers-docs
        - 500d3ebbaa Add link to provider override docs to all group providers
        - 83ca01f620 dd link to provider override docs to all shadow providers
        - c5fe38789d Add link to provider override docs to all user providers
        - 5c1c1dda59 Add link to provider override docs to all service providers
        - 736f2befc9 Add link to provider override docs to all package providers
        - f9306347cc Clarify the scope of the provider param in states.
        - af24c82ab0 Add documentation on virtual module provider overrides to the module docs
        - 0bc6c97a63 Improve docstrings
        - 1948920674 Add external ref to windows package manager docs
        - e7fa21438c Add new doc pages to toctree
        - f0de1236ec Move the tables of virtual modules to individual documentation pages
- **ISSUE** #30183: (jakehilton) Minion startup extremely delayed when first master in failover multi master setup is down (refs: #31364, #31382, #32143)
- **ISSUE** #29643: (matthayes) Can't get batch mode and –failhard to work as expected (refs: #31164)
- **ISSUE** #28569: (andrejohansson) Reactor alert on highstate fail (refs: #31164)
    - **PR** #32474: (DmitryKuzmenko) Backport 31164 and 31364
    - **PR** #32441: (cachedout) Backport 31164 31364 (refs: #32474)
    - **PR** #31364: (DmitryKuzmenko) Don't send REQ while another one is waiting for response. (refs: #32441, #32474)
    - **PR** #31164: (DmitryKuzmenko) Issues/29643 fix invalid retcode (refs: #32441, #32474)
- **ISSUE** #31738: (igorwidlinski) salt –show-jid does not show job id when run in batch mode (refs: #32450)
- **PR** #32450: (cachedout) Pass parser options into batch mode (refs: #32996) @ *2016-04-08 23:03:49 UTC*
    - 7bf44aea72 Merge pull request #32450 from cachedout/issue_31738

  - 74d0fa06b4 Pass parser options into batch mode
- **ISSUE** #28706: (kkaig) user.present:groups vs group.present:members (refs: #30824, #32448)
  - **PR** #32448: (rallytime) Back-port #30824 to 2015.8
  - **PR** #30824: (alxf) Issue #28706: Fix state user.present behavior. (refs: #32448)
- **ISSUE** #31851: (rhansen) error using module.run -> saltutil.runner -> state.orchestrate: "The following arguments are missing: _fun" (refs: #32445)
  - **PR** #32445: (rallytime) Argument name in docs should match actual arg name
  - **PR** #26676: (rallytime) Back-port #26648 to 2015.5 (refs: #32445)
  - **PR** #26648: (whiteinge) Free 'fun' from the function signature namespace (refs: #26676)
- **ISSUE** #32033: (timcharper) SaltStack *modules.dockerng _compare* does not handle docker implicit Domainname properly (issue when using network_mode: host) (refs: #32116, #32432)
- **PR** #32432: (ticosax) [dockerng] Fix Domainname introspection @ *2016-04-08 16:12:19 UTC*
  - a36f9499fc Merge pull request #32432 from ticosax/fix-domainname-introspection
  - 505b5b0168 Fix Domainname introspection
- **PR** #32427: (rallytime) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2016-04-08 15:39:13 UTC*
  - def911974c Merge pull request #32427 from rallytime/merge-2015.8
  - 9531ea6ef5 Merge branch '2015.5' into '2015.8'
  - 0809126d8e Merge #32293 with test fixes (#32418)
  - bbd8260a42 Ignore Raspbian in service.py __virtual__ (#32421)
  - 690addf0b4 FreeBSD supports packages in format java/openjdk7 so the prior commit broke that functionality. Check freebsd/pkg#1409 for more info.
  - a36866d7db Merge pull request #32399 from amontalban/2015.5
    - e1ffbd615a Fixes saltstack/salt#28262 for 2015.5 branch
  - 3f03c5fcf9 Merge pull request #32374 from cachedout/issue_32066
    - 62389d1d1a Update proxmox documentation
  - 8578089beb Merge pull request #32339 from Ch3LL/fix_doc_multi-master
    - 2774da288d remove reference to master_alive_check
- **ISSUE** #32311: (rkgrunt) glusterfs module incorrectly indexes into name of bricks (refs: #32312)
- **PR** #32423: (justinta) Update glusterfs_test to be inline with #32312 @ *2016-04-07 21:53:03 UTC*
  - **PR** #32312: (rkgrunt) Fixed glusterfs module (refs: #32423)
  - 5bc8c326ce Merge pull request #32423 from jtand/glusterfs_test_fix
  - 6f98bd50eb Update glusterfs_test to be inline with #32312
- **ISSUE** #31632: (zieba88) salt-cloud map parallel provisioning -P option failed on 2015.8.5 (refs: #32425)
- **PR** #32425: (cachedout) Fix salt-cloud paralell provisioning @ *2016-04-07 21:52:06 UTC*
  - c07e02bacb Merge pull request #32425 from cachedout/issue_31632
  - 127c0829ee Fix salt-cloud paralell provisioning
- **PR** #32323: (mcalmer) fix sorting by latest version when called with an attribute @ *2016-04-07 06:24:35 UTC*

  - 2cc054bbc0 Merge pull request #32323 from mcalmer/fix-ensure-installed-latest-with-attributes
  - cb1f30ee10 fix sorting by latest version when called with an attribute
- **ISSUE** saltstack/salt#28262: (palica) FreeBSD pkgng provider raising error for minion (refs: #32376)
- **ISSUE** #28262: (palica) FreeBSD pkgng provider raising error for minion (refs: #32376, #32399)
- **PR** #32376: (amontalban) Fixes saltstack/salt#28262 (refs: #32399) @ *2016-04-06 20:30:10 UTC*
  - 802580ee1a Merge pull request #32376 from amontalban/2015.8
  - 823d0c362b Fixes saltstack/salt#28262
- **ISSUE** #32375: (truescotw) jinja template copying file but not replacing tags (refs: #32393)
- **PR** #32393: (jfindlay) modules.win_timezone: don't list all zones in debug log @ *2016-04-06 18:10:43 UTC*
  - ad77d76cad Merge pull request #32393 from jfindlay/win_zone
  - c01c1b9da2 modules.win_timezone: don't list all zones in debug log
- **PR** #32372: (rallytime) Back-port #32358 to 2015.8 @ *2016-04-06 16:35:05 UTC*
  - **PR** #32358: (arthurlogilab) outputter virt_list does not exist anymore (refs: #32372)
  - 76ae95863d Merge pull request #32372 from rallytime/bp-32358
  - 95e0fe7744 outputter virt_list does not exist anymore
- **PR** #32392: (multani) Fix documentation on boto_asg and boto_elb modules and states @ *2016-04-06 16:34:36 UTC*
  - c612baa119 Merge pull request #32392 from multani/2015.8
  - 77c4772752 Fix documentation on boto_asg and boto_elb modules and states
- **ISSUE** #32201: (boltronics) salt-minion memory leak waiting on master to accept key (refs: #32373)
- **PR** #32373: (cachedout) Resolve memory leak in authentication @ *2016-04-06 15:19:55 UTC*
  - b706d3aa4d Merge pull request #32373 from cachedout/issue_32201
  - d9e4a0f372 Resolve memory leak in authentication
- **PR** #32126: (cro) Add a couple CLI examples for the highstate outputter. @ *2016-04-05 17:23:29 UTC*
  - 097aa7ccfc Merge pull request #32126 from cro/outputter_terse_docs
  - dafe279e60 Lint
  - abc2de0119 More clarification.
  - 85221e515b Expand docs for highstate outputter. Add CLI examples for when 'state_output: filter' is set.
- **PR** #32353: (mcalmer) Prevent metadata download when listing installed products @ *2016-04-05 17:02:15 UTC*
  - eab3b99be2 Merge pull request #32353 from mcalmer/prevent-refresh-on-list-installed-products
  - e32212ad53 Prevent metadata download when listing installed products
- **ISSUE** #32255: (jakosky) Salt-minion 2015.8.8 should display helpful error when regular file /var/log/salt/minion exists but a directory is expected. (refs: #32321)
- **PR** #32321: (abednarik) Better message when minion fail to start @ *2016-04-05 16:28:06 UTC*
  - 64abec94e7 Merge pull request #32321 from abednarik/minion_start_fail_log
  - 4c72adc03a Better message when minion fail to start

- **ISSUE** #30147: (anandnevase) salt.cloud.CloudClient method create() not working for VMware driver (refs: #32344)
- **PR** #32345: (nmadhok) [2015.8] Check if profile key exists in vm_ dict @ *2016-04-05 16:16:36 UTC*
  - **PR** #32344: (nmadhok) Check if profile key exists in vm_ dict (refs: #32345)
  - 59aca733ea Merge pull request #32345 from nmadhok/patch-4
  - 42d7a54240 Check if profile key exists in vm_ dict
- **PR** #32343: (Ferbla) Fixed win_wua example documentation @ *2016-04-05 16:14:37 UTC*
  - bb033c238d Merge pull request #32343 from Ferbla/2015.8
  - e2f0f16564 Fixed win_wua example documentation
- **ISSUE** #32354: (elsmorian) Incorrect capitalisation when telling users to change hash_type to SHA256 (refs: #32360)
- **PR** #32360: (rallytime) Make sure hash_type is lowercase in master/minion config files @ *2016-04-05 16:10:46 UTC*
  - 3219a8d176 Merge pull request #32360 from rallytime/fix-32354
  - 8b47c205df Make sure hash_type is lowercase in master/minion config files
- **PR** #32361: (cro) SDB is no longer experimental @ *2016-04-05 16:10:23 UTC*
  - fb530256f6 Merge pull request #32361 from cro/remove_sdb_exp_flag
  - 3bbe284d89 Remove 'experimental' warning from SDB docs.
- **PR** #32336: (rallytime) Back-port #28639 to 2015.8 @ *2016-04-04 20:53:11 UTC*
  - **PR** #28639: (RuriRyan) Fixed handling of the disabled option for yumpkg (refs: #32336)
  - e1ef4a9d66 Merge pull request #32336 from rallytime/bp-28639
  - 0829143dd1 Fixed handling of the disabled option for yumpkg
- **ISSUE** #32305: (Ch3LL) Receiving NoResponse Errors when running commands that take a longer time (refs: #32332)
- **PR** #32332: (rallytime) Don't unsubscribe from open events on the CLI too early on long-running commands @ *2016-04-04 20:39:39 UTC*
  - **PR** #32145: (paclat) fixes 29817 (refs: #32332)
  - 6ee5a9729c Merge pull request #32332 from rallytime/fix-32305
  - 8dc1161c8a Don't unsubscribe from open events on the CLI too early on long-running commands
- **PR** #32333: (rallytime) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2016-04-04 20:06:02 UTC*
  - 22b296d2fd Merge pull request #32333 from rallytime/merge-2015.8
  - d7b4b8b081 Merge branch '2015.5' into '2015.8'
  - fbdc47cc55 Merge pull request #32284 from rallytime/config-audit
    - 0491513204 Don't be so explicit. Just use string_types.
    - 083c477fd3 Use six.string_types in config default tuples
    - 7e642b8381 Audit config.py default types and values - first sweep
  - 0a6d44e57b Merge pull request #32302 from terminalmage/fix-missing-release
    - 413c371ccd Properly support packages with blank "Release" param in pkg.latest_version

- **ISSUE** #32246: (danlsgiga) IMPORTANT: Do not use md5 hashing algorithm! Please set "hash_type" to SHA256 in Salt Minion config! (refs: #32289)
- **PR** #32289: (rallytime) New salt-cloud instances should not use old hash_type default. @ *2016-04-04 17:52:09 UTC*
  - **PR** #31162: (isbm) Remove MD5 digest from everywhere and default to SHA256 (refs: #32289)
  - 28cc054244 Merge pull request #32289 from rallytime/fix-32246
  - 66acc00c71 New salt-cloud instances should not use old hash_type default.
- **ISSUE** #14277: (Sacro) Chocolatey.version doesn't tell you anything informative. (refs: #32291)
- **PR** #32291: (twangboy) Fix bad output for chocolatey.version (fixes #14277) @ *2016-04-04 17:50:54 UTC*
  - 5fb90a1040 Merge pull request #32291 from twangboy/fix_14277
  - 53f6a28297 Fix problem with return on installed packages
  - f5bd004ab0 Fix chocolatey.version function
- **ISSUE** #32183: (llamallama) Salt Cloud 2015.8.8 not installing salt minions on new nodes (refs: #32295)
- **PR** #32295: (rallytime) Test the contents of 'deploy_scripts_search_path' in salt.config.cloud_config @ *2016-04-04 17:38:47 UTC*
  - edbab99164 Merge pull request #32295 from rallytime/test-cloud-deploy-dir
  - 4037476f40 Patch call to os.path.isdir so we know both search paths are in tuple
  - 49a4eec051 Test the contents of 'deploy_scripts_search_path' in salt.config.cloud_config
- **ISSUE** #23617: (porterjamesj) file.managed with proxy broken in 2015.5 (refs: #32315)
- **PR** #32315: (ahus1) fixing file.managed with requests lib @ *2016-04-04 17:20:11 UTC*
  - 4389680bc5 Merge pull request #32315 from ahus1/fix_file_managed_http_requests
  - a867d23383 ensure streaming mode (use for example by file.managed) will works for requests backend
- **ISSUE** saltstack/salt-bootstrap#782: (ninjada) Bootstrap and Links & Documentation still broken due to fedoraproject redirect to fedorainfracloud.org (refs: #32316)
- **ISSUE** saltstack/salt-bootstrap#742: (dennisfoconnor) Non-Development Script Broken on Amazon Linux (refs: #32316)
- **ISSUE** saltstack/salt-bootstrap#695: (mtippett) Install Failures With Raspbian Jessie (refs: #32316)
- **PR** #32316: (vutny) Update Salt Bootstrap tutorial @ *2016-04-04 17:18:12 UTC*
  - 9065201761 Merge pull request #32316 from vutny/update-bootstrap-tutorial
  - b9698f015d Update Salt Bootstrap tutorial
- **PR** #32325: (bdrung) Re-add shebang to ssh-id-wrapper shell script @ *2016-04-04 17:08:41 UTC*
  - 352f3c01d1 Merge pull request #32325 from bdrung/fix-shebang
  - ffe585f078 Re-add shebang to ssh-id-wrapper shell script
- **PR** #32326: (bdrung) Fix typos @ *2016-04-04 16:41:41 UTC*
  - f16e332b3a Merge pull request #32326 from bdrung/fix-typos
  - a7db152333 Fix typo dont -> don't
  - d4c037301b Fix typo missmatch -> mismatch
  - 70dba70ff0 Fix typo additonal -> addition

  - 68c60903aa Fix typo mutliple -> multiple
  - 0f2c779b90 Fix typo fucntion -> function
  - 0c9e4c8c80 Fix typo avilable -> available
  - 920abe2ec7 Fix typo formated -> formatted
  - e56dd4bb23 Fix typo ommitted -> omitted
  - f99e6f1f13 Fix typo ouptut -> output
  - d3804094f2 Fix typo wether -> whether
  - 538fb6fae2 Fix typo perfomed -> performed
  - db7af998ee Fix typo santized -> sanitized
  - d7af01da2b Fix typo coresponding -> corresponding
  - 301e78b5be Fix typo vaules -> values
  - 8cada9573f Fix typos of retrieve
  - b484d6f9c9 Fix typo directorys -> directories
- **PR** #32300: (twangboy) Add documentation to disable winrepo/winrepo_ng @ *2016-04-01 21:23:09 UTC*
  - 664043d7e7 Merge pull request #32300 from twangboy/fix_28767
  - c971a3b054 Add documentation for disabled the winrepos
- **ISSUE** #18429: (somenick) Pillars passed from command-line override pillar subtrees instead of merging (refs: #32288)
- **PR** #32288: (terminalmage) use dictupdate.merge instead of dict.update to merge CLI pillar overrides @ *2016-04-01 16:30:30 UTC*
  - 42a25f6b9d Merge pull request #32288 from terminalmage/issue18429
  - db31732137 use dictupdate.merge instead of dict.update to merge CLI pillar overrides
- **PR** #32243: (isbm) Ensure latest pkg.info_installed ensure latest @ *2016-03-31 16:09:59 UTC*
  - 3e374e7ec6 Merge pull request #32243 from isbm/isbm-zypper-list-installed-ensure-latest
  - fba3d509ac Fix the documentation
  - 73ad8a2bfc Fix lint
  - f07c7ea792 Add lowpkg tests for version comparison
  - afdf451d87 Remove tests from the zypper_test that belongs to rpm_test
  - 3706a21c29 Fix condition from returning None on 0
  - 0a68ebff16 Remove suse/redhat checks, refactor code.
  - 30c8f7216b Move "string to EVR" function to the utilities
  - fb014a40b0 Sort installed pkgs data by version_cmp
  - b57e439d57 Merge yumpkg's and zypper's version_cmp for a common use
  - ebd13a283c Remove version_cmp from Zypper module and use just lowpkg alias
  - b46d5b526a Remove version_cmp from the yumpkg and use just a lowpkg alias
  - f4d9881e61 Force-sort the RPM output to ensure latest version of the multi-package on top of the list.

- **ISSUE** #32261: (arthurlogilab) dockerng : AttributeError: 'module' object has no attribute 'version_info' (refs: #32262, #32268)
- **PR** #32268: (ticosax) [dockerng] Improve detection for older versions of docker-py @ *2016-03-31 14:51:46 UTC*
  - **PR** #32262: (arthurlogilab) Catch Attribute Error when docker.version_info doesn't exist (refs: #32268)
  - 88fa3c5f71 Merge pull request #32268 from ticosax/handle-dockerpy-old
  - 05116aaa40 Improve detection for older versions of docker-py
- **PR** #32258: (jacobhammons) Replaces incorrect reference to *master_alive_check* @ *2016-03-31 14:41:09 UTC*
  - a491897a3b Merge pull request #32258 from jacobhammons/alive-interval-docs
  - ff8ca5ac2e Replaces incorrect reference to *master_alive_check* with *master_alive_interval* in docs
- **PR** #32254: (twangboy) Fix Display Name with spaces in win_servermanager @ *2016-03-31 14:38:22 UTC*
  - 8c68d8ac41 Merge pull request #32254 from twangboy/fix_31334
  - e5f02c52be Fix a pylint error
  - 5ca4ad6675 Fix unit tests for state
  - 12d530f8f0 Fix win_servermanager state
  - b26cb76abb Fix unit tests
  - 1d5bcee390 Fix 31344
- **PR** #32248: (rallytime) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2016-03-30 21:10:01 UTC*
  - 0f5e67de5d Merge pull request #32248 from rallytime/merge-2015.8
  - d743f8cc4e Merge branch '2015.5' into '2015.8'
    - 5d08db7c92 Merge pull request #32162 from terminalmage/issue31963
      - 5c1bdb812c Fix pkgrepo integration test
      - e7fb3095ce Properly handle yum/zypper repositories in pkgrepo.managed
      - add2111fec Use six.iteritems instead of dict.items
      - 6c21881c38 Docstring tweaks
      - ecbb78b649 Remove useless function
      - 06f3309552 Normalize variable naming to match other functions
      - 690537ca8b Look for apt-add-repository in PATH instead of assuming it's there
      - 709d80bb1b aptpkg: Accept **kwargs instead of a dict for pkg.expand_repo_def
    - 4fcdaab428 Merge pull request #32223 from twangboy/fix_31976
      - b7fcae97ce Create minion.d directory, fixes #31976
    - 3309ff6a29 Merge pull request #32218 from cachedout/issue_31501
      - 6795d6aef0 Only display error when tty is True in salt-ssh
    - 6e0cb22c96 Merge pull request #32196 from jtand/cherrypy_pam_test_lint_fix
      - bd3942e0fd Fixed pylint error in app_pam_test.py
- **ISSUE** #32169: (sknutsonsf) CommandNotFoundError: update-rc.d during service.enabled on Centos 7 (refs: #32230)

- **PR** #32230: (terminalmage) systemd.py: Support both update-rc.d and chkconfig as managers of sysv services @ *2016-03-30 21:09:43 UTC*
  - 6216c37885 Merge pull request #32230 from terminalmage/issue32169
  - 45af3e902a systemd.py: Support both update-rc.d and chkconfig as managers of sysv services
- **PR** #32249: (jacobhammons) Fixes windows download paths to account for patch @ *2016-03-30 20:26:53 UTC*
  - bde2a1fc98 Merge pull request #32249 from jacobhammons/dot8
  - 50d1df2482 Fixes windows download paths to account for patch
- **PR** #32221: (dmurphy18) Fix version check, fix extracting Major and Minor versions from __ver… @ *2016-03-30 14:50:31 UTC*
  - 1d9321d043 Merge pull request #32221 from dmurphy18/fix_version_check
  - 96cf024e63 Fix version check, fix extracting Major and Minor versions from __version__
- **ISSUE** #32031: (travispaul) Unable to manage Windows services that contain a space in the service name (refs: #32227)
- **PR** #32227: (twangboy) Remove list2cmdline usage from win_service.py @ *2016-03-30 14:43:17 UTC*
  - 22bd1e6b29 Merge pull request #32227 from twangboy/fix_32031
  - 58772b036d Remove list2cmdline usage
- **PR** #32239: (anlutro) Add state file name to warning log line @ *2016-03-30 14:37:54 UTC*
  - 7fce438b67 Merge pull request #32239 from alprs/fix-file_log_warning
  - 72adae3702 add state file name to log line
- **ISSUE** #31365: (cwicklein) osrelease_info broken for CentOS 7 (refs: #32215)
- **PR** #32215: (DmitryKuzmenko) rhel oscodename @ *2016-03-29 19:14:50 UTC*
  - 3c3028f347 Merge pull request #32215 from DSRCompany/issues/rhel_oscodename
  - dc2a3b81ac Ignore lsb codename from os-release for newest RHEL
- **PR** #32217: (jacobhammons) 2015.8.8.2 release notes @ *2016-03-29 17:53:22 UTC*
  - bf59f06733 Merge pull request #32217 from jacobhammons/dot8
  - 596444e2b4 2015.8.8.2 release notes Adds banner notifiying user when they are viewing release notes for an old release
- **ISSUE** #31844: (Talkless) slspath is not documented (refs: #32197)
- **PR** #32212: (rallytime) Back-port #32197 to 2015.8 @ *2016-03-29 15:50:58 UTC*
  - **PR** #32197: (twellspring) documentation fix issue 31844 (refs: #32212)
  - ab8b70d985 Merge pull request #32212 from rallytime/bp-32197
  - 5fdd81ace9 documentation fix issue 31844
- **ISSUE** #31931: (gravyboat) Ordering States documentation should note top.sls adheres to this rule (refs: #32193)
- **PR** #32211: (rallytime) Back-port #32210 to 2015.8 @ *2016-03-29 15:50:42 UTC*
  - **PR** #32210: (rallytime) Merge #32193 with pylint fix (refs: #32211)
  - **PR** #32193: (twellspring) Documentation fix 31931 (refs: #32210, #32211)
  - 200d82cc3e Merge pull request #32211 from rallytime/bp-32210

  - 7b9c05487c Whitespace fix.
  - abd432746c documentation-fix-31931
  - 79086f8f04 service.py documentation update for 32084
- **ISSUE** #32084: (guettli) Docs: Please provide a link from "service.running" to "watch" (refs: #32192)
- **PR** #32209: (rallytime) Back-port #32208 to 2015.8 @ *2016-03-29 15:50:27 UTC*
  - **PR** #32208: (rallytime) Merge #32192 with pylint fix (refs: #32209)
  - **PR** #32192: (twellspring) service.py documentation update for 32084 (refs: #32208, #32209)
  - 32da8d4c57 Merge pull request #32209 from rallytime/bp-32208
  - 777a2c4e83 Whitespace fix.
  - e3db0640ec service.py documentation update for 32084
- **ISSUE** #31595: (dverbeek84) dockerng ports specified in Dockerfile must be in sls file otherwise salt gives an error (refs: #32204)
- **PR** #32204: (ticosax) [dockerng] Consider labels carried by the image when comparing user defined labels. @ *2016-03-29 14:39:22 UTC*
  - 7154104591 Merge pull request #32204 from ticosax/label-from-image
  - c989ae5a7e Merge user defined labels with one carried by the image
- **PR** #32186: (rallytime) Add some "best practices" information to test documentation @ *2016-03-29 00:22:48 UTC*
  - 5877a19f59 Merge pull request #32186 from rallytime/testing-docs
  - 40d09c822e Add some "best practices" information to test documentation
- **PR** #32176: (rallytime) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2016-03-28 23:16:09 UTC*
  - b44adffc12 Merge pull request #32176 from rallytime/merge-2015.8
  - e8658697a6 Pylint fix for integration import
  - 527bc3e491 Pylint fix
  - e9abd2d420 Merge branch '2015.5' into '2015.8'
  - 6b8b8b51c0 Merge pull request #32154 from Ch3LL/ch3ll_pam_2015.5
    - ba605b0128 fix more pylint and add ability to close cherrypy engine
    - 2d4dc4da05 add teardown call
    - d115878714 fix pylint error
    - 4c1ab082b6 add pam salt-api tests
  - 230443be6c Merge pull request #32170 from gtmanfred/lxc_cloud_name
    - eb7d82e7be add name for lxc for use with cloud cache
  - 32b0421a34 Merge pull request #32164 from terminalmage/issue31731-2015.5
    - 18439c4f89 Make __virtual__ for rhservice.py more robust (2015.5 branch)
  - 6212e9aa56 Merge pull request #32141 from paclat/issue_32108
    - 72c5d12d43 fixes 32108
- **ISSUE** #27605: (jmcook1) nacl module documentation/possible bug (refs: #32163)

- **PR** #32163: (rallytime) Update nacl.config docs to use key value instead of 'None' @ *2016-03-28 14:46:40 UTC*
  - 1afb048801 Merge pull request #32163 from rallytime/fix-27605
  - e2d09f57dc Update nacl.config docs to use key value instead of 'None'
- **PR** #32166: (vutny) *salt.states.file*: correct examples with multiline YAML string @ *2016-03-28 14:45:32 UTC*
  - c08ba3f8a9 Merge pull request #32166 from vutny/fix-multiline-yaml-string-example
  - 34aaea93b4 Another indentation fix in *salt.states.alternatives*
  - 85d0576583 *salt.states.file*: correct examples with multiline YAML string
- **PR** #32168: (rallytime) Lint 2015.8 @ *2016-03-27 18:26:50 UTC*
  - f2e986cf65 Merge pull request #32168 from rallytime/lint-2015.8
  - ba6b19d72c Lint 2015.8
- **ISSUE** #31731: (sjorge) rh_service references osrelease before it is available, also does not return bool (refs: #32165)
- **PR** #32165: (terminalmage) Make __virtual__ for rhservice.py more robust (refs: #32164) @ *2016-03-27 18:21:16 UTC*
  - **PR** #32164: (terminalmage) Make __virtual__ for rhservice.py more robust (2015.5 branch) (refs: #32165)
  - ae472617af Merge pull request #32165 from terminalmage/issue31731
  - 559eb7da52 Make __virtual__ for rhservice.py more robust
- **ISSUE** #31944: (Inveracity) traceback in _determine_beacon_config(…) in beacon/__init__.py line 105 (refs: #32160)
- **PR** #32160: (cachedout) Fix beacon tutorial docs @ *2016-03-25 22:32:51 UTC*
  - 63c8bf3542 Merge pull request #32160 from cachedout/issue_31944
  - 104ada5b6f Fix beacon tutorial docs
- **PR** #32145: (paclat) fixes 29817 (refs: #32332) @ *2016-03-25 16:55:47 UTC*
  - bff94a5160 Merge pull request #32145 from paclat/issue_29817
  - 5d970ca031 fixes 29817
- **PR** #32133: (basepi) Pass eauth user/groups through salt-api to destination functions @ *2016-03-25 16:49:46 UTC*
  - 245249d347 Merge pull request #32133 from basepi/api_user_passthrough
  - 41ba309839 Change the kwarg names to be more specific
  - 40f7e596d8 Pass eauth user/groups through salt-api to destination functions
- **PR** #32127: (rallytime) Add runners to __salt__ docs @ *2016-03-25 15:54:02 UTC*
  - a09aa18036 Merge pull request #32127 from rallytime/dunder-docs
  - 482690ef33 Add note to docs about __salt__ referencing runner modules
  - a11d2e413a Add runners to __salt__ docs
- **ISSUE** #30183: (jakehilton) Minion startup extremely delayed when first master in failover multi master setup is down (refs: #31364, #31382, #32143)
- **PR** #32143: (DmitryKuzmenko) Set auth retry count to 0 if multimaster mode is failover. @ *2016-03-25 15:23:09 UTC*

  - **PR** #31382: (DmitryKuzmenko) Set auth retry count to 0 if multimaster mode is failover (refs: #32143)
  - cc224b877a Merge pull request #32143 from DSRCompany/issues/30183_failover_fix
  - 93d34a2573 Set auth retry count to 0 if multimaster mode is failover.
- **PR** #32134: (rallytime) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2016-03-25 15:22:08 UTC*
  - 0679a61871 Merge pull request #32134 from rallytime/merge-2015.8
  - 6886681410 Fix test failures
  - 7554d0f42d Merge branch '2015.5' into '2015.8'
    - bdd7ea89d5 Merge pull request #32129 from terminalmage/issue32044
      - 34ca1ea12e Change type check errors to debug loglevel
      - 5462081488 Support multiple valid option types when performing type checks
    - c42014eb54 Merge pull request #32056 from bstevenson/fix-list_absent
      - 1500aae027 set deleted value to list
      - 1dc8f5f289 unit test update
      - 39adf86fec Fixed negation logic
      - be9388173b Removed has_key in lieu of in
      - e48593ed81 Comments and Changes output fixes
      - b98f5517de Updated to conform to proper ret values
      - d18b4be80b remove whitespace end of line 186:q
      - d2b89c85ad fix formating
      - 103cee9e29 cleaned up formating
      - 7a4d7f0bff added whitespace
      - 8ea5b545b0 Loop through list values in list_absent
    - 848ce5647f Merge pull request #32096 from rallytime/bp-32065
      - 36a9d6a374 Fix an issue with the minion targeting example
    - 9b332d48b9 Merge pull request #32104 from jacobhammons/dot10
      - b9fc882a1e One additional known issue for 2015.5.10 release notes
    - ff51d548e1 Merge pull request #32100 from jacobhammons/dot10
      - 544a1661ce 2015.5.10 release docs
    - 72a20f9799 Merge pull request #32038 from terminalmage/issue32037
      - 8b2d983324 Add reference to state tutorial to state.apply docstring
      - 9b4fe8443e Move highstate usage details to top of state.apply docstring
      - 74ee8c54bc Clarify prior role of state.highstate in states tutorial
      - 1b97e4a3df Improve state module docs, replace references to state.highstate/state.sls with state.apply
- **ISSUE** #26129: (GreatSnoopy) salt yumpkg implementation painfully slow in some circumstances (refs: #32091)
- **PR** #32091: (clarkperkins) Fixed the regression in 410da78 @ *2016-03-25 14:53:08 UTC*

  - ad924226ca Merge pull request #32091 from clarkperkins/bugfix/yumpkg-repoquery
  - d2119ea608 Added comment so this issue doesn't regress again
  - 1455fab9e3 Fixed the regression in 410da78
- **ISSUE** #32044: (ScoreUnder) Multiple masters throwing warnings? "Key master with value [...] has an invalid type of list, a str is required for this value" (refs: #32129)
- **PR** #32135: (rallytime) [2015.8] Support multiple valid option types when performing type checks @ *2016-03-24 22:42:28 UTC*
  - **PR** #32129: (terminalmage) Support multiple valid option types when performing type checks (refs: #32135, #32284)
  - b84908d51f Merge pull request #32135 from rallytime/32129-to-2915.8
  - 7d43bdd721 Change type check errors to debug loglevel
  - ed5abf4381 Support multiple valid option types when performing type checks
- **PR** #31760: (sakateka) SMinion need wait future from eval_master @ *2016-03-24 22:08:56 UTC*
  - b23a08f3f4 Merge pull request #31760 from sakateka/fix_master_switch
  - 3d7874029a Run self.eval_master in self.io_loop.run_sync
  - 3b4425652b SMinion need wait future from eval_master
- **PR** #32106: (jfindlay) update suse master service patch @ *2016-03-24 21:34:01 UTC*
  - 5efe37ddc8 Merge pull request #32106 from jfindlay/suse_patch
  - 8de84b4251 update suse master service patch
- **PR** #32130: (jacobhammons) Added known issues 32004 and 32044 to 2015.8.8 release notes @ *2016-03-24 19:59:41 UTC*
  - 939c1b17d5 Merge pull request #32130 from jacobhammons/dot8
  - 21eee08842 Added known issues 32004 and 32044 to 2015.8.8 release notes
- **PR** #32105: (clarkperkins) Fixed invalid deploy_scripts_search_path @ *2016-03-24 17:36:27 UTC*
  - 2d8abf4717 Merge pull request #32105 from clarkperkins/bugfix/invalid-deploy-script-path
  - 5a9f4e947e Fixed invalid deploy_scripts_search_path
- **ISSUE** #32114: (tomlaredo) Wrong validation type for file_ignore_glob key (refs: #32117)
- **PR** #32117: (tomlaredo) Fixed validation type for file_ignore_glob @ *2016-03-24 17:28:22 UTC*
  - fe4112d7f9 Merge pull request #32117 from rodacom/fix_32114
  - c6f83ba00b Fixed validation type for file_ignore_glob Fixes #32114
- **PR** #32113: (sakateka) Fix log message for AsyncAuth initialization @ *2016-03-24 17:27:04 UTC*
  - 93d86d249c Merge pull request #32113 from sakateka/correct_log_message
  - 71148d77ab Fix log message for AsyncAuth initialization
- **ISSUE** #32033: (timcharper) SaltStack *modules.dockerng _compare* does not handle docker implicit Domainname properly (issue when using network_mode: host) (refs: #32116, #32432)
- **PR** #32116: (ticosax) Obtain default value of *memory_swap* from the container. @ *2016-03-24 15:56:54 UTC*
  - 294177f428 Merge pull request #32116 from ticosax/memory_swap-default-from-container
  - fe439db4d3 Obtain default value of *memory_swap* from the container.

- **PR** #32098: (rallytime) Back-port #32083 to 2015.8 @ *2016-03-23 21:49:01 UTC*
  - **PR** #32083: (guettli) "Fire Event Notifications" moved down (refs: #32098)
  - d5bb8f6372 Merge pull request #32098 from rallytime/bp-32083
  - 4a3a6629ce "Fire Event Notifications" moved down
- **PR** #32099: (jacobhammons) 2015.8.8 release docs @ *2016-03-23 20:02:40 UTC*
  - e45107ce96 Merge pull request #32099 from jacobhammons/dot8
  - 8ec5d989ad 2015.8.8 release docs
- **PR** #32088: (rallytime) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2016-03-23 17:52:37 UTC*
  - 9e11f3aac5 Merge pull request #32088 from rallytime/merge-2015.8
  - 59c3b7e82e Merge branch '2015.5' into '2015.8'
  - 908a7bf5cd Merge pull request #32051 from terminalmage/fix-state-apply-output
    - 7d7cb45565 Fix outputter for state.apply
  - 0e66f678d4 Merge pull request #32002 from abednarik/pkg_manjaron_issue31788
    - 1b052d0a66 Added Manajro Linux to virtual. List extended with ManajaroLinux in order su load pacman module.
  - ba5bf62c1a Merge pull request #31957 from rallytime/merge-2015.5
    - 1b6ec5d445 Merge branch '2014.7' into '2015.5'
      - ba73deee46 Merge pull request #31929 from twangboy/fix_build_script
      - 2c5599d2bc Backport build script from 2015.8
      - ce74991dd0 Fix nsi script to work with new build process
  - a52e3ad7a1 Merge pull request #31972 from terminalmage/zh-584
    - 1e5639e495 Make lack of python-ldap module more explicit when LDAP eauth is enabled
- **PR** #32074: (Xiami2012) Fix code for proto args in modules.iptables @ *2016-03-23 16:37:58 UTC*
  - bc9a899bc8 Merge pull request #32074 from Xiami2012/fix_iptables
  - aae3af7e49 Fix code for proto args in modules.iptables
- **PR** #32053: (basepi) [2015.8] Fix rabbitmq_user.present tag handling @ *2016-03-22 20:33:51 UTC*
  - 3e08dd0a93 Merge pull request #32053 from basepi/fix_rabbitmq
  - 95c08f55e9 Tear out useless unit test
  - bed048e1e7 Remove leftover arg (lint)
  - 08868cb32a Fix tag handling code for rabbitmq_user.present
  - 3b6d25b4e9 Remove leading whitespace on tags
- **ISSUE** #32004: (sjorge) win_dacl module stacktrace: NameError: name 'ntsecuritycon' is not defined (refs: #32023)
- **PR** #32023: (sbreidba) Move constant declaration into member variable to avoid issues when m… @ *2016-03-21 20:18:23 UTC*
  - 553ecaca25 Merge pull request #32023 from sbreidba/bugfix_32004

  - 711a0a9844 Move constant declaration into member variable to avoid issues when modules can't be loaded.
- **PR** #32026: (techhat) Don't require the decode_out file to already exist @ *2016-03-21 20:17:05 UTC*
  - 65c634d197 Merge pull request #32026 from techhat/decodeout
  - f27da41b71 Don't require the decode_out file to already exist
- **PR** #32019: (rallytime) Back-port #32012 to 2015.8 @ *2016-03-21 15:54:31 UTC*
  - **PR** #32012: (jfray) There were two identical blocks concerning Windows Deploy Timeouts. This (refs: #32019)
  - 1d4246bfd7 Merge pull request #32019 from rallytime/bp-32012
  - 26eee1505f There were two identical blocks concerning Windows Deploy Timeouts. This pull request removes the extra block of text.
- **ISSUE** #32013: (timcharper) SaltStack dockerng.running state ports configuration responding to Docker's injection of UDP params (refs: #32015)
- **PR** #32015: (ticosax) [dockerng] Fix ports exposition when protocol is passed. @ *2016-03-21 15:22:19 UTC*
  - d117db3efb Merge pull request #32015 from ticosax/fix-port-comparison-udp
  - e511864a55 Fix ports exposition when protocol is passed.
- **PR** #31999: (jacobhammons) Fixes a doc build exception caused by missing mocks for modules.win_dacl @ *2016-03-19 15:49:40 UTC*
  - c72ab6a073 Merge pull request #31999 from jacobhammons/mock-modules2
  - 31bb573abc Fixes a doc build exception caused by missing mocks for modules.win_dacl
- **PR** #31992: (notpeter) salt-cloud: add D2 and G2 EC2 instance types @ *2016-03-18 21:37:21 UTC*
  - 398ab909f0 Merge pull request #31992 from notpeter/2015.8
  - e3854c8569 D2 and G2 EC2 instance types.
- **PR** #31981: (lloydoliver) include rotational disks in grains under linux @ *2016-03-18 15:54:00 UTC*
  - ad8ada7eef Merge pull request #31981 from lloydoliver/linux-disk-grain-fix
  - 9c44604438 include rotational disks in grains under linux
- **PR** #31970: (twangboy) Add apply_template_on_contents for windows @ *2016-03-18 15:37:29 UTC*
  - 9be508e8f0 Merge pull request #31970 from twangboy/fix_win_file
  - dfeae191c1 Add apply_template_on_contents for windows
- **PR** #31960: (aletourneau) fixed ec2 get_console_output @ *2016-03-18 15:13:48 UTC*
  - 810c6dbcbe Merge pull request #31960 from aletourneau/2015.8_ec2-getconsoleoutput
  - 8305978879 fixed ec2 get_console_output
- **PR** #31958: (rallytime) [2015.8] Merge forward from 2015.5 to 2015.8 @ *2016-03-18 15:12:44 UTC*
  - 1c7dc364ad Merge pull request #31958 from rallytime/merge-2015.8
- **PR** #31935: (twangboy) Back port nullsoft build script from 2015.8 @ *2016-03-17 14:54:50 UTC*
  - 2d1f2a0c2e Merge pull request #31935 from twangboy/fix_build_script2
  - 4af8c9dbfc Back port nullsoft build script from 2015.8
- **PR** #31912: (jfindlay) log.mixins: remove extermporaneous .record @ *2016-03-16 01:56:46 UTC*

  - 43240dc566 Merge pull request #31912 from jfindlay/log_mixin
  - 9f9c694654 log.mixins: remove extermporaneous .record

28.2.65 Salt 2015.5.0 Release Notes - Codename Lithium

The 2015.5.0 feature release of Salt is focused on hardening Salt and mostly on improving existing systems. A few major additions are present, primarily the new Beacon system. Most enhancements have been focused around improving existing features and interfaces.

As usual the release notes are not exhaustive and primarily include the most notable additions and improvements. Hundreds of bugs have been fixed and many modules have been substantially updated and added.

Warning: In order to fix potential shell injection vulnerabilities in salt modules, a change has been made to the various `cmd` module functions. These functions now default to `python_shell=False`, which means that the commands will not be sent to an actual shell.

The largest side effect of this change is that "shellisms", such as pipes, will not work by default. The modules shipped with salt have been audited to fix any issues that might have arisen from this change. Additionally, the `cmd` state module has been unaffected, and use of `cmd.run` in jinja is also unaffected. `cmd.run` calls on the CLI will also allow shellisms.

However, custom execution modules which use shellisms in `cmd` calls will break, unless you pass `python_shell=True` to these calls.

As a temporary workaround, you can set `cmd_safe: False` in your minion and master configs. This will revert the default, but is also less secure, as it will allow shell injection vulnerabilities to be written in custom code. We recommend you only set this setting for as long as it takes to resolve these issues in your custom code, then remove the override.

Note: Starting in this version of salt, `pillar_opts` defaults to False instead of True. This means that master opts will not be present in minion pillar, and as a result, `config.get` calls will not include master opts.

We recommend pillar is used for configuration options which need to make it to the minion.

Beacons

The beacon system allows the minion to hook into system processes and continually translate external events into the salt event bus. The primary example of this is the *`inotify`* beacon. This beacon uses inotify to watch configured files or directories on the minion for changes, creation, deletion etc.

This allows for the changes to be sent up to the master where the reactor can respond to changes.

Sudo Minion Settings

It is now possible to run the minion as a non-root user and for the minion to execute commands via sudo. Simply add *sudo_user: root* to the minion config, run the minion as a non-root user and grant that user sudo rights to execute salt-call.

Lazy Loader

The Lazy Loader is a significant overhaul of Salt's module loader system. The Lazy Loader will lazily load modules on access instead of all on start. In addition to a major performance improvement, this "sandboxes" modules so a bad/broken import of a single module will only affect jobs that require accessing the broken module. (:issue: *20274*)

Enhanced Active Directory Support

The eauth system for LDAP has been extended to support Microsoft Active Directory out of the box. This includes Active Directory and LDAP group support for eauth.

Salt LXC Enhancements

The LXC systems have been overhauled to be more consistent and to fix many bugs.

This overhaul makes using LXC with Salt much easier and substantially improves the underlying capabilities of Salt's LXC integration.

Salt SSH

- Additional configuration options and command line flags have been added to configure the scan roster on the fly
- Added support for `state.single` in `salt-ssh`
- Added support for `publish.publish`, `publish.full_data`, and `publish.runner` in `salt-ssh`
- Added support for `mine.get` in `salt-ssh`

New Windows Installer

The new Windows installer changes how Salt is installed on Windows. The old installer used bbfreeze to create an isolated python environment to execute in. This made adding modules and python libraries difficult. The new installer sets up a more flexible python environment making it easy to manage the python install and add python modules.

Instead of frozen packages, a full python implementation resides in the bin directory (`C:\salt\bin`). By executing pip or easy_install from within the Scripts directory (`C:\salt\bin\Scripts`) you can install any additional python modules you may need for your custom environment.

The .exe's that once resided at the root of the salt directory (`C:\salt`) have been replaced by .bat files and should function the same way as the .exe's in previous versions.

The new Windows Installer will not replace the minion config file and key if they already exist on the target system. Only the salt program files will be replaced. `C:\salt\conf` and `C:\salt\var` will remain unchanged.

Removed Requests Dependency

The hard dependency on the requests library has been removed. Requests is still required by a number of cloud modules but is no longer required for normal Salt operations.

This removal fixes issues that were introduced with requests and salt-ssh, as well as issues users experienced from the many different packaging methods used by requests package maintainers.

Python 3 Updates

While Salt does not YET run on Python 3 it has been updated to INSTALL on Python 3, taking us one step closer. What remains is getting the test suite to the point where it can run on Python 3 so that we can verify compatibility.

RAET Additions

The RAET support continues to improve. RAET now supports multi-master and many bugs and performance issues have been fixed. RAET is much closer to being a first class citizen.

Modified File Detection

A number of functions have been added to the RPM-based package managers to detect and diff files that are modified from the original package installs. This can be found in the new pkg.modified functions.

Reactor Update

Fix an infinite recursion problem for runner/wheel reactor jobs by passing a ”user” (Reactor) to all jobs that the reactor starts. The reactor skips all events created by that username – thereby only reacting to events not caused by itself. Because of this, runner and wheel executions from the runner will have user ”Reactor” in the job cache.

Misc Fixes/Additions

- SDB driver for etcd. (:issue: *22043*)
- Add `only_upgrade` argument to apt-based `pkg.install` to only install a package version if the package is already installed. (Great for security updates!)
- Joyent now requires a `keyname` to be specified in the provider configuration. This change was necessitated upstream by the 7.0+ API.
- Add `args` argument to `cmd.script_retcode` to match `cmd.script` in the `cmd module`. (:issue: *21122*)
- Fixed bug where TCP keepalive was not being sent on the defined interval on the return port (4506) from minion to master. (:issue: *21465*)
- LocalClient may now optionally raise SaltClientError exceptions. If using this class directly, checking for and handling this exception is recommended. (:issue: *21501*)
- The SAuth object is now a singleton, meaning authentication state is global (per master) on each minion. This reduces sign-ins of minions from 3->1 per startup.
- Nested outputter has been optimized, it is now much faster.
- Extensive fileserver backend updates.

Deprecations

- Removed `parameter` keyword argument from `eselect.exec_action` execution module.
- Removed `runas` parameter from the following `pip`` execution module functions: `install`, `uninstall`, `freeze`, `list_`, `list_upgrades`, `upgrade_available`, `upgrade`. Please migrate to `user`.

- Removed `runas` parameter from the following `pip` state module functions: `installed`, `removed`, `uptodate` . Please migrate to `user`.
- Removed `quiet` option from all functions in `cmdmod` execution module. Please use `output_loglevel=quiet` instead.
- Removed `parameter` argument from `eselect.set_` state. Please migrate to `module_parameter` or `action_parameter`.
- The `salt_events` table schema has changed to include an additional field called `master_id` to distinguish between events flowing into a database from multiple masters. If `event_return` is enabled in the master config, the database schema must first be updated to add the `master_id` field. This alteration can be accomplished as follows:

  ```
  ALTER TABLE salt_events ADD master_id VARCHAR(255) NOT NULL;
  ```

Known Issues

- In multi-master mode, a minion may become temporarily unresponsive if modules or pillars are refreshed at the same time that one or more masters are down. This can be worked around by setting 'auth_timeout' and 'auth_tries' down to shorter periods.

28.2.66 Salt 2015.5.1 Release Notes

release 2015-05-20

Version 2015.5.1 is a bugfix release for *2015.5.0*.

Statistics

- Total Merges: **203**
- Total Issue References: **30**
- Total PR References: **177**
- Contributors: **49** (Arabus, Lothiraldan, Snergster, TaiSHiNet, The-Loeki, UtahDave, aboe76, ahus1, basepi, bastiaanb, bradthurber, cachedout, cellscape, corywright, cro, dennisjac, dmyerscough, galet, garethgreenaway, gladiatr72, gtmanfred, iggy, ionutbalutoiu, jacobhammons, jayeshka, jfindlay, joejulian, jpic, justinta, kaidokert, kaithar, kiorky, lisa2lisa, msciciel, nleib, notpeter, optix2000, rahulhan, rallytime, rubic, ryan-lane, s0undt3ch, slinu3d, steverweber, techhat, terminalmage, ticosax, twangboy, whiteinge)

Cloud Runner Changes

The `fun` argument to the `cloud.action` runner has changed to `func`. Please update any calls to this runner.

Changelog for v2015.5.0..v2015.5.1

Generated at: 2018-05-27 20:58:00 UTC

- **PR** #23998: (rallytime) Update release note for 2015.5.1 @ *2015-05-20 20:58:55 UTC*
  - 2422760ebd Merge pull request #23998 from rallytime/release_notes
  - 113c6049f5 Update release note for 2015.5.1

- **PR** #23989: (rallytime) Backport #23980 to 2015.5 @ *2015-05-20 19:33:41 UTC*
  - **PR** #23980: (iggy) template: jinja2 -> jinja (refs: #23989)
  - 117ecb1fe0 Merge pull request #23989 from rallytime/bp-23980
  - 8f8557c47d template: jinja2 -> jinja
- **PR** #23988: (rallytime) Backport #23977 to 2015.5 @ *2015-05-20 19:13:36 UTC*
  - **PR** #23977: (ionutbalutoiu) Fixed glance image_create (refs: #23988)
  - d4f1ba02d7 Merge pull request #23988 from rallytime/bp-23977
  - 46fc7c6b69 Fixed glance image_create
- **PR** #23986: (basepi) [2015.5] Merge forward from 2014.7 to 2015.5 @ *2015-05-20 18:41:33 UTC*
  - 9566e7d412 Merge pull request #23986 from basepi/merge-forward-2015.5
  - 0b78156592 Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.5
    - 314e4db512 Merge pull request #23965 from hvnsweeting/20147-fix-gitfs-gitpython-exception
      - 2576301631 handle all exception gitpython can raise
- **PR** #23985: (UtahDave) Add 2014.7.5-2 and 2015.5.0-2 Windows installer download links @ *2015-05-20 18:32:44 UTC*
  - 9d1130ef8e Merge pull request #23985 from UtahDave/2015.5local
  - 10338d0c54 Add links to Windows 2015.5.0-2 install downloads
  - b84f9756c5 updated Windows 2014.7.5-2 installer download link
- **PR** #23983: (rallytime) Versionadded tags for https_user and https_pass args new in 2015.5.0 @ *2015-05-20 18:05:27 UTC*
  - ca7729d023 Merge pull request #23983 from rallytime/versionadded_git_options
  - 14eae22c91 Versionadded tags for https_user and https_pass args new in 2015.5.0
- **PR** #23970: (jayeshka) adding system unit test case @ *2015-05-20 17:12:57 UTC*
  - b06df57e03 Merge pull request #23970 from jayeshka/system-unit-test
  - 89eb00815e adding system unit test case
- **PR** #23967: (jayeshka) adding states/memcached unit test case @ *2015-05-20 17:12:26 UTC*
  - 38d5f75756 Merge pull request #23967 from jayeshka/memcached-states-unit-test
  - 8ef9240e25 adding states/memcached unit test case
- **PR** #23966: (jayeshka) adding states/modjk unit test case @ *2015-05-20 17:11:48 UTC*
  - 868e807d8a Merge pull request #23966 from jayeshka/modjk-states-unit-test
  - 422a96497d adding states/modjk unit test case
- **PR** #23942: (jacobhammons) Updates to sphinx saltstack2 doc theme @ *2015-05-20 15:43:54 UTC*
  - 63164900bd Merge pull request #23942 from jacobhammons/2015.5
  - 31023c8915 Updates to sphinx saltstack2 doc theme
- **ISSUE** #23872: (joejulian) create_ca_signed_cert can error if dereferenced dict is used for args (refs: #23874)
- **PR** #23874: (joejulian) Validate keyword arguments to be valid @ *2015-05-20 04:53:40 UTC*
  - 587957badc Merge pull request #23874 from joejulian/2015.5_tls_validate_kwargs

  - 30102acd04 Fix py3 and ordering inconsistency problems.
  - 493f7ad5f0 Validate keyword arguments to be valid
- **PR** #23960: (rallytime) Backport #22114 to 2015.5 @ *2015-05-20 04:37:09 UTC*
  - **PR** #22114: (dmyerscough) Fixing KeyError when there are no additional pages (refs: #23960)
  - 00c5c22867 Merge pull request #23960 from rallytime/bp-22114
  - f3e1d63fce Catch KeyError
  - 306b1ea6b8 Fixing KeyError
  - 6b2cda2861 Fix PEP8 complaint
  - 239e50f30d Fixing KeyError when there are no additional pages
- **PR** #23961: (rallytime) Backport #23944 to 2015.5 @ *2015-05-20 04:35:41 UTC*
  - **PR** #23944: (ryan-lane) Add missing loginclass argument to _changes call (refs: #23961)
  - 4648b46e05 Merge pull request #23961 from rallytime/bp-23944
  - 970d19a31e Add missing loginclass argument to _changes call
- **PR** #23948: (jfindlay) augeas.change state now returns changes as a dict @ *2015-05-20 04:00:10 UTC*
  - 0cb5cd3938 Merge pull request #23948 from jfindlay/augeas_changes
  - f09b80a8b5 augeas.change state now returns changes as a dict
- **PR** #23957: (rallytime) Backport #23951 to 2015.5 @ *2015-05-20 03:04:24 UTC*
  - **PR** #23951: (ryan-lane) Do not check perms in file.copy if preserve (refs: #23957)
  - 2d185f78f7 Merge pull request #23957 from rallytime/bp-23951
  - 996b431252 Update file.py
  - 85d461f748 Do not check perms in file.copy if preserve
- **ISSUE** #23839: (gladiatr72) wonky loader syndrome (refs: #23906)
- **ISSUE** #23373: (tnypex) reactor/orchestrate race condition on salt['pillar.get'] (refs: #23906)
- **PR** #23956: (rallytime) Backport #23906 to 2015.5 @ *2015-05-20 03:04:14 UTC*
  - **PR** #23906: (gladiatr72) Added exception handler to trap the RuntimeError raised when (refs: #23956)
  - ebff1ff967 Merge pull request #23956 from rallytime/bp-23906
  - 9d87fd335c add proper marker for format argument
  - 197688ef0c Added exception handler to trap the RuntimeError raised when Depends.enforce_dependency() class method fires unsuccessfully. There appears to be no synchronization within the Depends decorator class wrt the class global dependency_dict which results in incomplete population of any loader instantiation occuring at the time of one of these exceptions.
- **ISSUE** #19852: (TaiSHiNet) DigitalOcean APIv2 can't delete machines when there is only 1 page (refs: #23955)
- **ISSUE** #19304: (TaiSHiNet) DigitalOcean API v2 cannot delete VMs on 2nd page (refs: #19305)
- **PR** #23955: (rallytime) Backport #19305 to 2015.5 @ *2015-05-20 03:03:55 UTC*
  - **PR** #19305: (TaiSHiNet) Fixes droplet listing past page 1 (refs: #23955)
  - da3f9197d3 Merge pull request #23955 from rallytime/bp-19305
  - bbf2429bce Fixes droplet listing past page 1

- **PR** #23940: (basepi) [2015.5] Merge forward from 2014.7 to 2015.5 @ *2015-05-19 22:37:58 UTC*
  - 02a78fce3d Merge pull request #23940 from basepi/merge-forward-2015.5
  - 36f0065faf Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.5
    - 913391207a Merge pull request #23939 from basepi/v2014.7.6release
      - 32b65dc2a9 Add extended changelog to 2014.7.6 release notes
    - 0031ca2631 Merge pull request #23881 from garethgreenaway/23820_2014_7_schedule_list_issue
      - b207f2a433 Missing continue in the list function when deleting unused attributes.
    - 63bd21ecd2 Merge pull request #23887 from basepi/salt-ssh.pillar.get.22131
      - bc84502f46 Bring salt-ssh pillar.get in line with mainline pillar.get
- **PR** #23932: (rallytime) Backport #23908 to 2015.5 @ *2015-05-19 21:41:28 UTC*
  - **PR** #23908: (nleib) fix connection function to mongo (refs: #23932)
  - ee4c01bf30 Merge pull request #23932 from rallytime/bp-23908
  - 5d520c9377 fix connection function to mongo
- **PR** #23931: (rallytime) Backport #23880 to 2015.5 @ *2015-05-19 21:41:18 UTC*
  - **PR** #23880: (bastiaanb) if setting client_config_dir to '~', expand path (refs: #23931)
  - 70bd407920 Merge pull request #23931 from rallytime/bp-23880
  - 8ce59a2e16 if setting client_config_dir to '~', expand path
- **ISSUE** #23847: (kiorky) lxc: systemd containers cant be seeded (refs: #23806, #23898, #23897, #23808)
- **ISSUE** #23833: (kiorky) lxc.set_dns fails intermittently (refs: #23807, #23898, #23897, #23808)
- **ISSUE** #23772: (cheuschober) lxc.init fails to bootstrap container (refs: #23806, #23808, #23807, #23898, #23897)
- **ISSUE** #23658: (arthurlogilab) [salt-cloud lxc] too verbose, shows host: True multiple times when starting (refs: #23898, #23897)
- **ISSUE** #23657: (arthurlogilab) [salt-cloud lxc] NameError: global name '__salt__' is not defined (refs: #23898, #23727, #23897)
- **PR** #23898: (kiorky) Lxc profiles (refs: #23897) @ *2015-05-19 21:08:28 UTC*
  - **PR** #23897: (kiorky) Lxc seed and prof ports (refs: #23898)
  - **PR** #23808: (kiorky) Lxc seed and prof ports (refs: #23807, #23897)
  - **PR** #23807: (kiorky) Lxc profiles (refs: #23898)
  - **PR** #23806: (kiorky) Lxc seeding (refs: #23807)
  - 5bdbf0af9b Merge pull request #23898 from makinacorpus/lxc_profiles
  - d9051a047a lxc: systemd support
  - e8d674fed4 lxc: chroot fallback toggle
  - e2887a0d44 lxc: sync func name with develop
  - e96e345799 lxc more fixes (lxc.set_dns)
  - fdb64245d4 lxc: Fix salt config (no more a kwarg)
  - 63e63fa527 repair salt cloud lxc api on develop
  - 80eabe2703 lxc salt cloud doc

  - 73f229d966 lxc: unificate saltconfig/master/master_port
  - 0bc1f08a6b lxc: refactor a bit saltcloud/lxc interface
  - 7a80370da9 lxc: get networkprofile from saltcloud
  - 47acb2e159 lxc: default net profile has now correct options
  - 7eadf4863c lxc: select the appropriate default bridge
- **ISSUE** #23900: (hashi825) salt ubuntu network building issue 2015.5.0 (refs: #23922)
- **PR** #23922: (garethgreenaway) Fixes to debian_ip.py @ *2015-05-19 18:50:53 UTC*
  - b818f72dce Merge pull request #23922 from garethgreenaway/23900_2015_5_bonding_interface_fixes
  - 0bba536d6d Fixing issue reported when using bonded interfaces on Ubuntu. Attributes should be bond-, but the code was attempting to split just on bond_. Fix accounts for both, but the debian_ip.py module will write out bond attributes with bond-
- **PR** #23925: (jpic) Fixed wrong path in LXC cloud documentation @ *2015-05-19 18:23:56 UTC*
  - **PR** #23924: (jpic) Fixed wrong path in LXC cloud documentation (refs: #23925)
  - b1c98a38ed Merge pull request #23925 from jpic/fix/wrong_lxc_path
  - a4bcd75171 Fixed wrong path in LXC cloud documentation
- **PR** #23894: (whiteinge) Add __all__ attribute to Mock class for docs @ *2015-05-19 17:17:35 UTC*
  - 7f6a716a8a Merge pull request #23894 from whiteinge/doc-mock__all__
  - 6eeca46158 Add __all__ attribute to Mock class for docs
- **ISSUE** #23767: (chrimi) Salt system.locale fails on non existent default locale (refs: #23884)
- **PR** #23884: (jfindlay) Fix locale.set_locale on debian @ *2015-05-19 15:51:22 UTC*
  - 8108a9bd19 Merge pull request #23884 from jfindlay/fix_locale
  - 91c2d51400 use append_if_not_found in locale.set_locale
  - e63260391c (re)generate /etc/default/locale
- **PR** #23866: (jfindlay) backport #23834, change portage.dep.strip_empty to list comprehension @ *2015-05-19 15:50:43 UTC*
  - **PR** #23834: (Arabus) Avoid deprecation warning from portage.dep.strip_empty() (refs: #23866)
  - 6bae12fa8b Merge pull request #23866 from jfindlay/flag_strip
  - aa032ccfaf replace portage.dep.strip_empty() with list comprehension
  - 7576872280 Proper replacement for portage.dep.strip_empty() with list comprehension, pep8fix
  - 2851a5cf13 Switch portage.dep.strip_empty(…) to filter(None,…) to avoid deprecation warning and do essentially the same
- **ISSUE** #23904: (mbrgm) Network config bonding section cannot be parsed when attribute names use dashes (refs: #23917)
- **PR** #23917: (corywright) Split debian bonding options on dash instead of underscore @ *2015-05-19 15:44:35 UTC*
  - a67a008913 Merge pull request #23917 from corywright/issue23904
  - c06f8cf831 Split debian bonding options on dash instead of underscore
- **PR** #23909: (jayeshka) 'str' object has no attribute 'capitalized' @ *2015-05-19 15:41:53 UTC*

  - e8fcd0994d Merge pull request #23909 from jayeshka/file-exe-module
  - e422d9d200 'str' object has no attribute 'capitalized'
- **PR** #23903: (garethgreenaway) Adding docs for missing schedule state module parameters. @ *2015-05-19 06:29:34 UTC*
  - c73bf38927 Merge pull request #23903 from garethgreenaway/missing_docs_schedule_state
  - acd8ab9e1d Adding docs for missing schedule state module parameters.
  - a56697bd6e Merge branch '2015.5' of https://github.com/saltstack/salt into 2015.5
  - 1c2af5c685 Merge branch '2015.5' of https://github.com/saltstack/salt into 2015.5
  - ef581283fa Merge branch '2015.5' of https://github.com/saltstack/salt into 2015.5
  - 8664e8bc8d Merge branch '2015.5' of https://github.com/saltstack/salt into 2015.5-2
  - 46eb2655ee saltstack2 sphinx theme updates
  - e7442d3b1e Merge branch '2015.5' of https://github.com/saltstack/salt into 2015.5
  - ee3c1bd4a7 missed one
  - 3872921dd0 More updates to sphinx2 theme
  - fcd48657ef Merge branch '2015.5' of https://github.com/saltstack/salt into 2015.5
  - 8c32152be0 removed TOC numbering, additional tweaks to layout.html
  - 73dfaeff28 Merge branch '2015.5' of https://github.com/saltstack/salt into 2015.5
  - 16d8a753ad saltstack2 sphinx theme and build settings
- **ISSUE** #23847: (kiorky) lxc: systemd containers cant be seeded (refs: #23806, #23898, #23897, #23808)
- **ISSUE** #23772: (cheuschober) lxc.init fails to bootstrap container (refs: #23806, #23808, #23807, #23898, #23897)
- **PR** #23806: (kiorky) Lxc seeding (refs: #23807) @ *2015-05-18 23:18:33 UTC*
  - ff3cc7d331 Merge pull request #23806 from makinacorpus/lxc_seeding
  - 61b7aad308 runners/lxc: optim
- **PR** #23892: (basepi) [2015.5] Merge forward from 2014.7 to 2015.5 @ *2015-05-18 23:07:57 UTC*
  - 5f1a93d966 Merge pull request #23892 from basepi/merge-forward-2015.5
  - c2eed77691 Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.5
  - 17c5810c04 Merge pull request #23891 from basepi/releasenotes
    - dec153bcea Update the release notes index page
  - a93e58f80f Merge pull request #23888 from basepi/v2014.7.6release
    - 49921b6cb2 Update the 2014.7.6 release notes with CVE details
  - 50730287bb Merge pull request #23871 from rallytime/bp-23848
    - 379c09c3a5 Updated for SLES 12.
- **PR** #23875: (rallytime) Backport #23838 to 2015.5 @ *2015-05-18 22:28:55 UTC*
  - **PR** #23838: (gtmanfred) add refresh_beacons and sync_beacons (refs: #23875)
  - 66d13356b3 Merge pull request #23875 from rallytime/bp-23838
  - 3174227e8e Add versionadded directives to new beacon saltutil functions

  - 4a94b2c17b add refresh_beacons and sync_beacons
- **PR** #23876: (rallytime) Switch digital ocean tests to v2 driver @ *2015-05-18 22:17:13 UTC*
  - d294cf260b Merge pull request #23876 from rallytime/switch_digital_ocean_tests_v2
  - dce9b540a6 Remove extra line
  - 4acf58e758 Switch digital ocean tests to v2 driver
- **ISSUE** #23792: (neogenix) Salt Scheduler Incorrect Response (True, should be False) (refs: #23882)
- **PR** #23882: (garethgreenaway) Fixes to scheduler in 2015.5 @ *2015-05-18 22:09:24 UTC*
  - b97a48c7f5 Merge pull request #23882 from garethgreenaway/23792_2015_5_wrong_return_code
  - 37dbde6d57 Job already exists in schedule, should return False.
- **PR** #23868: (basepi) [2015.5] Merge forward from 2014.7 to 2015.5 @ *2015-05-18 18:35:54 UTC*
  - 61c922ea1a Merge pull request #23868 from basepi/merge-forward-2015.5
  - c9ed23394c Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.5
  - aee00c83df Merge pull request #23810 from rallytime/bp-23757
    - fb32c32065 use abspath, do not eliminating symlinks
  - 6b3352bb1a Merge pull request #23809 from rallytime/virt_get_nics_fix
    - 0616fb7884 Fix virtualport section of virt.get_nics loop
  - 188f03f567 Merge pull request #23823 from gtmanfred/2014.7
    - 5ef006d59d add link local for ipv6
  - f3ca682f92 Merge pull request #23802 from gtmanfred/2014.7
    - 2da98b58c8 if it is ipv6 ip_to_int will fail
- **PR** #23863: (rahulhan) Adding states/timezone.py unit test @ *2015-05-18 17:02:19 UTC*
  - 433f87372c Merge pull request #23863 from rahulhan/states_timezone_unit_test
  - 72fcabc690 Adding states/timezone.py unit test
- **PR** #23862: (rahulhan) Adding states/tomcat.py unit tests @ *2015-05-18 17:02:10 UTC*
  - 37b3ee5421 Merge pull request #23862 from rahulhan/states_tomcat_unit_test
  - 65d7752d2a Adding states/tomcat.py unit tests
- **PR** #23860: (rahulhan) Adding states/test.py unit tests @ *2015-05-18 17:01:49 UTC*
  - dde7207acb Merge pull request #23860 from rahulhan/states_test_unit_test
  - 1f4cf86500 Adding states/test.py unit tests
- **PR** #23859: (rahulhan) Adding states/sysrc.py unit tests @ *2015-05-18 17:01:46 UTC*
  - 3c9b8139e8 Merge pull request #23859 from rahulhan/states_sysrc_unit_test
  - 6a903b054d Adding states/sysrc.py unit tests
- **PR** #23812: (rallytime) Backport #23790 to 2015.5 @ *2015-05-18 15:30:34 UTC*
  - **PR** #23790: (aboe76) updated suse spec file to version 2015.5.0 (refs: #23812)
  - 4cf30a7ffa Merge pull request #23812 from rallytime/bp-23790
  - 3f65631cb6 updated suse spec file to version 2015.5.0

- **PR** #23811: (rallytime) Backport #23786 to 2015.5 @ *2015-05-18 15:30:27 UTC*
  - **PR** #23786: (kaithar) Log the error generated that causes returns.mysql.returner to except. (refs: #23811)
  - c6f939adfb Merge pull request #23811 from rallytime/bp-23786
  - 346f30bdda Log the error generated that causes returns.mysql.returner to except.
- **PR** #23850: (jayeshka) adding sysbench unit test case @ *2015-05-18 15:28:04 UTC*
  - ce60582de4 Merge pull request #23850 from jayeshka/sysbench-unit-test
  - 280abdec7c adding sysbench unit test case
- **PR** #23843: (The-Loeki) Fix erroneous virtual:physical core grain detection @ *2015-05-18 15:24:22 UTC*
  - 060902fefa Merge pull request #23843 from The-Loeki/patch-1
  - 9e2cf606eb Fix erroneous virtual:physical core grain detection
- **ISSUE** #23815: (Snergster) [beacons] inotify errors on subdir creation (refs: #23816)
- **ISSUE** #23685: (Snergster) inotify beacon on file. 'change' event to reactor to reset file to known state will cause loop (refs: #23816)
- **PR** #23816: (Snergster) Doc for #23685 Added prereq, caution, and additional mask information @ *2015-05-18 15:18:03 UTC*
  - 3257a9bead Merge pull request #23816 from Snergster/23685-doc-fix
  - 0fca49d52a Added prereq, caution, and additional mask information
- **PR** #23832: (ahus1) make saltify provider use standard boostrap procedure @ *2015-05-18 02:18:29 UTC*
  - **PR** #23829: (ahus1) make saltify provider use standard boostrap procedure (refs: #23832)
  - 3df3b85090 Merge pull request #23832 from ahus1/ahus1_saltify_bootstrap_2015.5
  - f5b1734782 fixing problem in unit test
  - cba47f6856 make saltify to use standard boostrap procedure, therefore providing all options like master_sign_pub_file
- **PR** #23791: (optix2000) Psutil compat @ *2015-05-16 04:05:54 UTC*
  - 8ec4fb2a73 Merge pull request #23791 from optix2000/psutil_compat
  - 5470cf58db Fix pylint errors and sloppy inline comments
  - 64634b6349 Update psutil.pid_list to use psutil.pids
  - 5dd6d69192 Fix imports that aren't in __all__
  - 8a1da33ada Fix test cases by mocking psutil_compat
  - 558798df1f Fix net_io_counters deprecation issue
  - 8140f92ba8 Override unecessary pylint errors
  - 7d02ad4f06 Fix some of the mock names for the new API
  - 9b3023e851 Fix overloaded getters/setters. Fix line lengths
  - 180eb87a46 Fix whitespace
  - f8edf72f98 Use new psutil API in ps module
  - e48982ff9c Fix version checking in psutil_compat
  - 93ee411fd5 Create compatability psutil. psutil 3.0 drops 1.0 API, but we still support old psutil versions.

- **PR** #23782: (terminalmage) Replace "command -v" with "which" and get rid of spurious log messages @ *2015-05-16 04:03:10 UTC*
  - 405517be8b Merge pull request #23782 from terminalmage/issue23772
  - 0f6f239052 More ignore_retcode to suppress spurious log msgs
  - b4c48e62ea Ignore return code in lxc.attachable
  - 08658c0177 Replace "command -v" with "which"
- **PR** #23783: (basepi) [2015.5] Merge forward from 2014.7 to 2015.5 @ *2015-05-15 21:38:51 UTC*
  - cb2eb401f3 Merge pull request #23783 from basepi/merge-forward-2015.5
  - 9df51caf28 __opts__.get
  - 51d23ed9d0 Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.5
    - d9af0c3e82 Merge pull request #23488 from cellscape/lxc-cloud-fixes
      - 64250a67e5 Remove profile from opts after creating LXC container
      - c4047d2a71 Set destroy=True in opts when destroying cloud instance
      - 9e1311a7cd Store instance names in opts when performing cloud action
      - 934bc57c73 Correctly pass custom env to lxc-attach
      - 7fb85f7be1 Preserve test=True option in cloud states
      - 9771b5a313 Fix detection of absent LXC container in cloud state
      - fb24f0cf02 Report failure when failed to create/clone LXC container
      - 2d9aa2bb97 Avoid shadowing variables in lxc module
      - 792e1021f2 Allow to override profile options in lxc.cloud_init_interface
      - 42bd64b9b3 Return changes on successful lxc.create from salt-cloud
      - 4409eabb83 Return correct result when creating cloud LXC container
      - 377015c881 Issue #16424: List all providers when creating salt-cloud instance without profile
    - 808bbe1cb2 Merge pull request #23748 from basepi/salt-ssh.roster.host.check
      - bc53e049e0 Log entire exception for render errors in roster
      - 753de6a621 Log render errors in roster to error level
      - e01a7a90b3 Always let the real YAML error through
    - 72cf360255 Merge pull request #23731 from twangboy/fix_22959
      - 88e5495b2d Fixes #22959: Trying to add a directory to an unmapped drive in windows
    - 2610195262 Merge pull request #23730 from rallytime/bp-23729
      - 1877caecba adding support for nested grains to grains.item
    - 3e9df883d6 Merge pull request #23688 from twangboy/fix_23415
      - 6a91169bae Fixed unused-import pylint error
      - 5e25b3f355 fixed pylint errors
      - 1a9676626f Added inet_pton to utils/validate/net.py for ip.set_static_ip in windows
- **PR** #23781: (jfindlay) fix unit test mock errors on arch @ *2015-05-15 19:40:07 UTC*

  - 982f87316d Merge pull request #23781 from jfindlay/fix_locale_tests
  - 14c711eeb3 fix unit test mock errors on arch
- **ISSUE** #23566: (rks2286) Salt-cp corrupting the file after transfer to minion (refs: #23740)
- **PR** #23740: (jfindlay) Binary write @ *2015-05-15 18:10:44 UTC*
  - 916b1c4f7c Merge pull request #23740 from jfindlay/binary_write
  - 626930a4e5 update incorrect comment wording
  - a978f5c091 always use binary file write mode on windows
- **ISSUE** #23682: (chrish42) Pip module requires system pip, even when not used (with env_bin) (refs: #23736)
- **PR** #23736: (jfindlay) always load pip execution module @ *2015-05-15 18:10:16 UTC*
  - 348645ecd5 Merge pull request #23736 from jfindlay/fix_pip
  - b8867a8c23 update pip tests
  - 040bbc42d2 only check pip version in one place
  - 6c453a5a2a check for executable status of bin_env
  - 3337257833 always load the pip module as pip could be anywhere
- **PR** #23770: (cellscape) Fix cloud LXC container destruction @ *2015-05-15 17:38:59 UTC*
  - 10cedfb174 Merge pull request #23770 from cellscape/fix-cloud-lxc-destruction
  - 4f6021c884 Fix cloud LXC container destruction
- **PR** #23759: (lisa2lisa) fixed the problem for not beable to revoke ., for more detail https… @ *2015-05-15 17:38:38 UTC*
  - ddea822b02 Merge pull request #23759 from lisa2lisa/iss23664
  - a29f161a58 fixed the problem for not beable to revoke ., for more detail https://github.com/saltstack/salt/issues/23201, fixed mysql cannot create user with pure digit password, for more info https://github.com/saltstack/salt/issues/23664
- **PR** #23769: (cellscape) Fix file_roots CA lookup in salt.utils.http.get_ca_bundle @ *2015-05-15 16:21:49 UTC*
  - 10615ff5a7 Merge pull request #23769 from cellscape/utils-http-ca-file-roots
  - 8e90f3291b Fix file_roots CA lookup in salt.utils.http.get_ca_bundle
- **PR** #23765: (jayeshka) adding states/makeconf unit test case @ *2015-05-15 14:29:43 UTC*
  - fd8a1b797f Merge pull request #23765 from jayeshka/makeconf_states-unit-test
  - 26e31afa31 adding states/makeconf unit test case
- **PR** #23760: (ticosax) [doc] document refresh argument @ *2015-05-15 14:23:47 UTC*
  - ee13b08027 Merge pull request #23760 from ticosax/2015.5
  - e3ca859ba6 document refresh argument
- **PR** #23766: (jayeshka) adding svn unit test case @ *2015-05-15 14:23:18 UTC*
  - a017f725a4 Merge pull request #23766 from jayeshka/svn-unit-test
  - 19939cfa98 adding svn unit test case
- **ISSUE** #23734: (bradthurber) 2015.5.0 modules/archive.py ZipFile instance has no attribute '__exit__' - only python 2.6? (refs: #23737)

- **PR** #23751: (rallytime) Backport #23737 to 2015.5 @ *2015-05-15 03:58:37 UTC*
  - **PR** #23737: (bradthurber) fix for 2015.5.0 modules/archive.py ZipFile instance has no attribute… (refs: #23751)
  - 0ed9d45114 Merge pull request #23751 from rallytime/bp-23737
  - 8d1eb326d0 fix for 2015.5.0 modules/archive.py ZipFile instance has no attribute '__exit__' - only python 2.6? #23734
- **ISSUE** #23709: (kiorky) cmdmod: enhancement is really needed for stateful commands (refs: #23710)
- **PR** #23710: (kiorky) Get more useful output from stateful commands @ *2015-05-14 21:58:10 UTC*
  - d73984ec9c Merge pull request #23710 from makinacorpus/i23709
  - c70690969e Get more useful output from stateful commands
- **ISSUE** #23608: (kaidokert) salt-cloud file_map with non-root user (refs: #23609)
- **PR** #23724: (rallytime) Backport #23609 to 2015.5 @ *2015-05-14 19:34:22 UTC*
  - **PR** #23609: (kaidokert) file_map: chown created directories if not root #23608 (refs: #23724)
  - cdf421b9ed Merge pull request #23724 from rallytime/bp-23609
  - fe3a762673 file_map: chmod created directories if not root
- **PR** #23723: (rallytime) Backport #23568 to 2015.5 @ *2015-05-14 19:34:11 UTC*
  - **PR** #23568: (techhat) Allow Salt Cloud to use either SCP or SFTP, as configured (refs: #23723)
  - 94f9099307 Merge pull request #23723 from rallytime/bp-23568
  - bbec34abd3 Allow Salt Cloud to use either SCP or SFTP, as configured
- **PR** #23725: (rallytime) Backport #23691 to 2015.5 @ *2015-05-14 19:32:30 UTC*
  - **PR** #23691: (dennisjac) add initial configuration documentation for varstack pillar (refs: #23725)
  - 137e5eefd0 Merge pull request #23725 from rallytime/bp-23691
  - 28a846ebe8 add initial configuration documentation for varstack pillar
- **PR** #23722: (rallytime) Backport #23472 to 2015.5 @ *2015-05-14 19:31:52 UTC*
  - **PR** #23472: (techhat) Allow neutron network list to be used as pillar data (refs: #23722)
  - 0c00995dfb Merge pull request #23722 from rallytime/bp-23472
  - c3d0f39515 Change versionadded tag for backport
  - 023e88f264 Allow neutron network list to be used as pillar data
- **ISSUE** #23657: (arthurlogilab) [salt-cloud lxc] NameError: global name '__salt__' is not defined (refs: #23898, #23727, #23897)
- **PR** #23727: (jfindlay) fix npm execution module stacktrace @ *2015-05-14 18:14:12 UTC*
  - cbf4ca8d91 Merge pull request #23727 from jfindlay/npm_salt
  - 05392f282e fix npm execution module stacktrace
- **PR** #23718: (rahulhan) Adding states/user.py unit tests @ *2015-05-14 17:15:38 UTC*
  - ef536d58de Merge pull request #23718 from rahulhan/states_user_unit_tests
  - aad27db513 Adding states/user.py unit tests
- **PR** #23720: (basepi) [2015.5] Merge forward from 2014.7 to 2015.5 @ *2015-05-14 17:13:02 UTC*

  - a529d74079 Merge pull request #23720 from basepi/merge-forward-2015.5
  - 06a3ebd9d1 Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.5
  - 1b86460d73 Merge pull request #23680 from cachedout/issue_23403
    - d5986c21b4 Rename kwarg in cloud runner
  - cd64af0ce4 Merge pull request #23674 from cachedout/issue_23548
    - da8a2f5cb3 Handle lists correctly in grains.list_prsesent
  - d322a19213 Merge pull request #23672 from twangboy/fix_user_present
    - 731e7af3dd Merge branch '2014.7' of https://github.com/saltstack/salt into fix_user_present
    - d6f70a4545 Fixed user.present to create password in windows
  - 43f7025000 Merge pull request #23670 from rallytime/bp-23607
    - ed30dc4642 Fix for #23604. No error reporting. Exitcode !=0 are ok
- **PR** #23704: (jayeshka) adding states/lvs_server unit test case @ *2015-05-14 14:22:10 UTC*
  - 13facbf077 Merge pull request #23704 from jayeshka/lvs_server_states-unit-test
  - da323dab0b adding states/lvs_server unit test case
- **PR** #23703: (jayeshka) adding states/lvs_service unit test case @ *2015-05-14 14:21:23 UTC*
  - f95ca3188f Merge pull request #23703 from jayeshka/lvs_service_states-unit-test
  - 66717c8133 adding states/lvs_service unit test case
- **PR** #23702: (jayeshka) Remove superfluous return statement. @ *2015-05-14 14:20:42 UTC*
  - 07e987e327 Merge pull request #23702 from jayeshka/fix_lvs_service
  - ecff2181e4 fix lvs_service
- **PR** #23686: (jfindlay) remove superflous return statement @ *2015-05-14 14:20:18 UTC*
  - 39973d4095 Merge pull request #23686 from jfindlay/fix_lvs_server
  - 5aaeb73532 remove superflous return statement
- **PR** #23690: (rallytime) Backport #23424 to 2015.5 @ *2015-05-13 23:04:36 UTC*
  - **PR** #23424: (justinta) Added python_shell=True for refresh_db in pacman.py (refs: #23690)
  - be7c7ef3fd Merge pull request #23690 from rallytime/bp-23424
  - 94574b7367 Added python_shell=True for refresh_db in pacman.py
- **PR** #23681: (cachedout) Start on 2015.5.1 release notes @ *2015-05-13 19:44:22 UTC*
  - 1a0db43097 Merge pull request #23681 from cachedout/2015_5_1_release_notes
  - bdbbfa6ee7 Start on 2015.5.1 release notes
- **PR** #23679: (jfindlay) Merge #23616 @ *2015-05-13 19:03:53 UTC*
  - **PR** #23616: (Snergster) virtual returning none warning fixed in dev but missed in 2015.5 (refs: #23679)
  - b54075a2ac Merge pull request #23679 from jfindlay/merge_23616
  - 6e15e19907 appease pylint's blank line strictures
  - 8750680d9e virtual returning none warning fixed in dev but missed in 2015.5
- **PR** #23675: (basepi) [2015.5] Merge forward from 2014.7 to 2015.5 @ *2015-05-13 18:35:54 UTC*

- e480f13688 Merge pull request #23675 from basepi/merge-forward-2015.5
- bd635488ef Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.5
  - 0f006ac1d8 Merge pull request #23661 from rallytime/merge-23640
    - 4427f42bb6 Whitespace fix
    - dd9115466e Add warning to get_or_set_hash about reserved chars
  - 84e2ef88fc Merge pull request #23639 from cachedout/issue_23452
    - d418b49a77 Syntax error!
    - 45b4015d7d Handle exceptions raised by __virtual__
  - bd9b94ba8c Merge pull request #23637 from cachedout/issue_23611
    - 56cb1f52e3 Fix typo
    - f6fcf19a7f Convert str master to list
  - f20c0e42ce Merge pull request #23595 from rallytime/bp-23549
    - 6efcac09ad Update __init__.py
  - 1acaf86da7 Merge pull request #23594 from rallytime/bp-23496
    - d5ae1d268a Fix for issue #23110 This resolves issues when the freshly created directory is removed by fileserver.update.
  - 2c221c7332 Merge pull request #23593 from rallytime/bp-23442
    - 39869a15bd check w/ low['name'] only
    - 304cc499e9 another fix for file defined w/ id, but require name
    - 8814d4180e add directory itself to keep list
  - fadd1ef63c Merge pull request #23606 from twangboy/fix_installer
    - 038331edab Fixed checkbox for starting service and actually starting it
- acdd3fc6bd Fix lint
- 680e88f058 Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.5
  - 10b3f0f643 Merge pull request #23592 from rallytime/bp-23389
    - 734cc43801 Correct fail_hard typo
  - cd34b9b6c4 Merge pull request #23573 from techhat/novaquery
    - f92db5e92f Linting
    - 26e00d3ccc Scan all available networks for public and private IPs
  - 2a72cd71c2 Merge pull request #23558 from jfindlay/fix_ebuild
    - 45404fb2a6 reorder emerge command line
  - a664a3c6fd Merge pull request #23530 from dr4Ke/fix_salt-ssh_to_include_pkg_sources
    - 5df6a8008c fix pylint warning
    - d0549e56ba salt-ssh state: fix including all salt:// references
  - 55c3869861 Merge pull request #23433 from twangboy/list_pkgs_fix
    - 8ab5b1b86f Fix pylint error

        - 2d11d6545e Obtain all software from the registry
    - 755bed0abd Merge pull request #23554 from jleroy/debian-hostname-fix
        - 5ff749e487 Debian: Hostname always updated
    - 6ec87ce9f5 Merge pull request #23551 from dr4Ke/grains.append_unit_tests
        - ebff9df5b2 fix pylint errors
        - c4954046ad unit tests for grains.append module function
        - 0c9a32326c use MagickMock
        - c838a22377 unit tests for grains.append module function
    - e96c5c5bf3 Merge pull request #23474 from dr4Ke/fix_grains.append_nested
        - a01a5bb51e grains.get, parameter delimititer, versionadded: 2014.7.6
        - b39f50475d remove debugging output
        - b6e15e295c fix grains.append in nested dictionnary grains #23411
    - ab7e1aed8e Merge pull request #23537 from t0rrant/patch-1
        - 8e03cc99d3 Update changelog
- **PR** #23669: (rallytime) Backport #23586 to 2015.5 @ *2015-05-13 18:27:11 UTC*
  - **PR** #23586: (Lothiraldan) Fix salt.state.file._unify_sources_and_hashes when sources is used without sources_hashes (refs: #23669)
  - 0dad6be0fc Merge pull request #23669 from rallytime/bp-23586
  - ef4c6adae3 Remove another unused import
  - 73cfda751a Remove unused import
  - 52b68d695a Use the zip_longest from six module for python 3 compatiblity
  - 18d5ff9a8e Fix salt.state.file._unify_sources_and_hashes when sources is used without sources_hashes
- **PR** #23662: (rallytime) Merge #23642 with pylint fix @ *2015-05-13 15:46:51 UTC*
  - **PR** #23642: (cachedout) Let saltmod handle lower-level exceptions gracefully (refs: #23662)
  - fabef759e0 Merge pull request #23662 from rallytime/merge-23642
  - aa7bbd84fa Remove unused import
  - 9e66d4c88e Let saltmod handle lower-level exceptions gracefully
- **PR** #23622: (jfindlay) merge #23508 @ *2015-05-13 15:36:49 UTC*
  - **PR** #23508: (cro) Port mysql returner to postgres using jsonb datatype (refs: #23622)
  - 072b92733d Merge pull request #23622 from jfindlay/pgjsonb
  - 454322c7e4 appease pylint's proscription on blank line excess
  - 57c617136d Get time with timezone correct also in job return.
  - e109d0f643 Get time with timezone correct.
  - 21e06b9112 Fix SQL, remove unneeded imports.
  - 653f360723 Stop making changes in 2 places.
  - d6daaa0292 Typo.

  - 7d748bff75 SSL is handled differently by Pg, so don't set it here.
  - cc7c377bcd Fill alter_time field in salt_events with current time with timezone.
  - 43defe9b20 Port mysql module to Postgres using jsonb datatypes
- **PR** #23651: (jayeshka) adding solr unit test case @ *2015-05-13 15:26:15 UTC*
  - c1bdd4d377 Merge pull request #23651 from jayeshka/solr-unit-test
  - 6e05148962 adding solr unit test case
- **PR** #23649: (jayeshka) adding states/libvirt unit test case @ *2015-05-13 15:24:48 UTC*
  - ee43411677 Merge pull request #23649 from jayeshka/libvirt_states-unit-test
  - 0fb923a283 adding states/libvirt unit test case
- **PR** #23648: (jayeshka) adding states/linux_acl unit test case @ *2015-05-13 15:24:11 UTC*
  - c7fc466f1e Merge pull request #23648 from jayeshka/linux_acl_states-unit-test
  - 3f0ab29eb0 removed error.
  - 11081c121c adding states/linux_acl unit test case
- **PR** #23650: (jayeshka) adding states/kmod unit test case @ *2015-05-13 15:09:18 UTC*
  - 4cba7ba35c Merge pull request #23650 from jayeshka/kmod_states-unit-test
  - 1987015033 adding states/kmod unit test case
- **PR** #23633: (jayeshka) made changes to test_interfaces function. @ *2015-05-13 06:51:07 UTC*
  - bc8faf1543 Merge pull request #23633 from jayeshka/win_network-2015.5-unit-test
  - 0936e1d386 made changes to test_interfaces function.
- **PR** #23619: (jfindlay) fix kmod.present processing of module loading @ *2015-05-13 01:16:56 UTC*
  - 7df3579bbc Merge pull request #23619 from jfindlay/fix_kmod_state
  - 73facbfc1f fix kmod.present processing of module loading
- **PR** #23598: (rahulhan) Adding states/win_dns_client.py unit tests @ *2015-05-12 21:47:36 UTC*
  - d4f30955fa Merge pull request #23598 from rahulhan/states_win_dns_client_unit_test
  - d08d885828 Adding states/win_dns_client.py unit tests
- **PR** #23597: (rahulhan) Adding states/vbox_guest.py unit tests @ *2015-05-12 21:46:30 UTC*
  - 811c6a1d89 Merge pull request #23597 from rahulhan/states_vbox_guest_unit_test
  - 6a2909eeea Removed errors
  - 4cde78a58a Adding states/vbox_guest.py unit tests
- **PR** #23615: (rallytime) Backport #23577 to 2015.5 @ *2015-05-12 21:19:11 UTC*
  - **PR** #23577: (msciciel) Fix find and remove functions to pass database param (refs: #23615)
  - 029ff1103d Merge pull request #23615 from rallytime/bp-23577
  - 6f74477129 Fix find and remove functions to pass database param
- **PR** #23603: (rahulhan) Adding states/winrepo.py unit tests @ *2015-05-12 18:40:12 UTC*
  - b8589532d1 Merge pull request #23603 from rahulhan/states_winrepo_unit_test
  - a66e7e7f1f Adding states/winrepo.py unit tests

- **PR** #23602: (rahulhan) Adding states/win_path.py unit tests @ *2015-05-12 18:39:37 UTC*
  - 3cbbd6d277 Merge pull request #23602 from rahulhan/states_win_path_unit_test
  - 122c29f71a Adding states/win_path.py unit tests
- **PR** #23600: (rahulhan) Adding states/win_network.py unit tests @ *2015-05-12 18:39:01 UTC*
  - 3c904e8739 Merge pull request #23600 from rahulhan/states_win_network_unit_test
  - b418404eb7 removed lint error
  - 1be802300b Adding states/win_network.py unit tests
- **PR** #23599: (rahulhan) Adding win_firewall.py unit tests @ *2015-05-12 18:37:49 UTC*
  - 10243a7742 Merge pull request #23599 from rahulhan/states_win_firewall_unit_test
  - 6cda890517 Adding win_firewall.py unit tests
- **PR** #23601: (basepi) Add versionadded for jboss module/state @ *2015-05-12 17:22:59 UTC*
  - e73071dbdf Merge pull request #23601 from basepi/jboss.version.added
  - 0174c8fe58 Add versionadded for jboss module/state
- **PR** #23469: (s0undt3ch) Call the windows specific function not the general one @ *2015-05-12 16:47:22 UTC*
  - 9beb7bc529 Merge pull request #23469 from s0undt3ch/hotfix/call-the-win-func
  - 83e88a3eb1 Call the windows specific function not the general one
- **PR** #23583: (jayeshka) adding states/ipset unit test case @ *2015-05-12 16:31:55 UTC*
  - d2f097584c Merge pull request #23583 from jayeshka/ipset_states-unit-test
  - 4330cf4a6e adding states/ipset unit test case
- **PR** #23582: (jayeshka) adding states/keyboard unit test case @ *2015-05-12 16:31:17 UTC*
  - 82a47e8cbf Merge pull request #23582 from jayeshka/keyboard_states-unit-test
  - fa94d7ab5c adding states/keyboard unit test case
- **PR** #23581: (jayeshka) adding states/layman unit test case @ *2015-05-12 16:30:36 UTC*
  - 77e5b28566 Merge pull request #23581 from jayeshka/layman_states-unit-test
  - 297b055b1c adding states/layman unit test case
- **PR** #23580: (jayeshka) adding smf unit test case @ *2015-05-12 16:29:58 UTC*
  - cbe32828ef Merge pull request #23580 from jayeshka/smf-unit-test
  - 4f9719157b adding smf unit test case
- **ISSUE** #21603: (ipmb) ssh_auth.present fails on key without comment (refs: #23572)
- **PR** #23572: (The-Loeki) Fix regression of #21355 introduced by #21603 @ *2015-05-12 16:28:05 UTC*
  - **PR** #21355: (The-Loeki) Fix for comments containing whitespaces (refs: #23572)
  - 16a333832a Merge pull request #23572 from The-Loeki/ssh_auth_fix
  - d8248dd368 Fix regression of #21355 introduced by #21603
- **ISSUE** #23490: (lichtamberg) salt.modules.aptpkg.upgrade should have default "dist_upgrade=False" (refs: #23565)
- **PR** #23565: (garethgreenaway) fix to aptpkg module @ *2015-05-12 16:25:46 UTC*

  - f843f89cd7 Merge pull request #23565 from garethgreenaway/2015_2_aptpkg_upgrade_default_to_upgrade
  - 97ae514641 aptpkg.upgrade should default to upgrade instead of dist_upgrade.
- **ISSUE** #23473: (terminalmage) unit.modules.rh_ip_test.RhipTestCase.test_build_bond is not properly mocked (refs: #23550)
- **PR** #23550: (jfindlay) additional mock for rh_ip_test test_build_bond @ *2015-05-12 15:17:16 UTC*
  - c1157cdaee Merge pull request #23550 from jfindlay/fix_rh_ip_test
  - e9b94d36d3 additional mock for rh_ip_test test_build_bond
- **PR** #23552: (garethgreenaway) Fix for an issue caused by a previous pull request @ *2015-05-11 21:54:59 UTC*
  - b593328176 Merge pull request #23552 from garethgreenaway/2015_5_returner_fix_broken_previous_pr
  - 7d70e2b334 Passed argumentes in the call _fetch_profile_opts to were in the wrong order
- **PR** #23547: (slinu3d) Added AWS v4 signature support for 2015.5 @ *2015-05-11 21:52:24 UTC*
  - d0f96825dd Merge pull request #23547 from slinu3d/2015.5
  - f3bfdb561b Fixed urlparse and urlencode calls
  - 802dbdb965 Added AWS v4 signature support for 2015.5
- **PR** #23544: (basepi) [2015.5] Merge forward from 2014.7 to 2015.5 @ *2015-05-11 18:02:06 UTC*
  - 06c6a1f44a Merge pull request #23544 from basepi/merge-forward-2015.5
  - f8a36bc155 Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.5
    - b79fed3a92 Merge pull request #23538 from cro/licupdate
      - 345efe25c9 Update date in LICENSE file
    - a123a36f05 Merge pull request #23505 from aneeshusa/remove-unused-ssh-config-validator
      - 90af1672ca Remove unused ssh config validator. Fixes #23159.
    - ca2c21a63c Merge pull request #23467 from slinu3d/2014.7
      - 0b4081d8f4 Fixed pylint error at line 363
      - 5be5eb5b14 Fixed pylink errors
      - e64f374ffa Fixed lint errors
      - b9d1ac4f1f Added AWS v4 signature support
    - e6f9eec02e Merge pull request #23444 from techhat/novacreateattach
      - ebdb7eae2d Add create_attach_volume to nova driver
    - e331463319 Merge pull request #23460 from s0undt3ch/hotfix/bootstrap-script-2014.7
      - edcd0c41f2 Update to latest stable bootstrap script v2015.05.07
    - 7a8ce1a954 Merge pull request #23439 from techhat/maxtries
      - 0ad3ff2c88 Add wait_for_passwd_maxtries variable
- **ISSUE** #23426: (twangboy) Can't restart salt-minion on 64 bit windows (2015.5.0) (refs: #23470)
- **PR** #23470: (twangboy) Fixed service.restart for salt-minion @ *2015-05-11 17:54:47 UTC*
  - aa5b896d3e Merge pull request #23470 from twangboy/fix_svc_restart
  - b3f284c517 Fixed tests

  - ad44d79f26 Fixed service.restart for salt-minion
- **PR** #23539: (rahulhan) Adding states/virtualenv_mod.py unit tests @ *2015-05-11 17:02:31 UTC*
  - 67988b21ee Merge pull request #23539 from rahulhan/states_virtualenv_mod_unit_test
  - 750bb07d1c Adding states/virtualenv_mod.py unit tests
  - c96619653e Merge pull request #23423 from cachedout/remove_jid_event_from_orch
    - f81aab7627 Remove jid_event from state.orch
  - 2bb09b7ee7 Merge pull request #23509 from keesbos/Catch_empty_environment
    - 6dedeaccd2 Catch the unset (empty/None) environment case
  - 6d42f30271 Merge pull request #23245 from freimer/issue_23244
    - 24cf6ebad5 Add Caller functionality to reactors.
- **PR** #23513: (gladiatr72) short-circuit auto-failure of iptables.delete state @ *2015-05-11 15:18:33 UTC*
  - c3f03d827d Merge pull request #23513 from gladiatr72/RFC_stop_iptables.check_from_short-circuiting_position-only_delete_rule
  - c71714c364 short-circuit auto-failure of iptables.delete state if position argument is set without the other accoutrements that check_rule requires.
- **PR** #23534: (jayeshka) adding states/ini_manage unit test case @ *2015-05-11 14:32:06 UTC*
  - 4e77f6f8c4 Merge pull request #23534 from jayeshka/ini_manage_states-unit-test
  - 831223c31c adding states/ini_manage unit test case
- **PR** #23533: (jayeshka) adding states/hipchat unit test case @ *2015-05-11 14:30:22 UTC*
  - 11ba9ed99b Merge pull request #23533 from jayeshka/hipchat-states-unit-test
  - 41d14b322d adding states/hipchat unit test case
- **PR** #23532: (jayeshka) adding states/ipmi unit test case @ *2015-05-11 14:28:15 UTC*
  - e5421139d3 Merge pull request #23532 from jayeshka/ipmi-states-unit-test
  - fc3e64a8a4 adding states/ipmi unit test case
- **PR** #23531: (jayeshka) adding service unit test case @ *2015-05-11 14:27:12 UTC*
  - 9ba85fd31a Merge pull request #23531 from jayeshka/service-unit-test
  - 3ad5314ee0 adding service unit test case
- **ISSUE** #23512: (mostafahussein) hipchat_returner / slack_returner not work correctly (refs: #23517)
- **PR** #23517: (garethgreenaway) fix to returners @ *2015-05-11 14:20:51 UTC*
  - 32838cd888 Merge pull request #23517 from garethgreenaway/23512_2015_5_returners_with_profiles
  - 81e31e27cf fix for returners that utilize profile attributes. code in the if else statement was backwards. #23512
- **PR** #23502: (rahulhan) Adding states/win_servermanager.py unit tests @ *2015-05-08 19:47:18 UTC*
  - 6be7d8d13b Merge pull request #23502 from rahulhan/states_win_servermanager_unit_test
  - 2490074aa2 Adding states/win_servermanager.py unit tests
- **PR** #23495: (jayeshka) adding seed unit test case @ *2015-05-08 17:30:38 UTC*
  - 604857811e Merge pull request #23495 from jayeshka/seed-unit-test

  - 3f134bc573 adding seed unit test case
- **PR** #23494: (jayeshka) adding sensors unit test case @ *2015-05-08 17:30:18 UTC*
  - 70bc3c1415 Merge pull request #23494 from jayeshka/sensors-unit-test
  - 1fb48a31a8 adding sensors unit test case
- **PR** #23493: (jayeshka) adding states/incron unit test case @ *2015-05-08 17:29:59 UTC*
  - b981b20d44 Merge pull request #23493 from jayeshka/incron-states-unit-test
  - cc7bc170f3 adding states/incron unit test case
- **PR** #23492: (jayeshka) adding states/influxdb_database unit test case @ *2015-05-08 17:29:51 UTC*
  - 4019c493a1 Merge pull request #23492 from jayeshka/influxdb_database-states-unit-test
  - e1fcac815d adding states/influxdb_database unit test case
- **PR** #23491: (jayeshka) adding states/influxdb_user unit test case @ *2015-05-08 16:24:07 UTC*
  - d317a77afb Merge pull request #23491 from jayeshka/influxdb_user-states-unit-test
  - 9d4043f9ff adding states/influxdb_user unit test case
- **PR** #23477: (galet) LDAP auth: Escape filter value for group membership search @ *2015-05-07 22:04:48 UTC*
  - e0b2a73eb4 Merge pull request #23477 from galet/ldap-filter-escaping
  - 33038b9f86 LDAP auth: Escape filter value for group membership search
- **PR** #23476: (cachedout) Lint becaon @ *2015-05-07 19:55:36 UTC*
  - **PR** #23431: (UtahDave) Beacon fixes (refs: #23476)
  - e1719fe26b Merge pull request #23476 from cachedout/lint_23431
  - 8d1ff209eb Lint becaon
- **PR** #23431: (UtahDave) Beacon fixes (refs: #23476) @ *2015-05-07 19:53:47 UTC*
  - 1e299ede4f Merge pull request #23431 from UtahDave/beacon_fixes
  - 152f2235c2 remove unused import
  - 81198f9399 fix interval logic and example
  - 5504778adf update to proper examples
  - 6890439d58 fix list for mask
  - ee7b579e90 remove custom interval code.
- **PR** #23468: (rahulhan) Adding states/win_system.py unit tests @ *2015-05-07 19:20:50 UTC*
  - ea55c44bbb Merge pull request #23468 from rahulhan/states_win_system_unit_test
  - 33f8c12e9f Adding states/win_system.py unit tests
- **PR** #23466: (UtahDave) minor spelling fix @ *2015-05-07 19:19:06 UTC*
  - e6e11147af Merge pull request #23466 from UtahDave/2015.5local
  - b2c399a137 minor spelling fix
- **ISSUE** #529: (rubic) run salt in user space (refs: #543)
  - **PR** saltstack/salt-bootstrap#563: (notpeter) Ubuntu alternate ppas (refs: #23461, #23460)

  - **PR** #543: (rubic) updated documentation for user, fixed configuration template links (refs: #‘saltstack/salt-bootstrap#563‘_)

- **PR** #23461: (s0undt3ch) [2015.5] Update to latest stable bootstrap script v2015.05.07 @ *2015-05-07 19:16:18 UTC*
  - 4eeb1e627a Merge pull request #23461 from s0undt3ch/hotfix/bootstrap-script
  - 638c63d635 Update to latest stable bootstrap script v2015.05.07
- **PR** #23450: (jayeshka) adding scsi unit test case @ *2015-05-07 19:00:28 UTC*
  - 865127844a Merge pull request #23450 from jayeshka/scsi-unit-test
  - e7269ff29b adding scsi unit test case
- **PR** #23449: (jayeshka) adding s3 unit test case @ *2015-05-07 18:59:45 UTC*
  - 8b374ae64d Merge pull request #23449 from jayeshka/s3-unit-test
  - 85786bfe7f adding s3 unit test case
- **PR** #23448: (jayeshka) adding states/keystone unit test case @ *2015-05-07 18:58:59 UTC*
  - 49b431c8e4 Merge pull request #23448 from jayeshka/keystone-states-unit-test
  - a3050eb3e2 adding states/keystone unit test case
- **PR** #23447: (jayeshka) adding states/grafana unit test case @ *2015-05-07 18:58:20 UTC*
  - 23d7e7ef92 Merge pull request #23447 from jayeshka/grafana-states-unit-test
  - 7e90a4aaca adding states/grafana unit test case
- **PR** #23438: (techhat) Gate requests import @ *2015-05-07 07:22:58 UTC*
  - 1fd0bc2011 Merge pull request #23438 from techhat/gaterequests
  - d5b15fc6ce Gate requests import
- **PR** #23429: (basepi) [2015.5] Merge forward from 2014.7 to 2015.5 @ *2015-05-07 05:35:13 UTC*
  - 3c4f734332 Merge pull request #23429 from basepi/merge-forward-2015.5
  - 7729834d92 Merge remote-tracking branch ’upstream/2014.7’ into merge-forward-2015.5
  - 644eb75fec Merge pull request #23422 from cro/gce_sh_home
    - 4ef9e6ba06 Don’t use $HOME to find user’s directory, some shells don’t set it
  - ef17ab4b2a Merge pull request #23425 from basepi/functionwrapper_typo
    - c390737f3e Fix typo in FunctionWrapper
  - 1b13ec04c2 Merge pull request #23385 from rallytime/bp-23346
    - 9efc13c810 more linting fixes
    - cf131c9a5a cleaned up some pylint errors
    - f981699c75 added logic to sftp_file and file_map to allow folder uploads using file_map
  - f8c7a62089 Merge pull request #23414 from jfindlay/update_branch
    - 8074d16d52 2015.2 -> 2015.5
  - 54b3bd43e4 Merge pull request #23404 from hvnsweeting/cherrypy-post-emptybody-fix
    - f85f8f954c initialize var when POST body is empty
  - 160f703296 Merge pull request #23409 from terminalmage/update-lithium-docstrings-2014.7

    - bc97d011ba Fix sphinx typo
    - 20006b06f6 Update Lithium docstrings in 2014.7 branch
  - aa5fb0aa46 Merge pull request #23397 from jfindlay/fix_locale_gen
    - 0941fefd2b add more flexible whitespace to locale_gen search
- **PR** #23396: (basepi) [2015.2] Merge forward from 2014.7 to 2015.2 @ *2015-05-06 21:42:35 UTC*
  - 1fb84450f4 Merge pull request #23396 from basepi/merge-forward-2015.2
  - 2766c8cb4b Fix typo in FunctionWrapper
  - fd09cdae6f Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.2
    - 0c76dd4d8a Merge pull request #23368 from kaithar/bp-23367
      - 577f41972e Pylint fix
      - 8d9acd1f89 Put the sed insert statement back in to the output.
    - 3493cc1fca Merge pull request #23350 from lorengordon/file.replace_assume_line
      - b60e224beb Append/prepend: search for full line
    - 7be5c48ad5 Merge pull request #23341 from cachedout/issue_23026
      - e98e65e787 Fix tests
      - 6011b437ca Fix syndic pid and logfile path
    - ea61abfa68 Merge pull request #23272 from basepi/salt-ssh.minion.config.19114
      - c223309bb7 Add versionadded
      - be7407feae Lint
      - c2c337567e Missing comma
      - 8e3e8e073a Pass the minion_opts through the FunctionWrapper
      - cb69cd07de Match the master config template in the master config reference
      - 87fc3161f9 Add Salt-SSH section to master config template
      - 91dd9dcbdc Add ssh_minion_opts to master config ref
      - c273ea14c6 Add minion config to salt-ssh doc
      - a0b6b760c3 Add minion_opts to roster docs
      - 5212c35260 Accept minion_opts from the target information
      - e2099b6e1b Process *ssh_minion_opts* from master config
      - 3b64214377 Revert "Work around bug in salt-ssh in config.get for gpg renderer"
      - 494953a208 Remove the strip (embracing multi-line YAML dump)
      - fe87f0fe39 Dump multi-line yaml into the SHIM
      - b751a7281c Inject local minion config into shim if available
    - 4f760dd9cb Merge pull request #23347 from basepi/salt-ssh.functionwrapper.contains.19114
      - 30595e3ff7 Backport FunctionWrapper.__contains__
    - 02658b1e60 Merge pull request #23344 from cachedout/issue_22742
      - 5adc96ce7f Explicitely set file_client on master

    - ba7605d1cb Merge pull request #23318 from cellscape/honor-seed-argument
      - 228b1be299 Honor seed argument in LXC container initializaton
    - 4ac4509c57 Merge pull request #23307 from jfindlay/fix_locale_gen
      - 101199ac14 check for /etc/locale.gen
    - f790f42ed6 Merge pull request #23324 from s0undt3ch/hotfix/bootstrap-script-2014.7
    - 6643e47ce5 Update to the latest stable release of the bootstrap script v2015.05.04
- **PR** #23412: (rahulhan) Adding states/win_update.py unit tests @ *2015-05-06 18:31:09 UTC*
  - b3c16720f6 Merge pull request #23412 from rahulhan/states_win_update_unit_test
  - 9bc1519ee7 Removed unwanted imports
  - f12bfcf248 Adding states/win_update.py unit tests
- **PR** #23413: (terminalmage) Update manpages for 2015.2 -> 2015.5 @ *2015-05-06 17:12:57 UTC*
  - f2d7646a58 Merge pull request #23413 from terminalmage/update-manpages
  - 23fa4402dc Update manpages to reflect 2015.2 rename to 2015.5
  - 0fdaa73c84 Fix missed docstring updates from 2015.2 -> 2015.5
  - 4fea5ba477 Add missing RST file
- **PR** #23410: (terminalmage) Update Lithium docstrings in 2015.2 branch @ *2015-05-06 15:53:52 UTC*
  - **PR** #23409: (terminalmage) Update Lithium docstrings in 2014.7 branch (refs: #23410)
  - bafbea7bc7 Merge pull request #23410 from terminalmage/update-lithium-docstrings-2015.2
  - d395565bf7 Update Lithium docstrings in 2015.2 branch
- **PR** #23407: (jayeshka) adding rsync unit test case @ *2015-05-06 15:52:23 UTC*
  - 02ef41a549 Merge pull request #23407 from jayeshka/rsync-unit-test
  - a4dd836125 adding rsync unit test case
- **PR** #23406: (jayeshka) adding states/lxc unit test case @ *2015-05-06 15:51:50 UTC*
  - 58ec2a24c1 Merge pull request #23406 from jayeshka/lxc-states-unit-test
  - 32a0d03093 adding states/lxc unit test case
- **PR** #23395: (basepi) [2015.2] Add note to 2015.2.0 release notes about master opts in pillar @ *2015-05-05 22:15:20 UTC*
  - 8837d0038e Merge pull request #23395 from basepi/2015.2.0masteropts
  - b261c95cd6 Add note to 2015.2.0 release notes about master opts in pillar
- **PR** #23393: (basepi) [2015.2] Add warning about python_shell changes to 2015.2.0 release notes @ *2015-05-05 22:12:46 UTC*
  - f79aed5fe1 Merge pull request #23393 from basepi/2015.2.0python_shell
  - b2f033f485 Add CLI note
  - 48e7b3ee4f Add warning about python_shell changes to 2015.2.0 release notes
- **PR** #23380: (gladiatr72) Fix for double output with static salt cli/v2015.2 @ *2015-05-05 21:44:28 UTC*
  - a9777761d8 Merge pull request #23380 from gladiatr72/fix_for_double_output_with_static__salt_CLI/v2015.2
  - c47fdd79c7 Actually removed the `static` bits from below the else: fold this time.

  - 4ee367956c Fix for incorrect output with salt CLI --static option
- **PR** #23379: (rahulhan) Adding states/rabbitmq_cluster.py @ *2015-05-05 21:44:06 UTC*
  - 5c9543c1d2 Merge pull request #23379 from rahulhan/states_rabbitmq_cluster_test
  - 04c22d1acf Adding states/rabbitmq_cluster.py
- **PR** #23377: (rahulhan) Adding states/xmpp.py unit tests @ *2015-05-05 21:43:35 UTC*
  - 430f080a3a Merge pull request #23377 from rahulhan/states_xmpp_test
  - 32923b53c3 Adding states/xmpp.py unit tests
- **PR** #23335: (steverweber) 2015.2: include doc in master config for module_dirs @ *2015-05-05 21:28:58 UTC*
  - 8c057e6794 Merge pull request #23335 from steverweber/2015.2
  - 5e3bae95d8 help installing python pysphere lib
  - 97513b060a include module_dirs
  - 36b1c87dd2 include module_dirs
- **PR** #23362: (jayeshka) adding states/zk_concurrency unit test case @ *2015-05-05 15:50:06 UTC*
  - 1648253675 Merge pull request #23362 from jayeshka/zk_concurrency-states-unit-test
  - f60dda4b1d adding states/zk_concurrency unit test case
- **PR** #23363: (jayeshka) adding riak unit test case @ *2015-05-05 14:23:05 UTC*
  - 1cdaeed868 Merge pull request #23363 from jayeshka/riak-unit-test
  - f9da6db459 adding riak unit test case

28.2.67 Salt 2015.5.10 Release Notes

release 2015-03-22

Version 2015.5.10 is a bugfix release for *2015.5.0*.

Security Fix

CVE-2016-3176 Insecure configuration of PAM external authentication service

This issue affects all Salt versions prior to 2015.8.8/2015.5.10 when PAM *external authentication* is enabled. This issue involves passing an alternative PAM authentication service with a command that is sent to *LocalClient*, enabling the attacker to bypass the configured authentication service. Thank you to Dylan Frese <dmfrese@gmail.com> for bringing this issue to our attention.

This update defines the PAM eAuth `service` that users authenticate against in the Salt Master configuration.

No additional fixes are included in this release.

Read Before Upgrading Debian 8 (Jessie) from Salt Versions Earlier than 2015.5.9

Salt `systemd` service files are missing the following statement in these versions:

```
[Service]
KillMode=process
```

This statement must be added to successfully upgrade on these earlier versions of Salt.

Changelog for v2015.5.9..v2015.5.10

Generated at: 2018-05-27 22:39:26 UTC

- 69ba1de71d Remove ability of authenticating user to specify pam service

28.2.68 Salt 2015.5.11 Release Notes

release 2015-07-22

Version 2015.5.11 is a bugfix release for *2015.5.0*.

Statistics

- Total Merges: **101**
- Total Issue References: **73**
- Total PR References: **162**
- Contributors: **46** (AndrewPashkin, Ch3LL, DmitryKuzmenko, TheNullByte, UtahDave, abednarik, amontalban, anlutro, attiasr, basepi, borgstrom, brejoc, bstevenson, cachedout, carlwgeorge, efficks, gerhardqux, gtmanfred, heyfife, jacobhammons, jfindlay, justinta, lomeroe, lorengordon, mtorromeo, nmadhok, notpeter, paclat, pcn, phistrom, rallytime, robgott, sacren, sastorsl, serge-p, sjmh, sjorge, techhat, terminalmage, thatch45, thegoodduke, toanju, tomwalsh, twangboy, whiteinge, yannis666)

Changelog for v2015.5.10..v2015.5.11

Generated at: 2018-05-27 22:41:56 UTC

- **PR** #33412: (jfindlay) update 2015.5.11 release notes
- **PR** #33405: (rallytime) Back-port #33386 to 2015.5
- **PR** #33386: (terminalmage) Fix traceback in logging for config validation (refs: #33405)
- **ISSUE** #33376: (tmehlinger) pip state broken in 2015.8.9 with pip <6.0 (refs: #33383)
- **PR** #33383: (thatch45) maintain the fallabck because I am totally sick of this crap @ *2016-05-20 00:03:59 UTC*
  - d15f5e2cef Merge pull request #33383 from thatch45/2015.5
  - f5ebcba21c restore whitespace
  - 1d8b289db1 blast, put the try/except int he right place
  - 081e6c5b83 maintain the fallabck because I am totally sick of this crap
  - **PR** #33379: (cachedout) Improve doc clarity for disable_modules documentation
- **ISSUE** #26574: (jfindlay) minion stacktrace on top file yaml syntax error (refs: #33375)
  - **PR** #33375: (cachedout) Better YAML syntax error handling
- **PR** #33372: (jacobhammons) revved 2015.8 branch to .9 in version selector @ *2016-05-19 20:05:35 UTC*
  - bb3e98cad2 Merge pull request #33372 from jacobhammons/release-update
  - 5ce502160b revved 2015.8 branch to .9 in version selector
  - **PR** #33341: (phistrom) Expanded documentation for boto_elb state and module

- **ISSUE** #32400: (rallytime) Document Default Config Values (refs: #33286, #33292, #32538, #33287, #32454, #33282)
  - **PR** #33292: (rallytime) Added some more docs for master and minion config settings
- **ISSUE** #23643: (falzm) Error in iptables module: argument –match-set: expected 2 argument(s) (refs: #33301)
  - **PR** #33301: (gerhardqux) Fix iptables –match-set (#23643)
  - **PR** #33290: (UtahDave) fix "loose" typo
- **ISSUE** #32400: (rallytime) Document Default Config Values (refs: #33286, #33292, #32538, #33287, #32454, #33282)
  - **PR** #33287: (rallytime) Add auth_tries config option to minion.rst docs
  - **PR** #33286: (rallytime) Document new master and minion config opts for 2016.3.0 (refs: #33287)
- **ISSUE** #33276: (sjmh) minion_id_caching has no documentation (refs: #33282)
- **ISSUE** #32400: (rallytime) Document Default Config Values (refs: #33286, #33292, #32538, #33287, #32454, #33282)
  - **PR** #33282: (rallytime) Document minion_id_caching config value
- **ISSUE** #33118: (saltuser) file.replace not working correctly on newer minions (refs: #33137)
  - **PR** #33137: (lorengordon) Clarify file.replace MULTILINE flag interaction with regex anchors
  - **PR** #33236: (jfindlay) update 2015.5.11 release notes
- **ISSUE** #32250: (ikryten) Cannot run salt-minion as unprivileged user using 'user' directive (refs: #33211)
- **PR** #33211: (cachedout) Don't try to kill a parent proc if we can't @ *2016-05-12 21:29:50 UTC*
  - 698f1eb657 Merge pull request #33211 from cachedout/user_kill
  - d4f2e5baa7 Don't try to kill a parent proc if we can't
- **ISSUE** #32198: (goatjam) State 'pkg.installed' was not found in SLS (refs: #33205)
  - **PR** #33205: (cachedout) Resolve issue with pkg module on Mint Linux
  - **PR** #33178: (justinta) Add pip installed and removed test
  - **PR** #33197: (jfindlay) update 2015.5.11 release notes
  - **PR** #33181: (twangboy) Fix file.managed for Windows
  - **PR** #33185: (rallytime) [2015.5] Update to latest bootstrap script v2016.05.11
- **ISSUE** #33163: (jaybocc2) Salt 2015.8.5 incompatible with Pip v8.1.2 (refs: #33180)
  - **PR** #33180: (thatch45) Pip fix
  - **PR** #33160: (jfindlay) add 2015.5.11 release notes
  - **PR** #33155: (rallytime) [2015.5] Update to latest bootstrap script v2016.05.10
- **PR** #33141: (justinta) Skipping salt-call –local test @ *2016-05-10 17:05:17 UTC*
  - 6cd1641840 Merge pull request #33141 from jtand/disable_local_pkg_install_test
  - 8b1e34fb17 Skipping salt-call –local test
- **ISSUE** #33085: (fmnisme) salt doc err (refs: #33132)
  - **PR** #33132: (whiteinge) Doc mock decorators
- **ISSUE** #33074: (robnagler) Critical error in msgpack exposes pillar data (refs: #33078)

  - **PR** #33078: (cachedout) Lower display of msgpack failure msg to debug
  - **PR** #33080: (justinta) Use saltstack repo in buildpackage.py on CentOS 5
  - **PR** #33025: (Ch3LL) add test for installing package while using salt-call –local
  - **PR** #33055: (justinta) File and User test fixes for 2015.5 on Fedora23
  - **PR** #33060: (Ch3LL) Test pillar.items output
  - **PR** #33067: (sacren) Fix minor document error of test.assertion
  - **PR** #33045: (Ch3LL) Saltfile with pillar tests
  - **PR** #33044: (thatch45) Backport #33021 manually to 2015.5
  - **PR** #33021: (UtahDave) Fix syndic regression (refs: #33044)
- **ISSUE** #22580: (ryanwalder) minion runs highstate on start if schedule set in pillar (refs: #32958)
  - **PR** #32958: (rallytime) Add run_on_start docs to schedule.rst
- **ISSUE** #23714: (naemono) file.copy force ignored during highstate, but not with 'salt-call state.sls_id' (refs: #32732, #32848)
  - **PR** #32848: (lomeroe) backport PR #32732 to 2015.5 fixes #23714
  - **PR** #32732: (lomeroe) correct use of force flag in file.copy #23714 (refs: #32848)
  - **PR** #32837: (jfindlay) *salt-cloud -u* downloads stable version from bootstrap.saltstack.com by default
  - **PR** #32667: (jfindlay) [2015.5] update bootstrap to 2016.04.18 release
- **PR** #32776: (rallytime) [2015.5] Merge forward from 2014.7 to 2015.5 @ *2016-04-25 15:18:12 UTC*
  - c842e1e437 Merge pull request #32776 from rallytime/merge-2015.5
  - 7ecbf9f885 Merge pull request #14 from whiteinge/runner-async-low
    - 211f7b4af1 Format low data correct for runner_async
  - ce72851861 Merge branch '2014.7' into '2015.5'
  - 2775edc176 Saltnado /run fix (#32590)
  - b19c5a5ce7 Verify auth in saltnado run (#32552)
  - **PR** #32691: (terminalmage) Support remote sources in a source list
- **ISSUE** #32661: (dergrunepunkt) Batch exception w/dulpicated minion IDs (refs: #32686)
- **PR** #32686: (cachedout) Fix stacktrace in batch with dup minion ids @ *2016-04-19 19:18:50 UTC*
  - bd5442d768 Merge pull request #32686 from cachedout/issue_32661
  - f704df90bc Fix stacktrace in batch with dup minion ids
  - **PR** #32675: (basepi) [2015.5] Update "Low Hanging Fruit" to "Help Wanted"
- **ISSUE** #32612: (oliver-dungey) Calling Salt Modules from Templates - more complex examples would be great (refs: #32657)
  - **PR** #32657: (cachedout) Additional documentation on calling exec modules from templates
  - **PR** #32639: (nmadhok) [2015.5] - Fixing critical bug to remove only the specified Host instead of the entire Host cluster
  - **PR** #32638: (nmadhok) [2015.5] Adding _syspaths.py to .gitignore
- **ISSUE** #32381: (tbaker57) user.present state includes shadow hash in return when user updated (refs: #32561)

- **PR** #32561: (gtmanfred) redact passwords and hashes from user.present updates @ *2016-04-14 15:48:59 UTC*
  - 027b502335 Merge pull request #32561 from gtmanfred/user_passwords
  - 3db5e78d5d redact passwords and hashes from user.present updates
- **ISSUE** #32400: (rallytime) Document Default Config Values (refs: #33286, #33292, #32538, #33287, #32454, #33282)
- **PR** #32538: (rallytime) Back-port #32528 to 2015.5 @ *2016-04-13 15:06:14 UTC*
  - **PR** #32528: (AndrewPashkin) Document "grains" setting in the minion configuration reference (refs: #32538)
  - 7307bcb88e Merge pull request #32538 from rallytime/bp-32528
  - 46a4e8a310 Remove merge conflict line
  - e0d947c707 Document "grains" setting in the minion configuration reference
- **ISSUE** #32400: (rallytime) Document Default Config Values (refs: #33286, #33292, #32538, #33287, #32454, #33282)
  - **PR** #32454: (rallytime) Add documentation for some master/minion configs
- **ISSUE** #32413: (commutecat) Raspbian detected by both systemd.py and service.py __virtual__ functions (refs: #32421, #32458)
- **PR** #32458: (terminalmage) Improve and clarify docs on provider overrides. @ *2016-04-09 14:25:42 UTC*
  - 100c6e1b25 Merge pull request #32458 from terminalmage/clarify-providers-docs
  - 500d3ebbaa Add link to provider override docs to all group providers
  - 83ca01f620 dd link to provider override docs to all shadow providers
  - c5fe38789d Add link to provider override docs to all user providers
  - 5c1c1dda59 Add link to provider override docs to all service providers
  - 736f2befc9 Add link to provider override docs to all package providers
  - f9306347cc Clarify the scope of the provider param in states.
  - af24c82ab0 Add documentation on virtual module provider overrides to the module docs
  - 0bc6c97a63 Improve docstrings
  - 1948920674 Add external ref to windows package manager docs
  - e7fa21438c Add new doc pages to toctree
  - f0de1236ec Move the tables of virtual modules to individual documentation pages
- **ISSUE** #11497: (eeaston) cmd.run cwd should not be checked before preconditions (refs: #32293)
  - **PR** #32418: (rallytime) Merge #32293 with test fixes
  - **PR** #32293: (efficks) Fix issue #11497 (refs: #32418)
- **ISSUE** #32413: (commutecat) Raspbian detected by both systemd.py and service.py __virtual__ functions (refs: #32421, #32458)
  - **PR** #32421: (terminalmage) Ignore Raspbian in service.py __virtual__
- **ISSUE** #1409: (twinshadow) module/network.py: Interfaces do not list multiple addesses
- **ISSUE** saltstack/salt#28262: (palica) FreeBSD pkgng provider raising error for minion (refs: #32376)
- **ISSUE** #28262: (palica) FreeBSD pkgng provider raising error for minion (refs: #32399, #32376)

- **PR** #32399: (amontalban) Backport to fix #28262 for 2015.5 as requested in PR #32376 @ *2016-04-06 22:48:23 UTC*
  - **PR** #32376: (amontalban) Fixes saltstack/salt#28262 (refs: #32399)
  - a36866d7db Merge pull request #32399 from amontalban/2015.5
  - e1ffbd615a Fixes saltstack/salt#28262 for 2015.5 branch
- **ISSUE** #32066: (guettli) Proxmox docs outdated (refs: #32374)
- **PR** #32374: (cachedout) Update proxmox documentation @ *2016-04-05 22:25:16 UTC*
  - 3f03c5fcf9 Merge pull request #32374 from cachedout/issue_32066
  - 62389d1d1a Update proxmox documentation
- **PR** #32339: (Ch3LL) remove reference to master_alive_check in 2015.5 @ *2016-04-04 20:39:24 UTC*
  - 8578089beb Merge pull request #32339 from Ch3LL/fix_doc_multi-master
  - 2774da288d remove reference to master_alive_check
- **ISSUE** #32044: (ScoreUnder) Multiple masters throwing warnings? "Key master with value [...] has an invalid type of list, a str is required for this value" (refs: #32129)
- **PR** #32284: (rallytime) Audit config.py default types and values @ *2016-04-02 02:00:38 UTC*
  - **PR** #32129: (terminalmage) Support multiple valid option types when performing type checks (refs: #32284)
  - fbdc47cc55 Merge pull request #32284 from rallytime/config-audit
  - 0491513204 Don't be so explicit. Just use string_types.
  - 083c477fd3 Use six.string_types in config default tuples
  - 7e642b8381 Audit config.py default types and values - first sweep
- **ISSUE** #32301: (terminalmage) pkg.latest_version returns inaccurate version when blank "Release" param set in package metadata (refs: #32302)
- **PR** #32302: (terminalmage) Properly support packages with blank "Release" param in pkg.latest_version @ *2016-04-01 22:13:27 UTC*
  - 0a6d44e57b Merge pull request #32302 from terminalmage/fix-missing-release
  - 413c371ccd Properly support packages with blank "Release" param in pkg.latest_version
- **ISSUE** #31963: (UtahDave) pkgrepo.managed state test=True doesn't actually test if changes need to be made. (refs: #32162)
- **PR** #32162: (terminalmage) Properly handle yum/zypper repositories in pkgrepo.managed @ *2016-03-30 17:51:05 UTC*
  - 5d08db7c92 Merge pull request #32162 from terminalmage/issue31963
  - 5c1bdb812c Fix pkgrepo integration test
  - e7fb3095ce Properly handle yum/zypper repositories in pkgrepo.managed
  - add2111fec Use six.iteritems instead of dict.items
  - 6c21881c38 Docstring tweaks
  - ecbb78b649 Remove useless function
  - 06f3309552 Normalize variable naming to match other functions

  - 690537ca8b Look for apt-add-repository in PATH instead of assuming it's there
  - 709d80bb1b aptpkg: Accept **kwargs instead of a dict for pkg.expand_repo_def
- **ISSUE** #31976: (moltob) Schedules not persisted on Windows minion (Installer issue) (refs: #32223)
- **PR** #32223: (twangboy) Create minion.d directory on install for Windows @ *2016-03-30 14:43:27 UTC*
  - 4fcdaab428 Merge pull request #32223 from twangboy/fix_31976
  - b7fcae97ce Create minion.d directory, fixes #31976
- **ISSUE** #31501: (grep4linux) Salt states fail with error 'Failed to return clean data' when using salt-ssh in Amazon EC2 (refs: #32218)
- **PR** #32218: (cachedout) Only display error when tty is True in salt-ssh @ *2016-03-29 19:13:44 UTC*
  - 3309ff6a29 Merge pull request #32218 from cachedout/issue_31501
  - 6795d6aef0 Only display error when tty is True in salt-ssh
- **PR** #32196: (justinta) Fixed pylint error in app_pam_test.py @ *2016-03-28 23:59:42 UTC*
  - 6e0cb22c96 Merge pull request #32196 from jtand/cherrypy_pam_test_lint_fix
  - bd3942e0fd Fixed pylint error in app_pam_test.py
- **PR** #32154: (Ch3LL) Add integration tests for salt-api using pam eauth @ *2016-03-28 16:06:36 UTC*
  - **PR** #31826: (gtmanfred) Remove ability of authenticating user to specify pam service (refs: #32154)
  - 6b8b8b51c0 Merge pull request #32154 from Ch3LL/ch3ll_pam_2015.5
  - ba605b0128 fix more pylint and add ability to close cherrypy engine
  - 2d4dc4da05 add teardown call
  - d115878714 fix pylint error
  - 4c1ab082b6 add pam salt-api tests
- **PR** #32170: (gtmanfred) add name for lxc for use with cloud cache @ *2016-03-28 14:34:16 UTC*
  - 230443be6c Merge pull request #32170 from gtmanfred/lxc_cloud_name
  - eb7d82e7be add name for lxc for use with cloud cache
- **ISSUE** #31731: (sjorge) rh_service references osrelease before it is available, also does not return bool (refs: #32165)
  - **PR** #32165: (terminalmage) Make __virtual__ for rhservice.py more robust (refs: #32164)
- **PR** #32164: (terminalmage) Make __virtual__ for rhservice.py more robust (2015.5 branch) (refs: #32165) @ *2016-03-27 18:21:52 UTC*
  - 32b0421a34 Merge pull request #32164 from terminalmage/issue31731-2015.5
  - 18439c4f89 Make __virtual__ for rhservice.py more robust (2015.5 branch)
- **PR** #32141: (paclat) fixes 32108 @ *2016-03-25 16:50:59 UTC*
  - 6212e9aa56 Merge pull request #32141 from paclat/issue_32108
  - 72c5d12d43 fixes 32108
- **ISSUE** #32044: (ScoreUnder) Multiple masters throwing warnings? "Key master with value [...] has an invalid type of list, a str is required for this value" (refs: #32129)
- **PR** #32129: (terminalmage) Support multiple valid option types when performing type checks (refs: #32284) @ *2016-03-24 21:16:29 UTC*

  - bdd7ea89d5 Merge pull request #32129 from terminalmage/issue32044
  - 34ca1ea12e Change type check errors to debug loglevel
  - 5462081488 Support multiple valid option types when performing type checks
- **ISSUE** #32052: (bstevenson) list_absent function doesn't loop through list of values (refs: #32056)
- **PR** #32056: (bstevenson) Fix list absent @ *2016-03-24 17:35:00 UTC*
  - c42014eb54 Merge pull request #32056 from bstevenson/fix-list_absent
  - 1500aae027 set deleted value to list
  - 1dc8f5f289 unit test update
  - 39adf86fec Fixed negation logic
  - be9388173b Removed has_key in lieu of in
  - e48593ed81 Comments and Changes output fixes
  - b98f5517de Updated to conform to proper ret values
  - d18b4be80b remove whitespace end of line 186:q
  - d2b89c85ad fix formating
  - 103cee9e29 cleaned up formating
  - 7a4d7f0bff added whitespace
  - 8ea5b545b0 Loop through list values in list_absent
- **PR** #32096: (rallytime) Back-port #32065 to 2015.5 @ *2016-03-23 22:01:36 UTC*
  - **PR** #32065: (TheNullByte) Fix an issue with the minion targeting example in docs (refs: #32096)
  - 848ce5647f Merge pull request #32096 from rallytime/bp-32065
  - 36a9d6a374 Fix an issue with the minion targeting example
- **PR** #32104: (jacobhammons) One additional known issue for 2015.5.10 release notes @ *2016-03-23 21:20:50 UTC*
  - 9b332d48b9 Merge pull request #32104 from jacobhammons/dot10
  - b9fc882a1e One additional known issue for 2015.5.10 release notes
- **PR** #32100: (jacobhammons) 2015.5.10 release docs @ *2016-03-23 20:05:21 UTC*
  - ff51d548e1 Merge pull request #32100 from jacobhammons/dot10
  - 544a1661ce 2015.5.10 release docs
- **ISSUE** #32037: (terminalmage) Increase the visibility of state.apply in Salt's documentation (refs: #32038)
- **PR** #32038: (terminalmage) Improve state module docs, replace references to state.highstate/state.sls with state.apply @ *2016-03-23 17:08:02 UTC*
  - 72a20f9799 Merge pull request #32038 from terminalmage/issue32037
  - 8b2d983324 Add reference to state tutorial to state.apply docstring
  - 9b4fe8443e Move highstate usage details to top of state.apply docstring
  - 74ee8c54bc Clarify prior role of state.highstate in states tutorial
  - 1b97e4a3df Improve state module docs, replace references to state.highstate/state.sls with state.apply
- **PR** #32051: (terminalmage) Fix outputter for state.apply @ *2016-03-23 16:42:43 UTC*

  - 908a7bf5cd Merge pull request #32051 from terminalmage/fix-state-apply-output
  - 7d7cb45565 Fix outputter for state.apply
- **ISSUE** #31788: (crocket) pkg.installed doesn't work on Manjaro. (refs: #32002)
- **PR** #32002: (abednarik) Added Manajro Linux to virtual. @ *2016-03-21 17:55:16 UTC*
  - 0e66f678d4 Merge pull request #32002 from abednarik/pkg_manjaron_issue31788
  - 1b052d0a66 Added Manajro Linux to virtual. List extended with ManajaroLinux in order su load pacman module.
- **PR** #31957: (rallytime) [2015.5] Merge forward from 2014.7 to 2015.5 @ *2016-03-18 15:12:22 UTC*
  - ba5bf62c1a Merge pull request #31957 from rallytime/merge-2015.5
  - 1b6ec5d445 Merge branch '2014.7' into '2015.5'
    - ba73deee46 Merge pull request #31929 from twangboy/fix_build_script
      - 2c5599d2bc Backport build script from 2015.8
      - ce74991dd0 Fix nsi script to work with new build process
- **PR** #31972: (terminalmage) Make lack of python-ldap module more explicit when LDAP eauth is enabled @ *2016-03-18 15:11:59 UTC*
  - a52e3ad7a1 Merge pull request #31972 from terminalmage/zh-584
  - 1e5639e495 Make lack of python-ldap module more explicit when LDAP eauth is enabled
- **PR** #31935: (twangboy) Back port nullsoft build script from 2015.8 @ *2016-03-17 14:54:50 UTC*
  - 2d1f2a0c2e Merge pull request #31935 from twangboy/fix_build_script2
  - 4af8c9dbfc Back port nullsoft build script from 2015.8
- **PR** #31912: (jfindlay) log.mixins: remove extermporaneous .record @ *2016-03-16 01:56:46 UTC*
  - 43240dc566 Merge pull request #31912 from jfindlay/log_mixin
  - 9f9c694654 log.mixins: remove extermporaneous .record
- **PR** #31825: (justinta) Updated .testing.pylintrc to match newer versions of pylint @ *2016-03-15 18:12:44 UTC*
  - 440e0dcbe0 Merge pull request #31825 from jtand/udpate_pylintrc
  - 9a14e02766 Updated beacons/sh.py to work with enumerate()
  - 0ecec691a0 Adjusted beacons to work with enumerate better
  - f509b4113e Fixed final lint error
  - 5945b3f11f Fix and disable pylint errors
  - 06ae6eaf55 Fixed pylint errors on jboss state and module
  - de96db97c8 Fixed more pylint errors, and disabled some more
  - c07b0a20b5 Merge branch 'lint_fixes' into udpate_pylintrc
    - 2e6a152308 Fixed lint error in lxc.py
    - 908ca1a439 Fixed lint error in ssh_py_shim
    - 404c1b50f7 Changed range(len()) to enumerate()
    - 1e13586546 Changed range(len()) to enumerate()
  - 9ccce7a9a5 Added more disables

  - 9c1aab3b4e Updated .testing.pylintrc to match newer versions of pylint
- **ISSUE** #31867: (damon-atkins) " __virtual__ returned False" is not a clear error message (refs: #31878, #31900)
- **PR** #31900: (rallytime) Add "python module" clarification to ps __virtual__ warning. @ *2016-03-15 17:59:35 UTC*
  - 471c9444a3 Merge pull request #31900 from rallytime/fix-psutil-warning
  - 22403d69ae Add "python module" clarification to ps __virtual__ warning.
- **ISSUE** #31867: (damon-atkins) " __virtual__ returned False" is not a clear error message (refs: #31878, #31900)
- **ISSUE** #19659: (wonderslug) state process.absent is failing on Ubuntu 14.04 because psutil is not installed (refs: #31878)
- **PR** #31878: (rallytime) Make sure __virtual__ error message is helpful when psutil is missing @ *2016-03-14 21:31:42 UTC*
  - c44c1b5e59 Merge pull request #31878 from rallytime/fix-psutil-warning
  - 44b29f72a1 Make sure __virtual__ error message is helpful when psutil is missing
- **PR** #31852: (rallytime) [2015.5] Merge forward from 2014.7 to 2015.5 @ *2016-03-13 02:47:02 UTC*
  - 5c592b6768 Merge pull request #31852 from rallytime/merge-2015.5
  - 1470de17fa Merge branch '2014.7' into '2015.5'
  - 218c902091 Merge pull request #31834 from jfindlay/2014.7
    - 358fdad0c8 add 2014.7.8 release notes
  - a423c6cd04 Merge pull request #31833 from jfindlay/2014.7
    - 6910fcc584 add 2014.7.9 release notes
  - c5e7c03953 Merge pull request #31826 from gtmanfred/2014.7
  - d73f70ebb2 Remove ability of authenticating user to specify pam service
- **PR** #31827: (gtmanfred) Remove ability of authenticating user to specify pam service @ *2016-03-11 20:40:19 UTC*
  - 0cc1d5db03 Merge pull request #31827 from gtmanfred/2015.5
  - 979173b78a Remove ability of authenticating user to specify pam service
- **PR** #31810: (whiteinge) Fix outdated Jinja 'env' variable reference @ *2016-03-11 03:52:21 UTC*
  - 8cf0b9eb3d Merge pull request #31810 from whiteinge/saltenv-jinja-var
  - cb72b19240 Fix outdated Jinja 'env' variable reference
- **ISSUE** #31729: (brejoc) Creating VM with salt-cloud fails for provider Exoscale (Cloudstack) (refs: #31744)
- **PR** #31744: (brejoc) Fix for AttributeError with libcloud <0.15 @ *2016-03-10 00:15:26 UTC*
  - 970ef0e445 Merge pull request #31744 from brejoc/fix-attribute-error-with-older-libcloud/2015.5
  - bb29dc2283 Added version to libcloud depends statement
  - 87f9534fce Added log message with update suggestion for libcloud
  - 72eab406cd Fix for AttributeError with libcloud <0.15
- **ISSUE** #31666: (sjorge) salt-call –local pillar.items is overly eager to give data (refs: #31740)
- **PR** #31740: (terminalmage) Assume pillar_opts is False when not specified in masterless mode @ *2016-03-09 22:57:57 UTC*

  - df2d23ba5d Merge pull request #31740 from terminalmage/issue31666
  - aeaf5864cd Fall back to False when pillar_opts not set
  - fe19d77eb4 Add default value for pillar_opts on minion
- **ISSUE** #31749: (milan-milo) salt-cloud spitting out error 'AttributeError: 'NoneType' object has no attribute 'pop'' (refs: #31750)
- **ISSUE** #26162: (nmadhok) VMware cloud driver create function failing with traceback on latest develop (refs: #26170)
- **PR** #31750: (rallytime) Back-port #26170 to 2015.5 @ *2016-03-09 17:44:14 UTC*
  - **PR** #26170: (nmadhok) [Backport] Make sure variable is a dictionary before popping something from it. (refs: #31750)
  - e22f5c0a26 Merge pull request #31750 from rallytime/bp-26170
  - 3c11234a05 Make sure variable is a dictionary before popping something from it.
- **ISSUE** #30559: (kaidokert) module.wait does not fail when called state fails (refs: #31689)
- **PR** #31689: (rallytime) Back-port #29467 to 2015.5 @ *2016-03-06 19:26:11 UTC*
  - **PR** #29467: (serge-p) Update module.py (refs: #31689)
  - 9162925dd0 Merge pull request #31689 from rallytime/bp-29467
  - 1f8f4cb99b Update module.py
- **PR** #31687: (cachedout) Removed useless GPG tests @ *2016-03-05 00:08:27 UTC*
  - d7914cdb14 Merge pull request #31687 from cachedout/rm_gpg_test
  - 8b00513ebb Removed useless tests
- **ISSUE** #31619: (alexxannar) 2015.8.7 pkg.installed problem with version parameter (refs: #31660)
- **PR** #31660: (terminalmage) Remove epoch from version string if present when installing with yum @ *2016-03-04 20:49:23 UTC*
  - bd4d12a155 Merge pull request #31660 from terminalmage/issue31619
  - da954d7b92 Add integration test for packages with epoch in version
  - 4fa7e4defe Move epoch removal
  - 290192af56 Remove epoch from version string if present when installing with yum
- **PR** #31683: (rallytime) Back-port #31578 to 2015.5 @ *2016-03-04 20:47:41 UTC*
  - **PR** #31578: (anlutro) Allow queueing of state runs through saltmod (refs: #31683)
  - e33c1f456a Merge pull request #31683 from rallytime/bp-31578
  - 8fe46789b7 allow queueing of state runs through saltmod
- **ISSUE** #31671: (guettli) Word "Job Cache" does not match (refs: #31682)
- **PR** #31682: (cachedout) Add definition of job cache to glossary @ *2016-03-04 20:07:19 UTC*
  - 27f443895d Merge pull request #31682 from cachedout/cache_meaning
  - a75e146125 Add definition of job cache to glossary
- **PR** #31658: (rallytime) Add mentioned of Salt's Coding Style docs to the Contributing docs @ *2016-03-03 22:14:57 UTC*
  - bd04c964d1 Merge pull request #31658 from rallytime/add-style-to-contrib

  - 6b526b5878 Add mentioned of Salt's Coding Style docs to the Contributing docs
- **ISSUE** #21932: (clinta) Salt Coding Style docs should list requirements for salt pylintrc (refs: #31655)
- **PR** #31655: (rallytime) Make note of pylint dependencies in docs @ *2016-03-03 18:37:06 UTC*
  - 10658dffe6 Merge pull request #31655 from rallytime/pylint-docs
  - 6e0377d376 Make note of pylint dependencies in docs
- **PR** #31440: (cachedout) Set correct type for master_tops config value @ *2016-03-02 21:17:14 UTC*
  - 6075774a01 Merge pull request #31440 from cachedout/master_tops_type
  - f49cc75049 Set correct type for master_tops config value
- **ISSUE** #31614: (frizzby) salt.utils.http.query() implementation contradicts it's documentation. decode arg (refs: #31622)
- **PR** #31622: (jfindlay) doc/topics/tutorials/http: update query decoding docs @ *2016-03-02 18:23:44 UTC*
  - 6d31b8918f Merge pull request #31622 from jfindlay/query_doc
  - 4e48fec806 doc/topics/tutorials/http: update query decoding docs
- **PR** #31558: (cachedout) Don't stacktrace if ssh binary is not installed with salt-ssh @ *2016-02-29 22:15:44 UTC*
  - dbf6e0786c Merge pull request #31558 from cachedout/ensure_ssh_installed
  - cecc6e0a5f Don't stacktrace if ssh binary is not installed with salt-ssh
- **PR** #31521: (terminalmage) salt-ssh: Fix race condition when caching files to build the thin tarball @ *2016-02-29 15:32:22 UTC*
  - 060a60fd90 Merge pull request #31521 from terminalmage/issue24753
  - 0d352bbc16 Add fileclient tests
  - d9370a8041 Update cp module salt-ssh wrapper to use new cachedir param
  - 0320494b1d Update the SSH state module wrappers to pass an alternate cachedir
  - 65bdcb3afa Accept and pass through the alternate cachedir when prepping the thin tar
  - c3f7a2f2e5 Add ability to specify an alternate base dir for file caching
- **PR** #31497: (rallytime) Remove duplicate "timeout" definition in Roster docs @ *2016-02-26 15:01:30 UTC*
  - 92f8f89218 Merge pull request #31497 from rallytime/remove-timeout-dup
  - 83e6480d20 Remove duplicate "timeout" definition in Roster docs
- **PR** #31472: (rallytime) Update contributing docs @ *2016-02-25 16:05:59 UTC*
  - da001bcb49 Merge pull request #31472 from rallytime/update-contributing-docs
  - 5871e4d1e0 Update contributing docs
- **ISSUE** #30183: (jakehilton) Minion startup extremely delayed when first master in failover multi master setup is down (refs: #31382)
- **PR** #31461: (DmitryKuzmenko) Set auth retry count to 0 if multimaster mode is failover. @ *2016-02-24 17:15:30 UTC*
  - **PR** #31382: (DmitryKuzmenko) Set auth retry count to 0 if multimaster mode is failover (refs: #31461)
  - f35e2dd1d3 Merge pull request #31461 from DSRCompany/issues/30183_fix_multimaster_failover_2015.5
  - 3d09c3b7a3 Set auth retry count to 0 if multimaster mode is failover.

- **ISSUE** #31356: (sastorsl) file.copy module with recurse=true and non-existing src dir does not fail and resets dst dir permissions (refs: #31442)
- **PR** #31442: (sastorsl) Add os.path.exists(src) to file.py, def copy @ *2016-02-23 23:40:03 UTC*
  - 26733ce988 Merge pull request #31442 from sastorsl/salt-modules-file.py-copy-check-src
  - 0a4132866d removed lint in the exception string
  - f8b5d498c3 Add os.path.exists(src) to file.py, def copy
- **ISSUE** #30739: (paclat) manage.present does not work when minion is using localhost (refs: #31441)
- **PR** #31441: (cachedout) Include localhost minions in presence detection for runner @ *2016-02-23 23:36:59 UTC*
  - e480727d27 Merge pull request #31441 from cachedout/issue_30739
  - ffcfad1570 Include localhost minions in presence detection for runner
- **PR** #31416: (carlwgeorge) selinux module documentation fix @ *2016-02-22 21:49:28 UTC*
  - 91ff95f093 Merge pull request #31416 from carlwgeorge/selinux_doc_fix
  - 0e6846d72e selinux module documentation fix
- **PR** #31336: (terminalmage) Improve config validation logging @ *2016-02-22 19:34:24 UTC*
  - 7d01979898 Merge pull request #31336 from terminalmage/config-validation-logging
  - 795008bad1 Improve config validation logging
- **ISSUE** #31369: (sjorge) illumos/solaris/smartos display compacted hwaddrs (refs: #31374)
- **PR** #31374: (sjorge) fix for #31369 @ *2016-02-22 16:22:21 UTC*
  - fed096a29d Merge pull request #31374 from sjorge/solarish_hwaddr
  - bdf2576dfb missed a .format and messed up the join
  - bbd2fdc96d fix for illumos/solaris hwaddr
- **PR** #31339: (jacobhammons) changed latest release to 2015.8.7 @ *2016-02-19 00:30:24 UTC*
  - 6ee17f905b Merge pull request #31339 from jacobhammons/dot7prev
  - 07120a8d48 changed latest release to 2015.8.7
- **PR** #31288: (notpeter) Improve salt.states.ssh_known_hosts documentation. @ *2016-02-17 22:09:18 UTC*
  - cd3400e67e Merge pull request #31288 from notpeter/ssh_known_hosts_docs
  - 3f573d89a2 Improve salt.states.ssh_known_hosts documentation.
- **PR** #31183: (heyfife) Fixed named external_ip reservation/re-use code in gce driver. @ *2016-02-17 19:02:27 UTC*
  - 875d9925fa Merge pull request #31183 from heyfife/fix-gce-named-static-ip-reservation
  - 26774e2323 Fixed named external_ip reservation/re-use code.
- **ISSUE** #31001: (toanju) Fedora 23 check installed packages fails (refs: #31032)
- **PR** #31032: (terminalmage) (2015.5 branch) yumpkg: ensure that dnf-plugins-core >= 0.1.15 is installed @ *2016-02-17 19:02:03 UTC*
  - e56c402c0c Merge pull request #31032 from terminalmage/issue31001
  - 42daea4509 yumpkg.py: Remove repoquery usage everywhere but check_db
  - 50befbc149 backport salt.utils.pkg.rpm to 2015.5
  - a1ad14994a Move salt.utils.itersplit() to salt.utils.itertools.split()

  - 5b8646ce64 Ignore failure to install new enough dnf-plugins-core
  - defe0859fd Ensure that dnf-plugins-core 0.1.15 is installed
- **ISSUE** #31174: (sjorge) salt.states.archive.extacted displays incorrect message: (refs: #31176)
- **PR** #31264: (sjorge) fix if_missing gets appended to dirs list, take III @ *2016-02-17 17:12:25 UTC*
  - **PR** #31250: (sjorge) if_missing append to array as far back as 2014.1 (refs: #31264)
  - **PR** #31176: (sjorge) if_missing incorrected appended to directories_created (refs: #31250, #31264)
  - cec69b74f0 Merge pull request #31264 from sjorge/if_missing-155-fix
  - 545edbf5e1 fix if_missing gets appended to dirs list, take III
- **PR** #31110: (cachedout) Fixup 30730 @ *2016-02-10 21:37:55 UTC*
  - fa3f474de9 Merge pull request #31110 from cachedout/fixup_30730
  - 5bf5848e04 Fixup unit test
  - f558f68e0a Fixes pylint warnings
  - 56a975ec43 Attempt to fix pylint warnings
  - 55d71be057 Make documentation and code examples consistent with code
  - 1f04fed6f8 Change parameter name from includes to skips
  - ccf5e13e7d Adding support for skipHidden in SetInclude
  - 4f2d4af2e7 Variable names standardization
  - f5917ac1e8 Fixes typo
  - 26e5236073 Invert RebootRequired logic
  - 8065a7abf6 Add basic documentation and define how the skips parameter works.
  - 389fea7508 Change parameter name from includes to skips
  - 30e1fef906 Adding support for skipHidden in SetInclude
  - 1244eea5be Variable names standardization, consistent if/else logic with states.win_update
- **ISSUE** #30900: (mchugh19) modules/qemu_nbd.py assumes versions of utilities that don't exist on ubuntu (refs: #30949)
- **PR** #30974: (rallytime) Back-port #30949 to 2015.5 @ *2016-02-08 16:38:46 UTC*
  - **PR** #30949: (techhat) Replace cfdisk with sfdisk (refs: #30974)
  - 1c699a1664 Merge pull request #30974 from rallytime/bp-30949
  - ff6542f593 Replace cfdisk with sfdisk
- **ISSUE** #28951: (ClaudiuPID) CloudLinux 7 changes (refs: #30897)
- **PR** #30942: (rallytime) Back-port #30897 to 2015.5 @ *2016-02-05 19:00:55 UTC*
  - **PR** #30897: (mtorromeo) Only remove the word linux from distroname when its not part of the name (refs: #30942)
  - c7f87cc371 Merge pull request #30942 from rallytime/bp-30897
  - 885e00ba54 Only remove the word linux from distroname when its not part of the name
- **PR** #30922: (jacobhammons) Rev latest version to 2015.8.5 @ *2016-02-05 01:20:27 UTC*
  - 35b7f62669 Merge pull request #30922 from jacobhammons/prev-rel-notes

  - 57c1ec637a Rev latest version to 2015.8.5
- **ISSUE** #30840: (HeathNaylor) Generic Error for SALT.STATES.BOTO_ELB (refs: #30865)
- **PR** #30865: (abednarik) Better boto elb error message. @ *2016-02-04 21:02:05 UTC*
  - 2488bb902e Merge pull request #30865 from abednarik/better_boto_elb_error
  - 3561e8c19b Better boto elb error message.
- **PR** #30831: (jacobhammons) Updated readme @ *2016-02-02 21:06:02 UTC*
  - 4da04f82c8 Merge pull request #30831 from jacobhammons/readme-update
  - 01a92f5d98 Updated readme
- **PR** #30829: (jacobhammons) Updated latest version to 2015.8.4 @ *2016-02-02 20:06:13 UTC*
  - 90c1ea9f6c Merge pull request #30829 from jacobhammons/release-2015.5
  - c95bb60148 Version to 2015.8.4
- **ISSUE** #24575: (BrandKNY) raid.present inside mdadm.py triggers IndexError: list index out of range (refs: #30784)
- **ISSUE** #23694: (gmolight) mdadm.py module (refs: #30784)
- **PR** #30784: (rallytime) Back-port #24952 to 2015.5 @ *2016-02-01 21:43:01 UTC*
  - **PR** #24952: (pcn) Don't split the string on a single line (refs: #30784)
  - 80a36793cb Merge pull request #30784 from rallytime/bp-24952
  - a07908bdea Don't split the string on a single line
- **ISSUE** #30560: (terminalmage) yumpkg.py: pkg.unhold fails in yum (refs: #30764)
- **PR** #30764: (terminalmage) Work around yum versionlock's inability to remove holds by package name alone @ *2016-02-01 18:14:27 UTC*
  - e978f5392f Merge pull request #30764 from terminalmage/issue30560
  - 39736afcd7 Work around yum versionlock's inability to remove holds by package name alone
- **PR** #30760: (toanju) Changed output format of arp_ip_target from list to comma delimited… @ *2016-01-31 19:05:02 UTC*
  - **PR** #27952: (tomwalsh) Corrected format of arp_ip_target in network config files and modprobe files (refs: #30760)
  - 6f565c0d76 Merge pull request #30760 from toanju/2015.5
  - dc4256f7df Changed output format of arp_ip_target from list to comma delimited string
- **ISSUE** #30722: (yannis666) mine config is not merged from minion config and pillar (refs: #30757)
- **PR** #30757: (yannis666) Fix to mine update to merge configuration @ *2016-01-31 19:02:44 UTC*
  - 1c205b4898 Merge pull request #30757 from yannis666/fix-for-mine-update-merge
  - 61bb23e256 Fix to mine update to merge configuration
- **ISSUE** #28751: (olfway) network.system state ignores test=True on debian/ubuntu (refs: #30749)
- **PR** #30749: (abednarik) Fix Netwotk hostname Module in Debian systems. @ *2016-01-29 23:01:09 UTC*
  - f9fde8f6a7 Merge pull request #30749 from abednarik/fix_network_system_test
  - 1e9e97df59 Fix Netwotk hostname Module in Debian systems.

- **ISSUE** #28438: (vakulich) Master failed to save job cache file: "Could not write job invocation cache file: [Errno 2] No such file or directory" (refs: #30699)
- **PR** #30699: (abednarik) Add Retry to save_load. @ *2016-01-29 16:08:30 UTC*
  - 076268089a Merge pull request #30699 from abednarik/save_load_retry_time
  - 186872cf49 Add Retry to save_load.
- **ISSUE** #30565: (heaje) scsi.ls fails to run both on CentOS 6 and CentOS 7 (refs: #30659)
- **PR** #30659: (sjmh) Fix lsscsi issues for certain platforms @ *2016-01-28 15:53:38 UTC*
  - 8d79d1b9c7 Merge pull request #30659 from sjmh/fix-scsi
  - 3544dd995e Fix lsscsi issues for certain platforms
- **ISSUE** #18980: (lrhazi) salt-cloud: ExtraData: unpack(b) received extra data. (refs: #30671)
- **PR** #30671: (techhat) Add file locking to cloud index @ *2016-01-27 17:14:55 UTC*
  - 516919525a Merge pull request #30671 from techhat/lockcloud
  - 4719f8d4ea Whitespace
  - 8e7eca23e4 Add file locking to cloud index
- **ISSUE** #28320: (Grokzen) file.comment & file.uncomment changes file permissions on edit (refs: #30586)
- **PR** #30586: (abednarik) Fix comment_line permissions. @ *2016-01-25 23:24:02 UTC*
  - 643c9c9616 Merge pull request #30586 from abednarik/fix_comment_line_perms
  - 8b395a42cb Fix comment_line permissions.
- **PR** #30582: (terminalmage) yumpkg.check_db: run separate repoquery commands when multiple names passed @ *2016-01-24 17:15:04 UTC*
  - a823e21428 Merge pull request #30582 from terminalmage/dnf-repoquery-multiple-targets
  - 410da789f9 yumpkg.check_db: run separate repoquery commands when multiple names passed
- **PR** #30548: (jacobhammons) Added placeholder release notes for 2015.5.10 @ *2016-01-22 18:36:01 UTC*
  - 8e56be7f4c Merge pull request #30548 from jacobhammons/doc-fixes
  - 03c51bb54d Added placeholder release notes for 2015.5.10 Changed old doc links from docs.saltstack.org to docs.saltstack.com
- **PR** #30530: (terminalmage) 2015.5 tweaks from #30529 @ *2016-01-22 16:26:21 UTC*
  - **PR** #30529: (terminalmage) Merge 2015.5 into 2015.8 (refs: #30530)
  - 1aafd4c5b5 Merge pull request #30530 from terminalmage/yumpkg-dnf-cleanup
  - 2586f71bcf 2015.5 tweaks from #30529
- **ISSUE** #23553: (aboe76) dnf a new package provider for fedora 22 (refs: #30484)
- **PR** #30484: (terminalmage) Backport DNF support to 2015.5 branch @ *2016-01-21 22:14:46 UTC*
  - 7798d42272 Merge pull request #30484 from terminalmage/dnf-yumpkg-2015.5
  - 330e26d1da Hide get_locked_packages
  - 5a637420e8 Backport DNF support to 2015.5 branch
- **PR** #30512: (jfindlay) disable pkgrepo test for ubuntu 15.10+ @ *2016-01-21 21:32:58 UTC*
  - b348f804b1 Merge pull request #30512 from jfindlay/repo_test

  - 66f06f2bd3 disable pkgrepo test for ubuntu 15.10+
- **PR** #30478: (justinta) Updated pip_state to work with pip 8.0 @ *2016-01-21 16:02:41 UTC*
  - a9348dfef8 Merge pull request #30478 from jtand/pip_8_update
  - 6227368830 Convert version to int, instead of comparing strings to ints
  - 20384a4810 Added InstallationError to except block
  - baa274bca9 Updated pip_state to work with pip 8.0
- **ISSUE** #30465: (alandrees) Nested imports with pyobjects (refs: #30482)
- **PR** #30482: (borgstrom) Pyobjects recursive import support (for 2015.5) @ *2016-01-21 15:54:32 UTC*
  - a30147c64f Merge pull request #30482 from borgstrom/pyobjects_recursive
  - 2c55a7580b Fixup lint errors
  - b46df0e4b5 Allow recursive salt:// imports
  - 51bfa16173 Add test to prove that recursive imports are currently broken
- **PR** #30459: (jfindlay) modules.pkg: disable repo int test for ubuntu 15.10 @ *2016-01-20 16:41:12 UTC*
  - 5c7cc51937 Merge pull request #30459 from jfindlay/pkg_tests
  - fb9972f590 modules.pkg: disable repo int test for ubuntu 15.10
- **PR** #30443: (justinta) Boto uses False for is_default instead of None @ *2016-01-19 18:28:08 UTC*
  - dd2ceb4c07 Merge pull request #30443 from jtand/boto_vpc_5
  - 2f77152479 Boto uses False for is_default instead of None
- **ISSUE** #26833: (twangboy) salt-cloud fails to spin up windows minion on 2015.8 Head (refs: #26853)
- **ISSUE** #21256: (dhs-rec) win.exe package for RH 6 (refs: #26853)
- **PR** #30420: (attiasr) Backport #26853 @ *2016-01-19 17:33:58 UTC*
  - **PR** #26853: (UtahDave) Fix salt-cloud on windows (refs: #30420)
  - 62d9dddced Merge pull request #30420 from attiasr/patch-1
  - 4de343c5a1 Backport #26853
- **ISSUE** #30341: (dnd) salt-cloud linode connection reset by peer (refs: #30364)
- **PR** #30364: (rallytime) Add TLS version imports and add linode driver documentation notices @ *2016-01-14 19:04:47 UTC*
  - 5a923b3aa9 Merge pull request #30364 from rallytime/fix-30341
  - 79bcf151cb Add TLS version imports and add linode driver documentation notices
- **ISSUE** #28822: (HerrBerg) saltenv url-parameter not working in file.managed for salt:// sources since 2015.8 (refs: #30166)
- **PR** #30184: (rallytime) Back-port #30166 to 2015.5 @ *2016-01-13 18:27:36 UTC*
  - **PR** #30166: (robgott) adding split_env call to cp.hash_file to pick up saltenv in file quer… (refs: #30184)
  - f037fd9c27 Merge pull request #30184 from rallytime/bp-30166
  - fa6b1b3022 adding split_env call to cp.hash_file to pick up saltenv in file query parameter
- **PR** #30291: (thegoodduke) ipset: fix test=true & add comment for every entry @ *2016-01-12 19:40:23 UTC*
  - **PR** #30170: (thegoodduke) ipset: fix comment and test (refs: #30291)

  - 1d8413fd2f Merge pull request #30291 from thegoodduke/for_fix_ipset
  - 62d6ccf561 ipset: fix test=true & add comment for every entry

28.2.69 Salt 2015.5.2 Release Notes

release 2015-06-10

Version 2015.5.2 is a bugfix release for *2015.5.0*.

Statistics

- Total Merges: **112**
- Total Issue References: **36**
- Total PR References: **145**
- Contributors: **49** (Sacro, The-Loeki, YanChii, aboe76, anlutro, awdrius, basepi, cdarwin, cedwards, clan, corywright, cro, djcrabhat, dmyerscough, dr4Ke, fayetted, galet, garethgreenaway, ghost, hazelesque, hvnsweeting, jacksontj, jacobhammons, jayeshka, jbq, jfindlay, joejulian, justinta, kartiksubbarao, kiorky, merll, msteed, neogenix, nicholascapo, nleib, pengyao, pruiz, rallytime, randybias, ryan-lane, steverweber, swdream, techhat, terminalmage, thcipriani, thusoy, trevor-h, twangboy, whiteinge)

Changelog for v2015.5.1..v2015.5.2

Generated at: 2018-05-27 21:13:02 UTC

- **PR** #24372: (rallytime) Add 2015.5.2 release notes @ *2015-06-03 19:30:46 UTC*
  - d71d75e2ec Merge pull request #24372 from rallytime/release_notes
  - f5ec1a1693 Add 2015.5.2 release notes
- **PR** #24346: (rallytime) Backport #24271 to 2015.5 @ *2015-06-03 18:44:31 UTC*
  - **PR** #24271: (randybias) Fixed the setup instructions (refs: #24346)
  - 76927c9ea1 Merge pull request #24346 from rallytime/bp-24271
  - 04067b6833 Fixed the setup instructions
- **ISSUE** #24012: (jbq) Enabling a service does not create the appropriate rc.d symlinks on Ubuntu (refs: #24013)
- **PR** #24345: (rallytime) Backport #24013 to 2015.5 @ *2015-06-03 18:39:41 UTC*
  - **PR** #24013: (jbq) Fix enabling a service on Ubuntu #24012 (refs: #24345)
  - 4afa03d8e3 Merge pull request #24345 from rallytime/bp-24013
  - 16e0732b50 Fix enabling a service on Ubuntu #24012
- **PR** #24365: (jacobhammons) Fixes for PDF build errors @ *2015-06-03 17:50:02 UTC*
  - c3392c246a Merge pull request #24365 from jacobhammons/DocFixes
  - 0fc190267f Fixes for PDF build errors
- **ISSUE** #22991: (nicholascapo) npm.installed ignores test=True (refs: #24313)
- **PR** #24313: (nicholascapo) Fix #22991 Correctly set result when test=True @ *2015-06-03 14:49:18 UTC*
  - ae681a4db1 Merge pull request #24313 from nicholascapo/fix-22991-npm.installed-test-true

  - ac9644cb19 Fix #22991 npm.installed correctly set result on test=True
- **ISSUE** #18966: (bechtoldt) file.serialize ignores test=True (refs: #24312)
- **PR** #24312: (nicholascapo) Fix #18966: file.serialize supports test=True @ *2015-06-03 14:49:06 UTC*
  - d57a9a267c Merge pull request #24312 from nicholascapo/fix-18966-file.serialize-test-true
  - e7328e7043 Fix #18966 file.serialize correctly set result on test=True
- **PR** #24302: (jfindlay) fix pkg hold/unhold integration test @ *2015-06-03 03:27:43 UTC*
  - 6b694e3495 Merge pull request #24302 from jfindlay/pkg_tests
  - c2db0b1758 fix pkg hold/unhold integration test
- **ISSUE** #14021: (emostar) EC2 doc mentions mount_point, but unable to use properly (refs: #24349)
- **PR** #24349: (rallytime) Remove references to mount_points in ec2 docs @ *2015-06-03 01:54:09 UTC*
  - aca8447ced Merge pull request #24349 from rallytime/fix-14021
  - a235b114d7 Remove references to mount_points in ec2 docs
- **PR** #24328: (dr4Ke) Fix state grains silently fails 2015.5 @ *2015-06-02 15:18:46 UTC*
  - 88a997e6ee Merge pull request #24328 from dr4Ke/fix_state_grains_silently_fails_2015.5
  - 8a63d1ebbe fix state grains silently fails #24319
  - ca1af20203 grains state: add some tests
- **ISSUE** #9772: (s0undt3ch) Delete VM's in a map does not delete them all (refs: #24310)
- **ISSUE** #24036: (arthurlogilab) [salt-cloud] Protect against passing command line arguments as names for the –destroy command in map files (refs: #24310)
- **PR** #24310: (techhat) Add warning about destroying maps @ *2015-06-02 03:01:28 UTC*
  - 7dcd9bb5de Merge pull request #24310 from techhat/mapwarning
  - ca535a6ff4 Add warning about destroying maps
- **PR** #24281: (steverweber) Ipmi docfix @ *2015-06-01 17:45:36 UTC*
  - 02bfb254d6 Merge pull request #24281 from steverweber/ipmi_docfix
  - dd36f2c555 yaml formating
  - f6deef3047 include api_kg kwarg in ipmi state
  - a7d4e97bb9 doc cleanup
  - 0ded2fdbef save more cleanup to doc
  - 08872f2da3 fix name api_key to api_kg
  - 165a387681 doc fix add api_kg kwargs
  - 1ec78887e4 cleanup docs
- **PR** #24287: (jfindlay) fix pkg test on ubuntu 12.04 for realz @ *2015-06-01 14:16:37 UTC*
  - 73cd2cbe1f Merge pull request #24287 from jfindlay/pkg_test
  - 98944d8c7f fix pkg test on ubuntu 12.04 for realz
- **PR** #24279: (rallytime) Backport #24263 to 2015.5 @ *2015-06-01 04:29:34 UTC*
  - **PR** #24263: (cdarwin) Correct usage of import_yaml in formula documentation (refs: #24279)

  - 02017a074c Merge pull request #24279 from rallytime/bp-24263
  - beff7c7785 Correct usage of import_yaml in formula documentation
- **ISSUE** #24226: (c4urself) iptables state needs to keep ordering of flags (refs: #24277)
- **PR** #24277: (rallytime) Put a space between after_jump commands @ *2015-06-01 04:28:26 UTC*
  - 2ba696d54a Merge pull request #24277 from rallytime/fix_iptables_jump
  - e2d1606b19 Move after_jump split out of loop
  - d14f1307b6 Remove extra loop
  - 42ed5320b6 Put a space between after_jump commands
- **PR** #24262: (basepi) More dictupdate after #24142 @ *2015-05-31 04:09:37 UTC*
  - **PR** #24142: (basepi) Optimize dictupdate.update and add #24097 functionality (refs: #24262)
  - **PR** #24097: (kiorky) Optimize dictupdate (refs: #24142)
  - 113eba34ec Merge pull request #24262 from basepi/dictupdatefix
  - 0c4832c0d4 Raise a typeerror if non-dict types
  - be21aaa122 Pylint
  - bb8a6c6cc9 More optimization
  - c933249d1a py3 compat
  - ff6b2a781f Further optimize dictupdate.update()
  - c73f5ba37c Remove unused valtype
- **PR** #24269: (kiorky) zfs: Fix spurious retcode hijacking in virtual @ *2015-05-30 17:47:49 UTC*
  - 785d5a1bfc Merge pull request #24269 from makinacorpus/zfs
  - 0bf23ce701 zfs: Fix spurious retcode hijacking in virtual
- **PR** #24257: (jfindlay) fix pkg mod integration test on ubuntu 12.04 @ *2015-05-29 23:09:00 UTC*
  - 3d885c04f0 Merge pull request #24257 from jfindlay/pkg_tests
  - 9508924c02 fix pkg mod integration test on ubuntu 12.04
- **ISSUE** #23883: (kaithar) max_event_size seems broken (refs: #24001, #24065)
- **ISSUE** #23657: (arthurlogilab) [salt-cloud lxc] NameError: global name '__salt__' is not defined (refs: #23982, #24080)
- **PR** #24260: (basepi) Fix some typos from #24080 @ *2015-05-29 22:54:58 UTC*
  - **PR** #24080: (kiorky) Lxc consistency2 (refs: #24066, #24260, #23982)
  - **PR** #24066: (kiorky) Merge forward 2015.5 -> develop (refs: #23982)
  - **PR** #24065: (kiorky) continue to fix #23883 (refs: #24066, #24080)
  - **PR** #23982: (kiorky) lxc: path support (refs: #24080)
  - 08a10755b3 Merge pull request #24260 from basepi/lxctypos24080
  - 0fa1ad3977 Fix another lxc typo
  - 669938f28d s/you ll/you'll/
- **ISSUE** #23883: (kaithar) max_event_size seems broken (refs: #24001, #24065)

- **ISSUE** #23657: (arthurlogilab) [salt-cloud lxc] NameError: global name '__salt__' is not defined (refs: #23982, #24080)
- **PR** #24080: (kiorky) Lxc consistency2 (refs: #24066, #24260, #23982) @ *2015-05-29 22:51:54 UTC*
  - **PR** #24066: (kiorky) Merge forward 2015.5 -> develop (refs: #23982)
  - **PR** #24065: (kiorky) continue to fix #23883 (refs: #24066, #24080)
  - **PR** #23982: (kiorky) lxc: path support (refs: #24080)
  - 75590cf490 Merge pull request #24080 from makinacorpus/lxc_consistency2
  - 81f80674a2 lxc: fix old lxc test
  - 458f50617b seed: lint
  - 96b8d55f14 Fix seed.mkconfig yamldump
  - 76ddb683f4 lxc/applynet: conservative
  - ce7096fdb7 variable collision
  - 8a8b28d652 lxc: lint
  - 458b18b7e6 more lxc docs
  - ef1f95231a lxc docs: typos
  - d67a43dc1f more lxc docs
  - 608da5ef5d modules/lxc: merge resolution
  - 27c4689a24 modules/lxc: more consistent comparsion
  - 07c365a23b lxc: merge conflict spotted
  - 999391551c modules/lxc: rework settings for consistency
  - ce11d8352e lxc: Global doc refresh
  - 61ed2f5e76 clouds/lxc: profile key is conflicting
- **ISSUE** #24210: (damonnk) salt-cloud vsphere.py should allow key_filename param (refs: #24220)
- **PR** #24247: (rallytime) Backport #24220 to 2015.5 @ *2015-05-29 21:40:01 UTC*
  - **PR** #24220: (djcrabhat) adding key_filename param to vsphere provider (refs: #24247)
  - da14f3b976 Merge pull request #24247 from rallytime/bp-24220
  - 0b1041dd72 adding key_filename param to vsphere provider
- **PR** #24254: (rallytime) Add deprecation warning to Digital Ocean v1 Driver @ *2015-05-29 21:39:25 UTC*
  - **PR** #22731: (dmyerscough) Decommission DigitalOcean APIv1 and have users use the new DigitalOcean APIv2 (refs: #24254)
  - 21d6126c34 Merge pull request #24254 from rallytime/add_deprecation_warning_digitalocean
  - cafe37bdf8 Add note to docs about deprecation
  - ea0f1e0921 Add deprecation warning to digital ocean driver to move to digital_ocean_v2
- **PR** #24252: (aboe76) Updated suse spec to 2015.5.1 @ *2015-05-29 21:38:45 UTC*
  - dac055dd8b Merge pull request #24252 from aboe76/opensuse_package
  - 0ad617df21 Updated suse spec to 2015.5.1
- **PR** #24251: (garethgreenaway) Returners broken in 2015.5 @ *2015-05-29 21:37:52 UTC*

  - 49e7fe8a5e Merge pull request #24251 from garethgreenaway/2015_5_returner_brokenness
  - 5df6b52568 The code calling cfg as a function vs treating it as a dictionary and using get is currently backwards causing returners to fail when used from the CLI and in scheduled jobs.
- **ISSUE** #21498: (rallytime) Clarify Digital Ocean Documentation (refs: #24255)
- **PR** #24255: (rallytime) Clarify digital ocean documentation and mention v1 driver deprecation @ *2015-05-29 21:37:07 UTC*
  - bfb946123e Merge pull request #24255 from rallytime/clarify_digital_ocean_driver_docs
  - 8d51f75aa5 Clarify digital ocean documentation and mention v1 driver deprecation
- **PR** #24232: (rallytime) Backport #23308 to 2015.5 @ *2015-05-29 21:36:46 UTC*
  - **PR** #23308: (thusoy) Don't merge: Add missing jump arguments to iptables module (refs: #24232)
  - 41f5756f36 Merge pull request #24232 from rallytime/bp-23308
  - 2733f66449 Import string
  - 9097cca099 Add missing jump arguments to iptables module
- **PR** #24245: (Sacro) Unset PYTHONHOME when starting the service @ *2015-05-29 20:00:31 UTC*
  - a95982c722 Merge pull request #24245 from Sacro/patch-2
  - 6632d06e94 Unset PYTHONHOME when starting the service
- **PR** #24121: (hvnsweeting) deprecate setting user permission in rabbitmq_vhost.present @ *2015-05-29 15:55:40 UTC*
  - 1504c76d3a Merge pull request #24121 from hvnsweeting/rabbitmq-host-deprecate-set-permission
  - 2223158e76 deprecate setting user permission in rabbitmq_host.present
- **PR** #24179: (merll) Changing user and group only possible for existing ids. @ *2015-05-29 15:52:43 UTC*
  - **PR** #24169: (merll) Changing user and group only possible for existing ids. (refs: #24179)
  - ba02f6509e Merge pull request #24179 from Precis/fix-file-uid-gid-2015.0
  - ee4c9d59ab Use ids if user or group is not present.
- **ISSUE** #24147: (paclat) Syndication issues when using authentication on master of masters. (refs: #24229)
- **PR** #24229: (msteed) Fix auth failure on syndic with external_auth @ *2015-05-29 15:04:06 UTC*
  - 9bfb066c2c Merge pull request #24229 from msteed/issue-24147
  - 482d1cfc64 Fix auth failure on syndic with external_auth
- **PR** #24234: (jayeshka) adding states/quota unit test case. @ *2015-05-29 14:14:27 UTC*
  - 19fa43c290 Merge pull request #24234 from jayeshka/quota-states-unit-test
  - c23356500b adding states/quota unit test case.
- **PR** #24217: (jfindlay) disable intermittently failing tests @ *2015-05-29 03:08:39 UTC*
  - **PR** #23623: (jfindlay) Fix /jobs endpoint's return (refs: #24217)
  - **PR** #22857: (jacksontj) Fix /jobs endpoint's return (refs: #23623)
  - e15142c629 Merge pull request #24217 from jfindlay/disable_bad_tests
  - 6b6280442c disable intermittently failing tests
- **PR** #24199: (ryan-lane) Various fixes for boto_route53 and boto_elb @ *2015-05-29 03:02:41 UTC*

  - ce8e43b774 Merge pull request #24199 from lyft/route53-fix-elb
  - d8dc9a7b5b Better unit tests for boto_elb state
  - 62f214b535 Remove cnames_present test
  - 7b9ae82951 Lint fix
  - b74b0d1413 Various fixes for boto_route53 and boto_elb
- **PR** #24142: (basepi) Optimize dictupdate.update and add #24097 functionality (refs: #24262) @ *2015-05-29 03:00:56 UTC*
  - **PR** #24097: (kiorky) Optimize dictupdate (refs: #24142)
  - a43465d235 Merge pull request #24142 from basepi/dictupdate24097
  - 5c6e210c8b Deepcopy on merge_recurse
  - a13c84ade8 Fix None check from #21968
  - 9ef2c64098 Add docstring
  - 8579429314 Add in recursive_update from #24097
  - 8599143200 if key not in dest, don't recurse
  - d8a84b3017 Rename klass to valtype
- **PR** #24208: (jayeshka) adding states/ports unit test case. @ *2015-05-28 23:06:33 UTC*
  - 526698ba8d Merge pull request #24208 from jayeshka/ports-states-unit-test
  - 657b709932 adding states/ports unit test case.
- **ISSUE** #20635: (dennisjac) 2015.2.0rc1: zfs errors in log after update (refs: #24219)
- **PR** #24219: (jfindlay) find zfs without modinfo @ *2015-05-28 21:07:26 UTC*
  - d00945fd40 Merge pull request #24219 from jfindlay/zfs_check
  - 15d401907c use the salt loader in the zfs mod
  - 5599b67a46 try to search for zfs if modinfo is unavailable
- **PR** #24190: (msteed) Fix issue 23815 @ *2015-05-28 20:10:34 UTC*
  - 3dc4b85295 Merge pull request #24190 from msteed/issue-23815
  - 086a1a94e8 lint
  - 65de62f852 fix #23815
  - d04e9162de spelling
  - db9f6820b8 add inotify beacon unit tests
- **PR** #24211: (rallytime) Backport #24205 to 2015.5 @ *2015-05-28 18:28:15 UTC*
  - **PR** #24205: (hazelesque) Docstring fix in salt.modules.yumpkg.hold (refs: #24211)
  - 436634b508 Merge pull request #24211 from rallytime/bp-24205
  - 23284b5d47 Docstring fix in salt.modules.yumpkg.hold
- **PR** #24212: (terminalmage) Clarify error in rendering template for top file @ *2015-05-28 18:26:20 UTC*
  - cc58624c7e Merge pull request #24212 from terminalmage/clarify-error-msg
  - ca807fb032 Clarify error in rendering template for top file

- **ISSUE** #23904: (mbrgm) Network config bonding section cannot be parsed when attribute names use dashes (refs: #23917)
- **ISSUE** #23900: (hashi825) salt ubuntu network building issue 2015.5.0 (refs: #23922)
- **PR** #24213: (The-Loeki) ShouldFix _- troubles in debian_ip @ *2015-05-28 18:24:39 UTC*
  - **PR** #23922: (garethgreenaway) Fixes to debian_ip.py (refs: #24213)
  - **PR** #23917: (corywright) Split debian bonding options on dash instead of underscore (refs: #24213)
  - 9825160b1a Merge pull request #24213 from The-Loeki/patch-3
  - a68d515973 ShouldFix _- troubles in debian_ip
- **PR** #24214: (basepi) 2015.5.1release @ *2015-05-28 16:23:57 UTC*
  - 071751d13f Merge pull request #24214 from basepi/2015.5.1release
  - e5ba31b5b5 2015.5.1 release date
  - 768494c819 Update latest release in docs
- **PR** #24202: (rallytime) Backport #24186 to 2015.5 @ *2015-05-28 05:16:48 UTC*
  - **PR** #24186: (thcipriani) Update salt vagrant provisioner info (refs: #24202)
  - c2f1fdb244 Merge pull request #24202 from rallytime/bp-24186
  - db793dd0de Update salt vagrant provisioner info
- **PR** #24192: (rallytime) Backport #20474 to 2015.5 @ *2015-05-28 05:16:18 UTC*
  - **PR** #20474: (djcrabhat) add sudo, sudo_password params to vsphere deploy to allow for non-root deploys (refs: #24192)
  - 8a085a2592 Merge pull request #24192 from rallytime/bp-20474
  - fd3c783f3e add sudo, sudo_password params to deploy to allow for non-root deploys
- **PR** #24184: (rallytime) Backport #24129 to 2015.5 @ *2015-05-28 05:15:08 UTC*
  - **PR** #24129: (pengyao) Wheel client doc (refs: #24184)
  - 7cc535bf4a Merge pull request #24184 from rallytime/bp-24129
  - 722a662479 fixed a typo
  - 565eb46ff5 Add cmd doc for WheelClient
- **PR** #24183: (rallytime) Backport #19320 to 2015.5 @ *2015-05-28 05:14:36 UTC*
  - **PR** #19320: (clan) add 'state_output_profile' option for profile output (refs: #24183)
  - eb0af70e5b Merge pull request #24183 from rallytime/bp-19320
  - 55db1bf8b5 sate_output_profile default to True
  - 991922703b fix type: statei -> state
  - 0549ca6266 add 'state_output_profile' option for profile output
- **PR** #24201: (whiteinge) Add list of client libraries for the rest_cherrypy module to the top-level documentation @ *2015-05-28 02:12:09 UTC*
  - 1b5bf23187 Merge pull request #24201 from whiteinge/rest_cherrypy-client-libs
  - 5f718027ca Add list of client libraries for the rest_cherrypy module
  - 28fc77f6f6 Fix rest_cherrypy config example indentation

- **PR** #24195: (rallytime) Merge #24185 with a couple of fixes @ *2015-05-27 22:18:37 UTC*
  - **PR** #24185: (jacobhammons) Fixes for doc build errors (refs: #24195)
  - 3307ec20d9 Merge pull request #24195 from rallytime/merge-24185
  - d8daa9dcd7 Merge #24185 with a couple of fixes
  - 634d56bca0 Fixed pylon error
  - 0689815d0e Fixes for doc build errors
- **PR** #24166: (jayeshka) adding states/pkgng unit test case. @ *2015-05-27 20:27:49 UTC*
  - 7e400bc3d7 Merge pull request #24166 from jayeshka/pkgng-states-unit-test
  - 2234bb0b70 adding states/pkgng unit test case.
- **PR** #24189: (basepi) [2015.5] Merge forward from 2014.7 to 2015.5 @ *2015-05-27 20:26:31 UTC*
  - 9fcda79cd4 Merge pull request #24189 from basepi/merge-forward-2015.5
  - 8839e9c22e Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.5
  - 9d7331c87d Merge pull request #24178 from rallytime/bp-24118
    - e2217a09e8 removed deprecated pymongo usage as no longer functional with pymongo > 3.x
  - 4e8c5031b0 Merge pull request #24159 from rallytime/keystone_doc_examples
    - dadac8d076 Fill out modules/keystone.py CLI Examples
  - fc10ee8ed5 Merge pull request #24158 from rallytime/fix_doc_error
    - 49a517e2ca Fix test_valid_docs test for tls module
- **PR** #24181: (justinta) Fixed error where file was evaluated as a symlink in test_absent @ *2015-05-27 18:26:28 UTC*
  - 2303dec0e9 Merge pull request #24181 from jtand/file_test
  - 5f0e601589 Fixed error where file was evaluated as a symlink in test_absent
- **PR** #24180: (terminalmage) Skip libvirt tests if not running as root @ *2015-05-27 18:18:47 UTC*
  - a16276852b Merge pull request #24180 from terminalmage/fix-libvirt-test
  - 72e7416ad2 Skip libvirt tests if not running as root
- **PR** #24165: (jayeshka) adding states/portage_config unit test case. @ *2015-05-27 17:15:08 UTC*
  - 1fbc5b25e6 Merge pull request #24165 from jayeshka/portage_config-states-unit-test
  - 8cf1505392 adding states/portage_config unit test case.
- **PR** #24164: (jayeshka) adding states/pecl unit test case. @ *2015-05-27 17:14:26 UTC*
  - 4747856411 Merge pull request #24164 from jayeshka/pecl-states-unit-test
  - 563a5b3c30 adding states/pecl unit test case.
- **PR** #24160: (The-Loeki) small enhancement to data module; pop() @ *2015-05-27 17:03:10 UTC*
  - cdaaa19324 Merge pull request #24160 from The-Loeki/patch-1
  - 2175ff3c75 doc & merge fix
  - eba382cdda small enhancement to data module; pop()
- **PR** #24153: (techhat) Batch mode sometimes improperly builds lists of minions to process @ *2015-05-27 16:21:53 UTC*

  - 4a8dbc7f13 Merge pull request #24153 from techhat/batchlist
  - 467ba64612 Make sure that minion IDs are strings
- **PR** #24167: (jayeshka) adding states/pagerduty unit test case. @ *2015-05-27 16:14:01 UTC*
  - ed8ccf57a2 Merge pull request #24167 from jayeshka/pagerduty-states-unit-test
  - 1af8c8334d adding states/pagerduty unit test case.
- **PR** #24156: (basepi) [2015.5] Merge forward from 2014.7 to 2015.5 @ *2015-05-27 15:05:01 UTC*
  - b9507d1567 Merge pull request #24156 from basepi/merge-forward-2015.5
  - e52b5ab2e2 Remove stray >>>>>
  - 7dfbd929ff Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.5
    - c0d32e0b5e Merge pull request #24125 from hvnsweeting/fix-rabbitmq-test-mode
      - 71862c69b9 enhance log
      - 28e2594162 change according to new output of rabbitmq module functions
      - cd0212e8ed processes and returns better output for rabbitmq module
    - 39a8f30f06 Merge pull request #24093 from msteed/issue-23464
      - fd35903d75 Fix failing test
      - 41b344c7d3 Make LocalClient.cmd_iter_no_block() not block
    - 5bffd3045e Merge pull request #24008 from davidjb/2014.7
      - 8b8d0293d4 Correct reST formatting for documentation
    - 1aa0420040 Merge pull request #23933 from jacobhammons/2014.7
    - a3613e68e4 removed numbering from doc TOC
    - 78b737c5e6 removed 2015.* release from release notes, updated index page to remove PDF/epub links
    - e867f7df77 Changed build settings to use saltstack2 theme and update release versions.
    - 81ed9c9f59 sphinx saltstack2 doc theme
- **ISSUE** #24102: (bormotov) win_update encondig problems (refs: #24145)
- **PR** #24145: (jfindlay) attempt to decode win update package @ *2015-05-26 23:20:20 UTC*
  - 05745fa931 Merge pull request #24145 from jfindlay/win_update_encoding
  - cc5e17e61f attempt to decode win update package
- **ISSUE** #24122: (kiorky) service.dead is no more stateful: services does not handle correctly enable/disable change state (refs: #24123)
- **PR** #24123: (kiorky) fix service enable/disable change @ *2015-05-26 21:24:19 UTC*
  - 70247890de Merge pull request #24123 from makinacorpus/ss
  - 2e2e1d262d fix service enable/disable change
- **PR** #24146: (rallytime) Fixes the boto_vpc_test failure on CentOS 5 tests @ *2015-05-26 20:15:19 UTC*
  - 51c3cec5d7 Merge pull request #24146 from rallytime/fix_centos_boto_failure
  - ac0f97de51 Fixes the boto_vpc_test failure on CentOS 5 tests

- **ISSUE** #24052: (twangboy) v2015.5.1 Changes the way it interprets the minion_master.pub file (refs: #24144, #24089)
- **ISSUE** #23566: (rks2286) Salt-cp corrupting the file after transfer to minion (refs: #24144, #23740)
- **PR** #24144: (twangboy) Compare Keys ignores all newlines and carriage returns @ *2015-05-26 19:25:48 UTC*
  - **PR** #23740: (jfindlay) Binary write (refs: #24144)
  - 1c91a2176f Merge pull request #24144 from twangboy/fix_24052
  - c197b41494 Compare Keys removing all newlines and carriage returns
- **PR** #24139: (rallytime) Backport #24118 to 2015.5 @ *2015-05-26 18:24:27 UTC*
  - **PR** #24118: (trevor-h) removed deprecated pymongo usage (refs: #24178, #24139)
  - 084166747c Merge pull request #24139 from rallytime/bp-24118
  - 4bb519b8da removed deprecated pymongo usage as no longer functional with pymongo > 3.x
- **PR** #24138: (rallytime) Backport #24116 to 2015.5 @ *2015-05-26 18:23:51 UTC*
  - **PR** #24116: (awdrius) Fixed typo in chown username (ending dot) that fails the command. (refs: #24138)
  - 742eca29f7 Merge pull request #24138 from rallytime/bp-24116
  - 7f08641800 Fixed typo in chown username (ending dot) that fails the command.
- **PR** #24137: (rallytime) Backport #24105 to 2015.5 @ *2015-05-26 18:23:40 UTC*
  - **PR** #24105: (cedwards) Updated some beacon-specific documentation formatting (refs: #24137)
  - e01536d098 Merge pull request #24137 from rallytime/bp-24105
  - f0778a0a60 Updated some beacon-specific documentation formatting
- **ISSUE** #23364: (pruiz) Unable to destroy host using proxmox cloud: There was an error destroying machines: 501 Server Error: Method 'DELETE /nodes/pmx1/openvz/openvz/100' not implemented (refs: #24104)
- **PR** #24136: (rallytime) Backport #24104 to 2015.5 @ *2015-05-26 15:58:47 UTC*
  - **PR** #24104: (pruiz) Only try to stop a VM if it's not already stopped. (fixes #23364) (refs: #24136)
  - 89cdf976e1 Merge pull request #24136 from rallytime/bp-24104
  - c53888415f Only try to stop a VM if it's not already stopped. (fixes #23364)
- **PR** #24135: (rallytime) Backport #24083 to 2015.5 @ *2015-05-26 15:58:27 UTC*
  - **PR** #24083: (swdream) fix code block syntax (refs: #24135)
  - 67c4373577 Merge pull request #24135 from rallytime/bp-24083
  - e1d06f9764 fix code block syntax
- **PR** #24131: (jayeshka) adding states/mysql_user unit test case @ *2015-05-26 15:58:10 UTC*
  - a83371e0ed Merge pull request #24131 from jayeshka/mysql_user-states-unit-test
  - ed1ef69856 adding states/mysql_user unit test case
- **PR** #24130: (jayeshka) adding states/ntp unit test case @ *2015-05-26 15:57:29 UTC*
  - 1dc1d2a6e5 Merge pull request #24130 from jayeshka/ntp-states-unit-test
  - ede4a9f2f1 adding states/ntp unit test case
- **PR** #24128: (jayeshka) adding states/openstack_config unit test case @ *2015-05-26 15:56:08 UTC*
  - 39434179a8 Merge pull request #24128 from jayeshka/openstack_config-states-unit-test

  - ca09e0f7c1 adding states/openstack_config unit test case
- **PR** #24127: (jayeshka) adding states/npm unit test case @ *2015-05-26 15:55:18 UTC*
  - 23f25c4298 Merge pull request #24127 from jayeshka/npm-states-unit-test
  - c3ecabbae0 adding states/npm unit test case
- **ISSUE** #24009: (hvnsweeting) state_verbose False summary is wrong (refs: #24077)
- **PR** #24077: (anlutro) Change how state_verbose output is filtered @ *2015-05-26 15:41:11 UTC*
  - 07488a4415 Merge pull request #24077 from alprs/fix-outputter_highstate_nonverbose_count
  - 7790408c3c Change how state_verbose output is filtered
- **PR** #24119: (jfindlay) Update contrib docs @ *2015-05-26 15:37:01 UTC*
  - 224820febf Merge pull request #24119 from jfindlay/update_contrib_docs
  - fa2d411f53 update example release branch in contrib docs
  - a0b76b57b3 clarify git rebase instructions
  - 3517e0095f fix contribution docs link typos
  - 651629c6a4 backport dev contrib doc updates to 2015.5
- **PR** #23928: (joejulian) Add the ability to replace existing certificates @ *2015-05-25 19:47:26 UTC*
  - 5488c4aaa2 Merge pull request #23928 from joejulian/2015.5_tls_module_replace_existing
  - 4a4cbdd266 Add the ability to replace existing certificates
- **ISSUE** #23221: (Reiner030) Debian Jessie: locale.present not working again (refs: #24078)
- **PR** #24078: (jfindlay) if a charmap is not supplied, set it to the codeset @ *2015-05-25 19:39:19 UTC*
  - dd90ef09b9 Merge pull request #24078 from jfindlay/locale_charmap
  - 5eb97f0973 if a charmap is not supplied, set it to the codeset
- **PR** #24088: (jfindlay) pkg module integration tests @ *2015-05-25 19:39:02 UTC*
  - 9cec5d3dc9 Merge pull request #24088 from jfindlay/pkg_tests
  - f1bd5ec404 adding pkg module integration tests
  - 739b2ef3bd rework yumpkg refresh_db so args are not mandatory
- **ISSUE** #24052: (twangboy) v2015.5.1 Changes the way it interprets the minion_master.pub file (refs: #24144, #24089)
- **PR** #24089: (jfindlay) allow override of binary file mode on windows @ *2015-05-25 19:38:44 UTC*
  - 517552caa6 Merge pull request #24089 from jfindlay/binary_write
  - b2259a6370 allow override of binary file mode on windows
- **ISSUE** #23973: (mschiff) state file.managed: setting contents_pillar to a pillar which is a list throws exception instead giving descriptive error message (refs: #24092)
- **PR** #24092: (jfindlay) collect scattered contents edits, ensure it's a str @ *2015-05-25 19:38:10 UTC*
  - 121ab9f857 Merge pull request #24092 from jfindlay/file_state
  - cfa0f1358e collect scattered contents edits, ensure it's a str
- **PR** #24112: (The-Loeki) thin_gen breaks when thinver doesn't exist @ *2015-05-25 19:37:47 UTC*
  - 84e65dece7 Merge pull request #24112 from The-Loeki/patch-1

  - 34646eae16 thin_gen breaks when thinver doesn't exist
- **PR** #24108: (jayeshka) adding states/mysql_query unit test case @ *2015-05-25 12:30:48 UTC*
  - ec509ed272 Merge pull request #24108 from jayeshka/mysql_query-states-unit-test
  - ec50450460 adding states/mysql_query unit test case
- **PR** #24110: (jayeshka) adding varnish unit test case @ *2015-05-25 12:30:21 UTC*
  - f2e5d6c2fd Merge pull request #24110 from jayeshka/varnish-unit-test
  - e11988969f adding varnish unit test case
- **PR** #24109: (jayeshka) adding states/mysql_grants unit test case @ *2015-05-25 12:29:53 UTC*
  - 4fca2b49e3 Merge pull request #24109 from jayeshka/mysql_grants-states-unit-test
  - 11a93cb80c adding states/mysql_grants unit test case
- **PR** #24028: (nleib) send a disable message to disable puppet @ *2015-05-25 04:02:11 UTC*
  - 6b43c9a8cb Merge pull request #24028 from nleib/2015.5
  - 15f24b42b2 update format of string in disabled msg
  - 7690e5b008 remove trailing whitespaces
  - 56a972034f Update puppet.py
  - 9686391d81 Update puppet.py
  - 33f3d68489 send a disable message to disable puppet
- **PR** #24100: (jfindlay) adding states/file unit test case @ *2015-05-24 05:17:54 UTC*
  - **PR** #23963: (jayeshka) adding states/file unit test case (refs: #24100)
  - 52c9acafc2 Merge pull request #24100 from jfindlay/merge_23963
  - 7d59deb3d6 adding states/file unit test case
- **ISSUE** #21446: (dpheasant) check for systemd on Oracle Linux (refs: #24098)
- **PR** #24098: (galet) Systemd not recognized properly on Oracle Linux 7 @ *2015-05-24 04:07:31 UTC*
  - 0eb9f15d20 Merge pull request #24098 from galet/2015.5
  - 4d6ab21c74 Systemd not recognized properly on Oracle Linux 7
- **PR** #24090: (jfindlay) adding states/mount unit test case @ *2015-05-22 23:02:57 UTC*
  - **PR** #24062: (jayeshka) adding states/mount unit test case (refs: #24090)
  - 8e04db76de Merge pull request #24090 from jfindlay/merge_24062
  - a81a9225b8 adding states/mount unit test case
- **ISSUE** #22574: (unicolet) error when which is not available (refs: #22806)
- **PR** #24086: (rallytime) Backport #22806 to 2015.5 @ *2015-05-22 21:18:20 UTC*
  - **PR** #22806: (jfindlay) use cmd.run_all instead of cmd.run_stdout (refs: #24086)
  - c0079f5dc7 Merge pull request #24086 from rallytime/bp-22806
  - f728f55160 use cmd.run_all instead of cmd.run_stdout
- **PR** #24024: (jayeshka) adding states/mongodb_user unit test case @ *2015-05-22 20:53:19 UTC*
  - 09de253373 Merge pull request #24024 from jayeshka/mongodb_user-states-unit-test

  - f31dc921f5 resolved errors
  - d038b1fdbb adding states/mongodb_user unit test case
- **ISSUE** #23883: (kaithar) max_event_size seems broken (refs: #24001, #24065)
- **PR** #24065: (kiorky) continue to fix #23883 (refs: #24066, #24080) @ *2015-05-22 18:59:21 UTC*
  - bfd812c56b Merge pull request #24065 from makinacorpus/real23883
  - 028282e01d continue to fix #23883
- **ISSUE** #24017: (arthurlogilab) [salt-cloud openstack] TypeError: unhashable type: 'dict' on map creation (refs: #24029)
- **PR** #24029: (kiorky) Fix providers handling @ *2015-05-22 16:56:06 UTC*
  - 429adfe00a Merge pull request #24029 from makinacorpus/fixproviders
  - 412b39b802 Fix providers handling
- **PR** #23936: (jfindlay) remove unreachable returns in file state @ *2015-05-22 16:26:49 UTC*
  - a42ccccd98 Merge pull request #23936 from jfindlay/file_state
  - ac29c0cdd0 also validate file.recurse source parameter
  - 57f73887fe remove unreachable returns in file state
- **PR** #24063: (jayeshka) removed tuple index error @ *2015-05-22 14:58:20 UTC*
  - 8b69b41a42 Merge pull request #24063 from jayeshka/mount-states-module
  - b9745d5c4f removed tuple index error
- **PR** #24057: (rallytime) Backport #22572 to 2015.5 @ *2015-05-22 05:36:25 UTC*
  - **PR** #22572: (The-Loeki) Small docfix for GitPillar (refs: #24057)
  - 02ac4aa288 Merge pull request #24057 from rallytime/bp-22572
  - 49aad84b17 Small docfix for GitPillar
- **ISSUE** #23088: (ghost) Segfault when adding a Zypper repo on SLES 11.3 (refs: #24027)
- **PR** #24040: (rallytime) Backport #24027 to 2015.5 @ *2015-05-21 23:43:54 UTC*
  - **PR** #24027: (ghost) Add baseurl to salt.modules.zypper.mod_repo (refs: #24040)
  - 82de059891 Merge pull request #24040 from rallytime/bp-24027
  - 37d25d8bc6 Added baseurl as alias for url and mirrorlist in salt.modules.zypper.mod_repo.
- **PR** #24039: (rallytime) Backport #24015 to 2015.5 @ *2015-05-21 23:43:25 UTC*
  - **PR** #24015: (YanChii) minor improvement of solarisips docs & fix typos (refs: #24039)
  - d909781d97 Merge pull request #24039 from rallytime/bp-24015
  - 6bfaa94a8c minor improovement of solarisips docs & fix typos
- **ISSUE** #19598: (fayetted) ssh_auth.present test=true incorectly reports changes will be made (refs: #19599)
- **PR** #24038: (rallytime) Backport #19599 to 2015.5 @ *2015-05-21 23:43:10 UTC*
  - **PR** #19599: (fayetted) Fix ssh_auth test mode, compare lines not just key (refs: #24038)
  - 4a0f254d22 Merge pull request #24038 from rallytime/bp-19599
  - ea00d3e786 Fix ssh_auth test mode, compare lines not just key

- **PR** #24046: (rallytime) Remove key management test from digital ocean cloud tests @ *2015-05-21 22:32:04 UTC*
  - 42b87f1049 Merge pull request #24046 from rallytime/remove_key_test
  - 1d031caa78 Remove key management test from digital ocean cloud tests
- **PR** #24044: (cro) Remove spurious log message, fix typo in doc @ *2015-05-21 22:31:49 UTC*
  - eff54b1c5a Merge pull request #24044 from cro/pgjsonb
  - de0663314a Remove spurious log message, fix typo in doc
- **ISSUE** #23883: (kaithar) max_event_size seems broken (refs: #24001, #24065)
- **PR** #24001: (msteed) issue #23883 @ *2015-05-21 20:32:30 UTC*
  - ac32000b5d Merge pull request #24001 from msteed/issue-23883
  - bea97a8b98 issue #23883
- **PR** #23995: (kiorky) Lxc path pre @ *2015-05-21 17:26:03 UTC*
  - f7fae26059 Merge pull request #23995 from makinacorpus/lxc_path_pre
  - 319282af5f lint
  - 1dc67e5678 lxc: versionadded
  - fcad7cb804 lxc: states improvments
  - 644bd729f7 lxc: more consistence for profiles
  - 139372c055 lxc: remove merge cruft
  - 725b0462ca lxc: Repair merge
- **ISSUE** #16383: (interjection) salt.states.augeas.change example from docs fails with exception (refs: #24032)
- **PR** #24032: (kartiksubbarao) Update augeas_cfg.py @ *2015-05-21 17:03:42 UTC*
  - 26d6851666 Merge pull request #24032 from kartiksubbarao/augeas_insert_16383
  - 3686dcd4c7 Update augeas_cfg.py
- **PR** #24025: (jayeshka) adding timezone unit test case @ *2015-05-21 16:50:53 UTC*
  - 55c9245075 Merge pull request #24025 from jayeshka/timezone-unit-test
  - 1ec33e22a7 removed assertion error
  - 16ecb28950 adding timezone unit test case
- **PR** #24023: (jayeshka) adding states/mongodb_database unit test case @ *2015-05-21 16:49:17 UTC*
  - e243617659 Merge pull request #24023 from jayeshka/mongodb_database-states-unit-test
  - 5a9ac7effb adding states/mongodb_database unit test case
- **PR** #24022: (jayeshka) adding states/modjk_worker unit test case @ *2015-05-21 16:48:29 UTC*
  - b377bd93e6 Merge pull request #24022 from jayeshka/modjk_worker-states-unit-test
  - 05c0a985db adding states/modjk_worker unit test case
- **ISSUE** #23776: (enblde) Presence change events constantly reporting all minions as new in 2015.5 (refs: #24005)
- **PR** #24005: (msteed) issue #23776 @ *2015-05-21 01:55:34 UTC*
  - 701c51ba7a Merge pull request #24005 from msteed/issue-23776
  - 62e67d8ca0 issue #23776

- **ISSUE** #23950: (neogenix) iptables state generates a 0 position which is invalid in iptables cli (refs: #23996)
- **PR** #23996: (neogenix) iptables state generates a 0 position which is invalid in iptables cli #23950 @ *2015-05-20 22:44:27 UTC*
  - 17b7c0b741 Merge pull request #23996 from neogenix/2015.5-23950
  - ad417a57c2 fix for #23950
- **PR** #23994: (rallytime) Skip the gpodder pkgrepo test for Ubuntu 15 - they don't have vivid ppa up yet @ *2015-05-20 21:18:21 UTC*
  - 4cb877307c Merge pull request #23994 from rallytime/skip_test_ubuntu_15
  - 9e0ec07d85 Skip the gpodder pkgrepo test - they don't have vivid ppa up yet

28.2.70 Salt 2015.5.3 Release Notes

release 2015-07-07

Version 2015.5.3 is a bugfix release for *2015.5.0*.

Statistics

- Total Merges: **178**
- Total Issue References: **69**
- Total PR References: **207**
- Contributors: **62** (CameronNemo, Lanzaa, Starblade42, The-Loeki, TheScriptSage, aboe76, ahus1, aneeshusa, anlutro, arthurlogilab, basepi, borutmrak, cachedout, cgtx, codertux, cro, dkiser, driskell, eliasp, garethgreenaway, grischa, gthb, heewa, infestdead, jacksontj, jacobhammons, jayeshka, jeanpralo, jfindlay, jodv, joejulian, justinta, kartiksubbarao, kev009, kiorky, lorengordon, msciciel, msteed, nmadhok, notpeter, obestwalter, pengyao, pille, porterjamesj, pruiz, quixoten, rallytime, rhertzog, ruzarowski, ryan-lane, steverweber, tankywoo, tbaker57, techhat, terminalmage, thatch45, thenewwazoo, trevor-h, twangboy, variia, zefrog, zhujinhe)

Changelog for v2015.5.2..v2015.5.3

Generated at: 2018-05-27 21:20:01 UTC

- **PR** #25109: (jfindlay) add 2015.5.3 release notes @ *2015-07-01 19:45:56 UTC*
  - f0f512a4da Merge pull request #25109 from jfindlay/2015.5
  - 3187d5d5aa add 2015.5.3 release notes
- **PR** #25096: (jfindlay) Postgres group test @ *2015-07-01 18:48:26 UTC*
  - **PR** #24330: (jayeshka) adding states/postgres_group unit test case. (refs: #25096)
  - 21709aa483 Merge pull request #25096 from jfindlay/postgres_group_test
  - 3c379dc115 declobber postgres state unit test mocking
  - a162ffa3d8 adding states/postgres_group unit test case.
- **ISSUE** #25041: (wt) REGRESSION: pillar.get of integer fails to render in sls (refs: #25085)
- **PR** #25085: (jfindlay) accept all sources in the file state @ *2015-07-01 18:23:45 UTC*
  - 0a846400c6 Merge pull request #25085 from jfindlay/fix_file

  - 937a252e16 remove unnecessary file state tests
  - 6f238e924c integration test file.managed sources
  - a5978d30c2 iterate an iterable source othwerise list+str it
- **PR** #25095: (jfindlay) Win groupadd unit tests @ *2015-07-01 18:18:53 UTC*
  - **PR** #24207: (jayeshka) adding win_groupadd unit test case. (refs: #25095)
  - a98394210e Merge pull request #25095 from jfindlay/win_groupadd_test
  - 564dffd14a depend on win libs rather than mocking them
  - 9b9aeb8628 resolved all erors.
  - aaf89354c0 adding win_groupadd unit test case.
- **ISSUE** #21520: (jfindlay) sudo.salt_call is broken (refs: #25089)
- **PR** #25089: (jfindlay) fix minion sudo @ *2015-07-01 15:53:16 UTC*
  - **PR** #20226: (thatch45) Allow sudo priv escalation (refs: #25089)
  - 7c8d2a8656 Merge pull request #25089 from jfindlay/fix_sudo
  - d8f91d4a19 add some apprehension to the sudo exec module
  - a9269c072a adding sudo exec module docs
  - e4a40b7bd8 comment whitespace in minion config
  - 44cb167744 adding sudo_user minion config docs
  - d461060efe adding sudo_user minion config to default
- **ISSUE** #24875: (ahammond) ValueError: list.remove(x): x not in list in File "/usr/lib/python2.6/site-packages/salt/cli/batch.py", line 179, in run active.remove(minion) (refs: #25099)
- **PR** #25099: (driskell) Fix broken batch results @ *2015-07-01 15:51:29 UTC*
  - 4d6078e5dd Merge pull request #25099 from driskell/patch-1
  - 59b23e5f6e Fix broken batch results
- **PR** #25083: (steverweber) ipmi: get_sensor_data would always fail @ *2015-06-30 20:57:21 UTC*
  - 46350796b6 Merge pull request #25083 from steverweber/fix_ipmi_stat
  - 836f48c378 include _ in IpmiCommand
  - 817e434591 get_sensor_data would always fail
- **PR** #25067: (The-Loeki) Fix for maxdepth=0 in find @ *2015-06-30 20:54:06 UTC*
  - 15f2a4077c Merge pull request #25067 from The-Loeki/patch-1
  - 61edad3a80 Fix for maxdepth=0 in find
- **PR** #25078: (terminalmage) Use smaller number for upper limit of mac_user's _first_avail_uid helper function @ *2015-06-30 20:53:24 UTC*
  - 58d933cfa8 Merge pull request #25078 from terminalmage/fix-mac-uid
  - df2ab7ee2b Use smaller number for upper limit of mac_user's _first_avail_uid helper function
- **ISSUE** #24521: (multani) State network.managed fails on Debian (Jessie) (refs: #25045)
- **PR** #25045: (garethgreenaway) Fixes to debian_ip.py in 2015.5 @ *2015-06-30 17:36:43 UTC*
  - ebd6cdc412 Merge pull request #25045 from garethgreenaway/24521_debian_networking

  - 6f2a6c940b having proto default to static since it's needed to build the template.
- **PR** #25065: (lorengordon) Add download links for 2015.5.1-3 and 2015.5.2 Windows installers @ *2015-06-30 15:29:31 UTC*
  - ae31b279cc Merge pull request #25065 from lorengordon/update-windows-installer-links
  - 40a0c132d4 Add download links for 2015.5.1-3 and 2015.5.2, Fixes #25057
- **PR** #25052: (basepi) [2015.5] Merge forward from 2014.7 to 2015.5 @ *2015-06-30 01:05:00 UTC*
  - ddaeb0fb8e Merge pull request #25052 from basepi/merge-forward-2015.5
  - 2c5e664a58 Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.5
  - a7154e7471 Merge pull request #25011 from notpeter/s3_2014.7_backport
    - 8b8af640f6 Add s3 to protocols for remote source_hash
- **PR** #25038: (jfindlay) versionadded @ *2015-06-29 19:49:27 UTC*
  - **PR** #24747: (msciciel) add get_route function to network module (refs: #25038)
  - c7003d4951 Merge pull request #25038 from jfindlay/versionadded
  - d6dc6f97b5 versionadded
- **PR** #24747: (msciciel) add get_route function to network module (refs: #25038) @ *2015-06-29 16:51:43 UTC*
  - 28c87cab17 Merge pull request #24747 from msciciel/2015.5
  - 79b4ec2da8 network module lint fix
  - 0b6ef784b2 network module: fix for ipv6
  - f3d184c478 add get_route function to network module
- **PR** #24975: (ryan-lane) Fix update of undefined env var in npm module @ *2015-06-29 16:45:05 UTC*
  - 46a96773aa Merge pull request #24975 from lyft/npm-module-fix
  - 6fde58182f Try byte literals rather than unicode strings in the env
  - c8514de334 Fix update of undefined env var in npm module
- **PR** #24986: (heewa) Don't modify empty change @ *2015-06-29 16:44:17 UTC*
  - 9cf8550cd8 Merge pull request #24986 from heewa/fix-pkg-hold-when-errored
  - d47a448a80 Don't modify empty change
- **ISSUE** #24969: (bradthurber) salt-cloud 2015.5.0: missing azure dependency results in misleading error (refs: #24999)
- **PR** #24999: (rallytime) Provide a less confusing error when cloud provider is misconfigured @ *2015-06-29 16:43:31 UTC*
  - ece897d8d6 Merge pull request #24999 from rallytime/cloud_error_help
  - 1e81a88625 Clean up
  - be19a6730e Provide a less confusing error when cloud provider is misconfigured
- **PR** #24987: (heewa) Don't try to cache a template when it's not a file @ *2015-06-29 14:02:59 UTC*
  - 4af15cfb90 Merge pull request #24987 from heewa/fix-trying-to-cache-no-file
  - 9ae0c78ffc Don't try to cache a template when it's not a file
- **PR** #25022: (jfindlay) revise label and milestone documentation @ *2015-06-29 13:51:24 UTC*

  - 8eeaddbff4 Merge pull request #25022 from jfindlay/label_docs
  - 8575192cc4 revise label and milestone documentation
- **PR** #25029: (jayeshka) adding redismod unit test case. @ *2015-06-29 13:50:33 UTC*
  - 89c2e01ac1 Merge pull request #25029 from jayeshka/redismod-unit-test
  - e3045be5a9 adding redismod unit test case.
- **PR** #24995: (rallytime) Fix deprecated pymongo usage causing errors in latest pymongo @ *2015-06-27 22:28:56 UTC*
  - **PR** #24175: (trevor-h) fix deprecated pymongo usage causing errors in latest pymongo (refs: #24995)
  - 642525298c Merge pull request #24995 from rallytime/tops_mongo
  - a3c1063a37 fix deprecated pymongo usage causing errors in latest pymongo
- **ISSUE** #24862: (dkatsanikakis) gpg.import_key returns error after succesfully completed (refs: #24994, #24966)
- **PR** #24994: (garethgreenaway) Another Fix to gpg.py in 2015.5 @ *2015-06-27 22:28:15 UTC*
  - e9aaa11b68 Merge pull request #24994 from garethgreenaway/2015_5_24862_gpg_import_key
  - d2f0d8fa96 variable was referenced before assignment. Just removing the variable and checking the return from distutils.version.LooseVersion directly.
- **PR** #24988: (jayeshka) adding states/supervisord unit test case. @ *2015-06-27 22:24:42 UTC*
  - ebd666e5ee Merge pull request #24988 from jayeshka/supervisord-states-unit-test
  - bb0a6d5625 adding states/supervisord unit test case.
- **PR** #25007: (basepi) [2015.5] Merge forward from 2014.7 to 2015.5 @ *2015-06-26 21:28:57 UTC*
  - 0487c3c59b Merge pull request #25007 from basepi/merge-forward-2015.5
  - 4980fd547b Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.5
  - a11e4c6eea Merge pull request #24944 from techhat/issue24915
    - 59c3081e49 Double-check main_cloud_config
  - d26a5447ba Merge pull request #24936 from jtand/psutil
    - bdb7a19c36 Fixed ps module to not use depreciated psutil commands
- **PR** #25003: (jacobhammons) Updated man pages @ *2015-06-26 19:13:41 UTC*
  - 91a60e198e Merge pull request #25003 from jacobhammons/man-pages
  - cf97a4ab17 Updated man pages
- **PR** #25002: (jacobhammons) sphinx html theme updates @ *2015-06-26 18:39:14 UTC*
  - a60a2c4222 Merge pull request #25002 from jacobhammons/doc-announcements
  - f88f344a28 sphinx html theme updates
- **PR** #24977: (rallytime) Only warn about digital ocean deprecation if digital ocean is configured @ *2015-06-25 23:54:46 UTC*
  - a791b23ff9 Merge pull request #24977 from rallytime/do_move_warning
  - 6b544227ab Only warn about digital ocean deprecation if digital ocean is configured
- **ISSUE** #24862: (dkatsanikakis) gpg.import_key returns error after succesfully completed (refs: #24994, #24966)
- **PR** #24966: (garethgreenaway) Fixes to gpg.py in 2015.5 @ *2015-06-25 19:58:49 UTC*

  - a71c1b7c8b Merge pull request #24966 from garethgreenaway/2015_5_24862_gpg_import_key
  - 55eb73b0c9 fixing unit tests.
  - 80c24be4fe Fixing an issue with the import_key method. Different results depending on which gnupg python module is installed.
- **ISSUE** #24846: (mavenAtHouzz) Memory leak issue in rest_tornado EventListener (refs: #24965)
- **PR** #24965: (jacksontj) Fix memory leak in saltnado @ *2015-06-25 18:48:03 UTC*
  - 86221846ac Merge pull request #24965 from jacksontj/2015.5
  - 48b5e1653e pylint
  - 87adca46e0 Fix memory leak in saltnado
- **PR** #24948: (jfindlay) fix some malformed doc links and anchors @ *2015-06-25 15:51:38 UTC*
  - 773c4cf8e4 Merge pull request #24948 from jfindlay/doc_links
  - 152a9b2a12 fix some malformed doc links and anchors
- **ISSUE** #24885: (anlutro) Master config - Directories starting with a dot have the dot stripped when root_dir is . (refs: #24886)
- **PR** #24886: (anlutro) Be more careful about stripping away root_dir from directory options @ *2015-06-25 15:50:11 UTC*
  - 4ebc01e662 Merge pull request #24886 from alprs/fix-root_dir_bug
  - 52ccafded3 os.sep is the correct directory separator constant
  - 0ecbf261ad Be more careful about stripping away root_dir from directory options
- **PR** #24930: (jacksontj) Don't refetch file templates 100% of the time– Performance optimization for templated files @ *2015-06-24 21:22:47 UTC*
  - f52f7e1d20 Merge pull request #24930 from jacksontj/2015.5
  - 5fb75346ef Only parse the source if we have one
  - c03a6fa9d1 Add support for sources of managed files to be local
  - 4cf78a0a95 pylint
  - d70914e473 Don't refetch the template 100% of the time– Performance optimization for templated files
- **PR** #24935: (basepi) [2015.5] Merge forward from 2014.7 to 2015.5 @ *2015-06-24 18:17:54 UTC*
  - 925a4d91ba Merge pull request #24935 from basepi/merge-forward-2015.5
  - 8d8bf3476f Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.5
  - eeb05a1b10 Merge pull request #24918 from BretFisher/minion-start-smartos-smf-fix
    - d7bfb0c7fd Smartos smf minion fix
- **ISSUE** #24826: (rakai93) rh_service.py: 'int' object has no attribute 'startswith' (refs: #24873)
- **PR** #24873: (jfindlay) convert osrelease grain to str before str op @ *2015-06-24 16:43:08 UTC*
  - 4e8ed0d8ed Merge pull request #24873 from jfindlay/rh_service
  - febe6efab7 convert osrelease grain to str before str op
- **PR** #24923: (jayeshka) adding states/status unit test case. @ *2015-06-24 15:50:07 UTC*
  - 90819f9c37 Merge pull request #24923 from jayeshka/status-states-unit-test

  - baec650674 adding states/status unit test case.
- **PR** #24902: (cro) Fix minion failover, document same @ *2015-06-24 15:20:43 UTC*
  - 2dd24ece71 Merge pull request #24902 from cro/fixfo2
  - 90c73ff446 References to documentation.
  - f0c9204d8b Add references to failover parameters in conf
  - 9da96a8b95 Docs
  - e2314f0e49 Move comment.
  - b9a756ff5f Fix master failover and add documentation for same. Factor in syndics. Syndics will not failover (yet).
- **PR** #24926: (rallytime) Back-port #22263 to 2015.5 @ *2015-06-24 15:09:40 UTC*
  - **PR** #22263: (cachedout) Prevent a load from being written if one already exists (refs: #24926)
  - 087ee09f46 Merge pull request #24926 from rallytime/bp-22263
  - 8c92d9c677 Prevent a load from being written if one already exists
- **PR** #24900: (rallytime) Back-port #24848 to 2015.5 @ *2015-06-24 15:09:18 UTC*
  - **PR** #24848: (nmadhok) Correcting bash code blocks (refs: #24900)
  - b34a74fe89 Merge pull request #24900 from rallytime/bp-24848
  - d2b5456f5d Correcting bash code blocks
- **PR** #24899: (rallytime) Back-port #24847 to 2015.5 @ *2015-06-24 15:09:01 UTC*
  - **PR** #24847: (borutmrak) unset size parameter for lxc.create when backing=zfs (refs: #24899)
  - a546e8e326 Merge pull request #24899 from rallytime/bp-24847
  - 1e4ec7a56b unset size parameter for lxc.create when backing=zfs
- **PR** #24898: (rallytime) Back-port #24845 to 2015.5 @ *2015-06-24 15:06:09 UTC*
  - **PR** #24845: (porterjamesj) fix bug in docker.loaded (refs: #24898)
  - d4dd8d288d Merge pull request #24898 from rallytime/bp-24845
  - 071049ae7a fix bug in docker.loaded
- **ISSUE** #24799: (infestdead) Forced remount because options changed when no options changed (glusterfs) (refs: #24839)
- **PR** #24897: (rallytime) Back-port #24839 to 2015.5 @ *2015-06-24 15:05:35 UTC*
  - **PR** #24839: (infestdead) fix for issue #24799 (refs: #24897)
  - 693085520f Merge pull request #24897 from rallytime/bp-24839
  - f3b20d5445 fix for issue #24799
- **PR** #24891: (jayeshka) adding states/ssh_known_hosts unit test case. @ *2015-06-23 16:46:58 UTC*
  - 1650233be9 Merge pull request #24891 from jayeshka/ssh_known_hosts-states-unit-test
  - ef1347f2b3 adding states/ssh_known_hosts unit test case.
- **ISSUE** #24870: (dkiser) salt-cloud fails on sudo password prompt when using ssh key to auth (refs: #24874)
- **PR** #24874: (dkiser) Fix for salt-cloud when ssh key used to auth and using sudo. @ *2015-06-22 23:46:08 UTC*
  - c32aae96aa Merge pull request #24874 from dkiser/salt-cloud-24870

  - 6c31143b22 Fix key error for the PR to fix #24870.
  - bdcf7d88c1 Fix pylint for #24874.
  - 8f66d193e0 Fix for salt-cloud when ssh key used to auth and using sudo.
- **ISSUE** #24871: (dkiser) salt-cloud fails to honor 'password' in cloud options before raising an exception (refs: #24880)
- **PR** #24880: (dkiser) Fix to allow password for salt-cloud to be set outside of a vm specif… @ *2015-06-22 23:44:59 UTC*
  - ddaa21c0ae Merge pull request #24880 from dkiser/salt-cloud-24871
  - 4f6c035673 Fix to allow password for salt-cloud to be set outside of a vm specific context.
- **PR** #24852: (pruiz) Fix issue 24851: regular expression so it now matches packages with '.' or '-' at pkg name @ *2015-06-22 20:37:13 UTC*
  - 3902b162a9 Merge pull request #24852 from pruiz/issue-24851
  - 73adb1df50 Fix regular expression so it now matches packages with '.' or '-' at pkg name.
- **PR** #24861: (jayeshka) adding states/ssh_auth unit test case. @ *2015-06-22 16:20:01 UTC*
  - 6c5b788afd Merge pull request #24861 from jayeshka/ssh_auth-states-unit-test
  - e5d7b0de80 adding states/ssh_auth unit test case.
- **ISSUE** #23478: (calvinhp) grains.get virtual reports "physical" on bhyve FreeBSD VM (refs: #24824)
- **PR** #24824: (kev009) Detect bhyve virtual type for FreeBSD guests @ *2015-06-22 15:24:35 UTC*
  - 9e3321c18e Merge pull request #24824 from kev009/grains-bhyve-bsd
  - a2262097a1 Detect bhyve virtual type for freebsd guests
- **ISSUE** #24746: (anlutro) state.apply doesn't seem to work (refs: #24795)
- **PR** #24795: (anlutro) Fix state.apply for salt-ssh @ *2015-06-22 15:23:57 UTC*
  - 7b07ef9f44 Merge pull request #24795 from alprs/fix-salt_ssh_state_apply
  - 905840b1fa Fix state.apply for salt-ssh
- **PR** #24832: (jacksontj) Don't incur a "_load_all" of the lazy_loader while looking for mod_init. @ *2015-06-22 15:17:10 UTC*
  - **PR** #20540: (jacksontj) Loader nomerge: Don't allow modules to "merge" (refs: #24832)
  - **PR** #20481: (jacksontj) Add submodule support to LazyLoader (refs: #20540)
  - **PR** #20473: (jacksontj) Add "disabled" support (refs: #20481)
  - **PR** #20274: (jacksontj) Loader overhaul to LazyLoader (refs: #20473)
  - **PR** #12327: (jacksontj) Add a LazyLoader class which will lazily load modules (with the given lo… (refs: #20274)
  - 31d4c131e9 Merge pull request #24832 from jacksontj/2015.5
  - cfa7c0a699 pylint
  - be18439736 Don't incur a "_load_all" of the lazy_loader while looking for mod_init.
- **ISSUE** #14666: (luciddr34m3r) salt-cloud GoGrid exception when using map file (refs: #24811)
- **PR** #24834: (rallytime) Back-port #24811 to 2015.5 @ *2015-06-19 18:43:49 UTC*
  - **PR** #24811: (rallytime) Add notes to map and gogrid docs – don't use -P with map files (refs: #24834)

  - 2d8148fb4d Merge pull request #24834 from rallytime/bp-24811
  - e2684ecf0b Add notes to map and gogrid docs – don't use -P with map files
- **PR** #24790: (rallytime) Back-port #24741 to 2015.5 @ *2015-06-19 17:25:58 UTC*
  - **PR** #24741: (CameronNemo) Improve Upstart enable/disable handling (refs: #24790)
  - d2edb63cff Merge pull request #24790 from rallytime/bp-24741
  - a54245f080 Add missing import
  - 4ce6370d6e salt.modules.upstart: fix lint errors
  - aec53ec32a Improve Upstart enable/disable handling
- **PR** #24789: (rallytime) Back-port #24717 to 2015.5 @ *2015-06-19 17:17:00 UTC*
  - **PR** #24717: (gthb) virtualenv.managed: document user and no_chown (refs: #24789)
  - 645e62a43c Merge pull request #24789 from rallytime/bp-24717
  - 95ac4eba13 virtualenv.managed: document user and no_chown
- **PR** #24823: (jayeshka) adding states/splunk_search unit test case. @ *2015-06-19 17:14:12 UTC*
  - 0a6c70f062 Merge pull request #24823 from jayeshka/splunk_search-states-unit-test
  - 98831a8cb0 adding states/splunk_search unit test case.
- **PR** #24809: (jodv) Correctly create single item list for failover master type with string value for master opt @ *2015-06-19 15:22:20 UTC*
  - 4c5a708599 Merge pull request #24809 from jodv/single_item_master_list
  - 18ceebc77f single item list vs. list of characters
- **PR** #24802: (basepi) [2015.5] Merge forward from 2014.7 to 2015.5 @ *2015-06-18 20:11:58 UTC*
  - ae05e70e94 Merge pull request #24802 from basepi/merge-forward-2015.5
  - 5b7a65d6d9 Merge pull request #19 from twangboy/merge-forward-fixes
    - 98e7e90299 Fixed test failures for Colton
  - b949856ae6 Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.5
    - 4281dfff0b Merge pull request #24780 from nmadhok/backport-2014.7-24777
      - c53b0d9a22 Backporting PR #24777 to 2014.7 branch
    - f3c5cb2d41 Merge pull request #24769 from msteed/issue-21318
      - f40a9d5cc0 Fix stacktrace in get_cli_returns()
    - 59db24602f Merge pull request #24690 from twangboy/fix_17041
      - 7a015389af Added additional reporting
      - d84ad5d519 Fixed capitalization… Failed and Already
      - e9552455c4 Merge branch '2014.7' of https://github.com/saltstack/salt into fix_17041
      - 144bff2f67 Report powershell output instead of error
  - **PR** saltstack/salt#24329: (jayeshka) adding states/postgres_database unit test case. (refs: #24798)
- **PR** #24798: (justinta) Revert "adding states/postgres_database unit test case." @ *2015-06-18 17:56:17 UTC*
  - daa76c34e4 Merge pull request #24798 from saltstack/revert-24329-postgres_database-states-unit-test

  - 179ce03d93 Revert "adding states/postgres_database unit test case."
- **PR** #24791: (rallytime) Back-port #24749 to 2015.5 @ *2015-06-18 17:43:15 UTC*
  - **PR** #24749: (obestwalter) add windows specfic default for multiprocessing (refs: #24791)
  - 7073a9f850 Merge pull request #24791 from rallytime/bp-24749
  - be43b2b394 add windows specfic default for multiprocessing
- **PR** #24792: (rallytime) Back-port #24757 to 2015.5 @ *2015-06-18 15:58:35 UTC*
  - **PR** #24757: (cachedout) Fix loader call in pyobjects (refs: #24792)
  - **PR** #24668: (grischa) enable virtual package names in pyobjects renderer (refs: #24721, #24757)
  - 1a158e8a3b Merge pull request #24792 from rallytime/bp-24757
  - 6c804f0789 Fix loader call in pyobjects
- **PR** #24768: (jfindlay) fix yum versionlock on RHEL/CentOS 5, disable corresponding test @ *2015-06-18 15:13:12 UTC*
  - 0f9298263b Merge pull request #24768 from jfindlay/pkg_mod
  - 7a26c2b5b9 disable pkg.hold test for RHEL/CentOS 5
  - 4cacd93c22 use correct yum versionlock pkg name on centos 5
- **ISSUE** #24776: (nmadhok) –static option in salt raises ValueError and has been broken for a very long time (refs: #24777)
  - **PR** #24779: (nmadhok) Backporting Changes to 2014.7 branch (refs: #24777)
- **PR** #24778: (nmadhok) Backporting PR #24777 to 2015.2 branch (refs: #24777) @ *2015-06-18 14:53:04 UTC*
  - **PR** #24777: (nmadhok) Fixing issue where –static option fails with ValueError Fixes #24776 (refs: #24778, #24780)
  - 39f088a74c Merge pull request #24778 from nmadhok/backport-2015.2-24777
  - ae3701f639 Backporting PR #24777 to 2015.2 branch
- **PR** #24774: (zefrog) Fix lxc lvname parameter command @ *2015-06-18 14:49:06 UTC*
  - 2a4f65f3f7 Merge pull request #24774 from zefrog/fix-lxc-lvname-param
  - 21e0cd4a5e Fixed typo in lxc module: lvname parameter typo
  - 283d86ec12 Fixed bug in lxc module: lvname using wrong parameter in cmd
- **PR** #24782: (jayeshka) adding states/slack unit test case. @ *2015-06-18 14:33:55 UTC*
  - fd7339014b Merge pull request #24782 from jayeshka/slack-states-unit-test
  - e2b6214764 adding states/slack unit test case.
- **ISSUE** #24770: (jacksontj) *Requisite* and *Requisite_in* don't play nice together (refs: #24771)
- **PR** #24771: (jacksontj) Always extend requisites, instead of replacing them @ *2015-06-18 14:29:09 UTC*
  - c9c90af512 Merge pull request #24771 from jacksontj/2015.5
  - b1211c5422 Re-enable tests for complex prereq and prereq_in
  - 378f6bfc36 Only merge when the merge is of requisites
- **PR** #24766: (msteed) Remove doc references to obsolete minion opt @ *2015-06-17 21:36:55 UTC*
  - 5fe4de8f62 Merge pull request #24766 from msteed/undoc-dns_check

  - f92a769d35 Remove doc references to obsolete minion opt
- **PR** #24329: (jayeshka) adding states/postgres_database unit test case. @ *2015-06-17 19:11:02 UTC*
  - a407ab7c51 Merge pull request #24329 from jayeshka/postgres_database-states-unit-test
  - ee06f1ad57 adding states/postgres_database unit test case.
- **ISSUE** #24560: (hydrosine) Documentation missing on parameter (refs: #24632)
- **ISSUE** #24547: (dragonpaw) Artifactory docs say module is 'jboss7'. (refs: #24632)
- **ISSUE** #24375: (companykitchen-dev) Custom grain won't sync under any circumstances (refs: #24632)
- **ISSUE** #24275: (kartiksubbarao) augeas issue with apache and recognizing changes that have been already made (refs: #24632)
- **ISSUE** #24163: (tbaker57) enable_gpu_grains default value confusion (refs: #24632)
- **PR** #24632: (jacobhammons) Doc bug fixes @ *2015-06-17 18:40:02 UTC*
  - 3ff6eff546 Merge pull request #24632 from jacobhammons/bug-fixes
  - 7c52012e31 Fixed typos
  - c7cdd416a2 Doc bug fixes Refs #24547 Refs #24275 Refs #24375 Refs #24560 Refs #24163
- **ISSUE** #24198: (ahammond) salt-call event.send doesn't send events from minion (refs: #24607)
- **PR** #24607: (garethgreenaway) fixes to minion.py @ *2015-06-17 18:16:42 UTC*
  - 9995f64428 Merge pull request #24607 from garethgreenaway/2015_5_sending_events_multi_master
  - 8abd3f0ee1 A fix if you have multiple masters configured and try to fire events to the minion. Currently they fail silently. Might be the cause of #24198.
- **PR** #24755: (rallytime) Remove SALT_CLOUD_REQS from setup.py @ *2015-06-17 17:42:25 UTC*
  - bf2dd94389 Merge pull request #24755 from rallytime/fix_setup_15
  - 48769a544d Remove SALT_CLOUD_REQS from setup.py
- **PR** #24740: (rallytime) Backport #24720 to 2015.5 @ *2015-06-17 16:43:37 UTC*
  - **PR** #24720: (TheScriptSage) Issue 24621 - AD/LDAP Group Auth Issue (refs: #24740)
  - 3d53d79476 Merge pull request #24740 from rallytime/bp-24720
  - a9bcdb5b77 Updating master.py to properly check against groups when user is only authed against group. Tested against unit.auth_test.
- **PR** #24723: (rallytime) Back-port #20124 to 2015.5 @ *2015-06-17 16:43:20 UTC*
  - **PR** #20124: (cgtx) add init system to default grains (refs: #24723)
  - ac2851be55 Merge pull request #24723 from rallytime/bp-20124
  - 4d0061b832 fix infinite loop introduced by #20124 when the init system is not in the supported_inits list
  - 0c7fa0fca2 Optimizations for #20124
  - f353454327 add init system to default grains (resolve #20124)
- **PR** #24754: (anlutro) salt-cloud documentation - Add information about linode location @ *2015-06-17 16:04:48 UTC*
  - 78cd09b6e9 Merge pull request #24754 from alprs/docs-add_linode_location_option
  - d88e071e98 add information about linode location

- **PR** #24748: (jayeshka) adding states/serverdensity_device unit test case. @ *2015-06-17 15:39:07 UTC*
  - d5554f76ec Merge pull request #24748 from jayeshka/serverdensity_device-states-unit-test
  - 1a4c241050 adding states/serverdensity_device unit test case.
- **PR** #24739: (rallytime) Back-port #24735 to 2015.5 @ *2015-06-17 15:16:47 UTC*
  - **PR** #24735: (notpeter) Add 2015.5 codename to version numbers docs (refs: #24739)
  - 0b7e7ef879 Merge pull request #24739 from rallytime/bp-24735
  - 64c565d9be Add .0 to version number
  - 5ed801b98f Add codenames for 2015.5 and future versions. Trailing newline.
- **ISSUE** #24111: (yermulnik) cli option '–summary' got broken after upgrade to 2015.5.1 (refs: #24732)
- **PR** #24732: (msteed) Fix stacktrace when *–summary* is used @ *2015-06-17 03:27:57 UTC*
  - c8713f2d00 Merge pull request #24732 from msteed/issue-24111
  - 54b33dd359 Fix stacktrace when –summary is used
- **PR** #24721: (rallytime) Back-port #24668 to 2015.5 @ *2015-06-17 03:23:47 UTC*
  - **PR** #24668: (grischa) enable virtual package names in pyobjects renderer (refs: #24721, #24757)
  - 70d37816bf Merge pull request #24721 from rallytime/bp-24668
  - 68fb5af970 fixing other test
  - ba4f262b9c fixing text for virtual support in pyobjects
  - b349d91a5f enable virtual package names in pyobjects renderer
- **ISSUE** #21923: (Fluro) Salt cloud not running provisioning script as root (refs: #24718)
- **ISSUE** #17241: (hasues) Salt-Cloud for vSphere needs additional documentation (refs: #24718)
- **PR** #24718: (rallytime) Added some missing config documentation to the vsphere driver @ *2015-06-17 03:19:35 UTC*
  - 1b9d6895c7 Merge pull request #24718 from rallytime/update_vsphere_docs
  - bfdebb6e18 Added some missing config documentation to the vsphere driver
- **PR** #24714: (rallytime) Remove cloud-requirements.txt @ *2015-06-17 03:17:04 UTC*
  - 64857c706d Merge pull request #24714 from rallytime/remove_cloud_reqs_15
  - 67b796d01e Remove cloud-requirements.txt
- **ISSUE** #24439: (bechtoldt) Add tornado version to versions report (refs: #24733)
- **PR** #24733: (msteed) Include Tornado in versions report @ *2015-06-17 03:13:53 UTC*
  - f96b1d68cd Merge pull request #24733 from msteed/issue-24439
  - 76cfef05ec Include Tornado in versions report
- **PR** #24737: (jacksontj) Move AES command logging to trace @ *2015-06-17 01:48:11 UTC*
  - a861fe0f4f Merge pull request #24737 from jacksontj/2015.5
  - a4ed41ae82 Move AES command logging to trace
- **PR** #24724: (basepi) [2015.5] Merge forward from 2014.7 to 2015.5 @ *2015-06-16 22:46:27 UTC*
  - 0d2dc46648 Merge pull request #24724 from basepi/merge-forward-2015.5

  - 4641028464 Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.5
  - a18dadad71 Merge pull request #24646 from twangboy/fix_24196
    - a208e1d60f Fixed user.present on existing user
- **PR** #24701: (jayeshka) adding states/selinux unit test case. @ *2015-06-16 15:27:29 UTC*
  - 3d33fe7676 Merge pull request #24701 from jayeshka/selinux-states-unit-test
  - 0c136fd9c2 adding states/selinux unit test case.
- **PR** #24687: (cachedout) Note about minimum worker_threads @ *2015-06-15 20:46:23 UTC*
  - 2e287a9e33 Merge pull request #24687 from cachedout/min_worker_threads
  - b7bb7eaeb2 Note about minimum worker_threads
- **PR** #24688: (cachedout) Update AUTHORS @ *2015-06-15 20:46:03 UTC*
  - 432478ccb7 Merge pull request #24688 from cachedout/update_authors
  - 3f6880e291 Better email
  - 6c7b773eae Update AUTHORS
- **ISSUE** #22385: (cachedout) States which require unavailable modules should display the reason (refs: #24649)
- **PR** #24649: (cachedout) Improved error reporting for failed states @ *2015-06-15 16:04:20 UTC*
  - 9a2b50d59f Merge pull request #24649 from cachedout/issue_22385
  - b9fe792534 States will now return the reason behind failure if a module could not be loaded
- **PR** #24673: (jayeshka) adding states/schedule unit test case. @ *2015-06-15 15:24:52 UTC*
  - 66e9e16753 Merge pull request #24673 from jayeshka/schedule-states-unit-test
  - 54aaaa5f12 adding states/schedule unit test case.
- **ISSUE** #24661: (kartiksubbarao) augeas.change doesn't support setting empty values (refs: #24663)
- **PR** #24663: (kartiksubbarao) Update augeas_cfg.py @ *2015-06-15 15:18:48 UTC*
  - 5eb19c4e4d Merge pull request #24663 from kartiksubbarao/patch-2
  - e18db50e0c Update augeas_cfg.py
- **ISSUE** #24583: (dkiser) salt-cloud keyring password referenced before assignment (refs: #24667)
- **PR** #24667: (dkiser) fix for #24583 clouds/openstack.py kerying first time succeeds @ *2015-06-14 21:58:58 UTC*
  - 4450432161 Merge pull request #24667 from dkiser/fix-cloud-keyring
  - c92c05fac0 fix for #24583 clouds/openstack.py kerying first time succeeds
- **ISSUE** #24537: (kartiksubbarao) alias.present doesn't update alias values that are substrings of the existing value (refs: #24659)
- **PR** #24659: (kartiksubbarao) Update aliases.py @ *2015-06-13 17:31:42 UTC*
  - 4c64ee9d94 Merge pull request #24659 from kartiksubbarao/patch-1
  - d6834749e2 Update aliases.py
- **PR** #24644: (cro) Merge forward 2014.7->2015.5 @ *2015-06-12 21:31:41 UTC*
  - 89eb616c29 Merge pull request #24644 from cro/2014.7-2015.5-20150612
  - 4136dc3160 Merge forward from 2014.7 to 2015.5

  - b99484fde2 Merge pull request #24643 from cro/saltannounce
    - ecb0623d7f Add salt-announce mailing list.
  - 635121e85d Merge pull request #24620 from twangboy/fix_24215
    - d7a9999be1 Fixed comment and uncomment functions in file.py
  - **PR** saltstack/salt#24595: (tankywoo) fix target rule, remove unneeded quotation mark (refs: #24642)
- **PR** #24642: (basepi) Revert "fix target rule, remove unneeded quotation mark" @ *2015-06-12 20:14:26 UTC*
  - b896a0d0e9 Merge pull request #24642 from saltstack/revert-24595-fix-iptables-target
  - 5ff3224ae1 Revert "fix target rule, remove unneeded quotation mark"
- **PR** #24628: (jayeshka) adding states/reg unit test case. @ *2015-06-12 17:29:11 UTC*
  - 01092c2337 Merge pull request #24628 from jayeshka/reg_states-unit-test
  - af1bd8f9ff adding states/reg unit test case.
- **ISSUE** #24494: (arount) Computed comments in jinja states (refs: #24591)
- **ISSUE** #23359: (BalintSzigeti) init.sls parsing issue (refs: #24591)
- **ISSUE** #21217: (Colstuwjx) Maybe a bug for jinja render? (refs: #24591)
- **PR** #24631: (rallytime) Back-port #24591 to 2015.5 @ *2015-06-12 16:54:32 UTC*
  - **PR** #24591: (tbaker57) Add some documentation surrounding Jinja vs yaml comments - (refs: #24631)
  - 5f491f911d Merge pull request #24631 from rallytime/bp-24591
  - f13cd418bc Add extra clarification why jinja comments are needed.
  - 23749718bb Fix typo
  - 6a917471d4 Add some documentation surrounding Jinja comments - refs #24492, #21217, #23359
- **PR** #24616: (garethgreenaway) additional logging in state.py module @ *2015-06-12 16:25:39 UTC*
  - f23f99ec35 Merge pull request #24616 from garethgreenaway/2015_5_logging_disabled_states
  - 4dbf0ef160 Adding some logging statement to give feedback when states, including highstate, are disabled. Useful when running from scheduler.
- **PR** #24595: (tankywoo) fix target rule, remove unneeded quotation mark @ *2015-06-12 16:23:22 UTC*
  - 6dccbb04a1 Merge pull request #24595 from tankywoo/fix-iptables-target
  - 10a5160d7c fix target rule, remove unneeded quotation mark
- **PR** #24604: (jfindlay) fix pkg module integration tests @ *2015-06-12 16:04:26 UTC*
  - 8ac3d94785 Merge pull request #24604 from jfindlay/pkg_tests
  - d88fb22fdc fix pkg module integration tests on CentOS 5
  - fb91b40ba0 fix pkg module integration tests on ubuntu 12
- **PR** #24600: (basepi) [2015.5] Remove __kwarg__ from salt-ssh keyword args @ *2015-06-12 04:21:29 UTC*
  - 0ff545c549 Merge pull request #24600 from basepi/salt-ssh.orchestrate.20615
  - 9b55683f6a Remove __kwarg__ from salt-ssh keyword args
- **ISSUE** #22843: (Xiol) salt-ssh roster doesn't support integers as host keys (refs: #24608)
- **PR** #24608: (basepi) [2015.5] Normalize salt-ssh flat roster minion IDs to strings @ *2015-06-11 21:35:07 UTC*

  - 832916f49f Merge pull request #24608 from basepi/salt-ssh.flat.roster.integers.22843
  - 381820f051 Normalize salt-ssh flat roster minion IDs to strings
- **PR** #24605: (basepi) [2015.5] Merge forward from 2014.7 to 2015.5 @ *2015-06-11 19:15:21 UTC*
  - 4eb5bb253b Merge pull request #24605 from basepi/merge-forward-2015.5
  - f96c5029bb Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.5
  - d83928a7f9 Merge pull request #24589 from BretFisher/patch-1
    - 65a11336dc Fixed Mine example for jinja code block
- **ISSUE** #24457: (ryan-lane) When selecting the version of docs on the docs site, it brings you to the homepage (refs: #24598)
- **ISSUE** #24250: (jfindlay) have version links on docs page link to that version of the current page (refs: #24598)
- **PR** #24598: (jacobhammons) 2015.5.2 release changes @ *2015-06-11 17:24:11 UTC*
  - e0bb177823 Merge pull request #24598 from jacobhammons/doc-fixes
  - f3f34ddff6 2015.5.2 release changes Refs #24250 Refs #24457
- **ISSUE** #20615: (aurynn) 2014.7.1: salt/states/saltmod using incorrect return dict for orchestrate (refs: #24588)
- **PR** #24588: (basepi) Fixes for saltmod.function for salt-ssh @ *2015-06-11 16:15:21 UTC*
  - 26930b45bd Merge pull request #24588 from basepi/salt-ssh.orchestrate.20615
  - 826936ce57 Move documentation into docstring instead of comments
  - de052e7135 Assign 'return' to 'ret' if necessary in saltmod.function
  - 34ff989d66 Convert keyword args to key=value strings in salt-ssh
- **PR** #24593: (jayeshka) adding states/redismod unit test case. @ *2015-06-11 15:55:27 UTC*
  - 5a21ad152e Merge pull request #24593 from jayeshka/redismod_states-unit-test
  - 3b95744840 adding states/redismod unit test case.
- **ISSUE** #40: (thatch45) Clean up timeouts (refs: #22857)
- **PR** #24581: (rallytime) Disabled some flaky tests until we can figure out how to make them more reliable @ *2015-06-11 15:51:41 UTC*
  - **PR** #24217: (jfindlay) disable intermittently failing tests (refs: #24581)
  - **PR** #23623: (jfindlay) Fix /jobs endpoint's return (refs: #24217)
  - **PR** #22857: (jacksontj) Fix /jobs endpoint's return (refs: #23623)
  - 8ffb86edd0 Merge pull request #24581 from rallytime/disable_some_flaky_tests
  - c82f135d2e Disabled some flaky tests until we can figure out how to make them more reliable
- **PR** #24566: (jayeshka) adding states/rdp unit test case. @ *2015-06-11 02:14:39 UTC*
  - a570d7f967 Merge pull request #24566 from jayeshka/rdp_states-unit-test
  - 273b994e91 adding states/rdp unit test case.
- **ISSUE** #24480: (kiorky) [CRITICAL] [2015.5] tls breaks tzinfo (refs: #24551)
- **PR** #24551: (joejulian) 2015.5 dont pollute environment @ *2015-06-11 02:13:06 UTC*
  - 20ada1f8a1 Merge pull request #24551 from joejulian/2015.5_dont_pollute_environment
  - cfc3b43ba2 Don't pollute the TZ environment variable

  - cba8d3f923 pep8
  - 9cb7015568 Mark keyword version adds
  - 76e2583265 Merge tls changes from develop
- **ISSUE** #19901: (clinta) State cache is not documented (refs: #24574, #24468)
- **PR** #24574: (jacobhammons) Refs #19901 @ *2015-06-10 20:09:23 UTC*
  - bb2fd6a970 Merge pull request #24574 from jacobhammons/19901
  - e2a2946dc7 Refs #19901
- **PR** #24577: (basepi) [2015.5] Merge forward from 2014.7 to 2015.5 @ *2015-06-10 19:46:22 UTC*
  - b03166cde3 Merge pull request #24577 from basepi/merge-forward-2015.5
  - e1d45ccf3b Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.5
  - d376390f76 Merge pull request #24530 from twangboy/fix_24427
    - 673e1d809e Added missing panel.bmp for installer
    - cc50218b01 Start Minion Service on Silent Install
- **ISSUE** #24235: (tomasfejfar) Difference between running from minion and from master (refs: #24571, #24468)
- **PR** #24571: (jacobhammons) Refs #24235 @ *2015-06-10 17:02:18 UTC*
  - 3ec457beef Merge pull request #24571 from jacobhammons/24235
  - 8df5d53bb8 Refs #24235
- **PR** #24565: (pille) fix backtrace, when listing plugins @ *2015-06-10 16:33:11 UTC*
  - fe07eb5653 Merge pull request #24565 from pille/munin-ignore-broken-symlinks
  - 8511a6c0a6 fix backtrace, when listing plugins
- **PR** #24554: (ryan-lane) Fix yes usage for pecl defaults @ *2015-06-09 23:59:49 UTC*
  - 251c8f9f5f Merge pull request #24554 from lyft/pecl-module-fix
  - 56a9cfcf24 Fix yes usage for pecl defaults
- **PR** #24535: (rallytime) Back-port #24518 to 2015.5 @ *2015-06-09 20:06:18 UTC*
  - **PR** #24518: (rallytime) Merge #24448 with Pylint Fixes (refs: #24535)
  - **PR** #24448: (codertux) Update modules path for operating systems using systemd (refs: #24518)
  - dbd49b4acb Merge pull request #24535 from rallytime/bp-24518
  - fc75197616 Pylint fix
  - 3e08840988 Update modules path for operating systems using systemd
- **PR** #24538: (basepi) [2015.5] Merge forward from 2014.7 to 2015.5 @ *2015-06-09 17:27:20 UTC*
  - 485ed3cff9 Merge pull request #24538 from basepi/merge-forward-2015.5
  - 6a8039d468 Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.5
  - 6ebc476bb3 Merge pull request #24513 from jquast/2014.7-bugfix-iteritem
    - 2be0180e5e bugfix use of 'iteritem' in 2014.7 branch
- **PR** #24495: (jayeshka) adding states/rabbitmq_vhost unit test case. @ *2015-06-09 15:33:23 UTC*
  - 73e6388acd Merge pull request #24495 from jayeshka/rabbitmq_vhost_states-unit-test

  - 31889e38eb cosmetic change.
  - cf501cf60d resolved error.
  - 4bb6087722 Merge branch '2015.5' of https://github.com/saltstack/salt into rabbitmq_vhost_states-unit-test
  - 3ad77143a8 adding states/rabbitmq_vhost unit test case.
- **PR** #24445: (jayeshka) adding states/pyrax_queues unit test case. @ *2015-06-09 15:28:45 UTC*
  - bf1abccebe Merge pull request #24445 from jayeshka/pyrax_queues_states-unit-test
  - ea27cefb10 adding states/pyrax_queues unit test case.
- **PR** #24490: (aneeshusa) Fix pacman.list_upgrades for new python_shell default. @ *2015-06-09 15:13:16 UTC*
  - 0247e8d10d Merge pull request #24490 from aneeshusa/fix-pacman-list-upgrades
  - 980e1cb4dc Lint fix.
  - dca33f1112 Fix pacman.list_upgrades for new python_shell default.
- **PR** #24517: (steverweber) small fixes to the ipmi docs @ *2015-06-09 15:10:14 UTC*
  - 6268ddb43a Merge pull request #24517 from steverweber/ipmi_doc
  - 6413712844 lint
  - e78aea9b01 more small fixes to the ipmi docs
- **PR** #24524: (jayeshka) any() takes list oy tuple. @ *2015-06-09 13:49:42 UTC*
  - 3728b3f327 Merge pull request #24524 from jayeshka/rabbitmq_vhost_states-module
  - 01c99ad767 any() takes list oy tuple.
- **PR** #24482: (eliasp) 'docker.running' needs now the 'image' param. @ *2015-06-09 04:43:04 UTC*
  - dd23de885b Merge pull request #24482 from eliasp/2015.5-states.dockerio-docker.running-doc
  - 5de741d626 'docker.running' needs now the 'image' param.
- **ISSUE** #23503: (jfindlay) salt-ssh fails on CentOS 7 when python-zmq is not installed (refs: #24515)
- **PR** #24515: (basepi) [2015.5] Add xml library to the salt-thin @ *2015-06-09 04:10:06 UTC*
  - 2a727c3f55 Merge pull request #24515 from basepi/susexml23503
  - 078b33eaaf Add xml library to the thin
- **PR** #24497: (jayeshka) adding states/rbenv unit test case. @ *2015-06-09 03:56:10 UTC*
  - fce998a58b Merge pull request #24497 from jayeshka/rbenv_states-unit-test
  - 79d343a62b adding states/rbenv unit test case.
- **PR** #24496: (jayeshka) adding states/rabbitmq_user unit test case. @ *2015-06-09 03:55:23 UTC*
  - 2bcb4b1eed Merge pull request #24496 from jayeshka/rabbitmq_user_states-unit-test
  - 7d96f27f91 adding states/rabbitmq_user unit test case.
- **PR** #24481: (eliasp) Fix typo (licnese → license). @ *2015-06-09 03:30:25 UTC*
  - 02a597bf49 Merge pull request #24481 from eliasp/2015.5-salt.states.powerpath-license_typo
  - 1280054bce Fix typo (licnese → license).
- **PR** #24467: (thenewwazoo) Fix dockerio bound volumes @ *2015-06-09 01:40:23 UTC*

  - 5ad3db5ffb Merge pull request #24467 from thenewwazoo/fix-dockerio-bound-volumes
  - db4e3dc69b Let's raise an exception if create fails
  - d1d85dd685 Add logging
  - ddc63f0f30 Fix volume handling when creating containers
- **PR** #24504: (rallytime) Move vsphere deprecation to 2015.5 @ *2015-06-08 22:43:05 UTC*
  - **PR** #24487: (nmadhok) Deprecating vsphere cloud driver in favor of vmware cloud driver (refs: #24504)
  - d236fbd38f Merge pull request #24504 from rallytime/move_vsphere_deprecation_2015.5
  - d876535d71 Add Getting Started with VSphere doc to 2015.5
  - b685ebc104 Add vSphere deprecation warnings to 2015.5
- **PR** #24506: (rallytime) Backport #24450 to 2015.5 @ *2015-06-08 22:42:14 UTC*
  - **PR** #24450: (ruzarowski) Fix salt cli runs with batch-size set (refs: #24506)
  - cb5546085c Merge pull request #24506 from rallytime/bp-24450
  - 1c0fca2b9d Backport #24450 to 2015.5
- **PR** #24498: (rallytime) Added "CLI Example" to make failing test happy on 2015.5 @ *2015-06-08 15:48:40 UTC*
  - 3173fd17ad Merge pull request #24498 from rallytime/fix_doc_failure_fifteen
  - d992ef4777 Added "CLI Example" to make failing test happy on 2015.5
- **PR** #24471: (anlutro) Set up salt-ssh file logging @ *2015-06-08 15:26:49 UTC*
  - 3639e411bd Merge pull request #24471 from alprs/fix-salt_ssh_logging
  - 6a11ec87b8 set up salt-ssh file logging
- **ISSUE** #24231: (tarwich) npm.bootstrap (refs: #24469)
- **PR** #24469: (jfindlay) correctly handle user environment info for npm @ *2015-06-08 15:26:02 UTC*
  - 551e70f3fb Merge pull request #24469 from jfindlay/npm_env
  - 8140c96949 update npm's user info envs
  - cb572f8c41 add *env* parameter to npm.uninstall
- **ISSUE** #24268: (tkent-xetus) Ability to specify revision for win_gitrepos undocumented (refs: #24468)
- **ISSUE** #24235: (tomasfejfar) Difference between running from minion and from master (refs: #24571, #24468)
- **ISSUE** #24193: (abng88) Update ext_pillar docs to mention that this feature is supported masterless as well (refs: #24468)
- **ISSUE** #24172: (zhujinhe) Can lists be passed in the pillar on the command line on version 2015.5.0? (refs: #24468)
- **ISSUE** #23211: (lloesche) Document that salt://| escapes special characters in filenames (refs: #24468)
- **ISSUE** #19901: (clinta) State cache is not documented (refs: #24574, #24468)
- **ISSUE** #19801: (ksalman) How are grains static? (refs: #24468)
- **PR** #24468: (jacobhammons) Bug fixes and build errors @ *2015-06-08 15:25:40 UTC*
  - 0d9e0c2b8c Merge pull request #24468 from jacobhammons/doc-fixes
  - 1035959459 Appended .0 to version added

  - d45c4ed11f Bug fixes and build errors Refs #23211 Refs #24268 Refs #24235 Refs #24193 Refs #24172 Refs #19901 Refs #19801
- **ISSUE** #24318: (favadi) uncaught exception for pkgrepo.absent for invalid PPA (refs: #24465)
- **PR** #24465: (jfindlay) catch exception from softwarerepositories @ *2015-06-08 15:25:19 UTC*
  - be6905a545 Merge pull request #24465 from jfindlay/unknown_ppa
  - 19c912866d catch exception from softwarerepositories
- **ISSUE** #24296: (objectx) mount.mount calls file.mkdir with incorrect named argument (refs: #24464)
- **PR** #24464: (jfindlay) fix typo in modules/mount.py @ *2015-06-08 15:25:07 UTC*
  - 58d1ea8fe8 Merge pull request #24464 from jfindlay/file_mkdir
  - 6e8cd44500 fix typo in modules/mount.py
- **ISSUE** #24434: (dkiser) multimaster failover fails due to logic from issue #23611 (refs: #24461)
- **PR** #24461: (dkiser) fix for #24434 @ *2015-06-08 15:24:53 UTC*
  - 4f332a71c6 Merge pull request #24461 from dkiser/multimaster_minion_fix
  - 1944a743d7 fix for #24434
- **PR** #24479: (ahus1) change "path" to "name" for "file" operations @ *2015-06-07 17:56:11 UTC*
  - 8917416d39 Merge pull request #24479 from ahus1/patch-1
  - 7d6b60c79d change "path" to "name" for "file" operations
- **PR** #24475: (rallytime) Back-port #24454 to 2015.5 @ *2015-06-07 01:29:32 UTC*
  - **PR** #24454: (rhertzog) Strip extraneous newline character added in last environment variable (refs: #24475)
  - 8618d5b6ea Merge pull request #24475 from rallytime/bp-24454
  - a793c192a6 Avoid extraneous newline character added in last environment variable
- **ISSUE** #24407: (aboe76) Please expand salt module random (refs: #24420)
- **PR** #24474: (rallytime) Back-port #24420 to 2015.5 @ *2015-06-07 01:29:11 UTC*
  - **PR** #24420: (aboe76) added random integer module to mod_random.py (refs: #24474)
  - 61658ffef7 Merge pull request #24474 from rallytime/bp-24420
  - 4219b404ad Fix lint error and update versionadded to 2015.5.3
  - 3613cc9659 added random integer module to mod_random.py
- **ISSUE** #24233: (variia) yumpkg.group_install keeps returning state change
- **PR** #24472: (variia) ensure {} output is not treated as change in module.py state, fixes #… @ *2015-06-06 14:45:44 UTC*
  - 508d7ddb91 Merge pull request #24472 from variia/Fix-yumpkg_group_install-return-change-#24233
  - 37e8827ce8 ensure {} output is not treated as change in module.py state, fixes #24233
- **ISSUE** #8585: (UtahDave) '#' in single quoted option on cli not making it into the execution module (refs: #24466)
- **ISSUE** #18045: (dstokes) Pillar kwargs parse error with # (refs: #24466)
- **PR** #24466: (basepi) [2015.5] Fix for # in inner strings in yaml arguments @ *2015-06-06 14:35:56 UTC*

  - 0292e67c8a Merge pull request #24466 from basepi/fixhashinargs18045
  - 2e0609f09e Fix for # in inner strings in yaml arguments
- **PR** #24456: (rallytime) Back-port #24441 to 2015.5 @ *2015-06-05 22:32:25 UTC*
  - **PR** #24441: (arthurlogilab) [doc] Alignement fix on external_auth documentation (refs: #24456)
  - ced558a6e6 Merge pull request #24456 from rallytime/bp-24441
  - 70028553c1 yaml indentations should be 2 spaces
  - 21b51abf25 [doc] Alignement fix on external_auth documentation
- **ISSUE** #24397: (kiorky) on debian: states.apt should use virtualname as it shadows system apt module (refs: #24398, #24400, #24399)
  - **PR** #24399: (kiorky) Versionvirtual (refs: #24398)
- **PR** #24398: (kiorky) VirtualName for states.apt (refs: #24399) @ *2015-06-05 17:40:04 UTC*
  - c0ff4110ab Merge pull request #24398 from makinacorpus/aptv
  - 785d27707f VirtualName for states.apt
- **PR** #24447: (jayeshka) adding states/rabbitmq_policy unit test case. @ *2015-06-05 15:26:11 UTC*
  - 36263405be Merge pull request #24447 from jayeshka/rabbitmq_policy_states-unit-test
  - 9b038abd63 adding states/rabbitmq_policy unit test case.
- **PR** #24446: (jayeshka) adding states/rabbitmq_plugin unit test case. @ *2015-06-05 15:25:33 UTC*
  - 8445a3f28d Merge pull request #24446 from jayeshka/rabbitmq_plugin_states-unit-test
  - cb0c99a012 adding states/rabbitmq_plugin unit test case.
- **PR** #24426: (basepi) [2015.5] Merge forward from 2014.7 to 2015.5 @ *2015-06-05 03:59:11 UTC*
  - 9cc3808758 Merge pull request #24426 from basepi/merge-forward-2015.5
  - eafa20cdfb Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.5
    - 83f853b6ea Merge pull request #24405 from jacksontj/2014.7
      - 2c7afaeebf Fix for #24276
    - cef919c602 Merge pull request #24395 from hvnsweeting/handle-exception-get-file
      - bb798a0224 handle exceptions when received data is not in good shape
    - efba1a94b4 Merge pull request #24305 from twangboy/win_path_docs
    - 36804253e6 Fixed pylint error caused by P… added r
    - bc42a4bb11 triple double quotes to triple single quotes
    - 77cd930bba Added documentation, fixed formatting
- **ISSUE** #24309: (steverweber) missing docs (refs: #24429)
- **PR** #24429: (jacobhammons) Salt cloud doc updates, build errors and bug fixes @ *2015-06-05 00:27:38 UTC*
  - 5d738b8dab Merge pull request #24429 from jacobhammons/cloud-doc-updates
  - 1f7a13d6f9 Salt cloud doc updates, build errors and bug fixes Refs #24309
- **PR** #24408: (rallytime) Backport #24392 to 2015.5 @ *2015-06-04 20:22:09 UTC*
  - **PR** #24392: (quixoten) Fix "No such file or directory" in grains/core.py (refs: #24408)

  - cdffc02cfe Merge pull request #24408 from rallytime/bp-24392
  - ff7461b3cd Use path found by salt.utils.which
- **PR** #24380: (rallytime) Backport #24357 to 2015.5 @ *2015-06-04 20:13:51 UTC*
  - **PR** #24357: (zhujinhe) fix invoke issues of Jinja Macros example (refs: #24380)
  - a6a1f87cd9 Merge pull request #24380 from rallytime/bp-24357
  - f08c875015 fix invoke issues of Jinja Macros example
- **ISSUE** #24358: (pengyao) Netapi SSH client don't support ssh_user and ssh_passwd arguments (refs: #24388)
- **PR** #24388: (pengyao) fixes #24358 @ *2015-06-04 20:07:40 UTC*
  - 86ce9dbbdf Merge pull request #24388 from pengyao/sshclient-kwargs
  - 5c08ca48b4 fixes #24358
- **ISSUE** #22958: (highlyunavailable) Weird error when typoing a command (refs: #24367)
- **PR** #24367: (terminalmage) Improve error message when module does not exist @ *2015-06-04 20:07:12 UTC*
  - 72d2eaeda9 Merge pull request #24367 from terminalmage/issue22958
  - d0d7a5481c Improve error message when module does not exist
- **ISSUE** #23101: (gravyboat) Create a docs page for labels (refs: #23387)
- **PR** #24412: (jfindlay) backport #23387 @ *2015-06-04 20:06:03 UTC*
  - **PR** #23387: (rallytime) Add some "What are all these labels for?" documentation (refs: #24412)
  - a628778e3c Merge pull request #24412 from jfindlay/bp-23387
  - bf85772042 Make sure the parameters are in the correct order
  - 9f53809cde Add "* Change" label parameters
  - b27a15e774 Remove "workaround" wording
  - 9fff35a959 Some small fixes
  - 54a7089fd6 Link the new labels doc in contributing and hacking docs
  - 375695e696 Add pull request label definitions
  - de945638d3 Add Feature Request label definition
  - 684f291bd4 Add issue definition and augment functional areas section
  - 2da13dd525 Start a "what are all of these labels for?" doc
- **ISSUE** #24154: (ssgward) Exception when running cp.get_url (refs: #24336)
- **PR** #24336: (twangboy) Added line to give more descriptive error @ *2015-06-04 19:56:00 UTC*
  - 485116c2cc Merge pull request #24336 from twangboy/fix_cp_get_url
  - 37b11f931c Added line to give more descriptive error
- **PR** #24413: (techhat) Add more namespaced functions to GoGrid driver @ *2015-06-04 19:51:22 UTC*
  - b3d39cc0e8 Merge pull request #24413 from techhat/gogridnamespace
  - 1b397cb6fe Adding blank line
  - da08cc9aac Add more namespaced functions to GoGrid driver

- **ISSUE** #24397: (kiorky) on debian: states.apt should use virtualname as it shadows system apt module (refs: #24398, #24400, #24399)
- **PR** #24399: (kiorky) Versionvirtual (refs: #24398) @ *2015-06-04 18:02:22 UTC*
  - **PR** #24398: (kiorky) VirtualName for states.apt (refs: #24399)
  - 27f109bd76 Merge pull request #24399 from makinacorpus/versionvirtual
  - 235c78ddfe Use apt_pkg.version_compare if available
  - 1c0cd459f8 reindent block to isolate conflict on merge forward
  - 699eceab64 use var to isolate conflict on merge forward
- **PR** #24371: (joejulian) 2015.5 tls module tests @ *2015-06-04 15:20:16 UTC*
  - deaee68b89 Merge pull request #24371 from joejulian/2015.5_tls_module_tests
  - 4c5dee1e25 Add @destructiveTest decorator to destructive tests
  - 274bbd4d43 Accept results from older pyOpenSSL
  - 161f913522 All cert info should be in UTC always
  - 9affcca766 See the whole diff if dict compare fails
  - 94f620857c Ignore extensions for now. Resolve this as part of fixing issue 24338.
  - 84904d31f1 Mask lint warning for unused imported module
  - 5675b78459 Do not test if PyOpenSSL is not installed
  - 563cc66311 Add tls tests
- **PR** #24403: (jayeshka) adding states/process unit test case. @ *2015-06-04 15:19:01 UTC*
  - 84686ee695 Merge pull request #24403 from jayeshka/process_states-unit-test
  - fcb71fb35e adding states/process unit test case.
- **PR** #24402: (jayeshka) adding states/pyenv unit test case. @ *2015-06-04 15:18:11 UTC*
  - 35de8d72db Merge pull request #24402 from jayeshka/pyenv_states-unit-test
  - 5f263ab48b adding states/pyenc unit test case.
- **PR** #24401: (jayeshka) adding states/powerpath unit test case. @ *2015-06-04 15:17:46 UTC*
  - 632f838838 Merge pull request #24401 from jayeshka/powerpath-states-unit-test
  - 49ff9272ce adding states/powerpath unit test case.
- **ISSUE** #24397: (kiorky) on debian: states.apt should use virtualname as it shadows system apt module (refs: #24398, #24400, #24399)
- **PR** #24400: (kiorky) Aptversion @ *2015-06-04 15:17:19 UTC*
  - 0a6e5e0d96 Merge pull request #24400 from makinacorpus/aptversion
  - e15cb936b5 Use apt_pkg.version_compare if available
  - 953725a563 Fix too much quoting in apt.version_cmp
- **PR** #24385: (jeanpralo) Fix salt.modules.dockerio.start method @ *2015-06-04 15:00:22 UTC*
  - a904055d28 Merge pull request #24385 from jeanpralo/Fix-binds-dockerio.start

  - a0fed313fa binds dict if not specified should remain to none otherwise docker-py will try to create a new host config and all volume and ports binds are lost. config should be done at the creation of the container not when we start it
- **PR** #24381: (justinta) Disabled flaky test to review later @ *2015-06-04 14:57:43 UTC*
  - 9890bc4e43 Merge pull request #24381 from jtand/seed_test
  - 7570ae9132 Disabled flaky test to review later
- **ISSUE** #23342: (philipsd6) salt-ssh 2015.2.0rc2 fails when target doesn't have lspci available (refs: #24382)
- **PR** #24382: (basepi) [2015.5] Handle CommandExecutionError in grains commands, Fixes #23342 @ *2015-06-04 12:44:04 UTC*
  - b3fa8fefcb Merge pull request #24382 from basepi/grainscommandnotfound23342
  - 85b91d64cc Handle CommandExecutionError in grains commands
- **PR** #24379: (Starblade42) Fixes an issue where Pagerduty states/modules couldn't find their profile in the Pillar @ *2015-06-04 12:41:13 UTC*
  - 52587a4fc1 Merge pull request #24379 from Starblade42/2015.5
  - b93dc5ef6c Linting!
  - 2dd5904119 Fixes an issue where Pagerduty states/modules couldn't find it's profile in the Pillar
- **PR** #24366: (terminalmage) Use yes $'\n' instead of printf '\n' for pecl commands @ *2015-06-03 21:28:58 UTC*
  - 3ca35d1ec3 Merge pull request #24366 from terminalmage/pecl-yes
  - dcd9ad8b6e Use yes $'n' instead of printf 'n' for pecl commands
- **ISSUE** #24284: (kiorky) systemd lxc containers need use_vt=True at lxc-start stage (refs: #24348)
  - **PR** #548: (Lanzaa) Salt is now platform dependent. Use get_python_lib(1) (refs: #24348)
- **PR** #24348: (kiorky) Try to close input pipes before calling lxc-start @ *2015-06-03 19:38:07 UTC*
  - 86a3b317c6 Merge pull request #24348 from makinacorpus/lxcpre
  - 0cb11a2767 lxc: typo
  - d71efa6d66 Try to close input pipes before calling lxc-start

28.2.71 Salt 2015.5.4 Release Notes

release 2015-08-13

Version 2015.5.4 is a bugfix release for *2015.5.0*.

Statistics

- Total Merges: **247**
- Total Issue References: **138**
- Total PR References: **312**
- Contributors: **92** (0xf10e, AkhterAli, BretFisher, DmitryKuzmenko, EvaSDK, GideonRed-zz, JohannesEbke, Oro, TheBigBear, TronPaul, UtahDave, ahus1, alekti, alexandrsushko, amontalban, andre-luiz-dos-santos, aneeshusa, anlutro, asyncsrc, attiasr, babilen, basepi, bbinet, bclermont, bechtoldt, blackduckx, bobrik, cachedout, colekowalski, cro, d–j, davidjb, denmat, derBroBro, dkiser, driskell, egarbi, fleaflicker, garethgreenaway,

gmcwhistler, gtmanfred, hasues, isbm, jacksontj, jacobhammons, jahamn, jarpy, jasonkeene, jayeshka, jfindlay, jleroy, jmdcal, jodv, joejulian, jquast, justinta, kev009, klyr, l2ol33rt, loa, lomeroe, martinhoefling, mgwilliams, nicholascapo, niq000, nmadhok, nyushi, oeuftete, opdude, pcdummy, pcn, peterdemin, puneetk, rallytime, rmatulat, s0undt3ch, silenius, sjorge, stanislavb, steverweber, supertom, t0rrant, tankywoo, techhat, terminalmage, thatch45, tony-cocco, twangboy, uvsmtid, vr-jack, yanatan16, zyio)

Bug Fixes

- The `cron.present` state now correctly defaults to state ID as identifier.

Salt-Cloud Changes

- When querying for VMs in the `digital_ocean_v2` cloud driver, the number of VMs to include in a page was changed from 20 (default) to 200 to reduce the number of API calls to Digital Ocean.
- The `vmware` salt-cloud driver was back-ported from the develop branch in order for installations of Salt that are older than 2015.8.0 to be able to use the `vmware` driver without stack-tracing on various deprecation paths that were implemented in the 2015.8.0 release.

Changelog for v2015.5.3..v2015.5.4

Generated at: 2018-05-27 21:59:14 UTC

- **PR** #26292: (jquast) Rabbitmq 3.2.4 on Ubuntu has ".…done.", not "…done" @ *2015-08-13 19:53:29 UTC*
  - 0a5d1307c4 Merge pull request #26292 from jquast/backport-ubuntu-rabbitmq-fix
  - 39ef653bc2 Rabbitmq 3.2.4 on Ubuntu has …done. not …done, change the if to be more portable
- **PR** #26296: (jquast) bugfix missing 'runas=None' for rabbitmqctl cmds (backport to 2015.5) @ *2015-08-13 19:52:40 UTC*
  - 21cc3c3bf6 Merge pull request #26296 from jquast/bugfix-runas-rabbitmqctl-2015.5
  - eb77320786 bugfix missing 'runas=None' for rabbitmqctl cmds
- **ISSUE** #25618: (twangboy) Fix reg.py to work with the registry properly (refs: #26268)
- **PR** #26293: (jfindlay) Fix #26268 @ *2015-08-13 19:48:06 UTC*
  - **PR** #26268: (twangboy) Multiple improvements to reg executionmod and state mod (refs: #26293)
  - ee59d154d7 Merge pull request #26293 from jfindlay/reggie
  - 91ea964556 add versionadded to reg exec and state mods
  - 3348b726c9 fix state/reg unit tests
  - 3f74a389ce return test results when test=True
  - a1274c438d I might have fixed some tests… I might have made them worse
  - 7393adf5a8 Fixed some lint
  - 787c88a283 Multiple improvements to reg executionmod and state mod
- **ISSUE** #25192: (deuscapturus) 2015.5.2 boto_cloudwatch_alarm.present not working. (refs: #26290)
- **PR** #26290: (rallytime) Only call convert_to_arn when action name is provided @ *2015-08-13 18:48:58 UTC*
  - 5dd5ac1198 Merge pull request #26290 from rallytime/fix-25192

  - a1f90fa070 Only call convert_to_arn when action name is provided
- **PR** #26288: (bbinet) allow to delete grains which value is False @ *2015-08-13 18:24:36 UTC*
  - c81dc0b62f Merge pull request #26288 from bbinet/grains-absent-fix
  - f46722aaeb allow to delete grains which value is False
- **ISSUE** #24882: (nmadhok) salt.states.openstack_config.present and salt.states.openstack_config.absent make changes when test=True (refs: #26263)
- **PR** #26263: (rallytime) Don't make changes when test=True for openstack present/absent funcs @ *2015-08-13 16:30:31 UTC*
  - 65ab5aa495 Merge pull request #26263 from rallytime/fix-24882
  - 86b8161d22 Mock test key in __opts__ dict
  - 298685bbb2 Don't make changes when test=True for openstack present/absent funcs
- **ISSUE** #24484: (bailsman) clouds/ec2.py: create_snapshot throws exception (refs: #26265)
- **PR** #26265: (rallytime) Don't stacktrace on query return in ec2.create_snapshot @ *2015-08-13 16:28:48 UTC*
  - 3d1a9cfedd Merge pull request #26265 from rallytime/fix-24484
  - 4975300591 Don't stacktrace on query return in ec2.create_snapshot
- **PR** #26285: (stanislavb) Remove explicit version from instance identity URL @ *2015-08-13 16:25:32 UTC*
  - 5778cb3f01 Merge pull request #26285 from stanislavb/2015.5
  - 1f18f4f91e Remove explicit version from instance identity URL
- **PR** #26275: (cachedout) Re-init modules on multi-master reconnect @ *2015-08-13 15:52:50 UTC*
  - 679dc089c0 Merge pull request #26275 from cachedout/mm_reinit
  - 1e0473c04a Re-init modules on multi-master reconnect
- **PR** #26273: (garethgreenaway) Fixes to schedule module in 2015.5 @ *2015-08-13 15:34:43 UTC*
  - 75fff28779 Merge pull request #26273 from garethgreenaway/2015_5_schedule_list_show_jobs_enabled
  - 1aad4b1b4f Jobs are enabled by default but schedule.list does not show an enabled jobs as being enabled by default. This change fixes that.
- **ISSUE** #24483: (bailsman) clouds/ec2.py: del_root_vol_on_destroy and del_all_vols_on_destroy not working (refs: #26271)
- **PR** #26271: (rallytime) Fix del_root_vol_on_destroy and del_all_vols_on_destroy functionality on ec2 @ *2015-08-12 23:22:47 UTC*
  - 10af22775a Merge pull request #26271 from rallytime/fix-24483
  - 139fbb93bc Fix del_root_vol_on_destroy and del_all_vols_on_destroy functionality on ec2
- **ISSUE** #25958: (anlutro) Cron identifier does not default to state ID as documented (refs: #26219)
- **PR** #26219: (anlutro) cron: make identifier default to state ID @ *2015-08-12 18:42:33 UTC*
  - 8e1b5da2e0 Merge pull request #26219 from alprs/fix-cron_identifier_default
  - 1f02e1671b cron: fix a typo in the tests
  - a86b1b7f94 add release note about cron state changes
  - 9511e392ce cron: read full length of multi-line comments
  - 9b18cd9050 cron: more descriptive tests, updated to reflect new behavior

  - f22ad837c3 cron: change identifier default value to False
  - ad444b6e7b cron identifier: default to state id
- **ISSUE** #26207: (fullermd) group members setting fails with obscure error message on FreeBSD (refs: #26237)
- **PR** #26257: (rallytime) Back-port #26237 to 2015.5 @ *2015-08-12 18:40:35 UTC*
  - **PR** #26237: (silenius) fix issue #26207 (refs: #26257)
  - eebcade533 Merge pull request #26257 from rallytime/bp-26237
  - d57fdbc6a0 Add versionadded to new members function
  - dad1920626 fix issue #26207
- **PR** #26258: (nmadhok) Fix permission on tests/runtests.py on 2015.5 branch @ *2015-08-12 18:40:04 UTC*
  - d7c8169dfb Merge pull request #26258 from nmadhok/fix-permission
  - d94485d336 Fix permission on tests/runtests.py on 2015.5 branch
- **PR** #26261: (nmadhok) Correct spelling of integration in docs @ *2015-08-12 18:14:48 UTC*
  - 74b70c37b7 Merge pull request #26261 from nmadhok/doc-fix-2015.5
  - 714f9766e7 Correct spelling of integration in docs
- **PR** #26247: (nmadhok) Initial commit of unit tests for vmware cloud driver @ *2015-08-12 16:58:24 UTC*
  - de00c181f8 Merge pull request #26247 from nmadhok/vmware-cloud-test-2015.5
  - 6cc5f97e92 Lint Fix
  - a8bfe5ec1f Initial commit of unit tests for vmware cloud driver
- **PR** #26246: (nmadhok) Backport additions to VMware cloud driver from develop to 2015.5 branch @ *2015-08-12 15:11:26 UTC*
  - d14d7b2c0e Merge pull request #26246 from nmadhok/vmware-cloud-driver-additions-2015.5
  - 5227aa94bc Backport additions to VMware cloud driver from develop to 2015.5 branch
- **PR** #26239: (opdude) Fixed documentation to match function name @ *2015-08-12 14:48:52 UTC*
  - 87b300d7b3 Merge pull request #26239 from Unity-Technologies/2015.5
  - fc18751710 Fixed documentation to match function name
- **PR** #26232: (garethgreenaway) Fix to trust_key in gpg module for 2015.5. @ *2015-08-12 04:48:27 UTC*
  - a93b96c9ba Merge pull request #26232 from garethgreenaway/2015_5_gpg_trust_key_fix
  - e174c41887 Fix to trust_key in gpg module for 2015.5.
- **ISSUE** #25802: (jefftucker) Running module "npm.list" fails on Windows for masterless minion (refs: #26084)
- **PR** #26084: (twangboy) Added python_shell=True, quoted user input @ *2015-08-10 21:29:35 UTC*
  - b57da552ff Merge pull request #26084 from twangboy/fix_25802
  - 4503ed5b34 Fixed but with multiple packages, was causing tests to fail
  - f05e3e72a3 Merge branch '2015.5' of https://github.com/saltstack/salt into fix_25802
- **PR** #26183: (cro) Fix LDAP configuration issue. @ *2015-08-10 19:09:41 UTC*
  - c3814137a3 Merge pull request #26183 from cro/anonldap2
  - aa5e9c80b5 Lint roller

  - 79833e3f8a Cherry pick index.rst change.
  - 99f2c27399 Documentation update for anonymous bind issue.
  - 793eed7b96 Cherry pick master.py groups check
- **PR** #26186: (jacobhammons) regenerated man pages @ *2015-08-10 19:07:44 UTC*
  - 3233ed4675 Merge pull request #26186 from jacobhammons/man-page-updates
  - bf2dad913f regenerated man pages
- **PR** #26182: (basepi) [2015.5] Merge forward from 2014.7 to 2015.5 @ *2015-08-10 19:00:10 UTC*
  - d48bcf7598 Merge pull request #26182 from basepi/merge-forward-2015.5
  - 32f5345d7d Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.5
    - abdf2935c4 Merge pull request #26116 from corux/fix-escape-content
      - fd913ddc36 Append/prepend: search for full line with escaped content
    - 106356d98d Merge pull request #26088 from jacobhammons/master-finger
      - 133d5f7885 some small changes
      - d220c83f77 master_finger configuration docs switch a script to use https:// instead of http:// Refs #25751
    - 4bd4bc41f2 Merge pull request #26047 from jacobhammons/win-downloads
      - 7c162d181c Updated windows download links in the docs to https://repo.saltstack.com Refs #25961
- **ISSUE** #25998: (driskell) Event subsystem discarding required events during –batch breaking it for slow running commands (refs: #26000)
- **PR** #26000: (driskell) Implement full event caching for subscribed tags @ *2015-08-10 18:57:17 UTC*
  - f39780f8ce Merge pull request #26000 from driskell/fix_discarded_events
  - 65acf975dd Implement full event caching for subscribed tags Require all multitasking contexts to subscribe to their events so one call to get_event for one tag does not discard events that should be saved for a subsequent call to get_event with another tag. Use blocking get_event in batching with very small timeout. Fixes #25998
- **PR** #26175: (rallytime) Back-port #26153 to 2015.5 @ *2015-08-10 18:22:32 UTC*
  - **PR** #26153: (loa) Fix dockerio state documentation typo (refs: #26175)
  - c01b4cf150 Merge pull request #26175 from rallytime/bp-26153
  - 9a263067e9 Fix dockerio state documentation typo
- **ISSUE** #26024: (jpic) lxc_conf_unset in cloud.profile is ignored (refs: #26147)
- **PR** #26177: (rallytime) Back-port #26147 to 2015.5 @ *2015-08-10 18:22:01 UTC*
  - **PR** #26147: (martinhoefling) Fixes #26024 (refs: #26177)
  - ca80f33bfd Merge pull request #26177 from rallytime/bp-26147
  - 323c3ab53c Fixes #26024
- **ISSUE** #21082: (clinta) master_type failover does not failover on DNS errors (refs: #25404)
- **PR** #26179: (rallytime) Back-port #25404 to 2015.5 @ *2015-08-10 18:21:50 UTC*
  - **PR** #25404: (DmitryKuzmenko) Fixed minion failover to next master on DNS errors. (refs: #26179)

  - 1213b8d706 Merge pull request #26179 from rallytime/bp-25404
  - 52ab9fc1fb Fixed minion failover to next master on DNS errors.
- **ISSUE** #26112: (wt) state.template fails with unclear error with template with only an include (refs: #26180)
- **PR** #26180: (jfindlay) fix processing of state.template @ *2015-08-10 18:21:38 UTC*
  - b319c5ec04 Merge pull request #26180 from jfindlay/templ_env
  - 5e46ea4441 check type of matches in render_state before iterating
  - c80299b918 insert saltenv to render_state args in state.template
- **ISSUE** #26162: (nmadhok) VMware cloud driver create function failing with traceback on latest develop (refs: #26172)
- **PR** #26172: (nmadhok) [Backport] Make sure variable is a dictionary before popping something from it. @ *2015-08-10 16:42:50 UTC*
  - ef5a4a47f6 Merge pull request #26172 from nmadhok/backport-cloud-fix-26163-2015.5
  - 0f2b5f8ac8 Make sure variable is a dictionary before popping something from it.
- **ISSUE** #26098: (rdinoff) SALT.STATES.SLACK Doc update (refs: #26168)
- **PR** #26168: (cachedout) Fix slack docs @ *2015-08-10 14:57:18 UTC*
  - 2545df052a Merge pull request #26168 from cachedout/fix_slack_docs
  - f421a936dc Fix slack docs
- **ISSUE** #24106: (nvx) fileclient.py#get_url ignores HTTP Auth again (2015.5 regression) (refs: #26127)
- **PR** #26127: (garethgreenaway) Fixes to salt.utils.http related to cp.get_file_str bug. @ *2015-08-10 14:38:25 UTC*
  - 9e6b0d6165 Merge pull request #26127 from garethgreenaway/2015_5_24106
  - 66f640086a one more lint error
  - 317a8ec75c Disabling pylint for W0633, auth should only ever be a sequence at this location.
  - 08eaca4fe4 lint fixes.
  - 7046b84ac8 Fixing a bug where cp.get_file_str would not work if using http(s) URLs with authentication. The salt.utils.http library in 2015.5 defaults to using urllib instead of requests and there was no authenitication support added. This PR adds authentication support. #24106
- **ISSUE** #26141: (nmadhok) salt-cloud VMware driver fails with error in parsing configuration file (refs: #26140)
- **ISSUE** #25809: (o-sleep) vmware cloud module error message (refs: #26140)
- **ISSUE** #25625: (steverweber) cloud vmware driver does not provide mac_address unless vmware tools is running (refs: #26137, #26140)
- **PR** #26140: (nmadhok) VMware cloud driver fixes @ *2015-08-10 13:15:58 UTC*
  - 3b65e1dd91 Merge pull request #26140 from nmadhok/vmware-cloud-fixes
  - a1899b436c Correct provider name in profile example
  - 1f21876d21 Lint fixes
  - 0bd4fce9c1 Additional fixes to format_instance functions to display more information available
  - 4ee1b777e9 Change double quotes to single quotes in add_host config example
  - e132f06a5c Change double quotes to single quotes in provider configuration example

  - ad9895de07 Display error in else condition if connection is unsuccessful and does not have msg attribute. Fixes #25809
- **ISSUE** #25625: (steverweber) cloud vmware driver does not provide mac_address unless vmware tools is running (refs: #26137, #26140)
- **PR** #26137: (steverweber) use device mac address if vmtools not active @ *2015-08-09 03:05:36 UTC*
  - 474a250414 Merge pull request #26137 from steverweber/vmware_macaddress_fix
  - 2589e389f0 use device mac address if vmtools not active
- **PR** #26119: (jodv) Backport eauth bugfix to 2015.5 @ *2015-08-09 02:19:52 UTC*
  - 8a33797737 Merge pull request #26119 from jodv/backport_eauth_bugfix
  - e1a7bb5e7b fix pylint error (unnecessary 'finally' clause may swallow exceptions unintentionally)
  - 5b5b4d8fe9 Fix issue with mixed user and group eauth perms
  - 0d2c6a67a5 Return all relevant perms on login
- **PR** #26135: (cro) Fix proxy minions in 2015.5 and significantly update documentation. @ *2015-08-09 02:19:21 UTC*
  - 2b8dcce0ca Merge pull request #26135 from cro/pm20155_2
  - 28329fff55 These tests make no sense now that the proxy interface is module based and not object based.
  - b17b65d4de Fix lint.
  - f4263c8f17 Fix lint.
  - 6927251c09 Fix lint.
  - 08f1a43ff0 Fix lint.
  - 8261158b5a Fix lint.
  - b5e643b9cd Whoops…Don't log the entire proxy dictionary–might have sensitive stuff in it.
  - 2acf3c5aa3 Remove some debugging statements, change some others to 'info' level.
  - 37de6af686 More proxy minion updates
  - e79a182108 More proxy minion updates
  - 3b746ac2f6 Update to reflect refactor to LazyLoader
  - 5d390d3a5f Updates post meeting with Rick
  - d1213ce4a0 Updates post meeting with Rick
  - dd0b7c6937 Fix proxyobject confusion, now called proxymodule
  - 9b1599d436 Update to reflect refactor to LazyLoader
- **PR** #26132: (TheBigBear) minor edit @ *2015-08-08 21:05:34 UTC*
  - 2705b4a36a Merge pull request #26132 from TheBigBear/patch-5
  - 1d624d77bc minor edit
- **ISSUE** #25915: (ari) FreeBSD pkg install fails (refs: #26133)
- **PR** #26133: (amontalban) Fixed #25915 in salt/modules/pkgng.py and salt/states/pkg.py @ *2015-08-08 21:05:05 UTC*
  - 3eac28f0f9 Merge pull request #26133 from amontalban/fix-bug-25915

  - 6b0f4fca05 Fixed #25915 in salt/modules/pkgng.py and salt/states/pkg.py
- **PR** #26111: (anlutro) Better error messages when virtualenv creation fails @ *2015-08-07 21:42:09 UTC*
  - 19c42b8b3a Merge pull request #26111 from alprs/fix-virtualenv_fail_message
  - b2913acc48 virtualenv: better error messages when creation fails
- **ISSUE** #26093: (freedba) archive.tar bug (refs: #26110)
- **PR** #26110: (jfindlay) check for sources before adding them to cmd str @ *2015-08-07 21:33:23 UTC*
  - 6d2835b464 Merge pull request #26110 from jfindlay/tar_sources
  - 1b2f8905eb check for sources before adding them to cmd str
- **PR** #26106: (vr-jack) Update __init__.py @ *2015-08-07 21:15:55 UTC*
  - 2d271b3612 Merge pull request #26106 from vr-jack/2015.5
  - 5664de6610 Update __init__.py
- **ISSUE** #25983: (jmdcal) Trying to get md5 of local zip (refs: #25984)
- **PR** #26101: (rallytime) Back-port #25984 to 2015.5 @ *2015-08-07 18:56:26 UTC*
  - **PR** #25984: (jmdcal) Support local files without md5sum (refs: #26101)
  - 40d41741c1 Merge pull request #26101 from rallytime/bp-25984
  - 3d279c0713 Pylint Fix
  - cced16a9f4 Support local files without md5sum
- **PR** #26080: (techhat) Fix string checking in s3fs @ *2015-08-06 23:36:09 UTC*
  - 0d3c2d549e Merge pull request #26080 from techhat/fixlower
  - 8717a36963 Fix string checking in s3fs
- **ISSUE** #26039: (basepi) Update scheduler docs to use orchestrate instead of overstate (refs: #26079)
- **PR** #26079: (cachedout) Update docs to remove state.over @ *2015-08-06 23:35:26 UTC*
  - dc9c9b5a34 Merge pull request #26079 from cachedout/issue_26039
  - f03f460af2 Update docs to remove state.over
    - 89d8faaeb1 Added python_shell=True, quoted user input
- **PR** #26058: (opdude) Fix choco version on chocolatey versions below 0.9.9 @ *2015-08-06 18:50:10 UTC*
  - aa023f25b8 Merge pull request #26058 from Unity-Technologies/hotfix/fix-choco-pkg-version-2015-5
  - beddb96b2b Fix choco version on chocolatey versions below 0.9.9
- **PR** #26068: (jfindlay) fix autoruns.list looking in wrong directory @ *2015-08-06 18:49:48 UTC*
  - fbe2584abe Merge pull request #26068 from jfindlay/auto_fix
  - 1e9a850e23 fix autoruns.list looking in wrong directory
- **ISSUE** saltstack/salt-bootstrap#640: (Deshke) salt-minon install bug on ubuntu 14.04 tornado>=4.0 (refs: #26065)
- **ISSUE** saltstack/salt-bootstrap#633: (neilmb) Bootstrap install fails on python-requests dependency (refs: #26065)
- **ISSUE** saltstack/salt-bootstrap#632: (JulianGindi) python-requests : Depends: python-urllib3 (>= 1.7.1) but it is not installable (refs: #26065)

- **ISSUE** saltstack/salt-bootstrap#631: (DavidJFelix) Stable broken in 15.04 even with -P (refs: #26065)
- **ISSUE** #636: (pille) restict access to salt:// filesystem (refs: #‘saltstack/salt-bootstrap#638‘_)
- **ISSUE** #613: (thatch45) Add timeout option to publish.publish (refs: #‘saltstack/salt-bootstrap#634‘_)
  - **PR** saltstack/salt-bootstrap#638: (stanislavb) Use prefix /usr for centos git install (refs: #26065)
  - **PR** saltstack/salt-bootstrap#634: (BretFisher) bugfix: exit git root before removing it (refs: #26065)
- **PR** #26065: (s0undt3ch) [2015.5] Update to latest bootstrap stable release v2015.06.08 @ *2015-08-06 17:09:35 UTC*
  - 5570408597 Merge pull request #26065 from s0undt3ch/hotfix/bootstrap-script-2015.5
  - a430a62b01 Update to latest bootstrap stable release v2015.06.08
- **ISSUE** #25994: (gmcwhistler) module.ilo tempfile creation in __execute_cmd results in TypeError: cannot concatenate ’str’ and ’int’ objects (refs: #26061)
- **PR** #26061: (gmcwhistler) Patch for issue #25994 @ *2015-08-06 17:07:34 UTC*
  - 83a1922196 Merge pull request #26061 from gmcwhistler/2015.5
  - b9e89d0f2d Patch for issue #25994
- **ISSUE** #26063: (saltstack-bot) not working with salt-cloud shows unknown locale error (refs: #26064)
- **PR** #26064: (s0undt3ch) Don’t stacktrace when trying to get the default locale. @ *2015-08-06 16:11:05 UTC*
  - 073fb2bdea Merge pull request #26064 from s0undt3ch/issues/26063-unknown-locale
  - 8c6ab78b1d Don’t stacktrace when trying to get the default locale.
- **PR** #26048: (jacobhammons) Updated windows download links in the docs to https://repo.saltstack.com @ *2015-08-05 22:59:50 UTC*
  - 0f44761d6e Merge pull request #26048 from jacobhammons/win-downloads2
  - 75243b61cf Updated windows download links in the docs to https://repo.saltstack.com
- **ISSUE** #25616: (rallytime) [2015.5] Provisioning Linodes Stacktraces (refs: #26044)
- **PR** #26044: (rallytime) Make sure the key we’re comparing is also lowercase @ *2015-08-05 19:23:54 UTC*
  - dedcadc37e Merge pull request #26044 from rallytime/fix-25616
  - c2e3803810 Make sure the key we’re comparing is also lowercase
- **PR** #26042: (jfindlay) fix test mode logic in state docs @ *2015-08-05 19:23:07 UTC*
  - f005bdfce6 Merge pull request #26042 from jfindlay/result
  - a83059ca01 fix test mode logic in state docs
- **ISSUE** #24460: (nicholascapo) Survey runner does not follow *–out* flag (refs: #26036)
- **PR** #26036: (nicholascapo) survey.hash: Remove manually printed text @ *2015-08-05 19:21:59 UTC*
  - 51ab6864b7 Merge pull request #26036 from nicholascapo/survey.hash_follow_out_flag
  - 439ee9831c survey.hash: Remove manually printed text
- **PR** #26030: (opdude) Fix a bug in choco version that returned odd data @ *2015-08-05 16:30:25 UTC*
  - 6a4d18eba6 Merge pull request #26030 from Unity-Technologies/hotfix/fix-choco-pkg-version-2015-5
  - 3dd96c0638 Fix a bug in choco version that returned odd data
- **PR** #26032: (jfindlay) add test logic to state reult doc @ *2015-08-05 16:28:32 UTC*

  - c96d3bb55e Merge pull request #26032 from jfindlay/result
  - 0fd180e106 add test logic to state reult doc
- **ISSUE** #23764: (es1o) source_hash from local file is not supported. (refs: #26031, #25750)
- **PR** #26031: (alekti) Revert "Add file as supported protocol for file source_hash. Fixes #23764" @ *2015-08-05 15:32:01 UTC*
  - bd14d85636 Merge pull request #26031 from alekti/merge-pull-25750-to-2015.5
  - 5a7cab4dcc Revert "Add file as supported protocol for file source_hash. Fixes #23764."
- **PR** #26021: (anlutro) Documentation: Specify versionadded for git.present shared argument @ *2015-08-05 14:17:38 UTC*
  - d55e6e5fe9 Merge pull request #26021 from alprs/docs-git_present_shared_versionadded
  - 8fa678aaa7 specify versionadded for git.present shared argument
- **ISSUE** #25701: (alekti) Issue #23764 regression (refs: #25750)
- **ISSUE** #23764: (es1o) source_hash from local file is not supported. (refs: #26031, #25750)
- **PR** #26020: (alekti) Correctly resolve conflict merging pull 25750 to 2015.5 @ *2015-08-05 14:16:58 UTC*
  - **PR** #25750: (alekti) Add file as supported protocol for file source_hash. Fixes #25701. (refs: #26020)
  - 5e17c5d230 Merge pull request #26020 from alekti/merge-pull-25750-to-2015.5
  - 4b9d7426cc Add file as supported protocol for file source_hash. Fixes #23764.
- **ISSUE** #22241: (masterkorp) Salt master not properly generating the map (refs: #25358)
- **PR** #26016: (basepi) Revert "Deep merge of pillar lists" @ *2015-08-05 04:59:52 UTC*
  - **PR** #25358: (dkiser) Deep merge of pillar lists (refs: #26016)
  - 53f7aadcd7 Merge pull request #26016 from basepi/revert.25358
  - 8a0e8e0460 Revert "Deep merge of pillar lists"
- **ISSUE** #12255: (eliasp) 'system.set_computer_desc' fails with non-ASCII chars (refs: #25992)
- **PR** #25992: (twangboy) Refactor win_system.py @ *2015-08-05 04:54:18 UTC*
  - 200bff7538 Merge pull request #25992 from twangboy/fix_12255
  - 0502897635 Fixed the lint… again
  - 6f85d6b9af Fixed some lint
  - 4195803e56 Merge pull request #3 from jfindlay/win_sys
    - 9156bbd33e update win_system exec mod unit tests
  - c92add95b5 Gated ctypes import, fixed some lint
  - d7670fda0a Refactor win_service.py
- **ISSUE** #25948: (twangboy) Fix uncomment function to handle spaces (refs: #26002)
- **PR** #26002: (twangboy) Fixed regex to account for comment character followed by whitespace @ *2015-08-04 22:28:11 UTC*
  - c168159750 Merge pull request #26002 from twangboy/fix_25948
  - ba1a57e582 Fixed regex to account for comment character followed by whitespace
- **ISSUE** #25949: (godlike64) layman.add does not work with unofficial overlays (refs: #25970)

- **PR** #25970: (jfindlay) accept addition of layman overlay @ *2015-08-04 15:42:28 UTC*
  - 4ad2422da1 Merge pull request #25970 from jfindlay/layman
  - 237a9e18b3 accept addition of layman overlay
- **PR** #25971: (basepi) [2015.5] salt.modules.reg Add spaces for strings split across multiple lines @ *2015-08-04 15:39:48 UTC*
  - f136c6c1c0 Merge pull request #25971 from basepi/reg.typos
  - bb001a6c0e Add spaces for strings split across multiple lines
- **PR** #25990: (rallytime) Back-port #25976 to 2015.5 @ *2015-08-04 14:36:53 UTC*
  - **PR** #25976: (fleaflicker) Typo in help output (refs: #25990)
  - 6383dd8a7d Merge pull request #25990 from rallytime/bp-25976
  - 5f6dc0cc85 Typo in help output
- **PR** #25996: (attiasr) fix msiexec package remove @ *2015-08-04 14:36:31 UTC*
  - 9f8bf75dc0 Merge pull request #25996 from attiasr/patch-1
  - 5fbc5fcd94 fix msiexec package remove
- **ISSUE** #25863: (peterdemin) pkg.installed fails on already installed package if it is in versionlock.list (refs: #25864)
- **PR** #25966: (rallytime) Back-port #25864 to 2015.5 @ *2015-08-03 18:48:26 UTC*
  - **PR** #25864: (peterdemin) #25863 state.pkg.installed fix (refs: #25966)
  - 2dca8d959b Merge pull request #25966 from rallytime/bp-25864
  - 0f7f9637b4 #25863 fix - state.pkg: do preflight check only for non-installed packages
- **PR** #25967: (rallytime) Back-port #25917 to 2015.5 @ *2015-08-03 18:48:02 UTC*
  - **PR** #25917: (jmdcal) adding missing format string (refs: #25967)
  - a6d8e541ed Merge pull request #25967 from rallytime/bp-25917
  - 82b7e14a1f adding missing format string
- **PR** #25895: (basepi) [2015.5] Merge forward from 2014.7 to 2015.5 @ *2015-08-03 17:12:37 UTC*
  - 87d028b302 Merge pull request #25895 from basepi/merge-forward-2015.5
  - 56e43c8f88 Fix lint
  - 93a182d9ea Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.5
    - d93eb87c16 Merge pull request #25750 from alekti/2014.7
      - 9ec3ae96d4 Add file as supported protocol for file source_hash. Fixes #23764.
    - 3a15df22ac Merge pull request #25704 from cachedout/master_type_2014_7
      - c95886c9a7 Ensure prior alignment with master_type in 2014.7
    - d1b9362a73 Merge pull request #25657 from MrCitron/pattern-carbon-returner-2014.7
      - f8b2f8079f Add the ability to specify a base pattern for metrics path used by the carbon returner
    - 9634351fc2 Merge pull request #25633 from AkhterAli/2014.7
      - 29be4bbe11 Update loader.py

- **ISSUE** #25850: (ssgward) Need to add packages to –versions-report (refs: #25941)
- **PR** #25941: (jfindlay) add timelib to dependency versions @ *2015-08-03 12:23:42 UTC*
    - 98955057e0 Merge pull request #25941 from jfindlay/time_lib
    - 464f7a404c add timelib to dependency versions
- **PR** #25951: (garethgreenaway) Log when event.fire and event.fire_master fail. @ *2015-08-03 00:19:45 UTC*
    - dcc6883b24 Merge pull request #25951 from garethgreenaway/event_fire_failed_log_why
    - 7f20454427 If we're unable to fire an event, log the cause so we know what happened
- **ISSUE** #25838: (grep4linux) docs disable_modules documentation typo (refs: #25942)
- **PR** #25942: (jfindlay) typo in minion doc @ *2015-07-31 23:34:55 UTC*
    - 4143cec3bf Merge pull request #25942 from saltstack/lover
    - 7e121de907 Update minion.rst
- **PR** #25938: (jacobhammons) Doc on using syndic with multimaster @ *2015-07-31 23:05:05 UTC*
    - **PR** #14690: (jacksontj) Multi syndic (refs: #25938)
    - 1f20c065b8 Merge pull request #25938 from jacobhammons/syndic-multimaster
    - ac0a8ff711 Doc on using syndic with multimaster
- **ISSUE** #25839: (twangboy) ALLUSERS="1" should be a default when installing MSI's (refs: #25848)
- **PR** #25848: (twangboy) Added allusers="1" when installing msi @ *2015-07-31 20:33:17 UTC*
    - 18a9e65e1f Merge pull request #25848 from twangboy/fix_25839
    - e797739a1b Removed normalize_name function
    - ad7fdda68b Adder allusers="1" when installing msi
- **PR** #25898: (jfindlay) clarify and expand syndic docs @ *2015-07-31 20:01:23 UTC*
    - de0a0593c2 Merge pull request #25898 from jfindlay/syndic_doc
    - 4795952847 rework syndic doc
    - a25d0eabef update syndic doc to conform to style
- **ISSUE** #25852: (UtahDave) Salt loader is not loading Salt vars in reactor python renderer (refs: #25927)
- **PR** #25927: (jacksontj) Pass actual renderers to the Reactor's Compiler @ *2015-07-31 20:00:17 UTC*
    - d1f3da548a Merge pull request #25927 from jacksontj/2015.5
    - cf7479aa0a Pass actual renderers to the Reactor's Compiler
- **ISSUE** #25810: (nvx) winpkg highstate fails when a new package name contains a unicide character (refs: #25921)
- **PR** #25921: (cachedout) Handle non-ascii in state log @ *2015-07-31 17:41:30 UTC*
    - 331fc121a8 Merge pull request #25921 from cachedout/issue_25810
    - 8074c545ea Handle non-ascii in state log
- **PR** #25919: (TheBigBear) Minor update to msi un-installer info @ *2015-07-31 17:39:48 UTC*
    - 20fb8da8d4 Merge pull request #25919 from TheBigBear/patch-4
    - c994d22696 Minor update to msi un-installer info

  - **PR** #25982: (sjorge) salt.modules.smartos_* limit to global zone only (refs: #25905)
- **PR** #25905: (rallytime) Back-port #25982 to 2015.5 @ *2015-07-30 23:24:19 UTC*
  - **PR** #25892: (TheBigBear) Update 7-zip msi un-installer instructions (refs: #25905)
  - 9a569da4ee Merge pull request #25905 from rallytime/bp-25892
  - 333fbdde30 Update 7-zip msi un-installer instructions
- **ISSUE** #25577: (yellow1912) Wrong indentation in document (refs: #25696)
- **PR** #25890: (rallytime) Back-port #25698 to 2015.5 @ *2015-07-30 23:12:09 UTC*
  - **PR** #25698: (rallytime) Back-port #25659 to 2015.8 (refs: #25890)
  - **PR** #25696: (AkhterAli) Update schedule.py
  - **PR** #25659: (isbm) Bugfix: crash at getting non-existing repo (refs: #25698)
  - 6a738c5c41 Merge pull request #25890 from rallytime/bp-25696
  - 7d68e49d98 Update schedule.py
- **ISSUE** #25650: (jacksontj) state.running documentation is incorrect (refs: #25894)
- **ISSUE** #24042: (whiteinge) The state_events setting is not documented (refs: #25894)
- **ISSUE** #23788: (k5jj) functions in drac.py module do not match documentation (refs: #25894)
- **ISSUE** #21296: (Lothiraldan) Possible minion enumeration using saltutil.find_job and eauth (refs: #25894)
- **PR** #25894: (jacobhammons) Minor doc bug fixes @ *2015-07-30 23:02:34 UTC*
  - 8abb21e206 Merge pull request #25894 from jacobhammons/bug-fixes
  - 3f3db4bd8e Additions for #24042
  - db2129b199 Minor doc bug fixes Refs #24042 Refs #25650 Refs #21296 Refs #23788
- **ISSUE** #24036: (arthurlogilab) [salt-cloud] Protect against passing command line arguments as names for the –destroy command in map files (refs: #25877)
- **PR** #25877: (rallytime) Protect against passing a map file in addition to VM names with –destroy @ *2015-07-30 21:55:45 UTC*
  - 59e1680182 Merge pull request #25877 from rallytime/fix-24036
  - 0211972fd7 Whitespace fix
  - c6715e0404 Protect against passing a map file in addition to VM names with –destroy
  - 3aa5045138 Clean up stacktrace when referenced map file doesn't exist
- **PR** #25870: (rallytime) Back-port #25824 to 2015.5 @ *2015-07-30 21:54:35 UTC*
  - **PR** #25824: (klyr) Fix get_managed() in file.py module for local files (refs: #25870)
  - c4c9e40be6 Merge pull request #25870 from rallytime/bp-25824
  - 1fd4837beb Fix get_managed() in file.py module for local files
- **PR** #25885: (t0rrant) Update Debian changelog @ *2015-07-30 20:05:59 UTC*
  - af2326af68 Merge pull request #25885 from t0rrant/patch-3
  - 3f73900c61 Update Debian changelog
- **ISSUE** #25478: (zyio) salt-ssh - Unable to locate current thin version (refs: #25862)
- **ISSUE** #25026: (sylvia-wang) salt-ssh "Failure deploying thin" when using salt module functions (refs: #25862)

- **PR** #25875: (rallytime) Back-port #25862 to 2015.5 @ *2015-07-30 17:34:02 UTC*
  - **PR** #25862: (zyio) Adding SCP_NOT_FOUND exit code (refs: #25875)
  - 6ce0b3e5b8 Merge pull request #25875 from rallytime/bp-25862
  - d7f448d501 Needed popen.wait().
  - 25f8042e41 Checking for scp existance. Using command -v should be POSIX
  - 6b2100a30b New exitcode for SCP not found Re: https://github.com/saltstack/salt/issues/25478 and https://github.com/saltstack/salt/issues/25026
- **PR** #25873: (rallytime) Back-port #25855 to 2015.5 @ *2015-07-30 17:33:55 UTC*
  - **PR** #25855: (puneetk) Patch 3 (refs: #25873)
  - 66dcc5525e Merge pull request #25873 from rallytime/bp-25855
  - f1f7ce25b7 Update saltmod.py
  - 23a6806008 Update saltmod.py
- **PR** #25871: (rallytime) Back-port #25829 to 2015.5 @ *2015-07-30 17:33:43 UTC*
  - **PR** #25829: (peterdemin) Fixed typo in salt.states.saltmod.function doc string (refs: #25871)
  - bf8bd38da7 Merge pull request #25871 from rallytime/bp-25829
  - a80c47ee10 Fixed typo in salt.states.saltmod.function doc string
- **ISSUE** #24002: (csakoda) File lock contention on windows minions causing highstate crash (refs: #25788)
- **PR** #25869: (rallytime) Back-port #25788 to 2015.5 @ *2015-07-30 17:33:33 UTC*
  - **PR** #25788: (opdude) Catch a hard crash when running highstate on windows (refs: #25869)
  - f26310ff0b Merge pull request #25869 from rallytime/bp-25788
  - 65b18e3b34 Catch a hard crash when running highstate on windows
- **ISSUE** #19532: (stolendog) salt-ssh running git clone with not root user (refs: #25853)
- **PR** #25853: (davidjb) Make ssh-id-wrapper accessible to non-root users @ *2015-07-30 16:49:47 UTC*
  - 810fbb8bfb Merge pull request #25853 from davidjb/ssh-id-wrapper-non-root
  - 6492bde192 Make ssh-id-wrapper accessible to non-root users
- **ISSUE** #25447: (spo0nman) SaltMaster is crippled with Minion Re-Authentication (refs: #25856)
- **PR** #25856: (jfindlay) expand minion reauth scalability documentation @ *2015-07-30 15:33:17 UTC*
  - b6805b068a Merge pull request #25856 from jfindlay/intro_scale
  - 5921461bb1 style and usage consistency in intro_scale
  - 51dc7cacfb whitespace adjustments in intro_scale
  - 39a82467f1 expand minion reauth scalability documentation
- **ISSUE** #25801: (themalkolm) Update docs that salt.states.winrepo requires *roles:salt-master* in grains. (refs: #25840)
- **PR** #25840: (jfindlay) add note to winrepo state docs about required grain @ *2015-07-30 14:38:27 UTC*
  - 423d528b73 Merge pull request #25840 from jfindlay/winrepo_master
  - b6cfd54f3b add note to winrepo state docs about required grain

- **ISSUE** #25827: (0xf10e) "Deprecating Code" doesn't mention Usage of warn_until() w/ Release Names (refs: #25846)
- **PR** #25846: (jfindlay) rework deprecation documentation for release names @ *2015-07-30 13:26:21 UTC*
  - 754c8be719 Merge pull request #25846 from jfindlay/depr_code
  - d377f42c48 rework deprecation documentation for release names
- **ISSUE** #23288: (UtahDave) cp.push fails to recreate empty files. (refs: #25833)
- **PR** #25833: (jahamn) Allows cp.push to recreate empty files @ *2015-07-29 16:14:48 UTC*
  - d9ab4bb989 Merge pull request #25833 from jahamn/fix-cp.push-not-recreating-empty-files
  - eac19fbf33 Allows cp.push to recreate empty files
- **ISSUE** #11474: (JensRantil) pkgrepo.managed key_url: salt:// always use *base* env (refs: #25831)
- **PR** #25831: (rallytime) Add salt:// to key_url options to docs for pkgrepo.managed @ *2015-07-29 15:38:43 UTC*
  - 6f93d64784 Merge pull request #25831 from rallytime/fix-11474
  - 067ea788e9 Add salt:// to key_url options to docs for pkgrepo.managed
- **ISSUE** #22699: (arthurlogilab) salt-cloud fails on KeyError when given a nonexistant action (refs: #25807)
- **PR** #25807: (rallytime) Provide helpful error when using actions with a mapfile @ *2015-07-29 15:30:15 UTC*
  - 72b3633383 Merge pull request #25807 from rallytime/fix-22699
  - 3f3005c746 Use handle_exception function in cloud cli.py
  - f91edf3a33 Provide helpful error when using actions with a mapfile
- **PR** #25818: (jfindlay) fix autoruns list @ *2015-07-29 15:29:20 UTC*
  - 71497adc0d Merge pull request #25818 from jfindlay/autoruns_users
  - c2dbb65982 fix autoruns list for modern windowsen
- **PR** #25826: (anlutro) Check that "onchanges" is a list @ *2015-07-29 15:00:28 UTC*
  - 98b324c5f8 Merge pull request #25826 from alprs/fix-onchanges_type_check
  - 7992a3f0f4 state.py: check that "onchanges" is a list
- **ISSUE** #25258: (nickw8) windows minion repo not updating (refs: #25798)
- **PR** #25798: (twangboy) Fixed stacktrace on package name not found @ *2015-07-28 22:40:14 UTC*
  - ad07dc1e27 Merge pull request #25798 from twangboy/fix_25258
  - aa19c2bf8f Fixed stacktrace on package name not found
- **ISSUE** #25437: (lorengordon) Stacktrace on Windows when running pkg.list_pkgs (refs: #25598, #25763)
- **PR** #25797: (twangboy) Changed repocache back to cached_repo @ *2015-07-28 22:39:32 UTC*
  - **PR** #25763: (twangboy) Fix 25437 (refs: #25797)
  - 4a38d4a606 Merge pull request #25797 from twangboy/fix_revert_in_25763
  - 81d5b5ee55 Changed repocache back to cached_repo
- **PR** #25793: (rallytime) Back-port #25730 to 2015.5 @ *2015-07-28 19:37:34 UTC*
  - **PR** #25730: (sjorge) patchelf lives in pkgsrc (refs: #25793)
  - 823f0ce350 Merge pull request #25793 from rallytime/bp-25730

  - 937779eb51 patchelf lives in pkgsrc
- **PR** #25792: (rallytime) Back-port #25688 to 2015.5 @ *2015-07-28 19:37:17 UTC*
  - **PR** #25688: (bclermont) Don't acquire lock if there is no formatter (refs: #25792)
  - 4109ae55f9 Merge pull request #25792 from rallytime/bp-25688
  - 0aa1416b6b Don't acquire lock if there is no formatter
- **PR** #25796: (cachedout) Remove debug from docs @ *2015-07-28 17:35:59 UTC*
  - 737fb1410c Merge pull request #25796 from cachedout/debug_doc
  - 33bfdf3b0b Remove debug from docs
- **ISSUE** #24920: (voileux) module.zpool.create on character device is not possible by salt (refs: #25749)
- **PR** #25749: (jahamn) Allow zpool.create on character devices @ *2015-07-28 16:01:40 UTC*
  - a658753eff Merge pull request #25749 from jahamn/fix-zpool-special-char-device-support
  - 361f6cc23f Allow zpool.create on character devices
- **PR** #25685: (twangboy) Fixed regex issues with comment and uncomment @ *2015-07-28 15:29:49 UTC*
  - 1fae76d53c Merge pull request #25685 from twangboy/fix_25594
  - a904e8329b Fixed another test failure…
  - aa077d3a86 Fixed more tests… justin findlay helped me…
  - 87c8f8dfb5 Fixed some tests… maybe…
  - 3c1a73f16c Fixed some lint
  - b3e44e342c Fixed states to work with comment_line
  - b1cedd1153 Fixed regex issues with comment and uncomment
- **ISSUE** #25437: (lorengordon) Stacktrace on Windows when running pkg.list_pkgs (refs: #25598, #25763)
- **PR** #25763: (twangboy) Fix 25437 (refs: #25797) @ *2015-07-28 15:29:27 UTC*
  - 0bdb29402a Merge pull request #25763 from twangboy/fix_25437
  - 9e70c800b9 The real fix for 25437 that doesn't break other crap
  - d7347e01e5 Revert "Fixed problem trying to load file with name of boolean type"
  - cf57712eeb Merge branch '2015.5' of https://github.com/saltstack/salt into fix_25437
- **PR** #25752: (thatch45) State top saltenv @ *2015-07-28 01:02:10 UTC*
  - c1236595f9 Merge pull request #25752 from thatch45/state_top_saltenv
  - 65d6ec0659 don't override the minion config unless requested
  - 26c858361c Add state_top_saltenv to the config chain
  - 36a3b674a7 Add raet support for state_top_saltnev
  - f6fa025b13 Add saltenv top file support to salt master_opts
  - 4a1c53309b Add state_top_saltenv support
- **ISSUE** #25717: (twangboy) Problem with chocolatey module not loading (refs: #25755)
- **PR** #25755: (twangboy) Fixed problem with dunder functions not being passed @ *2015-07-27 19:31:22 UTC*
  - f367acb253 Merge pull request #25755 from twangboy/fix_25717

  - 10e410504d Fixed problem with dunder functions not being passed
- **ISSUE** #25352: (m03) reg.absent reporting incorrect results (refs: #25648)
- **PR** #25648: (twangboy) Clarified functionality of reg module, fixed state to work with new module @ *2015-07-27 19:30:33 UTC*
  - f05ae95f9c Merge pull request #25648 from twangboy/fix_25352
  - d6496ce814 Merge pull request #1 from jfindlay/reg
    - 3b0cc6592a fix reg unit tests
  - b473fb7827 Fixed some tests… maybe…
  - ff7296d983 Fixed some more lint
  - 7a71f5ea6a Merge branch '2015.5' of https://github.com/saltstack/salt into fix_25352
  - f57b2b8e7a Fixed some line, added documentation
  - d78fa97a71 Merge branch '2015.5' of https://github.com/saltstack/salt into fix_25352
  - 99d9518af8 Clarified functionality of reg module, fixed state to work with new module
- **ISSUE** #25154: (uvsmtid) All data mixed on STDOUT together should generate valid JSON output (refs: #25722)
- **ISSUE** #25153: (uvsmtid) Multiple results should generate valid JSON output (refs: #25722)
- **PR** #25740: (rallytime) Back-port #25722 to 2015.5 @ *2015-07-27 16:08:40 UTC*
  - **PR** #25722: (uvsmtid) Minor docs changes to emphasize JSON output problems without *–static* option (refs: #25740)
  - 29c66d85a4 Merge pull request #25740 from rallytime/bp-25722
  - c33eb813ea Change docs for –static option with JSON - text B
  - 89dd2ec8fb Change docs for –static option with JSON - text A
- **PR** #25739: (rallytime) Back-port #25709 to 2015.5 @ *2015-07-27 16:08:27 UTC*
  - **PR** #25709: (colekowalski) add direct-io-mode to mount_invisible_options (refs: #25739)
  - **PR** #25699: (rallytime) Back-port #25660 to 2015.5 (refs: #25709)
  - **PR** #25660: (colekowalski) add glusterfs' direct-io-mode to mount_invisible_keys (refs: #25699, #25709)
  - 135b03e53b Merge pull request #25739 from rallytime/bp-25709
  - fda2ffa44e add direct-io-mode to mount_invisible_options
- **PR** #25738: (rallytime) Back-port #25671 to 2015.5 @ *2015-07-27 16:08:23 UTC*
  - **PR** #25671: (niq000) added a parameter so verifying SSL is now optional instead of hard-coded (refs: #25738)
  - 095a923b6e Merge pull request #25738 from rallytime/bp-25671
  - 525cd70589 added a parameter so verifying SSL is now optional instead of hard-coded
- **ISSUE** #25229: (rmatulat) Module git.latest kills target directory when test=True (refs: #25608)
- **PR** #25737: (rallytime) Back-port #25608 to 2015.5 @ *2015-07-27 16:08:18 UTC*
  - **PR** #25608: (rmatulat) Fix: prevent git.latest from removing target (refs: #25737)
  - 05fbfe64e9 Merge pull request #25737 from rallytime/bp-25608

  - df85d734bc Fix: prevent git.latest from removing target Fixes #25229 While force=True and test=True git.latest should not remove the target directory.
- **PR** #25733: (davidjb) Avoid IndexError when listing mounts if mount output ends in newline @ *2015-07-27 16:08:05 UTC*
  - 9817fc5556 Merge pull request #25733 from davidjb/mount-fix
  - 6d0bce2418 Test length of comps when listing mounts
- **ISSUE** #22460: (onmeac) Command setm is not supported (yet) (refs: #25705)
- **PR** #25705: (blackduckx) Support for setm augeas command. @ *2015-07-27 16:07:10 UTC*
  - 82ba390b7b Merge pull request #25705 from blackduckx/augeas-setm
  - cad0f2b46e Augeas: fix pylint and documentation
  - ee97896cba Support for setm augeas command.
- **PR** #25703: (cachedout) Return to *str* for master_type for 2015.5 @ *2015-07-27 16:06:22 UTC*
  - f732be365d Merge pull request #25703 from cachedout/master_type_2015_5
  - 0dc28ad3e4 Return to *str* for master_type for 2015.5
- **ISSUE** #25144: (johnccfm) user.present on Windows fails to add user to groups if group name contains a space (refs: #25702)
- **PR** #25702: (twangboy) Fixed win_user module for groups with spaces in the name @ *2015-07-27 15:06:33 UTC*
  - dea3d31578 Merge pull request #25702 from twangboy/fix_25144
  - d5be7a2fdf Fixed win_user moduele for groups with spaces in the name
- **ISSUE** #25351: (m03) win_servermanager.list_installed failing with "IndexError: list index out of range" (refs: #25711)
- **PR** #25711: (twangboy) Fixed problem with win_servermanager.list_installed @ *2015-07-27 15:05:48 UTC*
  - 186af9b54d Merge pull request #25711 from twangboy/fix_25351
  - 82fa911931 Fixed problem with win_servermanager.list_installed
- **ISSUE** #25435: (yee379) progressbar dependency missing (refs: #25714)
- **PR** #25714: (cachedout) Display warning when progressbar can't be loaded @ *2015-07-25 00:10:13 UTC*
  - ad8456eeed Merge pull request #25714 from cachedout/issue_25435
  - 44f34684ef Included note in help docs
  - 4e2fee17cc Display warning when progressbar can't be loaded
- **PR** #25699: (rallytime) Back-port #25660 to 2015.5 (refs: #25709) @ *2015-07-24 22:11:40 UTC*
  - **PR** #25660: (colekowalski) add glusterfs' direct-io-mode to mount_invisible_keys (refs: #25699, #25709)
  - a0969ff74a Merge pull request #25699 from rallytime/bp-25660
  - 85c636d7a1 add glusterfs' direct-io-mode to mount_invisible_keys
- **ISSUE** #25689: (anlutro) Minion log in salt-ssh (refs: #25694)
- **PR** #25694: (s0undt3ch) Salt-SSH fix for #25689 @ *2015-07-24 21:41:57 UTC*
  - fe829564f4 Merge pull request #25694 from s0undt3ch/2015.5
  - afba3bde90 Use a relative un-nested path to the salt-call logfile.

  - 6309f22a65 Fix wrong variable assignment
  - c312592c81 Have cookie JAR's respect the configured *cachedir*
- **ISSUE** #25250: (wipfs) 'force' option in copy state deletes target file (refs: #25461, #25710)
- **PR** #25710: (jahamn) Integration Testcase for Issue 25250 @ *2015-07-24 20:57:33 UTC*
  - fb4744b2f8 Merge pull request #25710 from jahamn/integration-test-for-issue-25250
  - 24f653e963 Integration Test for Issue 25250
- **PR** #25680: (basepi) [2015.5] Move cmd.run jinja aliasing to a wrapper class to prevent side effects @ *2015-07-24 19:52:10 UTC*
  - **PR** #25049: (terminalmage) Fix cmd.run when cross-called in a state/execution module (refs: #25680)
  - 18c9d5454d Merge pull request #25680 from basepi/jinja.alias.25049
  - e83a0f9b2b Use new-style classes
  - 4a50bac1c2 Fix typo
  - 36410389dc Name the Nitrogen release
  - 77679596f9 Make ALIASES global
  - 01c209efd9 Fix some aliases references
  - 1644641c57 Move cmd.run aliasing to a wrapper class to prevent side effects
- **PR** #25682: (basepi) [2015.5] Fix parsing args with just a hash (#) @ *2015-07-24 19:52:01 UTC*
  - 6a5c6dcd04 Merge pull request #25682 from basepi/fix.hash.parsing
  - 8d75c1b882 Fix parsing args with just a hash (#)
- **PR** #25695: (stanislavb) Configurable AWS region & region from IAM metadata @ *2015-07-24 19:36:40 UTC*
  - d330ef0d81 Merge pull request #25695 from stanislavb/expose-aws-region-config-and-fetch-region-from-metadata
  - 595da6252e Configurable AWS region & region from IAM metadata
- **PR** #25645: (kev009) Fix pkgng provider to work with a sources list and the underlying pkg… @ *2015-07-24 16:33:18 UTC*
  - ea0d295d49 Merge pull request #25645 from kev009/freebsd-pkgng-add
  - ee2cbb574a Fix pkgng provider to work with a sources list and the underlying pkg-add(8)
- **PR** #25677: (aneeshusa) Fix pacman.list_upgrades when refresh=True. @ *2015-07-24 16:30:06 UTC*
  - 2cad79c2f0 Merge pull request #25677 from aneeshusa/fix-pacman-list-upgrades-when-refreshing
  - 7062ae4eae Fix pacman.list_upgrades when refresh=True.
- **ISSUE** #25674: (UtahDave) file.managed with contents parameter uses wrong line endings on Windows (refs: #25675)
- **PR** #25675: (UtahDave) Use OS line endings with contents on file.managed @ *2015-07-24 16:29:50 UTC*
  - 18e739b812 Merge pull request #25675 from UtahDave/2015.5local
  - d0f9d001db Use OS line endings with contents on file.managed
- **PR** #25676: (basepi) Update release candidate docs to 2015.8.0rc2 @ *2015-07-23 20:29:37 UTC*
  - 7914f51636 Merge pull request #25676 from basepi/2015.8.0rc2releasedocs

  - 882d11836b Update release candidate docs to 2015.8.0rc2
- **ISSUE** #25665: (nmadhok) salt-cloud VMware driver fails with KeyErrors if there's any existing machine in the VMware infrastructure in (invalid state) (refs: #25666)
- **PR** #25666: (nmadhok) Check if the properties exist before looping over them causing KeyError @ *2015-07-23 17:55:40 UTC*
  - c36b714401 Merge pull request #25666 from nmadhok/vmware-cloud-fix_2015.5
  - 8e812296ef Check if the properties exist before looping over them causing KeyErrors Fixes #25665
- **PR** #25656: (anlutro) Fix locale detection in debian/gentoo @ *2015-07-23 16:46:40 UTC*
  - 36d04b2954 Merge pull request #25656 from alprs/fix-locale_detection
  - a260236942 change variable name
  - dd2a188c05 fix tests
  - aefd0fb374 code formatting
  - e58d222fb0 fix locale detection in debian/gentoo
- **PR** #25661: (rallytime) Back-port #25624 to 2015.5 @ *2015-07-23 16:26:48 UTC*
  - **PR** #25624: (bobrik) Fix typo in get_routes example for debian_ip (refs: #25661)
  - b1c1735aae Merge pull request #25661 from rallytime/bp-25624
  - 4e1fcfa15e Fix typo in get_routes example for debian_ip
- **ISSUE** #15209: (hubez) file.manage: source_hash not working with s3:// (2014.7.0rc1) (refs: #25638)
- **PR** #25662: (rallytime) Back-port #25638 to 2015.5 @ *2015-07-23 16:26:40 UTC*
  - **PR** #25638: (TronPaul) fix bad merge in 99fc7ec (refs: #25662)
  - 6a2843dee2 Merge pull request #25662 from rallytime/bp-25638
  - 90d833d5dc fix bad merge 99fc7ec
- **ISSUE** #25413: (zizkebab) pillar_opts default behavior is not reflected in the docs (refs: #25644)
- **PR** #25644: (cachedout) pillar doc fix @ *2015-07-22 22:57:23 UTC*
  - 00f4689fe3 Merge pull request #25644 from cachedout/issue_25413
  - 8cef61e6cc pillar doc fix
- **ISSUE** #25540: (dennisjac) salt highstate schedule cannot be removed (refs: #25642)
- **PR** #25642: (cachedout) Warn on pillar schedule delete @ *2015-07-22 22:04:12 UTC*
  - aeaeb53ed6 Merge pull request #25642 from cachedout/issue_25540
  - 74f6b6930c Warn on pillar schedule delete
- **ISSUE** #25437: (lorengordon) Stacktrace on Windows when running pkg.list_pkgs (refs: #25598, #25763)
- **PR** #25598: (twangboy) Fixed problem trying to load file with name of boolean type @ *2015-07-22 17:07:49 UTC*
  - 7b79e433f1 Merge pull request #25598 from twangboy/fix_25437
  - c53e11d42c Fixed problem trying to load file with name of boolean type
- **ISSUE** #25323: (terminalmage) unit.modules.tls_test fails with older mock (refs: #25604)
- **PR** #25604: (terminalmage) Move patching of mock_open to within test @ *2015-07-22 16:53:55 UTC*
  - f4a38a8aee Merge pull request #25604 from terminalmage/fix-mock_open

  - 123b8ee1cb Fix mock_open patch
  - af82835f42 Move patching of mock_open to within test
- **ISSUE** saltstack/salt-bootstrap#630: (jf) Ubuntu 12.04 (and maybe 12.x?): apt-get installing python-requests causes digital_ocean_v2 to fail with "[ERROR] Failed to get the output of 'digital_ocean.avail_sizes()': 'Response' object has no attribute 'text'" (refs: #25609)
  - **PR** saltstack/salt-bootstrap#627: (nyushi) Fix tornado installation on ubuntu (refs: #25609)
- **PR** #25609: (s0undt3ch) [2015.5] Update the bootstrap script to latest release v2015.07.22 @ *2015-07-22 16:28:52 UTC*
  - 224484df7e Merge pull request #25609 from s0undt3ch/hotfix/bootstrap-script-2015.5
  - 96a8568336 Update the bootstrap script to latest release v2015.07.22
- **ISSUE** #21912: (rvora) pkg.latest not updating the package on CentOS though yum reports an update available (refs: #25603)
- **PR** #25603: (terminalmage) Add version_cmp function to yumpkg.py @ *2015-07-22 15:42:29 UTC*
  - 07eb78c79f Merge pull request #25603 from terminalmage/issue21912
  - 99e532ba74 Add versionadded directive
  - 8a1765fc6f Add version_cmp function to yumpkg.py
  - 457e72e273 Fix refernces to __salt__['version_cmp']
  - a19fa2296a Avoid using single-letter variable
- **ISSUE** #25560: (dennisjac) scheduled highstate runs don't return results to the job cache (refs: #25590)
- **PR** #25590: (garethgreenaway) 2015.5 scheduled jobs return data @ *2015-07-21 21:57:42 UTC*
  - 69ef81caba Merge pull request #25590 from garethgreenaway/25560_2015_5_schedule_return_data
  - 19ca0c0b40 Switching default in 2015.5 for whether job data in returned to the mater job_cache.
- **PR** #25584: (rallytime) Back-port #24054 and #25576 to 2015.5 @ *2015-07-21 21:16:38 UTC*
  - **PR** #25576: (pcn) s3fs breaks when fetching files from s3 (refs: #25584)
  - **PR** #24054: (mgwilliams) s3.head: return useful data (refs: #25584)
  - 9ffefc867e Merge pull request #25584 from rallytime/bp-24054-and-25576
  - aa9598e3a5 s3fs breaks when fetching files from s3
  - 1667d67c3e s3.head: return useful data
- **ISSUE** #23626: (mirko) salt state 'ssh_known_hosts' doesn't take 'port' into account (refs: #25589)
- **PR** #25589: (jahamn) Fixes ssh_known_host not taking port into account @ *2015-07-21 21:15:06 UTC*
  - a966e439d1 Merge pull request #25589 from jahamn/Fix-ssh_known_host-not-taking-port-into-account
  - 8db7ada82d Fixed pylint e8303 errors
  - 6abad29f66 Fixed pylint errors
  - 8ae6ba1290 Merge branch '2015.5' of https://github.com/saltstack/salt into Fix-ssh_known_host-not-taking-port-into-account
- **PR** #25573: (EvaSDK) Do not execute bootstrap script twice @ *2015-07-21 18:20:04 UTC*
  - **PR** #25465: (EvaSDK) 2015.5.3 LXC module fixes (refs: #25573)
  - df74f2c3ad Merge pull request #25573 from EvaSDK/2015.5.3-lxc-fixes

  - 49cec9f9a1 Use a more persistent tmp directory
  - 96a672f8e0 Do not execute bootstrap script twice
- **ISSUE** #25532: (attiasr) salt/modules/win_pkg.py list_pkgs is broken (encoding issues) (refs: #25580, #25556)
- **PR** #25580: (attiasr) use explicit utf-8 decoding (#25532) @ *2015-07-21 15:40:49 UTC*
  - 79a809dd79 Merge pull request #25580 from attiasr/patch-1
  - 4b7dc96919 use explicit utf-8 decoding (#25532)
- **ISSUE** #25206: (jfindlay) fullname issues with user.add state on windows (refs: #25568)
- **PR** #25568: (twangboy) Fixed win_useradd module to add fullname @ *2015-07-21 14:30:25 UTC*
  - 6edf196533 Merge pull request #25568 from twangboy/fix_25206
  - fbee445c6d Commented out a pylint error
  - 4b56dc3893 Fixed win_useradd module to add fullname
- **ISSUE** #21041: (deuscapturus) state module gem.installed not working on Windows. (refs: #25430, #25561, #25428)
- **PR** #25561: (twangboy) Fixed the gem module to work on windows… without injection @ *2015-07-20 21:12:15 UTC*
  - **PR** #25428: (twangboy) Fixed the gem module to work on windows (refs: #25561)
  - 3c32b0b669 Merge pull request #25561 from twangboy/fix_21041_again
  - aaf3f3dcd0 Fixed some line and style issues
  - e6d0e5cda7 Merge branch '2015.5' of https://github.com/saltstack/salt into fix_21041_again
- **PR** #25521: (cachedout) Fix outputter for state.orch @ *2015-07-20 19:30:14 UTC*
  - 9e19142c35 Merge pull request #25521 from cachedout/orch_outputter
  - ea40816621 Try/except
  - dd609eb440 Fix outputter for state.orch
- **PR** #25563: (basepi) [2015.5] Merge forward from 2014.7 to 2015.5 @ *2015-07-20 19:27:36 UTC*
  - 2117ac8022 Merge pull request #25563 from basepi/merge-forward-2015.5
  - 3bf2f1a722 Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.5
  - 09ebaceca8 Merge pull request #25416 from cachedout/str_2014_7
    - cc514938a8 Fix broken keyword
      - d67491bb80 Removed the logger as it's not used anymore
      - 5008bfee96 Merge branch '2015.5' of https://github.com/saltstack/salt into fix_21041_again
- **PR** #25559: (cachedout) Lint win_pkg @ *2015-07-20 17:46:29 UTC*
  - 50c257b1d5 Merge pull request #25559 from cachedout/lint_win_pkg
  - 53a00add99 Lint win_pkg
- **ISSUE** #25532: (attiasr) salt/modules/win_pkg.py list_pkgs is broken (encoding issues) (refs: #25580, #25556)
- **PR** #25556: (attiasr) fix for #25532 @ *2015-07-20 17:45:11 UTC*
  - 7c7015ccda Merge pull request #25556 from attiasr/patch-1
  - 9b224e8d4e fix for #25532

- **ISSUE** #25538: (stanislavb) S3 ext_pillar configuration requires verify_ssl (refs: #25554)
- **PR** #25554: (jfindlay) verify_ssl=True for s3 ext pillar @ *2015-07-20 17:43:38 UTC*
  - 3c73dab2ce Merge pull request #25554 from jfindlay/verify_ssl
  - ca3ab4e737 verify_ssl=True for s3 ext pillar
- **PR** #25551: (rallytime) Backport #25530 to 2015.5 @ *2015-07-20 17:43:00 UTC*
  - **PR** #25530: (andre-luiz-dos-santos) The variable name must be last (refs: #25551)
  - e3e2e6718e Merge pull request #25551 from rallytime/bp-25530
  - df5003d7f9 The variable name must be last
- **PR** #25533: (attiasr) port 445 for windows bootstraping @ *2015-07-20 15:13:06 UTC*
  - 3e3441937f Merge pull request #25533 from attiasr/patch-2
  - c7fbf68597 fix windows bootstrapping
- **ISSUE** #25432: (gtmanfred) [2015.5.3][raet] raet error with SaltRaetRoadStackJoiner (refs: #25525)
- **PR** #25525: (gtmanfred) add make _prepare an alias for postinitio @ *2015-07-20 15:12:38 UTC*
  - 7fc051f56d Merge pull request #25525 from gtmanfred/2015.5
  - 43950a5bc5 add make _prepare an alias for postinitio
- **ISSUE** #25511: (rallytime) Make provider –> driver change backward compatible (refs: #25519)
- **ISSUE** #23574: (CedNantes) Failed to Deploy Salt-Minion on a Win 2012 R2 using wmware Cloud Driver from Develop branch (refs: #25519)
- **PR** #25519: (rallytime) Backport vmware driver to 2015.5 branch @ *2015-07-20 15:11:26 UTC*
  - 725d1a40d0 Merge pull request #25519 from rallytime/backport_vmware
  - 35e13eef1d Don't reference driver in older salt versions
  - f011890217 Add vmware back-port change to release notes
  - 0f4f560b38 Backport vmware driver to 2015.5 branch
- **PR** #25542: (Oro) Fix hipchat.send_message when using API v2 @ *2015-07-20 15:09:13 UTC*
  - 2f0d695bc0 Merge pull request #25542 from Oro/fix-hipchat-v2-sendmessage
  - 3a9f5b037f Fix hipchat.send_message when using API v2
- **PR** #25531: (rallytime) Back-port #25529 to 2015.5 @ *2015-07-18 19:16:10 UTC*
  - **PR** #25529: (davidjb) Fix minor typo in best practice example (refs: #25531)
  - 390aa7d28f Merge pull request #25531 from rallytime/bp-25529
  - 3e24381439 Fix minor typo in best practice example
- **PR** #25528: (davidjb) Fix typo in extend declaration doco @ *2015-07-18 14:22:06 UTC*
  - 6e811bfdd2 Merge pull request #25528 from davidjb/patch-7
  - bfc4f9fd85 Fix typo in extend declaration doco
- **ISSUE** #25486: (whiteinge) Highstate outputter not used for state.apply (refs: #25517)
- **PR** #25517: (rallytime) Back-port #25486 to 2015.5 @ *2015-07-17 21:49:26 UTC*
  - **PR** #25485: (attiasr) fix file downloads on windows

  - b9abd723a7 Merge pull request #25517 from rallytime/bp-25485
  - 6c2f3180c2 fix file downloads on windows
- **ISSUE** #25479: (alexandrsushko) multiple mount.mounted of one device (refs: #25483)
- **PR** #25516: (rallytime) Back-port #25483 to 2015.5 @ *2015-07-17 21:49:05 UTC*
  - **PR** #25483: (alexandrsushko) Added 'none' to the set of specialFSes (refs: #25516)
  - 9cb436fbae Merge pull request #25516 from rallytime/bp-25483
  - e0af6e3478 Added 'none' to the set of specialFSes
- **ISSUE** #25493: (blackduckx) Issue with job_args on schedule.add command (refs: #25513)
- **PR** #25513: (garethgreenaway) fixes to schedule.add documentation in 2015.5 @ *2015-07-17 17:03:24 UTC*
  - daf03efb7c Merge pull request #25513 from garethgreenaway/25493_2015_5_schedule_add_documentation
  - bc2414bc4d Fixing documentation for schedule.add when using the job_args parameter, value needs to be be in quotes for the value to be passed in as an array.
- **PR** #25465: (EvaSDK) 2015.5.3 LXC module fixes (refs: #25573) @ *2015-07-17 15:57:54 UTC*
  - 48050cd287 Merge pull request #25465 from EvaSDK/2015.5.3-lxc-fixes
  - 170eb52cc4 Fix use of undefined cmd when install of boostrap script fails
  - 86118f4a7b Install bootstrap script like dns and systemd check scripts in container
  - 978e6d56e2 Error out if configdir could not be created when preparing LXC container
  - 41b6c3c2bf Fix typo in redirecting shell output to /dev/null
  - 456393d4db Fix DNS script cleanup
- **ISSUE** saltstack/salt-bootstrap#611: (BretFisher) SmartOS doesn't detect missing git, fails install (refs: #25506)
- **ISSUE** saltstack/salt-bootstrap#607: (bechtoldt) (git install) change source of init scripts for debian based systems (refs: #25506)
- **ISSUE** saltstack/salt-bootstrap#602: (rallytime) Ubuntu 14.10 Won't Bootstrap with Latest Stable (refs: #25506)
- **ISSUE** saltstack/salt-bootstrap#598: (babilen) Installation fails on Debian 7 due to missing easy_install (refs: #25506)
- **ISSUE** saltstack/salt#25456: (julienlavergne) [2015.8.0rc1] salt-bootstrap fails to install salt master (refs: #25506)
- **ISSUE** saltstack/salt#25270: (iggy) [2015.8.0rc1] salt-bootstrap fails to properly install a minion (refs: #25506)
- **ISSUE** #619: (syphernl) Only send over changed files during state.highstate (refs: #'saltstack/salt-bootstrap#621'_)
  - **PR** saltstack/salt-bootstrap#625: (hasues) Modify bootstrap-salt.sh unbound error with CONFIG_PROTECT_MASK for Gentoo (refs: #25506)
  - **PR** saltstack/salt-bootstrap#624: (BretFisher) fix config and etc path on SmartOS (refs: #25506)
  - **PR** saltstack/salt-bootstrap#621: (lomeroe) python-jinja2 has been moved to rhui-...server-releases-optional repo... (refs: #25506)
  - **PR** saltstack/salt-bootstrap#606: (babilen) Switch to httpredir.debian.org as default Debian mirror (refs: #25506)
  - **PR** saltstack/salt-bootstrap#455: (denmat) PR: Issue 394 (refs: #25506)

- **PR** #25506: (s0undt3ch) [2015.5] Update bootstrap script to latest stable release, v2015.07.17 @ *2015-07-17 15:40:38 UTC*
  - f85f2b49fd Merge pull request #25506 from s0undt3ch/hotfix/bootstrap-script
  - ab6aaa6e60 Update bootstrap script to latest stable release, v2015.07.17
- **ISSUE** #25454: (mschiff) Regression: salt 2015.5 not working in secure chroot anymore. (refs: #25498)
- **PR** #25498: (jfindlay) only read /proc/1/cmdline if it exists @ *2015-07-17 15:35:33 UTC*
  - c8caf406b2 Merge pull request #25498 from jfindlay/jail_init
  - c63a6c206f only read /proc/1/cmdline if it exists
- **PR** #25487: (rallytime) Back-port #25464 to 2015.5 @ *2015-07-16 16:58:36 UTC*
  - **PR** #25464: (jquast) docfix: "cache_jobs: False" => grains_cache: False" (refs: #25487)
  - 3f695a17cf Merge pull request #25487 from rallytime/bp-25464
  - e947d8ec5a docfix: "cache_jobs: False" => grains_cache: False"
- **PR** #25482: (oeuftete) Fix docker.running detection of running container @ *2015-07-16 16:58:29 UTC*
  - 331808eb7d Merge pull request #25482 from oeuftete/docker-running-is-running-fix-2015-5
  - b69379ba50 Fix docker.running detection of running container
- **ISSUE** #25384: (rickh563) pyopenssl 0.14 requirement in 2015.5.3 does not work in RHEL6 : ZD-364 (refs: #25468)
- **PR** #25468: (joejulian) Add support for pyOpenSSL > 0.10 @ *2015-07-16 15:10:30 UTC*
  - 7a20ecbf46 Merge pull request #25468 from joejulian/use_pyopenssl_0_10
  - 1b7a56aa38 Add support for pyOpenSSL > 0.10
- **PR** #25467: (rallytime) Add lxml dependency to opennebula docs @ *2015-07-16 15:09:57 UTC*
  - d169905170 Merge pull request #25467 from rallytime/lxml_dep
  - d326f4f686 Add lxml dependency to opennebula docs
- **ISSUE** #25250: (wipfs) 'force' option in copy state deletes target file (refs: #25461, #25710)
- **ISSUE** #24647: (nmadhok) salt.states.file.copy does not copy the file if it already exists with force=True (refs: #25461)
- **PR** #25461: (jahamn) Update file, if force option and content not same @ *2015-07-15 20:15:07 UTC*
  - 89649456e0 Merge pull request #25461 from jahamn/fixed-file.copy-force-option-deleting-files-without-updating-them
  - 32cf1ebbb5 Update file, if force option and content not same
- **ISSUE** #25431: (namcois) Digital Ocean v2 reducing API calls by adding per_page (refs: #25438)
- **PR** #25438: (rallytime) Reduce digital_ocean_v2 API call frequency @ *2015-07-15 19:40:18 UTC*
  - 146a81b7c3 Merge pull request #25438 from rallytime/do_v2
  - faf49ea2a3 Add page number change release notes
  - da6ab82837 Reduce digital_ocean_v2 API call frequency
- **PR** #25457: (jacksontj) Saltnado @ *2015-07-15 17:50:12 UTC*
  - **PR** #25427: (tony-cocco) Saltnado runner client results in blocking call despite being set-up as Runner.async (refs: #25457)

  - cb98d79cdd Merge pull request #25457 from jacksontj/saltnado
  - bc32f66b98 Add runner_async endpoint to salnado
  - b043fa9b05 Better name of method process manager is starting
- **PR** #25459: (jahamn) Fixed 'defulats' typo in verify.py @ *2015-07-15 16:53:06 UTC*
  - 3f72eb5486 Merge pull request #25459 from jahamn/fix-defulats-typo-in-verify.py
  - 9bafd19f67 Fixed 'defulats' typo in verify.py
- **PR** #25426: (jquast) bugfix: trailing "…done" in rabbitmq output (backport from 'develop' to 2015.5) @ *2015-07-15 14:48:05 UTC*
  - 73566188cf Merge pull request #25426 from jquast/2015.5
  - 005a7ca2a3 bugfix: trailing "…done" in rabbitmq output
- **PR** #25433: (jleroy) Support for IPv6 addresses scopes in network.interfaces (ifconfig) @ *2015-07-15 14:44:09 UTC*
  - **PR** #25151: (jleroy) Support for IPv6 addresses scopes in network.interfaces (refs: #25274, #25433)
  - cfec990062 Merge pull request #25433 from jleroy/ipv6-scope-support-ifconfig
  - bc36d05c0c Support for IPv6 addresses scopes in network.interfaces (ifconfig)
- **ISSUE** #21041: (deuscapturus) state module gem.installed not working on Windows. (refs: #25430, #25561, #25428)
- **PR** #25430: (twangboy) Disabled rbenv execution module for Windows @ *2015-07-15 14:41:18 UTC*
  - a425230c19 Merge pull request #25430 from twangboy/fix_21041_2
  - 242fc21765 Disabled rbenv execution module for Windows
    - 8b2dc681f9 Fixed the gem module to work on windows… without injection
    - c7466e7894 Fixes ssh_known_host to take port into account
- **ISSUE** #1846: (seanchannel) development dependencies
- **PR** #25420: (techhat) Move S3 to use AWS Signature Version 4 @ *2015-07-14 22:03:09 UTC*
  - 9313804e27 Merge pull request #25420 from techhat/s3sig4
  - 3edf3a14e2 Linting
  - c63c2356be Move S3 to use AWS Signature Version 4
- **ISSUE** #20441: (deuscapturus) State module file.managed returns an error on Windows and test=Test (refs: #25418)
- **PR** #25418: (twangboy) Fixed problem with file.managed test=True @ *2015-07-14 21:26:59 UTC*
  - 30a41d3f51 Merge pull request #25418 from twangboy/fix_20441
  - d8957856cd Fixed problem with file.managed test=True
- **PR** #25417: (ahus1) extended documentation about dependencies for dig module @ *2015-07-14 20:49:51 UTC*
  - 3805677e93 Merge pull request #25417 from ahus1/patch-1
  - 3cd194ebaf Update dig.py
  - 287f8f76e3 extended documentation about dependencies
- **PR** #25411: (basepi) [2015.5] Merge forward from 2014.7 to 2015.5 @ *2015-07-14 17:55:26 UTC*

  - 4d929071e1 Merge pull request #25411 from basepi/merge-forward-2015.5
  - 33d2451fef Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.5
  - 2a1dd1113f Merge pull request #25375 from cachedout/config_fix_2014_7
    * c041f2905f Fix error in config.py for master_type
  - 2590e23d48 Merge pull request #25324 from jacobhammons/doc-theme-updates
    * 88f5fcf58d Latest help theme updates
- **PR** #25406: (anlutro) Force arguments to aptpkg.version_cmp into strings @ *2015-07-14 16:15:41 UTC*
  - 81bed62d16 Merge pull request #25406 from alprs/fix-apt_version_cmp_types
  - d56efd1341 force arguments to apt_pkg.version_compare into strings
- **PR** #25408: (rallytime) Back-port #25399 to 2015.5 @ *2015-07-14 16:09:06 UTC*
  - **PR** #25399: (jarpy) Demonstrate per-minion client_acl. (refs: #25408)
  - cd9ea63ff2 Merge pull request #25408 from rallytime/bp-25399
  - da9c0eb673 Typo in client_acl ref doc.
  - 50e0baf270 Demonstrate per-minion client_acl.
- **PR** #25240: (tankywoo) file make os.walk only be called one @ *2015-07-14 16:04:49 UTC*
  - ef9f6b0ce0 Merge pull request #25240 from tankywoo/fix-files-os-walk-multiple-times
  - 8044def1c0 file make os.walk only be called one
- **PR** #25395: (rallytime) Back-port #25389 to 2015.5 @ *2015-07-14 03:26:34 UTC*
  - **PR** #25389: (l2ol33rt) Adding entropy note for gpg renderer (refs: #25395)
  - d02f388b08 Merge pull request #25395 from rallytime/bp-25389
  - a086e5ad35 Adding entropy note
- **PR** #25392: (rallytime) Back-port #25256 to 2015.5 @ *2015-07-14 03:25:13 UTC*
  - **PR** #25256: (yanatan16) Dont assume source_hash exists (refs: #25392)
  - 008e3295c6 Merge pull request #25392 from rallytime/bp-25256
  - 6b2da4d582 Dont assume source_hash exists
- **PR** #25398: (twangboy) Fix date @ *2015-07-14 03:21:17 UTC*
  - 3f278963ae Merge pull request #25398 from twangboy/fix_date
  - 52824f9602 Added /V1 /Z to remove scheduled task after run
  - a055cca79f Changed date of scheduled task to work in other locales
- **PR** #25397: (GideonRed-zz) Introduce standard error output when cli exits with non-zero status @ *2015-07-14 03:20:24 UTC*
  - 978d9f7117 Merge pull request #25397 from GideonRed/2015.5
  - ea7ab27f31 Introduce standard error output when cli exits with non-zero status
- **ISSUE** #24444: (michaelkrupp) file.managed does not handle dead symlinks (refs: #25383)
- **PR** #25386: (cachedout) Lint #25383 @ *2015-07-13 21:01:10 UTC*
  - **PR** #25383: (jahamn) Fix manage_file function in salt/modules/file.py to handle broken sym… (refs: #25386)

  - 09442abbde Merge pull request #25386 from cachedout/lint_25383
  - 7694299170 Lint #25383
- **ISSUE** #24444: (michaelkrupp) file.managed does not handle dead symlinks (refs: #25383)
- **PR** #25383: (jahamn) Fix manage_file function in salt/modules/file.py to handle broken sym… (refs: #25386) @ *2015-07-13 20:58:23 UTC*
  - 47bcc61f55 Merge pull request #25383 from jahamn/Fix-file.managed_not_handling_dead_symlinks
  - ab17aa160e Fix manage_file function in salt/modules/file.py to handle broken symlinks
- **PR** #25369: (anlutro) Fix aptpkg.version_cmp @ *2015-07-13 20:18:45 UTC*
  - c9fe10e7aa Merge pull request #25369 from alprs/fix-apt_version_cmp
  - 6391b15b3e fix aptpkg.version_cmp
- **ISSUE** #25337: (eliasp) *salt-call* from non-existend cwd backtraces (refs: #25379)
- **PR** #25379: (jfindlay) check for cwd before getting it @ *2015-07-13 19:50:27 UTC*
  - beb0238392 Merge pull request #25379 from jfindlay/check_wd
  - 6e4547ff38 check for cwd before getting it
- **ISSUE** #25320: (podloucky-init) zypper module list_upgrades broken (2015.5.2) (refs: #25334)
- **PR** #25334: (jfindlay) return all cmd info back to zypper fcn @ *2015-07-13 17:03:29 UTC*
  - 274622ad9b Merge pull request #25334 from jfindlay/fix_zyp
  - c1e633903e return all cmd info back to zypper fcn
- **PR** #25339: (jfindlay) update orchestration docs @ *2015-07-13 16:04:26 UTC*
  - 71859c6593 Merge pull request #25339 from jfindlay/orch_doc
  - 0447808d95 clarify, motivate orchestration docs
- **ISSUE** #22241: (masterkorp) Salt master not properly generating the map (refs: #25358)
- **PR** #25358: (dkiser) Deep merge of pillar lists (refs: #26016) @ *2015-07-13 15:51:01 UTC*
  - 90a1ca02a3 Merge pull request #25358 from dkiser/22241_pillar_merge_lists
  - d030e289b3 Deep merge of pillar lists
- **ISSUE** #25281: (shinshenjs) Unless usage in Official Doc syntax error? (refs: #25346)
- **PR** #25346: (bechtoldt) set correct indention in states/requisites.rst (docs), fixes #25281 @ *2015-07-13 15:34:45 UTC*
  - 66c619fd71 Merge pull request #25346 from bechtoldt/issue25281
  - 8eb2ac1dbe set correct indention in states/requisites.rst (docs), fixes #25281
- **PR** #25336: (terminalmage) Don't try to read init binary if it wasn't found @ *2015-07-13 09:45:30 UTC*
  - b122ed931d Merge pull request #25336 from terminalmage/fix-init-grain
  - f473918a53 Don't try to read init binary if it wasn't found
- **PR** #25350: (davidjb) Fix documentation for file.blockreplace @ *2015-07-13 03:41:20 UTC*
  - 1805bafc89 Merge pull request #25350 from davidjb/patch-4
  - e13a9fd74e Fix documentation for file.blockreplace
- **ISSUE** #19288: (oba11) AssociatePublicIpAddress doesnt work with salt-cloud 2014.7.0 (refs: #25326, #20972)

- **PR** #25326: (rallytime) Back-port #20972 to 2015.5 @ *2015-07-10 18:49:44 UTC*
  - **PR** #20972: (JohannesEbke) Fix interface cleanup when using AssociatePublicIpAddress in #19288 (refs: #25326)
  - b0196fccb7 Merge pull request #25326 from rallytime/bp-20972
  - 51c941f59d Also fix cleanup of interfaces when using AssociatePublicIpAddress in #19288
- **ISSUE** #24433: (chrimi) Salt locale state fails, if locale has not been generated (refs: #25290)
- **PR** #25327: (rallytime) Back-port #25290 to 2015.5 @ *2015-07-10 18:49:37 UTC*
  - **PR** #25290: (pcdummy) Simple fix for locale.present on Ubuntu. (refs: #25327)
  - 28450d124e Merge pull request #25327 from rallytime/bp-25290
  - 20032c55f3 Simple fix for locale.present on Ubuntu.
- **ISSUE** #24827: (yermulnik) locale.present doesn't generate locales (refs: #25309)
- **PR** #25328: (rallytime) Back-port #25309 to 2015.5 @ *2015-07-10 17:22:59 UTC*
  - **PR** #25309: (davidjb) Format /etc/locale.gen correctly in salt.modules.localemod.gen_locale (refs: #25328)
  - 8f666a24f3 Merge pull request #25328 from rallytime/bp-25309
  - 44d44ec574 Format /etc/locale.gen correctly on gen_locale
- **PR** #25322: (jacobhammons) version change to 2015.5.3 @ *2015-07-10 16:11:24 UTC*
  - 0a33a1d8bb Merge pull request #25322 from jacobhammons/release-2015.5.3
  - 19f88920fa version change to 2015.5.3
- **PR** #25308: (jacksontj) Make clear commands trace level logging @ *2015-07-10 14:20:06 UTC*
  - **PR** #24737: (jacksontj) Move AES command logging to trace (refs: #25308)
  - 2f0f59b6cb Merge pull request #25308 from jacksontj/2015.5
  - 60fc770ba2 Make clear commands trace level logging
- **ISSUE** #24520: (nvx) Tomcat module fails to extract version number from snapshot builds (2015.5 regression) (refs: #24927)
- **PR** #25269: (jfindlay) Extract tomcat war version @ *2015-07-10 01:28:21 UTC*
  - **PR** #24927: (egarbi) Tomcat module fails to extract version number from snapshot builds #2… (refs: #25269)
  - 9b6646d578 Merge pull request #25269 from jfindlay/tomcat
  - fd4fca172d consolidate tomcat exec and state version extract
  - 59dc833567 update tomcat war_deployed state tests
  - edca458b6c Fixed 2 blank lines around import re
  - 7e528d1050 Tomcat module fails to extract version number from snapshot builds #24520
- **ISSUE** #18808: (amendlik) Add command line argument to select pillar environment (refs: #25238)
- **PR** #25238: (DmitryKuzmenko) Pillarenv backport 2015.5 @ *2015-07-10 01:25:07 UTC*
  - 0f82ac3e30 Merge pull request #25238 from DSRCompany/pillarenv_backport_2015.5
  - 98792eb179 Pillarenv support in minion config, cp and cmdmod modules.

  - 88ff576f39 Support pillarenv cmdline in state.sls. Backport of PR #23719
- **ISSUE** #13943: (Supermathie) Powershell commands that expect input hang forever (refs: #25299)
- **PR** #25299: (twangboy) Added -NonInteractive so powershell doesn't hang waiting for input @ *2015-07-09 21:00:16 UTC*
  - 219d4cad9c Merge pull request #25299 from twangboy/fix_13943
  - c05889031f Added -NonInteractive so powershell doesn't hang waiting for input
- **PR** #25301: (jacobhammons) bug fix for module function display in help @ *2015-07-09 20:46:34 UTC*
  - 1c43892a80 Merge pull request #25301 from jacobhammons/doc-bugs
  - f6561289af bug fix for module function display in help
- **ISSUE** #25277: (jacobhammons) CherryPy recommended versions (refs: #25279)
- **PR** #25279: (jacobhammons) Additional docs on external and master job cache, assorted doc fixes @ *2015-07-09 16:46:26 UTC*
  - 68149bc686 Merge pull request #25279 from jacobhammons/job-cache-docs
  - 57dfa92d5a Fixed typos
  - 2f9e5b9125 Additional docs on external and master job cache, assorted doc fixes Refs #25277
- **ISSUE** #25268: (lichtamberg) Salt not working anymore in 2015.8/develop: ValueError: 'scope' is not in list (refs: #25274)
- **PR** #25274: (jleroy) Fix for issue #25268 @ *2015-07-09 13:36:26 UTC*
  - **PR** #25151: (jleroy) Support for IPv6 addresses scopes in network.interfaces (refs: #25274, #25433)
  - 972fa2fb54 Merge pull request #25274 from jleroy/25268-fix
  - 2c698d204b Fix for issue #25268
- **PR** #25272: (twangboy) Fixed problem with service not starting @ *2015-07-08 23:29:48 UTC*
  - 8ebb73df2d Merge pull request #25272 from twangboy/service
  - e61eeba48b Fixed problem with service not starting
- **ISSUE** #25223: (nmadhok) Runner occasionally fails with a RuntimeError when fired by a reactor (refs: #25225)
- **PR** #25225: (nmadhok) Backporting fix for issue #25223 on 2015.5 branch @ *2015-07-08 15:16:18 UTC*
  - c6efd2356c Merge pull request #25225 from nmadhok/client-runtime-fix-backport-2015-2
  - 391b7d6730 Backporting fix for issue #25223 on 2015.2 branch
- **PR** #25214: (rallytime) A couple of doc fixes for the http tutorial @ *2015-07-07 22:23:07 UTC*
  - 207fbaeac4 Merge pull request #25214 from rallytime/http_doc
  - d0b61f3fc1 A couple of doc fixes for the http tutorial
- **ISSUE** #24272: (rallytime) Fix boto_vpc_test moto version check (refs: #25194)
- **PR** #25194: (rallytime) Update moto version check in boto_vpc_test and update min version @ *2015-07-07 18:27:32 UTC*
  - 9dd5cd8a8e Merge pull request #25194 from rallytime/fix-24272
  - f959e165a1 Clean up imports
  - fbc9c0d6bf Fix Pylint

  - fe2561f415 Update moto version check in boto_vpc_test and update min version
- **PR** #25205: (basepi) Update releasecandidate docs @ *2015-07-07 15:25:24 UTC*
  - a3e9486c28 Merge pull request #25205 from basepi/releasecandidatedocs
  - 452880d4aa Update releasecandidate docs
- **PR** #25187: (UtahDave) Doc fixes: Fix misspelling and remove extraneous double spaces @ *2015-07-07 01:07:04 UTC*
  - fbafd39a46 Merge pull request #25187 from UtahDave/fix_misspelling
  - 65abb63003 remove some extraneous double spaces
  - c423b62aa5 fix misspelling
- **PR** #25182: (cachedout) Try to re-pack long floats as strs @ *2015-07-07 01:06:43 UTC*
  - ddee90ce23 Merge pull request #25182 from cachedout/pack_long_floats
  - a192ecfd74 Try to re-pack long ints as strs
- **ISSUE** #23822: (sidcarter) Zip file extracted permissions are incorrect (refs: #25128)
- **PR** #25185: (rallytime) Back-port #25128 to 2015.5 @ *2015-07-07 00:58:00 UTC*
  - **PR** #25128: (stanislavb) Use cmd_unzip to preserve permissions (refs: #25185)
  - df9982b836 Merge pull request #25185 from rallytime/bp-25128
  - 1726057c8a Use cmd_unzip to preserve permissions
- **PR** #25181: (rallytime) Back-port #25102 to 2015.5 @ *2015-07-07 00:57:13 UTC*
  - **PR** #25102: (derBroBro) Update win_network.py (refs: #25181)
  - df0bb8c831 Merge pull request #25181 from rallytime/bp-25102
  - 64d8f14417 Update win_network.py
  - 6789c5b8e8 Update win_network.py
- **ISSUE** #24301: (iggy) influxdb_user and influxdb_database states need virtual functions (refs: #25059)
- **PR** #25179: (rallytime) Back-port #25059 to 2015.5 @ *2015-07-07 00:56:44 UTC*
  - **PR** #25059: (babilen) Add virtual functions to influxdb state modules (refs: #25179)
  - 04fdd7b0ee Merge pull request #25179 from rallytime/bp-25059
  - 1eeefbd2ab Add virtual functions to influxdb state modules
- **ISSUE** #18919: (giner) Windows: pkg.refresh_db returns false-positive success (refs: #25196)
- **PR** #25196: (twangboy) Fixed #18919 false-positive on pkg.refresh @ *2015-07-07 00:24:13 UTC*
  - 58b7d0e653 Merge pull request #25196 from twangboy/pkg_refresh
  - 12ffcd1062 Fixed #18919 false-positive on pkg.refresh
- **PR** #25180: (rallytime) Back-port #25088 to 2015.5 @ *2015-07-06 20:33:45 UTC*
  - **PR** #25088: (supertom) Update (refs: #25180)
  - 4a406aca45 Merge pull request #25180 from rallytime/bp-25088
  - 4078c8db25 added message recommending JSON file be used if the libcloud version is >= 0.17.0
- **PR** #25191: (basepi) Add extrndest back to fileclient.is_cached in 2015.5 @ *2015-07-06 19:35:24 UTC*

  - **PR** #25117: (basepi) Fix fileclient.is_cached (refs: #25191)
  - 01ed062ca7 Merge pull request #25191 from basepi/fix.fileclient.is_cached
  - 5fa74f4408 Add back in the extrndest stuff (which is now in develop)
- **ISSUE** #25016: (martinhoefling) salt-run doc.execution fails with AttributeError (refs: #25020)
- **PR** #25175: (rallytime) Back-port #25020 to 2015.5 @ *2015-07-06 18:53:19 UTC*
  - **PR** #25020: (martinhoefling) Fix for issue #25016 (refs: #25175)
  - a9404aea5c Merge pull request #25175 from rallytime/bp-25020
  - da2e1704ea Fix for issue #25016
- **ISSUE** #21879: (bechtoldt) Reference pages in documentation are outdated again (refs: #25019, #21880)
- **ISSUE** #19262: (bechtoldt) salt.pillar.file_tree doesn't appear in the documentation (refs: #25019)
- **PR** #25173: (rallytime) Partial back-port of #25019 @ *2015-07-06 18:52:59 UTC*
  - **PR** #25019: (bechtoldt) add missing module documentation to references (refs: #25173)
  - **PR** #24421: (bechtoldt) add missing module documentation (refs: #25019)
  - **PR** #21880: (bechtoldt) update references, fixes #21879 (refs: #25019)
  - **PR** #20039: (bechtoldt) completing some doc references (refs: #25019)
  - c70fec65b8 Merge pull request #25173 from rallytime/partial-bp-25019
  - c0c2463b64 Partial backport of #25019
- **PR** #25171: (rallytime) Back-port #25001 to 2015.5 @ *2015-07-06 18:51:53 UTC*
  - **PR** #25001: (jasonkeene) Add docs for key arg in ssh_known_hosts.present (refs: #25171)
  - c5ba9a90ba Merge pull request #25171 from rallytime/bp-25001
  - a891108793 Add docs for key arg in ssh_known_hosts.present
- **PR** #25170: (rallytime) Back-port #24982 to 2015.5 @ *2015-07-06 16:34:43 UTC*
  - **PR** #24982: (asyncsrc) ec2 network_interfaces fix (refs: #25170)
  - 3e06602545 Merge pull request #25170 from rallytime/bp-24982
  - 3e6eab3ae9 ec2 network_interfaces fix
- **PR** #25161: (aneeshusa) Allow checking for non-normalized systemd units. @ *2015-07-06 15:15:31 UTC*
  - 09602808a0 Merge pull request #25161 from aneeshusa/allow-checking-non-normalized-systemd-service-availability
  - b4d544fe70 Allow checking for non-normalized systemd units.
- **PR** #25151: (jleroy) Support for IPv6 addresses scopes in network.interfaces (refs: #25274, #25433) @ *2015-07-06 14:43:03 UTC*
  - 3599b8abab Merge pull request #25151 from jleroy/ipv6-scope-support
  - edce034e6c Support for IPv6 addresses scopes in network.interfaces
- **ISSUE** #24979: (mavenAtHouzz) [Discussion] Support for more than 1 netapi.rest_tornado server process (refs: #25149)
- **PR** #25166: (cachedout) Lint #25149 @ *2015-07-06 14:40:29 UTC*
  - **PR** #25149: (jacksontj) Saltnado multiprocess support (refs: #25166)

  - 66d6365a9f Merge pull request #25166 from cachedout/lint_saltnado
  - 2fe167edf8 Lint #25149
- **ISSUE** #24979: (mavenAtHouzz) [Discussion] Support for more than 1 netapi.rest_tornado server process (refs: #25149)
- **PR** #25149: (jacksontj) Saltnado multiprocess support (refs: #25166) @ *2015-07-06 14:38:43 UTC*
  - 2f1bad1c01 Merge pull request #25149 from jacksontj/saltnado
  - 6aa5548e2d Enable multiprocess support in saltnado
  - 9a1351eada Change print to logger, so we can set a level and log exc_info
- **PR** #25120: (d--j) add missing continue for exception case @ *2015-07-02 19:38:45 UTC*
  - a723af0f10 Merge pull request #25120 from d--j/patch-2
  - 81d5d15dce add missing continue for error case
- **PR** #25117: (basepi) Fix fileclient.is_cached (refs: #25191) @ *2015-07-02 19:38:26 UTC*
  - 6e2222241a Merge pull request #25117 from basepi/fix.fileclient.is_cached
  - 38e243fdfb Add fix from merge forward
  - 52f35f761a Add import
  - 23c32a7518 Backport develop version of salt.fileclient.is_cached
- **PR** #25087: (0xf10e) Fix execution module for glance - now based on 2015.5! @ *2015-07-02 19:36:27 UTC*
  - c80990ba4f Merge pull request #25087 from 0xf10e/fix_glance_2015.5
  - 7749cc081c PEP8 W601…
  - bbda079fa5 fix pylint E302, E502, E713, E1305
  - 3baacc72b4 use Glance API v1 for image_create
  - c3d6134da1 making pylint marginally happier
  - 19a20bf228 get valid properties for image_show() from the schema for "image"
  - 0c6a61173a add some debugging, fix a few AttributeErrors
  - aceca0e20d fix return of glance.image_show()
  - a47509e7dd fix return of image_list
  - 9f923edfab Change confusing "nt_ks" to "g_client"
  - fa2bd1a79c bit of docs/comments in image_create()
  - 5c34d0c494 merge 439b1e42053239b into 2015.5
  - 7a3cf27948 update attributes for image_show output
  - b1bec0f1a1 fix retry w/ user/pass if token fails
  - 2f4ef6683c update attributes for image_list output
  - eef3bc7048 use _auth() from neutron plus keystoneclient,
- **PR** #25129: (basepi) [2015.5] Merge forward from 2014.7 to 2015.5 @ *2015-07-02 17:37:40 UTC*
  - 549ee47420 Merge pull request #25129 from basepi/merge-forward-2015.5
  - 187268d879 Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.5

  - 36d53ef59e Merge pull request #25093 from jaybocc2/2014.7
    - c6a501ebda quick fix for issue #18447
  - 38903a94a1 Merge pull request #25069 from puneetk/patch-1
  - f0b4e600e6 Update Documentation to clarify version added
  - f8dc6030e7 Pylint updates , removing whitespace
  - 532d315dd1 [Code Review update] renamed function to is_enaled from list_enabled
  - 20b0462289 Update schedule.py
  - 4f1471d7fb Add a helper module function called list_enabled
  - **PR** saltstack/salt#24798: (justinta) Revert ”adding states/postgres_database unit test case.” (refs: #25114)
  - **PR** saltstack/salt#24329: (jayeshka) adding states/postgres_database unit test case. (refs: #‘saltstack/salt#24798‘_)
- **PR** #25114: (jfindlay) Revert ”Revert ”adding states/postgres_database unit test case.”” @ *2015-07-02 01:01:29 UTC*
  - 86f2791fdb Merge pull request #25114 from saltstack/revert-24798-revert-24329-postgres_database-states-unit-test
  - 071ee44d41 Revert ”Revert ”adding states/postgres_database unit test case.””
- **PR** #24362: (jayeshka) adding states/postgres_user unit test case. @ *2015-07-01 21:45:31 UTC*
  - bf8c7e7a9d Merge pull request #24362 from jayeshka/postgres_user-states-unit-test
  - fd1d834688 adding states/postgres_user unit test case.
- **PR** #24361: (jayeshka) adding states/postgres_schema unit test case. @ *2015-07-01 21:44:56 UTC*
  - 4195cea512 Merge pull request #24361 from jayeshka/postgres_schema-states-unit-test
  - 0558b0d744 adding states/postgres_schema unit test case.
- **PR** #24331: (jayeshka) adding states/postgres_extension unit test case. @ *2015-07-01 21:43:58 UTC*
  - ada8fe57d4 Merge pull request #24331 from jayeshka/postgres_extension-states-unit-test
  - 3d465a574a adding states/postgres_extension unit test case.

28.2.72 Salt 2015.5.5 Release Notes

release 2015-08-20

Version 2015.5.5 is a bugfix release for *2015.5.0*.

Statistics

- Total Merges: **33**
- Total Issue References: **28**
- Total PR References: **39**
- Contributors: **20** (TheBigBear, arthurlogilab, basepi, bastiaanb, cachedout, driskell, garethgreenaway, jacobhammons, jahamn, jfindlay, rallytime, s0undt3ch, scottjpack, silenius, sixninetynine, stanislavb, terminalmage, thusoy, twangboy, vr-jack)

Changelog for v2015.5.4..v2015.5.5

Generated at: 2018-05-27 22:04:18 UTC

- **ISSUE** #26484: (thusoy) Git state leaks HTTPS user/pw to log (refs: #26486)
- **ISSUE** #26482: (thusoy) Git states doesn't allow user-only auth (refs: #26483)
- **PR** #26486: (thusoy) Git: Don't leak https user/pw to log @ *2015-08-20 16:04:52 UTC*
  - **PR** #26483: (thusoy) Handle user-only http auth in git module (refs: #26486)
  - 28aa9b1058 Merge pull request #26486 from thusoy/git-confidential-auth
  - 5289165487 Git: Don't leak https user/pw to log
- **ISSUE** #26432: (centromere) Documentation incorrectly references salt-key on the minion (refs: #26476)
- **ISSUE** #26403: (adelcast) Grains documentation incorrectly states they are static (refs: #26476)
- **ISSUE** #26329: (cro) Add note to eauth docs indicating default PAM service. (refs: #26476)
- **ISSUE** #26264: (grep4linux) state trees cannot have 'dots' in the name (refs: #26476)
- **ISSUE** #26233: (dove-young) pip install salt, then start master failed on Fedora 22 (refs: #26476)
- **PR** #26476: (jacobhammons) Minor doc bug fixes @ *2015-08-19 22:52:35 UTC*
  - 679ba5ee0a Merge pull request #26476 from jacobhammons/doc-bugs
  - 499bd66378 Minor doc bug fixes Refs #26403 Refs #26432 Refs #26233 Refs #26264 Refs #26329
- **ISSUE** #26366: (GreatSnoopy) The development tree produces hanging, 100%cpu salt-master processes (refs: #26443)
- **ISSUE** #26301: (waynew) CPU pegged out running salt-master (after running command) (refs: #26443)
- **ISSUE** #25998: (driskell) Event subsystem discarding required events during –batch breaking it for slow running commands (refs: #26000)
- **PR** #26443: (cachedout) Fix connect issue in event init @ *2015-08-19 22:50:22 UTC*
  - **PR** #26000: (driskell) Implement full event caching for subscribed tags (refs: #26443)
  - 42b8c1b3f4 Merge pull request #26443 from cachedout/fix_event_sub
  - 560977bc7e Fix connect issue in event init
- **ISSUE** #26343: (jfindlay) batch error when no minions match target (refs: #26445)
- **PR** #26445: (cachedout) Raise clean error when no minions targeted in batch mode @ *2015-08-19 22:50:07 UTC*
  - d2df1a86ad Merge pull request #26445 from cachedout/issue_26343
  - 1600f3eccd Raise clean error when no minions targeted in batch mode
- **ISSUE** #26482: (thusoy) Git states doesn't allow user-only auth (refs: #26483)
- **PR** #26483: (thusoy) Handle user-only http auth in git module (refs: #26486) @ *2015-08-19 22:47:41 UTC*
  - a9b28e9577 Merge pull request #26483 from thusoy/git-user-only-auth
  - 09fc934acc Handle user-only http auth in git module
- **PR** #26496: (jfindlay) add dateutil dependency reporting @ *2015-08-19 22:46:31 UTC*
  - edc04930ae Merge pull request #26496 from jfindlay/dateutil
  - cbe330e78b add dateutil dependency reporting

- **PR** #26494: (cachedout) Remove unecessary debug statements @ *2015-08-19 20:46:00 UTC*
    - 4fff53b842 Merge pull request #26494 from cachedout/remove_debug_statements
    - d717a43dcc Remove unecessary debug statements
- **PR** #26465: (rallytime) Back-port #26457 to 2015.5 @ *2015-08-19 16:08:16 UTC*
    - **PR** #26457: (arthurlogilab) docstring improvement for network.ping module execution (refs: #26465)
    - f46a0dab5d Merge pull request #26465 from rallytime/bp-26457
    - b3f638ff0f docstring improvement for network.ping module execution
- **PR** #26434: (s0undt3ch) Fix missed typo @ *2015-08-18 18:14:29 UTC*
    - c1458980f3 Merge pull request #26434 from s0undt3ch/2015.5
    - 06dcaefcaa Fix missed typo
- **ISSUE** #26426: (alxbse) Private/public IPs are interchanged when listing nova driver cloud nodes (refs: #26430)
- **PR** #26430: (rallytime) List public and private ips under the correct label @ *2015-08-18 16:20:32 UTC*
    - 0f64be710f Merge pull request #26430 from rallytime/fix-26426
    - 2ba97316c9 List public and private ips under the correct label
- **PR** #26431: (rallytime) Back-port #26417 to 2015.5 @ *2015-08-18 15:41:58 UTC*
    - **PR** #26417: (scottjpack) Changed t1 -> t2 micro (refs: #26431)
    - 913451a414 Merge pull request #26431 from rallytime/bp-26417
    - 0254a2e90e Changed t1 -> t2 micro
- **PR** #26378: (stanislavb) Fix EC2 credentials from IAM roles for s3fs and s3 ext_pillar in 2015.5 @ *2015-08-18 14:01:53 UTC*
    - 952da7abaf Merge pull request #26378 from stanislavb/2015.5
    - 39ce3127cd Let utils.aws query instance metadata
- **ISSUE** #26245: (bradthurber) salt v2015.5.3 gitfs.py using newer pygit2 feature than required minimum (refs: #26420)
- **PR** #26420: (terminalmage) Only use pygit2.errors if it exists (2015.5 branch) @ *2015-08-18 14:00:01 UTC*
    - 09e96dce39 Merge pull request #26420 from terminalmage/issue26245-2015.5
    - 19a1149067 Only use pygit2.errors if it exists (2015.5 branch)
- **PR** #26409: (basepi) [2015.5] Merge forward from 2014.7 to 2015.5 @ *2015-08-17 23:19:56 UTC*
    - c5eb6bbd3e Merge pull request #26409 from basepi/merge-forward-2015.5
    - dafed10a9e Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.5
    - da8bca09aa Merge pull request #26242 from cro/anonldap4
        - a0d2ab1eed Remove dead code
    - 1ecf23773e Merge pull request #26216 from cro/anonldap3
    - af132d7b89 Documentation update for anonymous bind issue.
    - 2ef54b6b13 Documentation update for anonymous bind issue.
    - 5b1836bb00 Fix issue with LDAP anonymous binds.
- **ISSUE** #26404: (ssgward) Syntax error in lvm.vg_absent state causing failure (refs: #26406)

- **PR** #26406: (jfindlay) fix syntax error in lvm exec module @ *2015-08-17 21:18:25 UTC*
    - 741ca6b4db Merge pull request #26406 from jfindlay/lvm
    - 81d351ff8f fix syntax error in lvm exec module
- **PR** #26405: (TheBigBear) dependency zip files moved to new site @ *2015-08-17 21:17:24 UTC*
    - a7e2d30e2a Merge pull request #26405 from TheBigBear/patch-8
    - 8898d64918 dependency zip files moved to new site
- **PR** #26298: (vr-jack) Keep $HOME from being interpretted by Master shell @ *2015-08-17 21:15:11 UTC*
    - cf0523a12e Merge pull request #26298 from vr-jack/2015.5
    - 1fd6fc6ce3 Keep $HOME from being interpretted by Master shell
- **PR** #26324: (s0undt3ch) Salt is now pip install'able in windows @ *2015-08-17 20:41:34 UTC*
    - c0811d3302 Merge pull request #26324 from s0undt3ch/2015.5
    - e7cb3be2a0 Document the added options
    - 92af1c9572 Fix argument name
    - 72d2fdb512 Add *pypiwin32 >= 219* as a windows install requires.
    - b1105fc706 Allow mimicking the install setup command for develop/editable installations.
    - 26246a72ee Allow writing Salt's _version.py when installing in develop mode.
    - 71928f2194 Prefer HTTPS, fix url argument
    - 7b25430cc7 Download the necessary DLLs for windows
    - 86692a92cd Install PyCrypto from a wheel in repo.saltstack.com under Windows
    - 915da594c2 Skip M2Crypto in Windows.
    - 1ea426e299 Move code to properly handle default requirements.
    - 8fda8c0db3 M2CryptoWin{32,64} should only be installed on Salt < 2015.8.0
    - 0ff2f19aee Override the develop command in cmdclass
    - a5aa752a85 Override the develop command when WITH_SETUPTOOLS is set
    - 4d6841c761 Install M2CryptoWin{32,64} while installing Salt
- **ISSUE** #26161: (bastiaanb) salt initscripts do not set lock file in /var/lock/subsys as required on RedHat family OSes (refs: #26371)
- **PR** #26371: (bastiaanb) fix issue #26161: on RedHat family systems touch /var/lock/subsys/$SE… @ *2015-08-17 20:39:28 UTC*
    - 87151736c5 Merge pull request #26371 from bastiaanb/fix/issue-26161-salt-initscripts-dont-set-lockfile
    - ec8d4b0470 test wether RETVAL is 0 with -eq rather than =.
    - a83a5de41e fix issue #26161: on RedHat family systems touch /var/lock/subsys/$SERVICE to ensure the daemon will be stopped on shutdown.
- **ISSUE** #25801: (themalkolm) Update docs that salt.states.winrepo requires *roles:salt-master* in grains. (refs: #26328)
- **ISSUE** #25562: (jefftucker) winrepo state does not run on masterless minion (refs: #26328)
- **PR** #26402: (twangboy) Removed documentation no longer required @ *2015-08-17 20:35:37 UTC*

  - **PR** #26328: (twangboy) Removed salt-master role requirement (refs: #26402)
  - 89602f56ad Merge pull request #26402 from twangboy/fix_26328
  - ad5fa03b76 Removed documentation no longer required
- **PR** #26392: (rallytime) Back-port #26376 to 2015.5 @ *2015-08-17 19:39:51 UTC*
  - **PR** #26376: (TheBigBear) minor edit spelling (refs: #26392)
  - eb373e5904 Merge pull request #26392 from rallytime/bp-26376
  - a013bb5b3d minor edit
- **ISSUE** #16049: (ryan-lane) boto_elb.present state requires attributes argument (refs: #26342)
- **PR** #26342: (rallytime) Don't call boto_elb._attributes_present if no attributes were provided @ *2015-08-17 19:19:08 UTC*
  - 8bb57d1631 Merge pull request #26342 from rallytime/fix-16049
  - 211f6feaf5 Fix test failures - get_attributes shouldn't be called if none are provided
  - d8ad023e88 Don't call boto_elb._attributes_present if no attributes were provided
- **ISSUE** #26155: (silenius) pip availability in states/pip_state (refs: #26160)
- **PR** #26389: (rallytime) Back-port #26160 to 2015.5 @ *2015-08-17 19:09:16 UTC*
  - **PR** #26160: (silenius) proposed fix for #26155 (refs: #26389)
  - 2fd1e06343 Merge pull request #26389 from rallytime/bp-26160
  - f0bc3765d9 No logging should happen on __virtual__
  - ca406eaf3c proposed fix for #26155
- **ISSUE** #26266: (o-sleep) limit pw_user.getent() from returning entire corporate list (refs: #26300)
- **PR** #26300: (jfindlay) mock pwd function calls in pw_user exec module @ *2015-08-17 18:56:41 UTC*
  - 0046c6cfed Merge pull request #26300 from jfindlay/pw_test
  - 7e94989403 mock pwd calls in pw_user exec mod test
  - 26f5b466f5 check for pwd on linux and BSD user exec mods
- **ISSUE** #24334: (afletch) autosign_timeout not honoured (refs: #26386)
- **PR** #26386: (jahamn) Fixes autosign_timeout usage in check_autosign_dir @ *2015-08-17 18:34:40 UTC*
  - 709499438b Merge pull request #26386 from jahamn/fix-autosign_timeout
  - b2fa2ac9d3 Fixes autosign_timeout usage in check_autosign_dir
- **ISSUE** #25801: (themalkolm) Update docs that salt.states.winrepo requires *roles:salt-master* in grains. (refs: #26328)
- **ISSUE** #25562: (jefftucker) winrepo state does not run on masterless minion (refs: #26328)
- **PR** #26328: (twangboy) Removed salt-master role requirement (refs: #26402) @ *2015-08-17 18:30:17 UTC*
  - 8d901d7b15 Merge pull request #26328 from twangboy/fix_25562
  - d4ca1dccbf Removed salt-master role requirement
- **ISSUE** #26327: (bradthurber) mount.mounted opts incorrect "forced unmount and mount because options (tcp) changed" (refs: #26362)
- **PR** #26362: (garethgreenaway) Fixes to mount state. @ *2015-08-17 17:44:55 UTC*

  - 74558f5743 Merge pull request #26362 from garethgreenaway/2015_5_26327_more_invisible_mount_options
  - cf532d46dd Some mount options are translated to different options once a share has been mounted, eg. when specifying a protocol for NFS as either tcp or udp this option is translated into either proto=tcp or proto=udp. Change adds a lookup dictionary for these options so that a re-mount isn't forced each time.
- **PR** #26379: (s0undt3ch) [2015.5] Backport #26353 @ *2015-08-17 17:19:29 UTC*
  - **PR** #26353: (sixninetynine) fixed a typo in setup.py (refs: #26379)
  - 7dbbd90c98 Merge pull request #26379 from s0undt3ch/issues/backport-26353
  - 33ed315c85 fixed Packaing -> Packaging typo and added a couple comments on the setuptools/distutils abstract methods
- **ISSUE** #26240: (0xf10e) keystone.user_get raises exception when user is not found (refs: #26277)
- **PR** #26277: (rallytime) Handle exception when user is not found in keystone.user_get @ *2015-08-14 19:41:59 UTC*
  - bcca1b4c5a Merge pull request #26277 from rallytime/fix-26240
  - 0b6977335e Clean it up
  - 5edabfd271 It's a dict - git problems…
  - 39d3eb66f0 Log error and return error - make returns consistent.
  - 496474d862 Handle exception when user is not found in keystone.get_user
- **ISSUE** #24484: (bailsman) clouds/ec2.py: create_snapshot throws exception (refs: #26326)
- **PR** #26326: (rallytime) Make ec2.create_snapshot return less unweildly and more relevant @ *2015-08-14 19:40:47 UTC*
  - 78be3a826f Merge pull request #26326 from rallytime/create_snapshot_return
  - c5395db851 Make ec2.create_snapshot return less unweildly and more relevant
- **ISSUE** #16179: (UtahDave) Salt Cloud -l debug includes the entire bootstrap script twice in its output (refs: #26306)
- **PR** #26306: (rallytime) Move VM creation details dict to log.trace @ *2015-08-14 17:39:52 UTC*
  - 44c9d3063b Merge pull request #26306 from rallytime/fix-16179
  - 670464258f Move VM creation details dict to log.trace

28.2.73 Salt 2015.5.6 Release Notes

release 2015-10-13

Version 2015.5.6 is a bugfix release for *2015.5.0*.

Statistics

- Total Merges: **145**
- Total Issue References: **71**
- Total PR References: **178**

- Contributors: **53** (Arabus, JensRantil, PierreR, SaltyCharles, TheBigBear, abh, aboe76, anlutro, arthurlogilab, aspyatkin, basepi, benhosmer, bersace, cachedout, carlpett, damonzheng, derphilipp, dmyerscough, dsumsky, efficks, eguven, garethgreenaway, hexedpackets, jacksontj, jacobhammons, jfindlay, joejulian, johanek, julianbrost, kev009, lorengordon, madprog, marccardinal, netroby, nmadhok, plastikos, rallytime, serge-p, spudfkc, stanislavb, styro, systembell, tankywoo, techhat, terminalmage, thatch45, tjstansell, twangboy, vakulich, vtek21, whiteinge, zmalone, zyio)

Security Fixes

CVE-2015-6941 The Windows *user* module and `salt-cloud` display passwords in log when log level is set to `debug` or more verbose.

For the Windows *user* module, the password is now replaced with the string `XXX-REDACTED-XXX`.

For salt-cloud, debug logging no longer displays `win_password` and `sudo_password` authentication credentials.

CVE-2015-6918 Git state/execution modules log HTTPS auth credentials when log level is set to `debug` or more verbose.

These credentials are now replaced with `REDACTED` in the debug output. Thanks to Andreas Stieger <asteiger@suse.com> for bringing this to our attention.

Changelog for v2015.5.5..v2015.5.6

Generated at: 2018-05-27 22:13:00 UTC

- **PR** #27582: (jfindlay) add 2015.5.6 release notes @ *2015-09-30 22:33:48 UTC*
  - 304dc68f7f Merge pull request #27582 from jfindlay/2015.5
  - 4f0d55cda6 add 2015.5.6 release notes
- **ISSUE** #27518: (srkunze) [Docs] Relationship between Mine and Grains (refs: #27557)
- **PR** #27557: (jfindlay) add doc motivating mine vs grains @ *2015-09-30 17:49:46 UTC*
  - 7201ce71e4 Merge pull request #27557 from jfindlay/mine_doc
  - 3727d79bad edit mine doc for style and markup
  - 7e037a4666 add doc motivating mine vs grains
- **ISSUE** #27478: (rominf) iptables state fails to save rules (refs: #27515)
- **PR** #27515: (jfindlay) save iptables rules on SuSE @ *2015-09-30 16:09:42 UTC*
  - 59c3d5f93e Merge pull request #27515 from jfindlay/suse_fire
  - 4460ad2785 save iptables rules on SuSE
- **ISSUE** #27460: (llevar) Orchestrate runner not resolving reference to a built in state (refs: #27509)
- **PR** #27509: (jfindlay) tell the user why the gluster module does not work @ *2015-09-30 15:49:16 UTC*
  - 9b26357b19 Merge pull request #27509 from jfindlay/gluster_reason
  - 1ccda538d2 tell the user why the gluster module does not work
- **ISSUE** #27372: (GregMeno) pip.installed state fails when env_vars is not a dict (refs: #27379)
- **PR** #27379: (jfindlay) document and check dict type for pip env_vars @ *2015-09-30 02:56:52 UTC*
  - 989733ea86 Merge pull request #27379 from jfindlay/pip_vars

  - aee51ffdef document and check dict type for pip env_vars
- **PR** #27516: (basepi) [2015.5] Merge forward from 2014.7 to 2015.5 @ *2015-09-29 17:53:33 UTC*
  - 6d773f66c3 Merge pull request #27516 from basepi/merge-forward-2015.5
  - a08951f0fa Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.5
  - 5262f01325 Merge pull request #27335 from rallytime/cloud-logging-7
    - adeb1dcad4 Pylint Fix
    - 588c13783c Salt-cloud logging clean up for windows functions
    - 9b6000135c [2014.7] Fixup salt-cloud logging
- **ISSUE** #27447: (junster1) Fix mysql table size for salt_events (refs: #27472)
- **PR** #27472: (cachedout) Change recommeded schema for data field in mysql event table @ *2015-09-29 15:49:37 UTC*
  - 68d784c3dd Merge pull request #27472 from cachedout/fix_27447
  - 5e745ad6da Change recommeded schema for data field in mysql event table
- **PR** #27468: (cachedout) Fix 27351 @ *2015-09-29 15:35:29 UTC*
  - **PR** #27351: (SaltyCharles) fix sysctl truncating newline on os x (refs: #27468)
  - ee6e0ed057 Merge pull request #27468 from cachedout/fix_27351
  - 0bc37c0d41 Fix test
  - f9a19720de fix sysctl truncating newline on os x
- **ISSUE** #27438: (aboe76) can't set system locale on OpenSuse SUse (refs: #27479)
- **PR** #27479: (aboe76) fix locale on opensuse and suse #27438 @ *2015-09-29 15:34:48 UTC*
  - a214c7f84e Merge pull request #27479 from aboe76/fix_locale_suse
  - a8f2dad1be fix locale on opensuse and suse #27438
- **ISSUE** #17103: (arthurlogilab) salt is looking for outputters in /var/cache/salt/minion/extmods/output not /var/cache/salt/minion/extmods/outputputters (refs: #27483)
- **PR** #27483: (rallytime) Outputters should sync to output, not outputters, on the minion. @ *2015-09-29 15:33:08 UTC*
  - 931f593b51 Merge pull request #27483 from rallytime/fix-17103
  - 441241eb90 Change sync_outputters to sync_output for consistency, but alias sync_outputters
  - 105528720b Outputters should sync to output, not outputters, on the minion.
- **PR** #27484: (rallytime) Back-port #27434 and #27470 to 2015.5 @ *2015-09-29 15:32:03 UTC*
  - **PR** #27470: (cachedout) Minor doc fixup. (refs: #27484)
  - **PR** #27434: (netroby) Doc: copy key to server via ssh-copy-id (refs: #27484, #27470)
  - 9c2c028953 Merge pull request #27484 from rallytime/bp-27434-and-27470
  - 5de2ee35ab Minor doc fixup.
  - af656c7e87 Doc: copy key to server via ssh-copy-id
- **ISSUE** #27433: (TheBigBear) winrepo - drops "trailing zeroes" from version numbers on un-install? (refs: #27469)

- **PR** #27469: (twangboy) Added quotes to version numbers example @ *2015-09-28 21:54:43 UTC*
  - 927874d316 Merge pull request #27469 from twangboy/fix_27433
  - a996ea46e2 Added quotes to version numbers example
- **ISSUE** #27342: (ariscn) File.managed silent fail for contents_pillar (refs: #27375, #27467)
- **PR** #27467: (cachedout) file.managed: check contents_{pillar|grain} result @ *2015-09-28 20:22:16 UTC*
  - **PR** #27375: (jfindlay) file.managed: check contents_{pillar|grain} result (refs: #27467)
  - 382a53403f Merge pull request #27467 from cachedout/lint_27375
  - 4e54a98f5e Lint #27375
  - 278ade52d2 file.managed: check contents_{pillar|grain} result
- **ISSUE** #9856: (jeremyBass) for grant in grants: TypeError: 'bool' object is not iterable (refs: #27419)
- **PR** #27419: (rallytime) Amend error log to include multiple tips for troubleshooting. @ *2015-09-28 17:53:19 UTC*
  - ed6207a438 Merge pull request #27419 from rallytime/fix-9856
  - 551396564a Ammend error log to include multiple tips for troubleshooting.
- **ISSUE** #16753: (johtso) Duplicate selector in top file gives unhelpful traceback (refs: #27426)
- **PR** #27426: (rallytime) Don't stacktrace if there are conflicting id errors in highstate @ *2015-09-28 14:52:51 UTC*
  - 73fa89edf7 Merge pull request #27426 from rallytime/fix-16753
  - f6cbd81e66 Don't stacktrace if there are conflicting id errors in highstate
- **ISSUE** #27406: (s-iraheta) salt-cloud error with Softlayer (Bare Metal Instance): TypeError: 'bool' object is not iterable and with –list-locations: Failed to get the output of 'softlayer_hw.avail_locations()': 142776 (refs: #27408)
- **PR** #27408: (rallytime) Fix avail_locations function for the softlayer_hw driver in 2015.5 @ *2015-09-25 23:34:50 UTC*
  - 5dd1b70475 Merge pull request #27408 from rallytime/fix-27406-for-2015.5
  - 39a4ae5a6c Remove hdd: 19 refs from SL docs - no longer available from SoftLayer.
  - de2f9234d3 Use correct default for bandwith
  - 42d8127f79 Don't set the optional_products default to a boolean, and then try to loop.
  - 9d8a3d8303 Fix avail_locations function for the softlayer_hw driver in 2015.5
- **ISSUE** #27389: (ryan-lane) Docs layout issue (refs: #27410)
- **PR** #27410: (jacobhammons) Fix css layout Refs #27389 @ *2015-09-25 22:38:48 UTC*
  - 8f9a3cfbaf Merge pull request #27410 from jacobhammons/doc-updates
  - a9fdecada1 Fix css layout Refs #27389 sample typo fix in linux_acl additional module folders listed in dynamic-modules
- **PR** #27336: (rallytime) [2015.5] Fixup salt-cloud logging @ *2015-09-24 15:02:52 UTC*
  - 3746085587 Merge pull request #27336 from rallytime/cloud-logging-five
  - 7956b36076 [2015.5] Fixup salt-cloud logging
- **ISSUE** #27356: (lorengordon) file.replace fails if *repl* contains special regex characters and *append_if_not_found=True* (refs: #27358)

- **PR** #27358: (lorengordon) Escape search replacement text, fixes #27356 @ *2015-09-24 13:52:46 UTC*
  - 5a3be10a3e Merge pull request #27358 from lorengordon/escape-search-replacement-text
  - 88bb1fbfff Escape search replacement text, fixes #27356
- **ISSUE** #19236: (bramhg) salt-cloud : Unable to add SSD disk and unable to auto-delete disk on instance termination on GCE (refs: #27345)
- **PR** #27345: (rallytime) Allow use of rst header links by separating options out from yaml example @ *2015-09-23 19:48:56 UTC*
  - 6759f79d6d Merge pull request #27345 from rallytime/docs-for-19236
  - 1d3925bbfb Added version tag for ex_disk_type option
  - f23369300c Allow use of rst header links by separating options out from yaml example
- **PR** #26903: (bersace) Review defaults.get @ *2015-09-23 14:52:20 UTC*
  - c2efb291e2 Merge pull request #26903 from bersace/fix-defaults-modules
  - 474d7afc95 fixup! Review defaults loading
  - 36141d226e fixup! Review defaults loading
  - 62b6495358 fixup! Review defaults loading
  - cf0624e8b8 fixup! Review defaults loading
  - 2c58bab977 fixup! Review defaults loading
  - 82c5b1d8fd Review defaults loading
- **ISSUE** #27316: (efficks) Extracted state with zip format failed on Windows (refs: #27317)
- **PR** #27317: (efficks) State unzip should use unzip command instead of unzip_cmd. @ *2015-09-23 14:41:36 UTC*
  - a372466922 Merge pull request #27317 from efficks/fix27316
  - bf216c101e State unzip should use unzip command instead of unzip_cmd. Issue #27316
- **ISSUE** #15514: (flyaruu) Calling a boto_route53.present state fails if the record is already there (refs: #27309)
- **PR** #27309: (rallytime) Change a value list to a comma-separated string in boto_route53.present @ *2015-09-23 14:30:50 UTC*
  - bd3771e80f Merge pull request #27309 from rallytime/fix-15514
  - 9383d91ff8 Change a value list to a comma-separated string in boto_route53.present
- **ISSUE** #27297: (JensRantil) file.replace documentation improvement (refs: #27311)
- **PR** #27311: (jfindlay) discuss replacement occurrences in file doc @ *2015-09-22 22:23:10 UTC*
  - b5fe944875 Merge pull request #27311 from jfindlay/maxoc
  - 8ec2e921bd discuss replacement occurrences in file doc
- **PR** #27310: (basepi) [2015.5] Merge forward from 2014.7 to 2015.5 @ *2015-09-22 21:08:41 UTC*
  - ca4597b93a Merge pull request #27310 from basepi/merge-forward-2015.5
  - 7b75e4aed1 Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.5
  - e90412d3b8 Merge pull request #27252 from jfindlay/version.2014.7
    - 3d28307a00 2014.7 -> 2014.7.0
- **ISSUE** #27307: (terminalmage) Regression in yumpkg's refresh_db function (refs: #27308)

- **PR** #27308: (terminalmage) Fix refresh_db regression in yumpkg.py @ *2015-09-22 21:07:28 UTC*
  - 982c21c79f Merge pull request #27308 from terminalmage/fix-refresh_db-regression
  - 77686fb7ce Fix refresh_db regression in yumpkg.py
- **PR** #27286: (terminalmage) Add a configurable timer for minion return retries @ *2015-09-22 16:35:07 UTC*
  - 775a4f9ad0 Merge pull request #27286 from terminalmage/return_retry_timer
  - 540a7dfcf1 Add default values for new minion config options
  - 453b883820 Add a configurable timer for minion return retries
- **PR** #27278: (rallytime) Back-port #27256 to 2015.5 @ *2015-09-21 19:27:51 UTC*
  - **PR** #27256: (julianbrost) Fix error handling in salt.modules.file.statvfs (refs: #27278)
  - 02482c0572 Merge pull request #27278 from rallytime/bp-27256
  - 1beddf6311 Fix error handling in salt.modules.file.statvfs
- **PR** #27277: (rallytime) Back-port #27230 to 2015.5 @ *2015-09-21 19:06:14 UTC*
  - **PR** #27230: (benhosmer) Fix typo in AWS doc config (refs: #27277)
  - e36c019c37 Merge pull request #27277 from rallytime/bp-27230
  - 3ce77db1bc Fix typo in AWS doc config
- **PR** #27253: (jfindlay) 2015.5 -> 2015.5.0 @ *2015-09-18 23:44:43 UTC*
  - b22286476e Merge pull request #27253 from jfindlay/version.2015.5
  - 967e3bb72a 2015.5 -> 2015.5.0
- **PR** #27244: (garethgreenaway) Exception in cloud.ec2.create_snapshot @ *2015-09-18 21:41:11 UTC*
  - 51a0193b54 Merge pull request #27244 from garethgreenaway/ec2_create_snapshot_no_return_data_exception
  - 820fd576b9 Fixing the cause when the r_data from aws.query is empty and an exception happens when looking for the snapshotID
- **ISSUE** #27215: (wfhu) cron.file override the crontab file even if there's no change (refs: #27231)
- **PR** #27231: (jfindlay) only write cron file if it is changed @ *2015-09-18 18:23:10 UTC*
  - 26540f15bc Merge pull request #27231 from jfindlay/cronchange
  - 1e335297e2 only write cron file if it is changed
- **PR** #27233: (basepi) [2015.5] Add stub release notes for 2015.5.6 @ *2015-09-18 16:55:40 UTC*
  - 579f375f74 Merge pull request #27233 from basepi/release.notes.stubs
  - f4563ea9b7 Add stub release notes for 2015.5.6
- **ISSUE** #25423: (tweenk) Impossible to define a file.managed for use only as a template in "use" requisites (refs: #27208)
- **PR** #27208: (basepi) [2015.5] Add test.nop state @ *2015-09-18 16:50:17 UTC*
  - f5a322e3f2 Merge pull request #27208 from basepi/nop.state.25423
  - 9414b05b2c Add test.nop example
  - a84ce67b8f Add test.nop state
- **ISSUE** #27187: (SeverinLeonhardt) ssh_known_hosts.present hashes other entries even with hash_hostname: false (refs: #27201)

- **PR** #27201: (jfindlay) rename hash_hostname to hash_known_hosts @ *2015-09-18 15:45:03 UTC*
    - 59a07cae68 Merge pull request #27201 from jfindlay/sshhash
    - 1b620b77cd rename hash_host arg to hash_known_hosts
    - 12f14ae37c update hash_known_hosts docs in ssh module
- **PR** #27214: (jacksontj) Correctly support https, port 443 is not a requirement @ *2015-09-18 15:43:05 UTC*
    - 560545c4c5 Merge pull request #27214 from jacksontj/2015.5
    - e7526bdb44 Correctly support https, port 443 is not a requirement
- **ISSUE** #18582: (mainframe) Allow merging file_roots and pillar_roots from different config files included from master.d (refs: #27150)
- **PR** #27172: (rallytime) Back-port #27150 to 2015.5 @ *2015-09-17 17:25:51 UTC*
    - **PR** #27150: (cachedout) Merge config values from master.d/minion.d conf files (refs: #27172)
    - 7a34c7742d Merge pull request #27172 from rallytime/bp-27150
    - 0d7ee4b209 Merge config values from master.d/minion.d conf files
- **PR** #27194: (rallytime) Back-port #27180 to 2015.5 @ *2015-09-17 16:17:24 UTC*
    - **PR** #27180: (tankywoo) file copy ret result True if no change in test mode (refs: #27194)
    - e956d88f5f Merge pull request #27194 from rallytime/bp-27180
    - 327d343fef file copy ret result True if no change in test mode
- **PR** #27176: (basepi) [2015.5] Merge forward from 2014.7 to 2015.5 @ *2015-09-17 15:00:40 UTC*
    - a02d043309 Merge pull request #27176 from basepi/merge-forward-2015.5
    - 66f4641be3 Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.5
    - c186e51764 Merge pull request #27117 from jacobhammons/release-docs-2014.7
        - b69e11e0a4 made 2014.7 an archived release minor doc site updates
    - 69d758ee2b Merge pull request #27114 from cachedout/warn_on_insecure_log
        - 507fb04683 Issue warning that some log levels may contain sensitive data
    - aa71bae8aa Merge pull request #27075 from twangboy/fix_password_2014.7
        - c0689e3215 Replaced password with redacted when displayed
- **PR** #27170: (rallytime) Update Getting Started with GCE docs to use cloud.profiles or cloud.profiles.d examples @ *2015-09-16 22:23:51 UTC*
    - de2027426e Merge pull request #27170 from rallytime/gce-docs
    - a07db909bd Update Getting Started with GCE docs to use cloud.profiles or cloud.profiles.d examples
- **PR** #27167: (rallytime) Back-port #27148 to 2015.5 @ *2015-09-16 19:56:01 UTC*
    - **PR** #27148: (hexedpackets) Pass file pointers to the serialize load functions. (refs: #27167)
    - 28cfdfd067 Merge pull request #27167 from rallytime/bp-27148
    - d12be52355 Pass filepointers to the serialize load functions.
- **ISSUE** #27157: (alxbse) salt.util.smb loads even when impacket library is missing (refs: #27168)
- **PR** #27168: (techhat) Add further gating of impacket library @ *2015-09-16 18:55:56 UTC*
    - 4495f4f4d0 Merge pull request #27168 from techhat/gateimpacket

  - cc448bfdc1 Add further gating of impacket library
- **ISSUE** #27100: (hexedpackets) salt-cloud –full-query does nothing when no VM profiles are configured (refs: #27166)
- **PR** #27166: (rallytime) Allow a full-query for EC2, even if there are no profiles defined @ *2015-09-16 17:41:40 UTC*
  - 3e5ef0dc30 Merge pull request #27166 from rallytime/fix-27100
  - 50fb3a489a Allow a full-query for EC2, even if there are no profiles defined
- **PR** #27162: (rallytime) Be explicit in using "SoftLayer" for service queries in SoftLayer drivers @ *2015-09-16 16:43:26 UTC*
  - f1c9de7ed9 Merge pull request #27162 from rallytime/softlayer-service
  - d281068c70 Be explicit in using "SoftLayer" for service queries in SoftLayer drivers
- **ISSUE** #27133: (deniswal) win_path.add causes the value data to be set as the value and vice versa (refs: #27149)
- **PR** #27149: (twangboy) Fixed problem with add/remove path @ *2015-09-16 15:01:48 UTC*
  - 59e9dfd8de Merge pull request #27149 from twangboy/fix_27133
  - 7992b7e20a Fixed some tests… hopefully…
  - d4c8e30f5d Fixed problem with add/remove path
- **ISSUE** #11669: (jcockhren) salt.cloud is out of date for new google compute engine dashboard and API (refs: #27147)
- **PR** #27147: (rallytime) Enforce bounds in the GCE Regex @ *2015-09-15 21:51:55 UTC*
  - 097fcd1017 Merge pull request #27147 from rallytime/fix-11669
  - 55312ea03f Provide a more friendly error message.
  - 36555856c7 Enforce bounds in the GCE Regex
- **PR** #27128: (eguven) don't show diff for test run if show_diff=False @ *2015-09-15 14:11:55 UTC*
  - f5c3f157dd Merge pull request #27128 from eguven/2015.5-fix-test-diff
  - ec2d68a84a don't show diff for test run if show_diff=False
- **PR** #27116: (jacobhammons) Update latest to 2015.8, 2015.5 is now previous @ *2015-09-15 07:34:28 UTC*
  - 088b1dbb3e Merge pull request #27116 from jacobhammons/release-docs-2015.5
  - 6e323b6dd3 Update latest to 2015.8, 2015.5 is now previous Assorted style and minor updates
- **ISSUE** #25352: (m03) reg.absent reporting incorrect results (refs: #27019)
- **PR** #27033: (jfindlay) Merge #27019 @ *2015-09-15 07:32:17 UTC*
  - **PR** #27019: (twangboy) Fixed reg state module for None, 0, and " values (refs: #27033)
  - 440855b182 Merge pull request #27033 from jfindlay/n0ne
  - 3334b9d548 fix comment and unit test for reg state
  - 391a09d5ac update reg state unit tests
  - ebbf2b05ca Fixed reg state module for None, 0, and " values
- **ISSUE** #17088: (umireon) state.dockerio.run: docked_onlyif and docked_unless do not work (refs: #26942)
- **PR** #26942: (Arabus) Fix docker.run @ *2015-09-14 18:10:54 UTC*

  - 35fc74132a Merge pull request #26942 from Arabus/fix-docker.run
  - e61e1de1f5 Fixes value typo for dockerio.loaded state
  - 39fa11b696 further linting
  - 4aec37397c Further Linting to quiet the linter
  - 7eff8ad070 Code Linting and cmd call fix
  - a51676e0eb Fixes #17088 olyif and unless should run on the host
  - d0c6128b8f Fixes #17088 retcode now returns True or False based on return status
  - 8b2e7cc4f5 Syntax clarification
- **PR** #26977: (abh) Add support for PEERNTP network interface configuration @ *2015-09-14 17:59:00 UTC*
  - 59f2a0c7ae Merge pull request #26977 from abh/2015.5-ntppeer
  - df3d6e817f Add support for PEERNTP network interface configuration on RH derived systems
- **ISSUE** #27021: (SEJeff) webutil.user_exists state does not respect test=true (refs: #27023)
- **ISSUE** #21533: (aspyatkin) Add option specifying user to run htpasswd module functions (refs: #21649)
- **PR** #27023: (jfindlay) add test support for htpasswd state mod @ *2015-09-14 17:48:00 UTC*
  - **PR** #21649: (aspyatkin) Make enhancements to htpasswd modules (refs: #27023)
  - e05b1f3951 Merge pull request #27023 from jfindlay/htwebutilpass
  - 9f3d7890a6 add test support for htpasswd state mod
- **PR** #27074: (twangboy) Replaced password with redacted when displayed @ *2015-09-14 16:27:26 UTC*
  - 9f999c0027 Merge pull request #27074 from twangboy/fix_password_2015.5
  - fdd3537456 Replaced password with redacted when displayed
- **PR** #27073: (rallytime) Remove "use develop branch" warning from LXC tutorial @ *2015-09-11 23:51:06 UTC*
  - 46b44f85ed Merge pull request #27073 from rallytime/remove-lxc-warning
  - 76c056d02b Remove "use develop branch" warning from LXC tutorial now that 2015.5.0 has been released
- **PR** #27054: (rallytime) Back-port #27029 to 2015.5 @ *2015-09-11 22:29:45 UTC*
  - **PR** #27029: (spudfkc) Removed check for no package name (refs: #27054)
  - caab21d99c Merge pull request #27054 from rallytime/bp-27029
  - 0be393be22 Removed check for no package name
- **PR** #27053: (rallytime) Back-port #26992 to 2015.5 @ *2015-09-11 22:29:30 UTC*
  - **PR** #26992: (plastikos) Summary requires full return information. (refs: #27053)
  - 0227e1cb57 Merge pull request #27053 from rallytime/bp-26992
  - 83798aff3c Do not use full return for documentation.
  - d9d5bbaa68 Summary requires full return information.
- **PR** #27052: (rallytime) Back-port #26930 to 2015.5 @ *2015-09-11 22:28:11 UTC*
  - **PR** #26930: (madprog) aptpkg.mod_repo: Raise when key_url doesn't exist (refs: #27052)
  - b72a0ef86d Merge pull request #27052 from rallytime/bp-26930
  - d9787aa318 aptpkg.mod_repo: Raise when key_url doesn't exist

- **PR** #27049: (johanek) Run repoquery less @ *2015-09-11 22:26:12 UTC*
  - 8b554dd16f Merge pull request #27049 from johanek/repoquery-dedupe
  - c113916a23 When running repoquery to check for available versions of packages, run once for all packages rather than once per package
- **PR** #27070: (stanislavb) Deprecate salt.utils.iam in Carbon @ *2015-09-11 22:01:57 UTC*
  - **PR** #26561: (stanislavb) Leave salt.utils.s3 location fallback to salt.utils.aws (refs: #27070)
  - **PR** #26446: (stanislavb) Fetch AWS region from EC2 instance metadata (refs: #26561)
  - **PR** #26378: (stanislavb) Fix EC2 credentials from IAM roles for s3fs and s3 ext_pillar in 2015.5 (refs: #26446)
  - cc2cbf9869 Merge pull request #27070 from stanislavb/2015.5
  - 1e6e5ddc9c Deprecate salt.utils.iam in Carbon
- **PR** #27030: (jfindlay) Backport #26938 @ *2015-09-11 15:10:46 UTC*
  - **PR** #27004: (vtek21) Fix 'dict' object has no attribute split (refs: #27024, #27030)
  - **PR** #26938: (derphilipp) Fixes win_path module, migrates from reg.(set|get)_key to reg.(set|get)_value (refs: #27030)
  - e23caa8ccf Merge pull request #27030 from jfindlay/winreg
  - 120fbe78e0 remove trailing line in win_path exec module
  - b36a7107b2 update win_path exec module unit tests
  - a2dc6f2dd7 Fixes win_path module, migrates from reg.(set|get)_key to reg.(set|get)_value
- **ISSUE** #25581: (b18) Salt 2015.5.2 - Could not deserialize msgpack message error. (refs: #27025)
- **PR** #27025: (cachedout) Better try and error handling for prep_jid @ *2015-09-11 07:40:10 UTC*
  - 843c28b435 Merge pull request #27025 from cachedout/issue_25581
  - ecc09d9b93 Lint
  - bfcaab9ef4 Better try and error handling for prep_jid
- **PR** #27035: (terminalmage) useradd.py: Use contextmanager to prevent leaked filehandles @ *2015-09-11 07:39:41 UTC*
  - b9baa0b39a Merge pull request #27035 from terminalmage/useradd-contextmanager
  - e430e97f6c Update user states to reflect changes to login class handling
  - f24b979c7c useradd.py: Use contextmanager to prevent leaked filehandles
- **PR** #27034: (rallytime) Update softlayer docs for where to find apikey @ *2015-09-10 22:29:56 UTC*
  - 1cdfdf7a92 Merge pull request #27034 from rallytime/softlayer-doc-fix
  - cb641f8145 Update softlayer docs for where to find apikey
- **PR** #27024: (rallytime) Back-port #27004 to 2015.5 @ *2015-09-10 21:14:21 UTC*
  - **PR** #27004: (vtek21) Fix 'dict' object has no attribute split (refs: #27024, #27030)
  - 9e06d3f01a Merge pull request #27024 from rallytime/bp-27004
  - 54d6fcf4c7 Fix 'dict' object has no attribute split
  - bb29d73c71 Fix 'dict' object has no attribute split

  - 5f1a9c46aa Fix 'dict' object has no attribute split
  - 2bfdd9724e Fix 'dict' object has no attribute split
- **PR** #27027: (rallytime) Back-port #27013 to 2015.5 @ *2015-09-10 21:13:52 UTC*
  - **PR** #27013: (nmadhok) Remove unwanted debug statement (refs: #27027)
  - 9ab2cae1e4 Merge pull request #27027 from rallytime/bp-27013
  - 19a6e9cb1c Remove unwanted debug statement.
- **PR** #27026: (rallytime) Back-port #27011 to 2015.5 @ *2015-09-10 21:13:45 UTC*
  - **PR** #27011: (whiteinge) Move giant eventlisten.sh example out of the state.event docstring (refs: #27026)
  - 2c8beb238f Merge pull request #27026 from rallytime/bp-27011
  - f8518d545f Move giant eventlisten.sh example out of the state.event docstring
- **ISSUE** #20522: (eliasp) *modules.win_pkg.install()* blindly trusts *fileclient.get_url()*/unhandled exceptions (refs: #26972)
- **PR** #26972: (twangboy) Catch the 404 error from fileclient @ *2015-09-10 20:53:12 UTC*
  - e8cdcc62f7 Merge pull request #26972 from twangboy/fix_20522
  - 0110786fa9 Catch the 404 error from fileclient
- **PR** #26951: (terminalmage) Fix timezone module for CentOS @ *2015-09-10 20:46:07 UTC*
  - fbc95f4685 Merge pull request #26951 from terminalmage/fix-timezone
  - 30a4915762 Update tests to reflect changes to timezone module
  - b6f926919f Fix timezone module for CentOS
- **PR** #26875: (marccardinal) LXC gateway provisioned only when IP is provided @ *2015-09-10 19:31:32 UTC*
  - f2ad3c333c Merge pull request #26875 from marccardinal/patch-2
  - 36d5a62262 LXC gateway provisioned only when IP is provided
- **ISSUE** #26730: (styro) __opts__['user'] on Windows minion incorrect (eg for file.symlink) (refs: #26997, #'saltstack/salt'#26899'_'_, #26899)
- **PR** #26997: (twangboy) Fixed symlinks for windows (don't use user root) @ *2015-09-10 18:54:50 UTC*
  - **PR** #26899: (twangboy) file.symlink gets windows account instead of root (refs: #26997)
  - 7b2e7b1b37 Merge pull request #26997 from twangboy/fix_symlink_windows
  - 89cc02d4e0 Added *versionadded*
  - 835177b0c8 Fixed symlinks for windows (don't use user root)
- **PR** #27001: (twangboy) Added CLI Example for reg.delete_key_recursive @ *2015-09-10 17:19:43 UTC*
  - 5389a85894 Merge pull request #27001 from twangboy/fix_reg_docs
  - 2980bbda17 Minor clarification
  - 4684b2ddd1 Added CLI example for reg.delete_key_recursive
- **PR** #26996: (jacobhammons) Beacon doc updates @ *2015-09-10 16:47:49 UTC*
  - 37814f5dff Merge pull request #26996 from jacobhammons/beacon-doc
  - e475ea688e Fixed typo
  - 2401533d9e New content added to beacon docs.

- **ISSUE** #26867: (joejulian) lvm pv's can show as not belonging to their vg if symlink is used (refs: #26868)
- **PR** #26868: (joejulian) Use the actual device name when checking vgdisplay @ *2015-09-10 16:08:16 UTC*
  - 4ba7eed711 Merge pull request #26868 from joejulian/2015.5_lvm_vg_symlink_fix
  - 3dfb33849a Use the actual device name when checking vgdisplay
- **PR** #26955: (dsumsky) S3 ext_pillar module has broken caching mechanism (backport to 2015.5) @ *2015-09-10 14:54:01 UTC*
  - 1537e945be Merge pull request #26955 from dsumsky/s3-pillar-module-cache-fix-2015.5
  - 8219acffe7 - fixed pylint warnings
  - a3b10e8ab1 - fixed broken caching in S3 ext_pillar module (file_md5 was a list) - added debugging messages - static parameters are available as module parameters now
- **PR** #26987: (rallytime) Back-port #26966 to 2015.5 @ *2015-09-09 18:42:51 UTC*
  - **PR** #26966: (TheBigBear) URL has changed (refs: #26987)
  - 3e902e86b1 Merge pull request #26987 from rallytime/bp-26966
  - 6a29eac003 URL has changed
- **PR** #26915: (rallytime) Update Joyent Cloud Tests @ *2015-09-09 15:04:50 UTC*
  - eddb532713 Merge pull request #26915 from rallytime/joyent-tests
  - d4ad42d697 Update Joyent Cloud Tests
- **PR** #26971: (rallytime) Fix a couple of typos in reactor docs @ *2015-09-09 15:03:54 UTC*
  - f86814b2a4 Merge pull request #26971 from rallytime/reactor-doc-fix
  - 0214daad19 Fix a couple of typos in reactor docs
- **ISSUE** #26730: (styro) __opts__['user'] on Windows minion incorrect (eg for file.symlink) (refs: #26997, #'saltstack/salt'#26899'_'_, #26899)
  - **PR** saltstack/salt#26899: (twangboy) file.symlink gets windows account instead of root (refs: #26976)
- **PR** #26976: (thatch45) Revert "file.symlink gets windows account instead of root" @ *2015-09-08 22:44:19 UTC*
  - 57b1080f94 Merge pull request #26976 from saltstack/revert-26899-fix_26730
  - 6dd54e6bec Revert "file.symlink gets windows account instead of root"
- **PR** #26975: (whiteinge) Remove mocks from rest_cherrypy integration tests; fix groups check bug @ *2015-09-08 22:34:08 UTC*
  - 67be01f5fe Merge pull request #26975 from whiteinge/rest_cherrypy-integration
  - 9a0989585b Add additional 'groups' check to rest_cherrypy if groups are not used
  - d68aefcfde Remove mocks from rest_cherrypy integration tests
  - 2aa3da8911 Rename the rest_cherrypy tests to conform to our convention
- **ISSUE** #26730: (styro) __opts__['user'] on Windows minion incorrect (eg for file.symlink) (refs: #26997, #'saltstack/salt'#26899'_'_, #26899)
- **PR** #26899: (twangboy) file.symlink gets windows account instead of root (refs: #26997) @ *2015-09-08 21:14:30 UTC*
  - 20a48f7f2e Merge pull request #26899 from twangboy/fix_26730
  - 9d9b3bb47a file.symlink gets windows account instead of root

- **PR** #26960: (rallytime) Fix bash code block formatting in CherryPy netapi docs @ *2015-09-08 18:14:11 UTC*
  - dbc6b862f4 Merge pull request #26960 from rallytime/cherrypy-docs
  - c1420711db Fix bash code block formatting
- **PR** #26940: (rallytime) Fix minor doc typo in client api @ *2015-09-08 04:15:00 UTC*
  - f733e048c9 Merge pull request #26940 from rallytime/api-doc-fix
  - 00fe6a225c Fix minor doc typo in client api
- **ISSUE** #26850: (jfindlay) salt-ssh error on 2015.8 (refs: #26852)
- **PR** #26871: (rallytime) Back-port #26852 to 2015.5 @ *2015-09-08 03:43:08 UTC*
  - **PR** #26852: (basepi) [2015.8] Only reference msgpack if it imported successfully (refs: #26871)
  - de9350466e Merge pull request #26871 from rallytime/bp-26852
  - 5a4c8dd2f5 Only reference msgpack if it imported successfully
- **ISSUE** #26644: (gravyboat) pkgrepo should note that for ubuntu/debian all options should not be used (refs: #26800, #26851)
- **ISSUE** #26638: (WackyOne) Suse install documentation (refs: #26800, #26851)
- **PR** #26851: (jacobhammons) states/pkgrepo examples, suse installation updates @ *2015-09-02 18:29:09 UTC*
  - a563af29d3 Merge pull request #26851 from jacobhammons/doc-bugs
  - ac3bd47440 states/pkgrepo examples, suse installation updates Refs #26644 Refs #26638
- **ISSUE** #26804: (lrhazi) gpasswd error on RHEL 5 (refs: #26817)
- **PR** #26817: (jfindlay) modify groupadd for rhel 5 @ *2015-09-02 14:52:53 UTC*
  - 5b1b934192 Merge pull request #26817 from jfindlay/grouparg
  - 82d33939f3 modify groupadd for rhel 5
- **ISSUE** #22724: (ty2u) digital_ocean_v2.py doesn't restore snapshot (refs: #26824)
- **PR** #26824: (systembell) [salt-cloud] Fix creating droplet from snapshot in digital_ocean provider @ *2015-09-02 05:18:37 UTC*
  - cdc0ea2fe3 Merge pull request #26824 from pravka/fix-droplet-creation-from-snapshot-in-dov2
  - 00e3192536 removing log
  - e4a82d78d9 removing stringification of every value in the image dict
  - cdc2b4584a fixing condition for slug check
- **ISSUE** #26805: (joejulian) cur_param referenced before assignment (refs: #26823, #26820)
- **PR** #26823: (joejulian) use dbus instead of localectl @ *2015-09-02 00:25:25 UTC*
  - 4af6951a4c Merge pull request #26823 from joejulian/ctlfix
  - a9928cb143 pep8 fixes
  - 6108ec4280 Gated dbus for os families that use it
  - e154c7b16f remove trailing spaces
  - c1c1266cc3 fix indent change
  - 0a35320aa7 Use dbus directly
- **ISSUE** #26805: (joejulian) cur_param referenced before assignment (refs: #26823, #26820)

- **PR** #26820: (jfindlay) add default param in _parse_localectl in locale mod @ *2015-09-01 22:02:17 UTC*
    - a1749b76b8 Merge pull request #26820 from jfindlay/ctlfix
    - 3a2c0d5fbb add default param in _parse_localectl in locale mod
- **ISSUE** #26788: (ssgward) Windows minion user.rename gives exception (refs: #26821)
- **PR** #26821: (twangboy) Fixed user.rename function in windows @ *2015-09-01 22:01:50 UTC*
    - ff733547c4 Merge pull request #26821 from twangboy/fix_26788
    - cf979e4877 Fixed user.rename function in windows
- **ISSUE** #26754: (jefftucker) MySQLdb-python package should be included with windows minion installer (refs: #26803)
- **PR** #26803: (twangboy) Added check for PyMySQL if MySQLdb import fails @ *2015-09-01 21:44:41 UTC*
    - c892be3255 Merge pull request #26803 from twangboy/fix_26754
    - 23576c65eb Added check for PyMySQL if MySQLdb import fails
- **ISSUE** #26798: (jfindlay) stack trace from linode driver (refs: #26815)
- **PR** #26815: (jfindlay) stringify linode id before performing str actions @ *2015-09-01 17:56:29 UTC*
    - 6edfa36083 Merge pull request #26815 from jfindlay/linstr
    - 2ff5823944 stringify linode id before performing str actions
- **ISSUE** #26644: (gravyboat) pkgrepo should note that for ubuntu/debian all options should not be used (refs: #26800, #26851)
- **ISSUE** #26638: (WackyOne) Suse install documentation (refs: #26800, #26851)
- **ISSUE** #26192: (jefftucker) Logging documentation does not exist (refs: #26800)
- **ISSUE** #26108: (ahammond) documentation around scheduling and orchestration is unclear (refs: #26800)
- **ISSUE** #24510: (ahammond) lack of documentation around Denied Keys (refs: #26800)
- **PR** #26800: (jacobhammons) Doc bug fixes @ *2015-09-01 05:40:09 UTC*
    - 135a8a64af Merge pull request #26800 from jacobhammons/doc-fixes
    - 5cca52a3c1 Fixed windows installer paths Refs #25567
    - 0ec036350d Updates to salt-ssh and salt-key #24510
    - 992edc3bb8 Doc bug fixes Refs #26192 Refs #26638 Refs #26644 Refs #26108
- **ISSUE** #24021: (arthurlogilab) [salt-cloud saltify] AttributeError: 'str' object has no attribute 'setdefault' (refs: #26793)
- **PR** #26793: (rallytime) Don't stacktrace if "name" is specified as a minion id in a map file @ *2015-08-31 19:24:25 UTC*
    - da161b9516 Merge pull request #26793 from rallytime/fix-name-stacktrace
    - 8601e4b341 Don't stacktrace if "name" is specified as a minion id in a map file
- **ISSUE** #24020: (arthurlogilab) [salt-cloud saltify] cannot use –profile saltify machine{1..3} without a map (refs: #26790)
- **PR** #26790: (rallytime) Update Saltify docs to be more accurate and helpful @ *2015-08-31 18:17:31 UTC*
    - 7c8d0a09f6 Merge pull request #26790 from rallytime/saltify_docs
    - d53754f2b7 Update Saltify docs to be more accurate and helpful

- **ISSUE** #26773: (styro) salt-call minor breakage on Windows (refs: #26775)
- **PR** #26787: (jfindlay) merge #26775 @ *2015-08-31 17:52:45 UTC*
  - **PR** #26775: (styro) Fix some leftover non portable exitcodes. (refs: #26787)
  - 70d0268c83 Merge pull request #26787 from jfindlay/imp
  - e5bbf59ec7 disable import lint in run.py
  - 8aef725243 Restore blank lines again.
  - 1710070f61 Restore blank line.
  - 59d61a8dea os module no longer required.
  - f1b8d0d509 Add missing imports.
  - 7bd8809e23 Fix some non portable exitcodes. Fixes #26773
- **PR** #26759: (terminalmage) Backport PR #26726 to 2015.5 branch @ *2015-08-31 14:39:20 UTC*
  - **PR** #26726: (terminalmage) Redact HTTPS Basic Auth in states/funcs which deal with git remotes (refs: #26759)
  - 645998dbd3 Merge pull request #26759 from terminalmage/bp-26726
  - d7f7fca7e5 More cleanup from moving auth redaction to salt.utils.url
  - 07db5a7038 fix redaction
  - 399871e6dd Add auth redaction flags to git exec module and use them in git state
  - 776dc38d73 check for ValueError when adding http basic auth
  - d2eb1f4340 Rename arguments in salt.utils.url.add_http_basic_auth
  - b45f37a467 Add http basic auth tests
  - 1ed42ea4fd Remove git unit tests, moving them to salt.utils.url tests
  - 96a55cdb59 Remove unused imports
  - 1f25a859bd Redact HTTPS Basic Auth data from remote URLs in comments and changes dict
  - eafeb6c7bf Automatically redact HTTPS basic auth
  - 6be3f8f9e1 Add support for callbacks to influence what information about commands is logged
  - c36f240a87 Add HTTPS Basic Auth funcs to salt.utils.url
- **ISSUE** #26628: (MadsRC) state.ipset tries to parse wrong data (refs: #26768)
- **PR** #26768: (garethgreenaway) Fixes to ipset in 2015.5 for #26628 @ *2015-08-29 03:24:07 UTC*
  - 46a4bbd0e7 Merge pull request #26768 from garethgreenaway/26628_2015_5_ipset_fixes
  - f0c6090c7e Fixing issue when information returned from ipset isn't in the format we expect and it causes an exception.
- **ISSUE** #26732: (saltstack-bot) SmartOS pkgsrc dependency (refs: #26753)
- **PR** #26753: (jfindlay) import elementree from _compat in ilo exec mod @ *2015-08-28 20:56:45 UTC*
  - 7a58878ea8 Merge pull request #26753 from jfindlay/iloet
  - 211a02754f import elementree from _compat in ilo exec mod
- **ISSUE** #21256: (dhs-rec) win.exe package for RH 6 (refs: #26736)
- **PR** #26736: (twangboy) Changed import from smbconnection to smb3 @ *2015-08-28 17:23:42 UTC*

  - 22dbce8d61 Merge pull request #26736 from twangboy/fix_21256
  - 86f425c669 Changed import from smbconnection to smb3
- **ISSUE** #26705: (Galser) Salt-Master 2015.5.5-1 on Scientific Linux 6 fails loading some primitive pillars from YAML (refs: #26714)
- **PR** #26714: (jfindlay) add exception placeholder for older msgpacks @ *2015-08-28 16:02:35 UTC*
  - 16d4e0350d Merge pull request #26714 from jfindlay/pack_except
  - ebcfaf9050 add exception placeholder for older msgpacks
- **PR** #26710: (rallytime) Update GCE driver to return True, False or a new name in __virtual__() @ *2015-08-27 20:08:17 UTC*
  - 47faa8cc16 Merge pull request #26710 from rallytime/gce_virtual_return
  - e6b74879d7 Remove unused import
  - 78e31585cf Update GCE driver to return True, False or a new name in __virtual__()
- **ISSUE** #14612: (cachedout) Catch provider errors in salt cloud (refs: #26709)
- **PR** #26709: (rallytime) Ensure VM name is valid before trying to create Linode VM @ *2015-08-27 20:07:49 UTC*
  - cf487cf0f5 Merge pull request #26709 from rallytime/fix-14612
  - bc21094ea0 versionadded and more efficient checks
  - a3ac8e7008 Whitespace fix
  - 9a4228d906 Added unit tests for new _validate_name function and adjusted regex
  - 388815112c Ensure VM name is valid before trying to create Linode VM
- **ISSUE** #9592: (otrempe) pip module fails on Windows because of quoting (refs: #26617)
- **PR** #26617: (terminalmage) Fix Windows failures in pip module due to raw string formatting @ *2015-08-27 19:24:53 UTC*
  - c3a6280f8c Merge pull request #26617 from terminalmage/issue9592
  - 96c3df1ed5 Don't accept non-list input for pkgs arg
  - 419221535b Lint fix
  - ede057eebc Fix tests to reflect args being passed as lists instead of strings
  - 03250dbd9f Pass command to cmd.run_all as list instead of joining
  - 1c90cdb07e salt/modules/pip.py: Remove raw string format flags
  - cd35df5ff8 Catch TypeErrors in timed_subprocess
- **PR** #26700: (kev009) Ignore the first element of kern.disks split, which is the sysctl name @ *2015-08-27 17:48:02 UTC*
  - 24a4f54f39 Merge pull request #26700 from kev009/fbsd-disks-fix-2015.5
  - 3ac97f9de4 Ignore the first element of kern.disks split, which is the sysctl name
- **PR** #26695: (terminalmage) Better HTTPS basic auth redaction for 2015.5 branch @ *2015-08-27 15:10:38 UTC*
  - 58945131b5 Merge pull request #26695 from terminalmage/better-https-auth-redaction-2015.5
  - 752d260209 Use versioninfo tuple for comparison
  - b1d253483e Better HTTPS basic auth redaction for 2015.5 branch

- **PR** #26694: (terminalmage) Backport #26693 to 2015.5 @ *2015-08-27 08:16:30 UTC*
  - **PR** #26693: (serge-p) Update openbsdpkg.py (refs: #26694)
  - 4040a312f9 Merge pull request #26694 from terminalmage/bp-26693
  - 4aec926476 Update openbsdpkg.py
- **PR** #26681: (basepi) [2015.5] Merge forward from 2014.7 to 2015.5 @ *2015-08-26 22:03:07 UTC*
  - 0b17f80fe9 Merge pull request #26681 from basepi/merge-forward-2015.5
  - 64cad371f0 Remove overmocked test
  - 40718af1d5 Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.5
  - c2c7fe06c8 Merge pull request #26667 from nmadhok/doc-fix-2014.7
    - 26be189689 Doc fix. Fixes #26656
  - 6bd3dccae8 Merge pull request #26663 from jacobhammons/2014.7-version
    - b6af538070 version change for latest branch
  - 071a6112e5 Merge pull request #26636 from rallytime/cloud-test-fixes
    - c0d83d558d Don't use id as variable
    - 2b4bc1679d Keep ec2 instance creation test the same - it works better for the ec2 output
    - b5b58eb31f Skip digital ocean tests since we can't use API v1 with v2 tests
    - 9ae1539c62 Update cloud tests to be more efficient and accurate
  - 304542b4c6 Merge pull request #26640 from efficks/fixws2014
    - ebe5d9d85c Fix function spacing
- **PR** #26676: (rallytime) Back-port #26648 to 2015.5 @ *2015-08-26 19:46:01 UTC*
  - **PR** #26648: (whiteinge) Free 'fun' from the function signature namespace (refs: #26676)
  - 75675a6ba9 Merge pull request #26676 from rallytime/bp-26648
  - 1af42eed36 Free 'fun' from the function signature namespace
- **PR** #26677: (rallytime) Back-port #26653 to 2015.5 @ *2015-08-26 19:45:54 UTC*
  - **PR** #26653: (dmyerscough) You can provide a X-Auth-Token when requesting jobs (refs: #26677)
  - d7f682cb5b Merge pull request #26677 from rallytime/bp-26653
  - 497ca96039 You can provide a X-Auth-Token when requesting jobs
- **PR** #26675: (rallytime) Back-port #26631 to 2015.5 @ *2015-08-26 19:44:59 UTC*
  - **PR** #26631: (PierreR) Fix get_load in postgres returner (refs: #26675)
  - 960dbba7ed Merge pull request #26675 from rallytime/bp-26631
  - 20eecdc7be Fix get_load
- **PR** #26655: (damonzheng) Update win_dns_client.py @ *2015-08-26 16:05:26 UTC*
  - db30926ac9 Merge pull request #26655 from cheng0919/2015.5
  - fdebc01def Update win_dns_client.py
  - 1d23d5e797 Update win_dns_client.py
  - 1a45db0fb7 Update win_dns_client.py

- **PR** #26662: (jacobhammons) update version to 2015.5 @ *2015-08-26 13:45:44 UTC*
  - a04d243471 Merge pull request #26662 from jacobhammons/version
  - 4e5766fdde update version to 2015.5
- **PR** #26651: (jfindlay) add 2015.5.4 notes to 2015.5.5 notes @ *2015-08-26 00:25:28 UTC*
  - 8a9a076ad4 Merge pull request #26651 from jfindlay/2015.5
  - dc5cee5f8f add 2015.5.4 notes to 2015.5.5 notes
- **ISSUE** #26497: (JensRantil) Feature request: Make salt.states.managed support local file *source* (refs: #26525)
- **PR** #26525: (jfindlay) document check_file_meta args, remove unused arg @ *2015-08-25 21:43:46 UTC*
  - 5bdefdc234 Merge pull request #26525 from jfindlay/sum
  - 0297d49aa0 remove unused check_file_meta arg
  - 6a3cb1c0aa document args to file.check_file_meta exec fcn
- **PR** #26561: (stanislavb) Leave salt.utils.s3 location fallback to salt.utils.aws (refs: #27070) @ *2015-08-25 21:40:30 UTC*
  - **PR** #26446: (stanislavb) Fetch AWS region from EC2 instance metadata (refs: #26561)
  - **PR** #26378: (stanislavb) Fix EC2 credentials from IAM roles for s3fs and s3 ext_pillar in 2015.5 (refs: #26446)
  - 84e96458b3 Merge pull request #26561 from stanislavb/2015.5
  - 50332895a1 Leave salt.utils.s3 location fallback to salt.utils.aws
- **ISSUE** #22550: (amendlik) Error deleting SSH keys using salt-cloud –destroy (refs: #26573)
- **PR** #26573: (rallytime) Don't stacktrace if using private_ips and delete_sshkeys together @ *2015-08-25 20:00:23 UTC*
  - 1d729734cc Merge pull request #26573 from rallytime/destroy_ssh_keys_private_ips
  - 4267509c25 Don't stacktrace if using private_ips and delete_sshkeys
- **ISSUE** #20169: (flavianh) [salt-cloud] Add a meaningful error when /etc/salt/cloud is missing the master's address (refs: #26563)
- **PR** #26563: (rallytime) Fix error detection when salt-cloud config is missing a master's address @ *2015-08-25 20:00:11 UTC*
  - 000e5a2acf Merge pull request #26563 from rallytime/fix-20169
  - 65b285d02d Only warn if master IP is unset - must be compatible with masterless minions
  - a4c87fcf57 Simplify logic
  - 593ead08cf Fix error detection when salt-cloud config is missing a master's address
- **PR** #26641: (basepi) [2015.5] Merge forward from 2014.7 to 2015.5 @ *2015-08-25 18:17:46 UTC*
  - 19c7a6d575 Merge pull request #26641 from basepi/merge-forward-2015.5
  - a5dafa436c Already fixed on 2015.5
  - 71c0898fb5 Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.5
  - 4532f98a76 Merge pull request #26515 from bersace/salt-env-local-sls
  - 0727af9e3d Defaults to current saltenv in state.sls
- **PR** #26620: (rallytime) Also add -Z to script args for cloud tests @ *2015-08-24 22:03:24 UTC*

  - 2927859c8a Merge pull request #26620 from rallytime/more_script_args
  - 9ae27193d8 Also add -Z to script args for cloud tests
- **PR** #26618: (rallytime) Add script_args: '-P' to Ubuntu 14 profiles for nightly cloud tests @ *2015-08-24 21:15:24 UTC*
  - ed166ebd4f Merge pull request #26618 from rallytime/pip-undate-cloud-tests
  - 5a2c8825ba Extra lines
  - d28672b69e Add script_args: '-P' to Ubuntu 14 profiles for nightly cloud tests
- **PR** #26612: (rallytime) Use an available image to test against @ *2015-08-24 19:09:18 UTC*
  - 6d3927bed5 Merge pull request #26612 from rallytime/fix-do-list-images-test
  - 1401255287 Use an available image to test against
- **ISSUE** #15590: (jtratner) salt-cloud gce configuration check incorrect (refs: #26576)
- **PR** #26576: (rallytime) Ensure GCE and EC2 configuration checks are correct @ *2015-08-23 18:59:46 UTC*
  - 991bbf63fe Merge pull request #26576 from rallytime/fix-14604
  - ac67a1d238 Ensure GCE configuration check is correct
  - 421f1fde1e Ensure EC2 configuration check is correct
- **ISSUE** #12225: (arthurlogilab) [salt-cloud] Attribution of floating IPs works partially in parallel mode (refs: #26580)
- **PR** #26580: (rallytime) Avoid race condition when assigning floating IPs to new VMs @ *2015-08-23 18:58:48 UTC*
  - 746c0008a9 Merge pull request #26580 from rallytime/fix-12225
  - e3f7db17cc Avoid race condition when assigning floating IPs to new VMs
  - afda31be74 Create _assign_floating_ips function for DRY
- **PR** #26581: (terminalmage) Skip tests that don't work with older mock @ *2015-08-22 23:06:27 UTC*
  - 965a4ba7cf Merge pull request #26581 from terminalmage/fix-tests
  - 49d8bd1dbe Remove unused import
  - 81a0d4c915 Skip tests that don't work with older mock
- **ISSUE** #25478: (zyio) salt-ssh - Unable to locate current thin version (refs: #25862)
- **ISSUE** #25026: (sylvia-wang) salt-ssh "Failure deploying thin" when using salt module functions (refs: #25862)
- **PR** #26591: (rallytime) Back-port #26554 to 2015.5 @ *2015-08-22 21:19:02 UTC*
  - **PR** #26554: (tjstansell) /bin/sh is more portable than /bin/bash (refs: #26591)
  - **PR** #25862: (zyio) Adding SCP_NOT_FOUND exit code (refs: #26554)
  - 19992c1450 Merge pull request #26591 from rallytime/bp-26554
  - 6f8bed88cb /bin/sh is more portable than /bin/bash
- **PR** #26565: (cachedout) Fix many errors with __virtual__ in tests @ *2015-08-21 21:37:54 UTC*
  - 2cd36c7ed4 Merge pull request #26565 from cachedout/fix_virtual_warnings
  - 41541e4e2b Fix many errors with __virtual__ in tests
- **ISSUE** #19249: (ahetmanski) Cannot create cache_dir salt master exception. (refs: #26548)

- **PR** #26553: (rallytime) Back-port #26548 to 2015.5 @ *2015-08-21 17:40:21 UTC*
  - **PR** #26548: (vakulich) Catch OSError during cache directories creation, fixes #19249 (refs: #26553)
  - 5a32664efd Merge pull request #26553 from rallytime/bp-26548
  - ec2b2c3e40 Catch OSError during cache directories creation, fixes #19249
- **PR** #26552: (rallytime) Back-port #26542 to 2015.5 @ *2015-08-21 17:40:11 UTC*
  - **PR** #26542: (arthurlogilab) [doc] reactor documentation fix : returners (refs: #26552)
  - 7e67e48656 Merge pull request #26552 from rallytime/bp-26542
  - 0976b1e23b [doc] reactor documentation fix : returners
- **PR** #26551: (rallytime) Back-port #26539 to 2015.5 @ *2015-08-21 17:39:22 UTC*
  - **PR** #26539: (carlpett) Doc-fix: Escape backslash in domain\\username (refs: #26551)
  - bcd462545d Merge pull request #26551 from rallytime/bp-26539
  - 94ff4cff40 Doc-fix: Escape backslash in domainusername
- **PR** #26549: (rallytime) Back-port #26524 to 2015.5 @ *2015-08-21 17:38:50 UTC*
  - **PR** #26524: (JensRantil) Gracefully handle package comparison not in (-1, 0, 1) (refs: #26549)
  - **PR** #25369: (anlutro) Fix aptpkg.version_cmp (refs: #26524)
  - 4dbf61c5af Merge pull request #26549 from rallytime/bp-26524
  - 4763f28725 logging(cmp_version): output assertion
  - 673b6c683d utils(version_cmp): handle comparison not in (0,1,-1)
- **ISSUE** #26502: (ryan-lane) Adding a listener with None as ports doesn't result in an invocation error in boto_elb (refs: #26527)
- **PR** #26527: (jfindlay) check exists and values in boto_elb listeners @ *2015-08-21 15:27:52 UTC*
  - 1ac8287588 Merge pull request #26527 from jfindlay/elb
  - 343e47f00c check exists and values in boto_elb listeners
- **PR** #26446: (stanislavb) Fetch AWS region from EC2 instance metadata (refs: #26561) @ *2015-08-21 15:11:08 UTC*
  - **PR** #26378: (stanislavb) Fix EC2 credentials from IAM roles for s3fs and s3 ext_pillar in 2015.5 (refs: #26446)
  - e4b2534aa8 Merge pull request #26446 from stanislavb/2015.5-ec2-metadata-region
  - 57943ff4f7 Fetch AWS region from EC2 instance metadata
- **PR** #26546: (nmadhok) Do not raise KeyError when calling avail_images if VM/template is in disconnected state @ *2015-08-21 14:17:49 UTC*
  - d721b7b2be Merge pull request #26546 from nmadhok/vmware-key-error-patch-2015.5
  - 1dcf157256 Do not raise KeyError when calling avail_images if VM/template is in disconnected state
- **ISSUE** #25360: (BretFisher) file.replace removes line feed if using YAML's multiline string syntax (refs: #26481)
- **PR** #26537: (jfindlay) Merge #26481 @ *2015-08-21 05:37:24 UTC*
  - **PR** #26481: (TheBigBear) minor note: added (refs: #26537)
  - 7da87fabf1 Merge pull request #26537 from jfindlay/note

  - 662e723ae0 fixup note lint in file.replace state mod
  - 332535f2e6 Update file.py
  - 598500034f Update file.py
  - ec7c7d738d minor note: added
- **PR** #26528: (zmalone) Fixing encrypt to instructions in the 2015.5 branch @ *2015-08-20 21:49:06 UTC*
  - c6d8e34730 Merge pull request #26528 from zmalone/2015.5
  - 39b111c465 Fixing encrypt to instructions in the 2015.5 branch, –homedir is not necessary here.

28.2.74 Salt 2015.5.7 Release Notes

release 2015-10-13

Version 2015.5.7 is a bugfix release for *2015.5.0*.

Statistics

- Total Merges: **103**
- Total Issue References: **66**
- Total PR References: **135**
- Contributors: **46** (0xf10e, JaseFace, MasterNayru, MrCitron, Sacro, ajacoutot, arthurlogilab, basepi, belvedere-trading, beverlcl, blast-hardcheese, blueyed, bogdanr, cachedout, cbuechler, chrigl, dmyerscough, eguven, eliasp, erchn, eyj, garethgreenaway, gashev, gnubyexample, gracinet, gravyboat, gwaters, hedinfaok, iggy, jacksontj, jacobhammons, jfindlay, lorengordon, mbologna, msciciel, nmadhok, pass-by-value, plastikos, rallytime, rominf, s0undt3ch, silenius, sjmh, stephen144, terminalmage, twangboy)

Important: A significant orchestrate issue (issue ##29110) was discovered during the release process of 2015.5.7, so it has not been officially released. Please use *2015.5.8* instead.

Changelog for v2015.5.6..v2015.5.7

Generated at: 2018-05-27 22:16:54 UTC

- **PR** #28864: (jfindlay) add 2015.5.7 release notes @ *2015-11-13 17:15:00 UTC*
  - ec7fdc539b Merge pull request #28864 from jfindlay/2015.5
  - 648b697951 add 2015.5.7 release notes
- **ISSUE** #27392: (ahammond) schedule running state.orchestrate fails (refs: #28731)
- **PR** #28731: (garethgreenaway) Fixes to salt scheduler in 2015.5, ensuring that return_job is only used on minion scheduler @ *2015-11-13 16:58:06 UTC*
  - bed45f4208 Merge pull request #28731 from garethgreenaway/27392_2015_5_scheduler_return_job_master
  - 771e9f7b6f Fixing the salt scheduler so that it only attempts to return the job data to the master if the scheduled job is running from a minion's scheduler.
- **PR** #28857: (rallytime) Back-port #28851 to 2015.5 @ *2015-11-13 13:56:53 UTC*
  - **PR** #28851: (rominf) [states/schedule] docstring: args, kwargs -> job_args, job_kwargs (refs: #28857)

  - 06f4932876 Merge pull request #28857 from rallytime/bp-28851
  - aa4b193f87 [states/schedule] docstring: args, kwargs -> job_args, job_kwargs
- **PR** #28856: (rallytime) Back-port #28853 to 2015.5 @ *2015-11-13 13:46:10 UTC*
  - **PR** #28853: (eliasp) Typo (with → which) (refs: #28856)
  - 0934a52b34 Merge pull request #28856 from rallytime/bp-28853
  - 37eeab2683 Typo (with → which)
- **ISSUE** #28828: (basepi) salt-ssh doesn't package tornado's new deps in the thin (refs: #28826)
- **PR** #28832: (basepi) [2015.5] Backport #28826 @ *2015-11-12 19:32:03 UTC*
  - **PR** #28826: (basepi) [2015.8] Add new tornado deps to salt-ssh thin (refs: #28832)
  - eb904665dc Merge pull request #28832 from basepi/backport.28826
  - 57be72eb91 Add backports_abc and singledispatch_helpers to thin as well
  - 897cad627b Add singledispatch to the thin
- **ISSUE** #8647: (Mrten) salt '*' highstate returns 'minion did not return', salt [minion] highstate works (refs: #28833)
- **PR** #28833: (basepi) [2015.5] Increase the default gather_job_timeout @ *2015-11-12 19:31:58 UTC*
  - eff811a0ad Merge pull request #28833 from basepi/increase.gather_job_timeout.8647
  - c09243dd01 Increase the default gather_job_timeout
- **PR** #28829: (basepi) [2015.5] Merge forward from 2014.7 to 2015.5 @ *2015-11-12 18:50:51 UTC*
  - e4a036365d Merge pull request #28829 from basepi/merge-forward-2015.5
  - f8b8441485 Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.5
  - 76e69b4bff Merge pull request #28777 from rallytime/bp-28740-2014.7
    - da5fac2b36 Back-port #28740 to 2014.7
  - 45c73ebf2f Merge pull request #28716 from rallytime/bp-28705
    - 32e7bd3ea0 Account for new headers class in tornado 4.3
  - f4fe921965 Merge pull request #28717 from cachedout/umask_note
    - 1874300e08 Add note about recommended umask
- **ISSUE** #25775: (trimbleagvendoraccounta) blockdev.formatted formats but fails. Second highstate shows success. (refs: #28756)
- **ISSUE** #20235: (joejulian) blockdev.format state can fail even if it succeeds (refs: #28756)
- **PR** #28756: (MrCitron) Fix #25775 @ *2015-11-12 17:47:51 UTC*
  - 93562631aa Merge pull request #28756 from MrCitron/fix-25775
  - 82075c809c Add logs and correct pylint error
  - e31e22d96a Fix 25775
- **ISSUE** #28783: (chrigl) iptables.get_saved_rules does not handle family=ipv6 (refs: #28786)
- **PR** #28786: (chrigl) closes #28783 @ *2015-11-11 21:01:19 UTC*
  - 30cc48e37f Merge pull request #28786 from chrigl/fix-28783
  - ba6d814553 closes #28783

- **PR** #28776: (rallytime) Back-port #28740 to 2015.5 @ *2015-11-11 18:02:03 UTC*
    - **PR** #28740: (MasterNayru) Add missing S3 module import (refs: #28776, #28777)
    - 8f1d0b636e Merge pull request #28776 from rallytime/bp-28740-2015.5
    - 49256b7d90 Back-port #28740 to 2015.5
- **ISSUE** #28732: (dmyerscough) cherrypy API endpoint (refs: #28760)
- **ISSUE** #22452: (whiteinge) rest_cherrypy /keys URL returns empty keys for minion IDs that already exist (refs: #28760)
- **ISSUE** #22451: (whiteinge) rest_cherrypy /keys URL throws a 500 on the first request (refs: #28760)
- **ISSUE** #22442: (allanliu) rest_cherrypy /keys URL does not handle JSON requests (refs: #28760)
- **PR** #28760: (dmyerscough) Fixing CherryPy key bug @ *2015-11-11 15:11:18 UTC*
    - 77d4b980f1 Merge pull request #28760 from dmyerscough/28732-Fix-cherrypi-api-keys-endpoint
    - 206d1684b2 Fixing CherryPy key bug
- **ISSUE** #28714: (gravyboat) Salt-api doesn't work with post unless data is included. (refs: #28718)
- **PR** #28746: (rallytime) Back-port #28718 to 2015.5 @ *2015-11-10 18:16:40 UTC*
    - **PR** #28718: (sjmh) Account for no POST data (refs: #28746)
    - 6f8f04975f Merge pull request #28746 from rallytime/bp-28718
    - 092f441cad Account for no POST data
- **PR** #28705: (cachedout) Account for new headers class in tornado 4.3 (refs: #28716) @ *2015-11-09 19:24:34 UTC*
    - f40c617bad Merge pull request #28705 from cachedout/tornado_http_headers
    - 7ac6cde1ee Account for new headers class in tornado 4.3
- **PR** #28699: (rallytime) Back-port #28670 to 2015.5 @ *2015-11-09 18:10:58 UTC*
    - **PR** #28670: (plastikos) psutil can fail to look-up a uid and raise a KeyError (refs: #28699)
    - 604a7b4199 Merge pull request #28699 from rallytime/bp-28670
    - e436b23296 psutil can fail to look-up a uid and raise a KeyError
- **PR** #28703: (rallytime) Back-port #28690 to 2015.5 @ *2015-11-09 18:01:57 UTC*
    - **PR** #28690: (MrCitron) Fix 28689 : Check s3 ext pillar cache file before calculating expiration (refs: #28703)
    - 7bd3eb8370 Merge pull request #28703 from rallytime/bp-28690
    - a0988dab58 Fix 28689 : Check s3 ext pillar cache file before calculating expiration
    - **PR** saltstack/salt-bootstrap#868: (cachedout) Always refresh the Arch Linux keyring if needed (refs: #28694)
- **PR** #28694: (s0undt3ch) [2015.5] Update to latest bootstrap script v2015.11.09 @ *2015-11-09 17:49:53 UTC*
    - 2a40f57b93 Merge pull request #28694 from s0undt3ch/2015.5
    - 0910c6ffe4 Update to latest bootstrap script v2015.11.09
- **ISSUE** #26592: (centromere) rabbitmq.list_vhosts removes final line from rabbitmqctl output (refs: #28669)
- **PR** #28669: (rallytime) Use the -q argument to strip extraneous messages from rabbitmq @ *2015-11-08 01:07:25 UTC*

  - 3249b322e8 Merge pull request #28669 from rallytime/fix-26592
  - 098fb815af Use the -q argument to strip extraneous messages from rabbitmq
- **ISSUE** #28577: (jacksontj) Increase in master CPU usage after upgrading to 2015.8 (refs: #28645)
- **PR** #28645: (jacksontj) Rework minion return_retry_timer @ *2015-11-07 03:40:28 UTC*
  - **PR** #27286: (terminalmage) Add a configurable timer for minion return retries (refs: #28645)
  - 29e8250d0c Merge pull request #28645 from jacksontj/2015.5
  - f63c2d70a7 Rework minion return_retry_timer
- **ISSUE** #15177: (baskinomics) system.join_domain() does not join domain on Windows Server 2012 R2 (refs: #28668)
- **PR** #28668: (twangboy) Fixed join_domain and unjoin_domain for Windows @ *2015-11-07 03:40:04 UTC*
  - 1bbaea8aad Merge pull request #28668 from twangboy/fix_15177
  - 745b8f75f6 Fixed some lint
  - a43eb53f28 Added version added notes in docs
  - 6b537c8640 Fixed join_domain and unjoin_domain for Windows
- **ISSUE** #8051: (regilero) Problems with fileinput.input inplace editing in salt.states.file.replace (refs: #28174)
- **ISSUE** #7999: (regilero) MULTILINE pattern cannot work in file.replace, fileinput always reads line by line. (refs: #28174)
- **PR** #28666: (jfindlay) define r_data before using it in file module @ *2015-11-07 00:46:27 UTC*
  - **PR** #28174: (lorengordon) Add support for multiline regex in file.replace (refs: #28666)
  - 4ad5056066 Merge pull request #28666 from jfindlay/r_data
  - 29228f445f define r_data before using it in file module
- **ISSUE** #24758: (zerthimon) salt-minion uses 100% CPU for periodic status.master task on a server with a lot of TCP connections (a LB). (refs: #28662)
- **PR** #28662: (cachedout) Add note about disabling master_alive_interval @ *2015-11-07 00:38:12 UTC*
  - e129e889ad Merge pull request #28662 from cachedout/issue_24758
  - 78f4894333 Add note about disabling master_alive_interval
- **PR** #28627: (twangboy) Backport win_useradd @ *2015-11-06 16:57:49 UTC*
  - df121d0cec Merge pull request #28627 from twangboy/backport_win_useradd
  - 87282b6354 Backport win_useradd
- **ISSUE** #28398: (L4rS6) Permissions /var/cache/salt/minion/extmods (refs: #28617)
- **PR** #28617: (cachedout) Set restrictive umask on module sync @ *2015-11-05 23:43:28 UTC*
  - 64a20228c6 Merge pull request #28617 from cachedout/umask_module_sync
  - 227792e158 Set restrictive umask on module sync
- **ISSUE** #28621: (gravyboat) Puppet module documentation should be less insulting (refs: #28622)
- **PR** #28622: (gravyboat) Update puppet module wording @ *2015-11-05 20:34:07 UTC*
  - 065f8c7fb3 Merge pull request #28622 from gravyboat/update_puppet_module_docs
  - 4ea28bed30 Update puppet module wording

- **ISSUE** #655: (thatch45) Add general command management to service (refs: #‘saltstack/salt-bootstrap#656‘_)
  - **PR** saltstack/salt-bootstrap#674: (jfindlay) add support for repo.saltstack.com (refs: #28563)
  - **PR** saltstack/salt-bootstrap#665: (mbologna) Change to 'dnf' as package manager for Fedora 22-> (refs: #28563)
  - **PR** saltstack/salt-bootstrap#656: (eyj) Add bootstrap -b flag (don't install dependencies) (refs: #28563)
  - **PR** saltstack/salt-bootstrap#654: (hedinfaok) Fixes error finding python-jinja2 in RHEL 7 (refs: #28563)
  - **PR** saltstack/salt-bootstrap#653: (cbuechler) Make bootstrap work with FreeBSD 11-CURRENT. (refs: #28563)
- **PR** #28563: (s0undt3ch) [2015.5] Update to latest bootstrap script v2015.11.04 @ *2015-11-04 15:16:31 UTC*
  - 08295de5a5 Merge pull request #28563 from s0undt3ch/2015.5
  - 16f4db79a0 Update to latest bootstrap script v2015.11.04
- **ISSUE** #28173: (twangboy) system.computer_name does not work in windows (refs: #28541)
- **PR** #28541: (twangboy) Fixed problem with system.set_computer_name @ *2015-11-04 14:48:54 UTC*
  - 1e09f186ce Merge pull request #28541 from twangboy/fix_28173
  - 7edf5ce370 Fixed problem with system.set_computer_name
- **ISSUE** #28524: (bmcorser) UnicodeDecodeError in states.file (refs: #28538, #28537)
  - **PR** #28538: (jfindlay) decode path and url to utf-8 in url.create (refs: #28537)
- **PR** #28537: (jfindlay) decode filename to utf-8 in file.recurse state @ *2015-11-04 14:48:18 UTC*
  - f44ed780b5 Merge pull request #28537 from jfindlay/decode_state_2015.5
  - 06e514940c decode filename to utf-8 in file.recurse state
- **ISSUE** #28272: (gravyboat) Update documentation contributing docs to explain how to PR against different releases (refs: #28529)
- **PR** #28529: (rallytime) Update contributing and documentation pages to recommend submitting against branches @ *2015-11-04 14:47:21 UTC*
  - 6acf87593f Merge pull request #28529 from rallytime/fix-28272
  - a959681858 Add link to Sending a GH PR to documentation docs
  - 1c612e2772 Update contributing and documentation pages to recommend submitting against branches
- **ISSUE** #28511: (nghgd) vmware clone task fails instead of waiting to completion (refs: #28546)
- **PR** #28548: (nmadhok) [Backport] [2015.5] Tasks can be in queued state instead of running @ *2015-11-04 04:14:25 UTC*
  - **PR** #28546: (nmadhok) Tasks can be in queued state instead of running. (refs: #28548)
  - 025bff2bf0 Merge pull request #28548 from nmadhok/2015.5-task-error
  - 804a0a6537 Tasks can be in queued state instead of running. Fixes #28511
- **ISSUE** #24585: (utahcon) No version data for SALT.STATES.VIRTUALENV in wiki (refs: #28531)
- **PR** #28531: (rallytime) Add versionadded directives to virtualenv_mod state/module @ *2015-11-03 21:34:49 UTC*
  - 63bd3e52b3 Merge pull request #28531 from rallytime/fix-24585
  - bc577b2531 Add versionadded directives to virtualenv_mod state/module

- **PR** #28508: (twangboy) Fixed windows tests @ *2015-11-03 19:31:12 UTC*
    - ea3bf972c4 Merge pull request #28508 from twangboy/fix_unit_tests_windows
    - 0da6ff7c50 Fixed some logic
    - cf1e059be5 Fixed windows tests
- **PR** #28525: (rallytime) Fix spacing in doc examples for boto_route53 state and module @ *2015-11-03 19:30:24 UTC*
    - 73c5735fc1 Merge pull request #28525 from rallytime/route53_spacing
    - 6ab2ce615c Fix spacing in doc examples for boto_route53 state and module
- **ISSUE** #28243: (guettli) Docs: default value of state_auto_order ? (refs: #28517)
- **PR** #28517: (rallytime) Add state_auto_order defaults to True note to ordering docs @ *2015-11-03 14:04:40 UTC*
    - 2d7f934f67 Merge pull request #28517 from rallytime/fix-28243
    - be8f650901 Punctuation.
    - fd846822c1 Add state_auto_order defaults to True note to ordering docs
- **PR** #28512: (basepi) [2015.5] Merge forward from 2014.7 to 2015.5 @ *2015-11-03 00:38:08 UTC*
    - 63ce8f78d5 Merge pull request #28512 from basepi/merge-forward-2015.5
    - 61c382133a Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.5
    - 4bf56cad3f Merge pull request #28461 from cachedout/issue_28455
        - 097838ec0c Wrap all cache calls in state.sls in correct umask
    - f3e61db045 Merge pull request #28407 from DSRCompany/issues/24910_token_auth_fix_2014
        - b7b5bec309 Don't request creds if auth with key.
- **PR** #28448: (gwaters) added a note to the tutorial for redhat derivatives @ *2015-10-30 18:49:53 UTC*
    - 37ceae1e88 Merge pull request #28448 from gwaters/add-redhat-notes
    - e70990704a added a note to the tutorial for those that redhat so they can use the state file too.
- **PR** #28406: (rallytime) Back-port #28381 to 2015.5 @ *2015-10-29 19:10:37 UTC*
    - **PR** #28381: (JaseFace) Add FreeBSD detection for VirtualBox (refs: #28406)
    - 5ef50d60cd Merge pull request #28406 from rallytime/bp-28381
    - e5322d2c44 Add FreeBSD detection for VirtualBox
- **PR** #28413: (rallytime) Back-port #28400 to 2015.5 @ *2015-10-29 18:06:46 UTC*
    - **PR** #28400: (msciciel) State pkg.installed: do not execute _preflight_check if not_installed list is empty in _find_install_targets (refs: #28413)
    - 30d5f7bbae Merge pull request #28413 from rallytime/bp-28400
    - ae1921b922 Do not execute _preflight_check if not_installed list is empty in _find_install_targets. Calling with empty list on rhel/centos cause execution of repoquery –whatprovides without pkg list which is memory consumptive task for host and also for red hat satellite server.
- **PR** #28366: (erchn) mark repo not enabled when pkgrepo state passes in disable: True @ *2015-10-29 15:55:54 UTC*
    - 045d540aff Merge pull request #28366 from erchn/fix_yumpkg_mod_repo_disabled
    - 8187a4ce20 re-arrange things a bit to have less overall changes

  - f1d570ff18 move todelete above disabled check, add comment
  - 64feec413f also remove disabled key from repo_opts
  - 2f2ebb7bb6 mark repo not enabled when pkgrepo state passes in disable: True
- **ISSUE** #28372: (beverlcl) use_carrier option for bonding network interfaces are setting invalid values (refs: #28373)
- **PR** #28373: (beverlcl) Fixing bug #28372 for use_carrier option on bonding network interfaces. @ *2015-10-29 14:45:57 UTC*
  - 3923f4a569 Merge pull request #28373 from beverlcl/fix-use_carrier-28372
  - 32cffeceb6 Fixing bug #28372 for use_carrier option on bonding network interfaces.
- **PR** #28359: (rallytime) Back-port #28358 to 2015.5 @ *2015-10-28 20:43:05 UTC*
  - **PR** #28358: (arthurlogilab) docstring typo fix - list returners not runners (refs: #28359)
  - e07e3f257b Merge pull request #28359 from rallytime/bp-28358
  - 9cacbf582b docstring typo fix - list returners not runners
- **ISSUE** #28000: (hrumph) No option to stop windows minion installer from starting service in silent mode. (refs: #28346)
- **ISSUE** #27923: (twangboy) Salt Windows Installer fails to grab existing config (refs: #28346)
- **PR** #28346: (twangboy) Fix installer @ *2015-10-28 14:21:34 UTC*
  - 282be7ba5a Merge pull request #28346 from twangboy/fix_installer
  - f65e3e5275 Updated documentation to reflect the new parameter
  - a0c5223554 Fixes #27923 and #28000
- **PR** #28315: (gwaters) Adding a working example of setting pillar data on the cli @ *2015-10-27 15:27:49 UTC*
  - 7858f04ebc Merge pull request #28315 from gwaters/update-pillar-doc
  - b15285c0b4 adding a working example of setting pillar data on the cli
- **ISSUE** #28209: (basepi) Legacy git_pillar configs cause duplicate ext_pillar calls (refs: #28210)
- **PR** #28211: (terminalmage) Fix for ext_pillar being compiled twice in legacy git_pillar code (2015.5 branch) @ *2015-10-26 14:14:02 UTC*
  - **PR** #28210: (terminalmage) Fix for ext_pillar being compiled twice in legacy git_pillar code (refs: #28211)
  - 45305ccf29 Merge pull request #28211 from terminalmage/legacy_git_pillar-2015.5
  - 0d6a4ac115 Remove non-functional test
  - ab991d61d9 Fix for ext_pillar being compiled twice in legacy git_pillar code (2015.5 branch)
- **ISSUE** #26411: (whiteinge) salt-call cannot send custom events without Minion daemon running (refs: #28263)
- **PR** #28263: (cachedout) New channel for event.send @ *2015-10-26 14:07:06 UTC*
  - a6cc84c407 Merge pull request #28263 from cachedout/issue_26411-1
  - 3b880a5f07 New channel for event.fire_master
  - 29e9533aab Stand up a new channel if using salt-call
- **PR** #28293: (cachedout) Minor grammar changes @ *2015-10-26 12:15:42 UTC*
  - **PR** #28271: (gwaters) Update tutorial documentation (refs: #28293)

  - 788e1463d8 Merge pull request #28293 from cachedout/fix_28271
  - 499ed8519b Minor grammar changes to #28271
- **PR** #28271: (gwaters) Update tutorial documentation (refs: #28293) @ *2015-10-26 12:12:37 UTC*
  - e178af0b90 Merge pull request #28271 from gwaters/update-tutorial-documentation
  - f96d39483d updated the tutorial with gravyboat's suggestions
  - b1f4a2bdf4 i think i changed the wrong header, updated to fix
  - 846b3aece1 I found you can not run the cp.push commands until after enabling the feature in the conf, so I wanted to update the docs so others who try these commands wont bump into the same issue I had.
- **ISSUE** #28248: (0xf10e) conventions/formula.rst: "Gather external data" suggests unavailable jinja functionality (refs: #28280)
- **PR** #28280: (0xf10e) Correct Jinja function load_* to import_* @ *2015-10-25 04:11:10 UTC*
  - e3eff9b909 Merge pull request #28280 from 0xf10e/patch-1
  - 6d4316b0ac Correct Jinja function load_* to import_*
- **PR** #28255: (cachedout) Add __cli opt @ *2015-10-23 18:44:30 UTC*
  - 909fa3dc97 Merge pull request #28255 from cachedout/cli_opt
  - a2408157de Add __cli opt
- **ISSUE** #27374: (mool) boto_route53 state doesn't create a record (refs: #28213)
- **PR** #28213: (rallytime) If record returned None, don't continue with the state. Something went wrong @ *2015-10-23 13:54:50 UTC*
  - 0fa094ae11 Merge pull request #28213 from rallytime/boto_route53_state
  - 237d64ff11 If record returned None, don't continue with the state. Something went wrong.
- **ISSUE** #28217: (Ch3LL) Scheduler.present tries to add the scheudler each time (refs: #28238)
- **PR** #28238: (basepi) [2015.5] Fix schedule.present always diffing @ *2015-10-23 13:31:32 UTC*
  - 1768014705 Merge pull request #28238 from basepi/fix.schedule.present.28217
  - 087a8dc3c2 Only insert enabled if it's a dict
  - 5b49f41fab Fix schedule comparison to adjust for 'enabled' being added in schedule.list
  - 2dc1226ab8 Build new item with 'enabled' if available
- **ISSUE** #8051: (regilero) Problems with fileinput.input inplace editing in salt.states.file.replace (refs: #28174)
- **ISSUE** #7999: (regilero) MULTILINE pattern cannot work in file.replace, fileinput always reads line by line. (refs: #28174)
- **PR** #28174: (lorengordon) Add support for multiline regex in file.replace (refs: #28666) @ *2015-10-22 14:02:43 UTC*
  - bdd48c92de Merge pull request #28174 from lorengordon/file-replace-multiline
  - acdef2da60 Update docstrings with new guidance
  - 0835b005b7 Use a test that makes the extra file read unnecessary
  - 6d6121a6e5 Use *flags* when checking whether content was added previously
  - b25e609e9e Set *flags=8* since now the file is read as a MULTILINE string by default
  - 89e8dcdffd Use *finally* block to ensure mmap object is closed

  - 5aea6647c9 Add support for multiline regex in file.replace
- **ISSUE** #19673: (holyzhou) partition.mkpart in parted modules doesn't work (refs: #28175)
- **PR** #28175: (twangboy) Fixes #19673 @ *2015-10-21 20:48:24 UTC*
  - 2225925fb5 Merge pull request #28175 from twangboy/fix_19673
  - ae8fbb208f Fixes #19673
- **PR** #28140: (rallytime) Add OpenBSD installation documentation to 2015.5 branch @ *2015-10-20 16:31:34 UTC*
  - **PR** #28103: (ajacoutot) OpenBSD salt package: update list of dependencies. (refs: #28140)
  - ab18dcf637 Merge pull request #28140 from rallytime/bsd-installation-doc
  - 458a544d83 Add OpenBSD installation documentation to 2015.5 branch
- **ISSUE** #28101: (bogdanr) salt-cloud ec2 list-sizes doesn't show all available sizes (refs: #28138)
- **PR** #28138: (rallytime) Back-port #28130 EC2 Sizes Only portion to 2015.5 @ *2015-10-20 16:29:09 UTC*
  - **PR** #28130: (bogdanr) Ec2 upload public key and updated instances size list (refs: #28138)
  - fad38eb3c3 Merge pull request #28138 from rallytime/bp-28130-sizes-only
  - 6ab31e1886 Pylint
  - 37e4ed58a9 Added missing comma
  - 667f5e669f Added a bunch of instance sizes and updated some outdated ones
- **ISSUE** #26844: (double-yaya) The function "state.sls" is running as PID XXXX and was started at …. with jid XXXX always shows the current jid (refs: #28097)
- **PR** #28097: (jacksontj) For all multi-part messages, check the headers. If the header is not … @ *2015-10-20 15:00:18 UTC*
  - ce8f858536 Merge pull request #28097 from jacksontj/2015.5
  - 75e04bcbbc For all multi-part messages, check the headers. If the header is not your minion_id, skip the message
- **ISSUE** #23655: (arthurlogilab) salt-cloud with lxc should not traceback when minion is unreacheable (refs: #28117)
- **PR** #28117: (rallytime) Clean up stacktrace when master can't be reached in lxc cloud driver @ *2015-10-20 12:41:12 UTC*
  - 9cdb970289 Merge pull request #28117 from rallytime/fix-23655
  - dfb908e405 Clean up stacktrace when master can't be reached in lxc cloud driver
- **PR** #28110: (terminalmage) Add explanation of file_client: local setting masterless mode @ *2015-10-20 12:28:05 UTC*
  - bf7ed0a397 Merge pull request #28110 from terminalmage/masterless-mode
  - ed90103124 Add explanation of file_client: local setting masterless mode
- **ISSUE** #27940: (multani) salt-cloud creating lxc containers doesn't fire "salt/cloud/*/created" event (refs: #28109)
- **PR** #28109: (rallytime) Add created reactor event to lxc cloud driver @ *2015-10-19 20:32:41 UTC*
  - a569ef4980 Merge pull request #28109 from rallytime/fix-27940
  - 18b2245611 Add created reactor event to lxc cloud driver

- **ISSUE** #21845: (kitsemets) pip.install: fails in v2015.2.0rc1 when the package is already installed (pip v1.0) (refs: #27996)
- **PR** #27996: (rallytime) Don't fail if pip package is already present and pip1 is installed @ *2015-10-19 12:59:17 UTC*
  - d4604fdb26 Merge pull request #27996 from rallytime/fix-21845
  - f8380d751e Provide empty string as default stdout instead of None
  - f9406b5828 Don't fail if pip package is already present and pip1 is installed
- **PR** #28056: (rallytime) Back-port #28033 to 2015.5 @ *2015-10-19 12:55:10 UTC*
  - **PR** #28033: (twangboy) Fixed win_useradd.py (refs: #28056)
  - 28b97c514f Merge pull request #28056 from rallytime/bp-28033
  - af2c5ab759 Fixed win_useradd.py
- **PR** #28059: (rallytime) Back-port #28040 to 2015.5 @ *2015-10-18 16:17:29 UTC*
  - **PR** #28040: (erchn) Swift rackspace fixes (refs: #28059)
  - dfc3aaec74 Merge pull request #28059 from rallytime/bp-28040
  - 76a0d4937b Revert "Allow passing in auth_version, defaulting to 2."
  - 63d5675d34 default auth_version = 2
  - 8072716888 remove extra spaces
  - 9770f56f04 cleanup whitespace, default to None to be consistent with profile
  - f4adfe98c0 Allow passing in auth_version, defaulting to 2.
  - fab1ad39af Rackspace support for switft module.
- **ISSUE** #27534: (llevar) file.managed can't retrieve file via ftp (refs: #28047)
- **PR** #28047: (cachedout) Restore FTP functionality to file client @ *2015-10-18 16:16:46 UTC*
  - d1fa036b55 Merge pull request #28047 from cachedout/issue_27534
  - 6ea37ddbca Context manager
  - 4d6f6bb371 Lint
  - 59018289dc Restore FTP functionality to file client
- **PR** #28032: (twangboy) Fixed win_path.py @ *2015-10-17 15:16:15 UTC*
  - fd2ca2df1b Merge pull request #28032 from twangboy/fix_win_path
  - 2bcac93314 Fixed win_path.py
- **ISSUE** #26336: (jfindlay) windows user.present broken (refs: #28003)
- **PR** #28037: (rallytime) Back-port #28003 to 2015.5 @ *2015-10-16 20:59:52 UTC*
  - **PR** #28003: (twangboy) Fix #26336 (refs: #28037)
  - 88c1770be4 Merge pull request #28037 from rallytime/bp-28003
  - 4fcf51fb1e Fix PR #26336
- **PR** #28031: (jacobhammons) Updated release notes with additional CVE information @ *2015-10-16 16:19:37 UTC*
  - de727d8bd2 Merge pull request #28031 from jacobhammons/relnotes6

  - 05927bb6f0 Updated release notes with additional CVE information
- **ISSUE** #27897: (Inveracity) request to add \\r escape character for salt.states.host for windows (refs: #28008)
- **PR** #28008: (jfindlay) platform independent line endings in hosts mod @ *2015-10-16 13:20:28 UTC*
  - 16c0272849 Merge pull request #28008 from jfindlay/host_path
  - 9f7047dd3c platform independent line endings in hosts mod
- **ISSUE** #28010: (vakulich) Error "KeyError: 'ret'" appeared during salt.state run in orchestrate module if minion had an exception (refs: #28012)
- **PR** #28012: (rallytime) Clean up stack trace when something goes wrong with minion output @ *2015-10-16 12:40:59 UTC*
  - d41018fa8e Merge pull request #28012 from rallytime/fix-28010
  - 0d7059e0c2 Clean up stack trace when something goes wrong with minion output
- **PR** #27995: (jacobhammons) added link to grains security FAQ to targeting and pillar topics. @ *2015-10-15 21:15:31 UTC*
  - f728307001 Merge pull request #27995 from jacobhammons/pillar-doc
  - 2870af2ba3 added link to grains security FAQ to targeting and pillar topics.
- **PR** #27986: (jacobhammons) Changed current release to 5.6 and added CVE to release notes @ *2015-10-15 17:25:41 UTC*
  - efede904a7 Merge pull request #27986 from jacobhammons/dot6
  - bb61c68c11 Changed current release to 5.6 and added CVE to release notes
- **PR** #27913: (pass-by-value) Set default @ *2015-10-14 14:03:36 UTC*
  - 831ec680d9 Merge pull request #27913 from pass-by-value/proxmox_verify_ssl
  - 0b721efe37 Set default
- **PR** #27876: (terminalmage) 2015.5 branch: Fix traceback when 2015.8 git ext_pillar config schema used @ *2015-10-13 14:58:45 UTC*
  - 41cccb3a30 Merge pull request #27876 from terminalmage/git_pillar-AttributeError-2015.5
  - 07794c837a 2015.5 branch: Fix traceback when 2015.8 git ext_pillar config schema used
- **ISSUE** #27610: (benburkert) PR #27201 broke ssh_known_hosts with :port (refs: #27726)
- **ISSUE** #27187: (SeverinLeonhardt) ssh_known_hosts.present hashes other entries even with hash_hostname: false (refs: #27201)
- **PR** #27726: (jfindlay) deprecate hash_hostname in favor of hash_known_hosts @ *2015-10-12 16:19:09 UTC*
  - **PR** #27201: (jfindlay) rename hash_hostname to hash_known_hosts (refs: #27726)
  - c9c3b7760e Merge pull request #27726 from jfindlay/hashhosts
  - ebce47de7c add docs to ssh.recv_known_host exec module fcn
  - b6ee16b1e5 deprecate hash_hostname in favor of hash_known_hosts
- **ISSUE** #27735: (go8ose) saltutils.find_cached_job doesn't work (refs: #27776)
- **PR** #27776: (jfindlay) return message when local jobs_cache not found @ *2015-10-12 16:11:41 UTC*
  - 18e31584b0 Merge pull request #27776 from jfindlay/local_msg
  - 03afa3cffa return message when local jobs_cache not found

- **ISSUE** #27665: (ahammond) user.absent should not "fail" if /var/spool/mail/<user> already does not exist. (refs: #27766)
- **PR** #27766: (jfindlay) better check for debian userdel error @ *2015-10-12 15:14:33 UTC*
  - 86cc7b5537 Merge pull request #27766 from jfindlay/debmail
  - ee78da2c27 better check for debian userdel error
- **ISSUE** #27756: (iggy) syslog returner formats line incorrectly (refs: #27758)
- **PR** #27758: (iggy) Remove redundant text from syslog returner @ *2015-10-12 15:09:49 UTC*
  - c224386c9a Merge pull request #27758 from iggy/patch-1
  - 0994fb6a8c Remove redundant text from syslog returner
- **ISSUE** #27832: (viking60) Salt fails to recognize Manjaro (as an Arch derivate) (refs: #27841)
- **PR** #27841: (terminalmage) Detect Manjaro Linux as Arch derivative @ *2015-10-12 14:53:46 UTC*
  - 34a005041f Merge pull request #27841 from terminalmage/issue27832
  - 8e09fbd6a3 Detect Manjaro Linux as Arch derivative
- **ISSUE** #26538: (seanjnkns) salt.states.file.managed generates warning when used in place of salt.states.file.touch (refs: #27806)
- **PR** #27852: (rallytime) Back-port #27806 to 2015.5 @ *2015-10-12 14:53:17 UTC*
  - **PR** #27806: (blast-hardcheese) Empty string is falsy (refs: #27852)
  - 3944a498ad Merge pull request #27852 from rallytime/bp-27806
  - a84bf18bc4 Empty string is falsy
- **ISSUE** #27831: (basepi) v2015.5.5 highstate outputter stacktracing for jobs.lookup_jid (refs: #27838)
- **PR** #27838: (basepi) [2015.5] Fix highstate outputter for jobs.lookup_jid @ *2015-10-09 22:26:28 UTC*
  - **PR** #25521: (cachedout) Fix outputter for state.orch (refs: #27838)
  - 7508a1c474 Merge pull request #27838 from basepi/fix.runner.highstate.outputter.27831
  - 8ae9b66fd9 Don't pop 'outputter', we expect it further down
- **PR** #27791: (eguven) 2015.5 postgres_user groups backport @ *2015-10-08 23:59:08 UTC*
  - d178315f93 Merge pull request #27791 from eguven/2015.5-postgres-user-groups-backport
  - 2caf1d21d6 fix test
  - bc90c5bffe improve change reporting for postgres_user groups
  - 8712bce91a backport postgres_user groups
- **PR** #27759: (basepi) [2015.5] Merge forward from 2014.7 to 2015.5 @ *2015-10-07 18:01:54 UTC*
  - b2937b6a16 Merge pull request #27759 from basepi/merge-forward-2015.5
  - 792ee084bb Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.5
  - d284eb165b Merge pull request #27390 from JaseFace/schedule-missing-enabled
  - 563db71bfd Ensure we pass on the enable setting if present, or use the default of True if not in build_schedule_item() Prior to this, when schedule.present compares the existing schedule to the one crafted by this function, enabled will actually be removed at each run. schedule.present sees a modification needs to be made, and invokes schedule.modify, which does so with enabled: True, creating and endless loop of an 'enabled' removal and addition.

- **ISSUE** #26673: (robkinyon) __virtual__() doesn't work without __virtualname__ (refs: #27732)
- **PR** #27732: (jacobhammons) update docs for __virtual__ and __virtualname__ @ *2015-10-07 17:29:31 UTC*
  - 4b9128b491 Merge pull request #27732 from jacobhammons/26673
  - 75cc07cf10 noted that __virtual__ can return False and an error string
  - b928e1afa8 update docs for __virtual__ and __virtualname__ Refs #26673
- **PR** #27747: (Sacro) Chocolatey doesn't have a help command. @ *2015-10-07 16:06:53 UTC*
  - a130896d1c Merge pull request #27747 from Sacro/fix-chocolatey-version
  - 8f1fa9e78e Chocolatey doesn't have a help command.
- **PR** #27733: (jacobhammons) hardening topic - updates to docs.saltstack.com theme @ *2015-10-07 01:44:00 UTC*
  - 4e48651de0 Merge pull request #27733 from jacobhammons/bug-fixes
  - cbecd4f553 Updated saltstack2 theme to add SaltConf16 banner
  - 117e0c2bcc Added hardening topic based on the information in Refs #27088
- **ISSUE** #9051: (olenz) Add bash completion to the docs (refs: #27706)
- **ISSUE** #27005: (johanek) grains precedence (refs: #27706)
- **ISSUE** #21475: (quantonganh) Targeting with pillar should be added in to the main targeting page (refs: #27706)
- **ISSUE** #14876: (whiteinge) Create a pre-Salted tutorial VM (refs: #27706)
- **ISSUE** #13407: (gravyboat) Create page explaining how to pass variables on the command line (refs: #27706)
- **PR** #27706: (jacobhammons) Assorted doc bugs @ *2015-10-06 05:35:29 UTC*
  - c58da846bf Merge pull request #27706 from jacobhammons/bug-fixes
  - 76dc8de71b Assorted doc bugs Refs #9051 Refs #13407 Refs #21475 Refs #14876 Refs #27005
- **PR** #27695: (rallytime) Back-port #27671 to 2015.5 @ *2015-10-05 21:57:36 UTC*
  - **PR** #27671: (gashev) Added skip test_ext_pillar_env_mapping if git module does not exist. (refs: #27695)
  - 43fba89865 Merge pull request #27695 from rallytime/bp-27671
  - 2a88028595 Added skip test_ext_pillar_env_mapping if git module does not exist.
- **ISSUE** #27501: (yermulnik) [FreeBSD] "pkg search" behavior changed since 1.5 series (refs: #27524)
- **PR** #27524: (jfindlay) parse pkgng output in quiet mode for >= 1.6.1 @ *2015-10-05 21:22:40 UTC*
  - cb3d92676e Merge pull request #27524 from jfindlay/pkgng_quiet
  - 5e9107b970 parse pkgng output in quiet mode for >= 1.6.0
- **PR** #27686: (rallytime) Back-port #27476 to 2015.5 @ *2015-10-05 21:17:59 UTC*
  - **PR** #27476: (belvedere-trading) fix for: https://github.com/saltstack/salt/issues/27373 (refs: #27686)
  - 5b88c55cc3 Merge pull request #27686 from rallytime/bp-27476
  - 3e08d3de8a fix for: https://github.com/saltstack/salt/issues/27373
- **ISSUE** #27655: (gracinet) postgres_local_cache handling of success (refs: #27656)
- **PR** #27684: (rallytime) Back-port #27656 to 2015.5 @ *2015-10-05 21:17:55 UTC*
  - **PR** #27656: (gracinet) Fix #27655: handling of success in postgres_local_cache (refs: #27684)
  - f9ddd4647f Merge pull request #27684 from rallytime/bp-27656

  - d3780cba00 Fix #27655: handling of success in postgres_local_cache
- **PR** #27683: (rallytime) Back-port #27659 to 2015.5 @ *2015-10-05 21:17:30 UTC*
  - **PR** #27659: (gnubyexample) .pub as public key is what we should send to remote (refs: #27683)
  - 7ca6f854ff Merge pull request #27683 from rallytime/bp-27659
  - 84b6ee0c58 .pub as public key is what we should send to remote
- **PR** #27682: (rallytime) Back-port #27566 to 2015.5 @ *2015-10-05 21:17:26 UTC*
  - **PR** #27566: (blueyed) returners.local_cache: fix endless loop on OSError (refs: #27682)
  - a0f3e34656 Merge pull request #27682 from rallytime/bp-27566
  - 2a44255748 minor: fix/format doc for returners.local_cache.prep_jid
  - fd485e2396 returners.local_cache: fix endless loop on OSError
- **ISSUE** #25813: (whytewolf) debconf.set throwing exception in 2015.8.0rc2 (refs: #25928)
- **PR** #27681: (rallytime) Back-port #25928 to 2015.5 @ *2015-10-05 21:17:19 UTC*
  - **PR** #25928: (cachedout) Fix stacktrace for non-existant states (refs: #27681)
  - 0b9ba911c4 Merge pull request #27681 from rallytime/bp-25928
  - 17e1ddf137 Fix stacktrace for non-existant states
- **ISSUE** #27505: (silenius) [FreeBSD] state.service + provider daemontools is broken (refs: #27535)
- **PR** #27680: (rallytime) Back-port #27535 to 2015.5 @ *2015-10-05 21:17:10 UTC*
  - **PR** #27535: (silenius) Issue 27505 (refs: #27680)
  - 23da0d316a Merge pull request #27680 from rallytime/bp-27535
  - 04aed5e105 Versionadded change since 2015.5.6 has already been tagged
  - 579f2646ba .. versionadded:: 2015.5.6
  - cbaf46e066 python <2.7 compability (pylint issue)
  - ecde499478 s/bin/b to avoid confusion with bin()
  - 4237c5db80 add a __virtual__ to check that daemontools is installed properly
  - 623935a1bc fix doc
  - 573de3abd6 fix pylint issue
  - 5eb6a30d40 fix pep8 issues
  - 298cf4f5c0 import missing logging module
  - fe0ad36609 log was missing
  - e457083465 s/systemd/FreeBSD
  - 3512712e89 forgot service name..
  - 8f193a7bcc fixes #27505
- **PR** #27442: (JaseFace) Ensure we pass on the enable setting if present, or use the default of True if not in build_schedule_item() @ *2015-10-05 18:01:29 UTC*
  - 7d7b97eab6 Merge pull request #27442 from JaseFace/fix-27391-for-2015.5

  - bfbf63e1cc Ensure we pass on the enable setting if present, or use the default of True if not in build_schedule_item() Prior to this, when schedule.present compares the existing schedule to the one crafted by this function, enabled will actually be removed at each run. schedule.present sees a modification needs to be made, and invokes schedule.modify, which does so with enabled: True, creating and endless loop of an 'enabled' removal and addition.

- **ISSUE** #26320: (schlagify) pkg & diskusage beacons not sending alerts (refs: #27641)
- **PR** #27641: (rallytime) Gate the psutil import and add depends doc for diskusage beacon @ *2015-10-05 17:00:48 UTC*
  - ccbba8656b Merge pull request #27641 from rallytime/gate-psutil-diskusage
  - da2d93a3dd Gate the psutil import and add depends doc for diskusage beacon
- **PR** #27644: (rallytime) Back-port #27640 to 2015.5 @ *2015-10-05 14:55:31 UTC*
  - **PR** #27640: (stephen144) fix typo in default pillar path (refs: #27644)
  - 09183994f9 Merge pull request #27644 from rallytime/bp-27640
  - a9063a9745 fix typo in default pillar path
- **ISSUE** #27609: (rallytime) GCE with various external_ip settings cause stacktraces (refs: #27612)
- **PR** #27612: (rallytime) Fix GCE external_ip stacktraces in 2015.5 @ *2015-10-02 15:42:20 UTC*
  - 27fcecccbe Merge pull request #27612 from rallytime/fix-27609
  - 8dc047dc18 If external_up is set to None, don't stacktrace, just use the private ip.
  - 2ebf790f9f [salt-cloud] gce: don't stacktrace if Ephemeral is given instead of ephemeral
- **PR** #27568: (jacobhammons) regenerated man pages @ *2015-10-01 15:39:37 UTC*
  - c84a1edc1b Merge pull request #27568 from jacobhammons/man-pages-five
  - b59c03d20d regenerated man pages

28.2.75 Salt 2015.5.8 Release Notes

release 2015-12-01

Version 2015.5.8 is a bugfix release for *2015.5.0*.

Statistics

- Total Merges: **17**
- Total Issue References: **12**
- Total PR References: **27**
- Contributors: **12** (MasterNayru, TronPaul, basepi, cachedout, cxmcc, jfindlay, kevinlondon, messa, rallytime, tehmaspc, twangboy, whiteinge)

Security Fix

CVE-2015-8034 Saving `state.sls` cache data to disk with insecure permissions

This affects users of the `state.sls` function. The state run cache on the minion was being created with incorrect permissions. This file could potentially contain sensitive data that was inserted via jinja into the state SLS files. The permissions for this file are now being set correctly. Thanks to zmalone for bringing this issue to our attention.

Changelog for v2015.5.7..v2015.5.8

Generated at: 2018-05-27 22:25:07 UTC

- **ISSUE** #28883: (ldelossa) Issues running select states - local variable 'salt' referenced before assignment (refs: #29113)
- **PR** #29164: (jfindlay) Backport #29113 @ *2015-11-24 21:26:17 UTC*
  - **PR** #29113: (TronPaul) Kill unneeded import (refs: #29164)
  - **PR** #28740: (MasterNayru) Add missing S3 module import (refs: #28839, #29113)
  - a26c10a811 Merge pull request #29164 from jfindlay/bp-29113
  - 50fab35188 kill unneeded import
- **PR** #29138: (jfindlay) add 2015.5.8 release notes @ *2015-11-23 23:22:48 UTC*
  - 4f03196e7d Merge pull request #29138 from jfindlay/2015.5
  - be045f5cb1 add 2015.5.8 release notes
- **ISSUE** #29110: (mohshami) 2015.8.2 broke orchestration (refs: #29122)
- **ISSUE** #28010: (vakulich) Error "KeyError: 'ret'" appeared during salt.state run in orchestrate module if minion had an exception (refs: #28012)
- **PR** #29128: (cachedout) Set a safer default value for ret in saltmod @ *2015-11-23 17:07:40 UTC*
  - **PR** #29122: (cachedout) Fix broken state orchestration (refs: #29128)
  - **PR** #28012: (rallytime) Clean up stack trace when something goes wrong with minion output (refs: #29122)
  - 219367a23d Merge pull request #29128 from cachedout/tweak_29122
  - b08858b040 Missed check
  - 584efe81ee Set a safer default value for ret in saltmod
- **ISSUE** #29110: (mohshami) 2015.8.2 broke orchestration (refs: #29122)
- **ISSUE** #28010: (vakulich) Error "KeyError: 'ret'" appeared during salt.state run in orchestrate module if minion had an exception (refs: #28012)
- **PR** #29122: (cachedout) Fix broken state orchestration (refs: #29128) @ *2015-11-23 16:24:18 UTC*
  - **PR** #28012: (rallytime) Clean up stack trace when something goes wrong with minion output (refs: #29122)
  - 2250a36647 Merge pull request #29122 from cachedout/issue_29110
  - 4b9302d794 Fix broken state orchestration
- **PR** #29096: (rallytime) Back-port #29093 to 2015.5 @ *2015-11-22 17:02:51 UTC*
  - **PR** #29093: (cxmcc) Compare gem versions as a string. (refs: #29096)
  - 200e771efb Merge pull request #29096 from rallytime/bp-29093
  - f5734423a4 Compare gem versions as a string.

- **PR** #29084: (rallytime) Back-port #29055 to 2015.5 @ *2015-11-20 20:57:54 UTC*
  - **PR** #29055: (cachedout) Add section to style guide (refs: #29084)
  - d8a2018bc8 Merge pull request #29084 from rallytime/bp-29055
  - 52e650aed9 Add section to style guide
- **PR** #29083: (rallytime) Back-port #29053 to 2015.5 @ *2015-11-20 20:57:38 UTC*
  - **PR** #29053: (kevinlondon) Update rabbitmq_user.py (refs: #29083)
  - b5cff1a351 Merge pull request #29083 from rallytime/bp-29053
  - f1884de0e7 Update rabbitmq_user.py
- **ISSUE** #28928: (twangboy) Fix user.present 2015.5 (refs: #28932)
- **PR** #28932: (twangboy) Fixed user.present / user.absent in windows @ *2015-11-18 21:45:53 UTC*
  - **PR** #28627: (twangboy) Backport win_useradd (refs: #28932)
  - b3e3bebef0 Merge pull request #28932 from twangboy/fix_28928
  - 0653a04887 Fixed user.present / user.absent in windows
- **ISSUE** #26911: (dsumsky) file.manage state does not work with Amazon S3 URLs on Windows (refs: #28630)
- **ISSUE** #13850: (ryan-lane) s3:// urls in file.managed (and likely elsewhere) require s3.key and s3.keyid to be in minion config (refs: #28630)
- **PR** #29011: (rallytime) Back-port #28630 to 2015.5 @ *2015-11-18 17:50:05 UTC*
  - **PR** #28630: (messa) Use S3 credentials from Pillar (refs: #29011)
  - a2e4a227e0 Merge pull request #29011 from rallytime/bp-28630
  - 7baccc1b05 Lint - newline before def
  - 9e5c16d4da Reading S3 credentials from Pillar
  - a3216f813d Fixed requests HTTPError handler, it was still in urllib2 style
- **PR** #28982: (basepi) [2015.5] Merge forward from 2014.7 to 2015.5 @ *2015-11-18 00:49:32 UTC*
  - 1a4cd6002f Merge pull request #28982 from basepi/merge-forward-2015.5
  - bfbb109fbd Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.5
  - 4b8bdd0afb Merge pull request #28839 from cachedout/revert_28740
  - 215b26c06f Revert #28740
- **ISSUE** #28947: (dmyerscough) sdb modules are not synced out (refs: #28949)
- **PR** #28949: (whiteinge) Add sync_sdb execution function @ *2015-11-17 15:35:38 UTC*
  - edd26d763a Merge pull request #28949 from whiteinge/sync-sdb
  - b0ec9ab25b Add sync_sdb execution function
- **ISSUE** #28888: (twangboy) Fix file.comment (refs: #28930)
- **PR** #28930: (twangboy) Added missing import mmap required by file.py @ *2015-11-16 23:17:23 UTC*
  - 43da1bc4ce Merge pull request #28930 from twangboy/fix_28888
  - f5c489eaad Added missing import mmap required by file.py
- **PR** #28908: (rallytime) A couple of spelling fixes for doc conventions page. @ *2015-11-16 02:29:35 UTC*

  - 2488b873b8 Merge pull request #28908 from rallytime/doc-convention-spelling
  - 60e6eddb77 A couple of spelling fixes for doc conventions page.
- **ISSUE** #22442: (allanliu) rest_cherrypy /keys URL does not handle JSON requests (refs: #28902)
- **PR** #28902: (whiteinge) Fix missing JSON support for /keys endpoint @ *2015-11-15 15:36:05 UTC*
  - 827a1ae020 Merge pull request #28902 from whiteinge/json-keys
  - 9745903301 Fix missing JSON support for /keys endpoint
- **PR** #28897: (rallytime) Back-port #28873 to 2015.5 @ *2015-11-15 00:43:35 UTC*
  - **PR** #28873: (tehmaspc) Fix salt-cloud help output typo (refs: #28897)
  - d23bd49130 Merge pull request #28897 from rallytime/bp-28873
  - 077e671ead Fix salt-cloud help output typo
- **ISSUE** #28870: (basepi) mdadm commands failing (refs: #28871)
- **PR** #28871: (basepi) [2015.5] Fix command generation for mdadm.assemble @ *2015-11-13 21:54:33 UTC*
  - a9dc8b6ca6 Merge pull request #28871 from basepi/mdadm.fix.28870
  - 323bc2d2ac Fix command generation for mdadm.assemble

28.2.76 Salt 2015.5.9 Release Notes

release 2016-01-11

Version 2015.5.9 is a bugfix release for *2015.5.0*.

Statistics

- Total Merges: **45**
- Total Issue References: **21**
- Total PR References: **48**
- Contributors: **21** (abednarik, aletourneau, attiasr, basepi, cachedout, clan, clarkperkins, cro, dmyerscough, jacobhammons, jfindlay, jsutton, justinta, lorengordon, markckimball, mpreziuso, rallytime, terminalmage, titilambert, twangboy, zmalone)

Changelog for v2015.5.8..v2015.5.9

Generated at: 2018-05-27 22:31:06 UTC

- **PR** #30248: (jfindlay) add 2015.5.9 release notes @ *2016-01-08 23:13:10 UTC*
  - 92889db638 Merge pull request #30248 from jfindlay/2015.5
  - 741f7aba31 add 2015.5.9 release notes
- **PR** #30237: (jacobhammons) Updated man pages and doc version for 2015.5.9 @ *2016-01-08 18:10:05 UTC*
  - 7a329d89d7 Merge pull request #30237 from jacobhammons/man-pages-prev
  - 2431c4c5c3 Updated man page and doc conf.py copyright year to 2016
  - fe3da1c174 Updated man pages and doc version for 2015.5.9

- **PR** #30207: (rallytime) Use correct spacing in rabbitmq state examples @ *2016-01-07 18:37:35 UTC*
  - 2c0b725924 Merge pull request #30207 from rallytime/rabbitmq_states_doc_fix
  - 8d48c24182 Use correct spacing in rabbitmq state examples
- **PR** #30191: (jacobhammons) Updated doc site banners @ *2016-01-06 22:37:40 UTC*
  - b49cf910f4 Merge pull request #30191 from jacobhammons/banner-prev
  - c3390955b0 Updated doc site banners
- **ISSUE** #29633: (twellspring) user.present does not modify home directory (refs: #30125)
- **PR** #30125: (abednarik) Update user home event when createhome is set to False @ *2016-01-05 18:15:38 UTC*
  - 9363d6f5b6 Merge pull request #30125 from abednarik/update_user_home
  - 56544a77f6 Update user home event when createhome is set to False
- **ISSUE** #10155: (jhenry82) Option to select a random master in multi-master mode (refs: #30127)
- **PR** #30127: (jsutton) Updating documentation and example minion config for random_master/master_shuffle. @ *2016-01-04 19:30:50 UTC*
  - 1a5d585d91 Merge pull request #30127 from jsutton/clarify-documenation-for-random_master
  - 01dbf385ef Adding random_master to reference and updating master_shuffle. Adding master_shuffle to the minion example config file as it is needed for multi-master PKI.
- **PR** #30110: (markckimball) Fixed flag sent to salt.utils.http in order for verify_ssl to work correctly @ *2015-12-31 21:17:53 UTC*
  - 28b1bbbe77 Merge pull request #30110 from markckimball/fix-verify_ssl-in-joyent-cloud
  - e1c08cb269 Fixed flag sent to salt.utils.http in order for verify_ssl to work appropriately.
- **PR** #30093: (zmalone) Noting that file_roots and "state tree" should both be avoided @ *2015-12-30 22:40:05 UTC*
  - 040412b0b1 Merge pull request #30093 from zmalone/pillar-notes
  - cfbfd58afe Noting that file_roots and "state tree" should both be avoided, because in some environments, the actual states show up another level down. Adding notes about why this is undesirable.
- **ISSUE** #28120: (jtylers) Clear text passwords (refs: #30097)
- **PR** #30097: (cachedout) Note concern about cleartext password in docs for shadow.gen_password @ *2015-12-30 22:37:33 UTC*
  - 25edefc93a Merge pull request #30097 from cachedout/note_on_password_process_list
  - 58aec884ef Note concern about cleartext password in docs for shadow.gen_password
- **PR** #30089: (mpreziuso) Fixes terminology and adds more accurate details about the algorithms @ *2015-12-30 20:02:18 UTC*
  - 6b1c3a6bf2 Merge pull request #30089 from mpreziuso/patch-1
  - 50533add40 Fixes terminology and adds more accurate details about the algorithms
- **ISSUE** #29921: (anlutro) pygit 0.21 not fully supported? (refs: #30086)
- **PR** #30086: (cachedout) Document that gitfs needs recent libs @ *2015-12-30 19:26:05 UTC*
  - 200d09385d Merge pull request #30086 from cachedout/issue_29921
  - 8c29e2dd6a Document that gitfs needs recent libs
- **ISSUE** #27835: (bertjwregeer) [FreeBSD] salt-ssh hangs forever (refs: #30070)

- **PR** #30070: (cachedout) Add documentation on debugging salt-ssh @ *2015-12-29 23:00:06 UTC*
    - 404414bf57 Merge pull request #30070 from cachedout/issue_27835
    - 60431e342a Add documentation on debugging salt-ssh
- **PR** #30059: (mpreziuso) Fixes wrong function scope @ *2015-12-29 16:12:06 UTC*
    - 84db12212d Merge pull request #30059 from mpreziuso/patch-1
    - 1cb1c2da07 Fixes wrong function scope
- **PR** #30025: (justinta) Skipping some Boto tests until resolved moto issue @ *2015-12-28 15:21:45 UTC*
    - **PR** #29725: (cachedout) Disable some boto tests per resolution of moto issue (refs: #30025)
    - 1c6c9b1a06 Merge pull request #30025 from jtand/boto_tests
    - e706642152 Skipping some Boto tests until resolved moto issue
- **ISSUE** #28956: (racooper) Netscaler module doc enhancements (refs: #29949)
- **PR** #29949: (aletourneau) Enhanced netscaler docstring @ *2015-12-22 20:26:52 UTC*
    - 0f91021c59 Merge pull request #29949 from aletourneau/2015.5
    - cf855fe262 Fixed trailing white spaces
    - 864801e002 fixed version
    - 041d9346c4 Enhanced netscaler docstring
- **PR** #29941: (cachedout) Fix spelling error in boto_vpc @ *2015-12-22 15:49:54 UTC*
    - 229d3eb60b Merge pull request #29941 from cachedout/boto_spelling
    - b11bfd07b8 Fix spelling error in boto_vpc
- **ISSUE** #29880: (githubcdr) Salt pkg.uptodate fails on Arch linux (refs: #29908)
- **PR** #29908: (cachedout) Allow kwargs to be passed to pacman provide for update func @ *2015-12-22 15:04:18 UTC*
    - 69c5ada636 Merge pull request #29908 from cachedout/issue_29880
    - 4cd77b4118 Allow kwargs to be passed to pacman provide for update func
- **ISSUE** #27056: (oogali) pkgng provider on FreeBSD does not do BATCH=yes (refs: #29909)
- **PR** #29909: (abednarik) FreeBSD pkgng fix for non-interactive install. @ *2015-12-22 15:03:50 UTC*
    - ad0de4d563 Merge pull request #29909 from abednarik/freebsd_pkgng_non_interactive_fix
    - 8ac213001a FreeBSD pkgng fix for non-interactive install.
- **ISSUE** #24698: (cmhe) docker.installed not working (salt 2015.5.0, docker 1.6.2, dockerpy 0.5.3) (refs: #29730)
- **PR** #29730: (rallytime) Update docker-py version requirement to 0.6.0 for dockerio.py files @ *2015-12-16 14:44:40 UTC*
    - f43f3d166c Merge pull request #29730 from rallytime/fix-24698
    - 120fd5fdf0 Update docker-py version requirement to 0.6.0 for dockerio.py files
- **ISSUE** #23343: (micaelbergeron) npm state ignore the requested version (refs: #29715)
- **ISSUE** #18647: (hundt) Version number in npm state name does not result in correct version being installed (refs: #29715)
- **PR** #29715: (rallytime) Install correct package version, if provided, for npm state. @ *2015-12-15 23:19:45 UTC*

  - c393a4175a Merge pull request #29715 from rallytime/fix-23343
  - a0ed857c37 Install correct package version, if provided, for npm state.
- **PR** #29721: (terminalmage) Fix display of multiline strings when iterating over a list @ *2015-12-15 22:16:10 UTC*
  - 1310afbbc2 Merge pull request #29721 from terminalmage/nested-output-multiline-fix
  - 761be9cb93 Fix display of multiline strings when iterating over a list
- **ISSUE** #29488: (Shad0w1nk) salt.cloud.clouds.vmware.revert_to_snapshot crash when using the default value (refs: #29646)
- **PR** #29646: (rallytime) Don't stacktrace on kwargs.get if kwargs=None @ *2015-12-15 19:02:58 UTC*
  - 52cc07cec9 Merge pull request #29646 from rallytime/fix-29488
  - c5fa9e9351 Don't stacktrace on kwargs.get if kwargs=None
- **ISSUE** #29661: (mosuowhq) bug report when creating VM in /salt/cloud/clouds/nova.py (refs: #29673)
- **PR** #29673: (rallytime) Default value should be False and not 'False' @ *2015-12-14 18:08:44 UTC*
  - f606c23ea8 Merge pull request #29673 from rallytime/fix-29661
  - e4af7a1157 Default value should be False and not 'False'
- **PR** #29527: (jfindlay) 2015.5.7 notes: add note about not being released @ *2015-12-08 21:08:26 UTC*
  - f77c8e7baf Merge pull request #29527 from jfindlay/2015.5
  - 1a8044f0c9 2015.5.7 notes: add note about not being released
- **PR** #29539: (basepi) [2015.5] Merge forward from 2014.7 to 2015.5 @ *2015-12-08 19:14:51 UTC*
  - 867d550271 Merge pull request #29539 from basepi/merge-forward-2015.5
  - 2c9c4ba430 Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.5
  - 85aa70a6cb Merge pull request #29392 from jacobhammons/2014.7
    - d7f0db1dd8 updated version number to not reference a specific build from the latest branch
- **ISSUE** #12072: (vk00226) Passing user-data when provisioning ec2 instances (refs: #29504)
- **PR** #29504: (rallytime) Document userdata_file option for EC2 driver @ *2015-12-08 16:54:33 UTC*
  - de7f3d5a59 Merge pull request #29504 from rallytime/fix-12072
  - 8357c95dc2 Document userdata_file option for EC2 driver
- **ISSUE** #29101: (jessbreckenridge) Salt-cloud 2015.8.0 - del_*_vols_on_destroy does not work according to docs (refs: #29507)
- **PR** #29507: (rallytime) Switch volumes and del_*_on_destroy example ordering @ *2015-12-08 16:50:11 UTC*
  - 65deba8bb5 Merge pull request #29507 from rallytime/ec2-doc-fix
  - 90b4823bc2 Switch volumes and del_*_on_destroy example ordering
- **ISSUE** #28862: (trevor-h) salt-cloud uppercase timeout options no longer recognized (refs: #29469)
- **PR** #29469: (abednarik) Added Documentation note in salt cloud. @ *2015-12-07 18:27:46 UTC*
  - 0918c9294f Merge pull request #29469 from abednarik/doc_note_for_saltcloud_connection_timeout
  - 8e5c3e366a Added Documentation note in salt cloud.
- **PR** #29461: (dmyerscough) Fix resource limits, systemd sets the default too small @ *2015-12-05 16:26:34 UTC*

  - e43c7c05a6 Merge pull request #29461 from dmyerscough/fix-resource-limits
  - 85a8a3b033 Fix resource limits, systemd sets the default number of open files to 4096 causing te master to complain about limits when you have a large number of keys
- **ISSUE** #28526: (clarkperkins) yumpkg.installed broken in salt v2015.8.1 on CentOS 6 minions (refs: #28656)
- **PR** #29439: (rallytime) Back-port #28656 to 2015.5 @ *2015-12-04 22:56:17 UTC*
  - **PR** #28656: (clarkperkins) #28526 fixed yumpkg module issue with pkg.installed (refs: #29439)
  - 730f02fbdf Merge pull request #29439 from rallytime/bp-28656
  - 2f11bb021f #28526 fixed yumpkg module
- **PR** #29418: (jacobhammons) Added CVE 2015-8034 to 2015.5.8 release notes @ *2015-12-04 03:02:53 UTC*
  - 197210d52e Merge pull request #29418 from jacobhammons/dot8
  - 4f51a737f9 Added CVE 2015-8034 to 2015.5.8 release notes
- **PR** #29389: (jacobhammons) updated version numbers in documentation @ *2015-12-03 16:27:23 UTC*
  - b3452f2a1a Merge pull request #29389 from jacobhammons/2015.5
  - 824721ff36 updated version numbers
- **ISSUE** #26898: (twangboy) Symlinks in Windows (2015.8) (refs: #28191)
- **PR** #28501: (twangboy) Requested fixes for 26898 @ *2015-12-03 01:12:12 UTC*
  - **PR** #28420: (jfindlay) fix removal of symbolic links on windows in the file state (refs: #28501)
  - **PR** #28191: (twangboy) Fix 26898 (refs: #28420, #28501)
  - 6a7a95f28a Merge pull request #28501 from twangboy/jmoney_26898
  - c0cf33332c Fixed some Lint…
  - df17fc59d3 Merge pull request #6 from jfindlay/twang_test
    - bc7e0cfe64 add file.symlink unit tests
    - 9381dc7215 orthogonalize file.symlink unit tests
  - 8f462ba044 Merge pull request #5 from cachedout/fix_twangboy_test
    - 5293150d25 Fix tests
  - 7d39091c91 Fixed some more lint
  - 3dbd62af2c Fixed some tests… hopefully
  - f187db3288 Removed unnecessary logic
  - 89ebd268e6 Added file attributes restore on fail
  - 9ec72ca724 fix file state unit tests for win symlink feature
  - 69c32a663e Fixed some lint
  - 638dec5027 Fixed some tests… let's see if they're really are
  - 5ed7a99792 Replaced instances of shutil.rmtree in file state
  - 2651ce509f Fix file.remove for windows
- **ISSUE** #29344: (justinta) file.search broken on python 2.6 with empty files (refs: #29348)
- **PR** #29348: (justinta) Fixes an file.search on python2.6 @ *2015-12-02 23:26:36 UTC*

  - 760a521603 Merge pull request #29348 from jtand/file_search_fix
  - 04f82bd4fd Fixes an file.search on python2.6
- **ISSUE** #29206: (mschiff) ssh_known_hosts.present creates wrong known_hosts lines (refs: #29276)
- **PR** #29336: (rallytime) Back-port #29276 to 2015.5 @ *2015-12-02 19:37:42 UTC*
  - **PR** #29276: (abednarik) Prevent adding port twice when adding entry in known hosts (refs: #29336)
  - 51ea88d489 Merge pull request #29336 from rallytime/bp-29276
  - 3a0e19debb Prevent adding port twice when adding entry in known hosts
- **PR** #29333: (rallytime) Back-port #29280 to 2015.5 @ *2015-12-02 19:37:05 UTC*
  - **PR** #29280: (cachedout) [Doc] Add note for SVN state (refs: #29333)
  - **PR** #29165: (titilambert) [Doc] Add note for SVN state (refs: #29280, #29333)
  - 28255af52a Merge pull request #29333 from rallytime/bp-29280
  - 722d02ff4a Lint
  - 4a0040c1b4 [Doc] Add note for SVN state
- **PR** #29316: (basepi) [2015.5] Merge forward from 2014.7 to 2015.5 @ *2015-12-01 20:20:23 UTC*
  - 14e94b3593 Merge pull request #29316 from basepi/merge-forward-2015.5
  - 33f40b3c47 Merge remote-tracking branch 'upstream/2014.7' into merge-forward-2015.5
  - d2fb2109a3 Merge pull request #29296 from douardda/patch-3
  - d2885390f4 Use process KillMode on Debian systems also
- **PR** #29216: (clan) size is 0 doesn't mean no data, e.g, /proc/version @ *2015-11-30 20:01:43 UTC*
  - 6a2ffbfb7c Merge pull request #29216 from clan/file_search_on_proc_file
  - 91a20c07a1 try mmap first
  - 8aa4f2053e remove extra space to fix lint failure
  - d34e6b1a9a use read only if has read() method
  - 3209c1cdb5 size is 0 doesn't mean no data, e.g, /proc/version
- **PR** #29261: (attiasr) fix incorrect reinstallation of windows pkg @ *2015-11-30 18:28:42 UTC*
  - d6aaae8d7b Merge pull request #29261 from attiasr/patch-1
  - 7a99b90596 add log and return if pkg already installed
  - 1843c7ab8e fix incorrect reinstallation of windows pkg
- **PR** #29214: (cro) Doc for salt.utils.http should say verify_ssl not ssl_verify. @ *2015-11-25 23:55:38 UTC*
  - 9236188867 Merge pull request #29214 from cro/ssl_verify_ssl
  - e9c13c561b Doc bug–salt.utils.http takes verify_ssl not ssl_verify.
- **ISSUE** #29202: (lorengordon) Broken ca bundle lookup in *salt.utils.http.get_ca_bundle* (refs: #29204)
- **PR** #29204: (lorengordon) Use os.path.join to return full path to ca bundle @ *2015-11-25 20:00:42 UTC*
  - df7b35a86b Merge pull request #29204 from lorengordon/fix-29202
  - b1dae5e6fe Use os.path.join to return full path to ca bundle

28.2.77 Salt 2014.7.0 Release Notes - Codename Helium

This release is the largest Salt release ever, with more features and commits then any previous release of Salt. Everything from the new RAET transport to major updates in Salt Cloud and the merging of Salt API into the main project.

Important: The Fedora/RHEL/CentOS **salt-master** package has been modified for this release. The following components of Salt have been broken out and placed into their own packages:

- salt-syndic
- salt-cloud
- salt-ssh

When the **salt-master** package is upgraded, these components will be removed, and they will need to be manually installed.

Important: Compound/pillar matching have been temporarily disabled for the `mine` and `publish` modules for this release due to the possibility of inferring pillar data using pillar glob matching. A proper fix is now in the 2014.7 branch and scheduled for the 2014.7.1 release, and compound matching and non-globbing pillar matching will be re-enabled at that point.

Compound and pillar matching for normal salt commands are unaffected.

New Transport!

RAET Transport Option

This has been a HUGE amount of work, but the beta release of Salt with RAET is ready to go. RAET is a reliable queuing transport system that has been developed in partnership with a number of large enterprises to give Salt an alternative to ZeroMQ and a way to get Salt to scale well beyond tens of thousands of servers. Unlike ZeroMQ, RAET is completely asynchronous in every aspect of its operation and has been developed using the flow programming paradigm. This allows for many new capabilities to be added to Salt in the upcoming releases.

Please keep in mind that this is a beta release of RAET and we hope for bugs to be worked out, performance to be better realized and more in the 2015.5.0 release.

Simply stated, users running Salt with RAET should expect some hiccups as we hammer out the update. This is a BETA release of Salt RAET.

Salt SSH Enhancements

Salt SSH has just entered a new league, with substantial updates and improvements to make salt-ssh more reliable and easier then ever! From new features like the ansible roster and fileserver backends to the new pypi salt-ssh installer to lowered deps and a swath of bugfixes, salt-ssh is basically reborn!

Install salt-ssh Using pip

Salt-ssh is now pip-installable!

https://pypi.python.org/pypi/salt-ssh/

Pip will bring in all of the required deps, and while some deps are compiled, they all include pure python implementations, meaning that any compile errors which may be seen can be safely ignored.

```
pip install salt-ssh
```

Fileserver Backends

Salt-ssh can now use the salt fileserver backend system. This allows for the gitfs, hgfs, s3, and many more ways to centrally store states to be easily used with salt-ssh. This also allows for a distributed team to easily use a centralized source.

Saltfile Support

The new saltfile system makes it easy to have a user specific custom extended configuration.

Ext Pillar

Salt-ssh can now use the external pillar system. Making it easier then ever to use salt-ssh with teams.

No More sshpass

Thanks to the enhancements in the salt vt system, salt-ssh no longer requires sshpass to send passwords to ssh. This also makes the manipulation of ssh calls substantially more flexible, allowing for intercepting ssh calls in a much more fluid way.

Pure Python Shim

The salt-ssh call originally used a shell script to discover what version of python to execute with and determine the state of the ssh code deployment. This shell script has been replaced with a pure python version making it easy to increase the capability of the code deployment without causing platform inconsistency issues with different shell interpreters.

Custom Module Delivery

Custom modules are now seamlessly delivered. This makes the deployment of custom grains, states, execution modules and returners a seamless process.

CP Module Support

Salt-ssh now makes simple file transfers easier then ever! The *cp* module allows for files to be conveniently sent from the salt fileserver system down to systems.

More Thin Directory Options

Salt ssh functions by copying a subset of the salt code, or *salt thin* down to the target system. In the past this was always transferred to /tmp/.salt and cached there for subsequent commands.

Now, salt thin can be sent to a random directory and removed when the call is complete with the *-W* option. The new *-W* option still uses a static location but will clean up that location when finished.

The default *salt thin* location is now user defined, allowing multiple users to cleanly access the same systems.

State System Enhancements

New Imperative State Keyword "Listen"

The new `listen` and `listen_in` keywords allow for completely imperative states by calling the `mod_watch()` routine after all states have run instead of re-ordering the states.

Mod Aggregate Runtime Manipulator

The new `mod_aggregate` system allows for the state system to rewrite the state data during execution. This allows for state definitions to be aggregated dynamically at runtime.

The best example is found in the *`pkg`* state. If `mod_aggregate` is turned on, then when the first pkg state is reached, the state system will scan all of the other running states for pkg states and take all other packages set for install and install them all at once in the first pkg state.

These runtime modifications make it easy to run groups of states together. In future versions, we hope to fill out the `mod_aggregate` system to build in more and more optimizations.

For more documentation on `mod_aggregate`, see *the documentation*.

New Requisites: onchanges and onfail

The new `onchanges` and `onchanges_in` requisites make a state apply only if there are changes in the required state. This is useful to execute post hooks after changes occur on a system.

The other new requisites, `onfail`, and `onfail_in`, allow for a state to run in reaction to the failure of another state.

For more information about these new requisites, see the *requisites documentation*.

Global onlyif and unless

The `onlyif` and `unless` options can now be used for any state declaration.

Use `names` to expand and override values

The *names declaration* in Salt's state system can now override or add values to the expanded data structure. For example:

```
my_users:
  user.present:
    - names:
      - larry
      - curly
      - moe:
        - shell: /bin/zsh
        - groups:
          - wheel
    - shell: /bin/bash
```

Major Features

Scheduler Additions

The Salt scheduler system has received MAJOR enhancements, allowing for cron-like scheduling and much more granular timing routines. See *here* for more info.

Red Hat 7 Family Support

All the needed additions have been made to run Salt on RHEL 7 and derived OSes like CentOS and Scientific.

Fileserver Backends in salt-call

Fileserver backends like gitfs can now be used without a salt master! Just add the fileserver backend configuration to the minion config and execute salt-call. This has been a much-requested feature and we are happy to finally bring it to our users.

Amazon Execution Modules

An entire family of execution modules further enhancing Salt's Amazon Cloud support. They include the following:

- *Autoscale Groups* (includes *state support*) – related: *Launch Control* states
- *Cloud Watch* (includes *state support*)
- *Elastic Cache* (includes *state support*)
- *Elastic Load Balancer* (includes *state support*)
- *IAM Identity and Access Management* (includes *state support*)
- *Route53 DNS* (includes *state support*)
- *Security Groups* (includes *state support*)
- *Simple Queue Service* (includes *state support*)

LXC Runner Enhancements

BETA The Salt LXC management system has received a number of enhancements which make running an LXC cloud entirely from Salt an easy proposition.

Next Gen Docker Management

The Docker support in Salt has been increased at least ten fold. The Docker API is now completely exposed and Salt ships with Docker data tracking systems which make automating Docker deployments very easy.

Peer System Performance Improvements

The peer system communication routines have been refined to make the peer system substantially faster.

SDB

Encryption at rest for configs

GPG Renderer

Encrypted pillar at rest

OpenStack Expansion

Lots of new OpenStack stuff

Queues System

Ran change external queue systems into Salt events

Multi Master Failover Additions

Connecting to multiple masters is more dynamic then ever

Chef Execution Module

Managing Chef with Salt just got even easier!

salt-api Project Merge

The `salt-api` project has been merged into Salt core and is now available as part of the regular `salt-master` package install. No API changes were made, the **`salt-api`** script and init scripts remain intact.

`salt-api` has always provided Yet Another Pluggable Interface to Salt (TM) in the form of "netapi" modules. These are modules that bind to a port and start a service. Like many of Salt's other module types, netapi modules often have library and configuration dependencies. See the documentation for each module for instructions.

See also:

The full list of netapi modules.

Synchronous and Asynchronous Execution of Runner and Wheel Modules

`salt.runner.RunnerClient` and *`salt.wheel.WheelClient`* have both gained complimentary `cmd_sync` and `cmd_async` methods allowing for synchronous and asynchronous execution of any Runner or Wheel module function, all protected using Salt's *external authentication* system. `salt-api` benefits from this addition as well.

`rest_cherrypy` Additions

The *`rest_cherrypy`* netapi module provides the main REST API for Salt.

Web Hooks

This release of course includes the Web Hook additions from the most recent `salt-api` release, which allows external services to signal actions within a Salt infrastructure. External services such as Amazon SNS, Travis-CI, or GitHub, as well as internal services that cannot or should not run a Salt minion daemon can be used as first-class components in Salt's rich orchestration capabilities.

The raw HTTP request body is now available in the event data. This is sometimes required information for checking an HMAC signature in order to verify a HTTP request. As an example, Amazon or GitHub requests are signed this way.

Generating and Accepting Minion Keys

The *`/key`* convenience URL generates a public and private key for a minion, automatically pre-accepts the public key on the Salt Master, and returns both keys as a tarball for download.

This allows for easily bootstrapping the key on a new minion with a single HTTP call, such as with a Kickstart script, all using regular shell tools.

```
curl -sS http://salt-api.example.com:8000/keys \
        -d mid=jerry \
        -d username=kickstart \
        -d password=kickstart \
        -d eauth=pam \
        -o jerry-salt-keys.tar
```

Fileserver Backend Enhancements

All of the fileserver backends have been overhauled to be faster, lighter, and more reliable. The VCS backends (*`gitfs`*, *`hgfs`*, and *`svnfs`*) have also received a **lot** of new features.

Additionally, most config parameters for the VCS backends can now be configured on a per-remote basis, allowing for global config parameters to be overridden for a specific gitfs/hgfs/svnfs remote.

New `gitfs` Features

Pygit2 and Dulwich

In addition to supporting GitPython, support for pygit2 (0.20.3 and newer) and dulwich have been added. Provided a compatible version of pygit2 is installed, it will now be the default provider. The config parameter *`gitfs_provider`* has been added to allow one to choose a specific provider for gitfs.

Mountpoints

Prior to this release, to serve a file from gitfs at a salt fileserver URL of `salt://foo/bar/baz.txt`, it was necessary to ensure that the parent directories existed in the repository. A new config parameter *`gitfs_mountpoint`* allows gitfs remotes to be exposed starting at a user-defined `salt://` URL.

Environment Whitelisting/Blacklisting

By default, gitfs will expose all branches and tags as Salt fileserver environments. Two new config parameters, `gitfs_env_whitelist`, and `gitfs_env_blacklist`, allow more control over which branches and tags are exposed. More detailed information on how these two options work can be found in the *Gitfs Walkthrough*.

Expanded Authentication Support

As of pygit2 0.20.3, both http(s) and SSH key authentication are supported, and Salt now also supports both authentication methods when using pygit2. Keep in mind that pygit2 0.20.3 is not yet available on many platforms, so those who had been using authenticated git repositories with a passphraseless key should stick to GitPython if a new enough pygit2 is not yet available for the platform on which the master is running.

A full explanation of how to use authentication can be found in the *Gitfs Walkthrough*.

New `hgfs` Features

Mountpoints

This feature works exactly like its *gitfs counterpart*. The new config parameter is called *`hgfs_mountpoint`*.

Environment Whitelisting/Blacklisting

This feature works exactly like its *gitfs counterpart*. The new config parameters are called `hgfs_env_whitelist` and `hgfs_env_blacklist`.

New `svnfs` Features

Mountpoints

This feature works exactly like its *gitfs counterpart*. The new config parameter is called *`svnfs_mountpoint`*.

Environment Whitelisting/Blacklisting

This feature works exactly like its *gitfs counterpart*. The new config parameters are called `svnfs_env_whitelist` and `svnfs_env_blacklist`.

Configurable Trunk/Branches/Tags Paths

Prior to this release, the paths where trunk, branches, and tags were located could only be in directories named "trunk", "branches", and "tags" directly under the root of the repository. Three new config parameters (`svnfs_trunk`, `svnfs_branches`, and `svnfs_tags`) allow SVN repositories which are laid out differently to be used with svnfs.

New `minionfs` Features

Mountpoint

This feature works exactly like its *gitfs counterpart*. The new config parameter is called `minionfs_mountpoint`. The one major difference is that, as minionfs doesn't use multiple remotes (it just serves up files pushed to the master using `cp.push`) there is no such thing as a per-remote configuration for `minionfs_mountpoint`.

Changing the Saltenv from Which Files are Served

A new config parameter (`minionfs_env`) allows minionfs files to be served from a Salt fileserver environment other than `base`.

Minion Whitelisting/Blacklisting

By default, minionfs will expose the pushed files from all minions. Two new config parameters, `minionfs_whitelist`, and `minionfs_blacklist`, allow minionfs to be restricted to serve files from only the desired minions.

Pyobjects Renderer

Salt now ships with with the `Pyobjects Renderer` that allows for construction of States using pure Python with an idiomatic object interface.

New Modules

In addition to the Amazon modules mentioned above, there are also several other new execution modules:

- `Oracle`
- `Random`
- `Redis`
- `Amazon Simple Queue Service`
- `Block Device Management`
- `CoreOS etcd`

- *Genesis*
- `InfluxDB`
- *Server Density*
- *Twilio Notifications*
- *Varnish*
- *ZNC IRC Bouncer*
- *SMTP*

New Runners

- *Map/Reduce Style*
- *Queue*

New External Pillars

- *CoreOS etcd*

New Salt-Cloud Providers

- *Aliyun ECS Cloud*
- *LXC Containers*
- *Proxmox (OpenVZ containers & KVM)*

Salt Call Change

When used with a returner, salt-call now contacts a master if `--local` is not specicified.

Deprecations

`salt.modules.virtualenv_mod`

- Removed deprecated `memoize` function from `salt/utils/__init__.py` (deprecated)
- Removed deprecated `no_site_packages` argument from `create` function (deprecated)
- Removed deprecated `check_dns` argument from `minion_config` and `apply_minion_config` functions (deprecated)
- Removed deprecated `OutputOptionsWithTextMixIn` class from `salt/utils/parsers.py` (deprecated)
- Removed the following deprecated functions from `salt/modules/ps.py`: - `physical_memory_usage` (deprecated) - `virtual_memory_usage` (deprecated) - `cached_physical_memory` (deprecated) - `physical_memory_buffers` (deprecated)
- Removed deprecated cloud arguments from `cloud_config` function in `salt/config.py`: - `vm_config` (deprecated) - `vm_config_path` (deprecated)

- Removed deprecated `libcloud_version` function from `salt/cloud/libcloudfuncs.py` (deprecated)
- Removed deprecated `CloudConfigMixIn` class from `salt/utils/parsers.py` (deprecated)

28.2.78 Salt 2014.7.1 Release Notes

release 2015-01-12

Version 2014.7.1 is a bugfix release for *2014.7.0*.

The changes include:

- Fixed gitfs serving symlinks in `file.recurse` states (issue #17700)
- Fixed holding of multiple packages (YUM) when combined with version pinning (issue #18468)
- Fixed use of Jinja templates in masterless mode with non-roots fileserver backend (issue #17963)
- Re-enabled pillar and compound matching for mine and publish calls. Note that pillar globbing is still disabled for those modes, for security reasons. (issue #17194)
- Fix for `tty: True` in salt-ssh (issue #16847)
- Fix for supervisord states when supervisor not installed to system python (issue #18044)
- Fix for logging when `log_level='quiet'` for `cmd.run` (issue #19479)

28.2.79 Salt 2014.7.2 Release Notes

release 2015-02-09

Version 2014.7.2 is a bugfix release for *2014.7.0*.

The changes include:

- Fix erroneous warnings for systemd service enabled check (issue #19606)
- Fix FreeBSD kernel module loading, listing, and persistence `kmod` (issue #197151, issue #19682)
- Allow case-sensitive npm package names in the `npm state`. This may break behavior for people expecting the state to lowercase their npm package names for them. The `npm module` was never affected by mandatory lowercasing. (issue #20329)
- Deprecate the `activate` parameter for pip.install for both the `module` and the `state`. If `bin_env` is given and points to a virtualenv, there is no need to activate that virtualenv in a shell for pip to install to the virtualenv.
- Fix a file-locking bug in gitfs (issue #18839)
- Deprecated `archive_user` in favor of standardized `user` parameter in `state` and added `group` parameter.

28.2.80 Salt 2014.7.3 Release Notes

release 2015-03-25

Version 2014.7.3 is a bugfix release for *2014.7.0*.

Changes:

- Multi-master minions mode no longer route fileclient operations asymetrically. This fixes the source of many multi-master bugs where the minion would become unrepsonsive from one or more masters.
- Fix bug wherein network.iface could produce stack traces.
- net.arp will no longer be made available unless arp is installed on the system.
- Major performance improvements to Saltnado
- Allow KVM module to operate under KVM itself or VMware Fusion
- Various fixes to the Windows installation scripts
- Fix issue where the syndic would not correctly propagate loads to the master job cache.
- Improve error handling on invalid /etc/network/interfaces file in salt networking modules
- Fix bug where a response status was not checked for in fileclient.get_url
- Enable eauth when running salt in batch mode
- Increase timeout in Boto Route53 module
- Fix bugs with Salt's 'tar' module option parsing
- Fix parsing of NTP servers on Windows
- Fix issue with blockdev tuning not reporting changes correctly
- Update to the latest Salt bootstrap script
- Update Linode salt-cloud driver to use either linode-python or apache-libcloud
- Fix for s3.query function to return correct headers
- Fix for s3.head returning None for files that exist
- Fix the disable function in win_service module so that the service is disabled correctly
- Fix race condition between master and minion when making a directory when both daemons are on the same host
- Fix an issue where file.recurse would fail at the root of an svn repo when the repo has a mountpoint
- Fix an issue where file.recurse would fail at the root of an hgfs repo when the repo has a mountpoint
- Fix an issue where file.recurse would fail at the root of an gitfs repo when the repo has a mountpoint
- Add status.master capability for Windows.
- Various fixes to ssh_known_hosts
- Various fixes to states.network bonding for Debian
- The debian_ip.get_interfaces module no longer removes nameservers.
- Better integration between grains.virtual and systemd-detect-virt and virt-what
- Fix traceback in sysctl.present state output
- Fix for issue where mount.mounted would fail when superopts were not a part of mount.active (extended=True). Also mount.mounted various fixes for Solaris and FreeBSD.
- Fix error where datetimes were not correctly safeguarded before being passed into msgpack.
- Fix file.replace regressions. If the pattern is not found, and if dry run is False, and if *backup* is False, and if a pre-existing file exists with extension *.bak*, then that backup file will be overwritten. This backup behavior is a result of how *fileinput* works. Fixing it requires either passing through the file twice (the first time only to search for content and set a flag), or rewriting *file.replace* so it doesn't use *fileinput*

- VCS filreserver fixes/optimizations
- Catch fileserver configuration errors on master start
- Raise errors on invalid gitfs configurations
- set_locale when locale file does not exist (Redhat family)
- Fix to correctly count active devices when created mdadm array with spares
- Fix to correctly target minions in batch mode
- Support ssh:// urls using the gitfs dulwhich backend
- New fileserver runner
- Fix various bugs with argument parsing to the publish module.
- Fix disk.usage for Synology OS
- Fix issue with tags occurring twice with docker.pulled
- Fix incorrect key error in SMTP returner
- Fix condition which would remount loopback filesystems on every state run
- Remove requsites from listens after they are called in the state system
- Make system implementation of service.running aware of legacy service calls
- Fix issue where publish.publish would not handle duplicate responses gracefully.
- Accept Kali Linux for aptpkg salt execution module
- Fix bug where cmd.which could not handle a dirname as an argument
- Fix issue in ps.pgrep where exceptions were thrown on Windows.

Known issues:

- In multimaster mode, a minion may become temporarily unresponsive if modules or pillars are refreshed at the same time that one or more masters are down. This can be worked around by setting 'auth_timeout' and 'auth_tries' down to shorter periods.

28.2.81 Salt 2014.7.4 Release Notes

release 2015-03-30

Version 2014.7.4 is a bugfix release for *2014.7.0*.

This is a security release. The security issues fixed have only been present since 2014.7.0, and only users of the two listed modules are vulnerable. The following CVEs have been resolved:

- CVE-2015-1838 SaltStack: insecure /tmp file handling in salt/modules/serverdensity_device.py
- CVE-2015-1839 SaltStack: insecure /tmp file handling in salt/modules/chef.py

Changes:

- Multi-master minions mode no longer route fileclient operations asymetrically. This fixes the source of many multi-master bugs where the minion would become unrepsonsive from one or more masters.
- Fix bug wherein network.iface could produce stack traces.
- net.arp will no longer be made available unless arp is installed on the system.
- Major performance improvements to Saltnado

- Allow KVM module to operate under KVM itself or VMware Fusion
- Various fixes to the Windows installation scripts
- Fix issue where the syndic would not correctly propagate loads to the master job cache.
- Improve error handling on invalid /etc/network/interfaces file in salt networking modules
- Fix bug where a response status was not checked for in fileclient.get_url
- Enable eauth when running salt in batch mode
- Increase timeout in Boto Route53 module
- Fix bugs with Salt's 'tar' module option parsing
- Fix parsing of NTP servers on Windows
- Fix issue with blockdev tuning not reporting changes correctly
- Update to the latest Salt bootstrap script
- Update Linode salt-cloud driver to use either linode-python or apache-libcloud
- Fix for s3.query function to return correct headers
- Fix for s3.head returning None for files that exist
- Fix the disable function in win_service module so that the service is disabled correctly
- Fix race condition between master and minion when making a directory when both daemons are on the same host
- Fix an issue where file.recurse would fail at the root of an svn repo when the repo has a mountpoint
- Fix an issue where file.recurse would fail at the root of an hgfs repo when the repo has a mountpoint
- Fix an issue where file.recurse would fail at the root of an gitfs repo when the repo has a mountpoint
- Add status.master capability for Windows.
- Various fixes to ssh_known_hosts
- Various fixes to states.network bonding for Debian
- The debian_ip.get_interfaces module no longer removes nameservers.
- Better integration between grains.virtual and systemd-detect-virt and virt-what
- Fix traceback in sysctl.present state output
- Fix for issue where mount.mounted would fail when superopts were not a part of mount.active (extended=True). Also mount.mounted various fixes for Solaris and FreeBSD.
- Fix error where datetimes were not correctly safeguarded before being passed into msgpack.
- Fix file.replace regressions. If the pattern is not found, and if dry run is False, and if *backup* is False, and if a pre-existing file exists with extension *.bak*, then that backup file will be overwritten. This backup behavior is a result of how *fileinput* works. Fixing it requires either passing through the file twice (the first time only to search for content and set a flag), or rewriting *file.replace* so it doesn't use *fileinput*
- VCS filreserver fixes/optimizations
- Catch fileserver configuration errors on master start
- Raise errors on invalid gitfs configurations
- set_locale when locale file does not exist (Redhat family)
- Fix to correctly count active devices when created mdadm array with spares

- Fix to correctly target minions in batch mode
- Support ssh:// urls using the gitfs dulwhich backend
- New fileserver runner
- Fix various bugs with argument parsing to the publish module.
- Fix disk.usage for Synology OS
- Fix issue with tags occurring twice with docker.pulled
- Fix incorrect key error in SMTP returner
- Fix condition which would remount loopback filesystems on every state run
- Remove requsites from listens after they are called in the state system
- Make system implementation of service.running aware of legacy service calls
- Fix issue where publish.publish would not handle duplicate responses gracefully.
- Accept Kali Linux for aptpkg salt execution module
- Fix bug where cmd.which could not handle a dirname as an argument
- Fix issue in ps.pgrep where exceptions were thrown on Windows.

Known issues:

- In multimaster mode, a minion may become temporarily unresponsive if modules or pillars are refreshed at the same time that one or more masters are down. This can be worked around by setting 'auth_timeout' and 'auth_tries' down to shorter periods.
- There are known issues with batch mode operating on the incorrect number of minions. This bug can be patched with the change in Pull Request #22464.
- The *fun*, *state*, and *unless* keywords are missing from the state internals, which can cause problems running some states. This bug can be patched with the change in Pull Request #22365.

28.2.82 Salt 2014.7.5 Release Notes

release 2015-04-16

Version 2014.7.5 is a bugfix release for *2014.7.0*.

Changes:

- Fixed a key error bug in salt-cloud
- Updated man pages to better match documentation
- Fixed bug concerning high CPU usage with salt-ssh
- Fixed bugs with remounting cvfs and fuse filesystems
- Fixed bug with alowing requisite tracking of entire sls files
- Fixed bug with aptpkg.mod_repo returning OK even if apt-add-repository fails
- Increased frequency of ssh terminal output checking
- Fixed malformed locale string in localmod module
- Fixed checking of available version of package when accept_keywords were changed
- Fixed bug to make git.latest work with empty repositories

- Added **kwargs to service.mod_watch which removes warnings about *enable* and *__reqs__* not being supported by the function
- Improved state comments to not grow so quickly on failed requisites
- Added force argument to service to trigger force_reload
- Fixed bug to andle pkgrepo keyids that have been converted to int
- Fixed module.portage_config bug with appending accept_keywords
- Fixed bug to correctly report disk usage on windows minion
- Added the ability to specify key prefix for S3 ext_pillar
- Fixed issues with batch mode operating on the incorrect number of minions
- Fixed a bug with the proxmox cloud provider stacktracing on disk definition
- Fixed a bug with the changes dictionary in the file state
- Fixed the TCP keep alive settings to work better with SREQ caching
- Fixed many bugs within the iptables state and module
- Fixed bug with states by adding *fun*, *state*, and *unless* to the state runtime internal keywords listing
- Added ability to eAuth against Active Directory
- Fixed some salt-ssh issues when running on Fedora 21
- Fixed grains.get_or_set_hash to work with multiple entries under same key
- Added better explanations and more examples of how the Reactor calls functions to docs
- Fixed bug to not pass *ex_config_drive* to libcloud unless it's explicitly enabled
- Fixed bug with pip.install on windows
- Fixed bug where puppet.run always returns a 0 retcode
- Fixed race condition bug with minion scheduling via pillar
- Made efficiency improvements and bug fixes to the windows installer
- Updated environment variables to fix bug with pygit2 when running salt as non-root user
- Fixed cas behavior on data module – data.cas was not saving changes
- Fixed GPG rendering error
- Fixed strace error in virt.query
- Fixed stacktrace when running chef-solo command
- Fixed possible bug wherein uncaught exceptions seem to make zmq3 tip over when threading is involved
- Fixed argument passing to the reactor
- Fixed glibc caching to prevent bug where salt-minion getaddrinfo in dns_check() never got updated nameservers

Known issues:

- In multimaster mode, a minion may become temporarily unresponsive if modules or pillars are refreshed at the same time that one or more masters are down. This can be worked around by setting 'auth_timeout' and 'auth_tries' down to shorter periods.

28.2.83 Salt 2014.7.6 Release Notes

release 2015-05-18

Version 2014.7.6 is a bugfix release for *2014.7.0*.

Statistics

- Total Merges: **122**
- Total Issue References: **66**
- Total PR References: **166**
- Contributors: **49** (0xf10e, Azidburn, F30, JaseFace, JohannesEbke, aletourneau, aneeshusa, basepi, bastichelaar, bersace, cachedout, cedwards, cellscape, chris-prince, clan, clinta, cr1st1p, cro, dr4Ke, ericfode, ether42, gareth-greenaway, gtmanfred, hvnsweeting, jfindlay, jleroy, joejulian, justinta, kaithar, lorengordon, martinhoefling, mguegan, multani, notpeter, panticz, rallytime, rominf, rubic, s0undt3ch, skizunov, slinu3d, t0rrant, techhat, teizz, terminalmage, thatch45, twangboy, vdesjardins, vr-jack)

Security Fix

CVE-2015-4017 Certificates are not verified when connecting to server in the Aliyun and Proxmox modules

Only users of the Aliyun or Proxmox cloud modules are at risk. The vulnerability does not exist in the latest 2015.5.0 release of Salt.

Changelog for v2014.7.5..v2014.7.6

Generated at: 2018-05-27 20:42:49 UTC

- **PR** #23810: (rallytime) Backport #23757 to 2014.7 @ *2015-05-18 15:30:21 UTC*
  - **PR** #23757: (clan) use abspath, do not eliminating symlinks (refs: #23810)
  - aee00c83df Merge pull request #23810 from rallytime/bp-23757
  - fb32c32065 use abspath, do not eliminating symlinks
- **ISSUE** #20198: (jcftang) virt.get_graphics, virt.get_nics are broken, in turn breaking other things (refs: #23809, #21469)
- **PR** #23809: (rallytime) Fix virtualport section of virt.get_nics loop @ *2015-05-18 15:30:09 UTC*
  - **PR** #21487: (rallytime) Backport #21469 to 2014.7 (refs: #23809)
  - **PR** #21469: (vdesjardins) fixes #20198: virt.get_graphics and virt.get_nics calls in module virt (refs: #21487)
  - 6b3352bb1a Merge pull request #23809 from rallytime/virt_get_nics_fix
  - 0616fb7884 Fix virtualport section of virt.get_nics loop
- **PR** #23823: (gtmanfred) add link local for ipv6 @ *2015-05-17 12:48:25 UTC*
  - 188f03f567 Merge pull request #23823 from gtmanfred/2014.7
  - 5ef006d59d add link local for ipv6
- **PR** #23802: (gtmanfred) if it is ipv6 ip_to_int will fail @ *2015-05-16 04:06:59 UTC*
  - **PR** #23573: (techhat) Scan all available networks for public and private IPs (refs: #23802)

  - f3ca682f92 Merge pull request #23802 from gtmanfred/2014.7
  - 2da98b58c8 if it is ipv6 ip_to_int will fail
- **PR** #23488: (cellscape) LXC cloud fixes @ *2015-05-15 18:09:35 UTC*
  - d9af0c3e82 Merge pull request #23488 from cellscape/lxc-cloud-fixes
  - 64250a67e5 Remove profile from opts after creating LXC container
  - c4047d2a71 Set destroy=True in opts when destroying cloud instance
  - 9e1311a7cd Store instance names in opts when performing cloud action
  - 934bc57c73 Correctly pass custom env to lxc-attach
  - 7fb85f7be1 Preserve test=True option in cloud states
  - 9771b5a313 Fix detection of absent LXC container in cloud state
  - fb24f0cf02 Report failure when failed to create/clone LXC container
  - 2d9aa2bb97 Avoid shadowing variables in lxc module
  - 792e1021f2 Allow to override profile options in lxc.cloud_init_interface
  - 42bd64b9b3 Return changes on successful lxc.create from salt-cloud
  - 4409eabb83 Return correct result when creating cloud LXC container
  - 377015c881 Issue #16424: List all providers when creating salt-cloud instance without profile
- **ISSUE** #22332: (rallytime) [salt-ssh] Add a check for host in /etc/salt/roster (refs: #23748)
- **PR** #23748: (basepi) [2014.7] Log salt-ssh roster render errors more assertively and verbosely @ *2015-05-14 22:38:10 UTC*
  - 808bbe1cb2 Merge pull request #23748 from basepi/salt-ssh.roster.host.check
  - bc53e049e0 Log entire exception for render errors in roster
  - 753de6a621 Log render errors in roster to error level
  - e01a7a90b3 Always let the real YAML error through
- **ISSUE** #22959: (highlyunavailable) Windows Salt hangs if file.directory is trying to write to a drive that doesn't exist (refs: #23731)
- **PR** #23731: (twangboy) Fixes #22959: Trying to add a directory to an unmapped drive in windows @ *2015-05-14 21:59:14 UTC*
  - 72cf360255 Merge pull request #23731 from twangboy/fix_22959
  - 88e5495b2d Fixes #22959: Trying to add a directory to an unmapped drive in windows
- **PR** #23730: (rallytime) Backport #23729 to 2014.7 @ *2015-05-14 21:58:34 UTC*
  - **PR** #23729: (rallytime) Partially merge #23437 (grains fix) (refs: #23730)
  - **PR** #23437: (cedwards) Grains item patch (refs: #23729)
  - 2610195262 Merge pull request #23730 from rallytime/bp-23729
  - 1877caecba adding support for nested grains to grains.item
- **PR** #23688: (twangboy) Added inet_pton to utils/validate/net.py for ip.set_static_ip in windows @ *2015-05-14 16:15:56 UTC*
  - 3e9df883d6 Merge pull request #23688 from twangboy/fix_23415

  - 6a91169bae Fixed unused-import pylint error
  - 5e25b3f355 fixed pylint errors
  - 1a9676626f Added inet_pton to utils/validate/net.py for ip.set_static_ip in windows
- **ISSUE** #23403: (iamfil) salt.runners.cloud.action fun parameter is replaced (refs: #23680)
- **PR** #23680: (cachedout) Rename kwarg in cloud runner @ *2015-05-13 19:44:02 UTC*
  - 1b86460d73 Merge pull request #23680 from cachedout/issue_23403
  - d5986c21b4 Rename kwarg in cloud runner
- **ISSUE** #23548: (kkaig) grains.list_present produces incorrect (?) output (refs: #23674)
- **PR** #23674: (cachedout) Handle lists correctly in grains.list_prsesent @ *2015-05-13 18:34:58 UTC*
  - cd64af0ce4 Merge pull request #23674 from cachedout/issue_23548
  - da8a2f5cb3 Handle lists correctly in grains.list_prsesent
- **PR** #23672: (twangboy) Fix user present @ *2015-05-13 18:30:09 UTC*
  - d322a19213 Merge pull request #23672 from twangboy/fix_user_present
  - 731e7af3dd Merge branch '2014.7' of https://github.com/saltstack/salt into fix_user_present
  - d6f70a4545 Fixed user.present to create password in windows
- **ISSUE** #23604: (Azidburn) service.dead on systemd Minion create an Error Message (refs: #23607)
- **PR** #23670: (rallytime) Backport #23607 to 2014.7 @ *2015-05-13 18:27:17 UTC*
  - **PR** #23607: (Azidburn) Fix for #23604. No error reporting. Exitcode !=0 are ok (refs: #23670)
  - 43f7025000 Merge pull request #23670 from rallytime/bp-23607
  - ed30dc4642 Fix for #23604. No error reporting. Exitcode !=0 are ok
- **ISSUE** #22141: (Deshke) grains.get_or_set_hash render error if hash begins with "%" (refs: #23640)
- **PR** #23661: (rallytime) Merge #23640 with whitespace fix @ *2015-05-13 15:47:30 UTC*
  - **PR** #23640: (cachedout) Add warning to get_or_set_hash about reserved chars (refs: #23661)
  - 0f006ac1d8 Merge pull request #23661 from rallytime/merge-23640
  - 4427f42bb6 Whitespace fix
  - dd9115466e Add warning to get_or_set_hash about reserved chars
- **ISSUE** #23452: (landergate) minion crashed with empty grain (refs: #23639)
- **PR** #23639: (cachedout) Handle exceptions raised by __virtual__ @ *2015-05-13 15:11:12 UTC*
  - 84e2ef88fc Merge pull request #23639 from cachedout/issue_23452
  - d418b49a77 Syntax error!
  - 45b4015d7d Handle exceptions raised by __virtual__
- **ISSUE** #23611: (hubez) master_type set to 'failover' but 'master' is not of type list but of type <type 'str'> (refs: #23637)
- **PR** #23637: (cachedout) Convert str master to list @ *2015-05-13 15:08:19 UTC*
  - bd9b94ba8c Merge pull request #23637 from cachedout/issue_23611
  - 56cb1f52e3 Fix typo

  - f6fcf19a7f Convert str master to list
- **PR** #23595: (rallytime) Backport #23549 to 2014.7 @ *2015-05-12 21:19:40 UTC*
  - **PR** #23549: (vr-jack) Update __init__.py (refs: #23595)
  - f20c0e42ce Merge pull request #23595 from rallytime/bp-23549
  - 6efcac09ad Update __init__.py
- **ISSUE** #23110: (martinhoefling) Copying files from gitfs in file.recurse state fails (refs: #23496)
- **PR** #23594: (rallytime) Backport #23496 to 2014.7 @ *2015-05-12 21:19:34 UTC*
  - **PR** #23496: (martinhoefling) Fix for issue #23110 (refs: #23594)
  - 1acaf86da7 Merge pull request #23594 from rallytime/bp-23496
  - d5ae1d268a Fix for issue #23110 This resolves issues when the freshly created directory is removed by fileserver.update.
- **PR** #23593: (rallytime) Backport #23442 to 2014.7 @ *2015-05-12 21:19:26 UTC*
  - **PR** #23442: (clan) add directory itself to keep list (refs: #23593)
  - 2c221c7332 Merge pull request #23593 from rallytime/bp-23442
  - 39869a15bd check w/ low['name'] only
  - 304cc499e9 another fix for file defined w/ id, but require name
  - 8814d4180e add directory itself to keep list
- **PR** #23606: (twangboy) Fixed checkbox for starting service and actually starting it @ *2015-05-12 21:18:50 UTC*
  - fadd1ef63c Merge pull request #23606 from twangboy/fix_installer
  - 038331edab Fixed checkbox for starting service and actually starting it
- **ISSUE** #22908: (karanjad) Add failhard option to salt orchestration (refs: #23389)
- **PR** #23592: (rallytime) Backport #23389 to 2014.7 @ *2015-05-12 16:44:42 UTC*
  - **PR** #23389: (cachedout) Correct fail_hard typo (refs: #23592)
  - 10b3f0f643 Merge pull request #23592 from rallytime/bp-23389
  - 734cc43801 Correct fail_hard typo
- **PR** #23573: (techhat) Scan all available networks for public and private IPs (refs: #23802) @ *2015-05-12 15:22:22 UTC*
  - cd34b9b6c4 Merge pull request #23573 from techhat/novaquery
  - f92db5e92f Linting
  - 26e00d3ccc Scan all available networks for public and private IPs
- **ISSUE** #23479: (danielmorlock) Typo in pkg.removed for Gentoo? (refs: #23558)
- **PR** #23558: (jfindlay) reorder emerge command line @ *2015-05-12 15:17:46 UTC*
  - 2a72cd71c2 Merge pull request #23558 from jfindlay/fix_ebuild
  - 45404fb2a6 reorder emerge command line
- **ISSUE** #23355: (dr4Ke) salt-ssh: 'sources: salt://' files from 'pkg' state are not included in salt_state.tgz (refs: #23530)
- **PR** #23530: (dr4Ke) salt-ssh state: fix including all salt:// references @ *2015-05-12 15:13:43 UTC*

  - a664a3c6fd Merge pull request #23530 from dr4Ke/fix_salt-ssh_to_include_pkg_sources
  - 5df6a8008c fix pylint warning
  - d0549e56ba salt-ssh state: fix including all salt:// references
- **ISSUE** #23004: (b18) 2014.7.5 - Windows - pkg.list_pkgs - "nxlog" never shows up in output. (refs: #23433)
- **PR** #23433: (twangboy) Obtain all software from the registry @ *2015-05-11 22:47:52 UTC*
  - 55c3869861 Merge pull request #23433 from twangboy/list_pkgs_fix
  - 8ab5b1b86f Fix pylint error
  - 2d11d6545e Obtain all software from the registry
- **PR** #23554: (jleroy) Debian: Hostname always updated @ *2015-05-11 21:57:00 UTC*
  - 755bed0abd Merge pull request #23554 from jleroy/debian-hostname-fix
  - 5ff749e487 Debian: Hostname always updated
- **ISSUE** #23411: (dr4Ke) grains.append should work at any level of a grain (refs: #23440, #23474)
- **PR** #23551: (dr4Ke) grains.append unit tests, related to #23474 @ *2015-05-11 21:54:25 UTC*
  - **PR** #23474: (dr4Ke) Fix grains.append in nested dictionnary grains #23411 (refs: #23551)
  - **PR** #23440: (dr4Ke) fix grains.append in nested dictionnary grains #23411 (refs: #23474)
  - 6ec87ce9f5 Merge pull request #23551 from dr4Ke/grains.append_unit_tests
  - ebff9df5b2 fix pylint errors
  - c4954046ad unit tests for grains.append module function
  - 0c9a32326c use MagickMock
  - c838a22377 unit tests for grains.append module function
- **ISSUE** #23411: (dr4Ke) grains.append should work at any level of a grain (refs: #23440, #23474)
- **PR** #23474: (dr4Ke) Fix grains.append in nested dictionnary grains #23411 (refs: #23551) @ *2015-05-11 18:00:21 UTC*
  - **PR** #23440: (dr4Ke) fix grains.append in nested dictionnary grains #23411 (refs: #23474)
  - e96c5c5bf3 Merge pull request #23474 from dr4Ke/fix_grains.append_nested
  - a01a5bb51e grains.get, parameter delimititer, versionadded: 2014.7.6
  - b39f50475d remove debugging output
  - b6e15e295c fix grains.append in nested dictionnary grains #23411
- **PR** #23537: (t0rrant) Update changelog @ *2015-05-11 17:02:16 UTC*
  - ab7e1aed8e Merge pull request #23537 from t0rrant/patch-1
  - 8e03cc99d3 Update changelog
- **PR** #23538: (cro) Update date in LICENSE file @ *2015-05-11 15:19:25 UTC*
  - b79fed3a92 Merge pull request #23538 from cro/licupdate
  - 345efe25c9 Update date in LICENSE file
- **ISSUE** #23159: (aneeshusa) Unused validator (refs: #23505)
- **PR** #23505: (aneeshusa) Remove unused ssh config validator. Fixes #23159. @ *2015-05-09 13:24:15 UTC*

  - a123a36f05 Merge pull request #23505 from aneeshusa/remove-unused-ssh-config-validator
  - 90af1672ca Remove unused ssh config validator. Fixes #23159.
- **ISSUE** #20518: (ekle) module s3.get does not support eu-central-1 (refs: #23467)
- **PR** #23467: (slinu3d) Added AWS v4 signature support @ *2015-05-08 14:36:19 UTC*
  - ca2c21a63c Merge pull request #23467 from slinu3d/2014.7
  - 0b4081d8f4 Fixed pylint error at line 363
  - 5be5eb5b14 Fixed pylink errors
  - e64f374ffa Fixed lint errors
  - b9d1ac4f1f Added AWS v4 signature support
- **PR** #23444: (techhat) Add create_attach_volume to nova driver @ *2015-05-07 19:51:32 UTC*
  - e6f9eec02e Merge pull request #23444 from techhat/novacreateattach
  - ebdb7eae2d Add create_attach_volume to nova driver
- **ISSUE** #529: (rubic) run salt in user space (refs: #543)
  - **PR** saltstack/salt-bootstrap#563: (notpeter) Ubuntu alternate ppas (refs: #23460)
  - **PR** #543: (rubic) updated documentation for user, fixed configuration template links (refs: #‘saltstack/salt-bootstrap#563‘_)
- **PR** #23460: (s0undt3ch) [2014.7] Update to latest stable bootstrap script v2015.05.07 @ *2015-05-07 19:10:54 UTC*
  - e331463319 Merge pull request #23460 from s0undt3ch/hotfix/bootstrap-script-2014.7
  - edcd0c41f2 Update to latest stable bootstrap script v2015.05.07
- **PR** #23439: (techhat) Add wait_for_passwd_maxtries variable @ *2015-05-07 07:28:56 UTC*
  - 7a8ce1a954 Merge pull request #23439 from techhat/maxtries
  - 0ad3ff2c88 Add wait_for_passwd_maxtries variable
- **PR** #23422: (cro) $HOME should not be used, some shells don’t set it. @ *2015-05-06 21:02:36 UTC*
  - 644eb75fec Merge pull request #23422 from cro/gce_sh_home
  - 4ef9e6ba06 Don’t use $HOME to find user’s directory, some shells don’t set it
- **PR** #23425: (basepi) [2014.7] Fix typo in FunctionWrapper @ *2015-05-06 20:38:03 UTC*
  - ef17ab4b2a Merge pull request #23425 from basepi/functionwrapper_typo
  - c390737f3e Fix typo in FunctionWrapper
- **PR** #23385: (rallytime) Backport #23346 to 2014.7 @ *2015-05-06 20:12:29 UTC*
  - **PR** #23346: (ericfode) Allow file_map in salt-cloud to handle folders. (refs: #23385)
  - 1b13ec04c2 Merge pull request #23385 from rallytime/bp-23346
  - 9efc13c810 more linting fixes
  - cf131c9a5a cleaned up some pylint errors
  - f981699c75 added logic to sftp_file and file_map to allow folder uploads using file_map
- **PR** #23414: (jfindlay) 2015.2 -> 2015.5 @ *2015-05-06 20:04:02 UTC*
  - f8c7a62089 Merge pull request #23414 from jfindlay/update_branch

  - 8074d16d52 2015.2 -> 2015.5
- **PR** #23404: (hvnsweeting) saltapi cherrypy: initialize var when POST body is empty @ *2015-05-06 17:35:56 UTC*
  - 54b3bd43e4 Merge pull request #23404 from hvnsweeting/cherrypy-post-emptybody-fix
  - f85f8f954c initialize var when POST body is empty
- **PR** #23409: (terminalmage) Update Lithium docstrings in 2014.7 branch @ *2015-05-06 16:20:46 UTC*
  - 160f703296 Merge pull request #23409 from terminalmage/update-lithium-docstrings-2014.7
  - bc97d011ba Fix sphinx typo
  - 20006b06f6 Update Lithium docstrings in 2014.7 branch
- **ISSUE** #17245: (tomashavlas) localemod does not generate locale for Arch (refs: #23397, #23307)
- **PR** #23397: (jfindlay) add more flexible whitespace to locale_gen search @ *2015-05-06 03:44:11 UTC*
  - aa5fb0aa46 Merge pull request #23397 from jfindlay/fix_locale_gen
  - 0941fefd2b add more flexible whitespace to locale_gen search
- **PR** #23368: (kaithar) Backport #23367 to 2014.7 @ *2015-05-05 21:42:26 UTC*
  - **PR** #23367: (kaithar) Put the sed insert statement back in to the output. (refs: #23368)
  - **PR** #18368: (basepi) Merge forward from 2014.7 to develop (refs: #23368, #23367)
  - 0c76dd4d8a Merge pull request #23368 from kaithar/bp-23367
  - 577f41972e Pylint fix
  - 8d9acd1f89 Put the sed insert statement back in to the output.
- **ISSUE** #23294: (variia) file.replace fails to append if repl string partially available (refs: #23350)
- **PR** #23350: (lorengordon) Append/prepend: search for full line @ *2015-05-05 21:42:11 UTC*
  - 3493cc1fca Merge pull request #23350 from lorengordon/file.replace_assume_line
  - b60e224beb Append/prepend: search for full line
- **ISSUE** #23026: (adelcast) Incorrect salt-syndic logfile and pidfile locations (refs: #23341)
- **PR** #23341: (cachedout) Fix syndic pid and logfile path @ *2015-05-05 21:29:10 UTC*
  - 7be5c48ad5 Merge pull request #23341 from cachedout/issue_23026
  - e98e65e787 Fix tests
  - 6011b437ca Fix syndic pid and logfile path
- **ISSUE** #19114: (pykler) salt-ssh and gpg pillar renderer (refs: #23347, #23272, #23188)
- **PR** #23272: (basepi) [2014.7] Allow salt-ssh minion config overrides via master config and roster (refs: #23347) @ *2015-05-05 21:28:47 UTC*
  - **PR** #23188: (basepi) [2014.7] Work around bug in salt-ssh in config.get for gpg renderer (refs: #23272)
  - ea61abfa68 Merge pull request #23272 from basepi/salt-ssh.minion.config.19114
  - c223309bb7 Add versionadded
  - be7407feae Lint
  - c2c337567e Missing comma
  - 8e3e8e073a Pass the minion_opts through the FunctionWrapper

  - cb69cd07de Match the master config template in the master config reference
  - 87fc3161f9 Add Salt-SSH section to master config template
  - 91dd9dcbdc Add ssh_minion_opts to master config ref
  - c273ea14c6 Add minion config to salt-ssh doc
  - a0b6b760c3 Add minion_opts to roster docs
  - 5212c35260 Accept minion_opts from the target information
  - e2099b6e1b Process *ssh_minion_opts* from master config
  - 3b64214377 Revert "Work around bug in salt-ssh in config.get for gpg renderer"
  - 494953a208 Remove the strip (embracing multi-line YAML dump)
  - fe87f0fe39 Dump multi-line yaml into the SHIM
  - b751a7281c Inject local minion config into shim if available
- **ISSUE** #19114: (pykler) salt-ssh and gpg pillar renderer (refs: #23347, #23272, #23188)
- **PR** #23347: (basepi) [2014.7] Salt-SSH Backport FunctionWrapper.__contains__ @ *2015-05-05 14:13:21 UTC*
  - **PR** #23272: (basepi) [2014.7] Allow salt-ssh minion config overrides via master config and roster (refs: #23347)
  - **PR** #23188: (basepi) [2014.7] Work around bug in salt-ssh in config.get for gpg renderer (refs: #23272)
  - 4f760dd9cb Merge pull request #23347 from basepi/salt-ssh.functionwrapper.contains.19114
  - 30595e3ff7 Backport FunctionWrapper.__contains__
- **ISSUE** #22742: (hvnsweeting) salt-master says: "This master address: 'salt' was previously resolvable but now fails to resolve!" (refs: #23344)
- **PR** #23344: (cachedout) Explicitely set file_client on master @ *2015-05-04 23:21:48 UTC*
  - 02658b1e60 Merge pull request #23344 from cachedout/issue_22742
  - 5adc96ce7f Explicitely set file_client on master
- **PR** #23318: (cellscape) Honor seed argument in LXC container initializaton @ *2015-05-04 20:58:12 UTC*
  - **PR** #23311: (cellscape) Fix new container initialization in LXC runner (refs: #23318)
  - ba7605d1cb Merge pull request #23318 from cellscape/honor-seed-argument
  - 228b1be299 Honor seed argument in LXC container initializaton
- **ISSUE** #17245: (tomashavlas) localemod does not generate locale for Arch (refs: #23397, #23307)
- **PR** #23307: (jfindlay) check for /etc/locale.gen @ *2015-05-04 20:56:32 UTC*
  - 4ac4509c57 Merge pull request #23307 from jfindlay/fix_locale_gen
  - 101199ac14 check for /etc/locale.gen
- **ISSUE** saltstack/salt-bootstrap#580: (bradthurber) git develop broken in centos6/rhel6/others? due to missing python tornado dep (refs: #23324)
- **ISSUE** saltstack/salt-bootstrap#560: (bradthurber) param to avoid git install on CentOS/RHEL? (refs: #23324)
- **ISSUE** #552: (jhutchins) Support require and watch under the same state dec (refs: #23324)
  - **PR** saltstack/salt-bootstrap#589: (panticz) Fix Debian Squeeze backports mirror (refs: #23324)
  - **PR** saltstack/salt-bootstrap#504: (rominf) opensuse 13.2: fix installation (refs: #23324)

  - **PR** #567: (bastichelaar) Added upstart module (refs: #23324)
- **PR** #23324: (s0undt3ch) [2014.7] Update to the latest stable release of the bootstrap script v2015.05.04 @ *2015-05-04 16:28:30 UTC*
  - f790f42ed6 Merge pull request #23324 from s0undt3ch/hotfix/bootstrap-script-2014.7
  - 6643e47ce5 Update to the latest stable release of the bootstrap script v2015.05.04
- **PR** #23329: (cro) Require requests to verify cert when talking to aliyun and proxmox cloud providers @ *2015-05-04 16:18:17 UTC*
  - 5487367baa Merge pull request #23329 from cro/cloud_verify_cert
  - 860d4b7338 Turn on ssl verify for requests.
- **PR** #23311: (cellscape) Fix new container initialization in LXC runner (refs: #23318) @ *2015-05-04 09:55:29 UTC*
  - ea2017672d Merge pull request #23311 from cellscape/fix-salt-cloud-lxc-init
  - 76fbb34e7d Fix new container initialization in LXC runner
- **ISSUE** #18880: (johtso) npm installed breaks when a module is missing (refs: #23298)
- **PR** #23298: (chris-prince) Fixed issue #18880 in 2014.7 branch @ *2015-05-03 15:49:41 UTC*
  - c399b8f568 Merge pull request #23298 from chris-prince/2014.7
  - 0fa25dbb58 Fixed issue #18880 in 2014.7 branch
- **ISSUE** #23148: (cr1st1p) virt - error handling bogus if machine image location is wrong (refs: #23151)
- **PR** #23292: (rallytime) Merge #23151 with pylint fixes @ *2015-05-02 03:54:12 UTC*
  - **PR** #23151: (cr1st1p) Fixes #23148 (refs: #23292)
  - 16ecefd466 Merge pull request #23292 from rallytime/merge-23151
  - 8ff852a23a Merge #23151 with pylint fixes
  - 8ffa12e82d Fixes #23148
- **PR** #23274: (basepi) [2014.7] Reduce salt-ssh debug log verbosity @ *2015-05-01 20:19:23 UTC*
  - ce24315a4b Merge pull request #23274 from basepi/salt-ssh.debug.verbosity
  - ecee6c68f4 Log stdout and stderr to trace
  - 08f54d79c6 Log stdout and stderr to trace as well
  - 9b9c30f5ad Reduce salt-ssh debug log verbosity
- **ISSUE** #22605: (mavenAtHouzz) Tornado websockets event Handlers registration are incorrect (refs: #23261)
- **PR** #23261: (rallytime) Fix tornado websocket event handler registration @ *2015-05-01 18:20:31 UTC*
  - 7b55e4310f Merge pull request #23261 from rallytime/fix-22605
  - 4950fbf2b3 Fix tornado websocket event handler registration
- **PR** #23258: (teizz) TCP keepalives on the ret side, Revisited. @ *2015-05-01 16:13:49 UTC*
  - 83ef7cb114 Merge pull request #23258 from teizz/ret_keepalive_2014_7_5
  - 0b9fb6f9be The fixes by cachedout which were backported into 2015_2 were missing a single parameter thus not setting up the TCP keepalive for the ZeroMQ Channel by default.
- **ISSUE** #23224: (twellspring) iptables.append –log parameters must be after –jump LOG (refs: #23241)
- **PR** #23241: (techhat) Move iptables log options after the jump @ *2015-05-01 01:31:59 UTC*

  - 8de3c83956 Merge pull request #23241 from techhat/issue23224
  - 87f7948c99 Move iptables log options after the jump
- **PR** #23228: (rallytime) Backport #23171 to 2014.7 @ *2015-04-30 21:09:45 UTC*
  - **PR** #23171: (skizunov) Bugfix: 'clean_proc_dir' is broken (refs: #23228)
  - f20210e499 Merge pull request #23228 from rallytime/bp-23171
  - e670e99506 Bugfix: 'clean_proc_dir' is broken
- **ISSUE** #22703: (Xiol) salt-ssh does not work with list matcher (refs: #22808)
- **PR** #23227: (rallytime) Backport #22808 to 2014.7 @ *2015-04-30 21:09:14 UTC*
  - **PR** #22808: (basepi) [2015.2] Add list targeting to salt-ssh flat roster (refs: #23227)
  - 721cc285ee Merge pull request #23227 from rallytime/bp-22808
  - d208a00b2a Dict, not list
  - a3f529e003 It's already been converted to a list
  - dd57f2d1c1 Add list targeting to salt-ssh flat roster
- **PR** #22823: (hvnsweeting) 22822 file directory clean @ *2015-04-30 15:25:51 UTC*
  - 82c22afacc Merge pull request #22823 from hvnsweeting/22822-file-directory-clean
  - c749c276b4 fix lint - remove unnecessary parenthesis
  - cb3dfee969 refactor
  - 8924b5a911 refactor: use relpath instead of do it manually
  - d3060a51a3 refactor
  - 5759a0e8f0 bugfix: fix file.directory clean=True when it require parent dir
- **ISSUE** saltstack/salt#22941: (bersace) *_pillar* func breaks fileserver globals (refs: #22942)
- **ISSUE** #22941: (bersace) *_pillar* func breaks fileserver globals (refs: #22977)
- **PR** #22977: (bersace) Fix fileserver backends __opts__ overwritten by _pillar @ *2015-04-30 15:24:56 UTC*
  - **PR** #22942: (bersace) Fix fileserver backends global overwritten by _pillar (refs: #22977)
  - f6c0728bfb Merge pull request #22977 from bersace/fix-fileserver-backends-pillar-side-effect
  - 5f451f63cf Fix fileserver backends __opts__ overwritten by _pillar
- **ISSUE** #23166: (claudiupopescu) "Error in function _minion_event" resulting in modules not loaded (refs: #23180)
- **PR** #23180: (jfindlay) fix typos from 36841bdd in masterapi.py @ *2015-04-30 15:22:41 UTC*
  - 34206f7ae3 Merge pull request #23180 from jfindlay/remote_event
  - 72066e1073 fix typos from 36841bdd in masterapi.py
- **ISSUE** #23153: (cr1st1p) cmdmod : run_chroot - broken in 2014.7.5 - missing kwargs (refs: #23176)
- **PR** #23176: (jfindlay) copy standard cmd.run* kwargs into cmd.run_chroot @ *2015-04-30 15:22:12 UTC*
  - b6b82165c8 Merge pull request #23176 from jfindlay/run_chroot
  - 7dc3417b44 copy standard cmd.run* kwargs into cmd.run_chroot
- **ISSUE** #23192: (joejulian) supervisord mod_watch does not accept sfun (refs: #23193)

- **PR** #23193: (joejulian) supervisord.mod_watch should accept sfun @ *2015-04-30 04:34:21 UTC*
    - effacbe294 Merge pull request #23193 from joejulian/2014.7_supervisord_accept_sfun
    - efb59f9d9d supervisord.mod_watch should accept sfun
- **ISSUE** #19114: (pykler) salt-ssh and gpg pillar renderer (refs: #23347, #23272, #23188)
- **PR** #23188: (basepi) [2014.7] Work around bug in salt-ssh in config.get for gpg renderer (refs: #23272) @ *2015-04-30 04:34:10 UTC*
    - 72fe88e5c6 Merge pull request #23188 from basepi/salt-ssh.function.wrapper.gpg.19114
    - d73979ee12 Work around bug in salt-ssh in config.get for gpg renderer
- **ISSUE** #21480: (msciciel) TypeError: string indices must be integers, not str (refs: #23154)
- **PR** #23154: (cachedout) Re-establish channel on interruption in fileclient @ *2015-04-29 16:18:59 UTC*
    - 168508ec2a Merge pull request #23154 from cachedout/refresh_channel
    - 9f8dd80c38 Re-establish channel on interruption in fileclient
- **ISSUE** #20647: (ryan-lane) file.serialize fails to serialize due to ordered dicts (refs: #20779)
- **PR** #23146: (rallytime) Backport #20779 to 2014.7 @ *2015-04-28 20:45:06 UTC*
    - **PR** #20779: (cachedout) Use declared yaml options (refs: #23146)
    - 3b53e04534 Merge pull request #23146 from rallytime/bp-20779
    - ffd18493e8 compare OrderedDicts in serializer unit test
    - a22170627c Just change serialize
    - a111798e8e Use declared yaml options
- **PR** #23145: (rallytime) Backport #23089 to 2014.7 @ *2015-04-28 20:44:56 UTC*
    - **PR** #23089: (cachedout) Stringify version number before lstrip (refs: #23145)
    - 8bb4664bf9 Merge pull request #23145 from rallytime/bp-23089
    - 93c41afd23 Stringify version number before lstrip
- **ISSUE** #16188: (drawks) salt.modules.parted has various functions with bogus input validation. (refs: #23124)
- **PR** #23144: (rallytime) Backport #23124 to 2014.7 @ *2015-04-28 20:44:46 UTC*
    - **PR** #23124: (ether42) fix parsing the output of parted in parted.list_() (refs: #23144)
    - c85d36fd29 Merge pull request #23144 from rallytime/bp-23124-2014-7
    - 6b64da706c fix parsing the output of parted
- **PR** #23120: (terminalmage) Don't run os.path.relpath() if repo doesn't have a "root" param set @ *2015-04-28 15:46:54 UTC*
    - a27b158153 Merge pull request #23120 from terminalmage/fix-gitfs-relpath
    - 1860fffd68 Don't run os.path.relpath() if repo doesn't have a "root" param set
- **PR** #23132: (clinta) Backport b27c176 @ *2015-04-28 15:00:30 UTC*
    - fcba607978 Merge pull request #23132 from clinta/patch-2
    - a824d727d1 Backport b27c176
- **ISSUE** #18476: (Auha) Upgrading salt on my master caused dependency issues (refs: #18610, #23114)

- **PR** #23114: (rallytime) Adjust ZeroMQ 4 docs to reflect changes to Ubuntu 12 packages @ *2015-04-28 03:59:24 UTC*
  - **PR** #18610: (rallytime) Make ZMQ 4 installation docs for ubuntu more clear (refs: #23114)
  - b0f4b28487 Merge pull request #23114 from rallytime/remove_ubuntu_zmq4_docs
  - f6cc7c8f8a Adjust ZeroMQ 4 docs to reflect changes to Ubuntu 12 packages
- **ISSUE** #23085: (xenophonf) Use "s3fs" (not "s3") in fileserver_roots (refs: #23097)
- **PR** #23108: (rallytime) Backport #23097 to 2014.7 @ *2015-04-28 03:58:05 UTC*
  - **PR** #23097: (rallytime) Change s3 to s3fs in fileserver_roots docs example (refs: #23108)
  - 399857f20b Merge pull request #23108 from rallytime/bp-23097
  - fa889845df Change s3 to s3fs in fileserver_roots docs example
- **ISSUE** #22171: (basepi) We should only call returner.save_load once per jid (refs: #22199)
- **PR** #23112: (basepi) [2014.7] Backport #22199 to fix mysql returner save_load errors @ *2015-04-28 03:55:44 UTC*
  - **PR** #22199: (basepi) [2015.2] Put a bandaid on the save_load duplicate issue (mysql returner) (refs: #23112)
  - 5541537c32 Merge pull request #23112 from basepi/mysql_returner_save_load
  - 0127012ed3 Put a bandaid on the save_load duplicate issue
  - **PR** saltstack/salt#22925: (rallytime) Backport #22895 to 2014.7 (refs: #23113)
- **PR** #23113: (rallytime) Revert "Backport #22895 to 2014.7" @ *2015-04-28 03:27:29 UTC*
  - **PR** #22895: (aletourneau) pam_tally counter was not reset to 0 after a succesfull login (refs: #23113, #22925, #saltstack/salt`#22925`_)
  - dfe2066b25 Merge pull request #23113 from saltstack/revert-22925-bp-22895
  - b957ea8977 Revert "Backport #22895 to 2014.7"
- **ISSUE** #23013: (ghost) gitfs regression with authenticated repos (refs: #23094)
- **PR** #23094: (terminalmage) pygit2: disable cleaning of stale refs for authenticated remotes @ *2015-04-27 20:51:28 UTC*
  - 21515f3c23 Merge pull request #23094 from terminalmage/issue23013
  - aaf7b04f79 pygit2: disable cleaning of stale refs for authenticated remotes
- **PR** #23048: (jfindlay) py-2.6 compat for utils/boto.py ElementTree exception @ *2015-04-25 16:56:45 UTC*
  - d45aa21dca Merge pull request #23048 from jfindlay/ET_error
  - 64c42ccb5f py-2.6 compat for utils/boto.py ElementTree exception
- **ISSUE** #22981: (syphernl) Locale state throwing traceback when generating not (yet) existing locale (refs: #23025)
- **PR** #23025: (jfindlay) catch exceptions on bad system locales/encodings @ *2015-04-25 16:56:30 UTC*
  - d25a5c102f Merge pull request #23025 from jfindlay/fix_sys_locale
  - 9c4d62bb00 catch exceptions on bad system locales/encodings
- **PR** #22932: (hvnsweeting) bugfix: also manipulate dir_mode when source not defined @ *2015-04-25 16:54:58 UTC*
  - 5e44b59a14 Merge pull request #22932 from hvnsweeting/file-append-bugfix

  - 3f368de14a do not use assert in execution module
  - 9d4fd4a8c8 bugfix: also manipulate dir_mode when source not defined
- **ISSUE** #23021: (ether42) ps.pgrep raises NoSuchProcess (refs: #23055)
- **PR** #23055: (jfindlay) prevent ps module errors on accessing dead procs @ *2015-04-24 22:39:49 UTC*
  - c2416a425f Merge pull request #23055 from jfindlay/fix_ps
  - c2dc7adeb1 prevent ps module errors on accessing dead procs
- **PR** #23031: (jfindlay) convert exception e.message to just e @ *2015-04-24 18:38:13 UTC*
  - bfd9158a83 Merge pull request #23031 from jfindlay/exception
  - 856bad1c31 convert exception e.message to just e
- **PR** #23015: (hvnsweeting) if status of service is stop, there is not an error with it @ *2015-04-24 14:35:10 UTC*
  - 7747f3342e Merge pull request #23015 from hvnsweeting/set-non-error-lvl-for-service-status-log
  - 92ea163513 if status of service is stop, there is not an error with it
- **ISSUE** #22993: (jetpak) salt-minion restart causes all spawned daemons to die on centos7 (systemd) (refs: #23000)
- **PR** #23000: (jfindlay) set systemd service killMode to process for minion @ *2015-04-24 03:42:39 UTC*
  - 2e09789156 Merge pull request #23000 from jfindlay/systemd_kill
  - 3d575e29c4 set systemd service killMode to process for minion
- **ISSUE** #22707: (arthurlogilab) retry_dns of master configuration is missing from the documentation (refs: #22999)
- **PR** #22999: (justinta) Added retry_dns to minion doc. @ *2015-04-24 03:30:24 UTC*
  - b5c059ab26 Merge pull request #22999 from jtand/fix_22707
  - 8486e17ab3 Added retry_dns to minion doc.
- **PR** #22990: (techhat) Use the proper cloud conf variable @ *2015-04-23 17:48:07 UTC*
  - 27dc877bfd Merge pull request #22990 from techhat/2014.7
  - d33bcbc2c1 Use the proper cloud conf variable
- **PR** #22976: (multani) Improve state_output documentation @ *2015-04-23 12:24:22 UTC*
  - 13dff652c6 Merge pull request #22976 from multani/fix/state-output-doc
  - 19efd419b5 Improve state_output documentation
- **PR** #22955: (terminalmage) Fix regression introduced yesterday in dockerio module @ *2015-04-22 18:56:39 UTC*
  - 89fa18500c Merge pull request #22955 from terminalmage/dockerio-run-fix
  - b4472ad1b2 Fix regression introduced yesterday in dockerio module
- **PR** #22954: (rallytime) Backport #22909 to 2014.7 @ *2015-04-22 18:56:20 UTC*
  - **PR** #22909: (mguegan) Fix compatibility with pkgin > 0.7 (refs: #22954)
  - 46ef227911 Merge pull request #22954 from rallytime/bp-22909
  - 70c1cd3969 Fix compatibility with pkgin > 0.7
- **ISSUE** #18720: (Reiner030) timeouts when setting Route53 records (refs: #22856)
- **PR** #22856: (jfindlay) increase timeout and decrease tries for route53 records @ *2015-04-22 16:47:01 UTC*

  - c9ae593461 Merge pull request #22856 from jfindlay/route53_timeout
  - ba4a786984 add route53 record sync wait, default=False
  - ea2fd50660 increase timeout and tries for route53 records
- **PR** #22946: (s0undt3ch) Test with a more recent pip version to avoid a traceback @ *2015-04-22 16:25:17 UTC*
  - a178d444b8 Merge pull request #22946 from s0undt3ch/2014.7
  - bc87749e2c Test with a more recent pip version to avoid a traceback
- **ISSUE** #22571: (BoomerB) same error message as on issue #18504 (refs: #22945)
- **PR** #22945: (garethgreenaway) Fixes to scheduler @ *2015-04-22 16:25:00 UTC*
  - de339bef0a Merge pull request #22945 from garethgreenaway/22571_2014_7_schedule_pillar_refresh_seconds_exceptions
  - bfa6d25ed8 Fixing a reported issue when using a scheduled job from pillar with splay. _seconds element that acted as a backup of the actual seconds was being removed when pillar was refreshed and causing exceptions. This fix moves some splay related code out of the if else condition so it's checked whether the job is in the job queue or not.
- **ISSUE** #18843: (calvinhp) State user.present will fail to create home if user exists and homedir doesn't (refs: #22933, #22887)
- **PR** #22887: (hvnsweeting) fix #18843 @ *2015-04-22 15:47:05 UTC*
  - 12d2b91d85 Merge pull request #22887 from hvnsweeting/18843-fix-user-present-home
  - 7fe7b089fd run user.chhome once to avoid any side-effect when run it twice
  - 19de9954ee clarify the usage of home arg
  - d6dc09af64 enhance doc, as usermod on ubuntu 12.04 will not CREATE home
  - 0ce4d7feb6 refactor: force to use boolean
  - 849d19edd7 log debug the creating dir process
  - c4e95b9f48 fix #18843: usermod won't create a dir if old home does not exist
- **ISSUE** #2417: (ffa) Module standards (refs: #22829)
- **ISSUE** #21140: (holms) locale.present state executed successfully, although originally fails (refs: #22930, #22829)
- **PR** #22930: (jfindlay) localemod.gen_locale now always returns a boolean @ *2015-04-22 15:37:39 UTC*
  - **PR** #22829: (F30) Always return a boolean in gen_locale() (refs: #22930)
  - b7de7bdf47 Merge pull request #22930 from jfindlay/localegen_bool
  - 399399f89e localemod.gen_locale now always returns a boolean
- **ISSUE** #18843: (calvinhp) State user.present will fail to create home if user exists and homedir doesn't (refs: #22933, #22887)
- **PR** #22933: (hvnsweeting) add test for #18843 @ *2015-04-22 15:27:18 UTC*
  - 11bcf14979 Merge pull request #22933 from hvnsweeting/18843-test
  - b13db32fde add test for #18843
- **PR** #22925: (rallytime) Backport #22895 to 2014.7 @ *2015-04-22 02:30:26 UTC*
  - **PR** #22895: (aletourneau) pam_tally counter was not reset to 0 after a succesfull login (refs: #23113, #22925, #saltstack/salt`#22925`_)

  - 6890752dd3 Merge pull request #22925 from rallytime/bp-22895
  - 3852d96213 Pylint fix
  - 90f7829ad3 Fixed pylint issues
  - 5ebf159554 Cleaned up pull request
  - a08ac478f6 pam_tally counter was not reset to 0 after a succesfull login
- **ISSUE** #22790: (whiteinge) jobs.list_jobs runner tracebacks on 'missing' argument (refs: #22914)
- **PR** #22914: (cachedout) Call proper returner function in jobs.list_jobs @ *2015-04-22 00:49:01 UTC*
  - eca37ebc11 Merge pull request #22914 from cachedout/issue_22790
  - d828d6fd58 Call proper returner function in jobs.list_jobs
- **PR** #22918: (JaseFace) Add a note to the git_pillar docs stating that GitPython is the only currently supported provider @ *2015-04-22 00:48:26 UTC*
  - 44f3409b01 Merge pull request #22918 from JaseFace/git-pillar-provider-doc-note
  - 0aee5c23d4 Add a note to the git_pillar docs stating that GitPython is the only currently supported provider
- **PR** #22907: (techhat) Properly merge cloud configs to create profiles @ *2015-04-21 22:02:44 UTC*
  - 31c461f573 Merge pull request #22907 from techhat/cloudconfig
  - 3bf4e66112 Properly merge cloud configs to create profiles
- **ISSUE** #22782: (0xf10e) Turning everything into OrderedDicts broke states.keystone.user_present() (refs: #22894)
- **PR** #22894: (0xf10e) Fix issue #22782 @ *2015-04-21 18:55:18 UTC*
  - f0939754a0 Merge pull request #22894 from 0xf10e/2014.7
  - 58fa24c7fa Clarify doc on kwarg 'roles' for user_present().
  - f0ae2eb84f Improve readability by renaming tenant_role
- **ISSUE** #12003: (MarkusMuellerAU) [state.dockerio] docker.run TypeError: run() argument after ** must be a mapping, not str (refs: #22902)
- **PR** #22902: (rallytime) Change state example to use proper kwarg @ *2015-04-21 18:50:47 UTC*
  - c802ba7514 Merge pull request #22902 from rallytime/docker_doc_fix
  - 8f703461b0 Change state example to use proper kwarg
- **PR** #22898: (terminalmage) dockerio: better error message for native exec driver @ *2015-04-21 18:02:58 UTC*
  - 81771a7769 Merge pull request #22898 from terminalmage/issue12003
  - c375309434 dockerio: better error message for native exec driver
- **ISSUE** #22825: (paolodina) Issue using file.replace in state file (refs: #22897)
- **PR** #22897: (rallytime) Add param documentation for file.replace state @ *2015-04-21 17:31:04 UTC*
  - e2ec4ecc55 Merge pull request #22897 from rallytime/fix-22825
  - 9c51630002 Add param documentation for file.replace state
- **ISSUE** saltstack/salt#22844: (bersace) LocalClient file cache confuse pillar and state files (refs: #22850)
- **PR** #22850: (bersace) Fix pillar and salt fileserver mixed @ *2015-04-21 17:04:33 UTC*

  - fd53889f0e Merge pull request #22850 from bersace/fix-pillar-salt-mixed
  - 31b98e72eb Initialize state file client after pillar loading
  - f6bebb7a31 Use saltenv
- **PR** #22818: (twangboy) Added documentation regarding pip in windows @ *2015-04-21 03:58:59 UTC*
  - 1380fec1b9 Merge pull request #22818 from twangboy/upd_pip_docs
  - cb999c7d70 Update pip.py
  - 3cc5c970ad Added documentation regarding pip in windows
- **PR** #22872: (rallytime) Prevent stacktrace on os.path.exists in hosts module @ *2015-04-21 02:54:40 UTC*
  - b2bf17f5d5 Merge pull request #22872 from rallytime/fix_hosts_stacktrace
  - c88a1ea243 Prevent stacktrace on os.path.exists in hosts module
- **PR** #22853: (s0undt3ch) Don't assume package installation order. @ *2015-04-21 02:42:41 UTC*
  - 03af523de9 Merge pull request #22853 from s0undt3ch/2014.7
  - b62df62151 Don't assume package installation order.
- **PR** #22877: (s0undt3ch) Don't fail on *make clean* just because the directory does not exist @ *2015-04-21 02:40:47 UTC*
  - 9211e36564 Merge pull request #22877 from s0undt3ch/hotfix/clean-docs-fix
  - 95d6887949 Don't fail on *make clean* just because the directory does not exist
- **PR** #22873: (thatch45) Type check the version since it will often be numeric @ *2015-04-21 02:38:11 UTC*
  - 5bdbd08bbd Merge pull request #22873 from thatch45/type_check
  - 53b8376626 Type check the version since it will often be numeric
- **PR** #22870: (twangboy) Added ability to send a version with a space in it @ *2015-04-20 23:18:28 UTC*
  - c965b0a035 Merge pull request #22870 from twangboy/fix_installer_again
  - 3f180cfaae Added ability to send a version with a space in it
- **PR** #22863: (rallytime) Backport #20974 to 2014.7 @ *2015-04-20 19:29:37 UTC*
  - **PR** #20974: (JohannesEbke) Fix expr_match usage in salt.utils.check_whitelist_blacklist (refs: #22863)
  - 2973eb18bc Merge pull request #22863 from rallytime/bp-20974
  - 14913a4cb4 Fix expr_match usage in salt.utils.check_whitelist_blacklist
- **PR** #22578: (hvnsweeting) gracefully handle when salt-minion cannot decrypt key @ *2015-04-20 15:24:45 UTC*
  - c45b92bb4b Merge pull request #22578 from hvnsweeting/2014-7-fix-compile-pillar
  - f75b24ad68 gracefully handle when salt-minion cannot decrypt key
- **ISSUE** #21979: (yrdevops) gitfs: error message not descriptive enough when libgit2 was compiled without libssh2 (refs: #22800)
- **PR** #22800: (terminalmage) Improve error logging for pygit2 SSH-based remotes @ *2015-04-18 17:18:55 UTC*
  - 900c7a510f Merge pull request #22800 from terminalmage/issue21979
  - 8f1c0084cd Clarify that for pygit2, receiving 0 objects means repo is up-to-date
  - 98885f71d6 Add information about libssh2 requirement for pygit2 ssh auth
  - 09468d2607 Fix incorrect log message

  - 2093bf8d96 Adjust loglevels for gitfs errors
  - 9d394dfe46 Improve error logging for pygit2 SSH-based remotes
- **PR** #22813: (twangboy) Updated instructions for building salt @ *2015-04-18 04:10:07 UTC*
  - e99f2fdb28 Merge pull request #22813 from twangboy/win_doc_fix
  - adc421acdd Fixed some formatting issues
  - 8901b3b5a6 Updated instructions for building salt
- **ISSUE** #22708: (Bilge) salt-ssh file.accumulated error: NameError: global name 'msgpack' is not defined (refs: #22810)
- **PR** #22810: (basepi) [2014.7] More msgpack gating for salt-ssh @ *2015-04-17 22:28:24 UTC*
  - fe1de89ad7 Merge pull request #22810 from basepi/salt-ssh.more.msgpack.gating
  - d4da8e66a4 Gate msgpack in salt/modules/saltutil.py
  - 02303b22ce Gate msgpack in salt/modules/data.py
  - d7e8741f02 Gate salt.states.file.py msgpack
- **ISSUE** #17144: (xpender) salt-cloud -m fails with softlayer (refs: #22803)
- **PR** #22803: (rallytime) Allow map file to work with softlayer @ *2015-04-17 20:34:42 UTC*
  - 11df71e16d Merge pull request #22803 from rallytime/fix-17144
  - ce88b6ad41 Allow map file to work with softlayer
- **PR** #22807: (rallytime) Add 2014.7.5 links to windows installation docs @ *2015-04-17 20:32:13 UTC*
  - cd43a95212 Merge pull request #22807 from rallytime/windows_docs_update
  - 5931a582d1 Replace all 4s with 5s
  - eadaead755 Add 2014.7.5 links to windows installation docs
- **PR** #22795: (rallytime) Added release note for 2014.7.5 release @ *2015-04-17 18:05:36 UTC*
  - 0b295e2c87 Merge pull request #22795 from rallytime/release_notes
  - fde1feed46 Remove extra line
  - b19b95d992 Added release note for 2014.7.5 release
- **ISSUE** #22740: (lorengordon) New Windows installer assumes salt is installed to the current directory (refs: #22759)
- **PR** #22759: (twangboy) Final edits to the batch files for running salt @ *2015-04-17 04:31:15 UTC*
  - **PR** #22754: (twangboy) Removed redundant \\ and " (refs: #22759)
  - 3c91459de2 Merge pull request #22759 from twangboy/fix_bat_one_last_time
  - 075f82e046 Final edits to the batch files for running salt
- **PR** #22760: (thatch45) Fix issues with the syndic @ *2015-04-17 04:30:48 UTC*
  - 20d3f2bb83 Merge pull request #22760 from thatch45/syndic_fix
  - e2db624b37 Fix issues with the syndic not resolving the master when the interface is set
- **PR** #22762: (twangboy) Fixed version not showing in Add/Remove Programs @ *2015-04-17 04:29:46 UTC*
  - 54c45845ab Merge pull request #22762 from twangboy/fix_installer
  - 4d25af8acf Fixed version not showing in Add/Remove Programs

28.2.84 Salt 2014.7.7 Release Notes

release 2015-10-13

Version 2014.7.7 is a bugfix release for *2014.7.0*.

Statistics

- Total Merges: **54**
- Total Issue References: **20**
- Total PR References: **60**
- Contributors: **28** (AkhterAli, BretFisher, MrCitron, alekti, basepi, bersace, cachedout, corux, cro, davidjb, dumol, efficks, garethgreenaway, hvnsweeting, jacksontj, jacobhammons, jaybocc2, jfindlay, jquast, justinta, msteed, nmadhok, notpeter, puneetk, rallytime, techhat, trevor-h, twangboy)

Changelog for v2014.7.6..v2014.7.7

Generated at: 2018-05-27 20:45:04 UTC

- **PR** #27335: (rallytime) [2014.7] Fixup salt-cloud logging @ *2015-09-24 20:33:53 UTC*
  - 5262f01325 Merge pull request #27335 from rallytime/cloud-logging-7
  - adeb1dcad4 Pylint Fix
  - 588c13783c Salt-cloud logging clean up for windows functions
  - 9b6000135c [2014.7] Fixup salt-cloud logging
- **PR** #27252: (jfindlay) 2014.7 -> 2014.7.0 @ *2015-09-18 23:44:39 UTC*
  - e90412d3b8 Merge pull request #27252 from jfindlay/version.2014.7
  - 3d28307a00 2014.7 -> 2014.7.0
- **PR** #27117: (jacobhammons) made 2014.7 an archived release @ *2015-09-15 07:35:12 UTC*
  - c186e51764 Merge pull request #27117 from jacobhammons/release-docs-2014.7
  - b69e11e0a4 made 2014.7 an archived release minor doc site updates
- **PR** #27114: (cachedout) Issue warning that some log levels may contain sensitive data @ *2015-09-15 07:30:43 UTC*
  - 69d758ee2b Merge pull request #27114 from cachedout/warn_on_insecure_log
  - 507fb04683 Issue warning that some log levels may contain sensitive data
- **PR** #27075: (twangboy) Replaced password with redacted when displayed @ *2015-09-14 18:36:10 UTC*
  - aa71bae8aa Merge pull request #27075 from twangboy/fix_password_2014.7
  - c0689e3215 Replaced password with redacted when displayed
- **ISSUE** #26656: (ari) [documentation] error in example for salt.runner.pillar (refs: #26667)
- **PR** #26667: (nmadhok) [doc-fix] Removing special character from salt.runners.pillar and other changes @ *2015-08-26 18:24:37 UTC*
  - c2c7fe06c8 Merge pull request #26667 from nmadhok/doc-fix-2014.7
  - 26be189689 Doc fix. Fixes #26656

- **PR** #26663: (jacobhammons) version change for latest branch @ *2015-08-26 14:03:35 UTC*
    - 6bd3dccae8 Merge pull request #26663 from jacobhammons/2014.7-version
    - b6af538070 version change for latest branch
- **PR** #26636: (rallytime) Refactor cloud provider tests to be more accurate @ *2015-08-25 21:28:34 UTC*
    - 071a6112e5 Merge pull request #26636 from rallytime/cloud-test-fixes
    - c0d83d558d Don't use id as variable
    - 2b4bc1679d Keep ec2 instance creation test the same - it works better for the ec2 output
    - b5b58eb31f Skip digital ocean tests since we can't use API v1 with v2 tests
    - 9ae1539c62 Update cloud tests to be more efficient and accurate
- **ISSUE** #26630: (efficks) win_service: Function has_powershell does not works on Windows XP (refs: #26640)
- **PR** #26640: (efficks) Fix function spacing @ *2015-08-25 20:01:39 UTC*
    - 304542b4c6 Merge pull request #26640 from efficks/fixws2014
    - ebe5d9d85c Fix function spacing
- **PR** #26515: (bersace) Defaults to current saltenv in state.sls @ *2015-08-25 16:35:50 UTC*
    - 4532f98a76 Merge pull request #26515 from bersace/salt-env-local-sls
    - 0727af9e3d Defaults to current saltenv in state.sls
- **PR** #26242: (cro) Remove dead code @ *2015-08-12 15:14:20 UTC*
    - da8bca09aa Merge pull request #26242 from cro/anonldap4
    - a0d2ab1eed Remove dead code
- **PR** #26216: (cro) Fix LDAP configuration issue. @ *2015-08-11 18:33:43 UTC*
    - 1ecf23773e Merge pull request #26216 from cro/anonldap3
    - af132d7b89 Documentation update for anonymous bind issue.
    - 2ef54b6b13 Documentation update for anonymous bind issue.
    - 5b1836bb00 Fix issue with LDAP anonymous binds.
- **PR** #26116: (corux) file.replace fails if repl string is an invalid regex and append/prepend is used @ *2015-08-10 16:44:12 UTC*
    - abdf2935c4 Merge pull request #26116 from corux/fix-escape-content
    - fd913ddc36 Append/prepend: search for full line with escaped content
- **ISSUE** #25751: (basepi) Document *master_finger* more prominently (refs: #26088)
- **PR** #26088: (jacobhammons) Master finger @ *2015-08-07 14:31:33 UTC*
    - 106356d98d Merge pull request #26088 from jacobhammons/master-finger
    - 133d5f7885 some small changes
    - d220c83f77 master_finger configuration docs switch a script to use https:// instead of http:// Refs #25751
- **ISSUE** #25961: (getabc) [2015.5.3-2] salt-winrepo.git/salt-minion.sls fails certificate '*.wpengine.com' or 'wpengine.com' (refs: #26047)
- **PR** #26047: (jacobhammons) Updated windows download links in the docs to https://repo.saltstack.com @ *2015-08-05 22:59:44 UTC*

  - 4bd4bc41f2 Merge pull request #26047 from jacobhammons/win-downloads
  - 7c162d181c Updated windows download links in the docs to https://repo.saltstack.com Refs #25961
- **ISSUE** #25701: (alekti) Issue #23764 regression (refs: #25750)
- **ISSUE** #23764: (es1o) source_hash from local file is not supported. (refs: #25750)
- **PR** #25750: (alekti) Add file as supported protocol for file source_hash. Fixes #25701. @ *2015-07-29 02:31:27 UTC*
  - d93eb87c16 Merge pull request #25750 from alekti/2014.7
  - 9ec3ae96d4 Add file as supported protocol for file source_hash. Fixes #23764.
- **PR** #25704: (cachedout) Ensure prior alignment with master_type in 2014.7 @ *2015-07-27 16:06:35 UTC*
  - 3a15df22ac Merge pull request #25704 from cachedout/master_type_2014_7
  - c95886c9a7 Ensure prior alignment with master_type in 2014.7
- **PR** #25657: (MrCitron) Add the ability to specify a base pattern for carbon returner @ *2015-07-24 16:32:58 UTC*
  - d1b9362a73 Merge pull request #25657 from MrCitron/pattern-carbon-returner-2014.7
  - f8b2f8079f Add the ability to specify a base pattern for metrics path used by the carbon returner
- **PR** #25633: (AkhterAli) Update loader.py @ *2015-07-22 20:02:41 UTC*
  - 9634351fc2 Merge pull request #25633 from AkhterAli/2014.7
  - 29be4bbe11 Update loader.py
- **PR** #25416: (cachedout) Fix broken keyword @ *2015-07-14 19:47:10 UTC*
  - 09ebaceca8 Merge pull request #25416 from cachedout/str_2014_7
  - cc514938a8 Fix broken keyword
- **PR** #25375: (cachedout) Fix error in config.py for master_type @ *2015-07-13 16:49:27 UTC*
  - 2a1dd1113f Merge pull request #25375 from cachedout/config_fix_2014_7
  - c041f2905f Fix error in config.py for master_type
- **PR** #25324: (jacobhammons) Latest help theme updates @ *2015-07-10 16:11:31 UTC*
  - 2590e23d48 Merge pull request #25324 from jacobhammons/doc-theme-updates
  - 88f5fcf58d Latest help theme updates
- **ISSUE** #18447: (ryan-lane) Can't install salt with raet using pip -e git (refs: #25093)
- **PR** #25093: (jaybocc2) quick fix for issue #18447 @ *2015-07-01 15:56:53 UTC*
  - 36d53ef59e Merge pull request #25093 from jaybocc2/2014.7
  - c6a501ebda quick fix for issue #18447
- **PR** #25069: (puneetk) Add a helper module function called list_enabled @ *2015-06-30 20:53:51 UTC*
  - 38903a94a1 Merge pull request #25069 from puneetk/patch-1
  - f0b4e600e6 Update Documentation to clarify version added
  - f8dc6030e7 Pylint updates , removing whitespace
  - 532d315dd1 [Code Review update] renamed function to is_enaled from list_enabled
  - 20b0462289 Update schedule.py

  - 4f1471d7fb Add a helper module function called list_enabled
- **ISSUE** #15209: (hubez) file.manage: source_hash not working with s3:// (2014.7.0rc1) (refs: #25011)
- **PR** #25011: (notpeter) Add s3 to protocols for remote source_hash (2014.7 backport) @ *2015-06-27 22:35:44 UTC*
  - a7154e7471 Merge pull request #25011 from notpeter/s3_2014.7_backport
  - 8b8af640f6 Add s3 to protocols for remote source_hash
- **ISSUE** #24915: (justinta) Salt-cloud not working in 2014.7.6 (refs: #24944)
- **PR** #24944: (techhat) Double-check main_cloud_config @ *2015-06-25 12:29:55 UTC*
  - a11e4c6eea Merge pull request #24944 from techhat/issue24915
  - 59c3081e49 Double-check main_cloud_config
- **PR** #24936: (justinta) Fixed ps module to not use depreciated psutil commands @ *2015-06-24 22:38:19 UTC*
  - d26a5447ba Merge pull request #24936 from jtand/psutil
  - bdb7a19c36 Fixed ps module to not use depreciated psutil commands
- **ISSUE** saltstack/salt-bootstrap#473: (s1kbr0) salt-bootstrap.sh […] git v2014.1.11 on SmartOS base64 is broken (refs: #24918)
- **PR** #24918: (BretFisher) SmartOS SMF minion startup fix @ *2015-06-24 15:44:26 UTC*
  - eeb05a1b10 Merge pull request #24918 from BretFisher/minion-start-smartos-smf-fix
  - d7bfb0c7fd Smartos smf minion fix
- **ISSUE** #24776: (nmadhok) –static option in salt raises ValueError and has been broken for a very long time (refs: #24777)
- **PR** #24780: (nmadhok) Backporting PR #24777 to 2014.7 branch @ *2015-06-18 14:52:56 UTC*
  - **PR** #24779: (nmadhok) Backporting Changes to 2014.7 branch (refs: #24777)
  - **PR** #24778: (nmadhok) Backporting PR #24777 to 2015.2 branch (refs: #24777)
  - **PR** #24777: (nmadhok) Fixing issue where –static option fails with ValueError Fixes #24776 (refs: #24778, #24780)
  - 4281dfff0b Merge pull request #24780 from nmadhok/backport-2014.7-24777
  - c53b0d9a22 Backporting PR #24777 to 2014.7 branch
- **ISSUE** #21318: (thanatos) get_full_returns raises KeyError (refs: #24769)
- **ISSUE** #18994: (njhartwell) salt.client.get_cli_returns errors when called immediately after run_job (refs: #24769)
- **PR** #24769: (msteed) Fix stacktrace in get_cli_returns() @ *2015-06-18 14:31:46 UTC*
  - f3c5cb2d41 Merge pull request #24769 from msteed/issue-21318
  - f40a9d5cc0 Fix stacktrace in get_cli_returns()
- **ISSUE** #17041: (xenophonf) Confusing Salt error messages due to limited/incomplete PowerShell command error handling (refs: #24690)
- **PR** #24690: (twangboy) Report powershell output instead of error @ *2015-06-17 16:33:49 UTC*
  - 59db24602f Merge pull request #24690 from twangboy/fix_17041
  - 7a015389af Added additional reporting
  - d84ad5d519 Fixed capitalization… Failed and Already

  - e9552455c4 Merge branch '2014.7' of https://github.com/saltstack/salt into fix_17041
- **ISSUE** #24196: (johnccfm) Exception when using user.present with Windows (refs: #24646)
- **PR** #24646: (twangboy) Fixed user.present on existing user @ *2015-06-15 15:04:43 UTC*
  - a18dadad71 Merge pull request #24646 from twangboy/fix_24196
  - a208e1d60f Fixed user.present on existing user
  - 144bff2f67 Report powershell output instead of error
- **PR** #24643: (cro) Add reference to salt-announce mailing list @ *2015-06-12 20:21:15 UTC*
  - b99484fde2 Merge pull request #24643 from cro/saltannounce
  - ecb0623d7f Add salt-announce mailing list.
- **PR** #24620: (twangboy) Fixed comment and uncomment functions in file.py @ *2015-06-12 19:36:26 UTC*
  - 635121e85d Merge pull request #24620 from twangboy/fix_24215
  - d7a9999be1 Fixed comment and uncomment functions in file.py
- **PR** #24589: (BretFisher) Fixed Mine example for jinja code block @ *2015-06-11 15:48:02 UTC*
  - d83928a7f9 Merge pull request #24589 from BretFisher/patch-1
  - 65a11336dc Fixed Mine example for jinja code block
- **ISSUE** #24427: (fayetted) 2015.5.1-3 Windows 64Bit Minion fails to start after install (refs: #24530)
- **PR** #24530: (twangboy) Start Minion Service on Silent Install @ *2015-06-09 21:30:08 UTC*
  - d376390f76 Merge pull request #24530 from twangboy/fix_24427
  - 673e1d809e Added missing panel.bmp for installer
  - cc50218b01 Start Minion Service on Silent Install
- **PR** #24513: (jquast) bugfix use of 'iteritem' in 2014.7 branch @ *2015-06-09 04:06:36 UTC*
  - **PR** #24511: (jquast) bugfix: trailing "…done" in rabbitmq output (refs: #24513)
  - 6ebc476bb3 Merge pull request #24513 from jquast/2014.7-bugfix-iteritem
  - 2be0180e5e bugfix use of 'iteritem' in 2014.7 branch
- **ISSUE** #24276: (markuskramerIgitt) Live salt-master Profiling with SIGUSR2 fails (refs: #24405)
- **PR** #24405: (jacksontj) Fix for #24276 @ *2015-06-04 20:50:42 UTC*
  - 83f853b6ea Merge pull request #24405 from jacksontj/2014.7
  - 2c7afaeebf Fix for #24276
- **PR** #24395: (hvnsweeting) handle exceptions when received data is not in good shape @ *2015-06-04 20:08:22 UTC*
  - cef919c602 Merge pull request #24395 from hvnsweeting/handle-exception-get-file
  - bb798a0224 handle exceptions when received data is not in good shape
- **PR** #24305: (twangboy) Added documentation, fixed formatting @ *2015-06-04 19:40:54 UTC*
  - efba1a94b4 Merge pull request #24305 from twangboy/win_path_docs
  - 36804253e6 Fixed pylint error caused by P… added r
  - bc42a4bb11 triple double quotes to triple single quotes

  - 77cd930bba Added documentation, fixed formatting
- **PR** #24178: (rallytime) Backport #24118 to 2014.7, too. @ *2015-05-27 17:49:45 UTC*
  - **PR** #24118: (trevor-h) removed deprecated pymongo usage (refs: #24178)
  - 9d7331c87d Merge pull request #24178 from rallytime/bp-24118
  - e2217a09e8 removed deprecated pymongo usage as no longer functional with pymongo > 3.x
- **PR** #24159: (rallytime) Fill out modules/keystone.py CLI Examples @ *2015-05-27 15:07:11 UTC*
  - 4e8c5031b0 Merge pull request #24159 from rallytime/keystone_doc_examples
  - dadac8d076 Fill out modules/keystone.py CLI Examples
- **PR** #24158: (rallytime) Fix test_valid_docs test for tls module @ *2015-05-27 15:06:05 UTC*
  - fc10ee8ed5 Merge pull request #24158 from rallytime/fix_doc_error
  - 49a517e2ca Fix test_valid_docs test for tls module
- **PR** #24125: (hvnsweeting) Fix rabbitmq test mode @ *2015-05-26 15:40:18 UTC*
  - c0d32e0b5e Merge pull request #24125 from hvnsweeting/fix-rabbitmq-test-mode
  - 71862c69b9 enhance log
  - 28e2594162 change according to new output of rabbitmq module functions
  - cd0212e8ed processes and returns better output for rabbitmq module
- **ISSUE** #23464: (tibold) cmd_iter_no_block() blocks (refs: #24093)
- **PR** #24093: (msteed) Make LocalClient.cmd_iter_no_block() not block @ *2015-05-25 15:56:42 UTC*
  - 39a8f30f06 Merge pull request #24093 from msteed/issue-23464
  - fd35903d75 Fix failing test
  - 41b344c7d3 Make LocalClient.cmd_iter_no_block() not block
- **PR** #24008: (davidjb) Correct reST formatting for states.cmd documentation @ *2015-05-21 04:19:01 UTC*
  - 5bffd3045e Merge pull request #24008 from davidjb/2014.7
  - 8b8d0293d4 Correct reST formatting for documentation
- **PR** #23933: (jacobhammons) sphinx saltstack2 doc theme @ *2015-05-20 18:19:19 UTC*
  - 1aa0420040 Merge pull request #23933 from jacobhammons/2014.7
  - a3613e68e4 removed numbering from doc TOC
  - 78b737c5e6 removed 2015.* release from release notes, updated index page to remove PDF/epub links
  - e867f7df77 Changed build settings to use saltstack2 theme and update release versions.
  - 81ed9c9f59 sphinx saltstack2 doc theme
- **PR** #23965: (hvnsweeting) handle all exceptions gitpython can raise @ *2015-05-20 15:08:03 UTC*
  - 314e4db512 Merge pull request #23965 from hvnsweeting/20147-fix-gitfs-gitpython-exception
  - 2576301631 handle all exception gitpython can raise
- **PR** #23939: (basepi) Add extended changelog to 2014.7.6 release notes @ *2015-05-19 21:21:00 UTC*
  - 913391207a Merge pull request #23939 from basepi/v2014.7.6release
  - 32b65dc2a9 Add extended changelog to 2014.7.6 release notes

- **ISSUE** #23820: (UtahDave) 2014.7.5 schedule error (refs: #23881)
- **PR** #23881: (garethgreenaway) Fixes to schedule module in 2014.7 @ *2015-05-19 15:46:30 UTC*
  - 0031ca2631 Merge pull request #23881 from garethgreenaway/23820_2014_7_schedule_list_issue
  - b207f2a433 Missing continue in the list function when deleting unused attributes.
- **ISSUE** #22131: (quixoten) "unexpected keyword argument 'merge'" on 2014.7.2 (salt-ssh) (refs: #23887)
- **PR** #23887: (basepi) [2014.7] Bring salt-ssh pillar.get in line with mainline pillar.get @ *2015-05-18 23:11:34 UTC*
  - 63bd21ecd2 Merge pull request #23887 from basepi/salt-ssh.pillar.get.22131
  - bc84502f46 Bring salt-ssh pillar.get in line with mainline pillar.get
- **PR** #23891: (basepi) Update the release notes index page @ *2015-05-18 23:06:52 UTC*
  - 17c5810c04 Merge pull request #23891 from basepi/releasenotes
  - dec153bcea Update the release notes index page
- **PR** #23888: (basepi) Update the 2014.7.6 release notes with CVE details @ *2015-05-18 22:35:51 UTC*
  - a93e58f80f Merge pull request #23888 from basepi/v2014.7.6release
  - 49921b6cb2 Update the 2014.7.6 release notes with CVE details
- **PR** #23871: (rallytime) Backport #23848 to 2014.7 @ *2015-05-18 20:34:04 UTC*
  - **PR** #23848: (dumol) Updated installation docs for SLES 12. (refs: #23871)
  - 50730287bb Merge pull request #23871 from rallytime/bp-23848
  - 379c09c3a5 Updated for SLES 12.

28.2.85 Salt 2014.7.8 Release Notes

release 2015-11-13

Version 2014.7.8 is a bugfix release for *2014.7.0*.

Statistics

- Total Merges: **7**
- Total Issue References: **3**
- Total PR References: **10**
- Contributors: **5** (DmitryKuzmenko, JaseFace, MasterNayru, cachedout, rallytime)

Changelog for v2014.7.7..v2014.7.8

Generated at: 2018-05-27 20:47:34 UTC

- **PR** #28839: (cachedout) Revert #28740 @ *2015-11-12 22:54:28 UTC*
  - **PR** #28740: (MasterNayru) Add missing S3 module import (refs: #28777, #28839)
  - 4b8bdd0afb Merge pull request #28839 from cachedout/revert_28740
  - 215b26c06f Revert #28740
- **PR** #28777: (rallytime) Back-port #28740 to 2014.7 @ *2015-11-11 18:00:00 UTC*

  - **PR** #28740: (MasterNayru) Add missing S3 module import (refs: #28777, #28839)
  - 76e69b4bff Merge pull request #28777 from rallytime/bp-28740-2014.7
  - da5fac2b36 Back-port #28740 to 2014.7
- **PR** #28716: (rallytime) Back-port #28705 to 2014.7 @ *2015-11-10 16:15:03 UTC*
  - **PR** #28705: (cachedout) Account for new headers class in tornado 4.3 (refs: #28716)
  - 45c73ebf2f Merge pull request #28716 from rallytime/bp-28705
  - 32e7bd3ea0 Account for new headers class in tornado 4.3
- **ISSUE** #28199: (felskrone) Non-standard umasks might break the master (refs: #28717)
- **PR** #28717: (cachedout) Add note about recommended umask @ *2015-11-09 23:26:20 UTC*
  - f4fe921965 Merge pull request #28717 from cachedout/umask_note
  - 1874300e08 Add note about recommended umask
- **ISSUE** #28455: (zmalone) highstate.cache is world readable, and contains secrets (refs: #28461)
- **PR** #28461: (cachedout) Wrap all cache calls in state.sls in correct umask @ *2015-11-02 17:11:02 UTC*
  - 4bf56cad3f Merge pull request #28461 from cachedout/issue_28455
  - 097838ec0c Wrap all cache calls in state.sls in correct umask
- **ISSUE** #24910: (bocig) -T, –make-token flag does NOT work- LDAP Groups (refs: #28407)
- **PR** #28407: (DmitryKuzmenko) Don't request creds if auth with key. @ *2015-10-29 16:12:30 UTC*
  - f3e61db045 Merge pull request #28407 from DSRCompany/issues/24910_token_auth_fix_2014
  - b7b5bec309 Don't request creds if auth with key.
- **PR** #27390: (JaseFace) Ensure we pass on the enable setting if present, or use the default of True if not in build_schedule_item() @ *2015-10-05 18:09:33 UTC*
  - d284eb165b Merge pull request #27390 from JaseFace/schedule-missing-enabled
  - 563db71bfd Ensure we pass on the enable setting if present, or use the default of True if not in build_schedule_item() Prior to this, when schedule.present compares the existing schedule to the one crafted by this function, enabled will actually be removed at each run. schedule.present sees a modification needs to be made, and invokes schedule.modify, which does so with enabled: True, creating and endless loop of an 'enabled' removal and addition.

28.2.86 Salt 2014.7.9 Release Notes

release 2016-03-11

Version 2014.7.9 is a bugfix release for *2014.7.0*.

Statistics

- Total Merges: **5**
- Total Issue References: **1**
- Total PR References: **5**
- Contributors: **4** (douardda, gtmanfred, jacobhammons, jfindlay)

Changelog for v2014.7.8..v2014.7.9

Generated at: 2018-05-27 20:55:35 UTC

- **PR** #31834: (jfindlay) add 2014.7.8 release notes @ *2016-03-11 21:35:42 UTC*
  - 218c902091 Merge pull request #31834 from jfindlay/2014.7
  - 358fdad0c8 add 2014.7.8 release notes
- **PR** #31833: (jfindlay) add 2014.7.9 release notes @ *2016-03-11 21:19:55 UTC*
  - a423c6cd04 Merge pull request #31833 from jfindlay/2014.7
  - 6910fcc584 add 2014.7.9 release notes
- **PR** #31826: (gtmanfred) Remove ability of authenticating user to specify pam service @ *2016-03-11 20:41:01 UTC*
  - c5e7c03953 Merge pull request #31826 from gtmanfred/2014.7
  - d73f70ebb2 Remove ability of authenticating user to specify pam service
- **PR** #29392: (jacobhammons) updated version number to not reference a specific build from the lat… @ *2015-12-03 15:54:31 UTC*
  - 85aa70a6cb Merge pull request #29392 from jacobhammons/2014.7
  - d7f0db1dd8 updated version number to not reference a specific build from the latest branch
- **ISSUE** #29295: (douardda) systemd's service file should use the 'process' KillMode option on Debian also (refs: #29296)
- **PR** #29296: (douardda) Use process KillMode on Debian systems also @ *2015-12-01 16:00:16 UTC*
  - d2fb2109a3 Merge pull request #29296 from douardda/patch-3
  - d2885390f4 Use process KillMode on Debian systems also

28.2.87 Salt 2014.1.0 Release Notes - Codename Hydrogen

Note: Due to a change in master to minion communication, 2014.1.0 minions are not compatible with older-version masters. Please upgrade masters first. More info on backwards-compatibility policy *here*, under the "Upgrading Salt" subheading.

Note: A change in the grammar in the state compiler makes `module.run` in requisites illegal syntax. Its use is replaced simply with the word `module`. In other words you will need to change requisites like this:

```
require:
    module.run: some_module_name
```

to:

```
require:
    module: some_module_name
```

This is a breaking change. We apologize for the inconvenience, we needed to do this to remove some ambiguity in parsing requisites.

release 2014-02-24

The 2014.1.0 release of Salt is a major release which not only increases stability but also brings new capabilities in virtualization, cloud integration, and more. This release brings a great focus on the expansion of testing making roughly double the coverage in the Salt tests, and comes with many new features.

2014.1.0 is the first release to follow the new date-based release naming system. See the version numbers page for more details.

Major Features

Salt Cloud Merged into Salt

Salt Cloud is a tool for provisioning salted minions across various cloud providers. Prior to this release, Salt Cloud was a separate project but this marks its full integration with the Salt distribution. A Getting Started guide and additional documentation for Salt Cloud can be found *here*:

Google Compute Engine

Alongside Salt Cloud comes new support for the Google Compute Engine. Salt Stack can now deploy and control GCE virtual machines and the application stacks that they run.

For more information on Salt Stack and GCE, please see this blog post.

Documentation for Salt and GCE can be found *here*.

Salt Virt

Salt Virt is a cloud controller that supports virtual machine deployment, inspection, migration, and integration with many aspects of Salt.

Salt Virt has undergone a major overhaul with this release and now supports many more features and includes a number of critical improvements.

Docker Integration

Salt now ships with `states` and an `execution module` to manage Docker containers.

Substantial Testing Expansion

Salt continues to increase its unit/regression test coverage. This release includes over 300 new tests.

BSD Package Management

BSD package management has been entirely rewritten. FreeBSD 9 and older now default to using pkg_add, while FreeBSD 10 and newer will use pkgng. FreeBSD 9 can be forced to use pkgng, however, by specifying the following option in the minion config file:

```
providers:
  pkg: pkgng
```

In addition, support for installing software from the ports tree has been added. See the documentation for the ports *state* and *execution module* for more information.

Network Management for Debian/Ubuntu

Initial support for management of network interfaces on Debian-based distros has been added. See the documentation for the *network state* and the *debian_ip* for more information.

IPv6 Support for iptables State/Module

The iptables *state* and *module* now have IPv6 support. A new parameter `family` has been added to the states and execution functions, to distinguish between IPv4 and IPv6. The default value for this parameter is `ipv4`, specifying `ipv6` will use ip6tables to manage firewall rules.

GitFS Improvements

Several performance improvements have been made to the *Git fileserver backend*. Additionally, file states can now use any SHA1 commit hash as a fileserver environment:

```
/etc/httpd/httpd.conf:
  file.managed:
    - source: salt://webserver/files/httpd.conf
    - saltenv: 45af879
```

This applies to the functions in the *cp module* as well:

```
salt '*' cp.get_file salt://readme.txt /tmp/readme.txt saltenv=45af879
```

MinionFS

This new fileserver backend allows files which have been pushed from the minion to the master (using *cp.push*) to be served up from the salt fileserver. The path for these files takes the following format:

```
salt://minion-id/path/to/file
```

`minion-id` is the id of the ”source” minion, the one from which the files were pushed to the master. `/path/to/file` is the full path of the file.

The *MinionFS Walkthrough* contains a more thorough example of how to use this backend.

saltenv

To distinguish between fileserver environments and execution functions which deal with environment variables, fileserver environments are now specified using the `saltenv` parameter. `env` will continue to work, but is deprecated and will be removed in a future release.

Grains Caching

A caching layer has been added to the Grains system, which can help speed up minion startup. Disabled by default, it can be enabled by setting the minion config option `grains_cache`:

```
grains_cache: True

# Seconds before grains cache is considered to be stale.
grains_cache_expiration: 300
```

If set to `True`, the grains loader will read from/write to a msgpack-serialized file containing the grains data.

Additional command-line parameters have been added to salt-call, mainly for testing purposes:

- `--skip-grains` will completely bypass the grains loader when salt-call is invoked.
- `--refresh-grains-cache` will force the grains loader to bypass the grains cache and refresh the grains, writing a new grains cache file.

Improved Command Logging Control

When using the *`cmd module`*, either on the CLI or when developing Salt execution modules, a new keyword argument `output_loglevel` allows for greater control over how (or even if) the command and its output are logged. For example:

```
salt '*' cmd.run 'tail /var/log/messages' output_loglevel=debug
```

The package management modules (`apt`, `yumpkg`, etc.) have been updated to log the copious output generated from these commands at loglevel `debug`.

Note: To keep a command from being logged, `output_loglevel=quiet` can be used.

Prior to this release, this could be done using `quiet=True`. This argument is still supported, but will be removed in a future Salt release.

PagerDuty Support

Initial support for firing events via PagerDuty has been added. See the documentation for the *`pagerduty`* module.

Virtual Terminal

Sometimes the subprocess module is not good enough, and, in fact, not even `askpass` is. This virtual terminal is still in it's infant childhood, needs quite some love, and was originally created to replace `askpass`, but, while developing it, it immediately proved that it could do so much more. It's currently used by salt-cloud when bootstrapping salt on clouds which require the use of a password.

Proxy Minions

Initial basic support for Proxy Minions is in this release. Documentation can be found *here*.

Proxy minions are a developing feature in Salt that enables control of devices that cannot run a minion. Examples include network gear like switches and routers that run a proprietary OS but offer an API, or ”dumb” devices that just don’t have the horsepower or ability to handle a Python VM.

Proxy minions can be difficult to write, so a simple REST-based example proxy is included. A Python bottle-based webserver can be found at https://github.com/cro/salt-proxy-rest as an endpoint for this proxy.

This is an ALPHA-quality feature. There are a number of issues with it currently, mostly centering around process control, logging, and inability to work in a masterless configuration.

Additional Bugfixes (Release Candidate Period)

Below are many of the fixes that were implemented in salt during the release candidate phase.

- Fix mount.mounted leaving conflicting entries in fstab (issue #7079)
- Fix mysql returner serialization to use json (issue #9590)
- Fix `ZMQError: Operation cannot be accomplished in current state` errors (issue #6306)
- Rbenv and ruby improvements
- Fix quoting issues with mysql port (issue #9568)
- Update mount module/state to support multiple swap partitions (issue #9520)
- Fix `archive` state to work with `bsdtar`
- Clarify logs for minion ID caching
- Add numeric revision support to git state (issue #9718)
- Update `master_uri` with `master_ip` (issue #9694)
- Add comment to Debian `mod_repo` (issue #9923)
- Fix potential undefined loop variable in rabbitmq state (issue #8703)
- Fix for salt-virt runner to delete key on VM deletion
- Fix for `salt-run -d` to limit results to specific runner or function (issue #9975)
- Add tracebacks to jinja renderer when applicable (issue #10010)
- Fix parsing in monit module (issue #10041)
- Fix highstate output from syndic minions (issue #9732)
- Quiet logging when dealing with passwords/hashes (issue #10000)
- Fix for multiple remotes in git_pillar (issue #9932)
- Fix npm installed command (issue #10109)
- Add safeguards for utf8 errors in zcbuildout module
- Fix compound commands (issue #9746)
- Add systemd notification when master is started
- Many doc improvements

28.2.88 Salt 2014.1.1 Release Notes

release 2014-03-18

Version 2014.1.1 is a bugfix release for *2014.1.0*. The changes include:

- Various doc fixes, including up-to-date Salt Cloud installation documentation.
- Renamed state.sls runner to state.orchestrate, to reduce confusion with the *`state.sls`* execution function
- Fix various bugs in the `dig` module (issue #10367)
- Add retry for query on certain EC2 status codes (issue #10154)
- Fix various bugs in mongodb_user state module (issue #10430)
- Fix permissions on `~/.salt_token` (issue #10422)
- Add PyObjects support
- Fix launchctl module crash with missing files
- Fix *`saltutil.find_job`* for Windows (issue #10581)
- Fix OS detection for OpenSolaris (issue #10601)
- Fix broken salt-ssh key_deploy
- Add support for multiline cron comments (issue #10721)
- Fix timezone module for Arch (issue #10789)
- Fix symlink support for `file.recurse` (issue #10809)
- Fix multi-master bugs (issue #10732 and issue #10969)
- Fix file.patch to error when source file is unavailable (issue #10380)
- Fix pkg to handle packages set as `purge` in `pkg.installed` (issue #10719)
- Add `zmqversion` grain
- Fix highstate summary for masterless minions (issue #10945)
- Fix *`saltutil.find_job`* for 2014.1 masters talking to 0.17 minions (issue #11020)
- Fix *`file.recurse`* states with trailing slashes in source (issue #11002)
- Fix *`pkg states`* to allow `pkgname.x86_64` (issue #7306)
- Make *`iptables states`* set a default table for flush (issue #11037)
- Added iptables `--reject-with` after final iptables call in *`iptables states`* (issue:*10757*)
- Fix improper passing of "family" in *`iptables states`* (issue #10774)
- Fix traceback in *`iptables.insert`* states (issue #10988)
- Fix zombie processes (issue #10867 and others)
- Fix batch mode to obey `--return` settings (issue #9146)
- Fix localclient issue that was causing batch mode breakage (issue #11094, issue #10470, and others)
- Multiple salt-ssh fixes
- FreeBSD: look in /usr/local/etc/salt for configuration by default, if installed using `pip --editable`.
- Add a `skip_suggestions` parameter to pkg.installed states which allows pre-flight check to be skipped (issue #11106)

- Fixed tag-based gitfs fileserver environments regression (issue #10956)
- Yum: fix cache of available pkgs not cleared when repos are changed (issue #11001)
- Yum: fix for plugin-provided repositories (i.e. RHN/Spacewalk) (issue #11145)
- Fix regression in *`chocolatey.bootstrap`* (issue #10541)
- Fix fail on unknown target in *`jobs runner`* (issue #11151)
- Don't log errors for commands which are expected to sometimes exit with non-zero exit status (issue #11154, issue #11090)
- Fix `test=True` CLI override of config option (issue #10877)
- Log sysctl key listing at loglevel TRACE (issue #10931)

28.2.89 Salt 2014.1.10 Release Notes

release 2014-08-01

Note: Version 2014.1.9 contained a regression which caused inaccurate Salt version detection, and thus was never packaged for general release. This version contains the version detection fix, but is otherwise identical to 2014.1.9.

Version 2014.1.10 is another bugfix release for *2014.1.0*. Changes include:

- Ensure salt-ssh will not continue if permissions on a temporary directory are not correct.
- Use the bootstrap script distributed with Salt instead of relying on an external resource
- Remove unused testing code
- Ensure salt states are placed into the `.salt` directory in salt-ssh
- Use a randomized path for temporary files in a salt-cloud deployment
- Clean any stale directories to ensure a fresh copy of salt-ssh during a deployment

Salt 2014.1.10 fixes security issues documented by CVE-2014-3563: "Insecure tmp-file creation in seed.py, salt-ssh, and salt-cloud." Upgrading is recommended.

28.2.90 Salt 2014.1.11 Release Notes

release 2014-08-29

Version 2014.1.11 is another bugfix release for *2014.1.0*. Changes include:

- Fix for minion_id with byte-order mark (BOM) (issue #12296)
- Fix `runas` deprecation in `at` module
- Fix trailing slash befhavior for `file.makedirs_` (issue #14019)
- Fix chocolatey path (issue #13870)
- Fix git_pillar infinite loop issues (issue #14671)
- Fix json outputter `null` case
- Fix for minion error if one of multiple masters are down (issue #14099)

28.2.91 Salt 2014.1.12 Release Notes

release 2014-10-08

Version 2014.1.12 is another bugfix release for *2014.1.0*. Changes include:

- Fix `scp_file` always failing (which broke salt-cloud) (issue #16437)
- Fix regression in pillar in masterless (issue #16210, issue #16416, issue #16428)

28.2.92 Salt 2014.1.13 Release Notes

release 2014-10-14

Version 2014.1.13 is another bugfix release for *2014.1.0*. Changes include:

- Fix `sftp_file` by checking the exit status code of scp (which broke salt-cloud) (issue #16599)

28.2.93 Salt 2014.1.2 Release Notes

release 2014-04-15

Version 2014.1.2 is another bugfix release for *2014.1.0*. The changes include:

- Fix username detection when su'ed to root on FreeBSD (issue #11628)
- Fix minionfs backend for file.recurse states
- Fix 32-bit packages of different arches than the CPU arch, on 32-bit RHEL/CentOS (issue #11822)
- Fix bug with specifying alternate home dir on user creation (FreeBSD) (issue #11790)
- Don't reload `site` module on module refresh for MacOS
- Fix regression with running execution functions in Pillar SLS (issue #11453)
- Fix some modules missing from Windows installer
- Don't log an error for yum commands that return nonzero exit status on non-failure (issue #11645)
- Fix bug in rabbitmq state (issue #8703)
- Fix missing ssh config options (issue #10604)
- Fix top.sls ordering (issue #10810 and issue #11691)
- Fix `salt-key --list all` (issue #10982)
- Fix win_servermanager install/remove function (issue #11038)
- Fix interaction with tokens when running commands as root (issue #11223)
- Fix overstate bug with `find_job` and `**kwargs` (issue #10503)
- Fix `saltenv` for `aptpkg.mod_repo` from `pkgrepo` state
- Fix environment issue causing file caching problems (issue #11189)
- Fix bug in `__parse_key` in registry state (issue #11408)
- Add minion auth retry on rejection (issue #10763)
- Fix publish_session updating the encryption key (issue #11493)
- Fix for bad `AssertionError` raised by GitPython (issue #11473)

- Fix `debian_ip` to allow disabling and enabling networking on Ubuntu (issue #11164)
- Fix potential memory leak caused by saved (and unused) events (issue #11582)
- Fix exception handling in the MySQL module (issue #11616)
- Fix environment-related error (issue #11534)
- Include `psutil` on Windows
- Add `file.replace` and `file.search` to Windows (issue #11471)
- Add additional `file` module helpers to Windows (issue #11235)
- Add `pid` to netstat output on Windows (issue #10782)
- Fix Windows not caching new versions of installers in winrepo (issue #10597)
- Fix hardcoded md5 hashing
- Fix kwargs in salt-ssh (issue #11609)
- Fix file backup timestamps (issue #11745)
- Fix stacktrace on `sys.doc` with invalid eauth (issue #11293)
- Fix `git.latest` with `test=True` (issue #11595)
- Fix `file.check_perms` hardcoded `follow_symlinks` (issue #11387)
- Fix certain `pkg` states for RHEL5/Cent5 machines (issue #11719)

28.2.94 Salt 2014.1.3 Release Notes

release 2014-04-15

Version 2014.1.3 is another bugfix release for *2014.1.0*. It was created as a hotfix for a regression found in 2014.1.2, which was not distributed. The only change made was as follows:

- Fix regression that caused `saltutil.find_job` to fail, causing premature terminations of salt CLI commands.

Changes in the not-distributed 2014.1.2, also included in 2014.1.3:

- Fix username detection when su'ed to root on FreeBSD (issue #11628)
- Fix minionfs backend for file.recurse states
- Fix 32-bit packages of different arches than the CPU arch, on 32-bit RHEL/CentOS (issue #11822)
- Fix bug with specifying alternate home dir on user creation (FreeBSD) (issue #11790)
- Don't reload `site` module on module refresh for MacOS
- Fix regression with running execution functions in Pillar SLS (issue #11453)
- Fix some modules missing from Windows installer
- Don't log an error for yum commands that return nonzero exit status on non-failure (issue #11645)
- Fix bug in rabbitmq state (issue #8703)
- Fix missing ssh config options (issue #10604)
- Fix top.sls ordering (issue #10810 and issue #11691)
- Fix `salt-key --list all` (issue #10982)
- Fix win_servermanager install/remove function (issue #11038)

- Fix interaction with tokens when running commands as root (issue #11223)
- Fix overstate bug with `find_job` and `**kwargs` (issue #10503)
- Fix `saltenv` for `aptpkg.mod_repo` from `pkgrepo` state
- Fix environment issue causing file caching problems (issue #11189)
- Fix bug in `__parse_key` in registry state (issue #11408)
- Add minion auth retry on rejection (issue #10763)
- Fix publish_session updating the encryption key (issue #11493)
- Fix for bad `AssertionError` raised by GitPython (issue #11473)
- Fix `debian_ip` to allow disabling and enabling networking on Ubuntu (issue #11164)
- Fix potential memory leak caused by saved (and unused) events (issue #11582)
- Fix exception handling in the MySQL module (issue #11616)
- Fix environment-related error (issue #11534)
- Include `psutil` on Windows
- Add `file.replace` and `file.search` to Windows (issue #11471)
- Add additional `file` module helpers to Windows (issue #11235)
- Add `pid` to netstat output on Windows (issue #10782)
- Fix Windows not caching new versions of installers in winrepo (issue #10597)
- Fix hardcoded md5 hashing
- Fix kwargs in salt-ssh (issue #11609)
- Fix file backup timestamps (issue #11745)
- Fix stacktrace on `sys.doc` with invalid eauth (issue #11293)
- Fix `git.latest` with `test=True` (issue #11595)
- Fix `file.check_perms` hardcoded `follow_symlinks` (issue #11387)
- Fix certain `pkg` states for RHEL5/Cent5 machines (issue #11719)

28.2.95 Salt 2014.1.4 Release Notes

release 2014-05-05

Version 2014.1.4 is another bugfix release for *2014.1.0*. Changes include:

- Fix setup.py dependency issue (issue #12031)
- Fix handling for IOErrors under certain circumstances (issue #11783 and issue #11853)
- Fix fatal exception when `/proc/1/cgroup` is not readable (issue #11619)
- Fix os grains for OpenSolaris (issue #11907)
- Fix `lvs.zero` module argument pass-through (issue #9001)
- Fix bug in `debian_ip` interaction with *`network.system`* state (issue #11164)
- Remove bad binary package verification code (issue #12177)
- Fix traceback in solaris package installation (issue #12237)

- Fix *`file.directory`* state symlink handling (issue #12209)
- Remove `external_ip` grain
- Fix *`file.managed`* makedirs issues (issue #10446)
- Fix hang on non-existent Windows drive letter for *`file`* module (issue #9880)
- Fix salt minion caching all users on the server (issue #9743)
- Add strftime formatting for *`ps.boot_time`* (issue #12428)

28.2.96 Salt 2014.1.5 Release Notes

release 2014-06-11

Version 2014.1.5 is another bugfix release for *2014.1.0*. Changes include:

- Add function for finding cached job on the minion
- Fix iptables save file location for Debian (issue #11730)
- Fix for minion caching jobs when master is down
- Bump default `syndic_wait` to 5 to fix syndic-related problems (issue #12262)
- Add OpenBSD, FreeBSD, and NetBSD support for `network.netstat` (issue #12121)
- Fix false positive error in logs for `makeconf` state (issue #9762)
- Fix for yum `fromrepo` package installs when repo is disabled by default (issue #12466)
- Fix for extra blank lines in `file.blockreplace` (issue #12422)
- Fix grain detection for OpenVZ guests (issue #11877)
- Fix `get_dns_servers` function for Windows `win_dns_client`
- Use system locale for ports package installations
- Use correct stop/restart procedure for Debian networking in `debian_ip` (issue #12614)
- Fix for `cmd_iter`/`cmd_iter_no_block` blocking issues (issue #12617)
- Fix traceback when syncing custom types (issue #12883)
- Fix cleaning directory symlinks in `file.directory`
- Add performance optimizations for `saltutil.sync_all` and `state.highstate`
- Fix possible error in `saltutil.running`
- Fix for kmod modules with dashes (issue #13239)
- Fix possible race condition for Windows minions in state module reloading (issue #12370)
- Fix bug with roster for `passwd` option that is loaded as a non-string object (issue #13249)
- Keep duplicate version numbers from showing up in `pkg.list_pkgs` output
- Fixes for Jinja renderer, timezone *`module`*/*`state`* (issue #12724)
- Fix timedatectl parsing for systemd>=210 (issue #12728)
- Fix `saltenv` being written to YUM repo config files (issue #12887)
- Removed the deprecated external nodes classifier (originally accessible by setting a value for external_nodes in the master configuration file). Note that this functionality has been marked deprecated for some time and was replaced by the more general *master tops* system.

- More robust escaping of ldap filter strings.
- Fix trailing slash in *`gitfs_root`* causing files not to be available (issue #13185)

28.2.97 Salt 2014.1.6 Release Notes

release 2014-07-08

Version 2014.1.6 is another bugfix release for *2014.1.0*. Changes include:

- Fix extra `iptables --help` output (Sorry!) (issue #13648, issue #13507, issue #13527, issue #13607)
- Fix `mount.active` for Solaris
- Fix support for `allow-hotplug` statement in debian_ip network module
- Add sqlite3 to esky builds
- Fix `jobs.active` output (issue #9526)
- Fix the `virtual` grain for Xen (issue #13534)
- Fix _ext_nodes unavailable on master (issue #13535)
- Fix eauth for batch mode (issue #9605)
- Fix force-related issues with `tomcat` support (issue #12889)
- Fix KeyError when cloud mapping
- Fix salt-minion restart loop in Windows (issue #12086)
- Fix detection of `service` virtual module on Fedora minions
- Fix traceback with missing ipv4 grain (issue #13838)
- Fix issue in roots backend with invalid data in mtime_map (issue #13836)
- Fix traceback in `jobs.active` (issue #11151)
- Fix `master_tops` and `_ext_nodes` issue (issue #13535, issue #13673)

28.2.98 Salt 2014.1.7 Release Notes

release 2014-07-09

Version 2014.1.7 is another bugfix release for *2014.1.0*. Changes include:

- Fix batch mode regression (issue #14046)

This release was a hotfix release for the regression listed above which was present in the 2014.1.6 release. The changes included in 2014.1.6 are listed below:

- Fix extra `iptables --help` output (Sorry!) (issue #13648, issue #13507, issue #13527, issue #13607)
- Fix `mount.active` for Solaris
- Fix support for `allow-hotplug` statement in debian_ip network module
- Add sqlite3 to esky builds
- Fix `jobs.active` output (issue #9526)
- Fix the `virtual` grain for Xen (issue #13534)
- Fix eauth for batch mode (issue #9605)

- Fix force-related issues with `tomcat` support (issue #12889)
- Fix KeyError when cloud mapping
- Fix salt-minion restart loop in Windows (issue #12086)
- Fix detection of `service` virtual module on Fedora minions
- Fix traceback with missing ipv4 grain (issue #13838)
- Fix issue in roots backend with invalid data in mtime_map (issue #13836)
- Fix traceback in `jobs.active` (issue #11151)
- Fix `master_tops` and `_ext_nodes` issue (issue #13535, issue #13673)

28.2.99 Salt 2014.1.8 Release Notes

release 2014-07-30

Note: This release contained a regression which caused inaccurate Salt version detection, and thus was never packaged for general release. Please use version 2014.1.10 instead.

Version 2014.1.8 is another bugfix release for *2014.1.0*. Changes include:

- Ensure salt-ssh will not continue if permissions on a temporary directory are not correct.
- Use the bootstrap script distributed with Salt instead of relying on an external resource
- Remove unused testing code
- Ensure salt states are placed into the `.salt` directory in salt-ssh
- Use a randomized path for temporary files in a salt-cloud deployment
- Clean any stale directories to ensure a fresh copy of salt-ssh during a deployment

28.2.100 Salt 2014.1.9 Release Notes

release 2014-07-31

Note: This release contained a regression which caused inaccurate Salt version detection, and thus was never packaged for general release. Please use version 2014.1.10 instead.

Note: Version 2014.1.8 contained a regression which caused inaccurate Salt version detection, and thus was never packaged for general release. This version contains the version detection fix, but is otherwise identical to 2014.1.8.

Version 2014.1.9 is another bugfix release for *2014.1.0*. Changes include:

- Ensure salt-ssh will not continue if permissions on a temporary directory are not correct.
- Use the bootstrap script distributed with Salt instead of relying on an external resource
- Remove unused testing code
- Ensure salt states are placed into the `.salt` directory in salt-ssh
- Use a randomized path for temporary files in a salt-cloud deployment

- Clean any stale directories to ensure a fresh copy of salt-ssh during a deployment

28.2.101 Salt 0.10.0 Release Notes

release 2012-06-16

0.10.0 has arrived! This release comes with MANY bug fixes, and new capabilities which greatly enhance performance and reliability. This release is primarily a bug fix release with many new tests and many repaired bugs. This release also introduces a few new key features which were brought in primarily to repair bugs and some limitations found in some of the components of the original architecture.

Major Features

Event System

The Salt Master now comes equipped with a new event system. This event system has replaced some of the back end of the Salt client and offers the beginning of a system which will make plugging external applications into Salt. The event system relies on a local ZeroMQ publish socket and other processes can connect to this socket and listen for events. The new events can be easily managed via Salt's event library.

Unprivileged User Updates

Some enhancements have been added to Salt for running as a user other than root. These new additions should make switching the user that the Salt Master is running as very painless, simply change the `user` option in the master configuration and restart the master, Salt will take care of all of the particulars for you.

Peer Runner Execution

Salt has long had the peer communication system used to allow minions to send commands via the salt master. 0.10.0 adds a new capability here, now the master can be configured to allow for minions to execute Salt runners via the `peer_run` option in the salt master configuration.

YAML Parsing Updates

In the past the YAML parser for sls files would return the incorrect numbers when the file mode was set with a preceding 0. The YAML parser used in Salt has been modified to no longer convert these number into octal but to keep them as the correct value so that sls files can be a little cleaner to write.

State Call Data Files

It was requested that the minion keep a local cache of the most recent executed state run. This has been added and now with state runs the data is stored in a msgpack file in the minion's cachedir.

Turning Off the Job Cache

A new option has been added to the master configuration file. In previous releases the Salt client would look over the Salt job cache to read in the minion return data. With the addition of the event system the Salt client can now watch for events directly from the master worker processes.

This means that the job cache is no longer a hard requirement. Keep in mind though, that turning off the job cache means that historic job execution data cannot be retrieved.

Test Updates

Minion Swarms Are Faster

To continue our efforts with testing Salt's ability to scale the minionswarm script has been updated. The minionswarm can now start up minions much faster than it could before and comes with a new feature allowing modules to be disabled, thus lowering the minion's footprint when making a swarm. These new updates have allows us to test

```
# python minionswarm.py -m 20 --master salt-master
```

Many Fixes

To get a good idea for the number of bugfixes this release offers take a look at the closed tickets for 0.10.0, this is a very substantial update:

https://github.com/saltstack/salt/issues?milestone=12&state=closed

Master and Minion Stability Fixes

As Salt deployments grow new ways to break Salt are discovered. 0.10.0 comes with a number of fixes for the minions and master greatly improving Salt stability.

28.2.102 Salt 0.10.1 Release Notes

release 2012-06-19

28.2.103 Salt 0.10.2 Release Notes

release 2012-07-30

0.10.2 is out! This release comes with enhancements to the pillar interface, cleaner ways to access the salt-call capabilities in the API, minion data caching and the event system has been added to salt minions.

There have also been updates to the ZeroMQ functions, many more tests (thanks to sponsors, the code sprint and many contributors) and a swath of bug fixes.

Major Features

Ext Pillar Modules

The ranks of available Salt modules directories sees a new member in 0.10.2. With the popularity of pillar a higher demand has arisen for `ext_pillar` interfaces to be more like regular Salt module additions. Now ext_pillar interfaces can be added in the same way as other modules, just drop it into the pillar directory in the salt source.

Minion Events

In 0.10.0 an event system was added to the Salt master. 0.10.2 adds the event system to the minions as well. Now event can be published on a local minion as well.

The minions can also send events back up to the master. This means that Salt is able to communicate individual events from the minions back up to the Master which are not associated with command.

Minion Data Caching

When pillar was introduced the landscape for available data was greatly enhanced. The minion's began sending grain data back to the master on a regular basis.

The new config option on the master called `minion_data_cache` instructs the Salt master to maintain a cache of the minion's grains and pillar data in the cachedir. This option is turned off by default to avoid hitting the disk more, but when enabled the cache is used to make grain matching from the salt command more powerful, since the minions that will match can be predetermined.

Backup Files

By default all files replaced by the file.managed and file.recurse states we simply deleted. 0.10.2 adds a new option. By setting the backup option to `minion` the files are backed up before they are replaced.

The backed up files are located in the cachedir under the file_backup directory. On a default system this will be at: `/var/cache/salt/file_backup`

Configuration files

`salt-master` and `salt-minion` automatically load additional configuration files from `master.d/*.conf` respective `minion.d/*.conf` where `master.d`/`minion.d` is a directory in the same directory as the main configuration file.

Salt Key Verification

A number of users complained that they had inadvertently deleted the wrong salt authentication keys. 0.10.2 now displays what keys are going to be deleted and verifies that they are the keys that are intended for deletion.

Key auto-signing

If `autosign_file` is specified in the configuration file incoming keys will be compared to the list of keynames in `autosign_file`. Regular expressions as well as globbing is supported.

The file must only be writable by the user otherwise the file will be ignored. To relax the permission and allow group write access set the `permissive_pki_access` option.

Module changes

Improved OpenBSD support

New modules for managing services and packages were provided by Joshua Elsasser to further improve the support for OpenBSD.

Existing modules like the *disk* module were also improved to support OpenBSD.

SQL Modules

The MySQL and PostgreSQL modules have both received a number of additions thanks to the work of Avi Marcus and Roman Imankulov.

ZFS Support on FreeBSD

A new ZFS module has been added by Kurtis Velarde for FreeBSD supporting various ZFS operations like creating, extending or removing zpools.

Augeas

A new Augeas module by Ulrich Dangel for editing and verifying config files.

Native Debian Service module

The support for the Debian was further improved with an new service module for Debian by Ahmad Khayyat supporting *disable* and *enable*.

Cassandra

Cassandra support has been added by Adam Garside. Currently only status and diagnostic information are supported.

Networking

The networking support for *RHEL* has been improved and supports bonding support as well as zeroconf configuration.

Monit

Basic monit support by Kurtis Velarde to control services via monit.

nzbget

Basic support for controlling nzbget by Joseph Hall

Bluetooth

Baisc `bluez` support for managing and controlling Bluetooth devices. Supports scanning as well as pairing/unpairing by Joseph Hall.

Test Updates

Consistency Testing

Another testing script has been added. A bug was found in pillar when many minions generated pillar data at the same time. The new `consist.py` script is the tests directory was created to reproduce bugs where data should always be consistent.

Many Fixes

To get a good idea for the number of bugfixes this release offers take a look at the closed tickets for 0.10.2, this is a very substantial update:

https://github.com/saltstack/salt/issues?milestone=24&page=1&state=closed

Master and Minion Stability Fixes

As Salt deployments grow new ways to break Salt are discovered. 0.10.2 comes with a number of fixes for the minions and master greatly improving Salt stability.

28.2.104 Salt 0.10.3 Release Notes

release 2012-09-30

The latest taste of Salt has come, this release has many fixes and feature additions. Modifications have been made to make ZeroMQ connections more reliable, the beginning of the ACL system is in place, a new command line parsing system has been added, dynamic module distribution has become more environment aware, the new *master_finger* option and many more!

Major Features

ACL System

The new ACL system has been introduced. The ACL system allows for system users other than root to execute salt commands. Users can be allowed to execute specific commands in the same way that minions are opened up to the peer system.

The configuration value to open up the ACL system is called `client_acl` and is configured like so:

```
client_acl:
  fred:
    - test..*
    - pkg.list_pkgs
```

Where *fred* is allowed access to functions in the test module and to the `pkg.list_pkgs` function.

Master Finger Option

The *master_finger* option has been added to improve the security of minion provisioning. The *master_finger* option allows for the fingerprint of the master public key to be set in the configuration file to double verify that the master is valid. This option was added in response to a motivation to pre-authenticate the master when provisioning new minions to help prevent man in the middle attacks in some situations.

Salt Key Fingerprint Generation

The ability to generate fingerprints of keys used by Salt has been added to `salt-key`. The new option *finger* accepts the name of the key to generate and display a fingerprint for.

```
salt-key -F master
```

Will display the fingerprints for the master public and private keys.

Parsing System

Pedro Algavio, aka s0undt3ch, has added a substantial update to the command line parsing system that makes the help message output much cleaner and easier to search through. Salt parsers now have *–versions-report* besides usual *–version* info which you can provide when reporting any issues found.

Key Generation

We have reduced the requirements needed for *salt-key* to generate minion keys. You're no longer required to have salt configured and it's common directories created just to generate keys. This might prove useful if you're batch creating keys to pre-load on minions.

Startup States

A few configuration options have been added which allow for states to be run when the minion daemon starts. This can be a great advantage when deploying with Salt because the minion can apply states right when it first runs. To use startup states set the `startup_states` configuration option on the minion to *highstate*.

New Exclude Declaration

Some users have asked about adding the ability to ensure that other sls files or ids are excluded from a state run. The exclude statement will delete all of the data loaded from the specified sls file or will delete the specified id:

```
exclude:
  - sls: http
  - id: /etc/vimrc
```

Max Open Files

While we're currently unable to properly handle ZeroMQ's abort signals when the max open files is reached, due to the way that's handled on ZeroMQ's, we have minimized the chances of this happening without at least warning the user.

More State Output Options

Some major changes have been made to the state output system. In the past state return data was printed in a very verbose fashion and only states that failed or made changes were printed by default. Now two options can be passed to the master and minion configuration files to change the behavior of the state output. State output can be set to verbose (default) or non-verbose with the `state_verbose` option:

```
state_verbose: False
```

It is noteworthy that the state_verbose option used to be set to *False* by default but has been changed to *True* by default in 0.10.3 due to many requests for the change.

Te next option to be aware of new and called `state_output`. This option allows for the state output to be set to *full* (default) or *terse*.

The *full* output is the standard state output, but the new *terse* output will print only one line per state making the output much easier to follow when executing a large state system.

```
state_output: terse
```

state.file.append Improvements

The salt state *file.append()* tries *not* to append existing text. Previously the matching check was being made line by line. While this kind of check might be enough for most cases, if the text being appended was multi-line, the check would not work properly. This issue is now properly handled, the match is done as a whole ignoring any white space addition or removal except inside commas. For those thinking that, in order to properly match over multiple lines, salt will load the whole file into memory, that's not true. For most cases this is not important but an erroneous order to read a 4GB file, if not properly handled, like salt does, could make salt chew that amount of memory. Salt has a buffered file reader which will keep in memory a maximum of 256KB and iterates over the file in chunks of 32KB to test for the match, more than enough, if not, explain your usage on a ticket. With this change, also *salt.modules.file.contains()*, *salt.modules.file.contains_regex()*, *salt.modules.file.contains_glob()* and *salt.utils.find* now do the searching and/or matching using the buffered chunks approach explained above.

Two new keyword arguments were also added, *makedirs*, and *source*. The first, *makedirs* will create the necessary directories in order to append to the specified file, of course, it only applies if we're trying to append to a non-existing file on a non-existing directory:

```
/tmp/salttest/file-append-makedirs:
    file.append:
        text: foo
        makedirs: True
```

The second, *source*, allows one to append the contents of a file instead of specifying the text.

```
/tmp/salttest/file-append-source:

file.append:
    - source: salt://testfile
```

Security Fix

A timing vulnerability was uncovered in the code which decrypts the AES messages sent over the network. This has been fixed and upgrading is strongly recommended.

28.2.105 Salt 0.10.4 Release Notes

release 2012-10-23

Salt 0.10.4 is a monumental release for the Salt team, with two new module systems, many additions to allow granular access to Salt, improved platform support and much more.

This release is also exciting because we have been able to shorten the release cycle back to under a month. We are working hard to keep up the aggressive pace and look forward to having releases happen more frequently!

This release also includes a serious security fix and all users are very strongly recommended to upgrade. As usual, upgrade the master first, and then the minion to ensure that the process is smooth.

Major Features

External Authentication System

The new external authentication system allows for Salt to pass through authentication to any authentication system to determine if a user has permission to execute a Salt command. The Unix PAM system is the first supported system with more to come!

The external authentication system allows for specific users to be granted access to execute specific functions on specific minions. Access is configured in the master configuration file, and uses the new access control system:

```
external_auth:
  pam:
    thatch:
      - 'web*':
        - test.*
        - network.*
```

The configuration above allows the user *thatch* to execute functions in the test and network modules on minions that match the web* target.

Access Control System

All Salt systems can now be configured to grant access to non-administrative users in a granular way. The old configuration continues to work. Specific functions can be opened up to specific minions from specific users in the case of external auth and client ACLs, and for specific minions in the case of the peer system.

Access controls are configured like this:

```
client_acl:
  fred:
    - web\*:
      - pkg.list_pkgs
      - test.*
      - apache.*
```

Target by Network

A new matcher has been added to the system which allows for minions to be targeted by network. This new matcher can be called with the *-S* flag on the command line and is available in all places that the matcher system is available. Using it is simple:

```
$ salt -S '192.168.1.0/24' test.ping
$ salt -S '192.168.1.100' test.ping
```

Nodegroup Nesting

Previously a nodegroup was limited by not being able to include another nodegroup, this restraint has been lifted and now nodegroups will be expanded within other nodegroups with the *N@* classifier.

Salt Key Delete by Glob

The ability to delete minion keys by glob has been added to `salt-key`. To delete all minion keys whose minion name starts with 'web':

```
$ salt-key -d 'web*'
```

Master Tops System

The *external_nodes* system has been upgraded to allow for modular subsystems to be used to generate the top file data for a highstate run.

The *external_nodes* option still works but will be deprecated in the future in favor of the new *master_tops* option.

Example of using *master_tops*:

```
master_tops:
  ext_nodes: cobbler-external-nodes
```

Next Level Solaris Support

A lot of work has been put into improved Solaris support by Romeo Theriault. Packaging modules (pkgadd/pkgrm and pkgutil) and states, cron support and user and group management have all been added and improved upon. These additions along with SMF (Service Management Facility) service support and improved Solaris grain detection in 0.10.3 add up to Salt becoming a great tool to manage Solaris servers with.

Security

A vulnerability in the security handshake was found and has been repaired, old minions should be able to connect to a new master, so as usual, the master should be updated first and then the minions.

Pillar Updates

The pillar communication has been updated to add some extra levels of verification so that the intended minion is the only one allowed to gather the data. Once all minions and the master are updated to salt 0.10.4 please activate pillar *2* by changing the *pillar_version* in the master config to *2*. This will be set to *2* by default in a future release.

28.2.106 Salt 0.10.5 Release Notes

release 2012-11-15

Salt 0.10.5 is ready, and comes with some great new features. A few more interfaces have been modularized, like the outputter system. The job cache system has been made more powerful and can now store and retrieve jobs archived in external databases. The returner system has been extended to allow minions to easily retrieve data from a returner interface.

As usual, this is an exciting release, with many noteworthy additions!

Major Features

External Job Cache

The external job cache is a system which allows for a returner interface to also act as a job cache. This system is intended to allow users to store job information in a central location for longer periods of time and to make the act of looking up information from jobs executed on other minions easier.

Currently the external job cache is supported via the mongo and redis returners:

```
ext_job_cache: redis
redis.host: salt
```

Once the external job cache is turned on the new *ret* module can be used on the minions to retrieve return information from the job cache. This can be a great way for minions to respond and react to other minions.

OpenStack Additions

OpenStack integration with Salt has been moving forward at a blistering pace. The new *nova*, *glance*, and *keystone* modules represent the beginning of ongoing OpenStack integration.

The Salt team has had many conversations with core OpenStack developers and is working on linking to OpenStack in powerful new ways.

Wheel System

A new API was added to the Salt Master which allows the master to be managed via an external API. This new system allows Salt API to easily hook into the Salt Master and manage configs, modify the state tree, manage the pillar and more. The main motivation for the wheel system is to enable features needed in the upcoming web UI so users can manage the master just as easily as they manage minions.

The wheel system has also been hooked into the external auth system. This allows specific users to have granular access to manage components of the Salt Master.

Render Pipes

Jack Kuan has added a substantial new feature. The render pipes system allows Salt to treat the render system like unix pipes. This new system enables sls files to be passed through specific render engines. While the default renderer is still recommended, different engines can now be more easily merged. So to pipe the output of Mako used in YAML use this shebang line:

#!mako|yaml

Salt Key Overhaul

The Salt Key system was originally developed as only a CLI interface, but as time went on it was pressed into becoming a clumsy API. This release marks a complete overhaul of Salt Key. Salt Key has been rewritten to function purely from an API and to use the outputter system. The benefit here is that the outputter system works much more cleanly with Salt Key now, and the internals of Salt Key can be used much more cleanly.

Modular Outputters

The outputter system is now loaded in a modular way. This means that output systems can be more easily added by dropping a python file down on the master that contains the function *output*.

Gzip from Fileserver

Gzip compression has been added as an option to the cp.get_file and cp.get_dir commands. This will make file transfers more efficient and faster, especially over slower network links.

Unified Module Configuration

In past releases of Salt, the minions needed to be configured for certain modules to function. This was difficult because it required pre-configuring the minions. 0.10.5 changes this by making all module configs on minions search the master config file for values.

Now if a single database server is needed, then it can be defined in the master config and all minions will become aware of the configuration value.

Salt Call Enhancements

The `salt-call` command has been updated in a few ways. Now, `salt-call` can take the –return option to send the data to a returner. Also, `salt-call` now reports executions in the minion proc system, this allows the master to be aware of the operation salt-call is running.

Death to pub_refresh and sub_timeout

The old configuration values *pub_refresh* and *sub_timeout* have been removed. These options were in place to alleviate problems found in earlier versions of ZeroMQ which have since been fixed. The continued use of these options has proven to cause problems with message passing and have been completely removed.

Git Revision Versions

When running Salt directly from git (for testing or development, of course) it has been difficult to know exactly what code is being executed. The new versioning system will detect the git revision when building and how many commits have been made since the last release. A release from git will look like this:

0.10.4-736-gec74d69

Svn Module Addition

Anthony Cornehl (twinshadow) contributed a module that adds Subversion support to Salt. This great addition helps round out Salt's VCS support.

Noteworthy Changes

Arch Linux Defaults to Systemd

Arch Linux recently changed to use systemd by default and discontinued support for init scripts. Salt has followed suit and defaults to systemd now for managing services in Arch.

Salt, Salt Cloud and Openstack

With the releases of Salt 0.10.5 and Salt Cloud 0.8.2, OpenStack becomes the first (non-OS) piece of software to include support both on the user level (with Salt Cloud) and the admin level (with Salt). We are excited to continue to extend support of other platforms at this level.

28.2.107 Salt 0.11.0 Release Notes

release 2012-12-14

Salt 0.11.0 is here, with some highly sought after and exciting features. These features include the new overstate system, the reactor system, a new state run scope component called __context__, the beginning of the search system (still needs a great deal of work), multiple package states, the MySQL returner and a better system to arbitrarily reference outputters.

It is also noteworthy that we are changing how we mark release numbers. For the life of the project we have been pushing every release with features and fixes as point releases. We will now be releasing point releases for only bug fixes on a more regular basis and major feature releases on a slightly less regular basis. This means that the next release will be a bugfix only release with a version number of 0.11.1. The next feature release will be named 0.12.0 and will mark the end of life for the 0.11 series.

Major Features

OverState

The overstate system is a simple way to manage rolling state executions across many minions. The overstate allows for a state to depend on the successful completion of another state.

Reactor System

The new reactor system allows for a reactive logic engine to be created which can respond to events within a salted environment. The reactor system uses sls files to match events fired on the master with actions, enabling Salt to react to problems in an infrastructure.

Your load-balanced group of webservers is under extra load? Spin up a new VM and add it to the group. Your fileserver is filling up? Send a notification to your sysadmin on call. The possibilities are endless!

Module Context

A new component has been added to the module loader system. The module context is a data structure that can hold objects for a given scope within the module.

This allows for components that are initialized to be stored in a persistent context which can greatly speed up ongoing connections. Right now the best example can be found in the *cp* execution module.

Multiple Package Management

A long desired feature has been added to package management. By definition Salt States have always installed packages one at a time. On most platforms this is not the fastest way to install packages. Erik Johnson, aka terminalmage, has modified the package modules for many providers and added new capabilities to install groups of packages. These package groups can be defined as a list of packages available in repository servers:

```
python_pkgs:
  pkg.installed:
    - pkgs:
      - python-mako
      - whoosh
      - python-git
```

or specify based on the location of specific packages:

```
python_pkgs:
  pkg.installed:
    - sources:
      - python-mako: http://some-rpms.org/python-mako.rpm
      - whoosh: salt://whoosh/whoosh.rpm
      - python-git: ftp://companyserver.net/python-git.rpm
```

Search System

The bones to the search system have been added. This is a very basic interface that allows for search backends to be added as search modules. The first supported search module is the whoosh search backend. Right now only the basic paths for the search system are in place, making this very experimental. Further development will involve improving the search routines and index routines for whoosh and other search backends.

The search system has been made to allow for searching through all of the state and pillar files, configuration files and all return data from minion executions.

Notable Changes

All previous versions of Salt have shared many directories between the master and minion. The default locations for keys, cached data and sockets has been shared by master and minion. This has created serious problems with running a master and a minion on the same systems. 0.11.0 changes the defaults to be separate directories. Salt will also attempt to migrate all of the old key data into the correct new directories, but if it is not successful it may need to be done manually. If your keys exhibit issues after updating make sure that they have been moved from `/etc/salt/pki` to `/etc/salt/pki/{master,minion}`.

The old setup will look like this:

```
/etc/salt/pki
|-- master.pem
|-- master.pub
|-- minions
|   `-- ragnarok.saltstack.net
|-- minions_pre
|-- minion.pem
|-- minion.pub
|-- minion_master.pub
|-- minions_pre
`-- minions_rejected
```

With the accepted minion keys in `/etc/salt/pki/minions`, the new setup places the accepted minion keys in `/etc/salt/pki/master/minions`.

```
/etc/salt/pki
|-- master
|   |-- master.pem
|   |-- master.pub
|   |-- minions
|   |   `-- ragnarok.saltstack.net
|   |-- minions_pre
|   `-- minions_rejected
|-- minion
|   |-- minion.pem
|   |-- minion.pub
|   `-- minion_master.pub
```

28.2.108 Salt 0.11.1 Release Notes

release 2012-12-19

28.2.109 Salt 0.12.0 Release Notes

release 2013-01-15

Another feature release of Salt is here! Some exciting additions are included with more ways to make salt modular and even easier management of the salt file server.

Major Features

Modular Fileserver Backend

The new modular fileserver backend allows for any external system to be used as a salt file server. The main benefit here is that it is now possible to tell the master to directly use a git remote location, or many git remote locations, automatically mapping git branches and tags to salt environments.

Windows is First Class!

A new Salt Windows installer is now available! Much work has been put in to improve Windows support. With this much easier method of getting Salt on your Windows machines, we hope even more development and progress will occur. Please file bug reports on the Salt GitHub repo issue tracker so we can continue improving.

One thing that is missing on Windows that Salt uses extensively is a software package manager and a software package repository. The Salt pkg state allows sys admins to install software across their infrastructure and across operating systems. Software on Windows can now be managed in the same way. The SaltStack team built a package manager that interfaces with the standard Salt pkg module to allow for installing and removing software on Windows. In addition, a software package repository has been built on top of the Salt fileserver. A small YAML file provides the information necessary for the package manager to install and remove software.

An interesting feature of the new Salt Windows software package repository is that one or more remote git repositories can supplement the master's local repository. The repository can point to software on the master's fileserver or on an HTTP, HTTPS, or ftp server.

New Default Outputter

Salt displays data to the terminal via the outputter system. For a long time the default outputter for Salt has been the python pretty print library. While this has been a generally reasonable outputter, it did have many failings. The new default outputter is called "nested", it recursively scans return data structures and prints them out cleanly.

If the result of the new nested outputter is not desired any other outputter can be used via the –out option, or the output option can be set in the master and minion configs to change the default outputter.

Internal Scheduler

The internal Salt scheduler is a new capability which allows for functions to be executed at given intervals on the minion, and for runners to be executed at given intervals on the master. The scheduler allows for sequences such as executing state runs (locally on the minion or remotely via an overstate) or continually gathering system data to be run at given intervals.

The configuration is simple, add the schedule option to the master or minion config and specify jobs to run, this in the master config will execute the state.over runner every 60 minutes:

```yaml
schedule:
  overstate:
    function: state.over
    minutes: 60
```

This example for the minion configuration will execute a highstate every 30 minutes:

```yaml
schedule:
  highstate:
    function: state.highstate
    minutes: 30
```

Optional DSL for SLS Formulas

Jack Kuan, our renderer expert, has created something that is astonishing. Salt, now comes with an optional Python based DSL, this is a very powerful interface that makes writing SLS files in pure python easier than it was with the raw py renderer. As usual this can be used with the renderer shebang line, so a single sls can be written with the DSL if pure python power is needed while keeping other sls files simple with YAML.

Set Grains Remotely

A new execution function and state module have been added that allows for grains to be set on the minion. Now grains can be set via a remote execution or via states. Use the *grains.present* state or the *grains.setval* execution functions.

Gentoo Additions

Major additions to Gentoo specific components have been made. The encompasses executions modules and states ranging from supporting the make.conf file to tools like layman.

28.2.110 Salt 0.12.1 Release Notes

release 2013-01-21

28.2.111 Salt 0.13.0 Release Notes

release 2013-02-12

The lucky number 13 has turned the corner! From CLI notifications when quitting a salt command, to substantial improvements on Windows, Salt 0.13.0 has arrived!

Major Features

Improved file.recurse Performance

The file.recurse system has been deployed and used in a vast array of situations. Fixes to the file state and module have led towards opening up new ways of running file.recurse to make it faster. Now the file.recurse state will download fewer files and will run substantially faster.

Windows Improvements

Minion stability on Windows has improved. Many file operations, including file.recurse, have been fixed and improved. The network module works better, to include network.interfaces. Both 32bit and 64bit installers are now available.

Nodegroup Targeting in Peer System

In the past, nodegroups were not available for targeting via the peer system. This has been fixed, allowing the new nodegroup expr_form argument for the publish.publish function:

> salt-call publish.publish group1 test.ping expr_form=nodegroup

Blacklist Additions

Additions allowing more granular blacklisting are available in 0.13.0. The ability to blacklist users and functions in client_acl have been added, as well as the ability to exclude state formulas from the command line.

Command Line Pillar Embedding

Pillar data can now be embedded on the command line when calling `state.sls` and `state.highstate`. This allows for on the fly changes or settings to pillar and makes parameterizing state formulas even easier. This is done via the keyword argument:

```
salt '*' state.highstate pillar='{"cheese": "spam"}'
```

The above example will extend the existing pillar to hold the `cheese` key with a value of `spam`. If the `cheese` key is already specified in the minion's pillar then it will be overwritten.

CLI Notifications

In the past hitting ctrl-C and quitting from the `salt` command would just drop to a shell prompt, this caused confusion with users who expected the remote executions to also quit. Now a message is displayed showing what command can be used to track the execution and what the job id is for the execution.

Version Specification in Multiple-Package States

Versions can now be specified within multiple-package *`pkg.installed`* states. An example can be found below:

```
mypkgs:
  pkg.installed:
    - pkgs:
      - foo
      - bar: 1.2.3-4
      - baz
```

Noteworthy Changes

The configuration subsystem in Salt has been overhauled to make the `opts` dict used by Salt applications more portable, the problem is that this is an incompatible change with salt-cloud, and salt-cloud will need to be updated to the latest git to work with Salt 0.13.0. Salt Cloud 0.8.5 will also require Salt 0.13.0 or later to function.

The SaltStack team is sorry for the inconvenience here, we work hard to make sure these sorts of things do not happen, but sometimes hard changes get in.

28.2.112 Salt 0.13.1 Release Notes

release 2013-02-15

28.2.113 Salt 0.13.2 Release Notes

release 2013-03-13

28.2.114 Salt 0.13.3 Release Notes

release 2013-03-18

28.2.115 Salt 0.14.0 Release Notes

release 2013-03-23

Salt 0.14.0 is here! This release was held up primarily by PyCon, Scale, and illness, but has arrived! 0.14.0 comes with many new features and is breaking ground for Salt in the area of cloud management with the introduction of Salt providing basic cloud controller functionality.

Major Features

Salt - As a Cloud Controller

This is the first primitive inroad to using Salt as a cloud controller is available in 0.14.0. Be advised that this is alpha, only tested in a few very small environments.

The cloud controller is built using kvm and libvirt for the hypervisors. Hypervisors are autodetected as minions and only need to have libvirt running and kvm installed to function. The features of the Salt cloud controller are as follows:

- Basic vm discovery and reporting
- Creation of new virtual machines
- Seeding virtual machines with Salt via qemu-nbd or libguestfs
- Live migration (shared and non shared storage)
- Delete existing VMs

It is noteworthy that this feature is still Alpha, meaning that all rights are reserved to change the interface if needs be in future releases!

Libvirt State

One of the problems with libvirt is management of certificates needed for live migration and cross communication between hypervisors. The new `libvirt` state makes the Salt Master hold a CA and manage the signing and distribution of keys onto hypervisors, just add a call to the libvirt state in the sls formulas used to set up a hypervisor:

```
libvirt_keys:
  libvirt.keys
```

New get Functions

An easier way to manage data has been introduced. The pillar, grains, and config execution modules have been extended with the new `get` function. This function works much in the same way as the get method in a python dict, but with an enhancement, nested dict components can be extracted using a *:* delimiter.

If a structure like this is in pillar:

```
foo:
  bar:
    baz: quo
```

Extracting it from the raw pillar in an sls formula or file template is done this way:

```
{{ pillar['foo']['bar']['baz'] }}
```

Now with the new get function the data can be safely gathered and a default can be set allowing the template to fall back if the value is not available:

```
{{ salt['pillar.get']('foo:bar:baz', 'qux') }}
```

This makes handling nested structures much easier, and defaults can be cleanly set. This new function is being used extensively in the new formulae repository of salt sls formulas.

28.2.116 Salt 0.14.1 Release Notes

release 2013-04-13

28.2.117 Salt 0.15.0 Release Notes

release 2013-05-03

The many new features of Salt 0.15.0 have arrived! Salt 0.15.0 comes with many smaller features and a few larger ones.

These features range from better debugging tools to the new Salt Mine system.

Major Features

The Salt Mine

First there was the peer system, allowing for commands to be executed from a minion to other minions to gather data live. Then there was the external job cache for storing and accessing long term data. Now the middle ground is being filled in with the Salt Mine. The Salt Mine is a system used to execute functions on a regular basis on minions and then store only the most recent data from the functions on the master, then the data is looked up via targets.

The mine caches data that is public to all minions, so when a minion posts data to the mine all other minions can see it.

IPV6 Support

0.13.0 saw the addition of initial IPV6 support but errors were encountered and it needed to be stripped out. This time the code covers more cases and must be explicitly enabled. But the support is much more extensive than before.

Copy Files From Minions to the Master

Minions have long been able to copy files down from the master file server, but until now files could not be easily copied from the minion up to the master.

A new function called `cp.push` can push files from the minions up to the master server. The uploaded files are then cached on the master in the master cachedir for each minion.

Better Template Debugging

Template errors have long been a burden when writing states and pillar. 0.15.0 will now send the compiled template data to the debug log, this makes tracking down the intermittent stage templates much easier. So running state.sls or state.highstate with *-l debug* will now print out the rendered templates in the debug information.

State Event Firing

The state system is now more closely tied to the master's event bus. Now when a state fails the failure will be fired on the master event bus so that the reactor can respond to it.

Major Syndic Updates

The Syndic system has been basically re-written. Now it runs in a completely asynchronous way and functions primarily as an event broker. This means that the events fired on the syndic are now pushed up to the higher level master instead of the old method used which waited for the client libraries to return.

This makes the syndic much more accurate and powerful, it also means that all events fired on the syndic master make it up the pipe as well making a reactor on the higher level master able to react to minions further downstream.

Peer System Updates

The Peer System has been updated to run using the client libraries instead of firing directly over the publish bus. This makes the peer system much more consistent and reliable.

Minion Key Revocation

In the past when a minion was decommissioned the key needed to be manually deleted on the master, but now a function on the minion can be used to revoke the calling minion's key:

```
$ salt-call saltutil.revoke_auth
```

Function Return Codes

Functions can now be assigned numeric return codes to determine if the function executed successfully. While not all functions have been given return codes, many have and it is an ongoing effort to fill out all functions that might return a non-zero return code.

Functions in Overstate

The overstate system was originally created to just manage the execution of states, but with the addition of return codes to functions, requisite logic can now be used with respect to the overstate. This means that an overstate stage can now run single functions instead of just state executions.

Pillar Error Reporting

Previously if errors surfaced in pillar, then the pillar would consist of only an empty dict. Now all data that was successfully rendered stays in pillar and the render error is also made available. If errors are found in the pillar, states will refuse to run.

Using Cached State Data

Sometimes states are executed purely to maintain a specific state rather than to update states with new configs. This is grounds for the new cached state system. By adding *cache=True* to a state call the state will not be generated fresh from the master but the last state data to be generated will be used. If no previous state data is available then fresh data will be generated.

Monitoring States

The new monitoring states system has been started. This is very young but allows for states to be used to configure monitoring routines. So far only one monitoring state is available, the `disk.status` state. As more capabilities are added to Salt UI the monitoring capabilities of Salt will continue to be expanded.

28.2.118 Salt 0.15.1 Release Notes

release 2013-05-08

The 0.15.1 release has been posted, this release includes fixes to a number of bugs in 0.15.1 and a three security patches.

Security Updates

A number of security issues have been resolved via the 0.15.1 release.

Path Injection in Minion IDs

Salt masters did not properly validate the id of a connecting minion. This can lead to an attacker uploading files to the master in arbitrary locations. In particular this can be used to bypass the manual validation of new unknown minions. Exploiting this vulnerability does not require authentication.

This issue affects all known versions of Salt.

This issue was reported by Ronald Volgers.

Patch

The issue is fixed in Salt 0.15.1. Updated packages are available in the usual locations.

Specific commits:

https://github.com/saltstack/salt/commit/5427b9438e452a5a8910d9128c6aafb45d8fd5d3

https://github.com/saltstack/salt/commit/7560908ee62351769c3cd43b03d74c1ca772cc52

https://github.com/saltstack/salt/commit/e200b8a7ff53780124e08d2bdefde7587e52bfca

RSA Key Generation Fault

RSA key generation was done incorrectly, leading to very insecure keys. It is recommended to regenerate all RSA keys.

This issue can be used to impersonate Salt masters or minions, or decrypt any transferred data.

This issue can only be exploited by attackers who are able to observe or modify traffic between Salt minions and the legitimate Salt master.

A tool was included in 0.15.1 to assist in mass key regeneration, the manage.regen_keys runner.

This issue affects all known versions of Salt.

This issue was reported by Ronald Volgers.

Patch

The issue is fixed in Salt 0.15.1. Updated packages are available in the usual locations.

Specific commits:

https://github.com/saltstack/salt/commit/5dd304276ba5745ec21fc1e6686a0b28da29e6fc

Command Injection Via ext_pillar

Arbitrary shell commands could be executed on the master by an authenticated minion through options passed when requesting a pillar.

Ext pillar options have been restricted to only allow safe external pillars to be called when prompted by the minion.

This issue affects Salt versions from 0.14.0 to 0.15.0.

This issue was reported by Ronald Volgers.

Patch

The issue is fixed in Salt 0.15.1. Updated packages are available in the usual locations.

Specific commits:

https://github.com/saltstack/salt/commit/43d8c16bd26159d827d1a945c83ac28159ec5865

28.2.119 Salt 0.15.2 Release Notes

release 2013-05-29

28.2.120 Salt 0.15.3 Release Notes

release 2013-06-01

28.2.121 Salt 0.16.0 Release Notes

release 2013-07-01

The 0.16.0 release is an exciting one, with new features in master redundancy, and a new, powerful requisite.

Major Features

Multi-Master

This new capability allows for a minion to be actively connected to multiple salt masters at the same time. This allows for multiple masters to send out commands to minions and for minions to automatically reconnect to masters that have gone down. A tutorial is available to help get started here:

Multi Master Tutorial

Prereq, the New Requisite

The new *prereq* requisite is very powerful! It allows for states to execute based on a state that is expected to make changes in the future. This allows for a change on the system to be preempted by another execution. A good example is needing to shut down a service before modifying files associated with it, allowing, for instance, a webserver to be shut down allowing a load balancer to stop sending requests while server side code is updated. In this case, the prereq will only run if changes are expected to happen in the prerequired state, and the prerequired state will always run after the prereq state and only if the prereq state succeeds.

Peer System Improvements

The peer system has been revamped to make it more reliable, faster, and like the rest of Salt, async. The peer calls when an updated minion and master are used together will be much faster!

Relative Includes

The ability to include an sls relative to the defined sls has been added, the new syntax id documented here:

Includes

More State Output Options

The `state_output` option in the past only supported *full* and *terse*, 0.16.0 add the *mixed* and *changes* modes further refining how states are sent to users' eyes.

Improved Windows Support

Support for Salt on Windows continues to improve. Software management on Windows has become more seamless with Linux/UNIX/BSD software management. Installed software is now recognized by the short names defined in the *repository SLS*. This makes it possible to run `salt '*' pkg.version firefox` and get back results from Windows and non-Windows minions alike.

When templating files on Windows, Salt will now correctly use Windows appropriate line endings. This makes it much easier to edit and consume files on Windows.

When using the cmd state the `shell` option now allows for specifying Windows Powershell as an alternate shell to execute cmd.run and cmd.script. This opens up Salt to all the power of Windows Powershell and its advanced Windows management capabilities.

Several fixes and optimizations were added for the Windows networking modules, especially when working with IPv6.

A system module was added that makes it easy to restart and shutdown Windows minions.

The Salt Minion will now look for its config file in `c:\salt\conf` by default. This means that it's no longer necessary to specify the `-c` option to specify the location of the config file when starting the Salt Minion on Windows in a terminal.

Multiple Targets for pkg.removed, pkg.purged States

Both *`pkg.removed`* and *`pkg.purged`* now support the `pkgs` argument, which allow for multiple packages to be targeted in a single state. This, as in *`pkg.installed`*, helps speed up these states by reducing the number of times that the package management tools (apt, yum, etc.) need to be run.

Random Times in Cron States

The temporal parameters in *`cron.present`* states (minute, hour, etc.) can now be randomized by using `random` instead of a specific value. For example, by using the `random` keyword in the `minute` parameter of a cron state, the same cron job can be pushed to hundreds or thousands of hosts, and they would each use a randomly-generated minute. This can be helpful when the cron job accesses a network resource, and it is not desirable for all hosts to run the job concurrently.

```
/path/to/cron/script:
  cron.present:
    - user: root
    - minute: random
    - hour: 2
```

Since Salt assumes a value of `*` for unspecified temporal parameters, adding a parameter to the state and setting it to `random` will change that value from `*` to a randomized numeric value. However, if that field in the cron entry on the minion already contains a numeric value, then using the `random` keyword will not modify it.

Confirmation Prompt on Key Acceptance

When accepting new keys with `salt-key -a minion-id` or `salt-key -A`, there is now a prompt that will show the affected keys and ask for confirmation before proceeding. This prompt can be bypassed using the `-y` or `--yes` command line argument, as with other `salt-key` commands.

Support for Setting Password Hashes on BSD Minions

FreeBSD, NetBSD, and OpenBSD all now support setting passwords in *`user.present`* states.

28.2.122 Salt 0.16.1 Release Notes

release 2013-07-29

28.2.123 Salt 0.16.2 Release Notes

release 2013-08-01

Version 0.16.2 is a bugfix release for *0.16.0*, and contains a number of fixes.

Windows

- Only allow Administrator's group and SYSTEM user access to C:\salt. This eliminates a race condition where a non-admin user could modify a template or managed file before it is executed by the minion (which is running as an elevated user), thus avoiding a potential escalation of privileges. (issue #6361)

Grains

- Fixed detection of `virtual` grain on OpenVZ hardware nodes
- Gracefully handle lsb_release data when it is enclosed in quotes
- LSB grains are now prefixed with `lsb_distrib_` instead of simply `lsb_`. The old naming is not preserved, so SLS may be affected.
- Improved grains detection on MacOS

Pillar

- Don't try to load *`git_pillar`* if not enabled in master config (issue #6052)
- Functions *`pillar.item`* and *`pillar.items`* added for parity with *`grains.item`*/*`grains.items`*. The old function `pillar.data` is preserved for backwards compatibility.
- Fixed minion traceback when Pillar SLS is malformed (issue #5910)

Peer Publishing

- More gracefully handle improperly quoted publish commands (issue #5958)
- Fixed traceback when timeout specified via the CLI fo *`publish.publish`*, *`publish.full_data`* (issue #5959)
- Fixed unintended change in output of *`publish.publish`* (issue #5928)

Minion

- Fixed salt-key usage in minionswarm script
- Quieted warning about **SALT_MINION_CONFIG** environment variable on minion startup and for CLI commands run via `salt-call` (issue #5956)
- Added minion config parameter *`random_reauth_delay`* to stagger re-auth attempts when the minion is waiting for the master to approve its public key. This helps prevent SYN flooding in larger environments.

User/Group Management

- Implement previously-ignored `unique` option for *user.present* states in FreeBSD
- Report in state output when a *group.present* state attempts to use a gid in use by another group
- Fixed regression that prevents a *user.present* state to set the password hash to the system default (i.e. an unset password)
- Fixed multiple *group.present* states with the same group (issue #6439)

File Management

- Fixed file.mkdir setting incorrect permissions (issue #6033)
- Fixed cleanup of source files for templates when `/tmp` is in file_roots (issue #6118)
- Fixed caching of zero-byte files when a non-empty file was previously cached at the same path
- Added HTTP authentication support to the cp module (issue #5641)
- Diffs are now suppressed when binary files are changed

Package/Repository Management

- Fixed traceback when there is only one target for *pkg.latest* states
- Fixed regression in detection of virtual packages (apt)
- Limit number of pkg database refreshes to once per *state.sls*/*state.highstate*
- YUM: Allow 32-bit packages with arches other than i686 to be managed on 64-bit systems (issue #6299)
- Fixed incorrect reporting in pkgrepo.managed states (issue #5517)
- Fixed 32-bit binary package installs on 64-bit RHEL-based distros, and added proper support for 32-bit packages on 64-bit Debian-based distros (issue #6303)
- Fixed issue where requisites were inadvertently being put into YUM repo files (issue #6471)

Service Management

- Fixed inaccurate reporting of results in *service.running* states when the service fails to start (issue #5894)
- Fixed handling of custom initscripts in RHEL-based distros so that they are immediately available, negating the need for a second state run to manage the service that the initscript controls

Networking

- Function network.hwaddr renamed to *network.hw_addr* to match *network.ip_addrs* and *network.ip_addrs6*. All three functions also now work without the underscore in the name, as well.
- Fixed traceback in *bridge.show* when interface is not present (issue #6326)

SSH

- Fixed incorrect result reporting for some *ssh_known_hosts.present* states
- Fixed inaccurate reporting when *ssh_auth.present* states are run with `test=True`, when rsa/dss is used for the `enc` param instead of ssh-rsa/ssh-dss (issue #5374)

pip

- Properly handle `-f` lines in pip freeze output
- Fixed regression in pip.installed states with specifying a requirements file (issue #6003)
- Fixed use of `editable` argument in `pip.installed` states (issue #6025)
- Deprecated `runas` parameter in execution function calls, in favor of `user`

MySQL

- Allow specification of *MySQL* connection arguments via the CLI, overriding/bypassing minion config params
- Allow *mysql_user.present* states to set a passwordless login (issue #5550)
- Fixed endless loop when *mysql.processlist* is run (issue #6297)

PostgreSQL

- Fixed traceback in *postgres.user_list* (issue #6352)

Miscellaneous

- Don't allow *npm states* to be used if *npm module* is not available
- Fixed *alternatives.install* states for which the target is a symlink (issue #6162)
- Fixed traceback in *sysbench module* (issue #6175)
- Fixed traceback in job cache
- Fixed tempfile cleanup for windows
- Fixed issue where SLS files using the *pydsl renderer* were not being run
- Fixed issue where returners were being passed incorrect information (issue #5518)
- Fixed traceback when numeric args are passed to *cmd.script* states
- Fixed bug causing *cp.get_dir* to return more directories than expected (issue #6048)
- Fixed traceback when *supervisord.running* states are run with `test=True` (issue #6053)
- Fixed tracebacks when Salt encounters problems running rbenv (issue #5888)
- Only make the *monit module* available if monit binary is present (issue #5871)
- Fixed incorrect behavior of `img.mount_image`
- Fixed traceback in *tomcat.deploy_war* in Windows
- Don't re-write /etc/fstab if mount fails
- Fixed tracebacks when Salt encounters problems running gem (issue #5886)

- Fixed incorrect behavior of `selinux.boolean` states (issue #5912)
- `RabbitMQ`: Quote passwords to avoid symbols being interpolated by the shell (issue #6338)
- Fixed tracebacks in `extfs.mkfs` and `extfs.tune` (issue #6462)
- Fixed a regression with the `module.run` state where the `m_name` and `m_fun` arguments were being ignored (issue #6464)

28.2.124 Salt 0.16.3 Release Notes

release 2013-08-09

Version 0.16.3 is another bugfix release for *0.16.0*. The changes include:

- Various documentation fixes
- Fix proc directory regression (issue #6502)
- Properly detect Linaro Linux (issue #6496)
- Fix regressions in `mount.mounted` (issue #6522, issue #6545)
- Skip malformed state requisites (issue #6521)
- Fix regression in gitfs from bad import
- Fix for watching prereq states (including recursive requisite error) (issue #6057)
- Fix mod_watch not overriding prereq (issue #6520)
- Don't allow functions which compile states to be called within states (issue #5623)
- Return error for malformed top.sls (issue #6544)
- Fix traceback in `mysql.query`
- Fix regression in binary package installation for 64-bit packages on Debian-based Linux distros (issue #6563)
- Fix traceback caused by running `cp.push` without having set `file_recv` in the master config file
- Fix scheduler configuration in pillar (issue #6201)

28.2.125 Salt 0.16.4 Release Notes

release 2013-09-07

Version 0.16.4 is another bugfix release for *0.16.0*, likely to be the last before 0.17.0 is released. The changes include:

- Multiple documentation improvements/additions
- Added the `osfinger` and `osarch` grains
- Properly handle 32-bit packages for debian32 on x86_64 (issue #6607)
- Fix regression in yum package installation in CentOS 5 (issue #6677)
- Fix bug in `hg.latest` state that would erroneously delete directories (issue #6661)
- Fix bug related to pid not existing for `ps.top` (issue #6679)
- Fix regression in `MySQL returner` (issue #6695)
- Fix IP addresses grains (`ipv4` and `ipv6`) to include all addresses (issue #6656)
- Fix regression preventing authenticated FTP (issue #6733)

- Fix setting password for windows users (issue #6824)
- Fix `file.contains` on values YAML parses as non-string (issue #6817)
- Fix `file.get_gid`, `file.get_uid`, and `file.chown` for broken symlinks (issue #6826)
- Fix comment for service reloads in service state (issue #6851)

28.2.126 Salt 0.17.0 Release Notes

release 2013-09-26

The 0.17.0 release is a very exciting release of Salt, this brings to Salt some very powerful new features and advances. The advances range from the state system to the test suite, covering new transport capabilities and making states easier and more powerful, to extending Salt Virt and much more!

The 0.17.0 release will also be the last release of Salt to follow the old 0.XX.X numbering system, the next release of Salt will change the numbering to be date based following this format:

<Year>.<Month>.<Minor>

So if the release happens in November of 2013 the number will be 13.11.0, the first bugfix release will be 13.11.1 and so forth.

Major Features

Halite

The new Halite web GUI is now available on PyPI. A great deal of work has been put into Halite to make it fully event driven and amazingly fast. The Halite UI can be started from within the Salt Master (after being installed from PyPI), or standalone, and does not require an external database to run. It is very lightweight!

This initial release of Halite is primarily the framework for the UI and the communication systems, making it easy to extend and build the UI up. It presently supports watching the event bus and firing commands over Salt.

At this time, Halite is not available as a package, but installation documentation is available at: http://docs.saltstack.com/topics/tutorials/halite.html

Halite is, like the rest of Salt, Open Source!

Much more will be coming in the future of Halite!

Salt SSH

The new `salt-ssh` command has been added to Salt. This system allows for remote execution and states to be run over ssh. The benefit here being, that salt can run relying only on the ssh agent, rather than requiring a minion to be deployed.

The `salt-ssh` system runs states in a compatible way as Salt and states created and run with salt-ssh can be moved over to a standard salt deployment without modification.

Since this is the initial release of salt-ssh, there is plenty of room for improvement, but it is fully operational, not just a bootstrap tool.

Rosters

Salt is designed to have the minions be aware of the master and the master does not need to be aware of the location of the minions. The new salt roster system was created and designed to facilitate listing the targets for salt-ssh.

The roster system, like most of Salt, is a plugin system, allowing for the list of systems to target to be derived from any pluggable backend. The rosters shipping with 0.17.0 are flat and scan. Flat is a file which is read in via the salt render system and the scan roster does simple network scanning to discover ssh servers.

State Auto Order

This is a major change in how states are evaluated in Salt. State Auto Order is a new feature that makes states get evaluated and executed in the order in which they are defined in the sls file. This feature makes it very easy to see the finite order in which things will be executed, making Salt now, fully imperative AND fully declarative.

The requisite system still takes precedence over the order in which states are defined, so no existing states should break with this change. But this new feature can be turned off by setting `state_auto_order: False` in the master config, thus reverting to the old lexicographical order.

state.sls Runner

The `state.sls` runner has been created to allow for a more powerful system for orchestrating state runs and function calls across the salt minions. This new system uses the state system for organizing executions.

This allows for states to be defined that are executed on the master to call states on minions via `salt-run state.sls`.

Salt Thin

Salt Thin is an exciting new component of Salt, this is the ability to execute Salt routines without any transport mechanisms installed, it is a pure python subset of Salt.

Salt Thin does not have any networking capability, but can be dropped into any system with Python installed and then `salt-call` can be called directly. The Salt Thin system, is used by the `salt-ssh` command, but can still be used to just drop salt somewhere for easy use.

Event Namespacing

Events have been updated to be much more flexible. The tags in events have all been namespaced allowing easier tracking of event names.

Mercurial Fileserver Backend

The popular git fileserver backend has been joined by the mercurial fileserver backend, allowing the state tree to be managed entirely via mercurial.

External Logging Handlers

The external logging handler system allows for Salt to directly hook into any external logging system. Currently supported are sentry and logstash.

Jenkins Testing

The testing systems in Salt have been greatly enhanced, tests for salt are now executed, via jenkins.saltstack.com, across many supported platforms. Jenkins calls out to salt-cloud to create virtual machines on Rackspace, then the minion on the virtual machine checks into the master running on Jenkins where a state run is executed that sets up the minion to run tests and executes the test suite.

This now automates the sequence of running platform tests and allows for continuous destructive tests to be run.

Salt Testing Project

The testing libraries for salt have been moved out of the main salt code base and into a standalone codebase. This has been done to ease the use of the testing systems being used in salt based projects other than Salt itself.

StormPath External Authentication

The external auth system now supports the fantastic Stormpath cloud based authentication system.

LXC Support

Extensive additions have been added to Salt for LXC support. This included the backend libs for managing LXC containers. Addition into the salt-virt system is still in the works.

macOS User/Group Support

Salt is now able to manage users and groups on Minions running macOS. However, at this time user passwords cannot be managed.

Django ORM External Pillar

Pillar data can now be derived from Django managed databases.

Fixes from RC to release

- Multiple documentation fixes
- Add multiple source files + templating for `file.append` (issue #6905)
- Support sysctl configuration files in systemd>=207 (issue #7351)
- Add `file.search` and `file.replace`
- Fix cross-calling execution functions in provider overrides
- Fix locale override for postgres (issue #4543)
- Fix Raspbian identification for service/pkg support (issue #7371)
- Fix `cp.push` file corruption (issue #6495)
- Fix ALT Linux password hash specification (issue #3474)
- Multiple salt-ssh-related fixes and improvements

28.2.127 Salt 0.17.1 Release Notes

release 2013-10-17

Note: THIS RELEASE IS NOT COMPATIBLE WITH PREVIOUS VERSIONS. If you update your master to 0.17.1, you must update your minions as well. Sorry for the inconvenience – this is a result of one of the security fixes listed below.

The 0.17.1 release comes with a number of improvements to salt-ssh, many bugfixes, and a number of security updates.

Salt SSH has been improved to be faster, more featureful and more secure. Since the original release of Salt SSH was primarily a proof of concept, it has been very exciting to see its rapid adoption. We appreciate the willingness of security experts to review Salt SSH and help discover oversights and ensure that security issues only exist for such a tiny window of time.

SSH Enhancements

Shell Improvements

Improvements to Salt SSH's communication have been added that improve routine execution regardless of the target system's login shell.

Performance

Deployment of routines is now faster and takes fewer commands to execute.

Security Updates

Be advised that these security issues all apply to a small subset of Salt users and mostly apply to Salt SSH.

Insufficient Argument Validation

This issue allowed for a user with limited privileges to embed executions inside of routines to execute routines that should be restricted. This applies to users using external auth or client ACL and opening up specific routines.

Be advised that these patches address the direct issue. Additional commits have been applied to help mitigate this issue from resurfacing.

CVE

CVE-2013-4435

Affected Versions

0.15.0 - 0.17.0

Patches

https://github.com/saltstack/salt/commit/6d8ef68b605fd63c36bb8ed96122a75ad2e80269 https://github.com/saltstack/salt/commit/ebdef37b7e5d2b95a01d34b211c61c61da67e46a https://github.com/saltstack/salt/commit/7f190ff890e47cdd591d9d7cefa5126574660824 https://github.com/saltstack/salt/commit/8e5afe59cef6743fe5dbd510dcf463dbdfca1ced https://github.com/saltstack/salt/commit/aca78f314481082862e96d4f0c1b75fa382bb885 https://github.com/saltstack/salt/commit/6a9752cdb1e8df2c9505ea910434c79d132eb1e2 https://github.com/saltstack/salt/commit/b73677435ba54ecfc93c1c2d840a7f9ba6f53410 https://github.com/saltstack/salt/commit/07972eb0a6f985749a55d8d4a2e471596591c80d https://github.com/saltstack/salt/commit/1e3f197726aa13ac5c3f2416000089f477f489b5

Found By

Feth Arezki, of Majerti

MITM SSH attack in salt-ssh

SSH host keys were being accepted by default and not enforced on future SSH connections. These patches set SSH host key checking by default and can be overridden by passing the -i flag to *salt-ssh*.

CVE

CVE-2013-4436

Affected Versions

0.17.0

Found By

Michael Scherer, Red Hat

Insecure Usage of /tmp in salt-ssh

The initial release of salt-ssh used the /tmp directory in an insecure way. These patches not only secure usage of files under /tmp in salt-ssh, but also add checksum validation for all packages sent into the now secure locations on target systems.

CVE

CVE-2013-4438

Affected Versions

0.17.0

Patches

https://github.com/saltstack/salt/commit/aa4bb77ef230758cad84381dde0ec660d2dc340a https://github.com/saltstack/salt/commit/8f92b6b2cb2e4ec3af8783eb6bf4ff06f5a352cf https://github.com/saltstack/salt/commit/c58e56811d5a50c908df0597a0ba0b643b45ebfd https://github.com/saltstack/salt/commit/0359db9b46e47614cff35a66ea6a6a76846885d2 https://github.com/saltstack/salt/commit/4348392860e0fd43701c331ac3e681cf1a8c17b0 https://github.com/saltstack/salt/commit/664d1a1cac05602fad2693f6f97092d98a72bf61 https://github.com/saltstack/salt/commit/bab92775a576e28ff9db262f32db9cf2375bba87 https://github.com/saltstack/salt/commit/c6d34f1acf64900a3c87a2d37618ff414e5a704e

Found By

Michael Scherer, Red Hat

YAML Calling Unsafe Loading Routine

It has been argued that this is not a valid security issue, as the YAML loading that was happening was only being called after an initial gateway filter in Salt has already safely loaded the YAML and would fail if non-safe routines were embedded. Nonetheless, the CVE was filed and patches applied.

CVE

CVE-2013-4438

Patches

https://github.com/saltstack/salt/commit/339b0a51befae6b6b218ebcb55daa9cd3329a1c5

Found By

Michael Scherer, Red Hat

Failure to Drop Supplementary Group on Salt Master

If a salt master was started as a non-root user by the root user, root's groups would still be applied to the running process. This fix changes the process to have only the groups of the running user.

CVE

CVE not considered necessary by submitter.

Affected Versions

0.11.0 - 0.17.0

Patches

https://github.com/saltstack/salt/commit/b89fa9135822d029795ab1eecd68cce2d1ced715

Found By

Michael Scherer, Red Hat

Failure to Validate Minions Posting Data

This issue allowed a minion to pose as another authorized minion when posting data such as the mine data. All minions now pass through the id challenge before posting such data.

CVE

CVE-2013-4439

Affected Versions

0.15.0 - 0.17.0

Patches

https://github.com/saltstack/salt/commit/7b850ff3d07ef6782888914ac4556c01e8a1c482 https://github.com/saltstack/salt/commit/151759b2a1e1c6ce29277aa81b054219147f80fd

Found By

David Anderson

Fix Reference

Version 0.17.1 is the first bugfix release for *0.17.0*. The changes include:

- Fix symbolic links in thin.tgz (issue #7482)
- Pass env through to file.patch state (issue #7452)
- Service provider fixes and reporting improvements (issue #7361)
- Add `--priv` option for specifying salt-ssh private key
- Fix salt-thin's salt-call on setuptools installations (issue #7516)
- Fix salt-ssh to support passwords with spaces (issue #7480)
- Fix regression in wildcard includes (issue #7455)
- Fix salt-call outputter regression (issue #7456)
- Fix custom returner support for startup states (issue #7540)

- Fix value handling in augeas (issue #7605)
- Fix regression in apt (issue #7624)
- Fix minion ID guessing to use `socket.getfqdn()` first (issue #7558)
- Add minion ID caching (issue #7558)
- Fix salt-key race condition (issue #7304)
- Add `--include-all` flag to salt-key (issue #7399)
- Fix custom grains in pillar (part of issue #5716, issue #6083)
- Fix race condition in salt-key (issue #7304)
- Fix regression in minion ID guessing, prioritize `socket.getfqdn()` (issue #7558)
- Cache minion ID on first guess (issue #7558)
- Allow trailing slash in `file.directory` state
- Fix reporting of file_roots in pillar return (issue #5449 and issue #5951)
- Remove pillar matching for mine.get (issue #7197)
- Sanitize args for multiple execution modules
- Fix yumpkg mod_repo functions to filter hidden args (issue #7656)
- Fix conflicting IDs in state includes (issue #7526)
- Fix mysql_grants.absent string formatting issue (issue #7827)
- Fix postgres.version so it won't return None (issue #7695)
- Fix for trailing slashes in mount.mounted state
- Fix rogue AttributErrors in the outputter system (issue #7845)
- Fix for incorrect ssh key encodings resulting in incorrect key added (issue #7718)
- Fix for pillar/grains naming regression in python renderer (issue #7693)
- Fix args/kwargs handling in the scheduler (issue #7422)
- Fix logfile handling for *file://*, *tcp://*, and *udp://* (issue #7754)
- Fix error handling in config file parsing (issue #6714)
- Fix RVM using sudo when running as non-root user (issue #2193)
- Fix client ACL and underlying logging bugs (issue #7706)
- Fix scheduler bug with returner (issue #7367)
- Fix user management bug related to default groups (issue #7690)
- Fix various salt-ssh bugs (issue #7528)
- Many various documentation fixes

28.2.128 Salt 0.17.2 Release Notes

release 2013-11-14

Version 0.17.2 is another bugfix release for *0.17.0*. The changes include:

- Add ability to delete key with grains.delval (issue #7872)

- Fix possible state compiler stack trace (issue #5767)
- Fix architecture regression in yumpkg (issue #7813)
- Use correct `ps` on Debian to prevent truncating (issue #5646)
- Fix grains targeting for new grains (issue #5737)
- Fix bug with merging in git_pillar (issue #6992)
- Fix print_jobs duplicate results
- Fix apt version specification for pkg.install
- Fix possible KeyError from ext_job_cache missing option
- Fix auto_order for `- names` states (issue #7649)
- Fix regression in new gitfs installs (directory not found error)
- Fix escape pipe issue on Windows for file.recurse (issue #7967)
- Fix fileclient in case of master restart (issue #7987)
- Try to output warning if CLI command malformed (issue #6538)
- Fix `--out=quiet` to actually be quiet (issue #8000)
- Fix for state.sls in salt-ssh (issue #7991)
- Fix for MySQL grants ordering issue (issue #5817)
- Fix traceback for certain missing CLI args (issue #8016)
- Add ability to disable lspci queries on master (issue #4906)
- Fail if sls defined in topfile does not exist (issue #5998)
- Add ability to downgrade MySQL grants (issue #6606)
- Fix ssh_auth.absent traceback (issue #8043)
- Add upstart detection for Debian/Raspbian (issue #8039)
- Fix ID-related issues (issue #8052, issue #8050, and others)
- Fix for jinja rendering issues (issue #8066 and issue #8079)
- Fix argument parsing in salt-ssh (issue #7928)
- Fix some GPU detection instances (issue #6945)
- Fix bug preventing includes from other environments in SLS files
- Fix for kwargs with dashes (issue #8102)
- Fix salt.utils.which for windows '.exe' (issue #7904)
- Fix apache.adduser without apachectl (issue #8123)
- Fix issue with evaluating `test` kwarg in states (issue #7788)
- Fix regression in `salt.client.Caller()` (issue #8078)
- Fix apt-key silent failure
- Fix bug where cmd.script would try to run even if caching failed (issue #7601)
- Fix apt `pkg.latest` regression (issue #8067)
- Fix for mine data not being updated (issue #8144)

- Fix for noarch packages in yum
- Fix a Xen detection edge case (issue #7839)
- Fix windows `__opts__` dictionary persistence (issue #7714)
- Fix version generation for when it's part of another git repo (issue #8090)
- Fix _handle_iorder stacktrace so that the real syntax error is shown (issue #8114 and issue #7905)
- Fix `git.latest` state when a commit SHA is used (issue #8163)
- Fix various small bugs in yumpkg.py (issue #8201)
- Fix for specifying identify file in git.latest (issue #8094)
- Fix for `--output-file` CLI arg (issue #8205)
- Add ability to specify shutdown time for system.shutdown (issue #7833)
- Fix for salt version using non-salt git repo info (issue #8266)
- Add additional hints at impact of `pkgrepo` states when `test=True` (issue #8247)
- Fix for salt-ssh files not being owned by root (issue #8216)
- Fix retry logic and error handling in fileserver (related to issue #7755)
- Fix file.replace with `test=True` (issue #8279)
- Add flag for limiting file traversal in fileserver (issue #6928)
- Fix for extra mine processes (issue #5729)
- Fix for unloading custom modules (issue #7691)
- Fix for salt-ssh opts (issue #8005 and issue #8271)
- Fix compound matcher for grains (issue #7944)
- Improve error reporting in ebuild module (related to issue #5393)
- Add `dir_mode` to `file.managed` (issue #7860)
- Improve traceroute support for FreeBSD and macOS (issue #4927)
- Fix for matching minions under syndics (issue #7671)
- Improve exception handling for missing ID (issue #8259)
- Fix grain mismatch for ScientificLinux (issue #8338)
- Add configuration option for minion_id_caching
- Fix open mode auth errors (issue #8402)

28.2.129 Salt 0.17.3 Release Notes

release 2013-12-08

Note: 0.17.3 had some regressions which were promptly fixed in the 0.17.4 release. Please use 0.17.4 instead.

Version 0.17.3 is another bugfix release for *0.17.0*. The changes include:

- Fix some jinja render errors (issue #8418)
- Fix `file.replace` state changing file ownership (issue #8399)

- Fix state ordering with the PyDSL renderer (issue #8446)
- Fix for new npm version (issue #8517)
- Fix for pip state requiring `name` even with requirements file (issue #8519)
- Fix yum logging to open terminals (issue #3855)
- Add sane maxrunning defaults for scheduler (issue #8563)
- Fix states duplicate key detection (issue #8053)
- Fix SUSE patch level reporting (issue #8428)
- Fix managed file creation umask (issue #8590)
- Fix logstash exception (issue #8635)
- Improve argument exception handling for salt command (issue #8016)
- Fix pecl success reporting (issue #8750)
- Fix launchctl module exceptions (issue #8759)
- Fix argument order in pw_user module
- Add warnings for failing grains (issue #8690)
- Fix hgfs problems caused by connections left open (issue #8811 and issue #8810)
- Add Debian iptables default for iptables-persistent package (issue #8889)
- Fix installation of packages with dots in pkg name (issue #8614)
- Fix noarch package installation on CentOS 6 (issue #8945)
- Fix portage_config.enforce_nice_config (issue #8252)
- Fix salt.util.copyfile umask usage (issue #8590)
- Fix rescheduling of failed jobs (issue #8941)
- Fix pkg on Amazon Linux (uses yumpkg5 now) (issue #8226)
- Fix conflicting options in postgres module (issue #8717)
- Fix ps modules for psutil >= 0.3.0 (issue #7432)
- Fix postgres module to return False on failure (issue #8778)
- Fix argument passing for args with pound signs (issue #8585)
- Fix pid of salt CLi command showing in status.pid output (issue #8720)
- Fix rvm to run gem as the correct user (issue #8951)
- Fix namespace issue in win_file module (issue #9060)
- Fix masterless state paths on windows (issue #9021)
- Fix timeout option in master config (issue #9040)

28.2.130 Salt 0.17.4 Release Notes

release 2013-12-10

Version 0.17.4 is another bugfix release for *0.17.0*. The changes include:

- Fix file.replace bug when replacement str is numeric (issue #9101)

- Fix regression in file.managed (issue #9131)
- Prevent traceback when job is None. (issue #9145)

28.2.131 Salt 0.17.5 Release Notes

release 2014-01-27

Version 0.17.5 is another bugfix release for *0.17.0*. The changes include:

- Fix `user.present` states with non-string fullname (issue #9085)
- Fix `virt.init` return value on failure (issue #6870)
- Fix reporting of `file.blockreplace` state when `test=True`
- Fix `network.interfaces` when used in cron (issue #7990)
- Fix bug in pkgrepo when switching to/from mirrorlist-based repo def (issue #9121)
- Fix infinite recursion when cache file is corrupted
- Add checking for rev and mirror/bare args in `git.latest` (issue #9107)
- Add `cmd.watch` alias (points to `cmd.wait`) (issue #8612)
- Fix stacktrace when prereq is not formed as a list (issue #8235)
- Fix stdin issue with lvdisplay command (issue #9128)
- Add pre-check function for range matcher (issue #9236)
- Add exception handling for psutil for processes that go missing (issue #9274)
- Allow `_in` requisites to match both on ID and name (issue #9061)
- Fix multiple client timeout issues (issue #7157 and issue #9302, probably others)
- Fix `ZMQError: Operation cannot be accomplished in current state` errors (issue #6306)
- Multiple optimization in minion auth routines
- Clarify logs for minion ID caching

28.2.132 Salt 0.6.0 release notes

The Salt remote execution manager has reached initial functionality! Salt is a management application which can be used to execute commands on remote sets of servers.

The whole idea behind Salt is to create a system where a group of servers can be remotely controlled from a single master, not only can commands be executed on remote systems, but salt can also be used to gather information about your server environment.

Unlike similar systems, like Func and MCollective, Salt is extremely simple to setup and use, the entire application is contained in a single package, and the master and minion daemons require no running dependencies in the way that Func requires Certmaster and MCollective requires activeMQ.

Salt also manages authentication and encryption. Rather than using SSL for encryption, salt manages encryption on a payload level, so the data sent across the network is encrypted with fast AES encryption, and authentication uses RSA keys. This means that Salt is fast, secure, and very efficient.

Messaging in Salt is executed with ZeroMQ, so the message passing interface is built into salt and does not require an external ZeroMQ server. This also adds speed to Salt since there is no additional bloat on the networking layer, and ZeroMQ has already proven itself as a very fast networking system.

The remote execution in Salt is "Lazy Execution", in that once the command is sent the requesting network connection is closed. This makes it easier to detach the execution from the calling process on the master, it also means that replies are cached, so that information gathered from historic commands can be queried in the future.

Salt also allows users to make execution modules in Python. Writers of these modules should also be pleased to know that they have access to the impressive information gathered from PuppetLabs' Facter application, making Salt module more flexible. In the future I hope to also allow Salt to group servers based on Facter information as well.

All in all Salt is fast, efficient, and clean, can be used from a simple command line client or through an API, uses message queue technology to make network execution extremely fast, and encryption is handled in a very fast and efficient manner. Salt is also VERY easy to use and VERY easy to extend.

You can find the source code for Salt on my GitHub page, I have also set up a few wiki pages explaining how to use and set up Salt. If you are using Arch Linux there is a package available in the Arch Linux AUR.

GitHub page: https://github.com/saltstack/salt

Wiki: https://github.com/saltstack/salt/wiki

Arch Linux Package: https://aur.archlinux.org/packages/salt-git/

I am very open to contributions, for instance I need packages for more Linux distributions as well as BSD packages and testers.

Give Salt a try, this is the initial release and is not a 1.0 quality release, but it has been working well for me! I am eager to get your feedback!

28.2.133 Salt 0.7.0 release notes

I am pleased to announce the release of Salt 0.7.0!

This release marks what is the first stable release of salt, 0.7.0 should be suitable for general use.

0.7.0 Brings the following new features to Salt:

- Integration with Facter data from puppet labs
- Allow for matching minions from the salt client via Facter information
- Minion job threading, many jobs can be executed from the master at once
- Preview of master clustering support - Still experimental
- Introduce new minion modules for stats, virtualization, service management and more
- Add extensive logging to the master and minion daemons
- Add sys.reload_functions for dynamic function reloading
- Greatly improve authentication
- Introduce the saltkey command for managing public keys
- Begin backend development preparatory to introducing butter
- Addition of man pages for the core commands
- Extended and cleaned configuration

0.7.0 Fixes the following major bugs:

- Fix crash in minions when matching failed
- Fix configuration file lookups for the local client
- Repair communication bugs in encryption
- Numerous fixes in the minion modules

The next release of Salt should see the following features:

- Stabilize the cluster support
- Introduce a remote client for salt command tiers
- salt-ftp system for distributed file copies
- Initial support for "butter"

Coming up next is a higher level management framework for salt called Butter. I want salt to stay as a simple and effective communication framework, and allow for more complicated executions to be managed via Butter.

Right now Butter is being developed to act as a cloud controller using salt as the communication layer, but features like system monitoring and advanced configuration control (a puppet manager) are also in the pipe.

Special thanks to Joseph Hall for the status and network modules, and thanks to Matthias Teege for tracking down some configuration bugs!

Salt can be downloaded from the following locations;

Arch Linux Package:

https://aur.archlinux.org/packages/salt-git/

Please enjoy the latest Salt release!

28.2.134 Salt 0.8.0 release notes

Salt 0.8.0 is ready for general consumption! The source tarball is available on GitHub for download:

A lot of work has gone into salt since the last release just 2 weeks ago, and salt has improved a great deal. A swath of new features are here along with performance and threading improvements!

The main new features of salt 0.8.0 are:

Salt-cp

Cython minion modules

Dynamic returners

Faster return handling

Lowered required Python version to 2.6

Advanced minion threading

Configurable minion modules

Salt-cp

The salt-cp command introduces the ability to copy simple files via salt to targeted servers. Using salt-cp is very simple, just call salt-cp with a target specification, the source file(s) and where to copy the files on the minions. For instance:

# salt-cp '*' /etc/hosts /etc/hosts

Will copy the local /etc/hosts file to all of the minions.

Salt-cp is very young, in the future more advanced features will be added, and the functionality will much more closely resemble the cp command.

Cython minion modules

Cython is an amazing tool used to compile Python modules down to c. This is arguably the fastest way to run Python code, and since pyzmq requires cython, adding support to salt for cython adds no new dependencies.

Cython minion modules allow minion modules to be written in cython and therefore executed in compiled c. Simply write the salt module in cython and use the file extension ".pyx" and the minion module will be compiled when the minion is started. An example cython module is included in the main distribution called cytest.pyx:

https://github.com/saltstack/salt/blob/master/salt/modules/cytest.pyx

Dynamic Returners

By default salt returns command data back to the salt master, but now salt can return command data to any system. This is enabled via the new returners modules feature for salt. The returners modules take the return data and sends it to a specific module. The returner modules work like minion modules, so any returner can be added to the minions.

This means that a custom data returner can be added to communicate the return data so anything from MySQL, Redis, MongoDB, and more!

There are 2 simple stock returners in the returners directory:

https://github.com/saltstack/salt/blob/master/salt/returners

The documentation on writing returners will be added to the wiki shortly, and returners can be written in pure Python, or in cython.

Configurable Minion Modules

Minion modules may need to be configured, now the options passed to the minion configuration file can be accessed inside of the minion modules via the __opt__ dict.

Information on how to use this simple addition has been added to the wiki: *Writing modules*

The test module has an example of using the __opts__ dict, and how to set default options:

https://github.com/saltstack/salt/blob/master/salt/modules/test.py

Advanced Minion Threading

In 0.7.0 the minion would block after receiving a command from the master, now the minion will spawn a thread or multiprocess. By default Python threads are used because for general use they have proved to be faster, but the minion can now be configured to use the Python multiprocessing module instead. Using multiprocessing will cause executions that are CPU bound or would otherwise exploit the negative aspects of the Python GIL to run faster and more reliably, but simple calls will still be faster with Python threading. The configuration option can be found in the minion configuration file:

https://github.com/saltstack/salt/blob/master/conf/minion

Lowered Supported Python to 2.6

The requirement for Python 2.7 has been removed to support Python 2.6. I have received requests to take the minimum Python version back to 2.4, but unfortunately this will not be possible, since the ZeroMQ Python bindings do not support Python 2.4.

Salt 0.8.0 is a very major update, it also changes the network protocol slightly which makes communication with older salt daemons impossible, your master and minions need to be upgraded together!

I could use some help bringing salt to the people! Right now I only have packages for Arch Linux, Fedora 14 and Gentoo. We need packages for Debian and people willing to help test on more platforms. We also need help writing more minion modules and returner modules. If you want to contribute to salt please hop on the mailing list and send in patches, make a fork on GitHub and send in pull requests! If you want to help but are not sure where you can, please email me directly or post tot he mailing list!

I hope you enjoy salt, while it is not yet 1.0 salt is completely viable and usable!

-Thomas S. Hatch

28.2.135 Salt 0.8.7 release notes

It has been a month since salt 0.8.0, and it has been a long month! But Salt is still coming along strong. 0.8.7 has a lot of changes and a lot of updates. This update makes Salt's ZeroMQ back end better, strips Facter from the dependencies, and introduces interfaces to handle more capabilities.

Many of the major updates are in the background, but the changes should shine through to the surface. A number of the new features are still a little thin, but the back end to support expansion is in place.

I also recently gave a presentation to the Utah Python users group in Salt Lake City, the slides from this presentation are available here:

The video from this presentation will be available shortly.

The major new features and changes in Salt 0.8.7 are:

- Revamp ZeroMQ topology on the master for better scalability
- State enforcement
- Dynamic state enforcement managers
- Extract the module loader into salt.loader
- Make Job ids more granular
- Replace Facter functionality with the new salt grains interface
- Support for "virtual" salt modules
- Introduce the salt-call command
- Better debugging for minion modules

The new ZeroMQ topology allows for better scalability, this will be required by the need to execute massive file transfers to multiple machines in parallel and state management. The new ZeroMQ topology is available in the aforementioned presentation.

0.8.7 introduces the capability to declare states, this is similar to the capabilities of Puppet. States in salt are declared via state data structures. This system is very young, but the core feature set is available. Salt states work around rendering files which represent Salt high data. More on the Salt state system will be documented in the near future.

The system for loading salt modules has been pulled out of the minion class to be a standalone module, this has enabled more dynamic loading of Salt modules and enables many of the updates in 0.8.7 –

https://github.com/saltstack/salt/blob/master/salt/loader.py

Salt Job ids are now microsecond precise, this was needed to repair a race condition unveiled by the speed improvements in the new ZeroMQ topology.

The new grains interface replaces the functionality of Facter, the idea behind grains differs from Facter in that the grains are only used for static system data, dynamic data needs to be derived from a call to a salt module. This makes grains much faster to use, since the grains data is generated when the minion starts.

Virtual salt modules allows for a salt module to be presented as something other than its module name. The idea here is that based on information from the minion decisions about which module should be presented can be made. The best example is the pacman module. The pacman module will only load on Arch Linux minions, and will be called pkg. Similarly the yum module will be presented as pkg when the minion starts on a Fedora/RedHat system.

The new salt-call command allows for minion modules to be executed from the minion. This means that on the minion a salt module can be executed, this is a great tool for testing Salt modules. The salt-call command can also be used to view the grains data.

In previous releases when a minion module threw an exception very little data was returned to the master. Now the stack trace from the failure is returned making debugging of minion modules MUCH easier.

Salt is nearing the goal of 1.0, where the core feature set and capability is complete!

-Thomas S Hatch

28.2.136 Salt 0.8.8 release notes

Salt 0.8.8 is here! This release adds a great deal of code and some serious new features.

Improved Documentation has been set up for salt using sphinx thanks to the efforts of Seth House. This new documentation system will act as the back end to the salt website which is still under heavy development. The new sphinx documentation system has also been used to greatly clean up the salt manpages. The salt 7 manpage in particular now contains extensive information which was previously only in the wiki. The new documentation can be found at: http://docs.saltstack.com/ We still have a lot to add, and when the domain is set up I will post another announcement.

More additions have been made to the ZeroMQ setup, particularly in the realm of file transfers. Salt 0.8.8 introduces a built in, stateless, encrypted file server which allows salt minions to download files from the salt master using the same encryption system used for all other salt communications. The main motivation for the salt file server has been to facilitate the new salt state system.

Much of the salt code has been cleaned up and a new cleaner logging system has been introduced thanks to the efforts of Pedro Algarvio. These additions will allow for much more flexible logging to be executed by salt, and fixed a great deal of my poor spelling in the salt docstrings! Pedro Algarvio has also cleaned up the API, making it easier to embed salt into another application.

The biggest addition to salt found in 0.8.8 is the new state system. The salt module system has received a new front end which allows salt to be used as a configuration management system. The configuration management system allows for system configuration to be defined in data structures. The configuration management system, or as it is called in salt, the "salt state system" supports many of the features found in other configuration managers, but allows for system states to be written in a far simpler format, executes at blazing speeds, and operates via the salt minion matching system. The state system also operates within the normal scope of salt, and requires no additional configuration to use.

The salt state system can enforce the following states with many more to come: Packages Files Services Executing commands Hosts

The system used to define the salt states is based on a data structure, the data structure used to define the salt states has been made to be as easy to use as possible. The data structure is defined by default using a YAML file rendered via a Jinja template. This means that the state definition language supports all of the data structures that YAML supports, and all of the programming constructs and logic that Jinja supports. If the user does not like YAML or Jinja the states can be defined in yaml-mako, json-jinja, or json-mako. The system used to render the states is completely dynamic, and any rendering system can be added to the capabilities of Salt, this means that a rendering system that renders XML data in a cheetah template, or whatever you can imagine, can be easily added to the capabilities of salt.

The salt state system also supports isolated environments, as well as matching code from several environments to a single salt minion.

The feature base for Salt has grown quite a bit since my last serious documentation push. As we approach 0.9.0 the goals are becoming very clear, and the documentation needs a lot of work. The main goals for 0.9.0 are to further refine the state system, fix any bugs we find, get Salt running on as many platforms as we can, and get the documentation filled out. There is a lot more to come as Salt moves forward to encapsulate a much larger scope, while maintaining supreme usability and simplicity.

If you would like a more complete overview of Salt please watch the Salt presentation: Slides:

-Thomas S Hatch

28.2.137 Salt 0.8.9 Release Notes

Salt 0.8.9 has finally arrived! Unfortunately this is much later than I had hoped to release 0.8.9, life has been very crazy over the last month. But despite challenges, Salt has moved forward!

This release, as expected, adds few new features and many refinements. One of the most exciting aspect of this release is that the development community for salt has grown a great deal and much of the code is from contributors.

Also, I have filled out the documentation a great deal. So information on States is properly documented, and much of the documentation that was out of date has been filled in.

Download!

The Salt source can be downloaded from PyPI:

https://pypi.python.org/packages/source/s/salt/salt-0.8.9.tar.gz

Here s the md5sum:

7d5aca4633bc22f59045f59e82f43b56

For instructions on how to set up Salt please see the *Installation* instructions.

New Features

Salt Run

A big feature is the addition of Salt run, the `salt-run` command allows for master side execution modules to be made that gather specific information or execute custom routines from the master.

Documentation for salt-run can be found *here*

Refined Outputters

One problem often complained about in salt was the fact that the output was so messy. Thanks to help from Jeff Schroeder a cleaner interface for the command output for the Salt CLI has been made. This new interface makes adding new printout formats easy and additions to the capabilities of minion modules makes it possible to set the printout mode or `outputter` for functions in minion modules.

Cross Calling Salt Modules

Salt modules can now call each other, the `__salt__` dict has been added to the predefined references in minion modules. This new feature is documented in the *modules documentation*.

Watch Option Added to Salt State System

Now in Salt states you can set the watch option, this will allow watch enabled states to change based on a change in the other defined states. This is similar to subscribe and notify statements in puppet.

Root Dir Option

Travis Cline has added the ability to define the option `root_dir` which allows the salt minion to operate in a subdir. This is a strong move in supporting the minion running as an unprivileged user

Config Files Defined in Variables

Thanks again to Travis Cline, the master and minion configuration file locations can be defined in environment variables now.

New Modules

Quite a few new modules, states, returners, and runners have been made.

New Minion Modules

apt

Support for apt-get has been added, this adds greatly improved Debian and Ubuntu support to Salt!

useradd and groupadd

Support for manipulating users and groups on Unix-like systems.

moosefs

Initial support for reporting on aspects of the distributed file system, MooseFS. For more information on MooseFS please see: http://www.moosefs.org

Thanks to Joseph Hall for his work on MooseFS support.

mount

Manage mounts and the fstab.

puppet

Execute puppet on remote systems.

shadow

Manipulate and manage the user password file.

ssh

Interact with ssh keys.

New States

user and group

Support for managing users and groups in Salt States.

mount

Enforce mounts and the fstab.

New Returners

mongo_return

Send the return information to a MongoDB server.

New Runners

manage

Display minions that are up or down.

28.2.138 Salt 0.9.0 Release Notes

release 2011-08-27

Salt 0.9.0 is here. This is an exciting release, 0.9.0 includes the new network topology features allowing peer salt commands and masters of masters via the syndic interface.

0.9.0 also introduces many more modules, improvements to the API and improvements to the ZeroMQ systems.

Download!

The Salt source can be downloaded from PyPI:

https://pypi.python.org/packages/source/s/salt/salt-0.9.0.tar.gz

Here is the md5sum:

9a925da04981e65a0f237f2e77ddab37

For instructions on how to set up Salt please see the *Installation* instructions.

New Features

Salt Syndic

The new *Syndic interface* allows a master to be commanded via another higher level salt master. This is a powerful solution allowing a master control structure to exist, allowing salt to scale to much larger levels then before.

Peer Communication

0.9.0 introduces the capability for a minion to call a publication on the master and receive the return from another set of minions. This allows salt to act as a communication channel between minions and as a general infrastructure message bus.

Peer communication is turned off by default but can be enabled via the `peer` option in the master configuration file. Documentation on the new *Peer interface*.

Easily Extensible API

The minion and master classes have been redesigned to allow for specialized minion and master servers to be easily created. An example on how this is done for the master can be found in the `master.py` salt module:

https://github.com/saltstack/salt/blob/master/salt/master.py

The `Master` class extends the `SMaster` class and set up the main master server.

The minion functions can now also be easily added to another application via the `SMinion` class, this class can be found in the `minion.py` module:

https://github.com/saltstack/salt/blob/master/salt/minion.py

Cleaner Key Management

This release changes some of the key naming to allow for multiple master keys to be held based on the type of minion gathering the master key.

The -d option has also been added to the salt-key command allowing for easy removal of accepted public keys.

The –gen-keys option is now available as well for salt-key, this allows for a salt specific RSA key pair to be easily generated from the command line.

Improved 0MQ Master Workers

The 0MQ worker system has been further refined to be faster and more robust. This new system has been able to handle a much larger load than the previous setup. The new system uses the IPC protocol in 0MQ instead of TCP.

New Modules

Quite a few new modules have been made.

New Minion Modules

apache

Work directly with apache servers, great for managing balanced web servers

cron

Read out the contents of a systems crontabs

mdadm

Module to manage raid devices in Linux, appears as the `raid` module

mysql

Gather simple data from MySQL databases

ps

Extensive utilities for managing processes

publish

Used by the peer interface to allow minions to make publications

28.2.139 Salt 0.9.1 Release Notes

release 2011-08-29

28.2.140 Salt 0.9.2 Release Notes

release 2011-09-17

Salt 0.9.2 has arrived! 0.9.2 is primarily a bugfix release, the exciting component in 0.9.2 is greatly improved support for salt states. All of the salt states interfaces have been more thoroughly tested and the new salt-states git repo is growing with example of how to use states.

This release introduces salt states for early developers and testers to start helping us clean up the states interface and make it ready for the world!

0.9.2 also fixes a number of bugs found on Python 2.6.

Download!

The Salt source can be downloaded from PyPI:

https://pypi.python.org/packages/source/s/salt/salt-0.9.2.tar.gz

For instructions on how to set up Salt please see the *Installation* instructions.

New Features

Salt-Call Additions

The salt-call command has received an overhaul, it now hooks into the outputter system so command output looks clean, and the logging system has been hooked into salt-call, so the -l option allows the logging output from salt minion functions to be displayed.

The end result is that the salt-call command can execute the state system and return clean output:

```
# salt-call state.highstate
```

State System Fixes

The state system has been tested and better refined. As of this release the state system is ready for early testers to start playing with. If you are interested in working with the state system please check out the (still very small) salt-states GitHub repo:

https://github.com/saltstack/salt-states

This git repo is the active development branch for determining how a clean salt-state database should look and act. Since the salt state system is still very young a lot of help is still needed here. Please fork the salt-states repo and help us develop a truly large and scalable system for configuration management!

Notable Bug Fixes

Python 2.6 String Formatting

Python 2.6 does not support format strings without an index identifier, all of them have been repaired.

Cython Loading Disabled by Default

Cython loading requires a development tool chain to be installed on the minion, requiring this by default can cause problems for most Salt deployments. If Cython auto loading is desired it will need to be turned on in the minion config.

28.2.141 Salt 0.9.3 Release Notes

release 2011-11-05

Salt 0.9.3 is finally arrived. This is another big step forward for Salt, new features range from proper FreeBSD support to fixing issues seen when attaching a minion to a master over the Internet.

The biggest improvements in 0.9.3 though can be found in the state system, it has progressed from something ready for early testers to a system ready to compete with platforms such as Puppet and Chef. The backbone of the state system has been greatly refined and many new features are available.

Download!

The Salt source can be downloaded from PyPI:

https://pypi.python.org/packages/source/s/salt/salt-0.9.3.tar.gz

For instructions on how to set up Salt please see the *Installation* instructions.

New Features

WAN Support

Recently more people have been testing Salt minions connecting to Salt Masters over the Internet. It was found that Minions would commonly loose their connection to the master when working over the internet. The minions can now detect if the connection has been lost and reconnect to the master, making WAN connections much more reliable.

State System Fixes

Substantial testing has gone into the state system and it is ready for real world usage. A great deal has been added to the documentation for states and the modules and functions available to states have been cleanly documented.

A number of State System bugs have also been founds and repaired, the output from the state system has also been refined to be extremely clear and concise.

Error reporting has also been introduced, issues found in sls files will now be clearly reported when executing Salt States.

Extend Declaration

The Salt States have also gained the `extend` declaration. This declaration allows for states to be cleanly modified in a post environment. Simply said, if there is an apache.sls file that declares the apache service, then another sls can include apache and then extend it:

```
include:
  - apache

extend:
  apache:
    service:
      - require:
        - pkg: mod_python

mod_python:
  pkg:
    - installed
```

The notable behavior with the extend functionality is that it literally extends or overwrites a declaration set up in another sls module. This means that Salt will behave as though the modifications were made directly to the apache sls. This ensures that the apache service in this example is directly tied to all requirements.

Highstate Structure Specification

This release comes with a clear specification of the Highstate data structure that is used to declare Salt States. This specification explains everything that can be declared in the Salt SLS modules.

The specification is extremely simple, and illustrates how Salt has been able to fulfill the requirements of a central configuration manager within a simple and easy to understand format and specification.

SheBang Renderer Switch

It came to our attention that having many renderers means that there may be a situation where more than one State Renderer should be available within a single State Tree.

The method chosen to accomplish this was something already familiar to developers and systems administrators, a SheBang. The Python State Renderer displays this new capability.

Python State Renderer

Until now Salt States could only be declared in yaml or json using Jinja or Mako. A new, very powerful, renderer has been added, making it possible to write Salt States in pure Python:

```
#!py

def run():
    '''
    Install the python-mako package
    '''
    return {'include': ['python'],
            'python-mako': {'pkg': ['installed']}}
```

This renderer is used by making a run function that returns the Highstate data structure. Any capabilities of Python can be used in pure Python sls modules.

This example of a pure Python sls module is the same as this example in yaml:

```
include:
  - python

python-mako:
  pkg:
    - installed
```

FreeBSD Support

Additional support has been added for FreeBSD, this is Salt's first branch out of the Linux world and proves the viability of Salt on non-Linux platforms.

Salt remote execution already worked on FreeBSD, and should work without issue on any Unix-like platform. But this support comes in the form of package management and user support, so Salt States also work on FreeBSD now.

The new freebsdpkg module provides package management support for FreeBSD and the new pw_user and pw_group provide user and group management.

Module and State Additions

Cron Support

Support for managing the system crontab has been added, declaring a cron state can be done easily:

```
date > /tmp/datestamp:
  cron:
    - present
    - user: fred
    - minute: 5
    - hour: 3
```

File State Additions

The file state has been given a number of new features, primarily the directory, recurse, symlink, and absent functions.

file.directory Make sure that a directory exists and has the right permissions.

```
/srv/foo:
  file:
    - directory
    - user: root
    - group: root
    - mode: 1755
```

file.symlink Make a symlink.

```
/var/lib/www:
  file:
    - symlink
    - target: /srv/www
    - force: True
```

file.recurse The recurse state function will recursively download a directory on the master file server and place it on the minion. Any change in the files on the master will be pushed to the minion. The recurse function is very powerful and has been tested by pushing out the full Linux kernel source.

```
/opt/code:
  file:
    - recurse
    - source: salt://linux
```

file.absent Make sure that the file is not on the system, recursively deletes directories, files, and symlinks.

```
/etc/httpd/conf.d/somebogusfile.conf:
  file:
    - absent
```

Sysctl Module and State

The sysctl module and state allows for sysctl components in the kernel to be managed easily. the sysctl module contains the following functions:

sysctl.show Return a list of sysctl parameters for this minion

sysctl.get Return a single sysctl parameter for this minion

sysctl.assign Assign a single sysctl parameter for this minion

sysctl.persist Assign and persist a simple sysctl parameter for this minion

The sysctl state allows for sysctl parameters to be assigned:

```
vm.swappiness:
  sysctl:
    - present
    - value: 20
```

Kernel Module Management

A module for managing Linux kernel modules has been added. The new functions are as follows:

kmod.available Return a list of all available kernel modules

kmod.check_available Check to see if the specified kernel module is available

kmod.lsmod Return a dict containing information about currently loaded modules

kmod.load Load the specified kernel module

kmod.remove Unload the specified kernel module

The kmod state can enforce modules be either present or absent:

```
kvm_intel:
  kmod:
    - present
```

Ssh Authorized Keys

The ssh_auth state can distribute ssh authorized keys out to minions. Ssh authorized keys can be present or absent.

```
AAAAB3NzaC1kc3MAAACBAL0sQ9fJ5bYTEyYvlRBsJdDOo49CNfhlWHWXQRqul6rwL4KIuPrhY7hBw0tV7UNC7J9IZRNO4iGod9C+
↪xw6NtnQZVMcmZIre5Elrw3OKgxcDNomjYFNHuOYaQLBBMosyO++tJe1KTAr3A2zGj2xbWO9JhEzu8xvSdF8jRu0N5SRXPpzSyU
↪x75wolBDbVzeTlxWxgxhafj7P6Ncdv25Wz9wvc6ko/puww0b3rcLNqK+XCNJlsM/
↪7lB8Q26iK5mRZzNsGeGwGTyzNIMBekGYQ5MRdIcPv5dBIP/1M6fQDEsAXQ==:
  ssh_auth:
    - present
    - user: frank
    - enc: dsa
    - comment: 'Frank's key'
```

28.2.142 Salt 0.9.4 Release Notes

release 2011-11-27

Salt 0.9.4 has arrived. This is a critical update that repairs a number of key bugs found in 0.9.3. But this update is not without feature additions as well! 0.9.4 adds support for Gentoo portage to the pkg module and state system. Also there are 2 major new state additions, the failhard option and the ability to set up finite state ordering with the `order` option.

This release also sees our largest increase in community contributions. These contributors have and continue to be the life blood of the Salt project, and the team continues to grow. I want to put out a big thanks to our new and existing contributors.

Download!

The Salt source can be downloaded from PyPI:

https://pypi.python.org/packages/source/s/salt/salt-0.9.4.tar.gz

For instructions on how to set up Salt please see the *Installation* instructions.

New Features

Failhard State Option

Normally, when a state fails Salt continues to execute the remainder of the defined states and will only refuse to execute states that require the failed state.

But the situation may exist, where you would want all state execution to stop if a single state execution fails. The capability to do this is called `failing hard`.

State Level Failhard

A single state can have a failhard set, this means that if this individual state fails that all state execution will immediately stop. This is a great thing to do if there is a state that sets up a critical config file and setting a require for each state that reads the config would be cumbersome. A good example of this would be setting up a package manager early on:

```
/etc/yum.repos.d/company.repo:
  file:
    - managed
    - source: salt://company/yumrepo.conf
    - user: root
    - group: root
    - mode: 644
    - order: 1
    - failhard: True
```

In this situation, the yum repo is going to be configured before other states, and if it fails to lay down the config file, than no other states will be executed.

Global Failhard

It may be desired to have failhard be applied to every state that is executed, if this is the case, then failhard can be set in the master configuration file. Setting failhard in the master configuration file will result in failing hard when any minion gathering states from the master have a state fail.

This is NOT the default behavior, normally Salt will only fail states that require a failed state.

Using the global failhard is generally not recommended, since it can result in states not being executed or even checked. It can also be confusing to see states failhard if an admin is not actively aware that the failhard has been set.

To use the global failhard set failhard: True in the master configuration

Finite Ordering of State Execution

When creating salt sls files, it is often important to ensure that they run in a specific order. While states will always execute in the same order, that order is not necessarily defined the way you want it.

A few tools exist in Salt to set up the correct state ordering, these tools consist of requisite declarations and order options.

The Order Option

Before using the order option, remember that the majority of state ordering should be done with requisite statements, and that a requisite statement will override an order option.

The order option is used by adding an order number to a state declaration with the option *order*:

```
vim:
  pkg:
    - installed
    - order: 1
```

By adding the order option to *1* this ensures that the vim package will be installed in tandem with any other state declaration set to the order *1*.

Any state declared without an order option will be executed after all states with order options are executed.

But this construct can only handle ordering states from the beginning. Sometimes you may want to send a state to the end of the line, to do this set the order to last:

```
vim:
  pkg:
    - installed
    - order: last
```

Substantial testing has gone into the state system and it is ready for real world usage. A great deal has been added to the documentation for states and the modules and functions available to states have been cleanly documented.

A number of State System bugs have also been founds and repaired, the output from the state system has also been refined to be extremely clear and concise.

Error reporting has also been introduced, issues found in sls files will now be clearly reported when executing Salt States.

Gentoo Support

Additional experimental support has been added for Gentoo. This is found in the contribution from Doug Renn, aka nestegg.

28.2.143 Salt 0.9.5 Release Notes

release 2012-01-15

Salt 0.9.5 is one of the largest steps forward in the development of Salt.

0.9.5 comes with many milestones, this release has seen the community of developers grow out to an international team of 46 code contributors and has many feature additions, feature enhancements, bug fixes and speed improvements.

Warning: Be sure to *read the upgrade instructions* about the switch to msgpack before upgrading!

Community

Nothing has proven to have more value to the development of Salt that the outstanding community that has been growing at such a great pace around Salt. This has proven not only that Salt has great value, but also the expandability of Salt is as exponential as I originally intended.

0.9.5 has received over 600 additional commits since 0.9.4 with a swath of new committers. The following individuals have contributed to the development of 0.9.5:

- Aaron Bull Schaefer
- Antti Kaihola
- Bas Tichelaar
- Brad Barden
- Brian Wagner
- Byron Clark
- Chris Scheller
- Christer Edwards
- Clint Savage

- Corey Quinn
- David Boucha
- Eivind Uggedal
- Eric Poelke
- Evan Borgstrom
- Jed Glazner
- Jeff Schroeder
- Jeffrey C. Ollie
- Jonas Buckner
- Kent Tenney
- Martin Schnabel
- Maxim Burgerhout
- Mitch Anderson
- Nathaniel Whiteinge
- Seth House
- Thomas S Hatch
- Thomas Schreiber
- Tor Hveem
- lzyeval
- syphernl

This makes 21 new developers since 0.9.4 was released!

To keep up with the growing community follow Salt on Ohloh (http://www.ohloh.net/p/salt), to join the Salt development community, fork Salt on GitHub, and get coding (https://github.com/saltstack/salt)!

Major Features

SPEED! Pickle to msgpack

For a few months now we have been talking about moving away from Python pickles for network serialization, but a preferred serialization format had not yet been found. After an extensive performance testing period involving everything from JSON to protocol buffers, a clear winner emerged. Message Pack (http://msgpack.org/) proved to not only be the fastest and most compact, but also the most "salt like". Message Pack is simple, and the code involved is very small. The msgpack library for Python has been added directly to Salt.

This move introduces a few changes to Salt. First off, Salt is no longer a "noarch" package, since the msgpack lib is written in C. Salt 0.9.5 will also have compatibility issues with 0.9.4 with the default configuration.

We have gone through great lengths to avoid backwards compatibility issues with Salt, but changing the serialization medium was going to create issues regardless. Salt 0.9.5 is somewhat backwards compatible with earlier minions. A 0.9.5 master can command older minions, but only if the `serial` config value in the master is set to `pickle`. This will tell the master to publish messages in pickle format and will allow the master to receive messages in both msgpack and pickle formats.

Therefore **the suggested methods for upgrading** are either to just upgrade everything at once, or:

1. Upgrade the master to 0.9.5
2. Set `serial` to `pickle` in the master config
3. Upgrade the minions
4. Remove the `serial` option from the master config

Since pickles can be used as a security exploit the ability for a master to accept pickles from minions at all will be removed in a future release.

C Bindings for YAML

All of the YAML rendering is now done with the YAML C bindings. This speeds up all of the sls files when running states.

Experimental Windows Support

David Boucha has worked tirelessly to bring initial support to Salt for Microsoft Windows operating systems. Right now the Salt Minion can run as a native Windows service and accept commands.

In the weeks and months to come Windows will receive the full treatment and will have support for Salt States and more robust support for managing Windows systems. This is a big step forward for Salt to move entirely outside of the Unix world, and proves Salt is a viable cross platform solution. Big Thanks to Dave for his contribution here!

Dynamic Module Distribution

Many Salt users have expressed the desire to have Salt distribute in-house modules, states, renderers, returners, and grains. This support has been added in a number of ways:

Modules via States

Now when salt modules are deployed to a minion via the state system as a file, then the modules will be automatically loaded into the active running minion - no restart required - and into the active running state. So custom state modules can be deployed and used in the same state run.

Modules via Module Environment Directories

Under the file_roots each environment can now have directories that are used to deploy large groups of modules. These directories sync modules at the beginning of a state run on the minion, or can be manually synced via the Salt module *`salt.modules.saltutil.sync_all`*.

The directories are named:

- `_modules`
- `_states`
- `_grains`
- `_renderers`
- `_returners`

The modules are pushed to their respective scopes on the minions.

Module Reloading

Modules can now be reloaded without restarting the minion, this is done by calling the `salt.modules.sys.reload_modules` function.

But wait, there's more! Now when a salt module of any type is added via states the modules will be automatically reloaded, allowing for modules to be laid down with states and then immediately used.

Finally, all modules are reloaded when modules are dynamically distributed from the salt master.

Enable / Disable Added to Service

A great deal of demand has existed for adding the capability to set services to be started at boot in the service module. This feature also comes with an overhaul of the service modules and initial systemd support.

This means that the *`service state`* can now accept `- enable: True` to make sure a service is enabled at boot, and `- enable: False` to make sure it is disabled.

Compound Target

A new target type has been added to the lineup, the compound target. In previous versions the desired minions could only be targeted via a single specific target type, but now many target specifications can be declared.

These targets can also be separated by and/or operators, so certain properties can be used to omit a node:

```
salt -C 'webserv* and G@os:Debian or E@db.*' test.ping
```

will match all minions with ids starting with webserv via a glob and minions matching the `os:Debian` grain. Or minions that match the `db.*` regular expression.

Node Groups

Often the convenience of having a predefined group of minions to execute targets on is desired. This can be accomplished with the new nodegroups feature. Nodegroups allow for predefined compound targets to be declared in the master configuration file:

```
nodegroups:
  group1: 'L@foo.domain.com,bar.domain.com,baz.domain.com and bl*.domain.com'
  group2: 'G@os:Debian and foo.domain.com'
```

And then used via the `-N` option:

```
salt -N group1 test.ping
```

Minion Side Data Store

The data module introduces the initial approach into storing persistent data on the minions, specific to the minions. This allows for data to be stored on minions that can be accessed from the master or from the minion.

The Minion datastore is young, and will eventually provide an interface similar to a more mature key/value pair server.

Major Grains Improvement

The Salt grains have been overhauled to include a massive amount of extra data. this includes hardware data, os data and salt specific data.

Salt -Q is Useful Now

In the past the salt query system, which would display the data from recent executions would be displayed in pure Python, and it was unreadable.

0.9.5 has added the outputter system to the `-Q` option, thus enabling the salt query system to return readable output.

Packaging Updates

Huge strides have been made in packaging Salt for distributions. These additions are thanks to our wonderful community where the work to set up packages has proceeded tirelessly.

FreeBSD

Salt on FreeBSD? There a port for that:

http://svnweb.freebsd.org/ports/head/sysutils/py-salt/

This port was developed and added by Christer Edwards. This also marks the first time Salt has been included in an upstream packaging system!

Fedora and Red Hat Enterprise

Salt packages have been prepared for inclusion in the Fedora Project and in EPEL for Red Hat Enterprise 5 and 6. These packages are the result of the efforts made by Clint Savage (herlo).

Debian/Ubuntu

A team of many contributors have assisted in developing packages for Debian and Ubuntu. Salt is still actively seeking inclusion in upstream Debian and Ubuntu and the package data that has been prepared is being pushed through the needed channels for inclusion.

These packages have been prepared with the help of:

- Corey
- Aaron Toponce
- and‘

More to Come

We are actively seeking inclusion in more distributions. Primarily getting Salt into Gentoo, SUSE, OpenBSD, and preparing Solaris support are all turning into higher priorities.

Refinement

Salt continues to be refined into a faster, more stable and more usable application. 0.9.5 comes with more debug logging, more bug fixes and more complete support.

More Testing, More BugFixes

0.9.5 comes with more bugfixes due to more testing than any previous release. The growing community and the introduction a dedicated QA environment have unearthed many issues that were hiding under the covers. This has further refined and cleaned the state interface, taking care of things from minor visual issues to repairing misleading data.

Custom Exceptions

A custom exception module has been added to throw salt specific exceptions. This allows Salt to give much more granular error information.

New Modules

`data`

The new data module manages a persistent datastore on the minion. Big thanks to bastichelaar for his help refining this module

`freebsdkmod`

FreeBSD kernel modules can now be managed in the same way Salt handles Linux kernel modules.

This module was contributed thanks to the efforts of Christer Edwards

`gentoo_service`

Support has been added for managing services in Gentoo. Now Gentoo services can be started, stopped, restarted, enabled, disabled, and viewed.

`pip`

The pip module introduces management for pip installed applications. Thanks goes to whitinge for the addition of the pip module

`rh_service`

The rh_service module enables Red Hat and Fedora specific service management. Now Red Hat like systems come with extensive management of the classic init system used by Red Hat

saltutil

The saltutil module has been added as a place to hold functions used in the maintenance and management of salt itself. Saltutil is used to salt the salt minion. The saltutil module is presently used only to sync extension modules from the master server.

systemd

Systemd support has been added to Salt, now systems using this next generation init system are supported on systems running systemd.

virtualenv

The virtualenv module has been added to allow salt to create virtual Python environments. Thanks goes to whitinge for the addition of the virtualenv module

win_disk

Support for gathering disk information on Microsoft Windows minions The windows modules come courtesy of Utah_Dave

win_service

The win_service module adds service support to Salt for Microsoft Windows services

win_useradd

Salt can now manage local users on Microsoft Windows Systems

yumpkg5

The yumpkg module introduces in 0.9.4 uses the yum API to interact with the yum package manager. Unfortunately, on Red Hat 5 systems salt does not have access to the yum API because the yum API is running under Python 2.4 and Salt needs to run under Python 2.6.

The yumpkg5 module bypasses this issue by shelling out to yum on systems where the yum API is not available.

New States

mysql_database

The new mysql_database state adds the ability to systems running a mysql server to manage the existence of mysql databases.

The mysql states are thanks to syphernl

`mysql_user`

The mysql_user state enables mysql user management.

`virtualenv`

The virtualenv state can manage the state of Python virtual environments. Thanks to Whitinge for the virtualenv state

New Returners

`cassandra_returner`

A returner allowing Salt to send data to a cassandra server. Thanks to Byron Clark for contributing this returner

28.2.144 Salt 0.9.6 Release Notes

release 2012-01-21

Salt 0.9.6 is a release targeting a few bugs and changes. This is primarily targeting an issue found in the names declaration in the state system. But a few other bugs were also repaired, like missing support for grains in extmods.

Due to a conflict in distribution packaging msgpack will no longer be bundled with Salt, and is required as a dependency.

New Features

HTTP and ftp support in files.managed

Now under the source option in the file.managed state a HTTP or ftp address can be used instead of a file located on the salt master.

Allow Multiple Returners

Now the returner interface can define multiple returners, and will also return data back to the master, making the process less ambiguous.

Minion Memory Improvements

A number of modules have been taken out of the minion if the underlying systems required by said modules are not present on the minion system. A number of other modules need to be stripped out in this same way which should continue to make the minion more efficient.

Minions Can Locally Cache Return Data

A new option, cache_jobs, has been added to the minion to allow for all of the historically run jobs to cache on the minion, allowing for looking up historic returns. By default cache_jobs is set to False.

Pure Python Template Support For file.managed

Templates in the file.managed state can now be defined in a Python script. This script needs to have a run function that returns the string that needs to be in the named file.

28.2.145 Salt 0.9.7 Release Notes

release 2012-02-15

Salt 0.9.7 is here! The latest iteration of Salt brings more features and many fixes. This release is a great refinement over 0.9.6, adding many conveniences under the hood, as well as some features that make working with Salt much better.

A few highlights include the new Job system, refinements to the requisite system in states, the `mod_init` interface for states, external node classification, search path to managed files in the file state, and refinements and additions to dynamic module loading.

0.9.7 also introduces the long developed (and oft changed) unit test framework and the initial unit tests.

Major Features

Salt Jobs Interface

The new jobs interface makes the management of running executions much cleaner and more transparent. Building on the existing execution framework the jobs system allows clear introspection into the active running state of the running Salt interface.

The Jobs interface is centered in the new minion side proc system. The minions now store msgpack serialized files under `/var/cache/salt/proc`. These files keep track of the active state of processes on the minion.

Functions in the saltutil Module

A number of functions have been added to the saltutil module to manage and view the jobs:

`running` - Returns the data of all running jobs that are found in the proc directory.

`find_job` - Returns specific data about a certain job based on job id.

`signal_job` - Allows for a given jid to be sent a signal.

`term_job` - Sends a termination signal (`SIGTERM, 15`) to the process controlling the specified job.

`kill_job` Sends a kill signal (`SIGKILL, 9`) to the process controlling the specified job.

The jobs Runner

A convenience runner front end and reporting system has been added as well. The jobs runner contains functions to make viewing data easier and cleaner.

The jobs runner contains a number of functions…

active

The active function runs `saltutil.running` on all minions and formats the return data about all running jobs in a much more usable and compact format. The active function will also compare jobs that have returned and jobs that are still running, making it easier to see what systems have completed a job and what systems are still being waited on.

lookup_jid

When jobs are executed the return data is sent back to the master and cached. By default is cached for 24 hours, but this can be configured via the `keep_jobs` option in the master configuration.

Using the `lookup_jid` runner will display the same return data that the initial job invocation with the salt command would display.

list_jobs

Before finding a historic job, it may be required to find the job id. `list_jobs` will parse the cached execution data and display all of the job data for jobs that have already, or partially returned.

External Node Classification

Salt can now use external node classifiers like Cobbler's `cobbler-ext-nodes`.

Salt uses specific data from the external node classifier. In particular the classes value denotes which sls modules to run, and the environment value sets to another environment.

An external node classification can be set in the master configuration file via the `external_nodes` option: https://salt.readthedocs.io/en/latest/ref/configuration/master.html#external-nodes

External nodes are loaded in addition to the top files. If it is intended to only use external nodes, do not deploy any top files.

State Mod Init System

An issue arose with the pkg state. Every time a package was run Salt would need to refresh the package database. This made systems with slower package metadata refresh speeds much slower to work with. To alleviate this issue the `mod_init` interface has been added to salt states.

The `mod_init` interface is a function that can be added to a state file. This function is called with the first state called. In the case of the pkg state, the `mod_init` function sets up a tag which makes the package database only refresh on the first attempt to install a package.

In a nutshell, the `mod_init` interface allows a state to run any command that only needs to be run once, or can be used to set up an environment for working with the state.

Source File Search Path

The file state continues to be refined, adding speed and capabilities. This release adds the ability to pass a list to the source option. This list is then iterated over until the source file is found, and the first found file is used.

The new syntax looks like this:

```
/etc/httpd/conf/httpd.conf:
  file:
    - managed
    - source:
      - salt://httpd/httpd.conf
      - http://myserver/httpd.conf: md5=8c1fe119e6f1fd96bc06614473509bf1
```

The source option can take sources in the list from the salt file server as well as an arbitrary web source. If using an arbitrary web source the checksum needs to be passed as well for file verification.

Refinements to the Requisite System

A few discrepancies were still lingering in the requisite system, in particular, it was not possible to have a `require` and a `watch` requisite declared in the same state declaration.

This issue has been alleviated, as well as making the requisite system run more quickly.

Initial Unit Testing Framework

Because of the module system, and the need to test real scenarios, the development of a viable unit testing system has been difficult, but unit testing has finally arrived. Only a small amount of unit testing coverage has been developed, much more coverage will be in place soon.

A huge thanks goes out to those who have helped with unit testing, and the contributions that have been made to get us where we are. Without these contributions unit tests would still be in the dark.

Compound Targets Expanded

Originally only support for `and` and `or` were available in the compound target. 0.9.7 adds the capability to negate compound targets with `not`.

Nodegroups in the Top File

Previously the nodegroups defined in the master configuration file could not be used to match nodes for states. The nodegroups support has been expanded and the nodegroups defined in the master configuration can now be used to match minions in the top file.

28.2.146 Salt 0.9.8 Release Notes

release 2012-03-21

Salt 0.9.8 is a big step forward, with many additions and enhancements, as well as a number of precursors to advanced future developments.

This version of Salt adds much more power to the command line, making the old hard timeout issues a thing of the past and adds keyword argument support. These additions are also available in the salt client API, making the available API tools much more powerful.

The new pillar system allows for data to be stored on the master and assigned to minions in a granular way similar to the state system. It also allows flexibility for users who want to keep data out of their state tree similar to 'external lookup' functionality in other tools.

A new way to extend requisites was added, the ”requisite in” statement. This makes adding requires or watch statements to external state decs much easier.

Additions to requisites making them much more powerful have been added as well as improved error checking for sls files in the state system. A new provider system has been added to allow for redirecting what modules run in the background for individual states.

Support for openSUSE has been added and support for Solaris has begun serious development. Windows support has been significantly enhanced as well.

The matcher and target systems have received a great deal of attention. The default behavior of grain matching has changed slightly to reflect the rest of salt and the compound matcher system has been refined.

A number of impressive features with keyword arguments have been added to both the CLI and to the state system. This makes states much more powerful and flexible while maintaining the simple configuration everyone loves.

The new batch size capability allows for executions to be rolled through a group of targeted minions a percentage or specific number at a time. This was added to prevent the ”thundering herd” problem when targeting large numbers of minions for things like service restarts or file downloads.

Upgrade Considerations

Upgrade Issues

There was a previously missed oversight which could cause a newer minion to crash an older master. That oversight has been resolved so the version incompatibility issue will no longer occur. When upgrading to 0.9.8 make sure to upgrade the master first, followed by the minions.

Debian/Ubuntu Packages

The original Debian/Ubuntu packages were called salt and included all salt applications. New packages in the ppa are split by function. If an old salt package is installed then it should be manually removed and the new split packages need to be freshly installed.

On the master:

```
# apt-get purge salt
# apt-get install salt-{master,minion}
```

On the minions:

```
# apt-get purge salt
# apt-get install salt-minion
```

And on any Syndics:

```
# apt-get install salt-syndic
```

The official Salt PPA for Ubuntu is located at: https://launchpad.net/~saltstack/+archive/salt

Major Features

Pillar

Pillar offers an interface to declare variable data on the master that is then assigned to the minions. The pillar data is made available to all modules, states, sls files etc. It is compiled on the master and is declared using the existing renderer system. This means that learning pillar should be fairly trivial to those already familiar with salt states.

CLI Additions

The `salt` command has received a serious overhaul and is more powerful than ever. Data is returned to the terminal as it is received, and the salt command will now wait for all running minions to return data before stopping. This makes adding very large *–timeout* arguments completely unnecessary and gets rid of long running operations returning empty `{}` when the timeout is exceeded.

When calling salt via sudo, the user originally running salt is saved to the log for auditing purposes. This makes it easy to see who ran what by just looking through the minion logs.

The *salt-key* command gained the *-D* and *–delete-all* arguments for removing all keys. Be careful with this one!

Running States Without a Master

The addition of running states without a salt-master has been added to 0.9.8. This feature allows for the unmodified salt state tree to be read locally from a minion. The result is that the UNMODIFIED state tree has just become portable, allowing minions to have a local copy of states or to manage states without a master entirely.

This is accomplished via the new file client interface in Salt that allows for the `salt://` URI to be redirected to custom interfaces. This means that there are now two interfaces for the salt file server, calling the master or looking in a local, minion defined `file_roots`.

This new feature can be used by modifying the minion config to point to a local `file_roots` and setting the `file_client` option to `local`.

Keyword Arguments and States

State modules now accept the `**kwargs` argument. This results in all data in a sls file assigned to a state being made available to the state function.

This passes data in a transparent way back to the modules executing the logic. In particular, this allows adding arguments to the `pkg.install` module that enable more advanced and granular controls with respect to what the state is capable of.

An example of this along with the new debconf module for installing ldap client packages on Debian:

```yaml
ldap-client-packages:
  pkg:
    - debconf: salt://debconf/ldap-client.ans
    - installed
    - names:
      - nslcd
      - libpam-ldapd
      - libnss-ldapd
```

Keyword Arguments and the CLI

In the past it was required that all arguments be passed in the proper order to the *salt* and *salt-call* commands. As of 0.9.8, keyword arguments can be passed in the form of `kwarg=argument`.

```
# salt -G 'type:dev' git.clone \
    repository=https://github.com/saltstack/salt.git cwd=/tmp/salt user=jeff
```

Matcher Refinements and Changes

A number of fixes and changes have been applied to the Matcher system. The most noteworthy is the change in the grain matcher. The grain matcher used to use a regular expression to match the passed data to a grain, but now defaults to a shell glob like the majority of match interfaces in Salt. A new option is available that still uses the old style regex matching to grain data called `grain-pcre`. To use regex matching in compound matches use the letter *P*.

For example, this would match any ArchLinux or Fedora minions:

```
# salt --grain-pcre 'os:(Arch:Fed).*' test.ping
```

And the associated compound matcher suitable for `top.sls` is *P*:

```
P@os:(Arch|Fed).*
```

NOTE: Changing the grains matcher from pcre to glob is backwards incompatible.

Support has been added for matching minions with Yahoo's range library. This is handled by passing range syntax with *-R* or *–range* arguments to salt.

More information at: https://github.com/ytoolshed/range/wiki/%22yamlfile%22-module-file-spec

Requisite "in"

A new means to updating requisite statements has been added to make adding watchers and requires to external states easier. Before 0.9.8 the only way to extend the states that were watched by a state outside of the sls was to use an extend statement:

```
include:
  - http
extend:
  apache:
    service:
      - watch:
        - pkg: tomcat

tomcat:
  pkg:
    - installed
```

But the new `Requisite in` statement allows for easier extends for requisites:

```
include:
  - http
```

(continues on next page)

(continued from previous page)

```
tomcat:
  pkg:
    - installed
    - watch_in:
      - service: apache
```

Requisite in is part of the extend system, so still remember to always include the sls that is being extended!

Providers

Salt predetermines what modules should be mapped to what uses based on the properties of a system. These determinations are generally made for modules that provide things like package and service management. The apt module maps to pkg on Debian and the yum module maps to pkg on Fedora for instance.

Sometimes in states, it may be necessary for a non-default module to be used for the desired functionality. For instance, an Arch Linux system may have been set up with systemd support. Instead of using the default service module detected for Arch Linux, the systemd module can be used:

```
http:
  service:
    - running
    - enable: True
    - provider: systemd
```

Default providers can also be defined in the minion config file:

```
providers:
  service: systemd
```

When default providers are passed in the minion config, then those providers will be applied to all functionality in Salt, this means that the functions called by the minion will use these modules, as well as states.

Requisite Glob Matching

Requisites can now be defined with glob expansion. This means that if there are many requisites, they can be defined on a single line.

To watch all files in a directory:

```
http:
  service:
    - running
    - enable: True
    - watch:
      - file: /etc/http/conf.d/*
```

This example will watch all defined files that match the glob `/etc/http/conf.d/*`

Batch Size

The new batch size option allows commands to be executed while maintaining that only so many hosts are executing the command at one time. This option can take a percentage or a finite number:

```
salt '*' -b 10 test.ping

salt -G 'os:RedHat' --batch-size 25% apache.signal restart
```

This will only run test.ping on 10 of the targeted minions at a time and then restart apache on 25% of the minions matching `os:RedHat` at a time and work through them all until the task is complete. This makes jobs like rolling web server restarts behind a load balancer or doing maintenance on BSD firewalls using carp much easier with salt.

Module Updates

This is a list of notable, but non-exhaustive updates with new and existing modules.

Windows support has seen a flurry of support this release cycle. We've gained all new `file`, *`network`*, and *`shadow`* modules. Please note that these are still a work in progress.

For our ruby users, new *`rvm`* and *`gem`* modules have been added along with the *`associated states`*

The *`virt`* module gained basic Xen support.

The *`yum`* module gained Scientific Linux support.

The *`pkg`* module on Debian, Ubuntu, and derivatives force apt to run in a non-interactive mode. This prevents issues when package installation waits for confirmation.

A `pkg` module for OpenSUSE's zypper was added.

The `service` module on Ubuntu natively supports upstart.

A new *`debconf`* module was contributed by our community for more advanced control over deb package deployments on Debian based distributions.

The *`mysql.user`* state and *`mysql`* module gained a *password_hash* argument.

The *`cmd`* module and state gained a *shell* keyword argument for specifying a shell other than `/bin/sh` on Linux / Unix systems.

New *`git`* and *`mercurial`* modules have been added for fans of distributed version control.

In Progress Development

Master Side State Compiling

While we feel strongly that the advantages gained with minion side state compiling are very critical, it does prevent certain features that may be desired. 0.9.8 has support for initial master side state compiling, but many more components still need to be developed, it is hoped that these can be finished for 0.9.9.

The goal is that states can be compiled on both the master and the minion allowing for compilation to be split between master and minion. Why will this be great? It will allow storing sensitive data on the master and sending it to some minions without all minions having access to it. This will be good for handling ssl certificates on front-end web servers for instance.

Solaris Support

Salt 0.9.8 sees the introduction of basic Solaris support. The daemon runs well, but grains and more of the modules need updating and testing.

Windows Support

Salt states on windows are now much more viable thanks to contributions from our community! States for file, service, local user, and local group management are more fully fleshed out along with network and disk modules. Windows users can also now manage registry entries using the new "reg" module.

28.2.147 Salt 0.9.9 Release Notes

release 2012-04-27

0.9.9 is out and comes with some serious bug fixes and even more serious features. This release is the last major feature release before 1.0.0 and could be considered the 1.0.0 release candidate.

A few updates include more advanced kwargs support, the ability for salt states to more safely configure a running salt minion, better job directory management and the new state test interface.

Many new tests have been added as well, including the new minion swarm test that allows for easier testing of Salt working with large groups of minions. This means that if you have experienced stability issues with Salt before, particularly in larger deployments, that these bugs have been tested for, found, and killed.

Major Features

State Test Interface

Until 0.9.9 the only option when running states to see what was going to be changed was to print out the highstate with state.show_highstate and manually look it over. But now states can be run to discover what is going to be changed.

Passing the option `test=True` to many of the state functions will now cause the salt state system to only check for what is going to be changed and report on those changes.

```
salt '*' state.highstate test=True
```

Now states that would have made changes report them back in yellow.

State Syntax Update

A shorthand syntax has been added to sls files, and it will be the default syntax in documentation going forward. The old syntax is still fully supported and will not be deprecated, but it is recommended to move to the new syntax in the future. This change moves the state function up into the state name using a dot notation. This is in-line with how state functions are generally referred to as well:

The new way:

```
/etc/sudoers:
  file.present:
    - source: salt://sudo/sudoers
    - user: root
    - mode: 400
```

Use and Use_in Requisites

Two new requisite statements are available in 0.9.9. The use and use_in requisite and requisite-in allow for the transparent duplication of data between states. When a state "uses" another state it copies the other state's arguments as defaults. This was created in direct response to the new network state, and allows for many network interfaces to be configured in the same way easily. A simple example:

```
root_file:
  file.absent:
    - name: /tmp/nothing
    - user: root
    - mode: 644
    - group: root
    - use_in:
      - file: /etc/vimrc

fred_file:
  file.absent:
    - name: /tmp/nothing
    - user: fred
    - group: marketing
    - mode: 660

/files/marketing/district7.rst:
  file.present:
    - source: salt://marketing/district7.rst
    - template: jinja
    - use:
      - file: fred_file

/etc/vimrc:
  file.present:
    - source: salt://edit/vimrc
```

This makes the 2 lower state decs inherit the options from their respectively "used" state decs.

Network State

The new network state allows for the configuration of network devices via salt states and the ip salt module. This addition has been given to the project by Jeff Hutchins and Bret Palsson from Jive Communications.

Currently the only network configuration backend available is for Red Hat based systems, like Red Hat Enterprise, CentOS, and Fedora.

Exponential Jobs

Originally the jobs executed were stored on the master in the format: `<cachedir>/jobs/jid/{minion ids}` But this format restricted the number of jobs in the cache to the number of subdirectories allowed on the filesystem. Ext3 for instance limits subdirectories to 32000. To combat this the new format for 0.9.9 is: `<cachedir>/jobs/jid_hash[:2]/jid_hash[2:]/{minion ids}` So that now the number of maximum jobs that can be run before the cleanup cycle hits the job directory is substantially higher.

ssh_auth Additions

The original ssh_auth state was limited to accepting only arguments to apply to a public key, and the key itself. This was restrictive due to the way the we learned that many people were using the state, so the key section has been expanded to accept options and arguments to the key that over ride arguments passed in the state. This gives substantial power to using ssh_auth with names:

```
sshkeys:
  ssh_auth:
    - present
    - user: backup
    - enc: ssh-dss
    - options:
      - option1="value1"
      - option2="value2 flag2"
    - comment: backup
    - names:
      -⊠
↪AAAAB3NzaC1yc2EAAAABIwAAAQEAlyE26SMFFVY5YJvnL7AF5CRTPtAigSW1U887ASfBt6FDa7Qr1YdO5ochiLoz8aSiMKd5h4
↪SMjq2aycHI+abiVDn3sciQjsLsNW59t48Udivl2RjWG7Eo+LYiB17MKD5M40r5CP2K4B8nuL+r4oAZEHKOJUF3rzA20MZXHRQu
↪I4bz/l0UdGh18SpMB8zVnT3YF5nukQQ/ATspmhpU66s4ntMehULC+ljLvZL40ByNmF0TZc2sdSkA0111==
      -⊠
↪AAAAB3NzaC1yc2EAAAABIwAAAQEAlyE26SMFFVY5YJvnL7AF5CRTPtAigSW1U887ASfBt6FDa7Qr1YdO5ochiLoz8aSiMKd5h4
↪SMjq2aycHI+abiVDn3sciQjsLsNW59t48Udivl2RjWG7Eo+LYiB17MKD5M40r5CP2K4B8nuL+r4oAZEHKOJUF3rzA20MZXHRQu
↪I4bz/l0UdGh18SpMB8zVnT3YF5nukQQ/ATspmhpU66s4ntMehULC+ljLvZL40ByNmF0TZc2sdSkA0222==⊠
↪override
      - ssh-rsa⊠
↪AAAAB3NzaC1yc2EAAAABIwAAAQEAlyE26SMFFVY5YJvnL7AF5CRTPtAigSW1U887ASfBt6FDa7Qr1YdO5ochiLoz8aSiMKd5h4
↪SMjq2aycHI+abiVDn3sciQjsLsNW59t48Udivl2RjWG7Eo+LYiB17MKD5M40r5CP2K4B8nuL+r4oAZEHKOJUF3rzA20MZXHRQu
↪I4bz/l0UdGh18SpMB8zVnT3YF5nukQQ/ATspmhpU66s4ntMehULC+ljLvZL40ByNmF0TZc2sdSkA0333==⊠
↪override
      - ssh-rsa⊠
↪AAAAB3NzaC1yc2EAAAABIwAAAQEAlyE26SMFFVY5YJvnL7AF5CRTPtAigSW1U887ASfBt6FDa7Qr1YdO5ochiLoz8aSiMKd5h4
↪SMjq2aycHI+abiVDn3sciQjsLsNW59t48Udivl2RjWG7Eo+LYiB17MKD5M40r5CP2K4B8nuL+r4oAZEHKOJUF3rzA20MZXHRQu
↪I4bz/l0UdGh18SpMB8zVnT3YF5nukQQ/ATspmhpU66s4ntMehULC+ljLvZL40ByNmF0TZc2sdSkA0444==
      - option3="value3",option4="value4 flag4" ssh-rsa⊠
↪AAAAB3NzaC1yc2EAAAABIwAAAQEAlyE26SMFFVY5YJvnL7AF5CRTPtAigSW1U887ASfBt6FDa7Qr1YdO5ochiLoz8aSiMKd5h4
↪SMjq2aycHI+abiVDn3sciQjsLsNW59t48Udivl2RjWG7Eo+LYiB17MKD5M40r5CP2K4B8nuL+r4oAZEHKOJUF3rzA20MZXHRQu
↪I4bz/l0UdGh18SpMB8zVnT3YF5nukQQ/ATspmhpU66s4ntMehULC+ljLvZL40ByNmF0TZc2sdSkA0555==⊠
↪override
      - option3="value3" ssh-rsa⊠
↪AAAAB3NzaC1yc2EAAAABIwAAAQEAlyE26SMFFVY5YJvnL7AF5CRTPtAigSW1U887ASfBt6FDa7Qr1YdO5ochiLoz8aSiMKd5h4
↪SMjq2aycHI+abiVDn3sciQjsLsNW59t48Udivl2RjWG7Eo+LYiB17MKD5M40r5CP2K4B8nuL+r4oAZEHKOJUF3rzA20MZXHRQu
↪I4bz/l0UdGh18SpMB8zVnT3YF5nukQQ/ATspmhpU66s4ntMehULC+ljLvZL40ByNmF0TZc2sdSkA0666==
```

LocalClient Additions

To follow up the recent additions in 0.9.8 of additional kwargs support, 0.9.9 also adds the capability to send kwargs into commands via a dict. This addition to the LocalClient api can be used like so:

```
import salt.client

client = salt.client.LocalClient('/etc/salt/master')
ret = client.cmd('*', 'cmd.run', ['ls -l'], kwarg={'cwd': '/etc'})
```

This update has been added to all cmd methods in the LocalClient class.

Better Self Salting

One problem faced with running Salt states, is that it has been difficult to manage the Salt minion via states, this is due to the fact that if the minion is called to restart while a state run is happening then the state run would be killed. 0.9.9 slightly changes the process scope of the state runs, so now when salt is executing states it can safely restart the salt-minion daemon.

In addition to daemonizing the state run, the apt module also daemonizes. This update makes it possible to cleanly update the salt-minion package on Debian/Ubuntu systems without leaving apt in an inconsistent state or killing the active minion process mid-execution.

Wildcards for SLS Modules

Now, when including sls modules in include statements or in the top file, shell globs can be used. This can greatly simplify listing matched sls modules in the top file and include statements:

```
base:
  '*':
    - files*
    - core*
```

```
include:
  - users.dev.*
  - apache.ser*
```

External Pillar

Since the pillar data is just, data, it does not need to come expressly from the pillar interface. The external pillar system allows for hooks to be added making it possible to extract pillar data from any arbitrary external interface. The external pillar interface is configured via the `ext_pillar` option. Currently interfaces exist to gather external pillar data via hiera or via a shell command that sends yaml data to the terminal:

```
ext_pillar:
  - cmd_yaml: cat /etc/salt/ext.yaml
  - hiera: /etc/hirea.yaml
```

The initial external pillar interfaces and extra interfaces can be added to the file salt/pillar.py, it is planned to add more external pillar interfaces. If the need arises a new module loader interface will be created in the future to manage external pillar interfaces.

Single State Executions

The new state.single function allows for single states to be cleanly executed. This is a great tool for setting up a small group of states on a system or for testing out the behavior of single states:

```
salt '*' state.single user.present name=wade uid=2000
```

The test interface functions here as well, so changes can also be tested against as:

```
salt '*' state.single user.present name=wade uid=2000 test=True
```

New Tests

A few exciting new test interfaces have been added, the minion swarm allows not only testing of larger loads, but also allows users to see how Salt behaves with large groups of minions without having to create a large deployment.

Minion Swarm

The minion swarm test system allows for large groups of minions to be tested against easily without requiring large numbers of servers or virtual machines. The minion swarm creates as many minions as a system can handle and roots them in the /tmp directory and connects them to a master.

The benefit here is that we were able to replicate issues that happen only when there are large numbers of minions. A number of elusive bugs which were causing stability issues in masters and minions have since been hunted down. Bugs that used to take careful watch by users over several days can now be reliably replicated in minutes, and fixed in minutes.

Using the swarm is easy, make sure a master is up for the swarm to connect to, and then use the minionswarm.py script in the tests directory to spin up as many minions as you want. Remember, this is a fork bomb, don't spin up more than your hardware can handle!

```
python minionswarm.py -m 20 --master salt-master
```

Shell Tests

The new Shell testing system allows us to test the behavior of commands executed from a high level. This allows for the high level testing of salt runners and commands like salt-key.

Client Tests

Tests have been added to test the aspects of the client APIs and ensure that the client calls work, and that they manage passed data, in a desirable way.

See also:

Legacy salt-cloud release docs

See also:

Legacy salt-api release docs

CHAPTER

TWENTYNINE

VENAFI TOOLS FOR SALT

29.1 Introduction

First, you need to configure the `master` file. This is because all module functions require either a configured `api_key` (for Cloud) or a `ttp_user` with a `tpp_password` and a `base_url` (for Trust Platform).

For Venafi Cloud:

```
venafi:
  api_key: abcdef01-2345-6789-abcd-ef0123456789
  base_url: "https://cloud.venafi.example.com/"    (optional)
```

If you don't have a Venafi Cloud account, you can sign up for one on the enrollment page.

For Venafi Platform:

```
venafi:
  base_url: "https://tpp.example.com/"
  tpp_user: admin
  tpp_password: "Str0ngPa$$w0rd"
  trust_bundle: "/opt/venafi/bundle.pem"
```

It is not common for the Venafi Platform's REST API (WebSDK) to be secured using a certificate issued by a publicly trusted CA, therefore establishing trust for that server certificate is a critical part of your configuration. Ideally this is done by obtaining the root CA certificate in the issuing chain in PEM format and copying that file to your Salt Master (e.g. /opt/venafi/bundle.pem). You then reference that file using the 'trust_bundle' parameter as shown above.

For the Venafi module to create keys and certificates it is necessary to enable external pillars. This is done by adding the following to the `/etc/salt/master` file:

```
ext_pillar:
  - venafi: True
```

29.2 Runner Functions

29.2.1 request

This command is used to enroll a certificate from Venafi Cloud or Venafi Platform.

`minion_id` ID of the minion for which the certificate is being issued. Required.

`dns_name` DNS subject name for the certificate. Required if `csr_path` is not specified.

csr_path Full path name of certificate signing request file to enroll. Required if `dns_name` is not specified.

zone Venafi Cloud zone ID or Venafi Platform folder that specify key and certificate policy. Defaults to "Default". For Venafi Cloud, the Zone ID can be found in the Zone page for your Venafi Cloud project.

org_unit Business Unit, Department, etc. Do not specify if it does not apply.

org Exact legal name of your organization. Do not abbreviate.

loc City/locality where your organization is legally located.

state State or province where your organization is legally located. Must not be abbreviated.

country Country where your organization is legally located; two-letter ISO code.

key_password Password for encrypting the private key.

The syntax for requesting a new certificate with private key generation looks like this:

```
salt-run venafi.request minion.example.com dns_name=www.example.com \
country=US state=California loc=Sacramento org="Company Name" org_unit=DevOps \
zone=Internet key_password=SecretSauce
```

And the syntax for requesting a new certificate using a previously generated CSR looks like this:

```
salt-run venafi.request minion.example.com csr_path=/tmp/minion.req zone=Internet
```

29.2.2 show_cert

This command is used to show last issued certificate for domain.

dns_name DNS subject name of the certificate to look up.

```
salt-run venafi.show_cert www.example.com
```

29.2.3 list_domain_cache

This command lists domains that have been cached on this Salt Master.

```
salt-run venafi.list_domain_cache
```

29.2.4 del_cached_domain

This command deletes a domain from the Salt Master's cache.

domains A domain name, or a comma-separated list of domain names, to delete from this master's cache.

```
salt-run venafi.del_cached_domain www.example.com
```

29.3 Transfer certificate to a minion

To transfer a cached certificate to a minion, you can use Venafi pillar.

Example state (SLS) file:

```
/etc/ssl/cert/www.example.com.crt:
  file.managed:
      - contents_pillar: venafi:www.example.com:cert
      - replace: True

/etc/ssl/cert/www.example.com.key:
  file.managed:
      - contents_pillar: venafi:www.example.com:pkey
      - replace: True

/etc/ssl/cert/www.example.com-chain.pem:
  file.managed:
      - contents_pillar: venafi:www.example.com:chain
      - replace: True
```

CHAPTER

THIRTY

GLOSSARY

Auto-Order The evaluation of states in the order that they are defined in a SLS file. *See also*: *ordering*.

Bootstrap A stand-alone Salt project which can download and install a Salt master and/or a Salt minion onto a host. *See also*: salt-bootstrap.

Compound Matcher A combination of many target definitions that can be combined with boolean operators. *See also*: *targeting*.

EAuth Shorthand for 'external authentication'. A system for calling to a system outside of Salt in order to authenticate users and determine if they are allowed to issue particular commands to Salt. *See also*: *external auth*.

Environment A directory tree containing state files which can be applied to minions. *See also*: *top file*.

Execution Function A Python function inside an Execution Module that may take arguments and performs specific system-management tasks. *See also*: *the list of execution modules*.

External Job Cache An external data-store that can archive information about jobs that have been run. A default returner. *See also*: `ext_job_cache`, *the list of returners*.

Execution Module A Python module that contains execution functions which directly perform various system-management tasks on a server. Salt ships with a number of execution modules but users can also write their own execution modules to perform specialized tasks. *See also*: *the list of execution modules*.

External Pillar A module that accepts arbitrary arguments and returns a dictionary. The dictionary is automatically added to a pillar for a minion.

Event A notice emitted onto an event bus. Events are often driven by requests for actions to occur on a minion or master and the results of those actions. *See also*: *Salt Reactor*.

File Server A local or remote location for storing both Salt-specific files such as top files or SLS files as well as files that can be distributed to minions, such as system configuration files. *See also*: *Salt's file server*.

Grain A key-value pair which contains a fact about a system, such as its hostname, network addresses. *See also*: *targeting with grains*.

Highdata The data structure in a SLS file the represents a set of state declarations. *See also*: *state layers*.

Highstate The collection of states to be applied to a system. *See also*: *state layers*.

Idempotent An action that ensures the system is in a well-known state regardless of the system's state before the action is applied. A corollary to this is that applying the action multiple times results in no changes to the system. State module functions should be idempotent. Some state module functions, such as `cmd.run` are not idempotent by default but can be made idempotent with the proper use of requisites such as *unless* and *onlyif*. For more information, *see* wikipedia.

Jinja A templating language which allows variables and simple logic to be dynamically inserted into static text files when they are rendered. *See also*: `Salt's Jinja documentation`.

Job The complete set of tasks to be performed by the execution of a Salt command are a single job. *See also*: `jobs runner`.

Job Cache A storage location for job results, which may then be queried by a salt runner or an external system. May be local to a salt master or stored externally.

Job ID A unique identifier to represent a given *job*. This is often shortened to JID.

Low State The collection of processed states after requisites and order are evaluated. *See also*: *state layers*.

Master A central Salt daemon from which commands can be issued to listening minions.

Masterless A minion which does not require a Salt master to operate. All configuration is local. *See also*: `file_client`.

Master Tops A system for the master that allows hooks into external systems to generate top file data.

Mine A facility to collect arbitrary data from minions and store that data on the master. This data is then available to all other minions. (Sometimes referred to as Salt Mine.) *See also*: *Salt Mine*.

Minion A server running a Salt minion daemon which can listen to commands from a master and perform the requested tasks. Generally, minions are servers which are to be controlled using Salt.

Minion ID A globally unique identifier for a minion. *See also*: `id`.

Multi-Master The ability for a minion to be actively connected to multiple Salt masters at the same time in high-availability environments.

Node Group A pre-defined group of minions declared in the master configuration file. *See also*: *targeting*.

Outputter A formatter for defining the characteristics of output data from a Salt command. *See also*: *list of outputters*.

Peer Communication The ability for minions to communicate directly with other minions instead of brokering commands through the Salt master. *See also*: *peer communication*.

Pillar A simple key-value store for user-defined data to be made available to a minion. Often used to store and distribute sensitive data to minions. *See also*: *Pillar*, *list of Pillar modules*.

Proxy Minion A minion which can control devices that are unable to run a Salt minion locally, such as routers and switches.

PyDSL A Pythonic domain-specific-language used as a Salt renderer. PyDSL can be used in cases where adding pure Python into SLS files is beneficial. *See also*: `PyDSL`.

Reactor An interface for listening to events and defining actions that Salt should taken upon receipt of given events. *See also*: *Reactor*.

Render Pipe Allows SLS files to be rendered by multiple renderers, with each renderer receiving the output of the previous. *See also*: *composing renderers*.

Renderer Responsible for translating a given data serialization format such as YAML or JSON into a Python data structure that can be consumed by Salt. *See also*: *list of renderers*.

Returner Allows for the results of a Salt command to be sent to a given data-store such as a database or log file for archival. *See also*: *list of returners*.

Roster A flat-file list of target hosts. (Currently only used by salt-ssh.)

Runner Module A module containing a set of runner functions. *See also*: *list of runner modules*.

Runner Function A function which is called by the **`salt-run`** command and executes on the master instead of on a minion. *See also*: *Runner Module*.

Salt Cloud A suite of tools used to create and deploy systems on many hosted cloud providers. *See also*: *salt-cloud*.

Salt SSH A configuration management and remote orchestration system that does not require that any software besides SSH be installed on systems to be controlled.

Salt Thin A subset of the normal Salt distribution that does not include any transport routines. A Salt Thin bundle can be dropped onto a host and used directly without any requirement that the host be connected to a network. Used by Salt SSH. *See also*: `thin runner`.

Salt Virt Used to manage the creation and deployment of virtual machines onto a set of host machines. Often used to create and deploy private clouds. *See also*: `virt runner`.

SLS Module Contains a set of *state declarations*.

State Compiler Translates *highdata* into lowdata.

State Declaration A data structure which contains a unique ID and describes one or more states of a system such as ensuring that a package is installed or a user is defined. *See also*: *highstate structure*.

State Function A function contained inside a *state module* which can manages the application of a particular state to a system. State functions frequently call out to one or more *execution modules* to perform a given task.

State Module A module which contains a set of state functions. *See also*: *list of state modules*.

State Run The application of a set of states on a set of systems.

Syndic A forwarder which can relay messages between tiered masters. **See also**: *Syndic*.

Target Minion(s) to which a given salt command will apply. *See also*: *targeting*.

Top File Determines which SLS files should be applied to various systems and organizes those groups of systems into environments. *See also*: *top file*, *list of master top modules*.

__virtual__ A function in a module that is called on module load to determine whether or not the module should be available to a minion. This function commonly contains logic to determine if all requirements for a module are available, such as external libraries.

Worker A master process which can send notices and receive replies from minions. *See also*: `worker_threads`.

SALT MODULE INDEX

a

salt.auth.auto, 997
salt.auth.django, 997
salt.auth.file, 999
salt.auth.keystone, 1000
salt.auth.ldap, 1001
salt.auth.mysql, 1001
salt.auth.pam, 1002
salt.auth.pki, 1003
salt.auth.rest, 1004
salt.auth.sharedsecret, 1004
salt.auth.yubico, 1005

b

salt.beacons.adb, 1006
salt.beacons.avahi_announce, 1007
salt.beacons.bonjour_announce, 1008
salt.beacons.btmp, 1009
salt.beacons.diskusage, 1010
salt.beacons.glxinfo, 1011
salt.beacons.haproxy, 1011
salt.beacons.inotify, 1012
salt.beacons.journald, 1013
salt.beacons.load, 1013
salt.beacons.log, 1014
salt.beacons.memusage, 1014
salt.beacons.napalm_beacon, 1014
salt.beacons.network_info, 1017
salt.beacons.network_settings, 1018
salt.beacons.pkg, 1018
salt.beacons.proxy_example, 1019
salt.beacons.ps, 1019
salt.beacons.salt_proxy, 1019
salt.beacons.sensehat, 1020
salt.beacons.service, 1020
salt.beacons.sh, 1021
salt.beacons.status, 1021
salt.beacons.telegram_bot_msg, 1023
salt.beacons.twilio_txt_msg, 1023
salt.beacons.wtmp, 1024

c

salt.cache.consul, 1026
salt.cache.etcd_cache, 1026
salt.cache.localfs, 1027
salt.cache.mysql_cache, 1028
salt.cache.redis_cache, 1029
salt.cloud.clouds.aliyun, 1032
salt.cloud.clouds.azurearm, 1034
salt.cloud.clouds.cloudstack, 1038
salt.cloud.clouds.digitalocean, 1039
salt.cloud.clouds.dimensiondata, 1043
salt.cloud.clouds.ec2, 1044
salt.cloud.clouds.gce, 1053
salt.cloud.clouds.gogrid, 1059
salt.cloud.clouds.joyent, 1061
salt.cloud.clouds.linode, 1065
salt.cloud.clouds.lxc, 1070
salt.cloud.clouds.msazure, 1071
salt.cloud.clouds.oneandone, 1087
salt.cloud.clouds.opennebula, 1090
salt.cloud.clouds.openstack, 843
salt.cloud.clouds.parallels, 1114
salt.cloud.clouds.profitbricks, 1115
salt.cloud.clouds.proxmox, 1119
salt.cloud.clouds.pyrax, 1121
salt.cloud.clouds.qingcloud, 1122
salt.cloud.clouds.saltify, 1124
salt.cloud.clouds.scaleway, 1126
salt.cloud.clouds.softlayer, 1127
salt.cloud.clouds.softlayer_hw, 1128
salt.cloud.clouds.tencentcloud, 1130
salt.cloud.clouds.vagrant, 1133
salt.cloud.clouds.virtualbox, 1134
salt.cloud.clouds.vmware, 1136
salt.cloud.clouds.vultrpy, 1147
salt.cloud.clouds.xen, 1149

e

salt.engines.docker_events, 1153
salt.engines.fluent, 1154
salt.engines.http_logstash, 1154
salt.engines.ircbot, 1155

salt.engines.junos_syslog, 1157
salt.engines.logentries, 1159
salt.engines.logstash_engine, 1160
salt.engines.napalm_syslog, 1160
salt.engines.reactor, 1163
salt.engines.redis_sentinel, 1163
salt.engines.slack, 1164
salt.engines.sqs_events, 1168
salt.engines.stalekey, 1169
salt.engines.test, 1170
salt.engines.thorium, 1170
salt.engines.webhook, 1170
salt.exceptions, 3998
salt.executors.direct_call, 1171
salt.executors.docker, 1171
salt.executors.splay, 1171
salt.executors.sudo, 1172

f

salt.fileserver.azurefs, 1172
salt.fileserver.gitfs, 1173
salt.fileserver.hgfs, 1174
salt.fileserver.minionfs, 1174
salt.fileserver.roots, 1174
salt.fileserver.s3fs, 1175
salt.fileserver.svnfs, 1176

g

salt.grains.chronos, 1177
salt.grains.cimc, 1177
salt.grains.core, 1177
salt.grains.disks, 1179
salt.grains.esxi, 1179
salt.grains.extra, 1179
salt.grains.fx2, 1180
salt.grains.junos, 1180
salt.grains.marathon, 1180
salt.grains.mdadm, 1180
salt.grains.metadata, 1180
salt.grains.napalm, 1181
salt.grains.opts, 1185
salt.grains.panos, 1185
salt.grains.philips_hue, 1185
salt.grains.rest_sample, 1185

l

salt.log.handlers.fluent_mod, 248
salt.log.handlers.log4mongo_mod, 249
salt.log.handlers.logstash_mod, 250
salt.log.handlers.sentry_mod, 252

m

salt.modules.acme, 1200
salt.modules.aix_group, 1203
salt.modules.aixpkg, 1204
salt.modules.aliases, 1206
salt.modules.alternatives, 1206
salt.modules.ansiblegate, 1208
salt.modules.apache, 1209
salt.modules.apcups, 1211
salt.modules.apf, 1212
salt.modules.apkpkg, 1213
salt.modules.aptpkg, 1216
salt.modules.archive, 1226
salt.modules.arista_pyeapi, 1234
salt.modules.artifactory, 1240
salt.modules.at, 1242
salt.modules.at_solaris, 1243
salt.modules.augeas_cfg, 1244
salt.modules.aws_sqs, 1246
salt.modules.azurearm_compute, 1248
salt.modules.azurearm_network, 1253
salt.modules.azurearm_resource, 1271
salt.modules.bamboohr, 1279
salt.modules.bcache, 1280
salt.modules.beacons, 1283
salt.modules.bigip, 1285
salt.modules.bluez_bluetooth, 1299
salt.modules.boto3_elasticache, 1301
salt.modules.boto3_elasticsearch, 1308
salt.modules.boto3_route53, 1322
salt.modules.boto_apigateway, 1329
salt.modules.boto_asg, 1341
salt.modules.boto_cfn, 1344
salt.modules.boto_cloudfront, 1346
salt.modules.boto_cloudtrail, 1348
salt.modules.boto_cloudwatch, 1351
salt.modules.boto_cloudwatch_event, 1353
salt.modules.boto_cognitoidentity, 1355
salt.modules.boto_datapipeline, 1358
salt.modules.boto_dynamodb, 1359
salt.modules.boto_ec2, 1361
salt.modules.boto_efs, 1375
salt.modules.boto_elasticache, 1378
salt.modules.boto_elasticsearch_domain, 1382
salt.modules.boto_elb, 1386
salt.modules.boto_elbv2, 1391
salt.modules.boto_iam, 1393
salt.modules.boto_iot, 1406
salt.modules.boto_kinesis, 1412
salt.modules.boto_kms, 1415
salt.modules.boto_lambda, 1418
salt.modules.boto_rds, 1424
salt.modules.boto_route53, 1429
salt.modules.boto_s3_bucket, 1435
salt.modules.boto_secgroup, 1441
salt.modules.boto_sns, 1444

`salt.modules.boto_sqs`, 1445
`salt.modules.boto_vpc`, 1447
`salt.modules.bower`, 1462
`salt.modules.bridge`, 1464
`salt.modules.bsd_shadow`, 1465
`salt.modules.btrfs`, 1466
`salt.modules.cabal`, 1471
`salt.modules.capirca_acl`, 1472
`salt.modules.cassandra`, 1481
`salt.modules.cassandra_cql`, 1482
`salt.modules.celery`, 1490
`salt.modules.ceph`, 1491
`salt.modules.chassis`, 1498
`salt.modules.chef`, 1499
`salt.modules.chocolatey`, 1500
`salt.modules.chronos`, 1506
`salt.modules.cimc`, 1507
`salt.modules.ciscoconfparse_mod`, 1514
`salt.modules.cisconso`, 1518
`salt.modules.cloud`, 1519
`salt.modules.cmdmod`, 1523
`salt.modules.composer`, 1554
`salt.modules.config`, 1556
`salt.modules.consul`, 1560
`salt.modules.container_resource`, 1573
`salt.modules.cp`, 1574
`salt.modules.cpan`, 1579
`salt.modules.cron`, 1579
`salt.modules.csf`, 1581
`salt.modules.cyg`, 1583
`salt.modules.daemontools`, 1585
`salt.modules.data`, 1586
`salt.modules.datadog_api`, 1588
`salt.modules.ddns`, 1589
`salt.modules.deb_apache`, 1590
`salt.modules.deb_postgres`, 1592
`salt.modules.debconfmod`, 1595
`salt.modules.debian_ip`, 1596
`salt.modules.debian_service`, 1597
`salt.modules.debuild_pkgbuild`, 1593
`salt.modules.defaults`, 1599
`salt.modules.devmap`, 1601
`salt.modules.dig`, 1601
`salt.modules.disk`, 1602
`salt.modules.djangomod`, 1605
`salt.modules.dnsmasq`, 1607
`salt.modules.dnsutil`, 1608
`salt.modules.dockercompose`, 1609
`salt.modules.dockermod`, 1615
`salt.modules.dpkg_lowpkg`, 1661
`salt.modules.drac`, 1662
`salt.modules.dracr`, 1665
`salt.modules.drbd`, 1673
`salt.modules.dummyproxy_pkg`, 1673
`salt.modules.dummyproxy_service`, 1674
`salt.modules.ebuildpkg`, 1675
`salt.modules.eix`, 1680
`salt.modules.elasticsearch`, 1681
`salt.modules.environ`, 1693
`salt.modules.eselect`, 1694
`salt.modules.esxi`, 1696
`salt.modules.etcd_mod`, 1696
`salt.modules.ethtool`, 1698
`salt.modules.event`, 1700
`salt.modules.extfs`, 1701
`salt.modules.file`, 1703
`salt.modules.firewalld`, 1731
`salt.modules.freebsd_sysctl`, 1738
`salt.modules.freebsd_update`, 1738
`salt.modules.freebsdjail`, 1739
`salt.modules.freebsdkmod`, 1740
`salt.modules.freebsdpkg`, 1741
`salt.modules.freebsdports`, 1744
`salt.modules.freebsdservice`, 1746
`salt.modules.gem`, 1749
`salt.modules.genesis`, 1752
`salt.modules.gentoo_service`, 1754
`salt.modules.gentoolkitmod`, 1756
`salt.modules.git`, 1757
`salt.modules.github`, 1799
`salt.modules.glance`, 1808
`salt.modules.glanceng`, 1810
`salt.modules.glusterfs`, 1811
`salt.modules.gnomedesktop`, 1815
`salt.modules.google_chat`, 1817
`salt.modules.gpg`, 1817
`salt.modules.grafana4`, 1823
`salt.modules.grains`, 1831
`salt.modules.group`, 1186
`salt.modules.groupadd`, 1836
`salt.modules.grub_legacy`, 1838
`salt.modules.guestfs`, 1838
`salt.modules.hadoop`, 1839
`salt.modules.haproxyconn`, 1840
`salt.modules.hashutil`, 1842
`salt.modules.heat`, 1845
`salt.modules.hg`, 1847
`salt.modules.hosts`, 1849
`salt.modules.http`, 1850
`salt.modules.icinga2`, 1855
`salt.modules.ifttt`, 1852
`salt.modules.ilo`, 1852
`salt.modules.incron`, 1856
`salt.modules.influxdb08mod`, 1862
`salt.modules.influxdbmod`, 1857
`salt.modules.infoblox`, 1867
`salt.modules.ini_manage`, 1872
`salt.modules.inspectlib`, 1876

salt.modules.inspectlib.collector, 1874
salt.modules.inspectlib.dbhandle, 1875
salt.modules.inspectlib.entities, 1876
salt.modules.inspectlib.exceptions, 1875
salt.modules.inspectlib.fsdb, 1876
salt.modules.inspectlib.kiwiproc, 1878
salt.modules.inspectlib.query, 1876
salt.modules.inspector, 1878
salt.modules.introspect, 1880
salt.modules.iosconfig, 1881
salt.modules.ipmi, 1883
salt.modules.ipset, 1896
salt.modules.iptables, 1898
salt.modules.iwtools, 1902
salt.modules.jboss7, 1903
salt.modules.jboss7_cli, 1907
salt.modules.jenkinsmod, 1908
salt.modules.jinja, 1911
salt.modules.jira_mod, 1911
salt.modules.junos, 1913
salt.modules.k8s, 1919
salt.modules.kapacitor, 1922
salt.modules.kerberos, 1923
salt.modules.kernelpkg, 1186
salt.modules.kernelpkg_linux_apt, 1925
salt.modules.kernelpkg_linux_yum, 1926
salt.modules.key, 1928
salt.modules.keyboard, 1928
salt.modules.keystone, 1929
salt.modules.keystoneng, 1936
salt.modules.keystore, 1944
salt.modules.kmod, 1945
salt.modules.kubernetesmod, 1946
salt.modules.launchctl_service, 1951
salt.modules.layman, 1952
salt.modules.ldap3, 1953
salt.modules.ldapmod, 1958
salt.modules.libcloud_compute, 1959
salt.modules.libcloud_dns, 1965
salt.modules.libcloud_loadbalancer, 1968
salt.modules.libcloud_storage, 1971
salt.modules.linux_acl, 1975
salt.modules.linux_ip, 1976
salt.modules.linux_lvm, 1976
salt.modules.linux_sysctl, 1979
salt.modules.localemod, 1979
salt.modules.locate, 1980
salt.modules.logadm, 1981
salt.modules.logmod, 1982
salt.modules.logrotate, 1983
salt.modules.lvs, 1984
salt.modules.lxc, 1986
salt.modules.lxd, 2003
salt.modules.mac_assistive, 2022
salt.modules.mac_brew_pkg, 2023
salt.modules.mac_desktop, 2026
salt.modules.mac_group, 2027
salt.modules.mac_keychain, 2028
salt.modules.mac_pkgutil, 2031
salt.modules.mac_portspkg, 2032
salt.modules.mac_power, 2034
salt.modules.mac_service, 2038
salt.modules.mac_shadow, 2041
salt.modules.mac_softwareupdate, 2045
salt.modules.mac_sysctl, 2048
salt.modules.mac_system, 2049
salt.modules.mac_timezone, 2054
salt.modules.mac_user, 2056
salt.modules.mac_xattr, 2059
salt.modules.macdefaults, 2025
salt.modules.macpackage, 2029
salt.modules.makeconf, 2060
salt.modules.mandrill, 2068
salt.modules.marathon, 2069
salt.modules.match, 2070
salt.modules.mattermost, 2073
salt.modules.mdadm_raid, 2074
salt.modules.mdata, 2076
salt.modules.memcached, 2077
salt.modules.mine, 2078
salt.modules.minion, 2080
salt.modules.mod_random, 2082
salt.modules.modjk, 2083
salt.modules.mongodb, 2086
salt.modules.monit, 2089
salt.modules.moosefs, 2091
salt.modules.mount, 2091
salt.modules.mssql, 2095
salt.modules.msteams, 2098
salt.modules.munin, 2098
salt.modules.mysql, 2099
salt.modules.nacl, 2107
salt.modules.nagios, 2111
salt.modules.nagios_rpc, 2112
salt.modules.namecheap_domains, 2115
salt.modules.namecheap_domains_dns, 2113
salt.modules.namecheap_domains_ns, 2116
salt.modules.namecheap_ssl, 2118
salt.modules.namecheap_users, 2124
salt.modules.napalm_bgp, 2148
salt.modules.napalm_formula, 2152
salt.modules.napalm_mod, 2125
salt.modules.napalm_netacl, 2139
salt.modules.napalm_network, 2155
salt.modules.napalm_ntp, 2175
salt.modules.napalm_probes, 2178
salt.modules.napalm_route, 2183
salt.modules.napalm_snmp, 2184

`salt.modules.napalm_users`, 2186
`salt.modules.napalm_yang_mod`, 2188
`salt.modules.netaddress`, 2195
`salt.modules.netbox`, 2196
`salt.modules.netbsd_sysctl`, 2202
`salt.modules.netbsdservice`, 2202
`salt.modules.netmiko_mod`, 2205
`salt.modules.netscaler`, 2211
`salt.modules.network`, 2215
`salt.modules.neutron`, 2221
`salt.modules.neutronng`, 2238
`salt.modules.nfs3`, 2243
`salt.modules.nftables`, 2243
`salt.modules.nginx`, 2248
`salt.modules.nilrt_ip`, 2248
`salt.modules.nix`, 2251
`salt.modules.nova`, 2252
`salt.modules.npm`, 2257
`salt.modules.nspawn`, 2259
`salt.modules.nxos`, 2265
`salt.modules.nxos_api`, 2266
`salt.modules.omapi`, 2269
`salt.modules.openbsd_sysctl`, 2270
`salt.modules.openbsdpkg`, 2270
`salt.modules.openbsdrcctl_service`, 2272
`salt.modules.openbsdservice`, 2274
`salt.modules.openscap`, 2276
`salt.modules.openstack_config`, 2276
`salt.modules.openstack_mng`, 2277
`salt.modules.openvswitch`, 2278
`salt.modules.opkg`, 2280
`salt.modules.opsgenie`, 2286
`salt.modules.oracle`, 2286
`salt.modules.osquery`, 2288
`salt.modules.out`, 2296
`salt.modules.pacmanpkg`, 2297
`salt.modules.pagerduty`, 2302
`salt.modules.pagerduty_util`, 2303
`salt.modules.pam`, 2305
`salt.modules.panos`, 2305
`salt.modules.parallels`, 2320
`salt.modules.parted_partition`, 2324
`salt.modules.pcs`, 2328
`salt.modules.pdbedit`, 2331
`salt.modules.pecl`, 2333
`salt.modules.peeringdb`, 2333
`salt.modules.pf`, 2336
`salt.modules.philips_hue`, 2338
`salt.modules.pillar`, 2338
`salt.modules.pip`, 2344
`salt.modules.pkg`, 1186
`salt.modules.pkg_resource`, 2349
`salt.modules.pkgin`, 2351
`salt.modules.pkgng`, 2354
`salt.modules.pkgutil`, 2368
`salt.modules.portage_config`, 2370
`salt.modules.postfix`, 2372
`salt.modules.postgres`, 2374
`salt.modules.poudriere`, 2385
`salt.modules.powerpath`, 2387
`salt.modules.proxy`, 2387
`salt.modules.ps`, 2389
`salt.modules.publish`, 2393
`salt.modules.puppet`, 2395
`salt.modules.purefa`, 2396
`salt.modules.purefb`, 2402
`salt.modules.pushbullet`, 2404
`salt.modules.pushover_notify`, 2405
`salt.modules.pw_group`, 2405
`salt.modules.pw_user`, 2406
`salt.modules.pyenv`, 2409
`salt.modules.qemu_img`, 2410
`salt.modules.qemu_nbd`, 2411
`salt.modules.quota`, 2411
`salt.modules.rabbitmq`, 2412
`salt.modules.rallydev`, 2418
`salt.modules.random_org`, 2420
`salt.modules.rbac_solaris`, 2422
`salt.modules.rbenv`, 2424
`salt.modules.rdp`, 2426
`salt.modules.redismod`, 2427
`salt.modules.reg`, 2433
`salt.modules.rest_pkg`, 2438
`salt.modules.rest_sample_utils`, 2439
`salt.modules.rest_service`, 2439
`salt.modules.restartcheck`, 2440
`salt.modules.ret`, 2441
`salt.modules.rh_ip`, 2441
`salt.modules.rh_service`, 2443
`salt.modules.riak`, 2445
`salt.modules.rpm_lowpkg`, 2447
`salt.modules.rpmbuild_pkgbuild`, 2449
`salt.modules.rsync`, 2451
`salt.modules.runit`, 2452
`salt.modules.rvm`, 2456
`salt.modules.s3`, 2458
`salt.modules.s6`, 2461
`salt.modules.salt_proxy`, 2462
`salt.modules.salt_version`, 2462
`salt.modules.saltcheck`, 2463
`salt.modules.saltcloudmod`, 2469
`salt.modules.saltutil`, 2469
`salt.modules.schedule`, 2481
`salt.modules.scp_mod`, 2484
`salt.modules.scsi`, 2485
`salt.modules.sdb`, 2485
`salt.modules.seed`, 2486
`salt.modules.selinux`, 2487

`salt.modules.sensehat`, 2492
`salt.modules.sensors`, 2494
`salt.modules.serverdensity_device`, 2494
`salt.modules.service`, 1186
`salt.modules.servicenow`, 2495
`salt.modules.shadow`, 1187
`salt.modules.slack_notify`, 2496
`salt.modules.slsutil`, 2498
`salt.modules.smartos_imgadm`, 2500
`salt.modules.smartos_nictagadm`, 2502
`salt.modules.smartos_virt`, 2503
`salt.modules.smartos_vmadm`, 2504
`salt.modules.smbios`, 2508
`salt.modules.smf_service`, 2510
`salt.modules.smtp`, 2512
`salt.modules.snapper`, 2543
`salt.modules.solaris_fmadm`, 2513
`salt.modules.solaris_group`, 2515
`salt.modules.solaris_shadow`, 2516
`salt.modules.solaris_system`, 2517
`salt.modules.solaris_user`, 2518
`salt.modules.solarisipspkg`, 2520
`salt.modules.solarispkg`, 2524
`salt.modules.solr`, 2527
`salt.modules.solrcloud`, 2533
`salt.modules.splunk`, 2536
`salt.modules.splunk_search`, 2536
`salt.modules.sqlite3`, 2538
`salt.modules.ssh`, 2539
`salt.modules.ssh_pkg`, 2542
`salt.modules.ssh_service`, 2542
`salt.modules.state`, 2547
`salt.modules.status`, 2558
`salt.modules.statuspage`, 2562
`salt.modules.supervisord`, 2566
`salt.modules.suse_apache`, 2568
`salt.modules.svn`, 2568
`salt.modules.swarm`, 2571
`salt.modules.swift`, 2573
`salt.modules.sysbench`, 2575
`salt.modules.sysfs`, 2575
`salt.modules.syslog_ng`, 2577
`salt.modules.sysmod`, 2582
`salt.modules.sysrc`, 2588
`salt.modules.system`, 2588
`salt.modules.system_profiler`, 2591
`salt.modules.systemd_service`, 2592
`salt.modules.telegram`, 2598
`salt.modules.telemetry`, 2598
`salt.modules.temp`, 2599
`salt.modules.test`, 2600
`salt.modules.test_virtual`, 2605
`salt.modules.testinframod`, 2604
`salt.modules.textfsm_mod`, 2605
`salt.modules.timezone`, 2608
`salt.modules.tls`, 2610
`salt.modules.tomcat`, 2618
`salt.modules.trafficserver`, 2622
`salt.modules.travisci`, 2624
`salt.modules.tuned`, 2625
`salt.modules.twilio_notify`, 2625
`salt.modules.udev`, 2626
`salt.modules.upstart_service`, 2627
`salt.modules.uptime`, 2629
`salt.modules.user`, 1187
`salt.modules.useradd`, 2630
`salt.modules.uwsgi`, 2634
`salt.modules.vagrant`, 2634
`salt.modules.varnish`, 2637
`salt.modules.vault`, 2638
`salt.modules.vbox_guest`, 2641
`salt.modules.vboxmanage`, 2642
`salt.modules.victorops`, 2644
`salt.modules.virt`, 2645
`salt.modules.virtualenv_mod`, 2676
`salt.modules.vmctl`, 2678
`salt.modules.vsphere`, 2679
`salt.modules.webutil`, 2719
`salt.modules.win_autoruns`, 2720
`salt.modules.win_certutil`, 2720
`salt.modules.win_dacl`, 2721
`salt.modules.win_disk`, 2723
`salt.modules.win_dism`, 2724
`salt.modules.win_dns_client`, 2728
`salt.modules.win_dsc`, 2729
`salt.modules.win_file`, 2733
`salt.modules.win_firewall`, 2745
`salt.modules.win_groupadd`, 2752
`salt.modules.win_iis`, 2754
`salt.modules.win_ip`, 2764
`salt.modules.win_lgpo`, 2766
`salt.modules.win_license`, 2772
`salt.modules.win_network`, 2772
`salt.modules.win_ntp`, 2776
`salt.modules.win_path`, 2776
`salt.modules.win_pkg`, 2777
`salt.modules.win_pki`, 2785
`salt.modules.win_powercfg`, 2787
`salt.modules.win_psget`, 2789
`salt.modules.win_servermanager`, 2792
`salt.modules.win_service`, 2794
`salt.modules.win_shadow`, 2801
`salt.modules.win_smtp_server`, 2802
`salt.modules.win_snmp`, 2804
`salt.modules.win_status`, 2806
`salt.modules.win_system`, 2812
`salt.modules.win_task`, 2820
`salt.modules.win_timezone`, 2831

salt.modules.win_useradd, 2833
salt.modules.win_wua, 2838
salt.modules.win_wusa, 2846
salt.modules.winrepo, 2791
salt.modules.x509, 2847
salt.modules.xapi_virt, 2853
salt.modules.xbpspkg, 2857
salt.modules.xfs, 2860
salt.modules.xml, 2861
salt.modules.xmpp, 2862
salt.modules.yumpkg, 2863
salt.modules.zabbix, 2878
salt.modules.zcbuildout, 2896
salt.modules.zenoss, 2897
salt.modules.zfs, 2898
salt.modules.zk_concurrency, 2905
salt.modules.znc, 2907
salt.modules.zoneadm, 2907
salt.modules.zonecfg, 2910
salt.modules.zookeeper, 2912
salt.modules.zpool, 2916
salt.modules.zypperpkg, 2924

n

salt.netapi.rest_cherrypy.app, 2935
salt.netapi.rest_cherrypy.wsgi, 2943
salt.netapi.rest_tornado.saltnado, 2960
salt.netapi.rest_tornado.saltnado_websockets, 2962
salt.netapi.rest_wsgi, 2977

o

salt.output.highstate, 2980
salt.output.json_out, 2981
salt.output.key, 2982
salt.output.nested, 2982
salt.output.newline_values_only, 2983
salt.output.no_out_quiet, 2984
salt.output.no_return, 2984
salt.output.overstatestage, 2985
salt.output.pony, 2985
salt.output.pprint_out, 2986
salt.output.profile, 2986
salt.output.progress, 2986
salt.output.raw, 2987
salt.output.table_out, 2987
salt.output.txt, 2988
salt.output.virt_query, 2989
salt.output.yaml_out, 2989

p

salt.pillar.cmd_json, 2991
salt.pillar.cmd_yaml, 2991
salt.pillar.cmd_yamlex, 2991
salt.pillar.cobbler, 2991
salt.pillar.confidant, 2992
salt.pillar.consul_pillar, 2992
salt.pillar.csvpillar, 2994
salt.pillar.digicert, 2995
salt.pillar.django_orm, 2995
salt.pillar.ec2_pillar, 2997
salt.pillar.etcd_pillar, 2998
salt.pillar.file_tree, 2999
salt.pillar.foreman, 3003
salt.pillar.git_pillar, 3003
salt.pillar.gpg, 3008
salt.pillar.hg_pillar, 3009
salt.pillar.hiera, 3009
salt.pillar.http_json, 3009
salt.pillar.http_yaml, 3010
salt.pillar.libvirt, 3011
salt.pillar.makostack, 3011
salt.pillar.mongo, 3017
salt.pillar.mysql, 3018
salt.pillar.netbox, 3019
salt.pillar.neutron, 3020
salt.pillar.nodegroups, 3021
salt.pillar.pepa, 3021
salt.pillar.pillar_ldap, 3026
salt.pillar.postgres, 3028
salt.pillar.puppet, 3029
salt.pillar.reclass_adapter, 3029
salt.pillar.redismod, 3030
salt.pillar.s3, 3030
salt.pillar.saltclass, 3032
salt.pillar.sql_base, 3032
salt.pillar.sqlcipher, 3035
salt.pillar.sqlite3, 3036
salt.pillar.stack, 3037
salt.pillar.svn_pillar, 3043
salt.pillar.varstack_pillar, 3044
salt.pillar.vault, 3044
salt.pillar.venafi, 3046
salt.pillar.virtkey, 3046
salt.pillar.vmware_pillar, 3046
salt.proxy.arista_pyeapi, 3049
salt.proxy.chronos, 3052
salt.proxy.cimc, 3050
salt.proxy.cisconso, 3053
salt.proxy.dummy, 3056
salt.proxy.esxi, 3057
salt.proxy.fx2, 3062
salt.proxy.junos, 3065
salt.proxy.marathon, 3065
salt.proxy.napalm, 3066
salt.proxy.netmiko_px, 3069
salt.proxy.nxos, 3073
salt.proxy.nxos_api, 3076

salt.proxy.panos, 3078
salt.proxy.philips_hue, 3082
salt.proxy.rest_sample, 3085
salt.proxy.ssh_sample, 3086

q

salt.queues.pgjsonb_queue, 3087
salt.queues.sqlite_queue, 3088

r

salt.renderers.cheetah, 361
salt.renderers.dson, 361
salt.renderers.genshi, 362
salt.renderers.gpg, 362
salt.renderers.hjson, 365
salt.renderers.jinja, 365
salt.renderers.json, 368
salt.renderers.json5, 368
salt.renderers.mako, 368
salt.renderers.msgpack, 368
salt.renderers.pass, 368
salt.renderers.py, 369
salt.renderers.pydsl, 371
salt.renderers.pyobjects, 375
salt.renderers.stateconf, 380
salt.renderers.wempy, 384
salt.renderers.yaml, 385
salt.renderers.yamlex, 386
salt.returners.carbon_return, 312
salt.returners.cassandra_cql_return, 313
salt.returners.cassandra_return, 316
salt.returners.couchbase_return, 316
salt.returners.couchdb_return, 317
salt.returners.django_return, 318
salt.returners.elasticsearch_return, 319
salt.returners.etcd_return, 320
salt.returners.highstate_return, 322
salt.returners.influxdb_return, 323
salt.returners.kafka_return, 324
salt.returners.librato_return, 324
salt.returners.local, 325
salt.returners.local_cache, 325
salt.returners.mattermost_returner, 326
salt.returners.memcache_return, 327
salt.returners.mongo_future_return, 328
salt.returners.mongo_return, 329
salt.returners.multi_returner, 330
salt.returners.mysql, 331
salt.returners.nagios_nrdp_return, 333
salt.returners.odbc, 334
salt.returners.pgjsonb, 337
salt.returners.postgres, 340
salt.returners.postgres_local_cache, 342
salt.returners.pushover_returner, 344
salt.returners.rawfile_json, 346
salt.returners.redis_return, 346
salt.returners.sentry_return, 348
salt.returners.slack_returner, 348
salt.returners.sms_return, 350
salt.returners.smtp_return, 350
salt.returners.splunk, 352
salt.returners.sqlite3_return, 352
salt.returners.syslog_return, 354
salt.returners.telegram_return, 355
salt.returners.xmpp_return, 356
salt.returners.zabbix_return, 357
salt.roster.ansible, 3089
salt.roster.cache, 3090
salt.roster.cloud, 3092
salt.roster.clustershell, 3092
salt.roster.flat, 3092
salt.roster.range, 3092
salt.roster.scan, 3093
salt.roster.sshconfig, 3093
salt.roster.terraform, 3093
salt.runners.asam, 3095
salt.runners.auth, 3096
salt.runners.bgp, 3097
salt.runners.cache, 3099
salt.runners.cloud, 3102
salt.runners.ddns, 3103
salt.runners.digicertapi, 3104
salt.runners.doc, 3107
salt.runners.drac, 3107
salt.runners.error, 3108
salt.runners.event, 3108
salt.runners.f5, 3109
salt.runners.fileserver, 3111
salt.runners.git_pillar, 3115
salt.runners.http, 3116
salt.runners.jobs, 3116
salt.runners.launchd, 3118
salt.runners.lxc, 3119
salt.runners.manage, 3121
salt.runners.mattermost, 3126
salt.runners.mine, 3127
salt.runners.nacl, 3127
salt.runners.net, 3130
salt.runners.network, 3136
salt.runners.pagerduty, 3137
salt.runners.pillar, 3138
salt.runners.pkg, 3139
salt.runners.queue, 3139
salt.runners.reactor, 3141
salt.runners.salt, 3142
salt.runners.saltutil, 3143
salt.runners.sdb, 3148
salt.runners.smartos_vmadm, 3149

salt.runners.spacewalk, 3151
salt.runners.ssh, 3152
salt.runners.state, 3152
salt.runners.survey, 3155
salt.runners.test, 3156
salt.runners.thin, 3156
salt.runners.vault, 3157
salt.runners.venafiapi, 3157
salt.runners.virt, 3158
salt.runners.vistara, 3160
salt.runners.winrepo, 3160

s

salt.sdb.cache, 3161
salt.sdb.confidant, 3162
salt.sdb.consul, 3163
salt.sdb.couchdb, 3163
salt.sdb.env, 3164
salt.sdb.etcd_db, 3165
salt.sdb.keyring_db, 3165
salt.sdb.memcached, 3166
salt.sdb.redis_sdb, 3167
salt.sdb.rest, 3167
salt.sdb.sqlite3, 3168
salt.sdb.tism, 3169
salt.sdb.vault, 3170
salt.sdb.yaml, 3170
salt.serializers.configparser, 3171
salt.serializers.json, 3172
salt.serializers.msgpack, 3172
salt.serializers.python, 3172
salt.serializers.toml, 3173
salt.serializers.yaml, 3173
salt.serializers.yamlex, 3173
salt.states.acme, 3182
salt.states.alias, 3183
salt.states.alternatives, 3184
salt.states.ansiblegate, 3184
salt.states.apache, 3185
salt.states.apache_conf, 3187
salt.states.apache_module, 3187
salt.states.apache_site, 3188
salt.states.aptpkg, 3188
salt.states.archive, 3188
salt.states.artifactory, 3195
salt.states.at, 3195
salt.states.augeas, 3197
salt.states.aws_sqs, 3199
salt.states.azurearm_compute, 3200
salt.states.azurearm_network, 3202
salt.states.azurearm_resource, 3216
salt.states.beacon, 3220
salt.states.bigip, 3222
salt.states.blockdev, 3233
salt.states.boto3_elasticache, 3233
salt.states.boto3_elasticsearch, 3242
salt.states.boto3_route53, 3246
salt.states.boto_apigateway, 3250
salt.states.boto_asg, 3255
salt.states.boto_cfn, 3260
salt.states.boto_cloudfront, 3262
salt.states.boto_cloudtrail, 3263
salt.states.boto_cloudwatch_alarm, 3265
salt.states.boto_cloudwatch_event, 3266
salt.states.boto_cognitoidentity, 3267
salt.states.boto_datapipeline, 3269
salt.states.boto_dynamodb, 3270
salt.states.boto_ec2, 3273
salt.states.boto_elasticache, 3280
salt.states.boto_elasticsearch_domain, 3283
salt.states.boto_elb, 3285
salt.states.boto_elbv2, 3290
salt.states.boto_iam, 3292
salt.states.boto_iam_role, 3298
salt.states.boto_iot, 3300
salt.states.boto_kinesis, 3303
salt.states.boto_kms, 3304
salt.states.boto_lambda, 3306
salt.states.boto_lc, 3309
salt.states.boto_rds, 3312
salt.states.boto_route53, 3316
salt.states.boto_s3_bucket, 3318
salt.states.boto_secgroup, 3321
salt.states.boto_sns, 3324
salt.states.boto_sqs, 3325
salt.states.boto_vpc, 3326
salt.states.bower, 3335
salt.states.cabal, 3336
salt.states.ceph, 3336
salt.states.chef, 3337
salt.states.chocolatey, 3338
salt.states.chronos_job, 3339
salt.states.cimc, 3340
salt.states.cisconso, 3342
salt.states.cloud, 3343
salt.states.cmd, 3344
salt.states.composer, 3354
salt.states.cron, 3355
salt.states.csf, 3359
salt.states.cyg, 3360
salt.states.ddns, 3361
salt.states.debconfmod, 3362
salt.states.dellchassis, 3364
salt.states.disk, 3368
salt.states.docker_container, 3369
salt.states.docker_image, 3390
salt.states.docker_network, 3393

salt.states.docker_volume, 3399
salt.states.drac, 3400
salt.states.elasticsearch, 3401
salt.states.elasticsearch_index, 3402
salt.states.elasticsearch_index_template, 3403
salt.states.environ, 3404
salt.states.eselect, 3404
salt.states.esxi, 3409
salt.states.etcd_mod, 3405
salt.states.ethtool, 3407
salt.states.event, 3415
salt.states.file, 3415
salt.states.firewall, 3451
salt.states.firewalld, 3451
salt.states.gem, 3453
salt.states.git, 3454
salt.states.github, 3463
salt.states.glance, 3465
salt.states.glance_image, 3465
salt.states.glusterfs, 3466
salt.states.gnomedesktop, 3468
salt.states.gpg, 3470
salt.states.grafana, 3470
salt.states.grafana4_dashboard, 3474
salt.states.grafana4_datasource, 3475
salt.states.grafana4_org, 3476
salt.states.grafana4_user, 3477
salt.states.grafana_dashboard, 3478
salt.states.grafana_datasource, 3479
salt.states.grains, 3479
salt.states.group, 3482
salt.states.heat, 3483
salt.states.hg, 3484
salt.states.host, 3485
salt.states.http, 3486
salt.states.icinga2, 3487
salt.states.ifttt, 3488
salt.states.incron, 3489
salt.states.influxdb08_database, 3490
salt.states.influxdb08_user, 3490
salt.states.influxdb_continuous_query, 3491
salt.states.influxdb_database, 3491
salt.states.influxdb_retention_policy, 3492
salt.states.influxdb_user, 3492
salt.states.infoblox_a, 3493
salt.states.infoblox_cname, 3493
salt.states.infoblox_host_record, 3494
salt.states.infoblox_range, 3495
salt.states.ini_manage, 3496
salt.states.ipmi, 3498
salt.states.ipset, 3500
salt.states.iptables, 3501
salt.states.jboss7, 3507
salt.states.jenkins, 3509
salt.states.junos, 3509
salt.states.kapacitor, 3514
salt.states.kernelpkg, 3515
salt.states.keyboard, 3517
salt.states.keystone, 3517
salt.states.keystone_domain, 3520
salt.states.keystone_endpoint, 3521
salt.states.keystone_group, 3522
salt.states.keystone_project, 3522
salt.states.keystone_role, 3523
salt.states.keystone_role_grant, 3524
salt.states.keystone_service, 3524
salt.states.keystone_user, 3525
salt.states.keystore, 3526
salt.states.kmod, 3526
salt.states.kubernetes, 3527
salt.states.layman, 3530
salt.states.ldap, 3531
salt.states.libcloud_dns, 3534
salt.states.libcloud_loadbalancer, 3535
salt.states.libcloud_storage, 3536
salt.states.linux_acl, 3538
salt.states.locale, 3540
salt.states.logadm, 3540
salt.states.logrotate, 3541
salt.states.loop, 3541
salt.states.lvm, 3543
salt.states.lvs_server, 3544
salt.states.lvs_service, 3544
salt.states.lxc, 3545
salt.states.mac_assistive, 3548
salt.states.mac_keychain, 3549
salt.states.mac_xattr, 3550
salt.states.macdefaults, 3548
salt.states.macpackage, 3549
salt.states.makeconf, 3551
salt.states.marathon_app, 3551
salt.states.mdadm_raid, 3552
salt.states.memcached, 3553
salt.states.modjk, 3553
salt.states.modjk_worker, 3554
salt.states.module, 3555
salt.states.mongodb_database, 3560
salt.states.mongodb_user, 3560
salt.states.monit, 3561
salt.states.mount, 3562
salt.states.msteams, 3565
salt.states.mysql_database, 3566
salt.states.mysql_grants, 3566
salt.states.mysql_query, 3568
salt.states.mysql_user, 3569

salt.states.net_napalm_yang, 3602
salt.states.netacl, 3570
salt.states.netconfig, 3581
salt.states.netntp, 3590
salt.states.netsnmp, 3591
salt.states.netusers, 3592
salt.states.network, 3596
salt.states.neutron_network, 3603
salt.states.neutron_secgroup, 3604
salt.states.neutron_secgroup_rule, 3605
salt.states.neutron_subnet, 3606
salt.states.nfs_export, 3607
salt.states.nftables, 3608
salt.states.npm, 3611
salt.states.ntp, 3612
salt.states.nxos, 3613
salt.states.openstack_config, 3615
salt.states.openvswitch_bridge, 3615
salt.states.openvswitch_port, 3615
salt.states.opsgenie, 3616
salt.states.pagerduty, 3617
salt.states.pagerduty_escalation_policy, 3617
salt.states.pagerduty_schedule, 3618
salt.states.pagerduty_service, 3619
salt.states.pagerduty_user, 3620
salt.states.panos, 3621
salt.states.pcs, 3630
salt.states.pdbedit, 3637
salt.states.pecl, 3637
salt.states.pip_state, 3638
salt.states.pkg, 3642
salt.states.pkgbuild, 3658
salt.states.pkgng, 3660
salt.states.pkgrepo, 3661
salt.states.portage_config, 3664
salt.states.ports, 3665
salt.states.postgres_cluster, 3665
salt.states.postgres_database, 3666
salt.states.postgres_extension, 3667
salt.states.postgres_group, 3668
salt.states.postgres_initdb, 3669
salt.states.postgres_language, 3670
salt.states.postgres_privileges, 3670
salt.states.postgres_schema, 3673
salt.states.postgres_tablespace, 3674
salt.states.postgres_user, 3675
salt.states.powerpath, 3676
salt.states.probes, 3676
salt.states.process, 3678
salt.states.proxy, 3678
salt.states.pushover, 3678
salt.states.pyenv, 3679
salt.states.pyrax_queues, 3680
salt.states.quota, 3681
salt.states.rabbitmq_cluster, 3681
salt.states.rabbitmq_plugin, 3682
salt.states.rabbitmq_policy, 3682
salt.states.rabbitmq_upstream, 3683
salt.states.rabbitmq_user, 3684
salt.states.rabbitmq_vhost, 3685
salt.states.rbac_solaris, 3686
salt.states.rbenv, 3686
salt.states.rdp, 3688
salt.states.redismod, 3688
salt.states.reg, 3689
salt.states.rsync, 3694
salt.states.rvm, 3695
salt.states.salt_proxy, 3697
salt.states.saltmod, 3698
salt.states.saltutil, 3701
salt.states.schedule, 3704
salt.states.selinux, 3707
salt.states.serverdensity_device, 3709
salt.states.service, 3710
salt.states.slack, 3713
salt.states.smartos, 3714
salt.states.smtp, 3717
salt.states.snapper, 3718
salt.states.solrcloud, 3720
salt.states.splunk, 3720
salt.states.splunk_search, 3721
salt.states.sqlite3, 3721
salt.states.ssh_auth, 3723
salt.states.ssh_known_hosts, 3725
salt.states.stateconf, 3726
salt.states.status, 3727
salt.states.statuspage, 3727
salt.states.supervisord, 3729
salt.states.svn, 3729
salt.states.sysctl, 3730
salt.states.syslog_ng, 3731
salt.states.sysrc, 3732
salt.states.telemetry_alert, 3732
salt.states.test, 3733
salt.states.testinframod, 3736
salt.states.timezone, 3736
salt.states.tls, 3736
salt.states.tomcat, 3736
salt.states.trafficserver, 3739
salt.states.tuned, 3741
salt.states.uptime, 3741
salt.states.user, 3742
salt.states.vagrant, 3744
salt.states.vault, 3746
salt.states.vbox_guest, 3747
salt.states.victorops, 3747
salt.states.virt, 3748

salt.states.virtualenv_mod, 3755
salt.states.webutil, 3756
salt.states.win_certutil, 3756
salt.states.win_dacl, 3757
salt.states.win_dism, 3758
salt.states.win_dns_client, 3761
salt.states.win_firewall, 3761
salt.states.win_iis, 3763
salt.states.win_lgpo, 3769
salt.states.win_license, 3772
salt.states.win_network, 3772
salt.states.win_path, 3774
salt.states.win_pki, 3774
salt.states.win_powercfg, 3775
salt.states.win_servermanager, 3776
salt.states.win_smtp_server, 3779
salt.states.win_snmp, 3781
salt.states.win_system, 3782
salt.states.win_wua, 3784
salt.states.win_wusa, 3787
salt.states.winrepo, 3787
salt.states.x509, 3788
salt.states.xml, 3792
salt.states.xmpp, 3793
salt.states.zabbix_action, 3793
salt.states.zabbix_host, 3795
salt.states.zabbix_hostgroup, 3796
salt.states.zabbix_mediatype, 3797
salt.states.zabbix_template, 3798
salt.states.zabbix_user, 3800
salt.states.zabbix_usergroup, 3802
salt.states.zabbix_valuemap, 3803
salt.states.zcbuildout, 3804
salt.states.zenoss, 3805
salt.states.zfs, 3806
salt.states.zk_concurrency, 3805
salt.states.zone, 3809
salt.states.zookeeper, 3814
salt.states.zpool, 3816

t

salt.thorium.calc, 3819
salt.thorium.check, 3820
salt.thorium.file, 3824
salt.thorium.key, 3825
salt.thorium.local, 3825
salt.thorium.reg, 3826
salt.thorium.runner, 3827
salt.thorium.status, 3827
salt.thorium.timer, 3828
salt.thorium.wheel, 3828
salt.tokens.localfs, 3828
salt.tokens.rediscluster, 3829
salt.tops.cobbler, 3830
salt.tops.ext_nodes, 3830
salt.tops.mongo, 3831
salt.tops.reclass_adapter, 3832
salt.tops.saltclass, 3832
salt.tops.varstack_top, 3836

u

salt.utils.aggregation, 3996
salt.utils.extend, 3928

w

salt.wheel.config, 3837
salt.wheel.error, 3838
salt.wheel.file_roots, 3838
salt.wheel.key, 3838
salt.wheel.minions, 3842
salt.wheel.pillar_roots, 3842

INDEX

Symbols

--args-separator=ARGS_SEPARATOR
  salt command line option, 971
--askpass
  salt-ssh command line option, 986
--async
  salt command line option, 970
--auto-create
  salt-key command line option, 979
--debug
  salt-extend command line option, 976
--description, -d
  salt-extend command line option, 976
--extension, -e
  salt-extend command line option, 976
--extra-filerefs=EXTRA_FILEREFS
  salt-ssh command line option, 985
--file-root=FILE_ROOT
  salt-call command line option, 968
--force-color
  salt command line option, 972
  salt-call command line option, 969
  salt-cloud command line option, 725
  salt-key command line option, 978
  salt-ssh command line option, 987
--gen-keys-dir=GEN_KEYS_DIR
  salt-key command line option, 979
--gen-keys=GEN_KEYS
  salt-key command line option, 979
--gen-signature
  salt-key command line option, 979
--grain-pcre
  salt command line option, 971
  salt-cp command line option, 974
--hard-crash
  salt-call command line option, 967
  salt-key command line option, 977
  salt-run command line option, 983
  salt-ssh command line option, 984
--hide-timeout
  salt command line option, 970
--id=ID
  salt-call command line option, 968
--identities-only
  salt-ssh command line option, 986
--include-all
  salt-key command line option, 978
--jid=JID
  salt-ssh command line option, 985
--key-deploy
  salt-ssh command line option, 986
--keysize=KEYSIZE
  salt-key command line option, 979
--list-images=LIST_IMAGES
  salt-cloud command line option, 724
--list-locations=LIST_LOCATIONS
  salt-cloud command line option, 724
--list-profiles
  salt-cloud command line option, 724
--list-providers
  salt-cloud command line option, 724
--list-sizes=LIST_SIZES
  salt-cloud command line option, 724
--local
  salt-call command line option, 968
--log-file-level=LOG_LEVEL_LOGFILE
  salt command line option, 971
  salt-api command line option, 990
  salt-call command line option, 968
  salt-cp command line option, 974
  salt-key command line option, 977
  salt-master command line option, 980
  salt-minion command line option, 981
  salt-proxy command line option, 982
  salt-run command line option, 984
  salt-ssh command line option, 986
  salt-syndic command line option, 988
  spm command line option, 991
--log-file=LOG_FILE
  salt command line option, 971
  salt-api command line option, 990
  salt-call command line option, 968
  salt-cp command line option, 974
  salt-key command line option, 977

salt-master command line option, 980
salt-minion command line option, 981
salt-proxy command line option, 982
salt-run command line option, 984
salt-ssh command line option, 986
salt-syndic command line option, 988
spm command line option, 991

--master=MASTER
salt-call command line option, 968

--max-procs
salt-ssh command line option, 985

--metadata
salt-call command line option, 968

--min-extra-modules=MIN_EXTRA_MODS
salt-ssh command line option, 985

--name, -n
salt-extend command line option, 976

--no-color
salt command line option, 972
salt-call command line option, 969
salt-cloud command line option, 725
salt-key command line option, 978
salt-ssh command line option, 987

--no-host-keys
salt-ssh command line option, 986

--no-merge
salt-extend command line option, 976

--out
salt command line option, 972
salt-call command line option, 969
salt-cloud command line option, 724
salt-key command line option, 977
salt-ssh command line option, 987

--out-file-append, --output-file-append
salt command line option, 972
salt-call command line option, 969
salt-cloud command line option, 724
salt-key command line option, 978
salt-ssh command line option, 987

--out-file=OUTPUT_FILE, --output-file=OUTPUT_FILE
salt command line option, 972
salt-call command line option, 969
salt-cloud command line option, 724
salt-key command line option, 977
salt-ssh command line option, 987

--out-indent OUTPUT_INDENT, --output-indent OUTPUT_INDENT
salt command line option, 972
salt-call command line option, 969
salt-cloud command line option, 724
salt-key command line option, 977
salt-ssh command line option, 987

--passwd
salt-ssh command line option, 986

--pid-file PIDFILE
salt-master command line option, 980
salt-minion command line option, 981
salt-proxy command line option, 982
salt-syndic command line option, 988

--pid-file=PIDFILE
salt-api command line option, 990

--pillar-root=PILLAR_ROOT
salt-call command line option, 968

--priv-passwd=SSH_PRIV_PASSWD
salt-ssh command line option, 985

--priv=PRIV
salt-key command line option, 979

--priv=SSH_PRIV
salt-ssh command line option, 985

--proxyid
salt-proxy command line option, 982

--pub=PUB
salt-key command line option, 979

--python2-bin=PYTHON2_BIN
salt-ssh command line option, 985

--python3-bin=PYTHON3_BIN
salt-ssh command line option, 985

--refresh, --refresh-cache
salt-ssh command line option, 985

--refresh-grains-cache
salt-call command line option, 968

--retcode-passthrough
salt-call command line option, 968

--return RETURNER
salt-call command line option, 968

--return=RETURNER
salt command line option, 970

--roster
salt-ssh command line option, 984

--roster-file
salt-ssh command line option, 985

--rotate-aes-key=ROTATE_AES_KEY
salt-key command line option, 977

--salt-directory, -o
salt-extend command line option, 976

--scan-ports=SSH_SCAN_PORTS
salt-ssh command line option, 986

--scan-timeout=SSH_SCAN_TIMEOUT
salt-ssh command line option, 986

--script-args=SCRIPT_ARGS
salt-cloud command line option, 723

--set-password=<USERNAME> <PROVIDER>
salt-cloud command line option, 724

--show-deploy-args
salt-cloud command line option, 723

--signature-path=SIGNATURE_PATH

salt-key command line option, 979
--skip-grains
salt-call command line option, 968
--state-output=STATE_OUTPUT, --state_output=STATE_OUTPUT
salt command line option, 972
salt-call command line option, 969
salt-cloud command line option, 725
salt-key command line option, 978
salt-ssh command line option, 987
--state-verbose=STATE_VERBOSE, --state_verbose=STATE_VERBOSE
salt command line option, 972
salt-call command line option, 969
salt-cloud command line option, 725
salt-key command line option, 978
salt-ssh command line option, 987
--subset=SUBSET
salt command line option, 970
--sudo
salt-ssh command line option, 986
--thin-extra-modules=THIN_EXTRA_MODS
salt-ssh command line option, 985
--user=SSH_USER
salt-ssh command line option, 986
--version
salt command line option, 970
salt-api command line option, 990
salt-call command line option, 967
salt-cloud command line option, 722
salt-cp command line option, 974
salt-key command line option, 977
salt-master command line option, 980
salt-minion command line option, 981
salt-proxy command line option, 982
salt-run command line option, 983
salt-ssh command line option, 984
salt-syndic command line option, 988
--versions-report
salt command line option, 970
salt-api command line option, 990
salt-call command line option, 967
salt-cloud command line option, 722
salt-cp command line option, 974
salt-key command line option, 977
salt-master command line option, 980
salt-minion command line option, 981
salt-proxy command line option, 982
salt-run command line option, 983
salt-ssh command line option, 984
salt-syndic command line option, 988
-A, --accept-all
salt-key command line option, 978
-C, --chunked
salt-cp command line option, 975
-C, --compound
salt command line option, 972
-D, --delete-all
salt-key command line option, 978
-E, --pcre
salt command line option, 971
salt-cp command line option, 974
salt-ssh command line option, 986
-F, --finger-all
salt-key command line option, 979
-F, --full-query
salt-cloud command line option, 723
-G, --grain
salt command line option, 971
salt-cp command line option, 974
-H, --hard
salt-cloud command line option, 723
-I, --pillar
salt command line option, 972
-L LOCATION, --location=LOCATION
salt-cloud command line option, 723
-L, --list
salt command line option, 971
salt-cp command line option, 974
-L, --list-all
salt-key command line option, 978
-N, --nodegroup
salt command line option, 971
salt-cp command line option, 974
-P, --parallel
salt-cloud command line option, 723
-P, --print-all
salt-key command line option, 978
-Q, --query
salt-cloud command line option, 723
-R, --range
salt command line option, 971
salt-cp command line option, 975
-R, --reject-all
salt-key command line option, 978
-S, --ipcidr
salt command line option, 972
-S, --select-query
salt-cloud command line option, 724
-T, --make-token
salt command line option, 970
-W, --rand-thin-dir
salt-ssh command line option, 985
-a ACCEPT, --accept=ACCEPT
salt-key command line option, 978
-a ACTION, --action=ACTION
salt-cloud command line option, 723
-a EAUTH, --auth=EAUTH

salt command line option, 970
-b BATCH, --batch-size=BATCH
salt command line option, 970
-c CONFIG_DIR, --config-dir=CONFIG_dir
salt command line option, 970
salt-api command line option, 990
salt-call command line option, 967
salt-cloud command line option, 722
salt-cp command line option, 974
salt-key command line option, 977
salt-master command line option, 980
salt-minion command line option, 981
salt-proxy command line option, 982
salt-run command line option, 983
salt-ssh command line option, 984
salt-syndic command line option, 988
-d DELETE, --delete=DELETE
salt-key command line option, 978
-d, --daemon
salt-api command line option, 990
salt-master command line option, 980
salt-minion command line option, 981
salt-proxy command line option, 982
salt-syndic command line option, 988
-d, --destroy
salt-cloud command line option, 723
-d, --doc, --documentation
salt command line option, 971
salt-call command line option, 968
salt-run command line option, 983
-f FINGER, --finger=FINGER
salt-key command line option, 979
-f <FUNC-NAME> <PROVIDER>, --function=<FUNC-NAME> <PROVIDER>
salt-cloud command line option, 723
-f, --force
spm command line option, 991
-g, --grains
salt-call command line option, 967
-h, --help
salt command line option, 970
salt-api command line option, 990
salt-call command line option, 967
salt-cloud command line option, 722
salt-cp command line option, 974
salt-key command line option, 977
salt-master command line option, 980
salt-minion command line option, 981
salt-proxy command line option, 982
salt-run command line option, 983
salt-ssh command line option, 984
salt-syndic command line option, 988
-i, --ignore-host-keys
salt-ssh command line option, 985
-k, --keep-tmp
salt-cloud command line option, 723
-l ARG, --list=ARG
salt-key command line option, 978
-l LOG_LEVEL, --log-level=LOG_LEVEL
salt command line option, 971
salt-api command line option, 990
salt-call command line option, 968
salt-cp command line option, 974
salt-master command line option, 980
salt-minion command line option, 981
salt-proxy command line option, 982
salt-run command line option, 984
salt-ssh command line option, 986
salt-syndic command line option, 988
spm command line option, 991
-m MAP, --map=MAP
salt-cloud command line option, 723
-m MODULE_DIRS, --module-dirs=MODULE_DIRS
salt-call command line option, 967
-n, --no-compression
salt-cp command line option, 975
-p PRINT, --print=PRINT
salt-key command line option, 978
-p PROFILE, --profile=PROFILE
salt-cloud command line option, 723
-q, --quiet
salt-key command line option, 977
-r REJECT, --reject=REJECT
salt-key command line option, 978
-r, --raw, --raw-shell
salt-ssh command line option, 984
-s, --static
salt command line option, 970
salt-ssh command line option, 985
-t TIMEOUT, --timeout=TIMEOUT
salt command line option, 970
salt-cp command line option, 974
salt-run command line option, 983
-t, --regen-thin, --thin
salt-ssh command line option, 985
-u USER, --user=USER
salt-key command line option, 977
salt-master command line option, 980
salt-minion command line option, 981
salt-proxy command line option, 982
salt-syndic command line option, 988
-u, --update-bootstrap
salt-cloud command line option, 723
-v VERBOSE, --verbose
salt command line option, 970
-v, --verbose
salt-ssh command line option, 985

-w, --wipe
  salt-ssh command line option, 985
-y, --assume-yes
  salt-cloud command line option, 723
  spm command line option, 991
-y, --yes
  salt-key command line option, 977
__virtual__, **5403**

A

A() (*in module salt.modules.dig*), 1601
A() (*in module salt.modules.dnsutil*), 1608
a2disconf() (*in module salt.modules.deb_apache*), 1591
a2dismod() (*in module salt.modules.deb_apache*), 1591
a2dismod() (*in module salt.modules.suse_apache*), 2568
a2dissite() (*in module salt.modules.deb_apache*), 1591
a2enconf() (*in module salt.modules.deb_apache*), 1591
a2enmod() (*in module salt.modules.deb_apache*), 1591
a2enmod() (*in module salt.modules.suse_apache*), 2568
a2ensite() (*in module salt.modules.deb_apache*), 1591
AAAA() (*in module salt.modules.dig*), 1601
AAAA() (*in module salt.modules.dnsutil*), 1608
abort_import() (*in module salt.modules.solr*), 2528
absent() (*in module salt.states.alias*), 3183
absent() (*in module salt.states.at*), 3195
absent() (*in module salt.states.aws_sqs*), 3199
absent() (*in module salt.states.beacon*), 3221
absent() (*in module salt.states.boto3_elasticsearch*), 3242
absent() (*in module salt.states.boto_apigateway*), 3251
absent() (*in module salt.states.boto_asg*), 3258
absent() (*in module salt.states.boto_cfn*), 3261
absent() (*in module salt.states.boto_cloudtrail*), 3264
absent() (*in module salt.states.boto_cloudwatch_alarm*), 3265
absent() (*in module salt.states.boto_cloudwatch_event*), 3267
absent() (*in module salt.states.boto_datapipeline*), 3269
absent() (*in module salt.states.boto_dynamodb*), 3272
absent() (*in module salt.states.boto_elasticache*), 3281
absent() (*in module salt.states.boto_elasticsearch_domain*), 3284
absent() (*in module salt.states.boto_elb*), 3289
absent() (*in module salt.states.boto_iam_role*), 3299
absent() (*in module salt.states.boto_kinesis*), 3304
absent() (*in module salt.states.boto_lc*), 3311
absent() (*in module salt.states.boto_rds*), 3313
absent() (*in module salt.states.boto_route53*), 3317
absent() (*in module salt.states.boto_s3_bucket*), 3320
absent() (*in module salt.states.boto_secgroup*), 3323
absent() (*in module salt.states.boto_sns*), 3324
absent() (*in module salt.states.boto_sqs*), 3326
absent() (*in module salt.states.boto_vpc*), 3328
absent() (*in module salt.states.chronos_job*), 3340
absent() (*in module salt.states.cloud*), 3343
absent() (*in module salt.states.cron*), 3357
absent() (*in module salt.states.ddns*), 3362
absent() (*in module salt.states.docker_container*), 3370
absent() (*in module salt.states.docker_image*), 3391
absent() (*in module salt.states.docker_network*), 3393
absent() (*in module salt.states.docker_volume*), 3399
absent() (*in module salt.states.drac*), 3400
absent() (*in module salt.states.elasticsearch_index*), 3402
absent() (*in module salt.states.elasticsearch_index_template*), 3403
absent() (*in module salt.states.file*), 3419
absent() (*in module salt.states.github*), 3463
absent() (*in module salt.states.glance_image*), 3466
absent() (*in module salt.states.gpg*), 3470
absent() (*in module salt.states.grafana4_dashboard*), 3474
absent() (*in module salt.states.grafana4_datasource*), 3475
absent() (*in module salt.states.grafana4_org*), 3476
absent() (*in module salt.states.grafana4_user*), 3477
absent() (*in module salt.states.grafana_dashboard*), 3478
absent() (*in module salt.states.grafana_datasource*), 3479
absent() (*in module salt.states.grains*), 3480
absent() (*in module salt.states.group*), 3483
absent() (*in module salt.states.heat*), 3484
absent() (*in module salt.states.host*), 3486
absent() (*in module salt.states.incron*), 3489
absent() (*in module salt.states.influxdb08_database*), 3490
absent() (*in module salt.states.influxdb08_user*), 3490
absent() (*in module salt.states.influxdb_continuous_query*), 3491
absent() (*in module salt.states.influxdb_database*), 3491
absent() (*in module salt.states.influxdb_retention_policy*), 3492
absent() (*in module salt.states.influxdb_user*), 3492
absent() (*in module salt.states.infoblox_a*), 3493
absent() (*in module salt.states.infoblox_cname*), 3493
absent() (*in module salt.states.infoblox_host_record*), 3494
absent() (*in module salt.states.infoblox_range*), 3495
absent() (*in module salt.states.ipset*), 3501
absent() (*in module salt.states.jenkins*), 3509
absent() (*in module salt.states.keystone_domain*), 3520

`absent()` (*in module salt.states.keystone_endpoint*), 3521
`absent()` (*in module salt.states.keystone_group*), 3522
`absent()` (*in module salt.states.keystone_project*), 3523
`absent()` (*in module salt.states.keystone_role*), 3523
`absent()` (*in module salt.states.keystone_role_grant*), 3524
`absent()` (*in module salt.states.keystone_service*), 3524
`absent()` (*in module salt.states.keystone_user*), 3525
`absent()` (*in module salt.states.kmod*), 3527
`absent()` (*in module salt.states.layman*), 3530
`absent()` (*in module salt.states.linux_acl*), 3539
`absent()` (*in module salt.states.lvs_server*), 3544
`absent()` (*in module salt.states.lvs_service*), 3544
`absent()` (*in module salt.states.lxc*), 3545
`absent()` (*in module salt.states.macdefaults*), 3548
`absent()` (*in module salt.states.makeconf*), 3551
`absent()` (*in module salt.states.marathon_app*), 3551
`absent()` (*in module salt.states.mdadm_raid*), 3552
`absent()` (*in module salt.states.memcached*), 3553
`absent()` (*in module salt.states.mongodb_database*), 3560
`absent()` (*in module salt.states.mongodb_user*), 3561
`absent()` (*in module salt.states.mysql_database*), 3566
`absent()` (*in module salt.states.mysql_grants*), 3567
`absent()` (*in module salt.states.mysql_user*), 3569
`absent()` (*in module salt.states.neutron_network*), 3604
`absent()` (*in module salt.states.neutron_secgroup*), 3605
`absent()` (*in module salt.states.neutron_secgroup_rule*), 3605
`absent()` (*in module salt.states.neutron_subnet*), 3606
`absent()` (*in module salt.states.nfs_export*), 3608
`absent()` (*in module salt.states.openstack_config*), 3615
`absent()` (*in module salt.states.openvswitch_bridge*), 3615
`absent()` (*in module salt.states.openvswitch_port*), 3615
`absent()` (*in module salt.states.pagerduty_escalation_policy*), 3618
`absent()` (*in module salt.states.pagerduty_schedule*), 3619
`absent()` (*in module salt.states.pagerduty_service*), 3619
`absent()` (*in module salt.states.pagerduty_user*), 3620
`absent()` (*in module salt.states.pdbedit*), 3637
`absent()` (*in module salt.states.pkgrepo*), 3662
`absent()` (*in module salt.states.postgres_cluster*), 3666
`absent()` (*in module salt.states.postgres_database*), 3666
`absent()` (*in module salt.states.postgres_extension*), 3667
`absent()` (*in module salt.states.postgres_group*), 3668
`absent()` (*in module salt.states.postgres_language*), 3670
`absent()` (*in module salt.states.postgres_privileges*), 3671
`absent()` (*in module salt.states.postgres_schema*), 3673
`absent()` (*in module salt.states.postgres_tablespace*), 3674
`absent()` (*in module salt.states.postgres_user*), 3675
`absent()` (*in module salt.states.process*), 3678
`absent()` (*in module salt.states.pyenv*), 3680
`absent()` (*in module salt.states.pyrax_queues*), 3681
`absent()` (*in module salt.states.rabbitmq_policy*), 3682
`absent()` (*in module salt.states.rabbitmq_upstream*), 3683
`absent()` (*in module salt.states.rabbitmq_user*), 3685
`absent()` (*in module salt.states.rabbitmq_vhost*), 3685
`absent()` (*in module salt.states.rbenv*), 3687
`absent()` (*in module salt.states.redismod*), 3688
`absent()` (*in module salt.states.reg*), 3690
`absent()` (*in module salt.states.schedule*), 3706
`absent()` (*in module salt.states.splunk*), 3720
`absent()` (*in module salt.states.splunk_search*), 3721
`absent()` (*in module salt.states.ssh_auth*), 3724
`absent()` (*in module salt.states.ssh_known_hosts*), 3726
`absent()` (*in module salt.states.sysrc*), 3732
`absent()` (*in module salt.states.telemetry_alert*), 3733
`absent()` (*in module salt.states.user*), 3742
`absent()` (*in module salt.states.win_dacl*), 3758
`absent()` (*in module salt.states.win_path*), 3774
`absent()` (*in module salt.states.zabbix_action*), 3793
`absent()` (*in module salt.states.zabbix_host*), 3795
`absent()` (*in module salt.states.zabbix_hostgroup*), 3796
`absent()` (*in module salt.states.zabbix_mediatype*), 3797
`absent()` (*in module salt.states.zabbix_template*), 3798
`absent()` (*in module salt.states.zabbix_user*), 3801
`absent()` (*in module salt.states.zabbix_usergroup*), 3802
`absent()` (*in module salt.states.zabbix_valuemap*), 3803
`absent()` (*in module salt.states.zone*), 3811
`absent()` (*in module salt.states.zookeeper*), 3814
`absent()` (*in module salt.states.zpool*), 3817
`accept()` (*in module salt.wheel.key*), 3839
`accept_dict()` (*in module salt.wheel.key*), 3839
`accept_vpc_peering_connection()` (*in module salt.modules.boto_vpc*), 1448
`accept_vpc_peering_connection()` (*in module salt.states.boto_vpc*), 3328
`acceptance_wait_time`
  `conf/minion`, 136
`acceptance_wait_time_max`
  `conf/minion`, 137
`access()` (*in module salt.modules.file*), 1703
`account_policy()` (*in module salt.states.boto_iam*), 3294
`accumulated()` (*in module salt.states.file*), 3419
`acl()` (*in module salt.auth.django*), 998

`acl_clone()` (*in module salt.modules.consul*), 1560
`acl_create()` (*in module salt.modules.consul*), 1560
`acl_delete()` (*in module salt.modules.consul*), 1560
`acl_info()` (*in module salt.modules.consul*), 1561
`acl_list()` (*in module salt.modules.consul*), 1561
`acl_update()` (*in module salt.modules.consul*), 1561
`acls()` (*in module salt.states.zookeeper*), 3814
`acpi_tables()` (*in module salt.modules.osquery*), 2288
`acquit()` (*in module salt.modules.solaris_fmadm*), 2513
`action()` (*in module salt.modules.cloud*), 1519
`action()` (*in module salt.runners.cloud*), 3102
`action()` (*salt.cloud.CloudClient method*), 3852
`activate()` (*in module salt.modules.namecheap_ssl*), 2118
`activate()` (*in module salt.modules.win_license*), 2772
`activate()` (*in module salt.states.modjk_worker*), 3554
`activate()` (*in module salt.states.win_license*), 3772
`activate_api_deployment()` (*in module salt.modules.boto_apigateway*), 1330
`activate_backup_image()` (*in module salt.modules.cimc*), 1507
`activate_pipeline()` (*in module salt.modules.boto_datapipeline*), 1358
`active()` (*in module salt.modules.kernelpkg_linux_apt*), 1925
`active()` (*in module salt.modules.kernelpkg_linux_yum*), 1926
`active()` (*in module salt.modules.mount*), 2091
`active()` (*in module salt.modules.tuned*), 2625
`active()` (*in module salt.runners.jobs*), 3116
`active_log_format()` (*in module salt.states.win_smtp_server*), 3779
`active_tcp()` (*in module salt.modules.network*), 2215
`add()` (*in module salt.modules.aix_group*), 1203
`add()` (*in module salt.modules.beacons*), 1283
`add()` (*in module salt.modules.bridge*), 1464
`add()` (*in module salt.modules.btrfs*), 1466
`add()` (*in module salt.modules.git*), 1757
`add()` (*in module salt.modules.groupadd*), 1836
`add()` (*in module salt.modules.ipset*), 1896
`add()` (*in module salt.modules.keystore*), 1944
`add()` (*in module salt.modules.layman*), 1952
`add()` (*in module salt.modules.ldap3*), 1954
`add()` (*in module salt.modules.mac_group*), 2027
`add()` (*in module salt.modules.mac_user*), 2056
`add()` (*in module salt.modules.mdadm_raid*), 2074
`add()` (*in module salt.modules.memcached*), 2077
`add()` (*in module salt.modules.pw_group*), 2405
`add()` (*in module salt.modules.pw_user*), 2406
`add()` (*in module salt.modules.schedule*), 2481
`add()` (*in module salt.modules.smartos_nictagadm*), 2502
`add()` (*in module salt.modules.solaris_group*), 2515
`add()` (*in module salt.modules.solaris_user*), 2518
`add()` (*in module salt.modules.supervisord*), 2566
`add()` (*in module salt.modules.svn*), 2568
`add()` (*in module salt.modules.useradd*), 2630
`add()` (*in module salt.modules.win_groupadd*), 2752
`add()` (*in module salt.modules.win_path*), 2776
`add()` (*in module salt.modules.win_useradd*), 2833
`add()` (*in module salt.modules.zpool*), 2916
`add()` (*in module salt.runners.reactor*), 3142
`add()` (*in module salt.thorium.calc*), 3819
`add()` (*salt.modules.win_timezone.TzMapper method*), 2831
`add_ace()` (*in module salt.modules.win_dacl*), 2721
`add_action()` (*in module salt.modules.win_task*), 2820
`add_argument()` (*salt.modules.syslog_ng.TypedParameterValue method*), 2580
`add_capability()` (*in module salt.modules.win_dism*), 2724
`add_capacity_to_diskgroup()` (*in module salt.modules.vsphere*), 2682
`add_child()` (*salt.modules.syslog_ng.Statement method*), 2579
`add_comment()` (*in module salt.modules.jira_mod*), 1912
`add_config()` (*in module salt.proxy.nxos*), 3074
`add_config_lock()` (*in module salt.modules.panos*), 2305
`add_config_lock()` (*in module salt.states.panos*), 3622
`add_device()` (*in module salt.modules.zenoss*), 2898
`add_dns()` (*in module salt.modules.win_dns_client*), 2728
`add_export()` (*in module salt.modules.nfs3*), 2243
`add_feature()` (*in module salt.modules.win_dism*), 2724
`add_gateway_router()` (*in module salt.modules.neutron*), 2222
`add_host()` (*in module salt.cloud.clouds.vmware*), 1137
`add_host()` (*in module salt.modules.ddns*), 1590
`add_host()` (*in module salt.modules.hosts*), 1849
`add_host()` (*in module salt.modules.omapi*), 2269
`add_host()` (*in module salt.runners.ddns*), 3103
`add_host_to_dvs()` (*in module salt.modules.vsphere*), 2682
`add_input_endpoint()` (*in module salt.cloud.clouds.msazure*), 1071
`add_interface()` (*in module salt.modules.firewalld*), 1731
`add_interface_router()` (*in module salt.modules.neutron*), 2222
`add_license()` (*in module salt.modules.powerpath*), 2387
`add_license()` (*in module salt.modules.vsphere*), 2684

add_lock() (*in module salt.modules.zypperpkg*), 2924
add_management_certificate() (*in module salt.cloud.clouds.msazure*), 1071
add_masquerade() (*in module salt.modules.firewalld*), 1731
add_package() (*in module salt.modules.win_dism*), 2724
add_parameter() (*salt.modules.syslog_ng.Option method*), 2578
add_permission() (*in module salt.modules.boto_lambda*), 1419
add_pkg() (*in module salt.modules.pkg_resource*), 2349
add_platform() (*in module salt.runners.asam*), 3096
add_pool_member() (*in module salt.modules.bigip*), 1285
add_pool_member() (*in module salt.runners.f5*), 3110
add_pool_member() (*in module salt.states.bigip*), 3222
add_pool_member() (*salt.runners.f5.F5Mgmt method*), 3110
add_port() (*in module salt.modules.firewalld*), 1731
add_port_fwd() (*in module salt.modules.firewalld*), 1731
add_proxymodule_to_opts
    conf/proxy, 168
add_record() (*in module salt.modules.boto_route53*), 1429
add_repo() (*in module salt.modules.github*), 1799
add_repo() (*in module salt.modules.xbpspkg*), 2857
add_repo_key() (*in module salt.modules.aptpkg*), 1216
add_resource() (*in module salt.modules.zonecfg*), 2910
add_rich_rule() (*in module salt.modules.firewalld*), 1732
add_rule() (*in module salt.modules.win_firewall*), 2745
add_rule() (*in module salt.states.win_firewall*), 3761
add_server() (*in module salt.modules.lvs*), 1984
add_service() (*in module salt.modules.firewalld*), 1732
add_service() (*in module salt.modules.lvs*), 1984
add_service_certificate() (*in module salt.cloud.clouds.msazure*), 1072
add_service_port() (*in module salt.modules.firewalld*), 1732
add_service_protocol() (*in module salt.modules.firewalld*), 1732
add_source() (*in module salt.modules.chocolatey*), 1500
add_source() (*in module salt.modules.firewalld*), 1732
add_store() (*in module salt.modules.win_certutil*), 2720
add_store() (*in module salt.states.win_certutil*), 3757
add_svc_avail_path() (*in module salt.modules.runit*), 2453
add_tags() (*in module salt.modules.boto3_elasticsearch*), 1308
add_tags() (*in module salt.modules.boto_cloudtrail*), 1349
add_tags() (*in module salt.modules.boto_elasticsearch_domain*), 1383
add_tags_to_resource() (*in module salt.modules.boto3_elasticache*), 1302
add_team() (*in module salt.modules.github*), 1800
add_team_member() (*in module salt.modules.github*), 1800
add_team_repo() (*in module salt.modules.github*), 1800
add_trigger() (*in module salt.modules.win_task*), 2821
add_user() (*in module salt.modules.github*), 1801
add_user() (*in module salt.modules.rabbitmq*), 2412
add_user_to_group() (*in module salt.modules.boto_iam*), 1393
add_value() (*salt.modules.syslog_ng.TypedParameter method*), 2579
add_veth() (*in module salt.modules.lxc*), 1986
add_vhost() (*in module salt.modules.rabbitmq*), 2413
add_volume_bricks() (*in module salt.modules.glusterfs*), 1811
add_volume_bricks() (*in module salt.states.glusterfs*), 3466
added (*salt.modules.file.AttrChanges attribute*), 1703
added() (*salt.states.cyg.DictDiffer method*), 3360
addgroup() (*in module salt.modules.win_useradd*), 2834
addGroupsToKey() (*in module salt.runners.spacewalk*), 3151
addif() (*in module salt.modules.bridge*), 1464
additions_install() (*in module salt.modules.vbox_guest*), 2641
additions_installed() (*in module salt.states.vbox_guest*), 3747
additions_mount() (*in module salt.modules.vbox_guest*), 2641
additions_remove() (*in module salt.modules.vbox_guest*), 2641
additions_removed() (*in module salt.states.vbox_guest*), 3747
additions_umount() (*in module salt.modules.vbox_guest*), 2642
additions_version() (*in module salt.modules.vbox_guest*), 2642
address_() (*in module salt.modules.bluez_bluetooth*), 1299
address_exists() (*in module salt.states.panos*),

3622
address_group_exists() (*in module salt.states.panos*), 3623
adduser() (*in module salt.modules.aix_group*), 1203
adduser() (*in module salt.modules.groupadd*), 1837
adduser() (*in module salt.modules.mac_group*), 2027
adduser() (*in module salt.modules.pw_group*), 2405
adduser() (*in module salt.modules.win_groupadd*), 2753
admin_password() (*in module salt.proxy.fx2*), 3064
admin_password_present() (*in module salt.states.zabbix_user*), 3801
admin_username() (*in module salt.proxy.fx2*), 3064
agent_check_deregister() (*in module salt.modules.consul*), 1561
agent_check_fail() (*in module salt.modules.consul*), 1562
agent_check_pass() (*in module salt.modules.consul*), 1562
agent_check_register() (*in module salt.modules.consul*), 1562
agent_check_warn() (*in module salt.modules.consul*), 1563
agent_checks() (*in module salt.modules.consul*), 1563
agent_join() (*in module salt.modules.consul*), 1563
agent_leave() (*in module salt.modules.consul*), 1563
agent_maintenance() (*in module salt.modules.consul*), 1564
agent_members() (*in module salt.modules.consul*), 1564
agent_self() (*in module salt.modules.consul*), 1564
agent_service_deregister() (*in module salt.modules.consul*), 1564
agent_service_maintenance() (*in module salt.modules.consul*), 1564
agent_service_register() (*in module salt.modules.consul*), 1565
agent_services() (*in module salt.modules.consul*), 1565
agent_settings() (*in module salt.states.win_snmp*), 3781
Aggregate (*class in salt.utils.aggregation*), 3997
aggregate() (*in module salt.utils.aggregation*), 3997
alarms() (*in module salt.modules.trafficserver*), 2622
alf() (*in module salt.modules.osquery*), 2288
alf_exceptions() (*in module salt.modules.osquery*), 2288
alf_explicit_auths() (*in module salt.modules.osquery*), 2288
alf_services() (*in module salt.modules.osquery*), 2288
alias() (*in module salt.states.solrcloud*), 3720
alias_absent() (*in module salt.states.boto_lambda*), 3306
alias_absent() (*in module salt.states.elasticsearch*), 3401
alias_create() (*in module salt.modules.elasticsearch*), 1681
alias_delete() (*in module salt.modules.elasticsearch*), 1682
alias_exists() (*in module salt.modules.boto_lambda*), 1419
alias_exists() (*in module salt.modules.elasticsearch*), 1682
alias_exists() (*in module salt.modules.solrcloud*), 2533
alias_get() (*in module salt.modules.elasticsearch*), 1682
alias_get_collections() (*in module salt.modules.solrcloud*), 2533
alias_present() (*in module salt.states.boto_lambda*), 3307
alias_present() (*in module salt.states.elasticsearch*), 3401
alias_set_collections() (*in module salt.modules.solrcloud*), 2534
align_check() (*in module salt.modules.parted_partition*), 2325
alive() (*in module salt.modules.napalm_mod*), 2125
alive() (*in module salt.proxy.junos*), 3065
alive() (*in module salt.proxy.napalm*), 3068
alive() (*in module salt.proxy.netmiko_px*), 3073
alive() (*in module salt.proxy.rest_sample*), 3085
alived() (*in module salt.runners.manage*), 3121
all_status() (*in module salt.modules.status*), 2558
allocate_eip_address() (*in module salt.modules.boto_ec2*), 1361
allow() (*in module salt.modules.apf*), 1212
allow() (*in module salt.modules.csf*), 1582
allow_icmp() (*in module salt.modules.firewalld*), 1732
allow_minion_key_revoke
  conf/master, 73
allow_missing_func() (*in module salt.executors.docker*), 1171
allow_port() (*in module salt.modules.csf*), 1582
allow_ports() (*in module salt.modules.csf*), 1582
allowed() (*in module salt.runners.manage*), 3121
AllowedDir (*class in salt.modules.inspectlib.entities*), 1876
alter_db() (*in module salt.modules.mysql*), 2099
alter_retention_policy() (*in module salt.modules.influxdbmod*), 1857
always_verify_signature
  conf/minion, 158
AnsibleModuleCaller (*class in salt.modules.ansiblegate*), 1208
AnsibleModuleResolver (*class in*

salt.modules.ansiblegate), 1208
AnsibleState (*class in salt.states.ansiblegate*), 3185
api() (*in module salt.runners.spacewalk*), 3151
api_exists() (*in module salt.modules.boto_apigateway*), 1330
api_logfile
    conf/master, 111
api_model_exists() (*in module salt.modules.boto_apigateway*), 1330
api_pidfile
    conf/master, 111
api_version() (*in module salt.modules.keystone*), 1929
apiinfo_version() (*in module salt.modules.zabbix*), 2878
app() (*in module salt.modules.marathon*), 2069
appdata_ptr (*salt.auth.pam.PamConv attribute*), 1002
append() (*in module salt.modules.file*), 1703
append() (*in module salt.modules.grains*), 1831
append() (*in module salt.modules.iptables*), 1898
append() (*in module salt.modules.nftables*), 2243
append() (*in module salt.states.file*), 3420
append() (*in module salt.states.grains*), 3480
append() (*in module salt.states.iptables*), 3505
append() (*in module salt.states.nftables*), 3610
append_cflags() (*in module salt.modules.makeconf*), 2060
append_cxxflags() (*in module salt.modules.makeconf*), 2061
append_dict_key_value
    jinja filters, 438
append_domain
    conf/minion, 131
append_domain() (*in module salt.grains.core*), 1177
append_emerge_default_opts() (*in module salt.modules.makeconf*), 2061
append_features() (*in module salt.modules.makeconf*), 2061
append_gentoo_mirrors() (*in module salt.modules.makeconf*), 2061
append_makeopts() (*in module salt.modules.makeconf*), 2061
append_minionid_config_dirs
    conf/minion, 132
append_to_package_conf() (*in module salt.modules.portage_config*), 2370
append_use_flags() (*in module salt.modules.portage_config*), 2370
append_var() (*in module salt.modules.makeconf*), 2062
applications() (*in module salt.modules.system_profiler*), 2591
apply() (*in module salt.wheel.config*), 3837
apply_() (*in module salt.modules.dockermod*), 1618
apply_() (*in module salt.modules.seed*), 2486
apply_() (*in module salt.modules.state*), 2547
apply_config() (*in module salt.modules.win_dsc*), 2729
apply_network_profile() (*in module salt.modules.lxc*), 1986
apply_network_settings() (*in module salt.modules.debian_ip*), 1596
apply_network_settings() (*in module salt.modules.nilrt_ip*), 2248
apply_network_settings() (*in module salt.modules.rh_ip*), 2441
apply_rollback() (*in module salt.modules.cisconso*), 1518
apply_rollback() (*in module salt.proxy.cisconso*), 3055
apply_security_groups() (*in module salt.modules.boto_elb*), 1386
apply_template() (*in module salt.utils.extend*), 3928
apply_template_on_contents() (*in module salt.modules.file*), 1704
apps() (*in module salt.modules.marathon*), 2069
apps() (*in module salt.modules.osquery*), 2288
apt_sources() (*in module salt.modules.osquery*), 2288
archive() (*in module salt.modules.git*), 1758
archive() (*in module salt.modules.hg*), 1847
arg() (*in module salt.modules.test*), 2600
arg() (*in module salt.runners.test*), 3156
arg_clean() (*in module salt.modules.test*), 2600
arg_repr() (*in module salt.modules.test*), 2600
arg_type() (*in module salt.modules.test*), 2600
args() (*in module salt.proxy.netmiko_px*), 3073
argspec() (*in module salt.modules.sysmod*), 2582
Argument (*class in salt.modules.syslog_ng*), 2577
ArgumentValueError, 3998
arp() (*in module salt.modules.napalm_network*), 2155
arp() (*in module salt.modules.network*), 2215
arp_cache() (*in module salt.modules.osquery*), 2288
ArtifactoryError, 1240
as_list (*salt.pillar.sql_base.SqlBaseExtPillar attribute*), 3034
assemble() (*in module salt.modules.mdadm_raid*), 2074
assertion() (*in module salt.modules.test*), 2600
assign() (*in module salt.modules.freebsd_sysctl*), 1738
assign() (*in module salt.modules.linux_sysctl*), 1979
assign() (*in module salt.modules.mac_sysctl*), 2048
assign() (*in module salt.modules.netbsd_sysctl*), 2202
assign() (*in module salt.modules.openbsd_sysctl*), 2270
assign_default_storage_policy_to_datastore() (*in module salt.modules.vsphere*), 2685
assign_floating_ip() (*in module salt.cloud.clouds.digitalocean*), 1040

`assign_issue()` (*in module salt.modules.jira_mod*), 1912

`assign_license()` (*in module salt.modules.vsphere*), 2685

`assign_private_ip_addresses()` (*in module salt.modules.boto_ec2*), 1362

`assign_templates()` (*in module salt.states.zabbix_host*), 3795

`associate_api_key_stagekeys()` (*in module salt.modules.boto_apigateway*), 1330

`associate_dhcp_options_to_vpc()` (*in module salt.modules.boto_vpc*), 1449

`associate_eip_address()` (*in module salt.modules.boto_ec2*), 1362

`associate_network_acl_to_subnet()` (*in module salt.modules.boto_vpc*), 1449

`associate_profile_to_role()` (*in module salt.modules.boto_iam*), 1394

`associate_route_table()` (*in module salt.modules.boto_vpc*), 1449

`associate_vpc_with_hosted_zone()` (*in module salt.modules.boto3_route53*), 1323

`asynchronous()` (*salt.runner.RunnerClient method*), 3850

`asynchronous()` (*salt.wheel.WheelClient method*), 3851

`at()` (*in module salt.modules.at*), 1242

`at()` (*in module salt.modules.at_solaris*), 1243

`atc()` (*in module salt.modules.at*), 1242

`atc()` (*in module salt.modules.at_solaris*), 1243

`atq()` (*in module salt.modules.at*), 1243

`atq()` (*in module salt.modules.at_solaris*), 1244

`atrm()` (*in module salt.modules.at*), 1243

`atrm()` (*in module salt.modules.at_solaris*), 1244

`attach()` (*in module salt.modules.zoneadm*), 2907

`attach()` (*in module salt.modules.zpool*), 2917

`attach_()` (*in module salt.modules.bcache*), 1280

`attach_disk()` (*in module salt.cloud.clouds.gce*), 1054

`attach_group_policy()` (*in module salt.modules.boto_iam*), 1394

`attach_lb()` (*in module salt.cloud.clouds.gce*), 1054

`attach_network_interface()` (*in module salt.modules.boto_ec2*), 1363

`attach_principal_policy()` (*in module salt.modules.boto_iot*), 1407

`attach_role_policy()` (*in module salt.modules.boto_iam*), 1394

`attach_subnets()` (*in module salt.modules.boto_elb*), 1386

`attach_tag()` (*in module salt.modules.vsphere*), 2685

`attach_usage_plan_to_apis()` (*in module salt.modules.boto_apigateway*), 1331

`attach_user_policy()` (*in module salt.modules.boto_iam*), 1394

`attach_volume()` (*in module salt.cloud.clouds.ec2*), 1046

`attach_volume()` (*in module salt.modules.boto_ec2*), 1363

`attach_volume()` (*in module salt.modules.libcloud_compute*), 1959

`attachable()` (*in module salt.modules.lxc*), 1986

`attached()` (*in module salt.states.zone*), 3811

`attr()` (*in module salt.modules.sysfs*), 2576

`attr_call()` (*in module salt.modules.test*), 2600

`AttrChanges` (*class in salt.modules.file*), 1703

`attributes()` (*in module salt.modules.extfs*), 1701

`audit()` (*in module salt.modules.pkgng*), 2354

`auth()` (*in module salt.auth.auto*), 997

`auth()` (*in module salt.auth.django*), 998

`auth()` (*in module salt.auth.file*), 1000

`auth()` (*in module salt.auth.keystone*), 1001

`auth()` (*in module salt.auth.ldap*), 1001

`auth()` (*in module salt.auth.mysql*), 1002

`auth()` (*in module salt.auth.pam*), 1003

`auth()` (*in module salt.auth.pki*), 1003

`auth()` (*in module salt.auth.rest*), 1004

`auth()` (*in module salt.auth.sharedsecret*), 1005

`auth()` (*in module salt.auth.yubico*), 1005

`auth()` (*in module salt.modules.keystone*), 1929

`auth()` (*in module salt.modules.pcs*), 2328

`auth()` (*in module salt.states.pcs*), 3633

`auth_add()` (*in module salt.modules.rbac_solaris*), 2422

`auth_events`
    `conf/master`, 64

`auth_get()` (*in module salt.modules.rbac_solaris*), 2422

`auth_keys()` (*in module salt.modules.ssh*), 2539

`auth_list()` (*in module salt.modules.rbac_solaris*), 2422

`auth_rm()` (*in module salt.modules.rbac_solaris*), 2423

`auth_safemode`
    `conf/minion`, 138

`auth_timeout`
    `conf/minion`, 138

`auth_traps_enabled()` (*in module salt.states.win_snmp*), 3781

`auth_tries`
    `conf/minion`, 138

`authenticate()` (*in module salt.auth.pam*), 1003

`authenticate()` (*in module salt.modules.lxd*), 2004

`AuthenticationError`, 3998

`AuthorizationError`, 3998

`authorize()` (*in module salt.modules.boto_secgroup*), 1441

`authorize_cache_security_group_ingress()` (*in module salt.modules.boto3_elasticache*), 1302

`authorize_cache_security_group_ingress()` (*in module salt.modules.boto_elasticache*), 1379

`auto()` (*in module salt.modules.alternatives*), 1206

`auto()` (*in module salt.states.alternatives*), 3184
`Auto-Order`, **5401**
`auto_accept`
    `conf/master`, 68
`autoload_dynamic_modules`
    `conf/minion`, 151
`automaster()` (*in module salt.modules.mount*), 2091
`autoreject_file`
    `conf/master`, 69
`autoremove()` (*in module salt.modules.aptpkg*), 1216
`autoremove()` (*in module salt.modules.pkgng*), 2354
`autosign_file`
    `conf/master`, 69
`autosign_grains`
    `conf/minion`, 158
`autosign_grains_dir`
    `conf/master`, 69
`autosign_timeout`
    `conf/master`, 68
`avail()` (*in module salt.modules.localemod*), 1979
`avail()` (*in module salt.modules.smartos_imgadm*), 2500
`avail_images()` (*in module salt.cloud.clouds.aliyun*), 1032
`avail_images()` (*in module salt.cloud.clouds.azurearm*), 1036
`avail_images()` (*in module salt.cloud.clouds.cloudstack*), 1038
`avail_images()` (*in module salt.cloud.clouds.digitalocean*), 1040
`avail_images()` (*in module salt.cloud.clouds.dimensiondata*), 1043
`avail_images()` (*in module salt.cloud.clouds.ec2*), 1046
`avail_images()` (*in module salt.cloud.clouds.gce*), 1054
`avail_images()` (*in module salt.cloud.clouds.gogrid*), 1059
`avail_images()` (*in module salt.cloud.clouds.joyent*), 1062
`avail_images()` (*in module salt.cloud.clouds.linode*), 1065
`avail_images()` (*in module salt.cloud.clouds.lxc*), 1070
`avail_images()` (*in module salt.cloud.clouds.msazure*), 1072
`avail_images()` (*in module salt.cloud.clouds.oneandone*), 1088
`avail_images()` (*in module salt.cloud.clouds.opennebula*), 1090
`avail_images()` (*in module salt.cloud.clouds.openstack*), 1112
`avail_images()` (*in module salt.cloud.clouds.parallels*), 1114
`avail_images()` (*in module salt.cloud.clouds.profitbricks*), 1116
`avail_images()` (*in module salt.cloud.clouds.proxmox*), 1119
`avail_images()` (*in module salt.cloud.clouds.qingcloud*), 1122
`avail_images()` (*in module salt.cloud.clouds.saltify*), 1124
`avail_images()` (*in module salt.cloud.clouds.scaleway*), 1126
`avail_images()` (*in module salt.cloud.clouds.softlayer*), 1127
`avail_images()` (*in module salt.cloud.clouds.softlayer_hw*), 1129
`avail_images()` (*in module salt.cloud.clouds.tencentcloud*), 1130
`avail_images()` (*in module salt.cloud.clouds.vagrant*), 1133
`avail_images()` (*in module salt.cloud.clouds.vmware*), 1138
`avail_images()` (*in module salt.cloud.clouds.vultrpy*), 1148
`avail_images()` (*in module salt.cloud.clouds.xen*), 1149
`avail_locations()` (*in module salt.cloud.clouds.aliyun*), 1032
`avail_locations()` (*in module salt.cloud.clouds.azurearm*), 1036
`avail_locations()` (*in module salt.cloud.clouds.cloudstack*), 1038
`avail_locations()` (*in module salt.cloud.clouds.digitalocean*), 1040
`avail_locations()` (*in module salt.cloud.clouds.dimensiondata*), 1043
`avail_locations()` (*in module salt.cloud.clouds.ec2*), 1046
`avail_locations()` (*in module salt.cloud.clouds.gce*), 1054
`avail_locations()` (*in module salt.cloud.clouds.gogrid*), 1059
`avail_locations()` (*in module salt.cloud.clouds.joyent*), 1062
`avail_locations()` (*in module salt.cloud.clouds.linode*), 1065
`avail_locations()` (*in module salt.cloud.clouds.msazure*), 1072
`avail_locations()` (*in module salt.cloud.clouds.oneandone*), 1088
`avail_locations()` (*in module salt.cloud.clouds.opennebula*), 1091
`avail_locations()` (*in module salt.cloud.clouds.profitbricks*), 1116
`avail_locations()` (*in module salt.cloud.clouds.proxmox*), 1119

`avail_locations()` (*in module salt.cloud.clouds.qingcloud*), 1122
`avail_locations()` (*in module salt.cloud.clouds.saltify*), 1124
`avail_locations()` (*in module salt.cloud.clouds.softlayer*), 1127
`avail_locations()` (*in module salt.cloud.clouds.softlayer_hw*), 1129
`avail_locations()` (*in module salt.cloud.clouds.tencentcloud*), 1130
`avail_locations()` (*in module salt.cloud.clouds.vagrant*), 1133
`avail_locations()` (*in module salt.cloud.clouds.vmware*), 1138
`avail_locations()` (*in module salt.cloud.clouds.vultrpy*), 1148
`avail_locations()` (*in module salt.cloud.clouds.xen*), 1150
`avail_modules()` (*in module salt.modules.win_psget*), 2789
`avail_platforms()` (*in module salt.modules.genesis*), 1752
`avail_scripts()` (*in module salt.cloud.clouds.vultrpy*), 1148
`avail_sizes()` (*in module salt.cloud.clouds.aliyun*), 1032
`avail_sizes()` (*in module salt.cloud.clouds.azurearm*), 1036
`avail_sizes()` (*in module salt.cloud.clouds.cloudstack*), 1038
`avail_sizes()` (*in module salt.cloud.clouds.digitalocean*), 1040
`avail_sizes()` (*in module salt.cloud.clouds.dimensiondata*), 1043
`avail_sizes()` (*in module salt.cloud.clouds.ec2*), 1046
`avail_sizes()` (*in module salt.cloud.clouds.gce*), 1054
`avail_sizes()` (*in module salt.cloud.clouds.gogrid*), 1059
`avail_sizes()` (*in module salt.cloud.clouds.joyent*), 1062
`avail_sizes()` (*in module salt.cloud.clouds.linode*), 1065
`avail_sizes()` (*in module salt.cloud.clouds.msazure*), 1072
`avail_sizes()` (*in module salt.cloud.clouds.oneandone*), 1088
`avail_sizes()` (*in module salt.cloud.clouds.opennebula*), 1091
`avail_sizes()` (*in module salt.cloud.clouds.openstack*), 1112
`avail_sizes()` (*in module salt.cloud.clouds.profitbricks*), 1116
`avail_sizes()` (*in module salt.cloud.clouds.qingcloud*), 1122
`avail_sizes()` (*in module salt.cloud.clouds.saltify*), 1124
`avail_sizes()` (*in module salt.cloud.clouds.softlayer*), 1127
`avail_sizes()` (*in module salt.cloud.clouds.softlayer_hw*), 1129
`avail_sizes()` (*in module salt.cloud.clouds.tencentcloud*), 1130
`avail_sizes()` (*in module salt.cloud.clouds.vagrant*), 1133
`avail_sizes()` (*in module salt.cloud.clouds.vmware*), 1138
`avail_sizes()` (*in module salt.cloud.clouds.vultrpy*), 1148
`avail_sizes()` (*in module salt.cloud.clouds.xen*), 1150
`availability_set_absent()` (*in module salt.states.azurearm_compute*), 3201
`availability_set_create_or_update()` (*in module salt.modules.azurearm_compute*), 1249
`availability_set_delete()` (*in module salt.modules.azurearm_compute*), 1249
`availability_set_get()` (*in module salt.modules.azurearm_compute*), 1250
`availability_set_present()` (*in module salt.states.azurearm_compute*), 3201
`availability_sets_list()` (*in module salt.modules.azurearm_compute*), 1250
`availability_sets_list_available_sizes()` (*in module salt.modules.azurearm_compute*), 1250
`available()` (*in module salt.modules.daemontools*), 1585
`available()` (*in module salt.modules.debian_service*), 1597
`available()` (*in module salt.modules.freebsdkmod*), 1740
`available()` (*in module salt.modules.freebsdservice*), 1746
`available()` (*in module salt.modules.gentoo_service*), 1754
`available()` (*in module salt.modules.kmod*), 1945
`available()` (*in module salt.modules.launchctl_service*), 1951
`available()` (*in module salt.modules.mac_service*), 2038
`available()` (*in module salt.modules.netbsdservice*), 2203
`available()` (*in module salt.modules.openbsdrcctl_service*), 2272
`available()` (*in module salt.modules.openbsdservice*), 2274
`available()` (*in module salt.modules.rh_service*), 2443
`available()` (*in module salt.modules.runit*), 2453
`available()` (*in module salt.modules.s6*), 2461

available() (*in module salt.modules.smf_service*), 2510
available() (*in module salt.modules.systemd_service*), 2592
available() (*in module salt.modules.upstart_service*), 2627
available() (*in module salt.modules.win_service*), 2794
available() (*in module salt.modules.win_wua*), 2839
available_capabilities() (*in module salt.modules.win_dism*), 2725
available_extensions() (*in module salt.modules.postgres*), 2374
available_features() (*in module salt.modules.win_dism*), 2725
available_version() (*in module salt.modules.aixpkg*), 1204
available_version() (*in module salt.modules.mac_brew_pkg*), 2023
available_version() (*in module salt.modules.mac_portspkg*), 2032
available_version() (*in module salt.modules.opkg*), 2280
available_version() (*in module salt.modules.pkgin*), 2351
available_version() (*in module salt.modules.solarisipspkg*), 2521
available_version() (*in module salt.modules.xbpspkg*), 2857
AvailablePackages (*class in salt.modules.yumpkg*), 2863
AvailablePagedPoolPages (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATION attribute*), 2806
AvailablePages (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATION attribute*), 2806
avg
    jinja filters, 432
aws_encode() (*in module salt.modules.boto3_route53*), 1324
azurefs_update_interval
    conf/master, 98

B

back_make() (*in module salt.modules.bcache*), 1281
backup() (*in module salt.modules.pkgng*), 2354
backup() (*in module salt.modules.solr*), 2529
backup_mode
    conf/minion, 136
backup_mode() (*in module salt.modules.config*), 1556
balancer_absent() (*in module salt.states.libcloud_loadbalancer*), 3536
balancer_attach_member() (*in module salt.modules.libcloud_loadbalancer*), 1968
balancer_detach_member() (*in module salt.modules.libcloud_loadbalancer*), 1969
balancer_present() (*in module salt.states.libcloud_loadbalancer*), 3536
ban() (*in module salt.modules.varnish*), 2638
ban_list() (*in module salt.modules.varnish*), 2638
bare_rac_cmd() (*in module salt.modules.dracr*), 1665
base64_b64decode() (*in module salt.modules.hashutil*), 1842
base64_b64encode() (*in module salt.modules.hashutil*), 1843
base64_decode
    jinja filters, 441
base64_decodefile() (*in module salt.modules.hashutil*), 1843
base64_decodestring() (*in module salt.modules.hashutil*), 1843
base64_encode
    jinja filters, 440
base64_encodefile() (*in module salt.modules.hashutil*), 1843
base64_encodestring() (*in module salt.modules.hashutil*), 1843
baseline_snapshot() (*in module salt.states.snapper*), 3719
basename() (*in module salt.modules.file*), 1704
batchEvent() (*salt.returners.splunk.http_event_collector method*), 352
beacon() (*in module salt.beacons.adb*), 1006
beacon() (*in module salt.beacons.avahi_announce*), 1007
beacon() (*in module salt.beacons.bonjour_announce*), 1008
beacon() (*in module salt.beacons.btmp*), 1010
beacon() (*in module salt.beacons.diskusage*), 1010
beacon() (*in module salt.beacons.glxinfo*), 1011
beacon() (*in module salt.beacons.haproxy*), 1011
beacon() (*in module salt.beacons.inotify*), 1012
beacon() (*in module salt.beacons.journald*), 1013
beacon() (*in module salt.beacons.load*), 1013
beacon() (*in module salt.beacons.memusage*), 1014
beacon() (*in module salt.beacons.napalm_beacon*), 1017
beacon() (*in module salt.beacons.network_info*), 1017
beacon() (*in module salt.beacons.network_settings*), 1018
beacon() (*in module salt.beacons.pkg*), 1018
beacon() (*in module salt.beacons.proxy_example*), 1019
beacon() (*in module salt.beacons.ps*), 1019
beacon() (*in module salt.beacons.salt_proxy*), 1019
beacon() (*in module salt.beacons.sensehat*), 1020
beacon() (*in module salt.beacons.service*), 1020
beacon() (*in module salt.beacons.sh*), 1021
beacon() (*in module salt.beacons.status*), 1023

beacon() (*in module salt.beacons.telegram_bot_msg*), 1023
beacon() (*in module salt.beacons.twilio_txt_msg*), 1023
beacon() (*in module salt.beacons.wtmp*), 1025
Best Practices, 576
bgrewriteaof() (*in module salt.modules.redismod*), 2427
bgsave() (*in module salt.modules.redismod*), 2427
bin_pkg_info() (*in module salt.modules.dpkg_lowpkg*), 1661
bin_pkg_info() (*in module salt.modules.rpm_lowpkg*), 2447
bindings_exist() (*in module salt.states.jboss7*), 3507
blade_idrac() (*in module salt.states.dellchassis*), 3367
blkid() (*in module salt.modules.disk*), 1602
block() (*in module salt.modules.bluez_bluetooth*), 1300
block_device_mappings() (*in module salt.cloud.clouds.cloudstack*), 1038
block_device_mappings() (*in module salt.cloud.clouds.ec2*), 1046
block_devices() (*in module salt.modules.osquery*), 2289
block_icmp() (*in module salt.modules.firewalld*), 1732
blockreplace() (*in module salt.modules.file*), 1704
blockreplace() (*in module salt.modules.napalm_network*), 2156
blockreplace() (*in module salt.states.file*), 3422
blocks() (*in module salt.modules.extfs*), 1701
bookmark() (*in module salt.modules.zfs*), 2898
bookmark_absent() (*in module salt.states.zfs*), 3807
bookmark_present() (*in module salt.states.zfs*), 3807
BOOL_PROPS_LIST (*in module salt.modules.solrcloud*), 2533
boolean() (*in module salt.states.selinux*), 3707
boot() (*in module salt.cloud.clouds.linode*), 1065
boot() (*in module salt.modules.nova*), 2253
boot() (*in module salt.modules.zoneadm*), 2908
boot_device() (*in module salt.states.ipmi*), 3498
boot_time() (*in module salt.modules.ps*), 2389
booted() (*in module salt.states.zone*), 3811
Bootstrap, **5401**
bootstrap() (*in module salt.modules.chocolatey*), 1500
bootstrap() (*in module salt.modules.genesis*), 1752
bootstrap() (*in module salt.modules.lxc*), 1987
bootstrap() (*in module salt.modules.win_psget*), 2790
bootstrap() (*in module salt.modules.zcbuildout*), 2896
bootstrap() (*in module salt.runners.manage*), 3121
bootstrap() (*in module salt.states.bower*), 3335
bootstrap() (*in module salt.states.npm*), 3611
bootstrap_container() (*in module salt.modules.nspawn*), 2259
bootstrap_psexec() (*in module salt.runners.manage*), 3122
bootstrap_salt() (*in module salt.modules.nspawn*), 2260
bounce_cluster() (*in module salt.modules.trafficserver*), 2622
bounce_cluster() (*in module salt.states.trafficserver*), 3739
bounce_local() (*in module salt.modules.trafficserver*), 2622
bounce_local() (*in module salt.states.trafficserver*), 3739
branch() (*in module salt.modules.git*), 1760
bridge_create() (*in module salt.modules.openvswitch*), 2278
bridge_delete() (*in module salt.modules.openvswitch*), 2278
bridge_exists() (*in module salt.modules.openvswitch*), 2278
bridge_list() (*in module salt.modules.openvswitch*), 2278
broadcast_change() (*in module salt.modules.reg*), 2434
build
    spm command line option, 991
build() (*in module salt.modules.debuild_pkgbuild*), 1593
build() (*in module salt.modules.dockercompose*), 1612
build() (*in module salt.modules.dockermod*), 1618
build() (*in module salt.modules.inspector*), 1878
build() (*in module salt.modules.rpmbuild_pkgbuild*), 2449
build() (*salt.modules.inspectlib.collector.Inspector method*), 1874
build() (*salt.modules.syslog_ng.Argument method*), 2577
build() (*salt.modules.syslog_ng.Buildable method*), 2578
build() (*salt.modules.syslog_ng.GivenStatement method*), 2578
build() (*salt.modules.syslog_ng.Option method*), 2578
build() (*salt.modules.syslog_ng.SimpleParameter method*), 2579
build() (*salt.modules.syslog_ng.SimpleParameterValue method*), 2579
build() (*salt.modules.syslog_ng.TypedParameter method*), 2579
build() (*salt.modules.syslog_ng.TypedParameterValue method*), 2580
build_body() (*salt.modules.syslog_ng.Buildable method*), 2578
build_bond() (*in module salt.modules.debian_ip*), 1596
build_bond() (*in module salt.modules.rh_ip*), 2442

`build_clonespec()` (*in module salt.cloud.clouds.vmware*), 1138
`build_directions()` (*in module salt.modules.csf*), 1582
`build_header()` (*salt.modules.syslog_ng.Buildable method*), 2578
`build_header()` (*salt.modules.syslog_ng.Statement method*), 2579
`build_info()` (*in module salt.modules.nginx*), 2248
`build_interface()` (*in module salt.modules.debian_ip*), 1596
`build_interface()` (*in module salt.modules.nilrt_ip*), 2248
`build_interface()` (*in module salt.modules.rh_ip*), 2442
`build_job()` (*in module salt.modules.jenkinsmod*), 1908
`build_network_settings()` (*in module salt.modules.debian_ip*), 1596
`build_network_settings()` (*in module salt.modules.nilrt_ip*), 2249
`build_network_settings()` (*in module salt.modules.rh_ip*), 2442
`build_policy()` (*in module salt.modules.boto_iam*), 1394
`build_routes()` (*in module salt.modules.debian_ip*), 1596
`build_routes()` (*in module salt.modules.rh_ip*), 2442
`build_rule()` (*in module salt.modules.iptables*), 1898
`build_rule()` (*in module salt.modules.nftables*), 2243
`build_schedule_item()` (*in module salt.modules.schedule*), 2481
`build_tail()` (*salt.modules.syslog_ng.Buildable method*), 2578
`build_tail()` (*salt.modules.syslog_ng.Statement method*), 2579
`Buildable` (*class in salt.modules.syslog_ng*), 2578
`buildmod()` (*in module salt.modules.znc*), 2907
`buildout()` (*in module salt.modules.zcbuildout*), 2896
`built()` (*in module salt.states.pkgbuild*), 3658
`bulk_activate()` (*in module salt.modules.modjk*), 2083
`bulk_build()` (*in module salt.modules.poudriere*), 2385
`bulk_disable()` (*in module salt.modules.modjk*), 2084
`bulk_recover()` (*in module salt.modules.modjk*), 2084
`bulk_stop()` (*in module salt.modules.modjk*), 2084

C

`ca_exists()` (*in module salt.modules.tls*), 2611
`cache`
    `conf/master`, 58
`cache_clean()` (*in module salt.modules.npm*), 2257
`cache_cleaned()` (*in module salt.states.npm*), 3611
`cache_cluster_absent()` (*in module salt.states.boto3_elasticache*), 3235
`cache_cluster_absent()` (*in module salt.states.boto_elasticache*), 3281
`cache_cluster_exists()` (*in module salt.modules.boto3_elasticache*), 1302
`cache_cluster_present()` (*in module salt.states.boto3_elasticache*), 3235
`cache_cluster_present()` (*in module salt.states.boto_elasticache*), 3281
`cache_dir()` (*in module salt.modules.cp*), 1574
`cache_file()` (*in module salt.modules.container_resource*), 1573
`cache_file()` (*in module salt.modules.cp*), 1574
`cache_files()` (*in module salt.modules.cp*), 1574
`cache_jobs`
    `conf/minion`, 132
`cache_list()` (*in module salt.modules.npm*), 2258
`cache_local_file()` (*in module salt.modules.cp*), 1575
`cache_make()` (*in module salt.modules.bcache*), 1281
`cache_master()` (*in module salt.modules.cp*), 1575
`cache_path()` (*in module salt.modules.npm*), 2258
`cache_security_group_exists()` (*in module salt.modules.boto3_elasticache*), 1302
`cache_sreqs`
    `conf/minion`, 140
`cache_subnet_group_absent()` (*in module salt.states.boto3_elasticache*), 3238
`cache_subnet_group_exists()` (*in module salt.modules.boto3_elasticache*), 1303
`cache_subnet_group_present()` (*in module salt.states.boto3_elasticache*), 3238
`cached()` (*in module salt.states.file*), 3424
`cachedir`
    `conf/master`, 55
    `conf/minion`, 132
`CacheIoCount` (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFO attribute*), 2806
`CacheReadCount` (*salt.modules.win_status.SYSTEM_PERFORMANCE_IN attribute*), 2806
`CacheTransitionCount` (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATIC attribute*), 2806
`calc()` (*in module salt.thorium.calc*), 3819
`calc_net()` (*in module salt.modules.network*), 2215
`call()` (*in module salt.cloud.clouds.openstack*), 1112
`call()` (*in module salt.modules.arista_pyeapi*), 1235
`call()` (*in module salt.modules.dockermod*), 1619
`call()` (*in module salt.modules.napalm_mod*), 2125
`call()` (*in module salt.modules.netmiko_mod*), 2208
`call()` (*in module salt.proxy.arista_pyeapi*), 3050

`call()` (*in module salt.proxy.napalm*), 3068
`call()` (*in module salt.proxy.netmiko_px*), 3073
`call()` (*in module salt.proxy.panos*), 3081
`call()` (*in module salt.states.cmd*), 3347
`call()` (*salt.modules.ansiblegate.AnsibleModuleCaller method*), 1208
`call_alert()` (*in module salt.proxy.philips_hue*), 3082
`call_blink()` (*in module salt.proxy.philips_hue*), 3082
`call_brightness()` (*in module salt.proxy.philips_hue*), 3083
`call_color()` (*in module salt.proxy.philips_hue*), 3083
`call_effect()` (*in module salt.proxy.philips_hue*), 3083
`call_hook()` (*in module salt.modules.slack_notify*), 2496
`call_lights()` (*in module salt.proxy.philips_hue*), 3083
`call_ping()` (*in module salt.proxy.philips_hue*), 3084
`call_rename()` (*in module salt.proxy.philips_hue*), 3084
`call_status()` (*in module salt.proxy.philips_hue*), 3084
`call_switch()` (*in module salt.proxy.philips_hue*), 3084
`call_temperature()` (*in module salt.proxy.philips_hue*), 3084
`Caller` (*class in salt.client*), 3849
`calling-salt-functions`
  `jinja filters`, 451
`can_user_run()` (*salt.engines.slack.SlackClient method*), 1166
`cancel_commit()` (*in module salt.modules.napalm_network*), 2157
`cancel_downtime()` (*in module salt.modules.datadog_api*), 1588
`cancel_elasticsearch_service_software_update()` (*in module salt.modules.boto3_elasticsearch*), 1309
`capabilities()` (*in module salt.modules.virt*), 2646
`capability_installed()` (*in module salt.states.win_dism*), 3758
`capability_removed()` (*in module salt.states.win_dism*), 3759
`cas()` (*in module salt.modules.data*), 1586
`catalog_datacenters()` (*in module salt.modules.consul*), 1565
`catalog_deregister()` (*in module salt.modules.consul*), 1566
`catalog_node()` (*in module salt.modules.consul*), 1566
`catalog_nodes()` (*in module salt.modules.consul*), 1566
`catalog_register()` (*in module salt.modules.consul*), 1567
`catalog_service()` (*in module salt.modules.consul*), 1567
`catalog_services()` (*in module salt.modules.consul*), 1567
`CcCopyReadNoWait` (*salt.modules.win_status.SYSTEM_PERFORMANCE_ attribute*), 2806
`CcCopyReadNoWaitMiss` (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATIC attribute*), 2806
`CcCopyReadWait` (*salt.modules.win_status.SYSTEM_PERFORMANCE_IN attribute*), 2806
`CcCopyReadWaitMiss` (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATIC attribute*), 2807
`CcDataFlushes` (*salt.modules.win_status.SYSTEM_PERFORMANCE_INF attribute*), 2807
`CcDataPages` (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFOR attribute*), 2807
`CcDirtyPagesThreshold` (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATIC attribute*), 2807
`CcFastMdlReadNotPossible` (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATIC attribute*), 2807
`CcFastMdlReadNoWait` (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATIC attribute*), 2807
`CcFastMdlReadResourceMiss` (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATIC attribute*), 2807
`CcFastMdlReadWait` (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATIC attribute*), 2807
`CcFastReadNotPossible` (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATIC attribute*), 2807
`CcFastReadNoWait` (*salt.modules.win_status.SYSTEM_PERFORMANCE_ attribute*), 2807
`CcFastReadResourceMiss` (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATIC attribute*), 2807
`CcFastReadWait` (*salt.modules.win_status.SYSTEM_PERFORMANCE_IN attribute*), 2807
`CcLazyWriteIos` (*salt.modules.win_status.SYSTEM_PERFORMANCE_IN attribute*), 2807
`CcLazyWritePages` (*salt.modules.win_status.SYSTEM_PERFORMANCE_ attribute*), 2807
`CcMapDataNoWait` (*salt.modules.win_status.SYSTEM_PERFORMANCE_I attribute*), 2807
`CcMapDataNoWaitMiss` (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATIC attribute*), 2807
`CcMapDataWait` (*salt.modules.win_status.SYSTEM_PERFORMANCE_INF attribute*), 2807

CcMapDataWaitMiss (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATION attribute*), 2807

CcMdlReadNoWait (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATION attribute*), 2807

CcMdlReadNoWaitMiss (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATION attribute*), 2807

CcMdlReadWait (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATION attribute*), 2807

CcMdlReadWaitMiss (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATION attribute*), 2808

CcPinMappedDataCount (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATION attribute*), 2808

CcPinReadNoWait (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATION attribute*), 2808

CcPinReadNoWaitMiss (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATION attribute*), 2808

CcPinReadWait (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATION attribute*), 2808

CcPinReadWaitMiss (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATION attribute*), 2808

CcReadAheadIos (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATION attribute*), 2808

CcTotalDirtyPages (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATION attribute*), 2808

ceph_version() (*in module salt.modules.ceph*), 1491

cert() (*in module salt.modules.acme*), 1200

cert() (*in module salt.states.acme*), 3182

cert_base_path() (*in module salt.modules.tls*), 2611

cert_info() (*in module salt.modules.tls*), 2612

certificate_managed() (*in module salt.states.x509*), 3790

certificates() (*in module salt.modules.osquery*), 2289

certs() (*in module salt.modules.acme*), 1201

cflags_contains() (*in module salt.modules.makeconf*), 2062

ch_config() (*in module salt.proxy.esxi*), 3061

chain_absent() (*in module salt.states.iptables*), 3506

chain_absent() (*in module salt.states.nftables*), 3610

chain_present() (*in module salt.states.iptables*), 3506

chain_present() (*in module salt.states.nftables*), 3611

change() (*in module salt.modules.ldap3*), 1954

change() (*in module salt.states.augeas*), 3197

change_password() (*in module salt.modules.drac*), 1662

change_password() (*in module salt.modules.dracr*), 1665

change_password() (*in module salt.modules.ilo*), 1852

change_password() (*in module salt.modules.rabbitmq*), 2413

change_resource_record_sets() (*in module salt.modules.boto3_route53*), 1324

change_username() (*in module salt.modules.ilo*), 1852

changed() (*salt.states.cyg.DictDiffer method*), 3360

changed_files() (*in module salt.modules.snapper*), 2544

channel (*salt.engines.ircbot.PrivEvent attribute*), 1156

chassis() (*in module salt.states.dellchassis*), 3367

chassis_credentials() (*in module salt.modules.chassis*), 1498

chattr() (*in module salt.modules.file*), 1705

chconfig() (*in module salt.proxy.fx2*), 3064

check() (*in module salt.modules.ipset*), 1896

check() (*in module salt.modules.iptables*), 1899

check() (*in module salt.modules.namecheap_domains*), 2115

check() (*in module salt.modules.nftables*), 2244

check() (*in module salt.modules.parted_partition*), 2325

check() (*in module salt.modules.pkgng*), 2355

check() (*in module salt.states.firewall*), 3451

check_ace() (*in module salt.modules.win_dacl*), 2721

check_antivirus() (*in module salt.modules.panos*), 2306

check_available() (*in module salt.modules.freebsdkmod*), 1740

check_available() (*in module salt.modules.kmod*), 1945

check_balances() (*in module salt.modules.namecheap_users*), 2124

check_chain() (*in module salt.modules.iptables*), 1899

check_chain() (*in module salt.modules.nftables*), 2244

check_conf_enabled() (*in module salt.modules.deb_apache*), 1591

check_db() (*in module salt.modules.ebuildpkg*), 1675

check_dns_name_availability() (*in module salt.modules.azurearm_network*), 1254

check_exists() (*in module salt.modules.alternatives*), 1207

check_exists() (*in module salt.modules.uptime*), 2629

check_extra_requirements() (*in module salt.modules.ebuildpkg*), 1675

check_extra_requirements() (*in module salt.modules.opkg*), 2281

check_extra_requirements() (*in module*

salt.modules.pkg_resource), 2349
check_file_meta() (*in module salt.modules.file*), 1705
check_hash() (*in module salt.modules.file*), 1706
check_inheritance() (*in module salt.modules.win_dacl*), 2722
check_installed() (*in module salt.modules.alternatives*), 1207
check_ip() (*in module salt.modules.dig*), 1602
check_ip() (*in module salt.modules.dnsutil*), 1608
check_ip_address_availability() (*in module salt.modules.azurearm_network*), 1254
check_key() (*in module salt.modules.ssh*), 2539
check_key_file() (*in module salt.modules.ssh*), 2539
check_known_host() (*in module salt.modules.ssh*), 2539
check_managed() (*in module salt.modules.file*), 1706
check_managed_changes() (*in module salt.modules.file*), 1707
check_member_pool() (*in module salt.runners.f5*), 3110
check_member_pool() (*salt.runners.f5.F5Mgmt method*), 3110
check_mod_enabled() (*in module salt.modules.deb_apache*), 1592
check_mod_enabled() (*in module salt.modules.suse_apache*), 2568
check_password() (*in module salt.modules.rabbitmq*), 2413
check_password() (*in module salt.proxy.nxos*), 3074
check_perms() (*in module salt.modules.file*), 1707
check_perms() (*in module salt.modules.win_file*), 2733
check_pillar() (*in module salt.states.test*), 3734
check_pool() (*in module salt.runners.f5*), 3110
check_pool() (*salt.runners.f5.F5Mgmt method*), 3110
check_request() (*in module salt.modules.state*), 2549
check_role() (*in module salt.proxy.nxos*), 3074
check_server() (*in module salt.modules.lvs*), 1984
check_service() (*in module salt.modules.lvs*), 1984
check_set() (*in module salt.modules.ipset*), 1896
check_site_enabled() (*in module salt.modules.deb_apache*), 1592
check_software() (*in module salt.modules.panos*), 2306
check_table() (*in module salt.modules.nftables*), 2244
check_upgrade_eligibility() (*in module salt.modules.boto3_elasticsearch*), 1309
check_valid_package() (*in module salt.modules.cyg*), 1583
check_virtualserver() (*in module salt.runners.f5*), 3110
check_virtualserver() (*salt.runners.f5.F5Mgmt method*), 3110
check_vpc() (*in module salt.modules.boto_vpc*), 1450
check_whitelist_blacklist
    jinja filters, 435
CheckError, 3998
checkout() (*in module salt.modules.git*), 1761
checkout() (*in module salt.modules.svn*), 2569
checks_list() (*in module salt.modules.uptime*), 2629
checksum() (*in module salt.modules.rpm_lowpkg*), 2447
chfullname() (*in module salt.modules.mac_user*), 2057
chfullname() (*in module salt.modules.pw_user*), 2407
chfullname() (*in module salt.modules.solaris_user*), 2518
chfullname() (*in module salt.modules.useradd*), 2631
chfullname() (*in module salt.modules.win_useradd*), 2834
chgid() (*in module salt.modules.aix_group*), 1203
chgid() (*in module salt.modules.groupadd*), 1837
chgid() (*in module salt.modules.mac_group*), 2027
chgid() (*in module salt.modules.mac_user*), 2057
chgid() (*in module salt.modules.pw_group*), 2406
chgid() (*in module salt.modules.pw_user*), 2407
chgid() (*in module salt.modules.solaris_group*), 2515
chgid() (*in module salt.modules.solaris_user*), 2518
chgid() (*in module salt.modules.useradd*), 2631
chgroups() (*in module salt.modules.mac_user*), 2057
chgroups() (*in module salt.modules.pw_user*), 2407
chgroups() (*in module salt.modules.solaris_user*), 2518
chgroups() (*in module salt.modules.useradd*), 2631
chgroups() (*in module salt.modules.win_useradd*), 2834
chgrp() (*in module salt.modules.file*), 1707
chgrp() (*in module salt.modules.win_file*), 2734
chhome() (*in module salt.modules.mac_user*), 2057
chhome() (*in module salt.modules.pw_user*), 2407
chhome() (*in module salt.modules.solaris_user*), 2519
chhome() (*in module salt.modules.useradd*), 2631
chhome() (*in module salt.modules.win_useradd*), 2834
chhomephone() (*in module salt.modules.pw_user*), 2407
chhomephone() (*in module salt.modules.solaris_user*), 2519
chhomephone() (*in module salt.modules.useradd*), 2631
chloginclass() (*in module salt.modules.pw_user*), 2407
chloginclass() (*in module salt.modules.useradd*), 2631
chocolatey_version() (*in module salt.modules.chocolatey*), 1500

chost_contains() (*in module salt.modules.makeconf*), 2062
chother() (*in module salt.modules.useradd*), 2632
chown() (*in module salt.modules.file*), 1707
chown() (*in module salt.modules.win_file*), 2734
chpgrp() (*in module salt.modules.win_file*), 2735
chprofile() (*in module salt.modules.win_useradd*), 2835
chrome_extensions() (*in module salt.modules.osquery*), 2289
chroomnumber() (*in module salt.modules.pw_user*), 2407
chroomnumber() (*in module salt.modules.solaris_user*), 2519
chroomnumber() (*in module salt.modules.useradd*), 2632
chshell() (*in module salt.modules.mac_user*), 2057
chshell() (*in module salt.modules.pw_user*), 2408
chshell() (*in module salt.modules.solaris_user*), 2519
chshell() (*in module salt.modules.useradd*), 2632
chuid() (*in module salt.modules.mac_user*), 2057
chuid() (*in module salt.modules.pw_user*), 2408
chuid() (*in module salt.modules.solaris_user*), 2519
chuid() (*in module salt.modules.useradd*), 2632
chworkphone() (*in module salt.modules.pw_user*), 2408
chworkphone() (*in module salt.modules.solaris_user*), 2519
chworkphone() (*in module salt.modules.useradd*), 2632
cib_create() (*in module salt.modules.pcs*), 2329
cib_present() (*in module salt.states.pcs*), 3633
cib_push() (*in module salt.modules.pcs*), 2329
cib_pushed() (*in module salt.states.pcs*), 3633
cidr_broadcast() (*in module salt.modules.netaddress*), 2195
cidr_netmask() (*in module salt.modules.netaddress*), 2196
cimc() (*in module salt.grains.cimc*), 1177
clean() (*in module salt.modules.iosconfig*), 1881
clean() (*in module salt.modules.pkgng*), 2355
clean_dynamic_modules
  conf/minion, 151
clean_locks() (*in module salt.modules.zypperpkg*), 2924
clean_metadata() (*in module salt.modules.yumpkg*), 2863
clean_old_jobs() (*in module salt.returners.etcd_return*), 321
clean_old_jobs() (*in module salt.returners.local_cache*), 325
clean_old_jobs() (*in module salt.returners.multi_returner*), 330
clean_old_jobs() (*in module salt.returners.mysql*), 333
clean_old_jobs() (*in module salt.returners.pgjsonb*), 339
clean_old_jobs() (*in module salt.returners.postgres_local_cache*), 344
clean_old_jobs() (*in module salt.returners.redis_return*), 347
cleanup() (*in module salt.modules.kernelpkg_linux_apt*), 1925
cleanup() (*in module salt.modules.kernelpkg_linux_yum*), 1926
cleanup_unattached_disks() (*in module salt.cloud.clouds.msazure*), 1072
clear() (*in module salt.modules.data*), 1586
clear() (*in module salt.modules.lvs*), 1984
clear() (*in module salt.modules.mac_xattr*), 2059
clear() (*in module salt.modules.qemu_nbd*), 2411
clear() (*in module salt.modules.sensehat*), 2492
clear() (*in module salt.modules.zpool*), 2917
clear() (*in module salt.thorium.reg*), 3826
clear_alarms() (*in module salt.modules.trafficserver*), 2622
clear_all() (*in module salt.runners.cache*), 3099
clear_cache() (*in module salt.modules.saltutil*), 2469
clear_cache() (*in module salt.modules.state*), 2549
clear_cache() (*in module salt.runners.fileserver*), 3111
clear_cluster() (*in module salt.modules.trafficserver*), 2622
clear_cluster() (*in module salt.states.trafficserver*), 3739
clear_commit_tasks() (*in module salt.modules.panos*), 2306
clear_file_list_cache() (*in module salt.runners.fileserver*), 3111
clear_git_lock() (*in module salt.runners.cache*), 3100
clear_grains() (*in module salt.runners.cache*), 3100
clear_job_cache() (*in module salt.modules.saltutil*), 2469
clear_lock() (*in module salt.runners.fileserver*), 3112
clear_mine() (*in module salt.runners.cache*), 3100
clear_mine_func() (*in module salt.runners.cache*), 3100
clear_node() (*in module salt.modules.trafficserver*), 2622
clear_node() (*in module salt.states.trafficserver*), 3739
clear_password() (*in module salt.modules.rabbitmq*), 2413
clear_pillar() (*in module salt.runners.cache*), 3101
clear_property() (*in module salt.modules.zonecfg*), 2910
clear_request() (*in module salt.modules.state*),

2550
clear_triggers() (*in module salt.modules.win_task*), 2825
cli() (*in module salt.modules.junos*), 1913
cli() (*in module salt.modules.napalm_network*), 2157
cli() (*in module salt.states.junos*), 3510
cli_summary
  conf/master, 57
client() (*in module salt.modules.chef*), 1499
client() (*in module salt.states.chef*), 3337
client_config() (*in module salt.config*), 3843
client_version() (*in module salt.modules.oracle*), 2287
clone() (*in module salt.cloud.clouds.linode*), 1066
clone() (*in module salt.modules.git*), 1762
clone() (*in module salt.modules.hg*), 1848
clone() (*in module salt.modules.lxc*), 1987
clone() (*in module salt.modules.parallels*), 2320
clone() (*in module salt.modules.zfs*), 2899
clone() (*in module salt.modules.zoneadm*), 2908
clone_config() (*in module salt.states.panos*), 3624
cloned() (*in module salt.states.git*), 3454
clonemedium() (*in module salt.modules.vboxmanage*), 2643
clonevm() (*in module salt.modules.vboxmanage*), 2643
close() (*in module salt.beacons.inotify*), 1013
close() (*salt.modules.inspectlib.dbhandle.DBHandleBase method*), 1875
close() (*salt.modules.inspectlib.fsdb.CsvDB method*), 1876
close() (*salt.pillar.hg_pillar.Repo method*), 3009
close_alert() (*in module salt.states.opsgenie*), 3616
close_connection() (*salt.engines.logentries.PlainTextSocketAppender method*), 1159
cloud() (*in module salt.runners.cache*), 3101
cloud_init() (*in module salt.modules.lxc*), 1987
cloud_init() (*in module salt.runners.lxc*), 3119
cloud_init_interface() (*in module salt.modules.lxc*), 1988
CloudClient (*class in salt.cloud*), 3852
cloudstack_displayname() (*in module salt.cloud.clouds.cloudstack*), 1038
cluster_commit() (*in module salt.modules.riak*), 2445
cluster_create() (*in module salt.modules.deb_postgres*), 1592
cluster_exists() (*in module salt.modules.deb_postgres*), 1592
cluster_get_settings() (*in module salt.modules.elasticsearch*), 1682
cluster_health() (*in module salt.modules.elasticsearch*), 1683
cluster_join() (*in module salt.modules.riak*), 2445
cluster_leave() (*in module salt.modules.riak*), 2445
cluster_list() (*in module salt.modules.deb_postgres*), 1592
cluster_node_add() (*in module salt.modules.pcs*), 2329
cluster_node_present() (*in module salt.states.pcs*), 3634
cluster_plan() (*in module salt.modules.riak*), 2446
cluster_put_settings() (*in module salt.modules.elasticsearch*), 1683
cluster_quorum() (*in module salt.modules.ceph*), 1491
cluster_remove() (*in module salt.modules.deb_postgres*), 1593
cluster_setup() (*in module salt.modules.pcs*), 2329
cluster_setup() (*in module salt.states.pcs*), 3634
cluster_stats() (*in module salt.modules.elasticsearch*), 1683
cluster_status() (*in module salt.modules.ceph*), 1491
cluster_status() (*in module salt.modules.rabbitmq*), 2413
cluster_status() (*in module salt.modules.solrcloud*), 2534
cmd() (*in module salt.modules.chassis*), 1498
cmd() (*in module salt.modules.esxi*), 1696
cmd() (*in module salt.modules.nxos*), 2265
cmd() (*in module salt.modules.saltutil*), 2470
cmd() (*in module salt.runners.salt*), 3142
cmd() (*in module salt.runners.ssh*), 3152
cmd() (*in module salt.thorium.local*), 3825
cmd() (*in module salt.thorium.runner*), 3827
cmd() (*in module salt.thorium.wheel*), 3828
cmd() (*salt.client.Caller method*), 3849
cmd() (*salt.client.LocalClient method*), 3846
cmd() (*salt.client.ProxyCaller method*), 3850
cmd() (*salt.client.ssh.client.SSHClient method*), 3853
cmd() (*salt.runner.RunnerClient method*), 3850
cmd() (*salt.wheel.WheelClient method*), 3851
cmd_async() (*salt.client.LocalClient method*), 3847
cmd_async() (*salt.runner.RunnerClient method*), 3850
cmd_async() (*salt.wheel.WheelClient method*), 3851
cmd_batch() (*salt.client.LocalClient method*), 3847
cmd_blacklist_glob
  conf/minion, 158
cmd_iter() (*in module salt.modules.saltutil*), 2470
cmd_iter() (*salt.client.LocalClient method*), 3847
cmd_iter() (*salt.client.ssh.client.SSHClient method*), 3853
cmd_iter_no_block() (*salt.client.LocalClient method*), 3848
cmd_subset() (*salt.client.LocalClient method*), 3848
cmd_sync() (*salt.runner.RunnerClient method*), 3851
cmd_sync() (*salt.wheel.WheelClient method*), 3851

`cmd_unzip()` (*in module salt.modules.archive*), 1226

`cmd_whitelist_glob`

    `conf/minion`, 159

`cmd_zip()` (*in module salt.modules.archive*), 1227

`coalesce()` (*in module salt.states.ethtool*), 3408

`code` (*salt.engines.ircbot.Event attribute*), 1156

`code` (*salt.engines.ircbot.PrivEvent attribute*), 1156

`collatz()` (*in module salt.modules.test*), 2601

`collect_garbage()` (*in module salt.modules.nix*), 2251

`collection()` (*in module salt.states.solrcloud*), 3720

`collection_backup()` (*in module salt.modules.solrcloud*), 2534

`collection_backup_all()` (*in module salt.modules.solrcloud*), 2534

`collection_check_options()` (*in module salt.modules.solrcloud*), 2534

`collection_create()` (*in module salt.modules.solrcloud*), 2534

`collection_creation_options()` (*in module salt.modules.solrcloud*), 2535

`collection_exists()` (*in module salt.modules.solrcloud*), 2535

`collection_get_options()` (*in module salt.modules.solrcloud*), 2535

`collection_list()` (*in module salt.modules.solrcloud*), 2535

`collection_reload()` (*in module salt.modules.solrcloud*), 2535

`collection_set_options()` (*in module salt.modules.solrcloud*), 2535

`collectstatic()` (*in module salt.modules.djangomod*), 1605

`color`

    `conf/master`, 56

`COLOR_BLUE` (*salt.proxy.philips_hue.Const attribute*), 3082

`COLOR_DAYLIGHT` (*salt.proxy.philips_hue.Const attribute*), 3082

`COLOR_GREEN` (*salt.proxy.philips_hue.Const attribute*), 3082

`COLOR_ORANGE` (*salt.proxy.philips_hue.Const attribute*), 3082

`COLOR_PINK` (*salt.proxy.philips_hue.Const attribute*), 3082

`COLOR_PURPLE` (*salt.proxy.philips_hue.Const attribute*), 3082

`COLOR_RED` (*salt.proxy.philips_hue.Const attribute*), 3082

`color_theme`

    `conf/master`, 56, 132

`COLOR_WHITE` (*salt.proxy.philips_hue.Const attribute*), 3082

`COLOR_YELLOW` (*salt.proxy.philips_hue.Const attribute*), 3082

`column_families()` (*in module salt.modules.cassandra*), 1481

`column_family_definition()` (*in module salt.modules.cassandra*), 1481

`command` (*salt.engines.ircbot.PrivEvent attribute*), 1156

`command()` (*in module salt.modules.djangomod*), 1605

`CommandExecutionError`, 3998

`commandline_to_list()` (*salt.engines.slack.SlackClient method*), 1166

`CommandNotFoundError`, 3998

`comment()` (*in module salt.modules.file*), 1707

`comment()` (*in module salt.states.file*), 3425

`comment_line()` (*in module salt.modules.file*), 1708

`commit()` (*in module salt.modules.dockermod*), 1619

`commit()` (*in module salt.modules.git*), 1763

`commit()` (*in module salt.modules.junos*), 1914

`commit()` (*in module salt.modules.napalm_network*), 2159

`commit()` (*in module salt.modules.netmiko_mod*), 2208

`commit()` (*in module salt.modules.panos*), 2306

`commit()` (*in module salt.modules.svn*), 2569

`commit()` (*in module salt.states.junos*), 3510

`commit_cancelled()` (*in module salt.states.netconfig*), 3582

`commit_check()` (*in module salt.modules.junos*), 1914

`commit_check()` (*in module salt.states.junos*), 3510

`commit_config()` (*in module salt.states.panos*), 3624

`commit_confirmed()` (*in module salt.states.netconfig*), 3582

`commit_transaction()` (*in module salt.modules.bigip*), 1286

`CommitLimit` (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFOR attribute*), 2808

`CommittedPages` (*salt.modules.win_status.SYSTEM_PERFORMANCE_IN attribute*), 2808

`community_names()` (*in module salt.states.win_snmp*), 3781

`compactionstats()` (*in module salt.modules.cassandra*), 1481

`compare_changes()` (*in module salt.modules.glanceng*), 1810

`compare_changes()` (*in module salt.modules.keystoneng*), 1937

`compare_changes()` (*in module salt.modules.neutronng*), 2239

`compare_config()` (*in module salt.modules.napalm_network*), 2159

`compare_container()` (*in module salt.modules.dockermod*), 1620

`compare_container_networks()` (*in module salt.modules.dockermod*), 1620

`compare_containers()` (*in module salt.modules.dockermod*), 1621

compare_dicts
    jinja filters, 434
compare_lists
    jinja filters, 434
compare_networks() (*in module salt.modules.dockermod*), 1621
compare_params() (*in module salt.modules.zabbix*), 2878
compare_versions() (*in module salt.modules.win_pkg*), 2778
compare_vm_configs() (*in module salt.modules.vsphere*), 2686
compile_config() (*in module salt.modules.win_dsc*), 2729
compliance_report() (*in module salt.modules.napalm_mod*), 2125
compliance_report() (*in module salt.modules.napalm_yang_mod*), 2188
Compound Matcher, **5401**
compound() (*in module salt.modules.match*), 2070
computer_desc() (*in module salt.states.win_system*), 3782
computer_name() (*in module salt.states.win_system*), 3782
conf() (*in module salt.modules.grub_legacy*), 1838
conf/logging
    log_datefmt, 247
    log_datefmt_logfile, 247
    log_file, 246
    log_fmt_console, 247
    log_fmt_jid, 248
    log_fmt_logfile, 247
    log_granular_levels, 248
    log_level, 246
    log_level_logfile, 246
    log_levels, 245
conf/master
    allow_minion_key_revoke, 73
    api_logfile, 111
    api_pidfile, 111
    auth_events, 64
    auto_accept, 68
    autoreject_file, 69
    autosign_file, 69
    autosign_grains_dir, 69
    autosign_timeout, 68
    azurefs_update_interval, 98
    cache, 58
    cachedir, 55
    cli_summary, 57
    color, 56
    color_theme, 56, 132
    conf_file, 53
    cython_enable, 75
    decrypt_pillar, 99
    decrypt_pillar_default, 100
    decrypt_pillar_delimiter, 99
    decrypt_pillar_renderers, 100
    default_include, 117
    eauth_acl_module, 71
    enable_gpu_grains, 57
    enforce_mine_cache, 61
    env_order, 76
    event_return, 59
    event_return_blacklist, 60
    event_return_queue, 59
    event_return_whitelist, 59
    ext_job_cache, 59
    ext_pillar, 101
    ext_pillar_first, 101
    extension_modules, 53
    external_auth, 70
    extmod_blacklist, 53
    extmod_whitelist, 53
    failhard, 79
    file_buffer_size, 83
    file_ignore_glob, 83
    file_ignore_regex, 83
    file_recv, 71
    file_recv_max_size, 72
    file_roots, 84
    fileserver_backend, 81
    fileserver_followsymlinks, 81
    fileserver_ignoresymlinks, 81
    fileserver_limit_traversal, 82
    fileserver_list_cache_time, 82
    fileserver_verify_config, 82
    gather_job_timeout, 55
    git_pillar_base, 102
    git_pillar_branch, 102
    git_pillar_env, 103
    git_pillar_global_lock, 104
    git_pillar_includes, 104
    git_pillar_insecure_auth, 105
    git_pillar_passphrase, 106
    git_pillar_password, 105
    git_pillar_privkey, 106
    git_pillar_provider, 102
    git_pillar_pubkey, 106
    git_pillar_refspecs, 106
    git_pillar_root, 103
    git_pillar_ssl_verify, 104
    git_pillar_user, 105
    git_pillar_verify_config, 106
    gitfs_base, 86
    gitfs_disable_saltenv_mapping, 87
    gitfs_global_lock, 88
    gitfs_insecure_auth, 89

gitfs_mountpoint, 86
gitfs_passphrase, 90
gitfs_password, 89
gitfs_privkey, 90
gitfs_provider, 85
gitfs_pubkey, 89
gitfs_ref_types, 87
gitfs_refspecs, 90
gitfs_remotes, 85
gitfs_root, 86
gitfs_saltenv, 86
gitfs_saltenv_blacklist, 88
gitfs_saltenv_whitelist, 87
gitfs_ssl_verify, 85
gitfs_update_interval, 88
gitfs_user, 89
hash_type, 83
hgfs_base, 92
hgfs_branch_method, 91
hgfs_mountpoint, 92
hgfs_remotes, 91
hgfs_root, 92
hgfs_saltenv_blacklist, 93
hgfs_saltenv_whitelist, 92
hgfs_update_interval, 93
http_connect_timeout, 64
http_request_timeout, 64
include, 117
interface, 51
ipc_mode, 63
ipv6, 51
jinja_env, 77
jinja_lstrip_blocks, 79
jinja_sls_env, 78
jinja_trim_blocks, 79
job_cache, 57
job_cache_store_endtime, 60
keep_acl_in_token, 71
keep_jobs, 55
keysize, 68
log_datefmt, 115
log_datefmt_logfile, 115
log_file, 114
log_fmt_console, 115
log_fmt_logfile, 115
log_granular_levels, 115
log_level, 114
log_level_logfile, 114
log_rotate_backup_count, 116
log_rotate_max_bytes, 116
loop_interval, 55
master_id, 52
master_job_cache, 60
master_pubkey_signature, 72
master_roots, 84
master_sign_key_name, 72
master_sign_pubkey, 72
master_stats, 62
master_stats_event_iter, 62
master_tops, 77
master_use_pubkey_signature, 72
max_event_size, 60
max_minions, 61
max_open_files, 74
memcache_debug, 59
memcache_expire_seconds, 58
memcache_full_cleanup, 58
memcache_max_items, 58
minion_data_cache, 58
minion_data_cache_events, 64
minionfs_blacklist, 97
minionfs_env, 96
minionfs_mountpoint, 97
minionfs_update_interval, 97
minionfs_whitelist, 97
module_dirs, 54
nodegroups, 116
on_demand_ext_pillar, 99
open_mode, 68
optimization_order, 73
order_masters, 111
output, 55
output_file, 56
outputter_dirs, 56
peer, 113
peer_run, 114
permissive_pki_access, 69
pidfile, 52
pillar_cache, 109
pillar_cache_backend, 109
pillar_cache_ttl, 109
pillar_includes_override_sls, 109
pillar_merge_lists, 108
pillar_opts, 100
pillar_raise_on_missing, 102
pillar_roots, 98
pillar_safe_render_error, 100
pillar_source_merging_strategy, 107
ping_on_rotate, 61
pki_dir, 53
presence_events, 61
preserve_minion_cache, 73
pub_hwm, 74
publish_port, 51
publish_session, 73
publisher_acl, 70
publisher_acl_blacklist, 70
range_server, 116

reactor, 110
reactor_refresh_interval, 110
reactor_worker_hwm, 110
reactor_worker_threads, 110
renderer, 77
rest_timeout, 111
ret_port, 52
root_dir, 53
roots_update_interval, 84
roster, 65
roster_defaults, 65
roster_file, 65
rosters, 65
rotate_aes_key, 72
runner_dirs, 75
s3fs_update_interval, 98
show_jid, 56
show_timeout, 56
sock_dir, 57
sock_pool_size, 62
ssh_identities_only, 67
ssh_list_nodegroups, 67
ssh_log_file, 67
ssh_minion_opts, 67
ssh_passwd, 65
ssh_port, 66
ssh_priv_passwd, 66
ssh_scan_ports, 66
ssh_scan_timeout, 66
ssh_sudo, 66
ssh_timeout, 66
ssh_use_home_key, 67
ssh_user, 66
ssl, 73
state_aggregate, 80
state_events, 80
state_output, 80
state_output_diff, 80
state_top, 75
state_top_saltenv, 76
state_verbose, 79
sudo_acl, 70
svnfs_branches, 95
svnfs_mountpoint, 94
svnfs_remotes, 94
svnfs_root, 94
svnfs_saltenv_blacklist, 96
svnfs_saltenv_whitelist, 95
svnfs_tags, 95
svnfs_trunk, 95
svnfs_update_interval, 96
syndic_failover, 113
syndic_forward_all_events, 113
syndic_log_file, 112
syndic_master, 112
syndic_master_port, 112
syndic_pidfile, 112
syndic_wait, 113
tcp_keepalive, 117
tcp_keepalive_cnt, 118
tcp_keepalive_idle, 118
tcp_keepalive_intvl, 118
tcp_master_pub_port, 63
tcp_master_publish_pull, 63
tcp_master_pull_port, 63
tcp_master_workers, 63
thin_extra_mods, 68
timeout, 55
token_expire, 71
token_expire_user_override, 71
top_file_merging_strategy, 76
transport, 62
transport_opts, 62
user, 52
userdata_template, 77
utils_dirs, 75
verify_env, 55
win_gitrepos, 119
win_repo, 118
win_repo_mastercachefile, 119
winrepo_branch, 120
winrepo_cachefile, 119
winrepo_dir, 118
winrepo_dir_ng, 118
winrepo_insecure_auth, 121
winrepo_passphrase, 121
winrepo_password, 120
winrepo_privkey, 121
winrepo_provider, 118
winrepo_pubkey, 121
winrepo_refspecs, 121
winrepo_remotes, 119
winrepo_remotes_ng, 119
winrepo_ssl_verify, 120
winrepo_user, 120
worker_threads, 74
yaml_utf8, 81
zmq_backlog, 75

conf/minion
  acceptance_wait_time, 136
  acceptance_wait_time_max, 137
  always_verify_signature, 158
  append_domain, 131
  append_minionid_config_dirs, 132
  auth_safemode, 138
  auth_timeout, 138
  auth_tries, 138
  autoload_dynamic_modules, 151

autosign_grains, 158
backup_mode, 136
cache_jobs, 132
cache_sreqs, 140
cachedir, 132
clean_dynamic_modules, 151
cmd_blacklist_glob, 158
cmd_whitelist_glob, 159
conf_file, 130
cython_enable, 146
decrypt_pillar, 154
decrypt_pillar_default, 155
decrypt_pillar_delimiter, 155
decrypt_pillar_renderers, 155
default_include, 163
default_top, 149
disable_modules, 144
disable_returners, 144
docker.compare_container_networks, 143
docker.update_mine, 143
enable_gpu_grains, 136
enable_legacy_startup_events, 126
enable_whitelist_modules, 144
enable_zip_modules, 146
env_order, 148
environment, 151
extmod_blacklist, 146
extmod_whitelist, 146
failhard, 163
fibre_channel_grains, 134
file_client, 152
file_recv_max_size, 156
file_roots, 152
fileserver_followsymlinks, 153
fileserver_ignoresymlinks, 153
fileserver_limit_traversal, 153
grains, 133
grains_blacklist, 133
grains_cache, 133
grains_deep_merge, 133
grains_dirs, 145
grains_refresh_every, 134
hash_type, 153
http_connect_timeout, 141
http_request_timeout, 141
id, 130
include, 164
ipc_mode, 140
ipv6, 124
iscsi_grains, 135
keysize, 157
lock_saltenv, 151
log_datefmt, 161
log_datefmt_logfile, 161
log_file, 161
log_fmt_console, 161
log_fmt_logfile, 162
log_granular_levels, 162
log_level, 161
log_level_logfile, 161
log_rotate_backup_count, 162
log_rotate_max_bytes, 162
loop_interval, 139
master, 123
master_alive_interval, 126
master_failback, 126
master_failback_interval, 126
master_finger, 157
master_port, 128
master_shuffle, 126
master_sign_key_name, 158
master_tops_first, 125
master_tries, 137
master_type, 125
master_uri_format, 124
max_event_size, 125
metadata_server_grains, 134
mine_enabled, 135
mine_interval, 136
mine_return_job, 135
minion_id_caching, 131
minion_id_lowercase, 131
minion_id_remove_domain, 131
minion_pillar_cache, 156
module_dirs, 145
modules_max_memory, 146
multiprocessing, 160
no_proxy, 142
nvme_grains, 135
on_demand_ext_pillar, 154
open_mode, 157
optimization_order, 143
outputter_dirs, 136
pass_to_ext_pillars, 156
permissive_pki_access, 157
pillar_raise_on_missing, 156
pillar_roots, 154
pillarenv, 155
pillarenv_from_saltenv, 101, 155
ping_interval, 138
pki_dir, 130
process_count_max, 160
providers, 146
proxy_host, 141
proxy_password, 142
proxy_port, 142
proxy_username, 142

    pub_ret, 139
    publish_port, 128
    random_master, 127
    random_reauth_delay, 137
    random_startup_delay, 139
    reactor, 159
    reactor_refresh_interval, 160
    reactor_worker_hwm, 160
    reactor_worker_threads, 160
    recon_default, 138
    recon_max, 139
    recon_randomize, 139
    rejected_retry, 137
    render_dirs, 145
    renderer, 150
    retry_dns, 127
    retry_dns_count, 127
    return_retry_timer, 140
    return_retry_timer_max, 140
    returner_dirs, 145
    root_dir, 130
    saltenv, 151
    sls_list, 149
    snapper_states, 151
    snapper_states_config, 152
    sock_dir, 136
    source_address, 128
    source_interface_name, 128
    source_publish_port, 129
    source_ret_port, 129
    ssl, 159
    start_event_grains, 149
    startup_states, 149
    state_output, 150
    state_output_diff, 150
    state_top, 147
    state_top_saltenv, 147
    state_verbose, 150
    states_dirs, 145
    sudo_user, 129
    syndic_finger, 141
    tcp_keepalive, 164
    tcp_keepalive_cnt, 164
    tcp_keepalive_idle, 164
    tcp_keepalive_intvl, 165
    tcp_pub_port, 140
    tcp_pull_port, 141
    test, 150
    top_file, 149
    top_file_merging_strategy, 148
    transport, 141
    update_restart_services, 165
    update_url, 165
    use_master_when_local, 152
    user, 129
    utils_dirs, 145
    verify_master_pubkey_sign, 157
    win_gitrepos, 167
    win_repo, 166
    win_repo_cachefile, 166
    winrepo_cache_expire_max, 165
    winrepo_cache_expire_min, 165
    winrepo_cachefile, 166
    winrepo_dir, 166
    winrepo_dir_ng, 166
    winrepo_remotes, 167
    winrepo_remotes_ng, 167
    winrepo_source_dir, 166
    zmq_monitor, 162
conf/proxy
    add_proxymodule_to_opts, 168
    proxy_always_alive, 169
    proxy_keep_alive, 169
    proxy_keep_alive_interval, 169
    proxy_merge_grains_in_module, 168
conf_file
    conf/master, 53
    conf/minion, 130
conf_test() (*in module salt.modules.test*), 2601
config() (*in module salt.grains.extra*), 1179
config() (*in module salt.modules.apache*), 1209
config() (*in module salt.modules.arista_pyeapi*), 1236
config() (*in module salt.modules.freebsdports*), 1745
config() (*in module salt.modules.napalm_bgp*), 2149
config() (*in module salt.modules.napalm_network*), 2160
config() (*in module salt.modules.napalm_probes*), 2179
config() (*in module salt.modules.napalm_snmp*), 2185
config() (*in module salt.modules.napalm_users*), 2187
config() (*in module salt.modules.nxos_api*), 2267
config() (*in module salt.modules.rsync*), 2451
config() (*in module salt.modules.solaris_fmadm*), 2514
config() (*in module salt.modules.syslog_ng*), 2580
config() (*in module salt.states.chronos_job*), 3340
config() (*in module salt.states.marathon_app*), 3551
config() (*in module salt.states.syslog_ng*), 3731
config() (*in module salt.states.trafficserver*), 3739
config_() (*in module salt.modules.bcache*), 1281
config_absent() (*in module salt.states.nxos*), 3613
config_absent() (*in module salt.states.smartos*), 3715
config_changed() (*in module salt.modules.napalm_network*), 2160
config_control() (*in module salt.modules.napalm_network*), 2160
config_diff_text() (*in module salt.modules.napalm_mod*), 2127

config_diff_tree() (*in module salt.modules.napalm_mod*), 2127

config_filter_lines() (*in module salt.modules.napalm_mod*), 2128

config_find_lines() (*in module salt.modules.napalm_mod*), 2128

config_get() (*in module salt.modules.git*), 1764

config_get() (*in module salt.modules.lxd*), 2004

config_get() (*in module salt.modules.redismod*), 2427

config_get_regexp() (*in module salt.modules.git*), 1765

config_lines_w_child() (*in module salt.modules.napalm_mod*), 2128

config_lines_wo_child() (*in module salt.modules.napalm_mod*), 2129

config_merge_diff() (*in module salt.modules.napalm_mod*), 2129

config_merge_text() (*in module salt.modules.napalm_mod*), 2129

config_merge_tree() (*in module salt.modules.napalm_mod*), 2130

config_present() (*in module salt.states.nxos*), 3613

config_present() (*in module salt.states.smartos*), 3715

config_set() (*in module salt.modules.git*), 1766

config_set() (*in module salt.modules.lxd*), 2004

config_set() (*in module salt.modules.redismod*), 2427

config_set() (*in module salt.states.git*), 3455

config_show() (*in module salt.modules.pcs*), 2329

config_test() (*in module salt.modules.syslog_ng*), 2580

config_tree() (*in module salt.modules.napalm_mod*), 2130

config_unset() (*in module salt.modules.git*), 1767

config_unset() (*in module salt.states.git*), 3456

configfile() (*in module salt.states.apache*), 3187

configmap_absent() (*in module salt.states.kubernetes*), 3528

configmap_present() (*in module salt.states.kubernetes*), 3528

configmaps() (*in module salt.modules.kubernetesmod*), 1946

configtest() (*in module salt.modules.monit*), 2089

configtest() (*in module salt.modules.nginx*), 2248

configurable_test_state() (*in module salt.states.test*), 3734

configuration_import() (*in module salt.modules.zabbix*), 2878

configure_host_cache() (*in module salt.modules.vsphere*), 2686

configure_network() (*in module salt.modules.ilo*), 1852

configure_proxy() (*in module salt.modules.salt_proxy*), 2462

configure_proxy() (*in module salt.states.salt_proxy*), 3697

configure_snmp() (*in module salt.modules.ilo*), 1852

configured() (*in module salt.states.net_napalm_yang*), 3602

confirm_commit() (*in module salt.modules.napalm_network*), 2160

conn() (*in module salt.proxy.arista_pyeapi*), 3050

conn() (*in module salt.proxy.junos*), 3065

conn() (*in module salt.proxy.netmiko_px*), 3073

connect() (*in module salt.modules.ldap3*), 1955

connect() (*in module salt.modules.network*), 2216

connect() (*in module salt.modules.qemu_nbd*), 2411

connect() (*in module salt.modules.win_network*), 2772

connect_container_to_network() (*in module salt.modules.dockermod*), 1621

connect_host() (*in module salt.cloud.clouds.vmware*), 1138

connected() (*in module salt.modules.dockermod*), 1622

connected() (*in module salt.modules.napalm_network*), 2161

connected() (*in module salt.wheel.minions*), 3842

connection_ip_list() (*in module salt.states.win_smtp_server*), 3779

Const (*class in salt.proxy.philips_hue*), 3082

constraint_present() (*in module salt.states.pcs*), 3634

consul_fetch() (*in module salt.pillar.consul_pillar*), 2994

container_absent() (*in module salt.states.libcloud_storage*), 3537

container_config_delete() (*in module salt.modules.lxd*), 2005

container_config_get() (*in module salt.modules.lxd*), 2005

container_config_set() (*in module salt.modules.lxd*), 2005

container_create() (*in module salt.modules.lxd*), 2005

container_delete() (*in module salt.modules.lxd*), 2006

container_device_add() (*in module salt.modules.lxd*), 2007

container_device_delete() (*in module salt.modules.lxd*), 2007

container_device_get() (*in module salt.modules.lxd*), 2007

container_execute() (*in module salt.modules.lxd*), 2008

container_file_get() (*in module salt.modules.lxd*), 2008

`container_file_put()` (*in module salt.modules.lxd*), 2008
`container_freeze()` (*in module salt.modules.lxd*), 2009
`container_get()` (*in module salt.modules.lxd*), 2009
`container_list()` (*in module salt.modules.lxd*), 2009
`container_migrate()` (*in module salt.modules.lxd*), 2010
`container_path()` (*in module salt.modules.napalm_formula*), 2153
`container_present()` (*in module salt.states.libcloud_storage*), 3538
`container_rename()` (*in module salt.modules.lxd*), 2010
`container_restart()` (*in module salt.modules.lxd*), 2011
`container_setting()` (*in module salt.states.win_iis*), 3763
`container_start()` (*in module salt.modules.lxd*), 2011
`container_state()` (*in module salt.modules.lxd*), 2011
`container_stop()` (*in module salt.modules.lxd*), 2011
`container_unfreeze()` (*in module salt.modules.lxd*), 2012
`contains()` (*in module salt.cache.consul*), 1026
`contains()` (*in module salt.cache.etcd_cache*), 1027
`contains()` (*in module salt.cache.localfs*), 1027
`contains()` (*in module salt.cache.mysql_cache*), 1028
`contains()` (*in module salt.cache.redis_cache*), 1031
`contains()` (*in module salt.modules.file*), 1708
`contains()` (*in module salt.thorium.check*), 3820
`contains_glob()` (*in module salt.modules.file*), 1708
`contains_regex()` (*in module salt.modules.file*), 1709
`contains_whitespace`
    `jinja filters`, 435
`context()` (*in module salt.states.stateconf*), 3727
`ContextSwitches` (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATION attribute*), 2808
`continuous_query_exists()` (*in module salt.modules.influxdbmod*), 1858
`control_message_target()` (*salt.engines.slack.SlackClient method*), 1166
`conv` (*salt.auth.pam.PamConv attribute*), 1003
`convert()` (*in module salt.modules.btrfs*), 1466
`convert()` (*in module salt.modules.qemu_img*), 2410
`convert_cidr()` (*in module salt.modules.network*), 2216
`convert_duration()` (*in module salt.states.influxdb_retention_policy*), 3492
`convert_to_arn()` (*in module salt.modules.boto_cloudwatch*), 1351
`convert_to_group_ids()` (*in module salt.modules.boto_secgroup*), 1441
`convert_to_template()` (*in module salt.cloud.clouds.vmware*), 1139
`copy()` (*in module salt.modules.file*), 1709
`copy()` (*in module salt.modules.schedule*), 2481
`copy_()` (*in module salt.states.file*), 3425
`copy_from()` (*in module salt.modules.dockermod*), 1622
`copy_image()` (*in module salt.modules.libcloud_compute*), 1960
`copy_snapshot()` (*in module salt.cloud.clouds.ec2*), 1046
`copy_snapshot()` (*in module salt.modules.boto3_elasticache*), 1303
`copy_to()` (*in module salt.modules.container_resource*), 1573
`copy_to()` (*in module salt.modules.dockermod*), 1622
`copy_to()` (*in module salt.modules.lxc*), 1989
`copy_to()` (*in module salt.modules.nspawn*), 2260
`CopyOnWriteCount` (*salt.modules.win_status.SYSTEM_PERFORMANCE_ attribute*), 2808
`core_status()` (*in module salt.modules.solr*), 2529
`coredump_configured()` (*in module salt.states.esxi*), 3410
`coredump_network_enable()` (*in module salt.modules.vsphere*), 2686
`cp()` (*in module salt.modules.parted_partition*), 2325
`cpu()` (*in module salt.modules.sysbench*), 2575
`cpu_baseline()` (*in module salt.modules.virt*), 2647
`cpu_percent()` (*in module salt.modules.ps*), 2389
`cpu_times()` (*in module salt.modules.ps*), 2390
`cpuid()` (*in module salt.modules.osquery*), 2289
`cpuinfo()` (*in module salt.modules.status*), 2558
`cpuload()` (*in module salt.modules.win_status*), 2810
`cpustats()` (*in module salt.modules.status*), 2559
`cpustats()` (*in module salt.modules.win_status*), 2810
`cql_query()` (*in module salt.modules.cassandra_cql*), 1483
`cql_query_with_prepare()` (*in module salt.modules.cassandra_cql*), 1484
`create()` (*in module salt.cloud.clouds.aliyun*), 1032
`create()` (*in module salt.cloud.clouds.azurearm*), 1036
`create()` (*in module salt.cloud.clouds.cloudstack*), 1038
`create()` (*in module salt.cloud.clouds.digitalocean*), 1040
`create()` (*in module salt.cloud.clouds.dimensiondata*), 1043
`create()` (*in module salt.cloud.clouds.ec2*), 1046
`create()` (*in module salt.cloud.clouds.gce*), 1054
`create()` (*in module salt.cloud.clouds.gogrid*), 1060
`create()` (*in module salt.cloud.clouds.joyent*), 1062
`create()` (*in module salt.cloud.clouds.linode*), 1066

`create()` (*in module salt.cloud.clouds.lxc*), 1070
`create()` (*in module salt.cloud.clouds.msazure*), 1072
`create()` (*in module salt.cloud.clouds.oneandone*), 1088
`create()` (*in module salt.cloud.clouds.opennebula*), 1091
`create()` (*in module salt.cloud.clouds.openstack*), 1112
`create()` (*in module salt.cloud.clouds.parallels*), 1114
`create()` (*in module salt.cloud.clouds.profitbricks*), 1117
`create()` (*in module salt.cloud.clouds.proxmox*), 1119
`create()` (*in module salt.cloud.clouds.qingcloud*), 1122
`create()` (*in module salt.cloud.clouds.saltify*), 1124
`create()` (*in module salt.cloud.clouds.scaleway*), 1126
`create()` (*in module salt.cloud.clouds.softlayer*), 1127
`create()` (*in module salt.cloud.clouds.softlayer_hw*), 1129
`create()` (*in module salt.cloud.clouds.tencentcloud*), 1131
`create()` (*in module salt.cloud.clouds.vagrant*), 1133
`create()` (*in module salt.cloud.clouds.virtualbox*), 1134
`create()` (*in module salt.cloud.clouds.vmware*), 1139
`create()` (*in module salt.cloud.clouds.vultrpy*), 1148
`create()` (*in module salt.cloud.clouds.xen*), 1150
`create()` (*in module salt.modules.boto_asg*), 1341
`create()` (*in module salt.modules.boto_cfn*), 1345
`create()` (*in module salt.modules.boto_cloudtrail*), 1349
`create()` (*in module salt.modules.boto_elasticache*), 1379
`create()` (*in module salt.modules.boto_elasticsearch_domain*), 1384
`create()` (*in module salt.modules.boto_elb*), 1386
`create()` (*in module salt.modules.boto_rds*), 1425
`create()` (*in module salt.modules.boto_s3_bucket*), 1435
`create()` (*in module salt.modules.boto_secgroup*), 1442
`create()` (*in module salt.modules.boto_sns*), 1444
`create()` (*in module salt.modules.boto_sqs*), 1446
`create()` (*in module salt.modules.boto_vpc*), 1450
`create()` (*in module salt.modules.cloud*), 1520
`create()` (*in module salt.modules.dockercompose*), 1612
`create()` (*in module salt.modules.dockermod*), 1623
`create()` (*in module salt.modules.lxc*), 1990
`create()` (*in module salt.modules.mdadm_raid*), 2074
`create()` (*in module salt.modules.namecheap_domains*), 2115
`create()` (*in module salt.modules.namecheap_domains_ns*), 2117
`create()` (*in module salt.modules.namecheap_ssl*), 2119
`create()` (*in module salt.modules.pdbedit*), 2331
`create()` (*in module salt.modules.saltcloudmod*), 2469
`create()` (*in module salt.modules.serverdensity_device*), 2494
`create()` (*in module salt.modules.smartos_vmadm*), 2504
`create()` (*in module salt.modules.splunk_search*), 2537
`create()` (*in module salt.modules.statuspage*), 2562
`create()` (*in module salt.modules.uptime*), 2630
`create()` (*in module salt.modules.vboxmanage*), 2643
`create()` (*in module salt.modules.virtualenv_mod*), 2676
`create()` (*in module salt.modules.win_service*), 2794
`create()` (*in module salt.modules.zfs*), 2899
`create()` (*in module salt.modules.zonecfg*), 2910
`create()` (*in module salt.modules.zookeeper*), 2913
`create()` (*in module salt.modules.zpool*), 2917
`create()` (*in module salt.runners.cloud*), 3102
`create()` (*in module salt.runners.ddns*), 3103
`create()` (*in module salt.states.statuspage*), 3727
`create()` (*salt.cloud.CloudClient method*), 3852
`create_a()` (*in module salt.modules.infoblox*), 1867
`create_access_key()` (*in module salt.modules.boto_iam*), 1394
`create_address()` (*in module salt.cloud.clouds.gce*), 1054
`create_affinity_group()` (*in module salt.cloud.clouds.msazure*), 1072
`create_alarm()` (*in module salt.modules.telemetry*), 2598
`create_alert()` (*in module salt.states.opsgenie*), 3616
`create_alias()` (*in module salt.modules.boto_kms*), 1415
`create_alias()` (*in module salt.modules.boto_lambda*), 1420
`create_api()` (*in module salt.modules.boto_apigateway*), 1331
`create_api_deployment()` (*in module salt.modules.boto_apigateway*), 1331
`create_api_integration()` (*in module salt.modules.boto_apigateway*), 1331
`create_api_integration_response()` (*in module salt.modules.boto_apigateway*), 1332
`create_api_key()` (*in module salt.modules.boto_apigateway*), 1332
`create_api_method()` (*in module salt.modules.boto_apigateway*), 1332
`create_api_method_response()` (*in module salt.modules.boto_apigateway*), 1333
`create_api_model()` (*in module salt.modules.boto_apigateway*), 1333
`create_api_resources()` (*in module salt.modules.boto_apigateway*), 1333
`create_api_stage()` (*in module salt.modules.boto_apigateway*), 1333
`create_app()` (*in module salt.modules.win_iis*), 2754
`create_app()` (*in module salt.states.win_iis*), 3764
`create_apppool()` (*in module salt.modules.win_iis*),

2755
create_apppool() (in module salt.states.win_iis), 3764
create_attach_volumes() (in module salt.cloud.clouds.ec2), 1046
create_attach_volumes() (in module salt.cloud.clouds.gce), 1054
create_attach_volumes() (in module salt.cloud.clouds.msazure), 1072
create_backup() (in module salt.modules.win_iis), 2755
create_balancer() (in module salt.modules.libcloud_loadbalancer), 1969
create_baseline() (in module salt.modules.snapper), 2544
create_binding() (in module salt.modules.win_iis), 2755
create_binding() (in module salt.states.win_iis), 3764
create_block_storage() (in module salt.cloud.clouds.oneandone), 1088
create_ca() (in module salt.modules.tls), 2612
create_ca_signed_cert() (in module salt.modules.tls), 2612
create_cache_cluster() (in module salt.modules.boto3_elasticache), 1303
create_cache_parameter_group() (in module salt.modules.boto3_elasticache), 1303
create_cache_security_group() (in module salt.modules.boto3_elasticache), 1303
create_cache_security_group() (in module salt.modules.boto_elasticache), 1379
create_cache_subnet_group() (in module salt.modules.boto3_elasticache), 1304
create_cert_binding() (in module salt.modules.win_iis), 2756
create_cert_binding() (in module salt.states.win_iis), 3765
create_certificate() (in module salt.modules.x509), 2847
create_circuit() (in module salt.modules.netbox), 2196
create_circuit_provider() (in module salt.modules.netbox), 2196
create_circuit_termination() (in module salt.modules.netbox), 2197
create_circuit_type() (in module salt.modules.netbox), 2197
create_cluster() (in module salt.cloud.clouds.vmware), 1139
create_cluster() (in module salt.modules.vsphere), 2686
create_cname() (in module salt.modules.infoblox), 1867
create_config() (in module salt.cloud.clouds.linode), 1066
create_config() (in module salt.modules.snapper), 2544
create_configmap() (in module salt.modules.kubernetesmod), 1947
create_container() (in module salt.modules.libcloud_storage), 1972
create_continuous_query() (in module salt.modules.influxdbmod), 1858
create_crl() (in module salt.modules.x509), 2849
create_csr() (in module salt.modules.tls), 2613
create_csr() (in module salt.modules.x509), 2850
create_customer_gateway() (in module salt.modules.boto_vpc), 1450
create_data_disk() (in module salt.cloud.clouds.linode), 1066
create_datacenter() (in module salt.cloud.clouds.profitbricks), 1117
create_datacenter() (in module salt.cloud.clouds.vmware), 1139
create_datacenter() (in module salt.modules.vsphere), 2687
create_datasource() (in module salt.modules.grafana4), 1823
create_datasource() (in module salt.modules.jboss7), 1903
create_datastore_cluster() (in module salt.cloud.clouds.vmware), 1139
create_db() (in module salt.modules.influxdbmod), 1858
create_deployment() (in module salt.modules.kubernetesmod), 1947
create_device() (in module salt.modules.netbox), 2197
create_device_role() (in module salt.modules.netbox), 2197
create_device_type() (in module salt.modules.netbox), 2197
create_dhcp_options() (in module salt.modules.boto_vpc), 1450
create_disk() (in module salt.cloud.clouds.gce), 1055
create_disk() (in module salt.modules.vmctl), 2678
create_disk_from_distro() (in module salt.cloud.clouds.linode), 1066
create_diskgroup() (in module salt.modules.vsphere), 2687
create_distribution() (in module salt.modules.boto_cloudfront), 1347
create_dvportgroup() (in module salt.modules.vsphere), 2687
create_dvs() (in module salt.modules.vsphere), 2687
create_elasticsearch_domain() (in module salt.modules.boto3_elasticsearch), 1310

`create_empty_crl()` (*in module salt.modules.tls*), 2615
`create_event()` (*in module salt.modules.pagerduty*), 2302
`create_event()` (*in module salt.modules.victorops*), 2645
`create_event()` (*in module salt.runners.pagerduty*), 3137
`create_event()` (*in module salt.states.pagerduty*), 3617
`create_event()` (*in module salt.states.victorops*), 3748
`create_event_source_mapping()` (*in module salt.modules.boto_lambda*), 1420
`create_extension()` (*in module salt.modules.postgres*), 2374
`create_file_system()` (*in module salt.modules.boto_efs*), 1376
`create_file_vdev()` (*in module salt.modules.zpool*), 2918
`create_firewall_rule()` (*in module salt.modules.neutron*), 2222
`create_floating_ip()` (*in module salt.cloud.clouds.digitalocean*), 1040
`create_floatingip()` (*in module salt.modules.neutron*), 2223
`create_folder()` (*in module salt.cloud.clouds.vmware*), 1139
`create_folder()` (*in module salt.modules.win_task*), 2825
`create_from_template()` (*in module salt.modules.zonecfg*), 2911
`create_function()` (*in module salt.modules.boto_lambda*), 1420
`create_fwrule()` (*in module salt.cloud.clouds.gce*), 1055
`create_global_secondary_index()` (*in module salt.modules.boto_dynamodb*), 1360
`create_grant()` (*in module salt.modules.boto_kms*), 1415
`create_group()` (*in module salt.modules.boto_iam*), 1394
`create_hc()` (*in module salt.cloud.clouds.gce*), 1055
`create_healthcheck()` (*in module salt.modules.boto_route53*), 1430
`create_host()` (*in module salt.modules.infoblox*), 1868
`create_hosted_zone()` (*in module salt.modules.boto3_route53*), 1325
`create_hosted_zone()` (*in module salt.modules.boto_route53*), 1431
`create_identity_pool()` (*in module salt.modules.boto_cognitoidentity*), 1356
`create_ikepolicy()` (*in module salt.modules.neutron*), 2223
`create_image()` (*in module salt.modules.boto_ec2*), 1363
`create_image()` (*in module salt.modules.libcloud_compute*), 1960
`create_instance_profile()` (*in module salt.modules.boto_iam*), 1395
`create_interface()` (*in module salt.modules.netbox*), 2198
`create_interface_connection()` (*in module salt.modules.netbox*), 2198
`create_internet_gateway()` (*in module salt.modules.boto_vpc*), 1451
`create_inventory_item()` (*in module salt.modules.netbox*), 2198
`create_ipaddress()` (*in module salt.modules.netbox*), 2198
`create_ipsec_site_connection()` (*in module salt.modules.neutron*), 2224
`create_ipsecpolicy()` (*in module salt.modules.neutron*), 2224
`create_ipv4_range()` (*in module salt.modules.infoblox*), 1868
`create_issue()` (*in module salt.modules.jira_mod*), 1912
`create_jail()` (*in module salt.modules.poudriere*), 2386
`create_job()` (*in module salt.modules.jenkinsmod*), 1908
`create_key()` (*in module salt.cloud.clouds.digitalocean*), 1040
`create_key()` (*in module salt.modules.boto_ec2*), 1364
`create_key()` (*in module salt.modules.boto_kms*), 1415
`create_key()` (*in module salt.modules.gpg*), 1817
`create_key_pair()` (*in module salt.modules.libcloud_compute*), 1960
`create_keypair()` (*in module salt.cloud.clouds.ec2*), 1046
`create_keyspace()` (*in module salt.modules.cassandra_cql*), 1485
`create_keytab()` (*in module salt.modules.kerberos*), 1924
`create_launch_configuration()` (*in module salt.modules.boto_asg*), 1341
`create_lb()` (*in module salt.cloud.clouds.dimensiondata*), 1043
`create_lb()` (*in module salt.cloud.clouds.gce*), 1055
`create_listeners()` (*in module salt.modules.boto_elb*), 1386
`create_loadbalancer()` (*in module salt.cloud.clouds.profitbricks*), 1117
`create_login_profile()` (*in module salt.modules.boto_iam*), 1395

create_manufacturer() (*in module salt.modules.netbox*), 2199
create_metadata() (*in module salt.modules.postgres*), 2374
create_monitor() (*in module salt.modules.bigip*), 1286
create_monitor() (*in module salt.states.bigip*), 3222
create_mount_target() (*in module salt.modules.boto_efs*), 1376
create_multi() (*salt.modules.xmpp.SendMsgBot class method*), 2863
create_namespace() (*in module salt.modules.k8s*), 1919
create_namespace() (*in module salt.modules.kubernetesmod*), 1947
create_nat_gateway() (*in module salt.modules.boto_vpc*), 1451
create_network() (*in module salt.cloud.clouds.gce*), 1055
create_network() (*in module salt.modules.dockermod*), 1633
create_network() (*in module salt.modules.neutron*), 2225
create_network_acl() (*in module salt.modules.boto_vpc*), 1451
create_network_acl_entry() (*in module salt.modules.boto_vpc*), 1451
create_network_interface() (*in module salt.cloud.clouds.azurearm*), 1036
create_network_interface() (*in module salt.modules.boto_ec2*), 1364
create_node() (*in module salt.cloud.clouds.aliyun*), 1032
create_node() (*in module salt.cloud.clouds.digitalocean*), 1040
create_node() (*in module salt.cloud.clouds.joyent*), 1062
create_node() (*in module salt.cloud.clouds.parallels*), 1114
create_node() (*in module salt.cloud.clouds.proxmox*), 1120
create_node() (*in module salt.cloud.clouds.scaleway*), 1126
create_node() (*in module salt.modules.bigip*), 1286
create_node() (*in module salt.states.bigip*), 3222
create_object() (*in module salt.modules.infoblox*), 1868
create_option_group() (*in module salt.modules.boto_rds*), 1425
create_or_update() (*in module salt.modules.boto_cloudwatch_event*), 1353
create_or_update_alarm() (*in module salt.modules.boto_cloudwatch*), 1351
create_or_update_resource() (*in module salt.modules.pagerduty_util*), 2303
create_or_update_vmextension() (*in module salt.cloud.clouds.azurearm*), 1036
create_org() (*in module salt.modules.grafana4*), 1824
create_org_user() (*in module salt.modules.grafana4*), 1824
create_parameter_group() (*in module salt.modules.boto_rds*), 1425
create_pipeline() (*in module salt.modules.boto_datapipeline*), 1358
create_pkcs12() (*in module salt.modules.tls*), 2615
create_platform() (*in module salt.modules.netbox*), 2199
create_pod() (*in module salt.modules.kubernetesmod*), 1947
create_policy() (*in module salt.modules.boto_elb*), 1387
create_policy() (*in module salt.modules.boto_iam*), 1395
create_policy() (*in module salt.modules.boto_iot*), 1407
create_policy_version() (*in module salt.modules.boto_iam*), 1395
create_policy_version() (*in module salt.modules.boto_iot*), 1407
create_pool() (*in module salt.modules.bigip*), 1286
create_pool() (*in module salt.runners.f5*), 3110
create_pool() (*in module salt.states.bigip*), 3222
create_pool() (*salt.runners.f5.F5Mgmt method*), 3110
create_port() (*in module salt.modules.neutron*), 2225
create_ports_tree() (*in module salt.modules.poudriere*), 2386
create_principal() (*in module salt.modules.kerberos*), 1924
create_private_ip() (*in module salt.cloud.clouds.linode*), 1067
create_private_key() (*in module salt.modules.x509*), 2850
create_profile() (*in module salt.modules.bigip*), 1288
create_profile() (*in module salt.states.bigip*), 3223
create_queue() (*in module salt.modules.aws_sqs*), 1246
create_read_replica() (*in module salt.modules.boto_rds*), 1425
create_record() (*in module salt.modules.libcloud_dns*), 1965
create_replication_group() (*in module salt.modules.boto3_elasticache*), 1304
create_replication_group() (*in module salt.modules.boto_elasticache*), 1380
create_repo

spm command line option, 991

create_retention_policy() (*in module salt.modules.influxdbmod*), 1858

create_role() (*in module salt.modules.boto_iam*), 1395

create_role_policy() (*in module salt.modules.boto_iam*), 1395

create_route() (*in module salt.modules.boto_vpc*), 1452

create_route_table() (*in module salt.modules.boto_vpc*), 1452

create_router() (*in module salt.modules.neutron*), 2225

create_saml_provider() (*in module salt.modules.boto_iam*), 1396

create_secret() (*in module salt.modules.k8s*), 1919

create_secret() (*in module salt.modules.kubernetesmod*), 1947

create_security_group() (*in module salt.modules.neutron*), 2225

create_security_group_rule() (*in module salt.modules.neutron*), 2226

create_self_signed_cert() (*in module salt.modules.tls*), 2615

create_service() (*in module salt.cloud.clouds.msazure*), 1072

create_service() (*in module salt.modules.kubernetesmod*), 1947

create_simple_binding() (*in module salt.modules.jboss7*), 1903

create_site() (*in module salt.modules.netbox*), 2199

create_site() (*in module salt.modules.win_iis*), 2756

create_snapshot() (*in module salt.cloud.clouds.ec2*), 1046

create_snapshot() (*in module salt.cloud.clouds.gce*), 1055

create_snapshot() (*in module salt.cloud.clouds.vmware*), 1140

create_snapshot() (*in module salt.modules.smartos_vmadm*), 2505

create_snapshot() (*in module salt.modules.snapper*), 2544

create_snapshot() (*salt.modules.inspectlib.collector.Inspector method*), 1874

create_ssh_key() (*in module salt.cloud.clouds.oneandone*), 1088

create_stack() (*in module salt.modules.heat*), 1846

create_storage() (*in module salt.cloud.clouds.msazure*), 1073

create_storage_container() (*in module salt.cloud.clouds.msazure*), 1073

create_storage_policy() (*in module salt.modules.vsphere*), 2687

create_stream() (*in module salt.modules.boto_kinesis*), 1412

create_subnet() (*in module salt.modules.boto_vpc*), 1452

create_subnet() (*in module salt.modules.neutron*), 2226

create_subnet_group() (*in module salt.modules.boto_elasticache*), 1380

create_subnet_group() (*in module salt.modules.boto_rds*), 1426

create_subnetwork() (*in module salt.cloud.clouds.gce*), 1055

create_swap_disk() (*in module salt.cloud.clouds.linode*), 1067

create_table() (*in module salt.modules.boto_dynamodb*), 1360

create_table_from_object() (*salt.modules.inspectlib.fsdb.CsvDB method*), 1876

create_tag() (*in module salt.modules.vsphere*), 2688

create_tag_category() (*in module salt.modules.vsphere*), 2688

create_tags() (*in module salt.modules.boto_ec2*), 1364

create_tags() (*in module salt.modules.boto_efs*), 1376

create_target_group() (*in module salt.modules.boto_elbv2*), 1391

create_target_group() (*in module salt.states.boto_elbv2*), 3291

create_task() (*in module salt.modules.win_task*), 2825

create_task_from_xml() (*in module salt.modules.win_task*), 2826

create_thing_type() (*in module salt.modules.boto_iot*), 1408

create_topic_rule() (*in module salt.modules.boto_iot*), 1408

create_update_dashboard() (*in module salt.modules.grafana4*), 1824

create_usage_plan() (*in module salt.modules.boto_apigateway*), 1333

create_user() (*in module salt.modules.boto_iam*), 1396

create_user() (*in module salt.modules.cassandra_cql*), 1485

create_user() (*in module salt.modules.cimc*), 1507

create_user() (*in module salt.modules.drac*), 1662

create_user() (*in module salt.modules.dracr*), 1665

create_user() (*in module salt.modules.grafana4*), 1824

create_user() (*in module salt.modules.ilo*), 1852

create_user() (*in module salt.modules.influxdbmod*), 1859

`create_user()` (*in module salt.modules.ipmi*), 1884
`create_user()` (*in module salt.modules.splunk*), 2536
`create_vdir()` (*in module salt.modules.win_iis*), 2756
`create_vdir()` (*in module salt.states.win_iis*), 3765
`create_virtual()` (*in module salt.modules.bigip*), 1288
`create_virtual()` (*in module salt.states.bigip*), 3224
`create_vm()` (*in module salt.modules.vsphere*), 2689
`create_vmfs_datastore()` (*in module salt.modules.vsphere*), 2690
`create_volume()` (*in module salt.cloud.clouds.ec2*), 1046
`create_volume()` (*in module salt.modules.boto_ec2*), 1364
`create_volume()` (*in module salt.modules.dockermod*), 1636
`create_volume()` (*in module salt.modules.glusterfs*), 1812
`create_volume()` (*in module salt.modules.libcloud_compute*), 1960
`create_volume_snapshot()` (*in module salt.modules.libcloud_compute*), 1961
`create_vpnservice()` (*in module salt.modules.neutron*), 2226
`create_vs()` (*in module salt.runners.f5*), 3111
`create_vs()` (*salt.runners.f5.F5Mgmt method*), 3110
`create_win_salt_restart_task()` (*in module salt.modules.win_service*), 2795
`create_xml_path()` (*in module salt.modules.virt*), 2647
`create_xml_str()` (*in module salt.modules.virt*), 2647
`create_zone()` (*in module salt.modules.boto_route53*), 1432
`create_zone()` (*in module salt.modules.libcloud_dns*), 1966
`creategroup()` (*in module salt.states.boto_elasticache*), 3281
`createsuperuser()` (*in module salt.modules.djangomod*), 1605
`critical()` (*in module salt.modules.logmod*), 1982
`crl_managed()` (*in module salt.states.x509*), 3791
`crontab()` (*in module salt.modules.osquery*), 2289
`cross_test()` (*in module salt.modules.test*), 2601
`csr_managed()` (*in module salt.states.x509*), 3791
`CsvDB` (*class in salt.modules.inspectlib.fsdb*), 1876
`CsvDBEntity` (*class in salt.modules.inspectlib.fsdb*), 1878
`ctrl_alt_del()` (*in module salt.modules.virt*), 2648
`current()` (*in module salt.modules.win_useradd*), 2835
`current_branch()` (*in module salt.modules.git*), 1767
`custom()` (*in module salt.modules.status*), 2559
`custom()` (*in module salt.modules.supervisord*), 2566
`custom-execution-modules`
    `jinja filters`, 452
`custom-jinja-filters`
    `jinja filters`, 452
`customer_gateway_exists()` (*in module salt.modules.boto_vpc*), 1452
`cwd()` (*in module salt.grains.core*), 1177
`cxxflags_contains()` (*in module salt.modules.makeconf*), 2062
`cython_enable`
    `conf/master`, 75
    `conf/minion`, 146

D

`daclConstants` (*class in salt.modules.win_dacl*), 2722
`dangling()` (*in module salt.modules.dockermod*), 1636
`dashboard_absent()` (*in module salt.states.grafana*), 3473
`dashboard_present()` (*in module salt.states.grafana*), 3473
`data()` (*in module salt.modules.match*), 2071
`datadir_exists()` (*in module salt.modules.postgres*), 2374
`datadir_init()` (*in module salt.modules.postgres*), 2375
`datasource_exists()` (*in module salt.states.jboss7*), 3507
`date_format`
    `jinja filters`, 436
`db_alter()` (*in module salt.modules.postgres*), 2375
`db_check()` (*in module salt.modules.mysql*), 2099
`db_create()` (*in module salt.modules.influxdb08mod*), 1862
`db_create()` (*in module salt.modules.mssql*), 2096
`db_create()` (*in module salt.modules.mysql*), 2099
`db_create()` (*in module salt.modules.postgres*), 2375
`db_exists()` (*in module salt.modules.influxdb08mod*), 1862
`db_exists()` (*in module salt.modules.influxdbmod*), 1859
`db_exists()` (*in module salt.modules.mongodb*), 2086
`db_exists()` (*in module salt.modules.mssql*), 2096
`db_exists()` (*in module salt.modules.mysql*), 2100
`db_exists()` (*in module salt.modules.postgres*), 2375
`DB_FILE` (*salt.modules.inspectlib.EnvLoader attribute*), 1876
`db_get()` (*in module salt.modules.mysql*), 2100
`db_list()` (*in module salt.modules.influxdb08mod*), 1862
`db_list()` (*in module salt.modules.mongodb*), 2087
`db_list()` (*in module salt.modules.mssql*), 2096
`db_list()` (*in module salt.modules.mysql*), 2100
`db_list()` (*in module salt.modules.postgres*), 2376
`db_optimize()` (*in module salt.modules.mysql*), 2100

`db_remove()` (*in module salt.modules.influxdb08mod*), 1863
`db_remove()` (*in module salt.modules.mongodb*), 2087
`db_remove()` (*in module salt.modules.mssql*), 2096
`db_remove()` (*in module salt.modules.mysql*), 2100
`db_remove()` (*in module salt.modules.postgres*), 2376
`db_repair()` (*in module salt.modules.mysql*), 2100
`db_tables()` (*in module salt.modules.mysql*), 2100
`DBHandle` (*class in salt.modules.inspectlib.dbhandle*), 1875
`DBHandleBase` (*class in salt.modules.inspectlib.dbhandle*), 1875
`dbsize()` (*in module salt.modules.redismod*), 2428
`deactivate_license()` (*in module salt.modules.panos*), 2306
`deactivate_mfa_device()` (*in module salt.modules.boto_iam*), 1396
`dead()` (*in module salt.states.service*), 3710
`dead()` (*in module salt.states.supervisord*), 3729
`deb_packages()` (*in module salt.modules.osquery*), 2289
`debug()` (*in module salt.modules.logmod*), 1982
`debugging`
    `jinja filters`, 451
`dec()` (*in module salt.modules.nacl*), 2109
`dec()` (*in module salt.runners.nacl*), 3129
`dec_file()` (*in module salt.modules.nacl*), 2109
`dec_file()` (*in module salt.runners.nacl*), 3129
`decode()` (*in module salt.states.file*), 3426
`decode()` (*salt.modules.dockermod.DockerJSONDecoder method*), 1618
`decrease_stream_retention_period()` (*in module salt.modules.boto_kinesis*), 1413
`decrement()` (*in module salt.modules.memcached*), 2077
`decrypt()` (*in module salt.modules.boto_kms*), 1415
`decrypt()` (*in module salt.modules.gpg*), 1818
`decrypt_pillar`
    `conf/master`, 99
    `conf/minion`, 154
`decrypt_pillar_default`
    `conf/master`, 100
    `conf/minion`, 155
`decrypt_pillar_delimiter`
    `conf/master`, 99
    `conf/minion`, 155
`decrypt_pillar_renderers`
    `conf/master`, 100
    `conf/minion`, 155
`deepcopy()` (*in module salt.modules.defaults*), 1599
`default()` (*in module salt.modules.pyenv*), 2409
`default()` (*in module salt.modules.rbenv*), 2424
`DEFAULT_CACHE_PATH` (*salt.modules.inspectlib.EnvLoader attribute*), 1876
`default_config()` (*in module salt.modules.linux_sysctl*), 1979
`default_gateway()` (*in module salt.grains.core*), 1177
`default_hash()` (*in module salt.modules.bsd_shadow*), 1465
`default_hash()` (*in module salt.modules.solaris_shadow*), 2516
`default_include`
    `conf/master`, 117
    `conf/minion`, 163
`default_keychain()` (*in module salt.states.mac_keychain*), 3549
`DEFAULT_MINION_CONFIG_PATH` (*salt.modules.inspectlib.collector.Inspector attribute*), 1874
`DEFAULT_PID_PATH` (*salt.modules.inspectlib.EnvLoader attribute*), 1876
`default_route()` (*in module salt.modules.network*), 2216
`default_security_rule_get()` (*in module salt.modules.azurearm_network*), 1255
`default_security_rules_list()` (*in module salt.modules.azurearm_network*), 1255
`DEFAULT_TIMEOUT` (*salt.modules.ansiblegate.AnsibleModuleCaller attribute*), 1208
`default_top`
    `conf/minion`, 149
`default_zone()` (*in module salt.modules.firewalld*), 1733
`defaults()` (*in module salt.grains.junos*), 1180
`defaults()` (*in module salt.modules.napalm_formula*), 2153
`define_task()` (*in module salt.modules.kapacitor*), 1922
`define_vol_xml_path()` (*in module salt.modules.virt*), 2648
`define_vol_xml_str()` (*in module salt.modules.virt*), 2648
`define_xml_path()` (*in module salt.modules.virt*), 2649
`define_xml_str()` (*in module salt.modules.virt*), 2649
`defragment()` (*in module salt.modules.btrfs*), 1467
`defragment()` (*in module salt.modules.xfs*), 2860
`deinstall()` (*in module salt.modules.freebsdports*), 1745
`del_cached_domain()` (*in module salt.runners.digicertapi*), 3105
`del_cached_domain()` (*in module salt.runners.venafiapi*), 3158
`del_export()` (*in module salt.modules.nfs3*), 2243
`del_password()` (*in module*

salt.modules.bsd_shadow), 1465
del_password() (*in module salt.modules.mac_shadow*), 2042
del_password() (*in module salt.modules.solaris_shadow*), 2516
del_repo() (*in module salt.modules.aptpkg*), 1216
del_repo() (*in module salt.modules.opkg*), 2281
del_repo() (*in module salt.modules.xbpspkg*), 2857
del_repo() (*in module salt.modules.yumpkg*), 2864
del_repo() (*in module salt.modules.zypperpkg*), 2924
del_repo_key() (*in module salt.modules.aptpkg*), 1217
del_store() (*in module salt.modules.win_certutil*), 2720
del_store() (*in module salt.states.win_certutil*), 3757
del_tags() (*in module salt.cloud.clouds.ec2*), 1047
del_token() (*in module salt.runners.auth*), 3096
delete() (*in module salt.modules.aix_group*), 1203
delete() (*in module salt.modules.beacons*), 1283
delete() (*in module salt.modules.boto_asg*), 1342
delete() (*in module salt.modules.boto_cfn*), 1345
delete() (*in module salt.modules.boto_cloudtrail*), 1349
delete() (*in module salt.modules.boto_cloudwatch_event*), 1354
delete() (*in module salt.modules.boto_dynamodb*), 1360
delete() (*in module salt.modules.boto_elasticache*), 1380
delete() (*in module salt.modules.boto_elasticsearch_domain*), 1384
delete() (*in module salt.modules.boto_elb*), 1387
delete() (*in module salt.modules.boto_rds*), 1426
delete() (*in module salt.modules.boto_s3_bucket*), 1436
delete() (*in module salt.modules.boto_secgroup*), 1442
delete() (*in module salt.modules.boto_sns*), 1444
delete() (*in module salt.modules.boto_sqs*), 1446
delete() (*in module salt.modules.boto_vpc*), 1453
delete() (*in module salt.modules.bridge*), 1464
delete() (*in module salt.modules.btrfs*), 1467
delete() (*in module salt.modules.consul*), 1568
delete() (*in module salt.modules.ddns*), 1590
delete() (*in module salt.modules.groupadd*), 1837
delete() (*in module salt.modules.inspector*), 1879
delete() (*in module salt.modules.ipset*), 1896
delete() (*in module salt.modules.iptables*), 1900
delete() (*in module salt.modules.layman*), 1953
delete() (*in module salt.modules.ldap3*), 1956
delete() (*in module salt.modules.mac_group*), 2027
delete() (*in module salt.modules.mac_user*), 2057
delete() (*in module salt.modules.mac_xattr*), 2059
delete() (*in module salt.modules.macdefaults*), 2025
delete() (*in module salt.modules.memcached*), 2077
delete() (*in module salt.modules.mine*), 2078
delete() (*in module salt.modules.namecheap_domains_ns*), 2117
delete() (*in module salt.modules.nftables*), 2245
delete() (*in module salt.modules.nova*), 2253
delete() (*in module salt.modules.openstack_config*), 2277
delete() (*in module salt.modules.parallels*), 2320
delete() (*in module salt.modules.pdbedit*), 2332
delete() (*in module salt.modules.postfix*), 2372
delete() (*in module salt.modules.pw_group*), 2406
delete() (*in module salt.modules.pw_user*), 2408
delete() (*in module salt.modules.redismod*), 2428
delete() (*in module salt.modules.rh_service*), 2443
delete() (*in module salt.modules.s3*), 2459
delete() (*in module salt.modules.schedule*), 2481
delete() (*in module salt.modules.sdb*), 2485
delete() (*in module salt.modules.serverdensity_device*), 2494
delete() (*in module salt.modules.smartos_imgadm*), 2500
delete() (*in module salt.modules.smartos_nictagadm*), 2502
delete() (*in module salt.modules.smartos_vmadm*), 2505
delete() (*in module salt.modules.solaris_group*), 2515
delete() (*in module salt.modules.solaris_user*), 2519
delete() (*in module salt.modules.splunk_search*), 2537
delete() (*in module salt.modules.statuspage*), 2563
delete() (*in module salt.modules.swift*), 2574
delete() (*in module salt.modules.uptime*), 2630
delete() (*in module salt.modules.useradd*), 2632
delete() (*in module salt.modules.win_groupadd*), 2753
delete() (*in module salt.modules.win_service*), 2796
delete() (*in module salt.modules.win_useradd*), 2835
delete() (*in module salt.modules.zonecfg*), 2911
delete() (*in module salt.modules.zookeeper*), 2914
delete() (*in module salt.queues.pgjsonb_queue*), 3087
delete() (*in module salt.queues.sqlite_queue*), 3088
delete() (*in module salt.runners.ddns*), 3104
delete() (*in module salt.runners.queue*), 3140
delete() (*in module salt.runners.reactor*), 3142
delete() (*in module salt.runners.sdb*), 3148
delete() (*in module salt.sdb.cache*), 3162
delete() (*in module salt.sdb.etcd_db*), 3165
delete() (*in module salt.sdb.redis_sdb*), 3167
delete() (*in module salt.states.iptables*), 3506
delete() (*in module salt.states.mac_xattr*), 3550
delete() (*in module salt.states.nftables*), 3611
delete() (*in module salt.states.statuspage*), 3727
delete() (*in module salt.thorium.reg*), 3826
delete() (*in module salt.wheel.key*), 3839

`delete()` (*salt.modules.inspectlib.fsdb.CsvDB method*), 1876
`delete_()` (*in module salt.modules.mdata*), 2076
`delete_a()` (*in module salt.modules.infoblox*), 1868
`delete_access_key()` (*in module salt.modules.boto_iam*), 1396
`delete_address()` (*in module salt.cloud.clouds.gce*), 1055
`delete_advanced_configs()` (*in module salt.modules.vsphere*), 2690
`delete_affinity_group()` (*in module salt.cloud.clouds.msazure*), 1073
`delete_alarm()` (*in module salt.modules.boto_cloudwatch*), 1352
`delete_alarms()` (*in module salt.modules.telemetry*), 2599
`delete_alias()` (*in module salt.modules.boto_lambda*), 1420
`delete_api()` (*in module salt.modules.boto_apigateway*), 1334
`delete_api_deployment()` (*in module salt.modules.boto_apigateway*), 1334
`delete_api_integration()` (*in module salt.modules.boto_apigateway*), 1334
`delete_api_integration_response()` (*in module salt.modules.boto_apigateway*), 1334
`delete_api_key()` (*in module salt.modules.boto_apigateway*), 1335
`delete_api_method()` (*in module salt.modules.boto_apigateway*), 1335
`delete_api_method_response()` (*in module salt.modules.boto_apigateway*), 1335
`delete_api_model()` (*in module salt.modules.boto_apigateway*), 1335
`delete_api_resources()` (*in module salt.modules.boto_apigateway*), 1335
`delete_api_stage()` (*in module salt.modules.boto_apigateway*), 1336
`delete_backup()` (*in module salt.modules.file*), 1709
`delete_blob()` (*in module salt.cloud.clouds.azurearm*), 1036
`delete_cache_cluster()` (*in module salt.modules.boto3_elasticache*), 1304
`delete_cache_parameter_group()` (*in module salt.modules.boto3_elasticache*), 1304
`delete_cache_security_group()` (*in module salt.modules.boto3_elasticache*), 1304
`delete_cache_security_group()` (*in module salt.modules.boto_elasticache*), 1380
`delete_cache_subnet_group()` (*in module salt.modules.boto3_elasticache*), 1305
`delete_chain()` (*in module salt.modules.iptables*), 1900
`delete_chain()` (*in module salt.modules.nftables*), 2245
`delete_cname()` (*in module salt.modules.infoblox*), 1869
`delete_config()` (*in module salt.proxy.nxos*), 3074
`delete_config()` (*in module salt.states.panos*), 3624
`delete_configmap()` (*in module salt.modules.kubernetesmod*), 1947
`delete_container()` (*in module salt.modules.libcloud_storage*), 1972
`delete_cors()` (*in module salt.modules.boto_s3_bucket*), 1436
`delete_customer_gateway()` (*in module salt.modules.boto_vpc*), 1453
`delete_dashboard()` (*in module salt.modules.grafana4*), 1825
`delete_datasource()` (*in module salt.modules.grafana4*), 1825
`delete_deployment()` (*in module salt.modules.kubernetesmod*), 1947
`delete_device()` (*in module salt.runners.vistara*), 3160
`delete_dhcp_options()` (*in module salt.modules.boto_vpc*), 1453
`delete_dict()` (*in module salt.wheel.key*), 3839
`delete_disk()` (*in module salt.cloud.clouds.gce*), 1056
`delete_disk()` (*in module salt.cloud.clouds.msazure*), 1073
`delete_elasticsearch_domain()` (*in module salt.modules.boto3_elasticsearch*), 1313
`delete_elasticsearch_service_role()` (*in module salt.modules.boto3_elasticsearch*), 1313
`delete_event_source_mapping()` (*in module salt.modules.boto_lambda*), 1421
`delete_file_system()` (*in module salt.modules.boto_efs*), 1377
`delete_firewall_rule()` (*in module salt.modules.neutron*), 2227
`delete_floating_ip()` (*in module salt.cloud.clouds.digitalocean*), 1040
`delete_floatingip()` (*in module salt.modules.neutron*), 2227
`delete_folder()` (*in module salt.modules.win_task*), 2826
`delete_function()` (*in module salt.modules.boto_lambda*), 1421
`delete_fwrule()` (*in module salt.cloud.clouds.gce*), 1056
`delete_group()` (*in module salt.modules.boto_iam*), 1396
`delete_group_policy()` (*in module salt.modules.boto_iam*), 1396
`delete_hc()` (*in module salt.cloud.clouds.gce*), 1056
`delete_host()` (*in module salt.modules.ddns*), 1590
`delete_host()` (*in module salt.modules.infoblox*),

1869
delete_host() (*in module salt.modules.omapi*), 2269
delete_host() (*in module salt.runners.ddns*), 3104
delete_hosted_zone() (*in module salt.modules.boto3_route53*), 1326
delete_hosted_zone_by_domain() (*in module salt.modules.boto3_route53*), 1326
delete_identity_pools() (*in module salt.modules.boto_cognitoidentity*), 1356
delete_ikepolicy() (*in module salt.modules.neutron*), 2227
delete_image() (*in module salt.modules.libcloud_compute*), 1961
delete_input_endpoint() (*in module salt.cloud.clouds.msazure*), 1073
delete_instance_profile() (*in module salt.modules.boto_iam*), 1397
delete_interface() (*in module salt.cloud.clouds.azurearm*), 1036
delete_interface() (*in module salt.modules.netbox*), 2199
delete_internet_gateway() (*in module salt.modules.boto_vpc*), 1453
delete_inventory_item() (*in module salt.modules.netbox*), 2199
delete_ipaddress() (*in module salt.modules.netbox*), 2199
delete_ipsec_site_connection() (*in module salt.modules.neutron*), 2227
delete_ipsecpolicy() (*in module salt.modules.neutron*), 2227
delete_ipv4_range() (*in module salt.modules.infoblox*), 1869
delete_jail() (*in module salt.modules.poudriere*), 2386
delete_job() (*in module salt.modules.jenkinsmod*), 1909
delete_key() (*in module salt.cloud.clouds.joyent*), 1062
delete_key() (*in module salt.modules.boto_ec2*), 1365
delete_key() (*in module salt.modules.gpg*), 1818
delete_key_pair() (*in module salt.modules.libcloud_compute*), 1961
delete_key_recursive() (*in module salt.modules.reg*), 2434
delete_keypair() (*in module salt.cloud.clouds.ec2*), 1047
delete_launch_configuration() (*in module salt.modules.boto_asg*), 1342
delete_lb() (*in module salt.cloud.clouds.gce*), 1056
delete_license() (*in module salt.modules.panos*), 2306
delete_lifecycle_configuration() (*in module salt.modules.boto_s3_bucket*), 1436
delete_listeners() (*in module salt.modules.boto_elb*), 1387
delete_login_profile() (*in module salt.modules.boto_iam*), 1397
delete_managed_disk() (*in module salt.cloud.clouds.azurearm*), 1036
delete_management_certificate() (*in module salt.cloud.clouds.msazure*), 1074
delete_message() (*in module salt.modules.aws_sqs*), 1247
delete_monitor() (*in module salt.modules.bigip*), 1290
delete_monitor() (*in module salt.states.bigip*), 3225
delete_mount_cache() (*in module salt.modules.mount*), 2091
delete_mount_target() (*in module salt.modules.boto_efs*), 1377
delete_namespace() (*in module salt.modules.kubernetesmod*), 1947
delete_nat_gateway() (*in module salt.modules.boto_vpc*), 1453
delete_network() (*in module salt.cloud.clouds.gce*), 1056
delete_network() (*in module salt.modules.neutron*), 2228
delete_network_acl() (*in module salt.modules.boto_vpc*), 1454
delete_network_acl_entry() (*in module salt.modules.boto_vpc*), 1454
delete_network_interface() (*in module salt.modules.boto_ec2*), 1365
delete_node() (*in module salt.modules.bigip*), 1291
delete_node() (*in module salt.states.bigip*), 3225
delete_object() (*in module salt.modules.infoblox*), 1869
delete_object() (*in module salt.modules.libcloud_storage*), 1972
delete_objects() (*in module salt.modules.boto_s3_bucket*), 1436
delete_option_group() (*in module salt.modules.boto_rds*), 1426
delete_org() (*in module salt.modules.grafana4*), 1825
delete_org_user() (*in module salt.modules.grafana4*), 1825
delete_parameter_group() (*in module salt.modules.boto_rds*), 1426
delete_peers() (*in module salt.modules.napalm_ntp*), 2176
delete_pipeline() (*in module salt.modules.boto_datapipeline*), 1358
delete_pod() (*in module salt.modules.kubernetesmod*), 1948
delete_policy() (*in module salt.modules.boto_elb*), 1387

delete_policy() (*in module salt.modules.boto_iam*), 1397
delete_policy() (*in module salt.modules.boto_iot*), 1408
delete_policy() (*in module salt.modules.boto_s3_bucket*), 1436
delete_policy() (*in module salt.modules.rabbitmq*), 2413
delete_policy_version() (*in module salt.modules.boto_iam*), 1397
delete_policy_version() (*in module salt.modules.boto_iot*), 1408
delete_pool() (*in module salt.modules.bigip*), 1291
delete_pool() (*in module salt.states.bigip*), 3225
delete_pool_member() (*in module salt.modules.bigip*), 1291
delete_pool_member() (*in module salt.states.bigip*), 3225
delete_port() (*in module salt.modules.neutron*), 2228
delete_principal() (*in module salt.modules.kerberos*), 1924
delete_probes() (*in module salt.modules.napalm_probes*), 2179
delete_profile() (*in module salt.modules.bigip*), 1291
delete_profile() (*in module salt.states.bigip*), 3225
delete_queue() (*in module salt.modules.aws_sqs*), 1247
delete_quota() (*in module salt.modules.neutron*), 2228
delete_record() (*in module salt.modules.boto_route53*), 1432
delete_record() (*in module salt.modules.libcloud_dns*), 1966
delete_record() (*in module salt.modules.servicenow*), 2495
delete_replication() (*in module salt.modules.boto_s3_bucket*), 1436
delete_replication_group() (*in module salt.modules.boto3_elasticache*), 1305
delete_replication_group() (*in module salt.modules.boto_elasticache*), 1380
delete_resource() (*in module salt.modules.pagerduty_util*), 2303
delete_role() (*in module salt.modules.boto_iam*), 1397
delete_role_policy() (*in module salt.modules.boto_iam*), 1397
delete_route() (*in module salt.modules.boto_vpc*), 1454
delete_route_table() (*in module salt.modules.boto_vpc*), 1455
delete_router() (*in module salt.modules.neutron*), 2228
delete_rule() (*in module salt.modules.win_firewall*), 2745
delete_saml_provider() (*in module salt.modules.boto_iam*), 1397
delete_secret() (*in module salt.modules.k8s*), 1920
delete_secret() (*in module salt.modules.kubernetesmod*), 1948
delete_secret() (*in module salt.modules.vault*), 2640
delete_security_group() (*in module salt.modules.neutron*), 2228
delete_security_group_rule() (*in module salt.modules.neutron*), 2229
delete_server() (*in module salt.modules.lvs*), 1985
delete_server_cert() (*in module salt.modules.boto_iam*), 1398
delete_servers() (*in module salt.modules.napalm_ntp*), 2176
delete_service() (*in module salt.cloud.clouds.msazure*), 1074
delete_service() (*in module salt.modules.firewalld*), 1733
delete_service() (*in module salt.modules.kubernetesmod*), 1948
delete_service() (*in module salt.modules.lvs*), 1985
delete_service_certificate() (*in module salt.cloud.clouds.msazure*), 1074
delete_set() (*in module salt.modules.ipset*), 1896
delete_snapshot() (*in module salt.cloud.clouds.ec2*), 1047
delete_snapshot() (*in module salt.cloud.clouds.gce*), 1056
delete_snapshot() (*in module salt.modules.parallels*), 2321
delete_snapshot() (*in module salt.modules.smartos_vmadm*), 2505
delete_snapshot() (*in module salt.modules.snapper*), 2545
delete_snapshots() (*in module salt.modules.virt*), 2649
delete_ssh_key() (*in module salt.modules.ilo*), 1853
delete_stack() (*in module salt.modules.heat*), 1846
delete_storage() (*in module salt.cloud.clouds.msazure*), 1074
delete_storage_container() (*in module salt.cloud.clouds.msazure*), 1074
delete_stream() (*in module salt.modules.boto_kinesis*), 1413
delete_subnet() (*in module salt.modules.boto_vpc*), 1455
delete_subnet() (*in module salt.modules.neutron*), 2229
delete_subnet_group() (*in module

salt.modules.boto_elasticache), 1380
`delete_subnet_group()` (*in module salt.modules.boto_rds*), 1426
`delete_subnetwork()` (*in module salt.cloud.clouds.gce*), 1056
`delete_table()` (*in module salt.modules.nftables*), 2245
`delete_tag()` (*in module salt.modules.vsphere*), 2690
`delete_tag_category()` (*in module salt.modules.vsphere*), 2691
`delete_tagging()` (*in module salt.modules.boto_s3_bucket*), 1437
`delete_tags()` (*in module salt.modules.boto_ec2*), 1365
`delete_tags()` (*in module salt.modules.boto_efs*), 1377
`delete_tags()` (*in module salt.modules.boto_elb*), 1387
`delete_tags()` (*in module salt.modules.boto_secgroup*), 1442
`delete_target_group()` (*in module salt.modules.boto_elbv2*), 1392
`delete_target_group()` (*in module salt.states.boto_elbv2*), 3291
`delete_task()` (*in module salt.modules.kapacitor*), 1923
`delete_task()` (*in module salt.modules.win_task*), 2826
`delete_thing_type()` (*in module salt.modules.boto_iot*), 1408
`delete_topic_rule()` (*in module salt.modules.boto_iot*), 1409
`delete_transaction()` (*in module salt.modules.bigip*), 1292
`delete_upstream()` (*in module salt.modules.rabbitmq*), 2413
`delete_usage_plan()` (*in module salt.modules.boto_apigateway*), 1336
`delete_user()` (*in module salt.modules.boto_iam*), 1398
`delete_user()` (*in module salt.modules.drac*), 1662
`delete_user()` (*in module salt.modules.dracr*), 1666
`delete_user()` (*in module salt.modules.grafana4*), 1825
`delete_user()` (*in module salt.modules.ilo*), 1853
`delete_user()` (*in module salt.modules.rabbitmq*), 2413
`delete_user()` (*in module salt.modules.splunk*), 2536
`delete_user_org()` (*in module salt.modules.grafana4*), 1826
`delete_user_policy()` (*in module salt.modules.boto_iam*), 1398
`delete_users()` (*in module salt.modules.napalm_users*), 2187
`delete_value()` (*in module salt.modules.reg*), 2434
`delete_vhost()` (*in module salt.modules.rabbitmq*), 2414
`delete_virtual()` (*in module salt.modules.bigip*), 1292
`delete_virtual()` (*in module salt.states.bigip*), 3225
`delete_virtual_mfa_device()` (*in module salt.modules.boto_iam*), 1398
`delete_vm()` (*in module salt.modules.vsphere*), 2691
`delete_volume()` (*in module salt.cloud.clouds.ec2*), 1047
`delete_volume()` (*in module salt.modules.boto_ec2*), 1365
`delete_volume()` (*in module salt.modules.glusterfs*), 1812
`delete_vpc_peering_connection()` (*in module salt.modules.boto_vpc*), 1455
`delete_vpc_peering_connection()` (*in module salt.states.boto_vpc*), 3329
`delete_vpnservice()` (*in module salt.modules.neutron*), 2229
`delete_website()` (*in module salt.modules.boto_s3_bucket*), 1437
`delete_zone()` (*in module salt.modules.boto_route53*), 1432
`delete_zone()` (*in module salt.modules.firewalld*), 1733
`delete_zone()` (*in module salt.modules.libcloud_dns*), 1966
`deleteAllActivationKeys()` (*in module salt.runners.spacewalk*), 3152
`deleteAllGroups()` (*in module salt.runners.spacewalk*), 3152
`deleteAllSystems()` (*in module salt.runners.spacewalk*), 3152
`delfacl()` (*in module salt.modules.linux_acl*), 1975
`delif()` (*in module salt.modules.bridge*), 1464
`delkey()` (*in module salt.modules.grains*), 1831
`delta_import()` (*in module salt.modules.solr*), 2529
`deluser()` (*in module salt.modules.aix_group*), 1203
`deluser()` (*in module salt.modules.groupadd*), 1837
`deluser()` (*in module salt.modules.mac_group*), 2027
`deluser()` (*in module salt.modules.pw_group*), 2406
`deluser()` (*in module salt.modules.win_groupadd*), 2753
`delval()` (*in module salt.modules.grains*), 1831
`delvol_on_destroy()` (*in module salt.cloud.clouds.ec2*), 1047
`DemandZeroCount` (*salt.modules.win_status.SYSTEM_PERFORMANCE_I attribute*), 2808
`deny()` (*in module salt.modules.apf*), 1212
`deny()` (*in module salt.modules.csf*), 1582
`depclean()` (*in module salt.modules.ebuildpkg*), 1675
`depends()` (*in module salt.modules.dockermod*), 1636

deploy() (*in module salt.modules.jboss7*), 1904
deploy_password() (*in module salt.modules.dracr*), 1666
deploy_snmp() (*in module salt.modules.dracr*), 1666
deploy_war() (*in module salt.modules.tomcat*), 2619
deployed() (*in module salt.states.heat*), 3484
deployed() (*in module salt.states.jboss7*), 3508
deployed() (*in module salt.states.win_iis*), 3766
deployment_absent() (*in module salt.states.kubernetes*), 3529
deployment_cancel() (*in module salt.modules.azurearm_resource*), 1271
deployment_check_existence() (*in module salt.modules.azurearm_resource*), 1272
deployment_create_or_update() (*in module salt.modules.azurearm_resource*), 1272
deployment_delete() (*in module salt.modules.azurearm_resource*), 1273
deployment_export_template() (*in module salt.modules.azurearm_resource*), 1273
deployment_get() (*in module salt.modules.azurearm_resource*), 1273
deployment_operation_get() (*in module salt.modules.azurearm_resource*), 1274
deployment_operations_list() (*in module salt.modules.azurearm_resource*), 1274
deployment_present() (*in module salt.states.kubernetes*), 3529
deployment_validate() (*in module salt.modules.azurearm_resource*), 1274
deployments() (*in module salt.modules.kubernetesmod*), 1948
deployments_list() (*in module salt.modules.azurearm_resource*), 1275
deprecate_thing_type() (*in module salt.modules.boto_iot*), 1409
depth (*salt.pillar.sql_base.SqlBaseExtPillar attribute*), 3034
deregister_instances() (*in module salt.modules.boto_elb*), 1387
deregister_targets() (*in module salt.modules.boto_elbv2*), 1392
describe() (*in module salt.modules.boto_cfn*), 1345
describe() (*in module salt.modules.boto_cloudtrail*), 1349
describe() (*in module salt.modules.boto_cloudwatch_event*), 1354
describe() (*in module salt.modules.boto_dynamodb*), 1360
describe() (*in module salt.modules.boto_elasticsearch_domain*), 1384
describe() (*in module salt.modules.boto_rds*), 1427
describe() (*in module salt.modules.boto_s3_bucket*), 1437
describe() (*in module salt.modules.boto_vpc*), 1455
describe() (*in module salt.modules.git*), 1768
describe() (*in module salt.modules.hg*), 1848
describe_alias() (*in module salt.modules.boto_lambda*), 1421
describe_api_deployment() (*in module salt.modules.boto_apigateway*), 1336
describe_api_deployments() (*in module salt.modules.boto_apigateway*), 1336
describe_api_integration() (*in module salt.modules.boto_apigateway*), 1336
describe_api_integration_response() (*in module salt.modules.boto_apigateway*), 1336
describe_api_key() (*in module salt.modules.boto_apigateway*), 1337
describe_api_keys() (*in module salt.modules.boto_apigateway*), 1337
describe_api_method() (*in module salt.modules.boto_apigateway*), 1337
describe_api_method_response() (*in module salt.modules.boto_apigateway*), 1337
describe_api_model() (*in module salt.modules.boto_apigateway*), 1337
describe_api_models() (*in module salt.modules.boto_apigateway*), 1337
describe_api_resource() (*in module salt.modules.boto_apigateway*), 1337
describe_api_resource_method() (*in module salt.modules.boto_apigateway*), 1338
describe_api_resources() (*in module salt.modules.boto_apigateway*), 1338
describe_api_stage() (*in module salt.modules.boto_apigateway*), 1338
describe_api_stages() (*in module salt.modules.boto_apigateway*), 1338
describe_apis() (*in module salt.modules.boto_apigateway*), 1338
describe_cache_clusters() (*in module salt.modules.boto3_elasticache*), 1305
describe_cache_parameter_groups() (*in module salt.modules.boto3_elasticache*), 1305
describe_cache_security_groups() (*in module salt.modules.boto3_elasticache*), 1305
describe_cache_subnet_groups() (*in module salt.modules.boto3_elasticache*), 1305
describe_db_instances() (*in module salt.modules.boto_rds*), 1427
describe_db_subnet_groups() (*in module salt.modules.boto_rds*), 1427
describe_elasticsearch_domain() (*in module salt.modules.boto3_elasticsearch*), 1313
describe_elasticsearch_domain_config() (*in module salt.modules.boto3_elasticsearch*),

1314
describe_elasticsearch_domains() (*in module salt.modules.boto3_elasticsearch*), 1314
describe_elasticsearch_instance_type_limits() (*in module salt.modules.boto3_elasticsearch*), 1314
describe_event_source_mapping() (*in module salt.modules.boto_lambda*), 1421
describe_function() (*in module salt.modules.boto_lambda*), 1421
describe_hosted_zones() (*in module salt.modules.boto_route53*), 1432
describe_identity_pools() (*in module salt.modules.boto_cognitoidentity*), 1356
describe_key() (*in module salt.modules.boto_kms*), 1416
describe_launch_configuration() (*in module salt.modules.boto_asg*), 1342
describe_nat_gateways() (*in module salt.modules.boto_vpc*), 1456
describe_parameter_group() (*in module salt.modules.boto_rds*), 1427
describe_parameters() (*in module salt.modules.boto_rds*), 1427
describe_pipelines() (*in module salt.modules.boto_datapipeline*), 1358
describe_policy() (*in module salt.modules.boto_iot*), 1409
describe_policy_version() (*in module salt.modules.boto_iot*), 1409
describe_replication_group() (*in module salt.modules.boto_elasticache*), 1381
describe_replication_groups() (*in module salt.modules.boto3_elasticache*), 1306
describe_reserved_elasticsearch_instance_offerings() (*in module salt.modules.boto3_elasticsearch*), 1315
describe_reserved_elasticsearch_instances() (*in module salt.modules.boto3_elasticsearch*), 1315
describe_role() (*in module salt.modules.boto_iam*), 1398
describe_route_table() (*in module salt.modules.boto_vpc*), 1456
describe_route_tables() (*in module salt.modules.boto_vpc*), 1456
describe_snapshots() (*in module salt.cloud.clouds.ec2*), 1047
describe_subnet() (*in module salt.modules.boto_vpc*), 1456
describe_subnets() (*in module salt.modules.boto_vpc*), 1456
describe_target_health() (*in module salt.modules.boto_elbv2*), 1392
describe_thing_type() (*in module salt.modules.boto_iot*), 1409
describe_topic_rule() (*in module salt.modules.boto_iot*), 1410
describe_usage_plans() (*in module salt.modules.boto_apigateway*), 1338
describe_volumes() (*in module salt.cloud.clouds.ec2*), 1048
describe_vpc_peering_connection() (*in module salt.modules.boto_vpc*), 1457
describe_vpcs() (*in module salt.modules.boto_vpc*), 1457
deserialize() (*in module salt.modules.slsutil*), 2498
deserialize() (*in module salt.serializers.configparser*), 3171
deserialize() (*in module salt.serializers.json*), 3172
deserialize() (*in module salt.serializers.msgpack*), 3172
deserialize() (*in module salt.serializers.toml*), 3173
deserialize() (*in module salt.serializers.yaml*), 3173
deserialize() (*in module salt.serializers.yamlex*), 3175
desktop_interface() (*in module salt.states.gnomedesktop*), 3468
desktop_lockdown() (*in module salt.states.gnomedesktop*), 3469
destroy() (*in module salt.cloud.clouds.aliyun*), 1032
destroy() (*in module salt.cloud.clouds.azurearm*), 1037
destroy() (*in module salt.cloud.clouds.cloudstack*), 1038
destroy() (*in module salt.cloud.clouds.digitalocean*), 1040
destroy() (*in module salt.cloud.clouds.dimensiondata*), 1043
destroy() (*in module salt.cloud.clouds.ec2*), 1048
destroy() (*in module salt.cloud.clouds.gce*), 1056
destroy() (*in module salt.cloud.clouds.gogrid*), 1060
destroy() (*in module salt.cloud.clouds.joyent*), 1062
destroy() (*in module salt.cloud.clouds.linode*), 1067
destroy() (*in module salt.cloud.clouds.lxc*), 1070
destroy() (*in module salt.cloud.clouds.msazure*), 1074
destroy() (*in module salt.cloud.clouds.oneandone*), 1088
destroy() (*in module salt.cloud.clouds.opennebula*), 1091
destroy() (*in module salt.cloud.clouds.openstack*), 1112
destroy() (*in module salt.cloud.clouds.parallels*), 1114
destroy() (*in module salt.cloud.clouds.profitbricks*), 1117
destroy() (*in module salt.cloud.clouds.proxmox*), 1120
destroy() (*in module salt.cloud.clouds.qingcloud*), 1122
destroy() (*in module salt.cloud.clouds.saltify*), 1125

`destroy()` (*in module salt.cloud.clouds.scaleway*), 1126
`destroy()` (*in module salt.cloud.clouds.softlayer*), 1127
`destroy()` (*in module salt.cloud.clouds.softlayer_hw*), 1129
`destroy()` (*in module salt.cloud.clouds.tencentcloud*), 1131
`destroy()` (*in module salt.cloud.clouds.vagrant*), 1134
`destroy()` (*in module salt.cloud.clouds.virtualbox*), 1135
`destroy()` (*in module salt.cloud.clouds.vmware*), 1140
`destroy()` (*in module salt.cloud.clouds.vultrpy*), 1148
`destroy()` (*in module salt.cloud.clouds.xen*), 1150
`destroy()` (*in module salt.modules.cloud*), 1520
`destroy()` (*in module salt.modules.lxc*), 1990
`destroy()` (*in module salt.modules.mdadm_raid*), 2075
`destroy()` (*in module salt.modules.vagrant*), 2634
`destroy()` (*in module salt.modules.vboxmanage*), 2643
`destroy()` (*in module salt.modules.zfs*), 2900
`destroy()` (*in module salt.modules.zpool*), 2918
`destroy()` (*in module salt.runners.cloud*), 3103
`destroy()` (*salt.cloud.CloudClient method*), 3852
`destroy_balancer()` (*in module salt.modules.libcloud_loadbalancer*), 1969
`destroy_dns_records()` (*in module salt.cloud.clouds.digitalocean*), 1041
`destroy_node()` (*in module salt.modules.libcloud_compute*), 1961
`destroy_template()` (*in module salt.cloud.clouds.xen*), 1150
`destroy_vm_vdis()` (*in module salt.cloud.clouds.xen*), 1150
`destroy_volume()` (*in module salt.modules.libcloud_compute*), 1962
`destroy_volume_snapshot()` (*in module salt.modules.libcloud_compute*), 1962
`destroyed()` (*in module salt.states.vagrant*), 3745
`detach()` (*in module salt.modules.bcache*), 1282
`detach()` (*in module salt.modules.zoneadm*), 2908
`detach()` (*in module salt.modules.zpool*), 2918
`detach_disk()` (*in module salt.cloud.clouds.gce*), 1056
`detach_group_policy()` (*in module salt.modules.boto_iam*), 1398
`detach_lb()` (*in module salt.cloud.clouds.gce*), 1057
`detach_network_interface()` (*in module salt.modules.boto_ec2*), 1366
`detach_principal_policy()` (*in module salt.modules.boto_iot*), 1410
`detach_role_policy()` (*in module salt.modules.boto_iam*), 1398
`detach_subnets()` (*in module salt.modules.boto_elb*), 1388
`detach_usage_plan_from_apis()` (*in module salt.modules.boto_apigateway*), 1339
`detach_user_policy()` (*in module salt.modules.boto_iam*), 1399
`detach_volume()` (*in module salt.cloud.clouds.ec2*), 1048
`detach_volume()` (*in module salt.modules.boto_ec2*), 1366
`detach_volume()` (*in module salt.modules.libcloud_compute*), 1962
`detached()` (*in module salt.states.git*), 3457
`detached()` (*in module salt.states.zone*), 3811
`detail()` (*in module salt.modules.mdadm_raid*), 2075
`device()` (*in module salt.modules.bcache*), 1282
`device_exists()` (*in module salt.modules.zenoss*), 2898
`devices()` (*in module salt.modules.btrfs*), 1467
`devices()` (*in module salt.modules.xfs*), 2860
`dfs()` (*in module salt.modules.hadoop*), 1839
`dfs_absent()` (*in module salt.modules.hadoop*), 1839
`dfs_present()` (*in module salt.modules.hadoop*), 1839
`dfsadmin_report()` (*in module salt.modules.hadoop*), 1839
`dhcp_options_absent()` (*in module salt.states.boto_vpc*), 3329
`dhcp_options_exists()` (*in module salt.modules.boto_vpc*), 1457
`dhcp_options_present()` (*in module salt.states.boto_vpc*), 3330
`DICT_OPTIONS_LIST` (*in module salt.modules.solrcloud*), 2533
`DictDiffer` (*class in salt.states.cyg*), 3360
`dictupdate()` (*in module salt.modules.napalm_formula*), 2153
`did_composer_install()` (*in module salt.modules.composer*), 1554
`diff()` (*in module salt.modules.dockermod*), 1637
`diff()` (*in module salt.modules.git*), 1769
`diff()` (*in module salt.modules.junos*), 1914
`diff()` (*in module salt.modules.napalm_yang_mod*), 2189
`diff()` (*in module salt.modules.rpm_lowpkg*), 2447
`diff()` (*in module salt.modules.snapper*), 2545
`diff()` (*in module salt.modules.svn*), 2569
`diff()` (*in module salt.modules.yumpkg*), 2864
`diff()` (*in module salt.modules.zfs*), 2900
`diff()` (*in module salt.modules.zypperpkg*), 2924
`diff()` (*in module salt.runners.survey*), 3155
`diff()` (*in module salt.states.junos*), 3511
`diff_jid()` (*in module salt.modules.snapper*), 2545
`diff_objects()` (*in module salt.modules.infoblox*), 1869
`diff_text()` (*in module salt.modules.iosconfig*), 1881
`diff_tree()` (*in module salt.modules.iosconfig*), 1881
`difference`
  `jinja filters`, 433
`dig()` (*in module salt.modules.network*), 2216

dig() (*in module salt.modules.win_network*), 2773
digest() (*in module salt.modules.hashutil*), 1844
digest_file() (*in module salt.modules.hashutil*), 1844
dir() (*in module salt.modules.temp*), 2599
dir_list() (*in module salt.runners.fileserver*), 3113
directives() (*in module salt.modules.apache*), 1210
directory() (*in module salt.states.etcd_mod*), 3406
directory() (*in module salt.states.file*), 3427
directory_exists() (*in module salt.modules.file*), 1709
dirinfo() (*in module salt.modules.moosefs*), 2091
dirname() (*in module salt.modules.file*), 1709
dirty() (*in module salt.states.svn*), 3730
DirtyPagesWriteCount (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATION attribute*), 2808
DirtyWriteIoCount (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATION attribute*), 2808
disable() (*in module salt.modules.apf*), 1212
disable() (*in module salt.modules.beacons*), 1283
disable() (*in module salt.modules.csf*), 1582
disable() (*in module salt.modules.debian_service*), 1597
disable() (*in module salt.modules.freebsdservice*), 1746
disable() (*in module salt.modules.gentoo_service*), 1754
disable() (*in module salt.modules.mac_service*), 2038
disable() (*in module salt.modules.netbsdservice*), 2203
disable() (*in module salt.modules.nilrt_ip*), 2249
disable() (*in module salt.modules.nspawn*), 2260
disable() (*in module salt.modules.openbsdrcctl_service*), 2272
disable() (*in module salt.modules.pf*), 2336
disable() (*in module salt.modules.puppet*), 2395
disable() (*in module salt.modules.rdp*), 2426
disable() (*in module salt.modules.rh_service*), 2443
disable() (*in module salt.modules.runit*), 2453
disable() (*in module salt.modules.schedule*), 2481
disable() (*in module salt.modules.smf_service*), 2510
disable() (*in module salt.modules.state*), 2550
disable() (*in module salt.modules.systemd_service*), 2592
disable() (*in module salt.modules.upstart_service*), 2627
disable() (*in module salt.modules.win_firewall*), 2746
disable() (*in module salt.modules.win_ip*), 2764
disable() (*in module salt.modules.win_service*), 2796
disable() (*in module salt.states.modjk_worker*), 3554
disable_api_key() (*in module salt.modules.boto_apigateway*), 1339
disable_auto_login() (*in module salt.modules.mac_user*), 2057
disable_availability_zones() (*in module salt.modules.boto_elb*), 1388
disable_beacon() (*in module salt.modules.beacons*), 1284
disable_detailed_monitoring() (*in module salt.cloud.clouds.ec2*), 1048
disable_dhcp() (*in module salt.modules.ilo*), 1853
disable_enhanced_monitoring() (*in module salt.modules.boto_kinesis*), 1413
disable_inheritance() (*in module salt.modules.win_dacl*), 2722
disable_job() (*in module salt.modules.jenkinsmod*), 1909
disable_job() (*in module salt.modules.schedule*), 2482
disable_key() (*in module salt.modules.boto_kms*), 1416
disable_key_rotation() (*in module salt.modules.boto_kms*), 1416
disable_modules
    conf/minion, 144
disable_plugin() (*in module salt.modules.rabbitmq*), 2414
disable_quota_volume() (*in module salt.modules.glusterfs*), 1812
disable_returners
    conf/minion, 144
disable_server() (*in module salt.modules.haproxyconn*), 1840
disable_source() (*in module salt.modules.chocolatey*), 1500
disable_ssh() (*in module salt.modules.ilo*), 1853
disable_task() (*in module salt.modules.kapacitor*), 1923
disable_term_protect() (*in module salt.cloud.clouds.ec2*), 1048
disable_testing_mode() (*in module salt.modules.csf*), 1582
disabled() (*in module salt.modules.daemontools*), 1585
disabled() (*in module salt.modules.debian_service*), 1598
disabled() (*in module salt.modules.freebsdservice*), 1747
disabled() (*in module salt.modules.gentoo_service*), 1754
disabled() (*in module salt.modules.launchctl_service*), 1951
disabled() (*in module salt.modules.mac_service*), 2038
disabled() (*in module salt.modules.netbsdservice*), 2203

disabled() (*in module salt.modules.openbsdrcctl_service*), 2273
disabled() (*in module salt.modules.openbsdservice*), 2274
disabled() (*in module salt.modules.rh_service*), 2443
disabled() (*in module salt.modules.runit*), 2453
disabled() (*in module salt.modules.smf_service*), 2510
disabled() (*in module salt.modules.systemd_service*), 2592
disabled() (*in module salt.modules.upstart_service*), 2627
disabled() (*in module salt.modules.win_service*), 2796
disabled() (*in module salt.states.apache_conf*), 3187
disabled() (*in module salt.states.apache_module*), 3187
disabled() (*in module salt.states.apache_site*), 3188
disabled() (*in module salt.states.beacon*), 3222
disabled() (*in module salt.states.rabbitmq_plugin*), 3682
disabled() (*in module salt.states.rdp*), 3688
disabled() (*in module salt.states.schedule*), 3706
disabled() (*in module salt.states.service*), 3711
disabled() (*in module salt.states.win_firewall*), 3762
disassociate_api_key_stagekeys() (*in module salt.modules.boto_apigateway*), 1339
disassociate_eip_address() (*in module salt.modules.boto_ec2*), 1366
disassociate_network_acl() (*in module salt.modules.boto_vpc*), 1458
disassociate_profile_from_role() (*in module salt.modules.boto_iam*), 1399
disassociate_route_table() (*in module salt.modules.boto_vpc*), 1458
disassociate_vpc_from_hosted_zone() (*in module salt.modules.boto3_route53*), 1326
disbatch() (*salt.netapi.rest_tornado.saltnado.SaltAPIHandler method*), 2967
discard_config() (*in module salt.modules.napalm_network*), 2161
discard_local_changes() (*in module salt.modules.git*), 1770
disconnect() (*in module salt.modules.vsphere*), 2691
disconnect_all_containers_from_network() (*in module salt.modules.dockermod*), 1637
disconnect_container_from_network() (*in module salt.modules.dockermod*), 1637
disconnect_host() (*in module salt.cloud.clouds.vmware*), 1140
disconnect_session() (*in module salt.modules.rdp*), 2426
discoverable() (*in module salt.modules.bluez_bluetooth*), 1300
disinherit() (*in module salt.states.win_dacl*), 3758
disk_io_counters() (*in module salt.modules.ps*), 2390
disk_partition_usage() (*in module salt.modules.ps*), 2390
disk_partitions() (*in module salt.modules.ps*), 2390
disk_set() (*in module salt.modules.parted_partition*), 2325
disk_toggle() (*in module salt.modules.parted_partition*), 2325
disk_usage() (*in module salt.modules.ps*), 2390
diskgroups_configured() (*in module salt.states.esxi*), 3410
disks() (*in module salt.grains.disks*), 1179
diskstats() (*in module salt.modules.status*), 2559
diskusage() (*in module salt.modules.file*), 1710
diskusage() (*in module salt.modules.status*), 2559
diskusage() (*in module salt.modules.win_status*), 2810
display() (*in module salt.modules.alternatives*), 1207
display() (*salt.output.nested.NestDisplay method*), 2982
display() (*salt.output.no_return.NestDisplay method*), 2984
display() (*salt.output.table_out.TableDisplay method*), 2988
display_rows() (*salt.output.table_out.TableDisplay method*), 2988
dns() (*in module salt.grains.core*), 1178
dns_check
    jinja filters, 446
dns_dhcp() (*in module salt.modules.win_dns_client*), 2728
dns_dhcp() (*in module salt.states.win_dns_client*), 3761
dns_exists() (*in module salt.states.win_dns_client*), 3761
do() (*in module salt.modules.pyenv*), 2409
do() (*in module salt.modules.rbenv*), 2424
do() (*in module salt.modules.rvm*), 2456
do_with_python() (*in module salt.modules.pyenv*), 2409
do_with_ruby() (*in module salt.modules.rbenv*), 2424
doc() (*in module salt.modules.sysmod*), 2582
docker.compare_container_networks
    conf/minion, 143
docker.update_mine
    conf/minion, 143
docker_to_uuid() (*in module salt.modules.smartos_imgadm*), 2500
DockerJSONDecoder (*class in salt.modules.dockermod*), 1618
document_create() (*in module salt.modules.elasticsearch*), 1683
document_delete() (*in module

salt.modules.elasticsearch), 1684
document_exists() (*in module salt.modules.elasticsearch*), 1684
document_get() (*in module salt.modules.elasticsearch*), 1684
domain_capabilities() (*in module salt.modules.virt*), 2649
domain_create() (*in module salt.modules.keystoneng*), 1937
domain_delete() (*in module salt.modules.keystoneng*), 1937
domain_get() (*in module salt.modules.keystoneng*), 1937
domain_list() (*in module salt.modules.keystoneng*), 1937
domain_search() (*in module salt.modules.keystoneng*), 1937
domain_update() (*in module salt.modules.keystoneng*), 1937
dot_vals() (*in module salt.modules.config*), 1556
down() (*in module salt.modules.debian_ip*), 1596
down() (*in module salt.modules.linux_ip*), 1976
down() (*in module salt.modules.nilrt_ip*), 2249
down() (*in module salt.modules.rh_ip*), 2442
down() (*in module salt.runners.manage*), 3122
download() (*in module salt.modules.mac_softwareupdate*), 2045
download() (*in module salt.modules.win_wua*), 2840
download() (*in module salt.modules.yumpkg*), 2864
download() (*in module salt.modules.zypperpkg*), 2925
download_all() (*in module salt.modules.mac_softwareupdate*), 2045
download_antivirus() (*in module salt.modules.panos*), 2306
download_object() (*in module salt.modules.libcloud_storage*), 1973
download_software() (*in module salt.states.panos*), 3624
download_software_file() (*in module salt.modules.panos*), 2306
download_software_version() (*in module salt.modules.panos*), 2307
downloaded() (*in module salt.states.artifactory*), 3195
downloaded() (*in module salt.states.pkg*), 3643
drop_continuous_query() (*in module salt.modules.influxdbmod*), 1859
drop_db() (*in module salt.modules.influxdbmod*), 1859
drop_extension() (*in module salt.modules.postgres*), 2376
drop_keyspace() (*in module salt.modules.cassandra_cql*), 1486
drop_retention_policy() (*in module salt.modules.influxdbmod*), 1859
dump() (*in module salt.modules.data*), 1587
dump() (*in module salt.modules.disk*), 1602
dump() (*in module salt.modules.extfs*), 1701
dump() (*in module salt.modules.xfs*), 2860
dump_config() (*in module salt.modules.modjk*), 2084
dumpconf() (*in module salt.modules.znc*), 2907

E

EAuth, **5401**
eauth_acl_module
  conf/master, 71
EauthAuthenticationError, 3998
ec2_credentials_create() (*in module salt.modules.keystone*), 1930
ec2_credentials_delete() (*in module salt.modules.keystone*), 1930
ec2_credentials_get() (*in module salt.modules.keystone*), 1930
ec2_credentials_list() (*in module salt.modules.keystone*), 1930
echo() (*in module salt.modules.test*), 2601
eclean_dist() (*in module salt.modules.gentoolkitmod*), 1756
eclean_pkg() (*in module salt.modules.gentoolkitmod*), 1757
edit_conf() (*in module salt.modules.lxc*), 1991
edit_config() (*in module salt.states.panos*), 3625
edit_repo() (*in module salt.modules.github*), 1801
edit_server() (*in module salt.modules.lvs*), 1985
edit_service() (*in module salt.modules.lvs*), 1985
edit_task() (*in module salt.modules.win_task*), 2827
edit_team() (*in module salt.modules.github*), 1801
edited_conf() (*in module salt.states.lxc*), 3545
email_alerts() (*in module salt.modules.drac*), 1663
email_alerts() (*in module salt.modules.dracr*), 1666
emerge_default_opts_contains() (*in module salt.modules.makeconf*), 2062
empty() (*in module salt.modules.boto_s3_bucket*), 1437
empty_dir_list() (*in module salt.runners.fileserver*), 3113
enable() (*in module salt.modules.apf*), 1212
enable() (*in module salt.modules.beacons*), 1284
enable() (*in module salt.modules.csf*), 1582
enable() (*in module salt.modules.debian_service*), 1598
enable() (*in module salt.modules.freebsdservice*), 1747
enable() (*in module salt.modules.gentoo_service*), 1754
enable() (*in module salt.modules.mac_assistive*), 2022
enable() (*in module salt.modules.mac_service*), 2039
enable() (*in module salt.modules.netbsdservice*), 2203
enable() (*in module salt.modules.nilrt_ip*), 2249
enable() (*in module salt.modules.nspawn*), 2260
enable() (*in module salt.modules.openbsdrcctl_service*), 2273
enable() (*in module salt.modules.pf*), 2337
enable() (*in module salt.modules.puppet*), 2395

`enable()` (*in module salt.modules.rdp*), 2426
`enable()` (*in module salt.modules.rh_service*), 2443
`enable()` (*in module salt.modules.runit*), 2453
`enable()` (*in module salt.modules.schedule*), 2482
`enable()` (*in module salt.modules.smf_service*), 2511
`enable()` (*in module salt.modules.state*), 2550
`enable()` (*in module salt.modules.systemd_service*), 2593
`enable()` (*in module salt.modules.upstart_service*), 2628
`enable()` (*in module salt.modules.win_firewall*), 2746
`enable()` (*in module salt.modules.win_ip*), 2764
`enable()` (*in module salt.modules.win_service*), 2796
`enable_api_key()` (*in module salt.modules.boto_apigateway*), 1339
`enable_auto_login()` (*in module salt.modules.mac_user*), 2058
`enable_availability_zones()` (*in module salt.modules.boto_elb*), 1388
`enable_beacon()` (*in module salt.modules.beacons*), 1284
`enable_detailed_monitoring()` (*in module salt.cloud.clouds.ec2*), 1048
`enable_dhcp()` (*in module salt.modules.ilo*), 1853
`enable_enhanced_monitoring()` (*in module salt.modules.boto_kinesis*), 1413
`enable_firewall_ruleset()` (*in module salt.modules.vsphere*), 2691
`enable_gpu_grains`
  `conf/master`, 57
  `conf/minion`, 136
`enable_inheritance()` (*in module salt.modules.win_dacl*), 2722
`enable_job()` (*in module salt.modules.jenkinsmod*), 1909
`enable_job()` (*in module salt.modules.schedule*), 2482
`enable_key()` (*in module salt.modules.boto_kms*), 1416
`enable_key_rotation()` (*in module salt.modules.boto_kms*), 1416
`enable_legacy_startup_events`
  `conf/minion`, 126
`enable_plugin()` (*in module salt.modules.rabbitmq*), 2414
`enable_quota_volume()` (*in module salt.modules.glusterfs*), 1812
`enable_server()` (*in module salt.modules.haproxyconn*), 1840
`enable_source()` (*in module salt.modules.chocolatey*), 1501
`enable_ssh()` (*in module salt.modules.ilo*), 1853
`enable_task()` (*in module salt.modules.kapacitor*), 1923
`enable_term_protect()` (*in module salt.cloud.clouds.ec2*), 1048
`enable_testing_mode()` (*in module salt.modules.csf*), 1582
`enable_whitelist_modules`
  `conf/minion`, 144
`enable_zip_modules`
  `conf/minion`, 146
`enabled()` (*in module salt.modules.daemontools*), 1585
`enabled()` (*in module salt.modules.debian_service*), 1598
`enabled()` (*in module salt.modules.dummyproxy_service*), 1674
`enabled()` (*in module salt.modules.freebsdservice*), 1747
`enabled()` (*in module salt.modules.gentoo_service*), 1754
`enabled()` (*in module salt.modules.launchctl_service*), 1951
`enabled()` (*in module salt.modules.mac_assistive*), 2022
`enabled()` (*in module salt.modules.mac_service*), 2039
`enabled()` (*in module salt.modules.netbsdservice*), 2203
`enabled()` (*in module salt.modules.openbsdrcctl_service*), 2273
`enabled()` (*in module salt.modules.openbsdservice*), 2275
`enabled()` (*in module salt.modules.rest_service*), 2439
`enabled()` (*in module salt.modules.rh_service*), 2444
`enabled()` (*in module salt.modules.runit*), 2453
`enabled()` (*in module salt.modules.smf_service*), 2511
`enabled()` (*in module salt.modules.ssh_service*), 2542
`enabled()` (*in module salt.modules.systemd_service*), 2593
`enabled()` (*in module salt.modules.upstart_service*), 2628
`enabled()` (*in module salt.modules.win_service*), 2796
`enabled()` (*in module salt.states.apache_conf*), 3187
`enabled()` (*in module salt.states.apache_module*), 3187
`enabled()` (*in module salt.states.apache_site*), 3188
`enabled()` (*in module salt.states.beacon*), 3222
`enabled()` (*in module salt.states.rabbitmq_plugin*), 3682
`enabled()` (*in module salt.states.rdp*), 3688
`enabled()` (*in module salt.states.schedule*), 3706
`enabled()` (*in module salt.states.service*), 3711
`enabled()` (*in module salt.states.win_firewall*), 3763
`enabled_service_owners()` (*in module salt.modules.introspect*), 1880
`enc()` (*in module salt.modules.nacl*), 2109
`enc()` (*in module salt.runners.nacl*), 3129
`enc_file()` (*in module salt.modules.nacl*), 2109
`enc_file()` (*in module salt.runners.nacl*), 3129
`encrypt()` (*in module salt.modules.boto_kms*), 1416
`encrypt()` (*in module salt.modules.gpg*), 1819

endpoint_absent() (*in module salt.states.keystone*), 3518
endpoint_create() (*in module salt.modules.keystone*), 1930
endpoint_create() (*in module salt.modules.keystoneng*), 1938
endpoint_delete() (*in module salt.modules.keystone*), 1931
endpoint_delete() (*in module salt.modules.keystoneng*), 1938
endpoint_get() (*in module salt.modules.keystone*), 1931
endpoint_get() (*in module salt.modules.keystoneng*), 1938
endpoint_list() (*in module salt.modules.keystone*), 1931
endpoint_list() (*in module salt.modules.keystoneng*), 1938
endpoint_present() (*in module salt.states.keystone*), 3518
endpoint_search() (*in module salt.modules.keystoneng*), 1938
endpoint_update() (*in module salt.modules.keystoneng*), 1938
enforce_mine_cache
    conf/master, 61
enforce_nice_config() (*in module salt.modules.portage_config*), 2370
eni_absent() (*in module salt.states.boto_ec2*), 3274
eni_present() (*in module salt.states.boto_ec2*), 3274
ensure_path() (*in module salt.modules.zookeeper*), 2914
ensure_views() (*in module salt.returners.couchdb_return*), 318
enter_config_mode() (*in module salt.modules.netmiko_mod*), 2208
enter_maintenance_mode() (*in module salt.cloud.clouds.vmware*), 1140
enter_root() (*salt.pillar.sql_base.SqlBaseExtPillar method*), 3034
enter_standby() (*in module salt.modules.boto_asg*), 1342
env() (*in module salt.modules.udev*), 2626
env_absent() (*in module salt.states.cron*), 3357
env_order
    conf/master, 76
    conf/minion, 148
env_present() (*in module salt.states.cron*), 3357
Environment, **5401**
environment
    conf/minion, 151
environment() (*in module salt.modules.napalm_network*), 2161
EnvLoader (*class in salt.modules.inspectlib*), 1876
envs() (*in module salt.modules.cp*), 1575
envs() (*in module salt.runners.fileserver*), 3113
eq() (*in module salt.thorium.check*), 3821
equal() (*in module salt.modules.salt_version*), 2463
equals() (*in module salt.modules.grains*), 1831
equalto
    jinja filters, 449
erase_disk_partitions() (*in module salt.modules.vsphere*), 2692
error() (*in module salt.modules.logmod*), 1982
error() (*in module salt.runners.error*), 3108
error() (*in module salt.wheel.error*), 3838
escaping-jinja
    jinja filters, 451
estimate() (*in module salt.modules.xfs*), 2860
esxcli_cmd() (*in module salt.modules.vsphere*), 2692
esxi() (*in module salt.grains.esxi*), 1179
etc_hosts() (*in module salt.modules.osquery*), 2289
etc_services() (*in module salt.modules.osquery*), 2289
Event, **5401**
Event, **669**, *see* Reactor
Event (*class in salt.engines.ircbot*), 1156
event bus, 669
event system, 669
event() (*in module salt.modules.state*), 2550
event() (*in module salt.runners.state*), 3152
event() (*in module salt.thorium.check*), 3821
event_bus_context() (*in module salt.engines.logentries*), 1159
event_bus_context() (*in module salt.engines.logstash_engine*), 1160
event_bus_context() (*in module salt.engines.test*), 1170
event_fire() (*in module salt.modules.consul*), 1568
event_list() (*in module salt.modules.consul*), 1568
event_return
    conf/master, 59
event_return() (*in module salt.returners.carbon_return*), 313
event_return() (*in module salt.returners.cassandra_cql_return*), 315
event_return() (*in module salt.returners.elasticsearch_return*), 320
event_return() (*in module salt.returners.local*), 325
event_return() (*in module salt.returners.mattermost_returner*), 326
event_return() (*in module salt.returners.mongo_future_return*), 329
event_return() (*in module salt.returners.mysql*), 333
event_return() (*in module salt.returners.pgjsonb*), 339
event_return() (*in module salt.returners.postgres*), 342

event_return() (*in module salt.returners.postgres_local_cache*), 344
event_return() (*in module salt.returners.rawfile_json*), 346
event_return() (*in module salt.returners.smtp_return*), 352
event_return_blacklist
  conf/master, 60
event_return_queue
  conf/master, 59
event_return_whitelist
  conf/master, 59
event_source_mapping_absent() (*in module salt.states.boto_lambda*), 3307
event_source_mapping_exists() (*in module salt.modules.boto_lambda*), 1422
event_source_mapping_present() (*in module salt.states.boto_lambda*), 3307
Events (*class in salt.netapi.rest_cherrypy.app*), 2951
EventsSaltAPIHandler (*class in salt.netapi.rest_tornado.saltnado*), 2974
ex_mod_init() (*in module salt.modules.ebuildpkg*), 1676
exactly_n_true
  jinja filters, 430
exactly_one_true
  jinja filters, 430
examine() (*in module salt.modules.mdadm_raid*), 2075
exception() (*in module salt.modules.logmod*), 1982
exception() (*in module salt.modules.test*), 2601
exec_() (*in module salt.modules.parallels*), 2321
exec_action() (*in module salt.modules.eselect*), 1694
exec_code() (*in module salt.modules.cmdmod*), 1523
exec_code_all() (*in module salt.modules.cmdmod*), 1523
execs() (*in module salt.modules.systemd_service*), 2593
execute() (*in module salt.executors.direct_call*), 1171
execute() (*in module salt.executors.docker*), 1171
execute() (*in module salt.executors.splay*), 1171
execute() (*in module salt.executors.sudo*), 1172
execute() (*in module salt.modules.augeas_cfg*), 1244
execute() (*in module salt.runners.salt*), 3143
execute_salt_restart_task() (*in module salt.modules.win_service*), 2797
Execution Function, **5401**
Execution Module, **5401**
execution() (*in module salt.runners.doc*), 3107
exists() (*in module salt.modules.boto3_elasticsearch*), 1316
exists() (*in module salt.modules.boto_asg*), 1342
exists() (*in module salt.modules.boto_cfn*), 1345
exists() (*in module salt.modules.boto_cloudtrail*), 1349
exists() (*in module salt.modules.boto_cloudwatch_event*), 1354
exists() (*in module salt.modules.boto_dynamodb*), 1360
exists() (*in module salt.modules.boto_ec2*), 1367
exists() (*in module salt.modules.boto_elasticache*), 1381
exists() (*in module salt.modules.boto_elasticsearch_domain*), 1384
exists() (*in module salt.modules.boto_elb*), 1388
exists() (*in module salt.modules.boto_kinesis*), 1413
exists() (*in module salt.modules.boto_rds*), 1427
exists() (*in module salt.modules.boto_s3_bucket*), 1437
exists() (*in module salt.modules.boto_secgroup*), 1442
exists() (*in module salt.modules.boto_sns*), 1444
exists() (*in module salt.modules.boto_sqs*), 1446
exists() (*in module salt.modules.boto_vpc*), 1458
exists() (*in module salt.modules.csf*), 1582
exists() (*in module salt.modules.dockermod*), 1638
exists() (*in module salt.modules.lxc*), 1991
exists() (*in module salt.modules.nspawn*), 2260
exists() (*in module salt.modules.parallels*), 2321
exists() (*in module salt.modules.parted_partition*), 2325
exists() (*in module salt.modules.redismod*), 2428
exists() (*in module salt.modules.smartos_nictagadm*), 2502
exists() (*in module salt.modules.win_path*), 2776
exists() (*in module salt.modules.zfs*), 2900
exists() (*in module salt.modules.zookeeper*), 2914
exists() (*in module salt.modules.zpool*), 2918
exists() (*in module salt.states.aws_sqs*), 3199
exists() (*in module salt.states.file*), 3429
exists() (*in module salt.states.grains*), 3480
exists() (*in module salt.states.mac_xattr*), 3550
exists() (*in module salt.states.win_path*), 3774
exit_config_mode() (*in module salt.modules.netmiko_mod*), 2209
exit_maintenance_mode() (*in module salt.cloud.clouds.vmware*), 1140
exit_standby() (*in module salt.modules.boto_asg*), 1343
exit_success() (*in module salt.runners.jobs*), 3116
expand_repo_def() (*in module salt.modules.aptpkg*), 1217
expire() (*in module salt.modules.redismod*), 2428
expireat() (*in module salt.modules.redismod*), 2428
expired() (*in module salt.modules.x509*), 2850
expires() (*in module salt.modules.acme*), 1201
export() (*in module salt.modules.dockermod*), 1638
export() (*in module salt.modules.inspector*), 1879
export() (*in module salt.modules.svn*), 2569

export() (*in module salt.modules.zonecfg*), 2911
export() (*in module salt.modules.zpool*), 2919
export() (*in module salt.states.svn*), 3730
export() (*in module salt.states.zone*), 3811
export() (*salt.modules.inspectlib.collector.Inspector method*), 1874
export() (*salt.modules.inspectlib.kiwiproc.KiwiExporter method*), 1878
export_cert() (*in module salt.modules.win_pki*), 2785
export_distributions() (*in module salt.modules.boto_cloudfront*), 1347
export_key() (*in module salt.modules.gpg*), 1819
export_roles() (*in module salt.modules.boto_iam*), 1399
export_users() (*in module salt.modules.boto_iam*), 1399
exportdb() (*in module salt.modules.udev*), 2626
ext() (*in module salt.modules.pillar*), 2338
ext_job_cache
    conf/master, 59
ext_pillar
    conf/master, 101
ext_pillar() (*in module salt.pillar.cmd_json*), 2991
ext_pillar() (*in module salt.pillar.cmd_yaml*), 2991
ext_pillar() (*in module salt.pillar.cmd_yamlex*), 2991
ext_pillar() (*in module salt.pillar.cobbler*), 2992
ext_pillar() (*in module salt.pillar.confidant*), 2992
ext_pillar() (*in module salt.pillar.consul_pillar*), 2994
ext_pillar() (*in module salt.pillar.csvpillar*), 2995
ext_pillar() (*in module salt.pillar.digicert*), 2995
ext_pillar() (*in module salt.pillar.django_orm*), 2997
ext_pillar() (*in module salt.pillar.ec2_pillar*), 2998
ext_pillar() (*in module salt.pillar.etcd_pillar*), 2999
ext_pillar() (*in module salt.pillar.file_tree*), 3001
ext_pillar() (*in module salt.pillar.foreman*), 3003
ext_pillar() (*in module salt.pillar.git_pillar*), 3008
ext_pillar() (*in module salt.pillar.gpg*), 3008
ext_pillar() (*in module salt.pillar.hg_pillar*), 3009
ext_pillar() (*in module salt.pillar.hiera*), 3009
ext_pillar() (*in module salt.pillar.http_json*), 3010
ext_pillar() (*in module salt.pillar.http_yaml*), 3011
ext_pillar() (*in module salt.pillar.libvirt*), 3011
ext_pillar() (*in module salt.pillar.makostack*), 3017
ext_pillar() (*in module salt.pillar.mongo*), 3018
ext_pillar() (*in module salt.pillar.mysql*), 3019
ext_pillar() (*in module salt.pillar.netbox*), 3020
ext_pillar() (*in module salt.pillar.neutron*), 3020
ext_pillar() (*in module salt.pillar.nodegroups*), 3021
ext_pillar() (*in module salt.pillar.pepa*), 3026
ext_pillar() (*in module salt.pillar.pillar_ldap*), 3028
ext_pillar() (*in module salt.pillar.postgres*), 3029
ext_pillar() (*in module salt.pillar.puppet*), 3029
ext_pillar() (*in module salt.pillar.reclass_adapter*), 3029
ext_pillar() (*in module salt.pillar.redismod*), 3030
ext_pillar() (*in module salt.pillar.s3*), 3031
ext_pillar() (*in module salt.pillar.saltclass*), 3032
ext_pillar() (*in module salt.pillar.sqlcipher*), 3036
ext_pillar() (*in module salt.pillar.sqlite3*), 3037
ext_pillar() (*in module salt.pillar.stack*), 3043
ext_pillar() (*in module salt.pillar.svn_pillar*), 3044
ext_pillar() (*in module salt.pillar.varstack_pillar*), 3044
ext_pillar() (*in module salt.pillar.vault*), 3046
ext_pillar() (*in module salt.pillar.venafi*), 3046
ext_pillar() (*in module salt.pillar.virtkey*), 3046
ext_pillar() (*in module salt.pillar.vmware_pillar*), 3048
ext_pillar_first
    conf/master, 101
extend_dict_key_value
    jinja filters, 439
extension_modules
    conf/master, 53
External Job Cache, **5401**
External Pillar, **5401**
external_auth
    conf/master, 70
extmod_blacklist
    conf/master, 53
    conf/minion, 146
extmod_whitelist
    conf/master, 53
    conf/minion, 146
extra() (*in module salt.modules.libcloud_compute*), 1962
extra() (*in module salt.modules.libcloud_dns*), 1966
extra() (*in module salt.modules.libcloud_loadbalancer*), 1970
extra() (*in module salt.modules.libcloud_storage*), 1973
extra_action() (*salt.cloud.CloudClient method*), 3852
extract() (*in module salt.modules.textfsm_mod*), 2605
extract_hash() (*in module salt.modules.file*), 1710
extract_index() (*in module salt.modules.boto_dynamodb*), 1361
extract_ipv4() (*in module salt.roster.cloud*), 3092
extract_queries() (*salt.pillar.mysql.MySQLExtPillar method*), 3019
extract_queries() (*salt.pillar.postgres.POSTGRESExtPillar method*), 3029
extract_queries() (*salt.pillar.sql_base.SqlBaseExtPillar method*),

3034
extract_war_version() (*in module salt.modules.tomcat*), 2619
extracted() (*in module salt.states.archive*), 3188

F

F5Mgmt (*class in salt.runners.f5*), 3110
fact() (*in module salt.modules.puppet*), 2395
facts() (*in module salt.grains.junos*), 1180
facts() (*in module salt.modules.junos*), 1914
facts() (*in module salt.modules.napalm_network*), 2162
facts() (*in module salt.modules.puppet*), 2395
facts_refresh() (*in module salt.modules.junos*), 1914
fail_with_changes() (*in module salt.states.test*), 3734
fail_without_changes() (*in module salt.states.test*), 3735
failhard
    conf/master, 79
    conf/minion, 163
false_() (*in module salt.modules.test*), 2601
fast_connect_test() (*in module salt.modules.ipmi*), 1884
faulty() (*in module salt.modules.solaris_fmadm*), 2514
fcontext_add_or_delete_policy() (*in module salt.modules.selinux*), 2487
fcontext_add_policy() (*in module salt.modules.selinux*), 2487
fcontext_apply_policy() (*in module salt.modules.selinux*), 2488
fcontext_delete_policy() (*in module salt.modules.selinux*), 2488
fcontext_get_policy() (*in module salt.modules.selinux*), 2488
fcontext_policy_absent() (*in module salt.states.selinux*), 3707
fcontext_policy_applied() (*in module salt.states.selinux*), 3707
fcontext_policy_is_applied() (*in module salt.modules.selinux*), 2489
fcontext_policy_present() (*in module salt.states.selinux*), 3707
feature_installed() (*in module salt.states.win_dism*), 3759
feature_removed() (*in module salt.states.win_dism*), 3759
features() (*in module salt.modules.btrfs*), 1467
features_contains() (*in module salt.modules.makeconf*), 2062
fetch() (*in module salt.cache.consul*), 1026
fetch() (*in module salt.cache.etcd_cache*), 1027
fetch() (*in module salt.cache.localfs*), 1028
fetch() (*in module salt.cache.mysql_cache*), 1028
fetch() (*in module salt.cache.redis_cache*), 1031
fetch() (*in module salt.modules.freebsd_update*), 1738
fetch() (*in module salt.modules.git*), 1770
fetch() (*in module salt.modules.grains*), 1832
fetch() (*in module salt.modules.pillar*), 2339
fetch() (*in module salt.modules.pkgng*), 2356
fetch() (*in module salt.modules.sqlite3*), 2538
fetch() (*in module salt.runners.cache*), 3101
fetch() (*salt.pillar.sql_base.SqlBaseExtPillar method*), 3035
fetch_license() (*in module salt.modules.panos*), 2307
fetch_tree() (*in module salt.pillar.consul_pillar*), 2994
fib() (*in module salt.modules.test*), 2601
fibre_channel_grains
    conf/minion, 134
field_names (*salt.pillar.sql_base.SqlBaseExtPillar attribute*), 3035
File Server, **5401**
file() (*in module salt.modules.temp*), 2600
file() (*in module salt.states.cron*), 3357
file_() (*in module salt.modules.osquery*), 2290
file_buffer_size
    conf/master, 83
file_changes() (*in module salt.modules.osquery*), 2290
file_client
    conf/minion, 152
file_copy() (*in module salt.modules.junos*), 1915
file_copy() (*in module salt.states.junos*), 3511
file_dict() (*in module salt.modules.apkpkg*), 1213
file_dict() (*in module salt.modules.aptpkg*), 1217
file_dict() (*in module salt.modules.dpkg_lowpkg*), 1661
file_dict() (*in module salt.modules.freebsdpkg*), 1742
file_dict() (*in module salt.modules.opkg*), 2281
file_dict() (*in module salt.modules.pacmanpkg*), 2297
file_dict() (*in module salt.modules.pkgin*), 2351
file_dict() (*in module salt.modules.rpm_lowpkg*), 2447
file_dict() (*in module salt.modules.yumpkg*), 2864
file_dict() (*in module salt.modules.zypperpkg*), 2925
file_exists() (*in module salt.modules.file*), 1710
file_exists() (*in module salt.modules.pillar*), 2340
file_hashsum
    jinja filters, 448
file_ignore_glob
    conf/master, 83
file_ignore_regex
    conf/master, 83

file_list() (*in module salt.modules.apkpkg*), 1213
file_list() (*in module salt.modules.aptpkg*), 1217
file_list() (*in module salt.modules.dpkg_lowpkg*), 1661
file_list() (*in module salt.modules.freebsdpkg*), 1742
file_list() (*in module salt.modules.opkg*), 2281
file_list() (*in module salt.modules.pacmanpkg*), 2297
file_list() (*in module salt.modules.pkgin*), 2351
file_list() (*in module salt.modules.rpm_lowpkg*), 2447
file_list() (*in module salt.modules.yumpkg*), 2864
file_list() (*in module salt.modules.zypperpkg*), 2925
file_list() (*in module salt.runners.fileserver*), 3114
file_present() (*in module salt.states.libcloud_storage*), 3538
file_query() (*in module salt.modules.mysql*), 2101
file_recv
    conf/master, 71
file_recv_max_size
    conf/master, 72
    conf/minion, 156
file_roots
    conf/master, 84
    conf/minion, 152
fileinfo() (*in module salt.modules.moosefs*), 2091
fileio() (*in module salt.modules.sysbench*), 2575
FileLockError, 3998
files
    spm command line option, 991
fileserver_backend
    conf/master, 81
fileserver_followsymlinks
    conf/master, 81
    conf/minion, 153
fileserver_ignoresymlinks
    conf/master, 81
    conf/minion, 153
fileserver_limit_traversal
    conf/master, 82
    conf/minion, 153
fileserver_list_cache_time
    conf/master, 82
fileserver_verify_config
    conf/master, 82
FileserverConfigError, 3998
filesystem_absent() (*in module salt.states.zfs*), 3807
filesystem_present() (*in module salt.states.zfs*), 3808
filesystems() (*in module salt.modules.mount*), 2092
filetype_id_to_string() (*in module salt.modules.selinux*), 2489
filter() (*in module salt.states.netacl*), 3570
filter() (*salt.modules.xmpp.SleekXMPPMUC method*), 2863
filter_() (*in module salt.modules.netbox*), 2200
filter_by() (*in module salt.modules.grains*), 1832
filter_by() (*in module salt.modules.match*), 2071
filter_by() (*in module salt.modules.pillar*), 2340
filter_flags() (*in module salt.modules.portage_config*), 2371
filter_lines() (*in module salt.modules.ciscoconfparse_mod*), 1514
find() (*in module salt.modules.file*), 1710
find() (*in module salt.modules.mongodb*), 2087
find() (*in module salt.proxy.nxos*), 3075
find() (*in module salt.runners.net*), 3131
find() (*in module salt.wheel.file_roots*), 3838
find() (*in module salt.wheel.pillar_roots*), 3842
find_cached_job() (*in module salt.modules.saltutil*), 2470
find_credentials() (*in module salt.proxy.esxi*), 3061
find_credentials() (*in module salt.proxy.fx2*), 3064
find_device() (*in module salt.modules.zenoss*), 2898
find_guest() (*in module salt.runners.lxc*), 3119
find_guests() (*in module salt.runners.lxc*), 3119
find_hosted_zone() (*in module salt.modules.boto3_route53*), 1327
find_images() (*in module salt.modules.boto_ec2*), 1367
find_instances() (*in module salt.modules.boto_ec2*), 1367
find_interfaces() (*in module salt.modules.bridge*), 1464
find_job() (*in module salt.modules.saltutil*), 2470
find_lines() (*in module salt.modules.ciscoconfparse_mod*), 1515
find_lines_w_child() (*in module salt.modules.ciscoconfparse_mod*), 1515
find_lines_wo_child() (*in module salt.modules.ciscoconfparse_mod*), 1516
find_objects() (*in module salt.modules.ciscoconfparse_mod*), 1516
find_objects_w_child() (*in module salt.modules.ciscoconfparse_mod*), 1517
find_objects_wo_child() (*in module salt.modules.ciscoconfparse_mod*), 1517
find_room() (*in module salt.modules.slack_notify*), 2497
find_user() (*in module salt.modules.slack_notify*), 2497
findarp() (*in module salt.runners.net*), 3132
findmac() (*in module salt.runners.net*), 3132
finger() (*in module salt.modules.key*), 1928

`finger()` (*in module salt.wheel.key*), 3840
`finger_master()` (*in module salt.modules.key*), 1928
`finger_master()` (*in module salt.wheel.key*), 3840
`fire()` (*in module salt.modules.event*), 1700
`fire()` (*salt.engines.slack.SlackClient method*), 1167
`fire_master()` (*in module salt.modules.event*), 1700
`firefox_addons()` (*in module salt.modules.osquery*), 2290
`firmware_update()` (*in module salt.states.dellchassis*), 3368
`FirstLevelTbFills` (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATION attribute*), 2808
`fix_outage()` (*in module salt.modules.rest_sample_utils*), 2439
`fix_outage()` (*in module salt.proxy.rest_sample*), 3085
`flags()` (*in module salt.states.portage_config*), 3665
`flavor_create()` (*in module salt.modules.nova*), 2254
`flavor_delete()` (*in module salt.modules.nova*), 2254
`flavor_list()` (*in module salt.modules.nova*), 2254
`flush()` (*in module salt.cache.consul*), 1026
`flush()` (*in module salt.cache.etcd_cache*), 1027
`flush()` (*in module salt.cache.localfs*), 1028
`flush()` (*in module salt.cache.mysql_cache*), 1028
`flush()` (*in module salt.cache.redis_cache*), 1031
`flush()` (*in module salt.modules.ipset*), 1896
`flush()` (*in module salt.modules.iptables*), 1900
`flush()` (*in module salt.modules.mine*), 2078
`flush()` (*in module salt.modules.nftables*), 2245
`flush()` (*in module salt.modules.pf*), 2337
`flush()` (*in module salt.modules.solaris_fmadm*), 2514
`flush()` (*in module salt.runners.cache*), 3101
`flush()` (*in module salt.states.ipset*), 3501
`flush()` (*in module salt.states.iptables*), 3506
`flush()` (*in module salt.states.nftables*), 3611
`flush()` (*salt.modules.inspectlib.dbhandle.DBHandleBase method*), 1875
`flush()` (*salt.modules.inspectlib.fsdb.CsvDB method*), 1877
`flush_api_stage_cache()` (*in module salt.modules.boto_apigateway*), 1339
`flush_synced()` (*in module salt.modules.elasticsearch*), 1684
`flushall()` (*in module salt.modules.redismod*), 2428
`flushBatch()` (*salt.returners.splunk.http_event_collector method*), 352
`flushdb()` (*in module salt.modules.redismod*), 2428
`fns()` (*in module salt.proxy.dummy*), 3056
`fns()` (*in module salt.proxy.napalm*), 3069
`fns()` (*in module salt.proxy.rest_sample*), 3085
`fns()` (*in module salt.proxy.ssh_sample*), 3086
`focus` (*salt.pillar.sql_base.SqlBaseExtPillar attribute*), 3035
`force_off()` (*in module salt.runners.virt*), 3158
`force_reload()` (*in module salt.modules.debian_service*), 1598
`force_reload()` (*in module salt.modules.netbsdservice*), 2203
`force_reload()` (*in module salt.modules.systemd_service*), 2593
`force_reload()` (*in module salt.modules.upstart_service*), 2628
`force_reset()` (*in module salt.modules.rabbitmq*), 2414
`forget()` (*in module salt.modules.mac_pkgutil*), 2031
`format_()` (*in module salt.modules.disk*), 1602
`format_pkg_list()` (*in module salt.modules.pkg_resource*), 2350
`format_return_text()` (*salt.engines.slack.SlackClient method*), 1167
`format_version()` (*in module salt.modules.pkg_resource*), 2350
`formatted()` (*in module salt.states.blockdev*), 3233
`ForwardingMapping` (*class in salt.states.firewalld*), 3452
`fqdns()` (*in module salt.grains.core*), 1178
`free_slave()` (*in module salt.modules.mysql*), 2101
`freecpu()` (*in module salt.modules.virt*), 2650
`freecpu()` (*in module salt.modules.xapi_virt*), 2853
`freemem()` (*in module salt.modules.virt*), 2650
`freemem()` (*in module salt.modules.xapi_virt*), 2853
`FreeSystemPtes` (*salt.modules.win_status.SYSTEM_PERFORMANCE_IN attribute*), 2808
`freeze()` (*in module salt.modules.lxc*), 1991
`freeze()` (*in module salt.modules.pip*), 2345
`freeze()` (*in module salt.runners.lxc*), 3119
`frozen()` (*in module salt.states.lxc*), 3545
`fs_create()` (*in module salt.modules.purefb*), 2402
`fs_delete()` (*in module salt.modules.purefb*), 2402
`fs_eradicate()` (*in module salt.modules.purefb*), 2403
`fs_extend()` (*in module salt.modules.purefb*), 2403
`fs_update()` (*in module salt.modules.purefb*), 2403
`fstab()` (*in module salt.modules.freebsdjail*), 1739
`fstab()` (*in module salt.modules.mount*), 2092
`fstab_absent()` (*in module salt.states.mount*), 3562
`fstab_present()` (*in module salt.states.mount*), 3563
`fstype()` (*in module salt.modules.disk*), 1603
`full_data()` (*in module salt.modules.publish*), 2393
`full_import()` (*in module salt.modules.solr*), 2529
`full_info()` (*in module salt.modules.virt*), 2650
`full_info()` (*in module salt.modules.xapi_virt*), 2853
`full_query()` (*in module salt.modules.cloud*), 1520
`full_query()` (*in module salt.runners.cloud*), 3103
`full_query()` (*salt.cloud.CloudClient method*), 3852

full_restart() (*in module salt.modules.daemontools*), 1585
full_restart() (*in module salt.modules.runit*), 2454
full_restart() (*in module salt.modules.s6*), 2461
full_restart() (*in module salt.modules.upstart_service*), 2628
fullversion() (*in module salt.modules.apache*), 1210
fullversion() (*in module salt.modules.dnsmasq*), 1607
fullversion() (*in module salt.modules.linux_lvm*), 1976
fullversion() (*in module salt.modules.tomcat*), 2620
function() (*in module salt.states.saltmod*), 3698
function_absent() (*in module salt.states.boto_lambda*), 3308
function_exists() (*in module salt.modules.boto_lambda*), 1422
function_present() (*in module salt.states.boto_lambda*), 3308
fx2() (*in module salt.grains.fx2*), 1180

G

gather_bootstrap_script() (*in module salt.modules.config*), 1557
gather_job_timeout
    conf/master, 55
gemset_copy() (*in module salt.modules.rvm*), 2456
gemset_create() (*in module salt.modules.rvm*), 2456
gemset_delete() (*in module salt.modules.rvm*), 2456
gemset_empty() (*in module salt.modules.rvm*), 2456
gemset_list() (*in module salt.modules.rvm*), 2456
gemset_list_all() (*in module salt.modules.rvm*), 2457
gemset_present() (*in module salt.states.rvm*), 3697
gen() (*in module salt.wheel.key*), 3840
gen_accept() (*in module salt.wheel.key*), 3840
gen_csr() (*in module salt.runners.digicertapi*), 3105
gen_hyper_keys() (*in module salt.pillar.libvirt*), 3011
gen_key() (*in module salt.runners.digicertapi*), 3105
gen_keys() (*in module salt.wheel.key*), 3841
gen_locale() (*in module salt.modules.localemod*), 1980
gen_mac
    jinja filters, 446
gen_password() (*in module salt.modules.solaris_shadow*), 2516
gen_signature() (*in module salt.wheel.key*), 3841
generate() (*in module salt.runners.thin*), 3156
generate_cert() (*in module salt.modules.icinga2*), 1855
generate_cert() (*in module salt.states.icinga2*), 3488
generate_data_key() (*in module salt.modules.boto_kms*), 1416
generate_data_key_without_plaintext() (*in module salt.modules.boto_kms*), 1417
generate_min() (*in module salt.runners.thin*), 3156
generate_nt_hash() (*in module salt.modules.pdbedit*), 2332
generate_random() (*in module salt.modules.boto_kms*), 1417
generate_ticket() (*in module salt.modules.icinga2*), 1855
generate_ticket() (*in module salt.states.icinga2*), 3488
generate_token() (*in module salt.runners.vault*), 3157
generate_triggered_messages() (*salt.engines.slack.SlackClient method*), 1167
generateBlobs() (*in module salt.modules.random_org*), 2420
generateDecimalFractions() (*in module salt.modules.random_org*), 2420
generateGaussians() (*in module salt.modules.random_org*), 2420
generateIntegers() (*in module salt.modules.random_org*), 2421
generateStrings() (*in module salt.modules.random_org*), 2421
generateUUIDs() (*in module salt.modules.random_org*), 2422
genrepo() (*in module salt.modules.win_pkg*), 2778
genrepo() (*in module salt.modules.winrepo*), 2791
genrepo() (*in module salt.runners.winrepo*), 3160
genrepo() (*in module salt.states.winrepo*), 3787
gentoo_mirrors_contains() (*in module salt.modules.makeconf*), 2062
get() (*in module salt.modules.augeas_cfg*), 1245
get() (*in module salt.modules.config*), 1557
get() (*in module salt.modules.consul*), 1569
get() (*in module salt.modules.data*), 1587
get() (*in module salt.modules.defaults*), 1600
get() (*in module salt.modules.dockercompose*), 1612
get() (*in module salt.modules.environ*), 1693
get() (*in module salt.modules.freebsd_sysctl*), 1738
get() (*in module salt.modules.gnomedesktop*), 1815
get() (*in module salt.modules.grains*), 1834
get() (*in module salt.modules.linux_sysctl*), 1979
get() (*in module salt.modules.logrotate*), 1983
get() (*in module salt.modules.mac_sysctl*), 2048
get() (*in module salt.modules.memcached*), 2077
get() (*in module salt.modules.mine*), 2078
get() (*in module salt.modules.netbsd_sysctl*), 2202
get() (*in module salt.modules.openbsd_sysctl*), 2270
get() (*in module salt.modules.openstack_config*), 2277
get() (*in module salt.modules.pillar*), 2341

get() (*in module salt.modules.rvm*), 2457
get() (*in module salt.modules.s3*), 2460
get() (*in module salt.modules.scp_mod*), 2484
get() (*in module salt.modules.sdb*), 2485
get() (*in module salt.modules.smartos_imgadm*), 2500
get() (*in module salt.modules.smartos_vmadm*), 2505
get() (*in module salt.modules.smbios*), 2508
get() (*in module salt.modules.splunk_search*), 2537
get() (*in module salt.modules.swift*), 2574
get() (*in module salt.modules.sysrc*), 2588
get() (*in module salt.modules.win_dacl*), 2723
get() (*in module salt.modules.win_lgpo*), 2766
get() (*in module salt.modules.win_wua*), 2841
get() (*in module salt.modules.zfs*), 2900
get() (*in module salt.modules.zookeeper*), 2914
get() (*in module salt.modules.zpool*), 2919
get() (*in module salt.runners.mine*), 3127
get() (*in module salt.runners.sdb*), 3148
get() (*in module salt.runners.smartos_vmadm*), 3149
get() (*in module salt.sdb.cache*), 3162
get() (*in module salt.sdb.confidant*), 3162
get() (*in module salt.sdb.consul*), 3163
get() (*in module salt.sdb.couchdb*), 3164
get() (*in module salt.sdb.env*), 3165
get() (*in module salt.sdb.etcd_db*), 3165
get() (*in module salt.sdb.keyring_db*), 3166
get() (*in module salt.sdb.memcached*), 3167
get() (*in module salt.sdb.redis_sdb*), 3167
get() (*in module salt.sdb.rest*), 3168
get() (*in module salt.sdb.sqlite3*), 3169
get() (*in module salt.sdb.tism*), 3170
get() (*in module salt.sdb.vault*), 3170
get() (*in module salt.sdb.yaml*), 3171
get() (*salt.modules.inspectlib.fsdb.CsvDB method*), 1877
GET() (*salt.netapi.rest_cherrypy.app.Events method*), 2952
GET() (*salt.netapi.rest_cherrypy.app.Jobs method*), 2948
GET() (*salt.netapi.rest_cherrypy.app.Keys method*), 2956
GET() (*salt.netapi.rest_cherrypy.app.Login method*), 2945
GET() (*salt.netapi.rest_cherrypy.app.LowDataAdapter method*), 2944
GET() (*salt.netapi.rest_cherrypy.app.Minions method*), 2947
GET() (*salt.netapi.rest_cherrypy.app.Stats method*), 2959
GET() (*salt.netapi.rest_cherrypy.app.WebsocketEndpoint method*), 2958
get() (*salt.netapi.rest_tornado.saltnado.EventsSaltAPIHandler method*), 2974
get() (*salt.netapi.rest_tornado.saltnado.JobsSaltAPIHandler method*), 2972
get() (*salt.netapi.rest_tornado.saltnado.MinionSaltAPIHandler method*), 2970
get() (*salt.netapi.rest_tornado.saltnado.SaltAPIHandler method*), 2967
get() (*salt.netapi.rest_tornado.saltnado.SaltAuthHandler method*), 2969
get_() (*in module salt.modules.etcd_mod*), 1696
get_() (*in module salt.modules.mdata*), 2076
get_() (*in module salt.modules.netbox*), 2200
get_a() (*in module salt.modules.infoblox*), 1869
get_account_created() (*in module salt.modules.mac_shadow*), 2042
get_account_id() (*in module salt.modules.boto_iam*), 1399
get_account_policy() (*in module salt.modules.boto_iam*), 1399
get_acls() (*in module salt.modules.zookeeper*), 2915
get_address() (*in module salt.modules.panos*), 2307
get_address_group() (*in module salt.modules.panos*), 2307
get_admins_active() (*in module salt.modules.panos*), 2307
get_admins_all() (*in module salt.modules.panos*), 2307
get_advanced_configs() (*in module salt.modules.vsphere*), 2692
get_affinity_group() (*in module salt.cloud.clouds.msazure*), 1075
get_agent_service_types() (*in module salt.modules.win_snmp*), 2804
get_agent_settings() (*in module salt.modules.win_snmp*), 2804
get_alarm() (*in module salt.modules.boto_cloudwatch*), 1352
get_alarms() (*in module salt.modules.telemetry*), 2599
get_alert_config() (*in module salt.modules.telemetry*), 2599
get_alias() (*in module salt.modules.hosts*), 1849
get_all() (*in module salt.modules.daemontools*), 1585
get_all() (*in module salt.modules.debian_service*), 1598
get_all() (*in module salt.modules.dummyproxy_service*), 1674
get_all() (*in module salt.modules.freebsdservice*), 1747
get_all() (*in module salt.modules.gentoo_service*), 1755
get_all() (*in module salt.modules.launchctl_service*), 1951
get_all() (*in module salt.modules.mac_service*), 2039
get_all() (*in module salt.modules.netbsdservice*), 2203
get_all() (*in module salt.modules.openbsdrcctl_service*), 2273
get_all() (*in module salt.modules.openbsdservice*), 2275

get_all() (*in module salt.modules.rest_service*), 2439
get_all() (*in module salt.modules.rh_service*), 2444
get_all() (*in module salt.modules.runit*), 2454
get_all() (*in module salt.modules.s6*), 2461
get_all() (*in module salt.modules.smf_service*), 2511
get_all() (*in module salt.modules.ssh_service*), 2542
get_all() (*in module salt.modules.systemd_service*), 2594
get_all() (*in module salt.modules.upstart_service*), 2628
get_all() (*in module salt.modules.win_service*), 2797
get_all_access_keys() (*in module salt.modules.boto_iam*), 1399
get_all_alarms() (*in module salt.modules.boto_cloudwatch*), 1352
get_all_cache_subnet_groups() (*in module salt.modules.boto_elasticache*), 1381
get_all_cpv_use() (*in module salt.modules.portage_config*), 2371
get_all_eip_addresses() (*in module salt.modules.boto_ec2*), 1367
get_all_elbs() (*in module salt.modules.boto_elb*), 1388
get_all_group_policies() (*in module salt.modules.boto_iam*), 1400
get_all_groups() (*in module salt.modules.boto_asg*), 1343
get_all_groups() (*in module salt.modules.boto_iam*), 1400
get_all_instance_profiles() (*in module salt.modules.boto_iam*), 1400
get_all_interfaces() (*in module salt.modules.win_ip*), 2764
get_all_launch_configurations() (*in module salt.modules.boto_asg*), 1343
get_all_mfa_devices() (*in module salt.modules.boto_iam*), 1400
get_all_profiles() (*in module salt.modules.win_firewall*), 2747
get_all_roles() (*in module salt.modules.boto_iam*), 1400
get_all_security_groups() (*in module salt.modules.boto_secgroup*), 1442
get_all_settings() (*in module salt.modules.win_firewall*), 2747
get_all_subscriptions_by_topic() (*in module salt.modules.boto_sns*), 1445
get_all_tags() (*in module salt.modules.boto_ec2*), 1367
get_all_topics() (*in module salt.modules.boto_sns*), 1445
get_all_user_policies() (*in module salt.modules.boto_iam*), 1400
get_all_users() (*in module salt.modules.boto_iam*), 1401
get_all_volumes() (*in module salt.modules.boto_ec2*), 1368
get_antivirus_info() (*in module salt.modules.panos*), 2308
get_api_versions() (*in module salt.cloud.clouds.azurearm*), 1037
get_args() (*salt.states.ansiblegate.AnsibleState method*), 3185
get_arn() (*in module salt.modules.boto_sns*), 1445
get_arp() (*in module salt.modules.panos*), 2308
get_attribute() (*in module salt.modules.boto_ec2*), 1368
get_attribute() (*in module salt.modules.xml*), 2861
get_attributes() (*in module salt.modules.boto_elb*), 1388
get_attributes() (*in module salt.modules.boto_sqs*), 1446
get_attributes() (*in module salt.modules.win_file*), 2735
get_auth_traps_enabled() (*in module salt.modules.win_snmp*), 2805
get_auth_url() (*in module salt.auth.keystone*), 1001
get_auto_login() (*in module salt.modules.mac_user*), 2058
get_availability_zone() (*in module salt.cloud.clouds.ec2*), 1048
get_available_extension() (*in module salt.modules.postgres*), 2376
get_backend() (*in module salt.modules.haproxyconn*), 1840
get_balancer() (*in module salt.modules.libcloud_loadbalancer*), 1970
get_balancer_by_name() (*in module salt.modules.libcloud_loadbalancer*), 1970
get_balances() (*in module salt.modules.namecheap_users*), 2124
get_bind_data() (*in module salt.modules.libcloud_dns*), 1967
get_bios_defaults() (*in module salt.modules.cimc*), 1507
get_bios_settings() (*in module salt.modules.cimc*), 1508
get_blob() (*in module salt.cloud.clouds.msazure*), 1075
get_blob_properties() (*in module salt.cloud.clouds.msazure*), 1075
get_blob_service_properties() (*in module salt.cloud.clouds.msazure*), 1076
get_block_device() (*in module salt.modules.parted_partition*), 2325
get_bond() (*in module salt.modules.debian_ip*), 1597
get_bond() (*in module salt.modules.rh_ip*), 2442
get_boot_arch() (*in module*

salt.modules.mac_system), 2049
`get_boot_order()` (*in module salt.modules.cimc*), 1508
`get_bootdev()` (*in module salt.modules.ipmi*), 1884
`get_bufsize()` (*in module salt.modules.network*), 2216
`get_ca()` (*in module salt.modules.tls*), 2616
`get_ca_signed_cert()` (*in module salt.modules.tls*), 2616
`get_ca_signed_key()` (*in module salt.modules.tls*), 2616
`get_cache_subnet_group()` (*in module salt.modules.boto_elasticache*), 1381
`get_capabilities()` (*in module salt.modules.win_dism*), 2725
`get_catalog()` (*in module salt.modules.mac_softwareupdate*), 2045
`get_cert_file()` (*in module salt.modules.win_pki*), 2785
`get_cert_serial()` (*in module salt.modules.win_certutil*), 2721
`get_certificate()` (*in module salt.runners.digicertapi*), 3105
`get_certs()` (*in module salt.modules.win_pki*), 2785
`get_cflags()` (*in module salt.modules.makeconf*), 2063
`get_change()` (*in module salt.modules.mac_shadow*), 2042
`get_channel_access()` (*in module salt.modules.ipmi*), 1885
`get_channel_info()` (*in module salt.modules.ipmi*), 1886
`get_channel_max_user_count()` (*in module salt.modules.ipmi*), 1886
`get_chassis_datacenter()` (*in module salt.modules.dracr*), 1666
`get_chassis_location()` (*in module salt.modules.dracr*), 1667
`get_chassis_name()` (*in module salt.modules.dracr*), 1667
`get_children()` (*in module salt.modules.zookeeper*), 2915
`get_chost()` (*in module salt.modules.makeconf*), 2063
`get_circuit_provider()` (*in module salt.modules.netbox*), 2200
`get_cleared_flags()` (*in module salt.modules.portage_config*), 2371
`get_cli_idle_timeout()` (*in module salt.modules.panos*), 2308
`get_cli_permissions()` (*in module salt.modules.panos*), 2308
`get_cli_returns()` (*salt.client.LocalClient method*), 3848
`get_client_args()` (*in module salt.modules.dockermod*), 1639
`get_clonespec_for_valid_snapshot()` (*in module salt.cloud.clouds.vmware*), 1141
`get_cloud_init_mime()` (*in module salt.modules.boto_asg*), 1343
`get_cluster_id()` (*in module salt.cloud.clouds.opennebula*), 1091
`get_cname()` (*in module salt.modules.infoblox*), 1869
`get_community_names()` (*in module salt.modules.win_snmp*), 2805
`get_compatible_elasticsearch_versions()` (*in module salt.modules.boto3_elasticsearch*), 1316
`get_computer_desc()` (*in module salt.modules.system*), 2588
`get_computer_desc()` (*in module salt.modules.win_system*), 2812
`get_computer_name()` (*in module salt.modules.mac_system*), 2049
`get_computer_name()` (*in module salt.modules.system*), 2588
`get_computer_name()` (*in module salt.modules.win_system*), 2812
`get_computer_sleep()` (*in module salt.modules.mac_power*), 2034
`get_config()` (*in module salt.modules.arista_pyeapi*), 1237
`get_config()` (*in module salt.modules.boto_asg*), 1343
`get_config()` (*in module salt.modules.boto_elasticache*), 1381
`get_config()` (*in module salt.modules.boto_secgroup*), 1443
`get_config()` (*in module salt.modules.dnsmasq*), 1607
`get_config()` (*in module salt.modules.napalm_yang_mod*), 2189
`get_config()` (*in module salt.modules.snapper*), 2545
`get_config()` (*in module salt.modules.win_dsc*), 2730
`get_config()` (*in module salt.modules.win_firewall*), 2747
`get_config_file()` (*in module salt.modules.syslog_ng*), 2580
`get_config_groups()` (*salt.engines.slack.SlackClient method*), 1167
`get_config_id()` (*in module salt.cloud.clouds.linode*), 1067
`get_config_resolver_class()` (*in module salt.proxy.cimc*), 3051
`get_config_status()` (*in module salt.modules.win_dsc*), 2730
`get_configured_provider()` (*in module salt.cloud.clouds.aliyun*), 1033
`get_configured_provider()` (*in module salt.cloud.clouds.azurearm*), 1037
`get_configured_provider()` (*in module

salt.cloud.clouds.cloudstack), 1038
get_configured_provider() (*in module salt.cloud.clouds.digitalocean*), 1041
get_configured_provider() (*in module salt.cloud.clouds.dimensiondata*), 1043
get_configured_provider() (*in module salt.cloud.clouds.ec2*), 1048
get_configured_provider() (*in module salt.cloud.clouds.gce*), 1057
get_configured_provider() (*in module salt.cloud.clouds.gogrid*), 1060
get_configured_provider() (*in module salt.cloud.clouds.joyent*), 1063
get_configured_provider() (*in module salt.cloud.clouds.linode*), 1067
get_configured_provider() (*in module salt.cloud.clouds.lxc*), 1070
get_configured_provider() (*in module salt.cloud.clouds.msazure*), 1076
get_configured_provider() (*in module salt.cloud.clouds.oneandone*), 1089
get_configured_provider() (*in module salt.cloud.clouds.opennebula*), 1091
get_configured_provider() (*in module salt.cloud.clouds.openstack*), 1112
get_configured_provider() (*in module salt.cloud.clouds.parallels*), 1114
get_configured_provider() (*in module salt.cloud.clouds.profitbricks*), 1117
get_configured_provider() (*in module salt.cloud.clouds.proxmox*), 1120
get_configured_provider() (*in module salt.cloud.clouds.pyrax*), 1121
get_configured_provider() (*in module salt.cloud.clouds.qingcloud*), 1123
get_configured_provider() (*in module salt.cloud.clouds.saltify*), 1125
get_configured_provider() (*in module salt.cloud.clouds.scaleway*), 1126
get_configured_provider() (*in module salt.cloud.clouds.softlayer*), 1128
get_configured_provider() (*in module salt.cloud.clouds.softlayer_hw*), 1129
get_configured_provider() (*in module salt.cloud.clouds.tencentcloud*), 1131
get_configured_provider() (*in module salt.cloud.clouds.vagrant*), 1134
get_configured_provider() (*in module salt.cloud.clouds.vultrpy*), 1148
get_configured_provider() (*in module salt.cloud.clouds.xen*), 1150
get_conn() (*in module salt.cloud.clouds.azurearm*), 1037
get_conn() (*in module salt.cloud.clouds.cloudstack*), 1038
get_conn() (*in module salt.cloud.clouds.dimensiondata*), 1043
get_conn() (*in module salt.cloud.clouds.gce*), 1057
get_conn() (*in module salt.cloud.clouds.msazure*), 1076
get_conn() (*in module salt.cloud.clouds.oneandone*), 1089
get_conn() (*in module salt.cloud.clouds.openstack*), 1113
get_conn() (*in module salt.cloud.clouds.profitbricks*), 1117
get_conn() (*in module salt.cloud.clouds.pyrax*), 1121
get_conn() (*in module salt.cloud.clouds.softlayer*), 1128
get_conn() (*in module salt.cloud.clouds.softlayer_hw*), 1129
get_conn() (*in module salt.pillar.consul_pillar*), 2994
get_conn() (*in module salt.sdb.consul*), 3163
get_conn_args() (*in module salt.proxy.nxos_api*), 3078
get_connection() (*in module salt.modules.arista_pyeapi*), 1238
get_connection() (*in module salt.modules.netmiko_mod*), 2209
get_connection_ip_list() (*in module salt.modules.win_smtp_server*), 2802
get_console_output() (*in module salt.cloud.clouds.ec2*), 1048
get_container() (*in module salt.modules.libcloud_storage*), 1973
get_container_object() (*in module salt.modules.libcloud_storage*), 1974
get_container_profile() (*in module salt.modules.lxc*), 1992
get_container_setting() (*in module salt.modules.win_iis*), 2757
get_continuous_query() (*in module salt.modules.influxdbmod*), 1859
get_coredump_network_config() (*in module salt.modules.vsphere*), 2693
get_cpu_details() (*in module salt.modules.cimc*), 1508
get_current_target() (*in module salt.modules.eselect*), 1695
get_cxxflags() (*in module salt.modules.makeconf*), 2063
get_dashboard() (*in module salt.modules.grafana4*), 1826
get_data() (*in module salt.modules.cisconso*), 1518
get_data() (*in module salt.proxy.cisconso*), 3056
get_data() (*salt.roster.sshconfig.RosterMatcher method*), 3093
get_data_disk() (*in module*

salt.cloud.clouds.linode), 1067
get_data_disk_size() (*in module salt.cloud.clouds.linode*), 1067
get_datacenter() (*in module salt.cloud.clouds.profitbricks*), 1117
get_datacenter_id() (*in module salt.cloud.clouds.linode*), 1067
get_datacenter_id() (*in module salt.cloud.clouds.profitbricks*), 1117
get_datasource() (*in module salt.modules.grafana4*), 1826
get_datasources() (*in module salt.modules.grafana4*), 1826
get_datastore_id() (*in module salt.cloud.clouds.opennebula*), 1092
get_date() (*in module salt.modules.mac_timezone*), 2054
get_default_gateway() (*in module salt.modules.win_ip*), 2764
get_default_keychain() (*in module salt.modules.mac_keychain*), 2028
get_dependencies() (*in module salt.cloud.clouds.aliyun*), 1033
get_dependencies() (*in module salt.cloud.clouds.azurearm*), 1037
get_dependencies() (*in module salt.cloud.clouds.cloudstack*), 1038
get_dependencies() (*in module salt.cloud.clouds.digitalocean*), 1041
get_dependencies() (*in module salt.cloud.clouds.dimensiondata*), 1043
get_dependencies() (*in module salt.cloud.clouds.ec2*), 1048
get_dependencies() (*in module salt.cloud.clouds.gce*), 1057
get_dependencies() (*in module salt.cloud.clouds.msazure*), 1076
get_dependencies() (*in module salt.cloud.clouds.oneandone*), 1089
get_dependencies() (*in module salt.cloud.clouds.opennebula*), 1092
get_dependencies() (*in module salt.cloud.clouds.openstack*), 1113
get_dependencies() (*in module salt.cloud.clouds.profitbricks*), 1117
get_dependencies() (*in module salt.cloud.clouds.proxmox*), 1120
get_dependencies() (*in module salt.cloud.clouds.pyrax*), 1121
get_dependencies() (*in module salt.cloud.clouds.qingcloud*), 1123
get_dependencies() (*in module salt.cloud.clouds.softlayer*), 1128
get_dependencies() (*in module salt.cloud.clouds.softlayer_hw*), 1129
get_dependencies() (*in module salt.cloud.clouds.tencentcloud*), 1131
get_deployment() (*in module salt.cloud.clouds.msazure*), 1076
get_details() (*in module salt.modules.esxi*), 1696
get_details() (*in module salt.proxy.esxi*), 3061
get_device() (*in module salt.proxy.napalm*), 3069
get_devmm() (*in module salt.modules.file*), 1712
get_dhcp_options() (*in module salt.modules.boto_vpc*), 1458
get_diff() (*in module salt.modules.file*), 1712
get_dir() (*in module salt.modules.cp*), 1575
get_disable_keyboard_on_lock() (*in module salt.modules.mac_system*), 2049
get_disabled() (*in module salt.modules.debian_service*), 1598
get_disabled() (*in module salt.modules.freebsdservice*), 1747
get_disabled() (*in module salt.modules.gentoo_service*), 1755
get_disabled() (*in module salt.modules.netbsdservice*), 2203
get_disabled() (*in module salt.modules.openbsdrcctl_service*), 2273
get_disabled() (*in module salt.modules.openbsdservice*), 2275
get_disabled() (*in module salt.modules.rh_service*), 2444
get_disabled() (*in module salt.modules.runit*), 2454
get_disabled() (*in module salt.modules.smf_service*), 2511
get_disabled() (*in module salt.modules.systemd_service*), 2594
get_disabled() (*in module salt.modules.upstart_service*), 2628
get_disabled() (*in module salt.modules.win_service*), 2797
get_disk() (*in module salt.cloud.clouds.msazure*), 1076
get_disk_size() (*in module salt.cloud.clouds.linode*), 1067
get_disk_timeout() (*in module salt.modules.win_powercfg*), 2787
get_disk_type() (*in module salt.cloud.clouds.profitbricks*), 1117
get_disk_usage() (*in module salt.modules.panos*), 2308
get_disks() (*in module salt.modules.cimc*), 1508
get_disks() (*in module salt.modules.virt*), 2651
get_disks() (*in module salt.modules.xapi_virt*), 2853
get_display_sleep() (*in module salt.modules.mac_power*), 2035
get_distribution() (*in module*

salt.modules.boto_cloudfront), 1347
get_distribution_id() (*in module salt.cloud.clouds.linode*), 1067
get_distribution_path() (*in module salt.modules.virtualenv_mod*), 2677
get_dns_config() (*in module salt.modules.win_dns_client*), 2728
get_dns_dracname() (*in module salt.modules.dracr*), 1667
get_dns_server_config() (*in module salt.modules.panos*), 2308
get_dns_servers() (*in module salt.modules.win_dns_client*), 2728
get_docker() (*in module salt.modules.mine*), 2079
get_domain_config() (*in module salt.modules.panos*), 2308
get_domain_workgroup() (*in module salt.modules.win_system*), 2812
get_dos_blocks() (*in module salt.modules.panos*), 2308
get_eip_address_info() (*in module salt.modules.boto_ec2*), 1369
get_elb_config() (*in module salt.modules.boto_elb*), 1388
get_emerge_default_opts() (*in module salt.modules.makeconf*), 2063
get_enabled() (*in module salt.modules.debian_service*), 1598
get_enabled() (*in module salt.modules.freebsdjail*), 1739
get_enabled() (*in module salt.modules.freebsdservice*), 1748
get_enabled() (*in module salt.modules.gentoo_service*), 1755
get_enabled() (*in module salt.modules.mac_service*), 2039
get_enabled() (*in module salt.modules.netbsdservice*), 2204
get_enabled() (*in module salt.modules.openbsdrcctl_service*), 2273
get_enabled() (*in module salt.modules.openbsdservice*), 2275
get_enabled() (*in module salt.modules.rh_service*), 2444
get_enabled() (*in module salt.modules.runit*), 2454
get_enabled() (*in module salt.modules.smf_service*), 2511
get_enabled() (*in module salt.modules.systemd_service*), 2594
get_enabled() (*in module salt.modules.upstart_service*), 2628
get_enabled() (*in module salt.modules.win_service*), 2797
get_endpoint() (*in module salt.modules.boto_rds*), 1427
get_endtime() (*in module salt.returners.local_cache*), 325
get_entity() (*in module salt.modules.keystoneng*), 1938
get_error_message() (*in module salt.exceptions*), 4001
get_escalation_policies() (*in module salt.modules.pagerduty_util*), 2304
get_ethernet_interfaces() (*in module salt.modules.cimc*), 1508
get_event_iter_returns() (*salt.client.LocalClient method*), 3848
get_event_source_mapping_ids() (*in module salt.modules.boto_lambda*), 1422
get_expiration_date() (*in module salt.modules.tls*), 2617
get_expire() (*in module salt.modules.mac_shadow*), 2042
get_extensions() (*in module salt.modules.tls*), 2617
get_fac() (*in module salt.modules.peeringdb*), 2334
get_features() (*in module salt.modules.makeconf*), 2063
get_features() (*in module salt.modules.win_dism*), 2725
get_fibre_channel_interfaces() (*in module salt.modules.cimc*), 1508
get_file() (*in module salt.modules.cp*), 1575
get_file_str() (*in module salt.modules.cp*), 1576
get_file_systems() (*in module salt.modules.boto_efs*), 1377
get_filter_config() (*in module salt.modules.capirca_acl*), 1473
get_filter_pillar() (*in module salt.modules.capirca_acl*), 1474
get_filter_pillar() (*in module salt.modules.napalm_netacl*), 2139
get_firewall_status() (*in module salt.modules.vsphere*), 2693
get_firmware() (*in module salt.modules.cimc*), 1508
get_flags_from_package_conf() (*in module salt.modules.portage_config*), 2371
get_fmri() (*in module salt.modules.solarisipspkg*), 2521
get_fqdn() (*in module salt.modules.network*), 2216
get_fqdn_cache() (*in module salt.modules.panos*), 2308
get_friendly_name() (*in module salt.modules.mac_keychain*), 2028
get_ftp_proxy() (*in module salt.modules.proxy*), 2387
get_fun() (*in module salt.modules.ret*), 2441
get_fun() (*in module salt.returners.cassandra_cql_return*), 315

get_fun() (*in module salt.returners.couchdb_return*), 318
get_fun() (*in module salt.returners.etcd_return*), 321
get_fun() (*in module salt.returners.influxdb_return*), 323
get_fun() (*in module salt.returners.memcache_return*), 327
get_fun() (*in module salt.returners.mongo_future_return*), 329
get_fun() (*in module salt.returners.mongo_return*), 330
get_fun() (*in module salt.returners.mysql*), 333
get_fun() (*in module salt.returners.odbc*), 336
get_fun() (*in module salt.returners.pgjsonb*), 339
get_fun() (*in module salt.returners.postgres*), 342
get_fun() (*in module salt.returners.redis_return*), 347
get_fun() (*in module salt.returners.sqlite3_return*), 353
get_general() (*in module salt.modules.dracr*), 1667
get_gentoo_mirrors() (*in module salt.modules.makeconf*), 2063
get_gid() (*in module salt.modules.file*), 1713
get_gid() (*in module salt.modules.win_file*), 2735
get_grains() (*in module salt.proxy.napalm*), 3069
get_graphics() (*in module salt.modules.virt*), 2651
get_group() (*in module salt.modules.boto_iam*), 1401
get_group() (*in module salt.modules.file*), 1713
get_group() (*in module salt.modules.win_file*), 2736
get_group_host() (*in module salt.modules.boto_elasticache*), 1381
get_group_id() (*in module salt.modules.boto_secgroup*), 1443
get_group_members() (*in module salt.modules.boto_iam*), 1401
get_group_policy() (*in module salt.modules.boto_iam*), 1401
get_ha_config() (*in module salt.modules.panos*), 2309
get_ha_link() (*in module salt.modules.panos*), 2309
get_ha_path() (*in module salt.modules.panos*), 2309
get_ha_state() (*in module salt.modules.panos*), 2309
get_ha_transitions() (*in module salt.modules.panos*), 2309
get_harddisk_sleep() (*in module salt.modules.mac_power*), 2035
get_hash() (*in module salt.modules.file*), 1713
get_hash() (*in module salt.modules.mac_keychain*), 2028
get_health() (*in module salt.modules.ipmi*), 1886
get_health_check() (*in module salt.modules.boto_elb*), 1389
get_hibernate_timeout() (*in module salt.modules.win_powercfg*), 2787
get_host() (*in module salt.modules.infoblox*), 1869
get_host_advanced() (*in module salt.modules.infoblox*), 1870
get_host_cache() (*in module salt.modules.vsphere*), 2693
get_host_datetime() (*in module salt.modules.vsphere*), 2694
get_host_domainname() (*in module salt.modules.infoblox*), 1870
get_host_hostname() (*in module salt.modules.infoblox*), 1870
get_host_id() (*in module salt.cloud.clouds.opennebula*), 1092
get_host_ipv4() (*in module salt.modules.infoblox*), 1870
get_host_ipv4addr_info() (*in module salt.modules.infoblox*), 1870
get_host_ipv6addr_info() (*in module salt.modules.infoblox*), 1871
get_host_mac() (*in module salt.modules.infoblox*), 1871
get_hosted_zone() (*in module salt.modules.boto3_route53*), 1327
get_hosted_zones_by_domain() (*in module salt.modules.boto3_route53*), 1327
get_hostname() (*in module salt.modules.cimc*), 1508
get_hostname() (*in module salt.modules.network*), 2217
get_hostname() (*in module salt.modules.panos*), 2309
get_hostname() (*in module salt.modules.win_system*), 2812
get_hosts() (*in module salt.modules.namecheap_domains_dns*), 2113
get_http_proxy() (*in module salt.modules.proxy*), 2387
get_https_proxy() (*in module salt.modules.proxy*), 2388
get_humidity() (*in module salt.modules.sensehat*), 2492
get_hwclock() (*in module salt.modules.mac_timezone*), 2054
get_hwclock() (*in module salt.modules.timezone*), 2608
get_hwclock() (*in module salt.modules.win_timezone*), 2831
get_hypervisor() (*in module salt.modules.virt*), 2651
get_icmp_types() (*in module salt.modules.firewalld*), 1733
get_id() (*in module salt.modules.boto_ec2*), 1369
get_id() (*in module salt.modules.boto_vpc*), 1458
get_id() (*in module salt.modules.parted_partition*), 2326
get_identity_pool_roles() (*in module salt.modules.boto_cognitoidentity*), 1356
get_image() (*in module salt.cloud.clouds.aliyun*), 1033

`get_image()` (*in module salt.cloud.clouds.cloudstack*), 1039
`get_image()` (*in module salt.cloud.clouds.digitalocean*), 1041
`get_image()` (*in module salt.cloud.clouds.dimensiondata*), 1044
`get_image()` (*in module salt.cloud.clouds.joyent*), 1063
`get_image()` (*in module salt.cloud.clouds.oneandone*), 1089
`get_image()` (*in module salt.cloud.clouds.opennebula*), 1092
`get_image()` (*in module salt.cloud.clouds.parallels*), 1114
`get_image()` (*in module salt.cloud.clouds.profitbricks*), 1117
`get_image()` (*in module salt.cloud.clouds.scaleway*), 1127
`get_image()` (*in module salt.modules.libcloud_compute*), 1963
`get_image_id()` (*in module salt.cloud.clouds.opennebula*), 1092
`get_imageid()` (*in module salt.cloud.clouds.ec2*), 1048
`get_info()` (*in module salt.modules.namecheap_domains*), 2115
`get_info()` (*in module salt.modules.namecheap_domains_ns*), 2117
`get_info()` (*in module salt.modules.namecheap_ssl*), 2120
`get_info_for_reshard()` (*in module salt.modules.boto_kinesis*), 1413
`get_ini()` (*in module salt.modules.ini_manage*), 1872
`get_input_endpoint()` (*in module salt.cloud.clouds.msazure*), 1076
`get_installed_extension()` (*in module salt.modules.postgres*), 2376
`get_installed_use()` (*in module salt.modules.portage_config*), 2371
`get_instance()` (*in module salt.modules.cloud*), 1520
`get_instance_health()` (*in module salt.modules.boto_elb*), 1389
`get_instances()` (*in module salt.modules.boto_asg*), 1343
`get_interface()` (*in module salt.modules.debian_ip*), 1597
`get_interface()` (*in module salt.modules.linux_ip*), 1976
`get_interface()` (*in module salt.modules.nilrt_ip*), 2249
`get_interface()` (*in module salt.modules.rh_ip*), 2442
`get_interface()` (*in module salt.modules.win_ip*), 2764
`get_interface_counters()` (*in module salt.modules.panos*), 2309
`get_interfaces()` (*in module salt.modules.firewalld*), 1733
`get_interfaces()` (*in module salt.modules.netbox*), 2200
`get_interfaces()` (*in module salt.modules.panos*), 2309
`get_interfaces_details()` (*in module salt.modules.nilrt_ip*), 2249
`get_ip()` (*in module salt.cloud.clouds.cloudstack*), 1039
`get_ip()` (*in module salt.modules.hosts*), 1850
`get_ipaddresses()` (*in module salt.modules.netbox*), 2200
`get_ips()` (*in module salt.cloud.clouds.linode*), 1068
`get_ipv4_range()` (*in module salt.modules.infoblox*), 1871
`get_issue()` (*in module salt.modules.github*), 1801
`get_issue_comments()` (*in module salt.modules.github*), 1802
`get_issues()` (*in module salt.modules.github*), 1802
`get_iuse()` (*in module salt.modules.portage_config*), 2371
`get_ix()` (*in module salt.modules.peeringdb*), 2334
`get_ixfac()` (*in module salt.modules.peeringdb*), 2334
`get_ixlan()` (*in module salt.modules.peeringdb*), 2334
`get_ixpfx()` (*in module salt.modules.peeringdb*), 2335
`get_jid()` (*in module salt.modules.ret*), 2441
`get_jid()` (*in module salt.returners.cassandra_cql_return*), 315
`get_jid()` (*in module salt.returners.couchbase_return*), 317
`get_jid()` (*in module salt.returners.couchdb_return*), 318
`get_jid()` (*in module salt.returners.etcd_return*), 321
`get_jid()` (*in module salt.returners.influxdb_return*), 323
`get_jid()` (*in module salt.returners.local_cache*), 325
`get_jid()` (*in module salt.returners.memcache_return*), 327
`get_jid()` (*in module salt.returners.mongo_future_return*), 329
`get_jid()` (*in module salt.returners.mongo_return*), 330
`get_jid()` (*in module salt.returners.multi_returner*), 330
`get_jid()` (*in module salt.returners.mysql*), 333
`get_jid()` (*in module salt.returners.odbc*), 336
`get_jid()` (*in module salt.returners.pgjsonb*), 339
`get_jid()` (*in module salt.returners.postgres*), 342
`get_jid()` (*in module salt.returners.postgres_local_cache*), 344
`get_jid()` (*in module salt.returners.redis_return*), 347
`get_jid()` (*in module salt.returners.sqlite3_return*), 353
`get_jids()` (*in module salt.modules.ret*), 2441
`get_jids()` (*in module*

salt.returners.cassandra_cql_return), 315
`get_jids()` (*in module salt.returners.couchbase_return*), 317
`get_jids()` (*in module salt.returners.couchdb_return*), 318
`get_jids()` (*in module salt.returners.etcd_return*), 321
`get_jids()` (*in module salt.returners.influxdb_return*), 323
`get_jids()` (*in module salt.returners.local_cache*), 325
`get_jids()` (*in module salt.returners.memcache_return*), 327
`get_jids()` (*in module salt.returners.mongo_future_return*), 329
`get_jids()` (*in module salt.returners.multi_returner*), 330
`get_jids()` (*in module salt.returners.mysql*), 333
`get_jids()` (*in module salt.returners.odbc*), 336
`get_jids()` (*in module salt.returners.pgjsonb*), 339
`get_jids()` (*in module salt.returners.postgres*), 342
`get_jids()` (*in module salt.returners.postgres_local_cache*), 344
`get_jids()` (*in module salt.returners.redis_return*), 347
`get_jids()` (*in module salt.returners.sqlite3_return*), 353
`get_jids_filter()` (*in module salt.returners.local_cache*), 325
`get_jids_filter()` (*in module salt.returners.mysql*), 333
`get_job()` (*in module salt.modules.panos*), 2310
`get_job_config()` (*in module salt.modules.jenkinsmod*), 1909
`get_job_info()` (*in module salt.modules.jenkinsmod*), 1909
`get_jobs()` (*in module salt.modules.jenkinsmod*), 1909
`get_jobs()` (*in module salt.modules.panos*), 2310
`get_jobs_from_runner()` (*salt.engines.slack.SlackClient method*), 1167
`get_key()` (*in module salt.cloud.clouds.cloudstack*), 1039
`get_key()` (*in module salt.modules.boto_ec2*), 1369
`get_key()` (*in module salt.modules.gpg*), 1819
`get_key()` (*in module salt.modules.redismod*), 2428
`get_key_filename()` (*in module salt.cloud.clouds.oneandone*), 1089
`get_key_filename()` (*in module salt.cloud.clouds.profitbricks*), 1117
`get_key_pair()` (*in module salt.modules.libcloud_compute*), 1963
`get_key_policy()` (*in module salt.modules.boto_kms*), 1417
`get_key_rotation_status()` (*in module salt.modules.boto_kms*), 1417
`get_keyid()` (*in module salt.cloud.clouds.digitalocean*), 1041
`get_keypair()` (*in module salt.cloud.clouds.cloudstack*), 1039
`get_keys()` (*in module salt.modules.boto_ec2*), 1369
`get_known_host_entries()` (*in module salt.modules.ssh*), 2539
`get_labels()` (*in module salt.modules.k8s*), 1920
`get_lacp()` (*in module salt.modules.panos*), 2310
`get_last_change()` (*in module salt.modules.mac_shadow*), 2042
`get_latest_release()` (*in module salt.modules.artifactory*), 1240
`get_latest_snapshot()` (*in module salt.modules.artifactory*), 1241
`get_lb_conn()` (*in module salt.cloud.clouds.dimensiondata*), 1044
`get_lb_conn()` (*in module salt.cloud.clouds.gce*), 1057
`get_lcm_config()` (*in module salt.modules.win_dsc*), 2730
`get_ldap()` (*in module salt.modules.cimc*), 1509
`get_license_info()` (*in module salt.modules.panos*), 2310
`get_license_tokens()` (*in module salt.modules.panos*), 2310
`get_linode()` (*in module salt.cloud.clouds.linode*), 1068
`get_linode_id_from_name()` (*in module salt.cloud.clouds.linode*), 1068
`get_list()` (*in module salt.modules.namecheap_domains*), 2116
`get_list()` (*in module salt.modules.namecheap_domains_dns*), 2114
`get_list()` (*in module salt.modules.namecheap_ssl*), 2121
`get_lldp_config()` (*in module salt.modules.panos*), 2310
`get_lldp_counters()` (*in module salt.modules.panos*), 2310
`get_lldp_local()` (*in module salt.modules.panos*), 2310
`get_lldp_neighbors()` (*in module salt.modules.panos*), 2311
`get_load()` (*in module salt.returners.cassandra_cql_return*), 315
`get_load()` (*in module salt.returners.couchbase_return*), 317
`get_load()` (*in module salt.returners.elasticsearch_return*), 320
`get_load()` (*in module salt.returners.etcd_return*), 321
`get_load()` (*in module salt.returners.influxdb_return*), 323
`get_load()` (*in module salt.returners.local_cache*), 325
`get_load()` (*in module salt.returners.memcache_return*), 327
`get_load()` (*in module*

salt.returners.mongo_future_return), 329
get_load() (*in module salt.returners.multi_returner*), 330
get_load() (*in module salt.returners.mysql*), 333
get_load() (*in module salt.returners.odbc*), 336
get_load() (*in module salt.returners.pgjsonb*), 339
get_load() (*in module salt.returners.postgres*), 342
get_load() (*in module salt.returners.postgres_local_cache*), 344
get_load() (*in module salt.returners.redis_return*), 347
get_load() (*in module salt.returners.sqlite3_return*), 354
get_local_admins() (*in module salt.modules.panos*), 2311
get_locale() (*in module salt.modules.localemod*), 1980
get_location() (*in module salt.cloud.clouds.aliyun*), 1033
get_location() (*in module salt.cloud.clouds.azurearm*), 1037
get_location() (*in module salt.cloud.clouds.cloudstack*), 1039
get_location() (*in module salt.cloud.clouds.digitalocean*), 1041
get_location() (*in module salt.cloud.clouds.ec2*), 1048
get_location() (*in module salt.cloud.clouds.joyent*), 1063
get_location() (*in module salt.cloud.clouds.opennebula*), 1092
get_location() (*in module salt.cloud.clouds.softlayer*), 1128
get_location() (*in module salt.cloud.clouds.softlayer_hw*), 1129
get_location_path() (*in module salt.cloud.clouds.joyent*), 1063
get_log_format() (*in module salt.modules.win_smtp_server*), 2802
get_log_format_types() (*in module salt.modules.win_smtp_server*), 2802
get_logdb_quota() (*in module salt.modules.panos*), 2311
get_login_failed_count() (*in module salt.modules.mac_shadow*), 2042
get_login_failed_last() (*in module salt.modules.mac_shadow*), 2043
get_loginclass() (*in module salt.modules.pw_user*), 2408
get_loginclass() (*in module salt.modules.useradd*), 2633
get_machine_id() (*in module salt.grains.core*), 1178
get_machine_id() (*in module salt.modules.vagrant*), 2635
get_macs() (*in module salt.modules.smartos_virt*), 2503
get_macs() (*in module salt.modules.virt*), 2651
get_macs() (*in module salt.modules.xapi_virt*), 2854
get_makeopts() (*in module salt.modules.makeconf*), 2063
get_managed() (*in module salt.modules.file*), 1714
get_management_certificate() (*in module salt.cloud.clouds.msazure*), 1076
get_management_interface() (*in module salt.modules.cimc*), 1509
get_masquerade() (*in module salt.modules.firewalld*), 1733
get_master() (*in module salt.grains.core*), 1178
get_master_ip() (*in module salt.modules.redismod*), 2429
get_master_key() (*in module salt.modules.panos*), 2311
get_master_status() (*in module salt.modules.mysql*), 2101
get_max_op_version() (*in module salt.modules.glusterfs*), 1813
get_maxdays() (*in module salt.modules.mac_shadow*), 2043
get_memory_token() (*in module salt.modules.cimc*), 1509
get_memory_unit() (*in module salt.modules.cimc*), 1509
get_milestone() (*in module salt.modules.github*), 1803
get_milestones() (*in module salt.modules.github*), 1803
get_minions() (*in module salt.modules.ret*), 2441
get_minions() (*in module salt.returners.cassandra_cql_return*), 315
get_minions() (*in module salt.returners.couchdb_return*), 318
get_minions() (*in module salt.returners.etcd_return*), 321
get_minions() (*in module salt.returners.influxdb_return*), 323
get_minions() (*in module salt.returners.memcache_return*), 327
get_minions() (*in module salt.returners.mongo_future_return*), 329
get_minions() (*in module salt.returners.mysql*), 333
get_minions() (*in module salt.returners.odbc*), 336
get_minions() (*in module salt.returners.pgjsonb*), 339
get_minions() (*in module salt.returners.postgres*), 342
get_minions() (*in module salt.returners.redis_return*), 347
get_minions() (*in module salt.returners.sqlite3_return*), 354
get_missing_flags() (*in module*

salt.modules.portage_config), 2371
get_mode() (*in module salt.modules.file*), 1714
get_mode() (*in module salt.modules.quota*), 2411
get_mode() (*in module salt.modules.win_file*), 2736
get_modules() (*in module salt.modules.eselect*), 1695
get_modules_list() (*salt.modules.ansiblegate.AnsibleModuleResolver method*), 1208
get_monitor_timeout() (*in module salt.modules.win_powercfg*), 2787
get_mount_targets() (*in module salt.modules.boto_efs*), 1378
get_mpkg_ids() (*in module salt.modules.macpackage*), 2029
get_namespaces() (*in module salt.modules.k8s*), 1920
get_needs_reboot() (*in module salt.modules.win_wua*), 2842
get_net() (*in module salt.modules.peeringdb*), 2335
get_netfac() (*in module salt.modules.peeringdb*), 2335
get_netixlan() (*in module salt.modules.peeringdb*), 2336
get_network() (*in module salt.modules.infoblox*), 1871
get_network_adapters() (*in module salt.modules.cimc*), 1509
get_network_interface() (*in module salt.modules.boto_ec2*), 1370
get_network_interface_id() (*in module salt.modules.boto_ec2*), 1370
get_network_profile() (*in module salt.modules.lxc*), 1992
get_network_settings() (*in module salt.modules.debian_ip*), 1597
get_network_settings() (*in module salt.modules.nilrt_ip*), 2249
get_network_settings() (*in module salt.modules.rh_ip*), 2442
get_networkid() (*in module salt.cloud.clouds.cloudstack*), 1039
get_nics() (*in module salt.modules.virt*), 2652
get_nics() (*in module salt.modules.xapi_virt*), 2854
get_node() (*in module salt.cloud.clouds.cloudstack*), 1039
get_node() (*in module salt.cloud.clouds.dimensiondata*), 1044
get_node() (*in module salt.cloud.clouds.joyent*), 1063
get_node() (*in module salt.cloud.clouds.oneandone*), 1089
get_node() (*in module salt.cloud.clouds.profitbricks*), 1117
get_node_host() (*in module salt.modules.boto_elasticache*), 1382
get_notification_channel_id() (*in module salt.modules.telemetry*), 2599
get_ntp() (*in module salt.modules.cimc*), 1509
get_ntp_config() (*in module salt.modules.panos*), 2311
get_ntp_config() (*in module salt.modules.vsphere*), 2694
get_ntp_servers() (*in module salt.modules.panos*), 2311
get_object() (*in module salt.modules.infoblox*), 1871
get_object_id_by_params() (*in module salt.modules.zabbix*), 2879
get_offset() (*in module salt.modules.mac_timezone*), 2054
get_offset() (*in module salt.modules.timezone*), 2609
get_offset() (*in module salt.modules.win_timezone*), 2832
get_one_version() (*in module salt.cloud.clouds.opennebula*), 1092
get_op_version() (*in module salt.modules.glusterfs*), 1813
get_openstack_cloud() (*in module salt.modules.glanceng*), 1810
get_openstack_cloud() (*in module salt.modules.keystoneng*), 1938
get_openstack_cloud() (*in module salt.modules.neutronng*), 2239
get_operation_status() (*in module salt.cloud.clouds.msazure*), 1077
get_operational_mode() (*in module salt.modules.panos*), 2311
get_operator_cloud() (*in module salt.modules.glanceng*), 1810
get_operator_cloud() (*in module salt.modules.keystoneng*), 1939
get_operator_cloud() (*in module salt.modules.neutronng*), 2239
get_option() (*in module salt.modules.csf*), 1582
get_option() (*in module salt.modules.ini_manage*), 1873
get_opts() (*in module salt.modules.test*), 2601
get_opts() (*in module salt.runners.test*), 3156
get_or_set_hash() (*in module salt.modules.grains*), 1834
get_or_set_hash() (*in module salt.modules.sdb*), 2485
get_or_set_hash() (*in module salt.runners.sdb*), 3149
get_org() (*in module salt.modules.grafana4*), 1826
get_org() (*in module salt.modules.peeringdb*), 2336
get_org_address() (*in module salt.modules.grafana4*), 1827
get_org_details() (*in module salt.runners.digicertapi*), 3105

get_org_prefs() (*in module salt.modules.grafana4*), 1827
get_org_users() (*in module salt.modules.grafana4*), 1827
get_orgs() (*in module salt.modules.grafana4*), 1827
get_output_volume() (*in module salt.modules.mac_desktop*), 2026
get_package_info() (*in module salt.modules.win_pkg*), 2778
get_panorama_status() (*in module salt.modules.panos*), 2311
get_parameter() (*in module salt.modules.lxc*), 1993
get_password() (*in module salt.cloud.clouds.cloudstack*), 1039
get_password() (*in module salt.cloud.clouds.linode*), 1068
get_password_data() (*in module salt.cloud.clouds.ec2*), 1049
get_path() (*in module salt.modules.win_path*), 2776
get_pauses() (*in module salt.modules.state*), 2551
get_pci_adapters() (*in module salt.modules.cimc*), 1509
get_pem_entries() (*in module salt.modules.x509*), 2850
get_pem_entry() (*in module salt.modules.x509*), 2850
get_pending_component_servicing() (*in module salt.modules.win_system*), 2813
get_pending_computer_name() (*in module salt.modules.win_system*), 2813
get_pending_domain_join() (*in module salt.modules.win_system*), 2813
get_pending_file_rename() (*in module salt.modules.win_system*), 2813
get_pending_reboot() (*in module salt.modules.win_system*), 2813
get_pending_servermanager() (*in module salt.modules.win_system*), 2813
get_pending_update() (*in module salt.modules.win_system*), 2814
get_permission_types() (*in module salt.modules.win_snmp*), 2805
get_permissions() (*in module salt.modules.boto_lambda*), 1422
get_permitted_ips() (*in module salt.modules.panos*), 2312
get_pgid() (*in module salt.modules.win_file*), 2736
get_pgroup() (*in module salt.modules.win_file*), 2737
get_pid() (*in module salt.modules.lxc*), 1993
get_pid_list() (*in module salt.modules.ps*), 2390
get_pipeline_definition() (*in module salt.modules.boto_datapipeline*), 1358
get_pixel() (*in module salt.modules.sensehat*), 2492
get_pixels() (*in module salt.modules.sensehat*), 2492
get_pkg_id() (*in module salt.modules.macpackage*), 2029
get_placementgroup() (*in module salt.cloud.clouds.ec2*), 1049
get_plan_id() (*in module salt.cloud.clouds.linode*), 1068
get_platform() (*in module salt.modules.panos*), 2312
get_poc() (*in module salt.modules.peeringdb*), 2336
get_policy() (*in module salt.modules.boto_iam*), 1401
get_policy() (*in module salt.modules.iptables*), 1900
get_policy() (*in module salt.modules.kerberos*), 1924
get_policy() (*in module salt.modules.win_lgpo*), 2767
get_policy_config() (*in module salt.modules.capirca_acl*), 1474
get_policy_info() (*in module salt.modules.win_lgpo*), 2768
get_policy_version() (*in module salt.modules.boto_iam*), 1401
get_ports() (*in module salt.modules.csf*), 1582
get_power() (*in module salt.modules.ipmi*), 1887
get_power_configuration() (*in module salt.modules.cimc*), 1509
get_power_supplies() (*in module salt.modules.cimc*), 1509
get_predefined_application() (*in module salt.modules.panos*), 2312
get_pressure() (*in module salt.modules.sensehat*), 2492
get_principal() (*in module salt.modules.kerberos*), 1924
get_private_ip() (*in module salt.cloud.clouds.linode*), 1068
get_private_key_size() (*in module salt.modules.x509*), 2851
get_privs() (*in module salt.modules.kerberos*), 1924
get_profiles() (*in module salt.modules.virt*), 2652
get_project() (*in module salt.cloud.clouds.cloudstack*), 1039
get_provider() (*in module salt.cloud.clouds.ec2*), 1049
get_provider() (*in module salt.cloud.clouds.lxc*), 1070
get_provider_client() (*in module salt.cloud.clouds.tencentcloud*), 1131
get_proxy_bypass() (*in module salt.modules.proxy*), 2388
get_proxy_type() (*in module salt.modules.vsphere*), 2694
get_proxy_win() (*in module salt.modules.proxy*), 2388
get_prs() (*in module salt.modules.github*), 1804
get_pub_key() (*in module salt.cloud.clouds.linode*),

1068
`get_public_key()` (*in module salt.modules.x509*), 2851
`get_public_keys()` (*in module salt.cloud.clouds.profitbricks*), 1117
`get_pv_args()` (*in module salt.cloud.clouds.xen*), 1150
`get_quotas_tenant()` (*in module salt.modules.neutron*), 2229
`get_reboot_required_witnessed()` (*in module salt.modules.system*), 2588
`get_reboot_required_witnessed()` (*in module salt.modules.win_system*), 2814
`get_record()` (*in module salt.modules.boto_route53*), 1433
`get_record()` (*in module salt.modules.libcloud_dns*), 1967
`get_relay_ip_list()` (*in module salt.modules.win_smtp_server*), 2802
`get_release()` (*in module salt.modules.artifactory*), 1241
`get_release_number()` (*in module salt.modules.salt_version*), 2463
`get_remote_events()` (*in module salt.modules.mac_system*), 2049
`get_remote_login()` (*in module salt.modules.mac_system*), 2049
`get_repo()` (*in module salt.modules.aptpkg*), 1217
`get_repo()` (*in module salt.modules.opkg*), 2281
`get_repo()` (*in module salt.modules.xbpspkg*), 2858
`get_repo()` (*in module salt.modules.yumpkg*), 2865
`get_repo()` (*in module salt.modules.zypperpkg*), 2925
`get_repo_data()` (*in module salt.modules.win_pkg*), 2778
`get_repo_info()` (*in module salt.modules.github*), 1804
`get_repo_keys()` (*in module salt.modules.aptpkg*), 1217
`get_repo_teams()` (*in module salt.modules.github*), 1804
`get_repository()` (*in module salt.modules.win_psget*), 2790
`get_resource()` (*in module salt.modules.pagerduty_util*), 2304
`get_resource_by_id()` (*in module salt.cloud.clouds.azurearm*), 1037
`get_resource_content()` (*in module salt.modules.virtualenv_mod*), 2677
`get_resource_id()` (*in module salt.modules.boto_vpc*), 1458
`get_resource_path()` (*in module salt.modules.virtualenv_mod*), 2677
`get_resource_records()` (*in module salt.modules.boto3_route53*), 1328
`get_resources_nodes()` (*in module salt.cloud.clouds.proxmox*), 1120
`get_resources_vms()` (*in module salt.cloud.clouds.proxmox*), 1120
`get_restart_delay()` (*in module salt.modules.mac_system*), 2049
`get_restart_freeze()` (*in module salt.modules.mac_power*), 2035
`get_restart_power_failure()` (*in module salt.modules.mac_power*), 2035
`get_retention_policy()` (*in module salt.modules.influxdbmod*), 1860
`get_rich_rules()` (*in module salt.modules.firewalld*), 1733
`get_role_policy()` (*in module salt.modules.boto_iam*), 1401
`get_roles()` (*in module salt.proxy.nxos*), 3075
`get_rollback()` (*in module salt.modules.cisconso*), 1519
`get_rollback()` (*in module salt.proxy.cisconso*), 3056
`get_rollbacks()` (*in module salt.modules.cisconso*), 1519
`get_rollbacks()` (*in module salt.proxy.cisconso*), 3056
`get_root_path()` (*in module salt.modules.lxc*), 1993
`get_route()` (*in module salt.modules.network*), 2217
`get_route()` (*in module salt.modules.win_network*), 2773
`get_routes()` (*in module salt.modules.debian_ip*), 1597
`get_routes()` (*in module salt.modules.linux_ip*), 1976
`get_routes()` (*in module salt.modules.rh_ip*), 2442
`get_rule()` (*in module salt.modules.win_firewall*), 2748
`get_rule_handle()` (*in module salt.modules.nftables*), 2246
`get_rules()` (*in module salt.modules.iptables*), 1900
`get_rules()` (*in module salt.modules.lvs*), 1985
`get_rules()` (*in module salt.modules.nftables*), 2246
`get_running()` (*in module salt.modules.modjk*), 2084
`get_running()` (*in module salt.modules.smf_service*), 2511
`get_running()` (*in module salt.modules.systemd_service*), 2594
`get_saml_provider()` (*in module salt.modules.boto_iam*), 1402
`get_saml_provider_arn()` (*in module salt.modules.boto_iam*), 1402
`get_saved_policy()` (*in module salt.modules.iptables*), 1901
`get_saved_rules()` (*in module salt.modules.iptables*), 1901
`get_saved_rules()` (*in module salt.modules.nftables*), 2246
`get_scaling_policy_arn()` (*in module*

salt.modules.boto_asg), 1343
get_schedules() (*in module salt.modules.pagerduty_util*), 2304
get_sd_auth() (*in module salt.modules.serverdensity_device*), 2494
get_secgroup_id() (*in module salt.cloud.clouds.opennebula*), 1092
get_secret_key() (*in module salt.modules.gpg*), 1820
get_secrets() (*in module salt.modules.k8s*), 1921
get_section() (*in module salt.modules.ini_manage*), 1873
get_security_groups() (*in module salt.cloud.clouds.cloudstack*), 1039
get_security_rule() (*in module salt.modules.panos*), 2312
get_securitygroup() (*in module salt.cloud.clouds.aliyun*), 1033
get_selections() (*in module salt.modules.aptpkg*), 1218
get_selections() (*in module salt.modules.debconfmod*), 1595
get_selinux_context() (*in module salt.modules.file*), 1714
get_sensor_data() (*in module salt.modules.ipmi*), 1887
get_serialized_facts() (*in module salt.proxy.junos*), 3065
get_server_certificate() (*in module salt.modules.boto_iam*), 1402
get_server_id() (*in module salt.grains.core*), 1178
get_server_setting() (*in module salt.modules.win_smtp_server*), 2803
get_servers() (*in module salt.modules.win_ntp*), 2776
get_servers() (*in module salt.modules.win_smtp_server*), 2803
get_service() (*in module salt.modules.panos*), 2312
get_service_certificate() (*in module salt.cloud.clouds.msazure*), 1077
get_service_group() (*in module salt.modules.panos*), 2312
get_service_instance_via_proxy() (*in module salt.modules.vsphere*), 2695
get_service_name() (*in module salt.modules.win_service*), 2797
get_service_policy() (*in module salt.modules.vsphere*), 2695
get_service_ports() (*in module salt.modules.firewalld*), 1734
get_service_protocols() (*in module salt.modules.firewalld*), 1734
get_service_running() (*in module salt.modules.vsphere*), 2695
get_services() (*in module salt.modules.firewalld*), 1734
get_services() (*in module salt.modules.pagerduty_util*), 2304
get_session() (*in module salt.modules.rdp*), 2426
get_session_info() (*in module salt.modules.panos*), 2313
get_sessions() (*in module salt.modules.haproxyconn*), 1840
get_settings() (*in module salt.modules.win_firewall*), 2748
get_signing_policy() (*in module salt.modules.x509*), 2851
get_site_packages() (*in module salt.modules.virtualenv_mod*), 2678
get_size() (*in module salt.cloud.clouds.aliyun*), 1033
get_size() (*in module salt.cloud.clouds.cloudstack*), 1039
get_size() (*in module salt.cloud.clouds.digitalocean*), 1041
get_size() (*in module salt.cloud.clouds.dimensiondata*), 1044
get_size() (*in module salt.cloud.clouds.joyent*), 1063
get_size() (*in module salt.cloud.clouds.oneandone*), 1089
get_size() (*in module salt.cloud.clouds.profitbricks*), 1118
get_skipped_nics() (*in module salt.modules.csf*), 1583
get_slack_channels() (*salt.engines.slack.SlackClient method*), 1167
get_slack_users() (*salt.engines.slack.SlackClient method*), 1167
get_slave_status() (*in module salt.modules.mysql*), 2101
get_sleep() (*in module salt.modules.mac_power*), 2035
get_sleep_on_power_button() (*in module salt.modules.mac_power*), 2035
get_slotname() (*in module salt.modules.dracr*), 1667
get_snapshot() (*in module salt.modules.artifactory*), 1242
get_snapshot() (*in module salt.modules.snapper*), 2545
get_snmp_config() (*in module salt.modules.cimc*), 1510
get_snmp_config() (*in module salt.modules.panos*), 2313
get_software_info() (*in module salt.modules.panos*), 2313
get_source_sum() (*in module salt.modules.file*), 1714
get_sources() (*in module salt.modules.firewalld*), 1734

get_spot_config() (*in module salt.cloud.clouds.ec2*), 1049
get_ssh_config() (*in module salt.modules.vagrant*), 2635
get_ssh_gateway_config() (*in module salt.cloud.clouds.ec2*), 1049
get_ssh_key() (*in module salt.modules.vsphere*), 2696
get_standby_timeout() (*in module salt.modules.win_powercfg*), 2788
get_startup_disk() (*in module salt.modules.mac_system*), 2050
get_static() (*in module salt.modules.systemd_service*), 2594
get_stopped() (*in module salt.modules.smf_service*), 2511
get_storage() (*in module salt.cloud.clouds.msazure*), 1077
get_storage_conn() (*in module salt.cloud.clouds.msazure*), 1077
get_storage_container() (*in module salt.cloud.clouds.msazure*), 1077
get_storage_container_acl() (*in module salt.cloud.clouds.msazure*), 1077
get_storage_container_metadata() (*in module salt.cloud.clouds.msazure*), 1078
get_storage_id() (*in module salt.cache.localfs*), 1028
get_storage_keys() (*in module salt.cloud.clouds.msazure*), 1078
get_stored_cert_serials() (*in module salt.modules.win_certutil*), 2721
get_stores() (*in module salt.modules.win_pki*), 2786
get_str() (*in module salt.modules.mod_random*), 2082
get_stream_when_active() (*in module salt.modules.boto_kinesis*), 1414
get_subnet_association() (*in module salt.modules.boto_vpc*), 1459
get_subnet_length() (*in module salt.modules.win_ip*), 2764
get_subnet_name() (*in module salt.modules.mac_system*), 2050
get_subnetid() (*in module salt.cloud.clouds.ec2*), 1049
get_sum() (*in module salt.modules.file*), 1715
get_svc_alias() (*in module salt.modules.runit*), 2454
get_svc_avail_path() (*in module salt.modules.runit*), 2454
get_svc_broken_path() (*in module salt.modules.runit*), 2454
get_swap_size() (*in module salt.cloud.clouds.linode*), 1068
get_sync() (*in module salt.modules.makeconf*), 2064
get_sys() (*in module salt.modules.keyboard*), 1928
get_syslog() (*in module salt.modules.cimc*), 1510
get_syslog_config() (*in module salt.modules.vsphere*), 2696
get_syslog_settings() (*in module salt.modules.cimc*), 1510
get_system_date() (*in module salt.modules.system*), 2589
get_system_date() (*in module salt.modules.win_system*), 2814
get_system_date_time() (*in module salt.modules.panos*), 2313
get_system_date_time() (*in module salt.modules.system*), 2589
get_system_files() (*in module salt.modules.panos*), 2313
get_system_info() (*in module salt.modules.cimc*), 1510
get_system_info() (*in module salt.modules.panos*), 2313
get_system_info() (*in module salt.modules.win_system*), 2814
get_system_services() (*in module salt.modules.panos*), 2313
get_system_state() (*in module salt.modules.panos*), 2313
get_system_time() (*in module salt.modules.system*), 2589
get_system_time() (*in module salt.modules.win_system*), 2814
get_tags() (*in module salt.cloud.clouds.ec2*), 1049
get_tags() (*in module salt.modules.boto_ec2*), 1370
get_tags() (*in module salt.modules.boto_efs*), 1378
get_target() (*in module salt.modules.aliases*), 1206
get_target() (*salt.engines.slack.SlackClient method*), 1167
get_target_list() (*in module salt.modules.eselect*), 1695
get_task() (*in module salt.modules.kapacitor*), 1923
get_team() (*in module salt.modules.github*), 1805
get_temperature() (*in module salt.modules.sensehat*), 2492
get_temperature_from_humidity() (*in module salt.modules.sensehat*), 2493
get_temperature_from_pressure() (*in module salt.modules.sensehat*), 2493
get_template() (*in module salt.cloud.clouds.opennebula*), 1092
get_template() (*in module salt.modules.boto_cfn*), 1345
get_template() (*in module salt.modules.cp*), 1576
get_template_id() (*in module salt.cloud.clouds.opennebula*), 1093
get_template_image() (*in module*

salt.cloud.clouds.opennebula), 1093
`get_tenancy()` (*in module salt.cloud.clouds.ec2*), 1049
`get_term_config()` (*in module salt.modules.capirca_acl*), 1476
`get_term_pillar()` (*in module salt.modules.capirca_acl*), 1481
`get_term_pillar()` (*in module salt.modules.napalm_netacl*), 2139
`get_test_string()` (*in module salt.modules.rest_sample_utils*), 2439
`get_testing_status()` (*in module salt.modules.csf*), 1583
`get_time()` (*in module salt.modules.mac_timezone*), 2054
`get_time_server()` (*in module salt.modules.mac_timezone*), 2054
`get_tld_list()` (*in module salt.modules.namecheap_domains*), 2116
`get_token()` (*in module salt.tokens.localfs*), 3828
`get_token()` (*in module salt.tokens.rediscluster*), 3829
`get_uid()` (*in module salt.modules.file*), 1715
`get_uid()` (*in module salt.modules.win_file*), 2737
`get_unassociated_eip_address()` (*in module salt.modules.boto_ec2*), 1370
`get_uncommitted_changes()` (*in module salt.modules.panos*), 2314
`get_unix()` (*salt.modules.win_timezone.TzMapper method*), 2831
`get_upgrade_history()` (*in module salt.modules.boto3_elasticsearch*), 1316
`get_upgrade_status()` (*in module salt.modules.boto3_elasticsearch*), 1316
`get_url()` (*in module salt.modules.cp*), 1576
`get_user()` (*in module salt.modules.boto_iam*), 1402
`get_user()` (*in module salt.modules.file*), 1716
`get_user()` (*in module salt.modules.github*), 1805
`get_user()` (*in module salt.modules.grafana4*), 1827
`get_user()` (*in module salt.modules.ilo*), 1853
`get_user()` (*in module salt.modules.ipmi*), 1887
`get_user()` (*in module salt.modules.pdbedit*), 2332
`get_user()` (*in module salt.modules.splunk*), 2536
`get_user()` (*in module salt.modules.win_file*), 2737
`get_user()` (*in module salt.proxy.nxos*), 3075
`get_user_access()` (*in module salt.modules.ipmi*), 1888
`get_user_data()` (*in module salt.modules.grafana4*), 1827
`get_user_name()` (*in module salt.modules.ipmi*), 1889
`get_user_orgs()` (*in module salt.modules.grafana4*), 1828
`get_user_policy()` (*in module salt.modules.boto_iam*), 1402
`get_users()` (*in module salt.modules.cimc*), 1510
`get_users()` (*in module salt.modules.grafana4*), 1828
`get_users()` (*in module salt.modules.ipmi*), 1889
`get_users()` (*in module salt.modules.pagerduty_util*), 2304
`get_users()` (*in module salt.modules.ps*), 2390
`get_users_config()` (*in module salt.modules.panos*), 2314
`get_using_network_time()` (*in module salt.modules.mac_timezone*), 2054
`get_valid_salt_views()` (*in module salt.returners.couchdb_return*), 318
`get_value()` (*in module salt.modules.xml*), 2862
`get_var()` (*in module salt.modules.makeconf*), 2064
`get_vcenter_version()` (*in module salt.cloud.clouds.vmware*), 1141
`get_version()` (*in module salt.modules.glusterfs*), 1813
`get_version()` (*in module salt.modules.jenkinsmod*), 1910
`get_vic_adapters()` (*in module salt.modules.cimc*), 1510
`get_vic_uplinks()` (*in module salt.modules.cimc*), 1510
`get_virtual_machine_scale_set_network_interface()` (*in module salt.modules.azurearm_network*), 1255
`get_vlans()` (*in module salt.modules.panos*), 2314
`get_vm()` (*in module salt.modules.vsphere*), 2697
`get_vm_config()` (*in module salt.modules.vsphere*), 2697
`get_vm_config_file()` (*in module salt.modules.vsphere*), 2697
`get_vm_id()` (*in module salt.cloud.clouds.opennebula*), 1093
`get_vm_info()` (*in module salt.modules.vagrant*), 2635
`get_vm_ip()` (*in module salt.cloud.clouds.xen*), 1150
`get_vm_size()` (*in module salt.cloud.clouds.linode*), 1068
`get_vm_status()` (*in module salt.cloud.clouds.proxmox*), 1120
`get_vmconfig()` (*in module salt.cloud.clouds.proxmox*), 1120
`get_vmotion_enabled()` (*in module salt.modules.vsphere*), 2697
`get_vn_id()` (*in module salt.cloud.clouds.opennebula*), 1093
`get_vsan_eligible_disks()` (*in module salt.modules.vsphere*), 2698
`get_vsan_enabled()` (*in module salt.modules.vsphere*), 2698
`get_wait_timeout()` (*in module salt.cloud.clouds.oneandone*), 1089
`get_wait_timeout()` (*in module*

salt.cloud.clouds.profitbricks), 1118
`get_wake_on_modem()` (*in module salt.modules.mac_power*), 2036
`get_wake_on_network()` (*in module salt.modules.mac_power*), 2036
`get_webapp_settings()` (*in module salt.modules.win_iis*), 2757
`get_webconfiguration_settings()` (*in module salt.modules.win_iis*), 2758
`get_weight()` (*in module salt.modules.haproxyconn*), 1841
`get_win()` (*salt.modules.win_timezone.TzMapper method*), 2831
`get_wu_settings()` (*in module salt.modules.win_wua*), 2842
`get_x()` (*in module salt.modules.keyboard*), 1928
`get_xml()` (*in module salt.modules.virt*), 2652
`get_xpath()` (*in module salt.modules.panos*), 2314
`get_yaml_loader()` (*in module salt.renderers.yaml*), 385
`get_zabbix_id_mapper()` (*in module salt.modules.zabbix*), 2879
`get_zone()` (*in module salt.modules.libcloud_dns*), 1967
`get_zone()` (*in module salt.modules.mac_timezone*), 2054
`get_zone()` (*in module salt.modules.panos*), 2314
`get_zone()` (*in module salt.modules.timezone*), 2609
`get_zone()` (*in module salt.modules.win_timezone*), 2832
`get_zonecode()` (*in module salt.modules.mac_timezone*), 2055
`get_zonecode()` (*in module salt.modules.timezone*), 2609
`get_zonecode()` (*in module salt.modules.win_timezone*), 2832
`get_zones()` (*in module salt.modules.boto_ec2*), 1370
`get_zones()` (*in module salt.modules.firewalld*), 1734
`get_zones()` (*in module salt.modules.panos*), 2314
`getAceTypeBit()` (*salt.modules.win_dacl.daclConstants method*), 2722
`getAceTypeText()` (*salt.modules.win_dacl.daclConstants method*), 2722
`getClockFormat()` (*in module salt.modules.gnomedesktop*), 1815
`getClockShowDate()` (*in module salt.modules.gnomedesktop*), 1816
`getconfig()` (*in module salt.modules.selinux*), 2489
`getenforce()` (*in module salt.modules.selinux*), 2489
`getent()` (*in module salt.modules.aix_group*), 1203
`getent()` (*in module salt.modules.groupadd*), 1837
`getent()` (*in module salt.modules.mac_group*), 2027
`getent()` (*in module salt.modules.mac_user*), 2058
`getent()` (*in module salt.modules.pw_group*), 2406
`getent()` (*in module salt.modules.pw_user*), 2408
`getent()` (*in module salt.modules.solaris_group*), 2515
`getent()` (*in module salt.modules.solaris_user*), 2520
`getent()` (*in module salt.modules.useradd*), 2633
`getent()` (*in module salt.modules.win_groupadd*), 2753
`getent()` (*in module salt.modules.win_useradd*), 2835
`getfacl()` (*in module salt.modules.linux_acl*), 1975
`getgoal()` (*in module salt.modules.moosefs*), 2091
`getIdleActivation()` (*in module salt.modules.gnomedesktop*), 1816
`getIdleDelay()` (*in module salt.modules.gnomedesktop*), 1816
`getObjectTypeBit()` (*salt.modules.win_dacl.daclConstants method*), 2722
`getos()` (*in module salt.grains.napalm*), 1181
`getPermissionBit()` (*salt.modules.win_dacl.daclConstants method*), 2722
`getPermissionText()` (*salt.modules.win_dacl.daclConstants method*), 2722
`getPropagationBit()` (*salt.modules.win_dacl.daclConstants method*), 2722
`getPropagationText()` (*salt.modules.win_dacl.daclConstants method*), 2722
`gets_service_instance_via_proxy()` (*in module salt.modules.vsphere*), 2699
`getsebool()` (*in module salt.modules.selinux*), 2489
`getSecurityHkey()` (*salt.modules.win_dacl.daclConstants method*), 2722
`getsemod()` (*in module salt.modules.selinux*), 2489
`getsid()` (*in module salt.modules.win_service*), 2798
`getUsage()` (*in module salt.modules.random_org*), 2422
`getUserSid()` (*in module salt.modules.win_useradd*), 2835
`gid_to_group()` (*in module salt.modules.file*), 1716
`gid_to_group()` (*in module salt.modules.win_file*), 2738
`git_pillar_base`
    `conf/master`, 102
`git_pillar_branch`
    `conf/master`, 102
`git_pillar_env`
    `conf/master`, 103
`git_pillar_global_lock`
    `conf/master`, 104
`git_pillar_includes`
    `conf/master`, 104
`git_pillar_insecure_auth`

conf/master, 105
git_pillar_passphrase
conf/master, 106
git_pillar_password
conf/master, 105
git_pillar_privkey
conf/master, 106
git_pillar_provider
conf/master, 102
git_pillar_pubkey
conf/master, 106
git_pillar_refspecs
conf/master, 106
git_pillar_root
conf/master, 103
git_pillar_ssl_verify
conf/master, 104
git_pillar_user
conf/master, 105
git_pillar_verify_config
conf/master, 106
gitfs_base
conf/master, 86
gitfs_disable_saltenv_mapping
conf/master, 87
gitfs_global_lock
conf/master, 88
gitfs_insecure_auth
conf/master, 89
gitfs_mountpoint
conf/master, 86
gitfs_passphrase
conf/master, 90
gitfs_password
conf/master, 89
gitfs_privkey
conf/master, 90
gitfs_provider
conf/master, 85
gitfs_pubkey
conf/master, 89
gitfs_ref_types
conf/master, 87
gitfs_refspecs
conf/master, 90
gitfs_remotes
conf/master, 85
gitfs_root
conf/master, 86
gitfs_saltenv
conf/master, 86
gitfs_saltenv_blacklist
conf/master, 88
gitfs_saltenv_whitelist
conf/master, 87
gitfs_ssl_verify
conf/master, 85
gitfs_update_interval
conf/master, 88
gitfs_user
conf/master, 89
github_signature() (*in module salt.modules.hashutil*), 1844
GitLockError, 3998
GitRemoteError, 3998
GivenStatement (*class in salt.modules.syslog_ng*), 2578
glob() (*in module salt.modules.match*), 2071
global_settings() (*in module salt.modules.ilo*), 1853
glsa_check_list() (*in module salt.modules.gentoolkitmod*), 1757
Grain, **5401**
grain() (*in module salt.modules.match*), 2071
grain_funcs() (*in module salt.loader*), 3845
grain_pcre() (*in module salt.modules.match*), 2071
grains
conf/minion, 133
grains() (*in module salt.loader*), 3845
grains() (*in module salt.proxy.cimc*), 3051
grains() (*in module salt.proxy.cisconso*), 3056
grains() (*in module salt.proxy.dummy*), 3057
grains() (*in module salt.proxy.esxi*), 3061
grains() (*in module salt.proxy.fx2*), 3064
grains() (*in module salt.proxy.nxos*), 3075
grains() (*in module salt.proxy.panos*), 3081
grains() (*in module salt.proxy.rest_sample*), 3085
grains() (*in module salt.proxy.ssh_sample*), 3086
grains() (*in module salt.runners.cache*), 3101
grains_blacklist
conf/minion, 133
grains_cache
conf/minion, 133
grains_deep_merge
conf/minion, 133
grains_dirs
conf/minion, 145
grains_refresh() (*in module salt.proxy.cimc*), 3052
grains_refresh() (*in module salt.proxy.dummy*), 3057
grains_refresh() (*in module salt.proxy.esxi*), 3061
grains_refresh() (*in module salt.proxy.fx2*), 3064
grains_refresh() (*in module salt.proxy.napalm*), 3069
grains_refresh() (*in module salt.proxy.nxos*), 3075
grains_refresh() (*in module salt.proxy.panos*), 3081

grains_refresh() (*in module salt.proxy.rest_sample*), 3085
grains_refresh() (*in module salt.proxy.ssh_sample*), 3086
grains_refresh_every
    conf/minion, 134
grant_access_to_shared_folders_to() (*in module salt.modules.vbox_guest*), 2642
grant_access_to_shared_folders_to() (*in module salt.states.vbox_guest*), 3747
grant_add() (*in module salt.modules.mysql*), 2102
grant_admin_privileges() (*in module salt.modules.influxdbmod*), 1860
grant_exists() (*in module salt.modules.mysql*), 2102
grant_permission() (*in module salt.modules.cassandra_cql*), 1486
grant_privilege() (*in module salt.modules.influxdbmod*), 1860
grant_revoke() (*in module salt.modules.mysql*), 2102
greater_than() (*in module salt.modules.salt_version*), 2463
grep() (*in module salt.modules.file*), 1716
group_absent() (*in module salt.states.boto_iam*), 3295
group_create() (*in module salt.modules.keystoneng*), 1939
group_create() (*in module salt.modules.postgres*), 2376
group_delete() (*in module salt.modules.keystoneng*), 1939
group_diff() (*in module salt.modules.pacmanpkg*), 2297
group_diff() (*in module salt.modules.yumpkg*), 2865
group_exists() (*in module salt.modules.boto_elasticache*), 1382
group_get() (*in module salt.modules.keystoneng*), 1939
group_info() (*in module salt.modules.pacmanpkg*), 2298
group_info() (*in module salt.modules.yumpkg*), 2865
group_install() (*in module salt.modules.yumpkg*), 2865
group_installed() (*in module salt.states.pkg*), 3644
group_list() (*in module salt.modules.keystoneng*), 1939
group_list() (*in module salt.modules.pacmanpkg*), 2298
group_list() (*in module salt.modules.yumpkg*), 2866
group_present() (*in module salt.states.boto_iam*), 3295
group_remove() (*in module salt.modules.postgres*), 2377
group_search() (*in module salt.modules.keystoneng*), 1939
group_to_gid() (*in module salt.modules.file*), 1716
group_to_gid() (*in module salt.modules.win_file*), 2738
group_update() (*in module salt.modules.keystoneng*), 1939
group_update() (*in module salt.modules.postgres*), 2377
groups() (*in module salt.auth.ldap*), 1001
groups() (*in module salt.auth.pam*), 1003
groups() (*in module salt.auth.yubico*), 1005
groups() (*in module salt.modules.osquery*), 2290
GsiNotUpdatableError, 3272
gt() (*in module salt.thorium.check*), 3821
gte() (*in module salt.thorium.check*), 3821
gunzip() (*in module salt.modules.archive*), 1228
gzip() (*in module salt.modules.archive*), 1228

H

halt() (*in module salt.modules.mac_system*), 2050
halt() (*in module salt.modules.solaris_system*), 2517
halt() (*in module salt.modules.system*), 2589
halt() (*in module salt.modules.win_system*), 2815
halt() (*in module salt.modules.zoneadm*), 2908
halted() (*in module salt.states.zone*), 3811
handle (*salt.auth.pam.PamHandle attribute*), 1003
handle_queue_creation() (*in module salt.queues.pgjsonb_queue*), 3087
handle_snapshot() (*in module salt.cloud.clouds.vmware*), 1141
hardlink() (*in module salt.states.file*), 3429
hardware_events() (*in module salt.modules.osquery*), 2290
has() (*in module salt.modules.acme*), 1202
has_app() (*in module salt.modules.marathon*), 2070
has_exec() (*in module salt.modules.cmdmod*), 1523
has_flag() (*in module salt.modules.portage_config*), 2371
has_instance() (*in module salt.modules.cloud*), 1520
has_job() (*in module salt.modules.chronos*), 1506
has_key() (*in module salt.modules.data*), 1587
has_method() (*in module salt.cloud.clouds.joyent*), 1063
has_pair() (*in module salt.modules.hosts*), 1850
has_powerpath() (*in module salt.modules.powerpath*), 2387
has_privileges() (*in module salt.modules.postgres*), 2377
has_settable_hwclock() (*in module salt.modules.system*), 2589
has_target() (*in module salt.modules.aliases*), 1206
has_use() (*in module salt.modules.portage_config*), 2372

has_value() (*in module salt.modules.environ*), 1693
has_value() (*in module salt.modules.grains*), 1835
hash() (*in module salt.modules.mod_random*), 2082
hash() (*in module salt.runners.survey*), 3155
hash_() (*in module salt.modules.osquery*), 2290
hash_file() (*in module salt.modules.cp*), 1577
hash_known_hosts() (*in module salt.modules.ssh*), 2540
hash_type
  conf/master, 83
  conf/minion, 153
Hashabledict (*class in salt.beacons.network_settings*), 1018
hdel() (*in module salt.modules.redismod*), 2429
hdparms() (*in module salt.modules.disk*), 1603
head() (*in module salt.modules.s3*), 2460
head() (*in module salt.modules.swift*), 2574
health_checks() (*in module salt.modules.consul*), 1569
health_node() (*in module salt.modules.consul*), 1569
health_service() (*in module salt.modules.consul*), 1570
health_state() (*in module salt.modules.consul*), 1570
healthy() (*in module salt.modules.solaris_fmadm*), 2514
healthy() (*in module salt.modules.zpool*), 2919
held() (*in module salt.states.aptpkg*), 3188
help() (*in module salt.modules.ansiblegate*), 1208
hexists() (*in module salt.modules.redismod*), 2429
hg_create() (*in module salt.modules.purefa*), 2397
hg_delete() (*in module salt.modules.purefa*), 2397
hg_remove() (*in module salt.modules.purefa*), 2397
hg_update() (*in module salt.modules.purefa*), 2397
hget() (*in module salt.modules.redismod*), 2429
hgetall() (*in module salt.modules.redismod*), 2429
hgfs_base
  conf/master, 92
hgfs_branch_method
  conf/master, 91
hgfs_mountpoint
  conf/master, 92
hgfs_remotes
  conf/master, 91
hgfs_root
  conf/master, 92
hgfs_saltenv_blacklist
  conf/master, 93
hgfs_saltenv_whitelist
  conf/master, 92
hgfs_update_interval
  conf/master, 93
high() (*in module salt.modules.state*), 2551
Highdata, **5401**
Highstate, **5401**
highstate() (*in module salt.modules.dockermod*), 1639
highstate() (*in module salt.modules.state*), 2551
hincrby() (*in module salt.modules.redismod*), 2429
hincrbyfloat() (*in module salt.modules.redismod*), 2429
history() (*in module salt.modules.dockermod*), 1639
history() (*in module salt.modules.zpool*), 2919
hlen() (*in module salt.modules.redismod*), 2430
hmac
  jinja filters, 441
hmac_compute() (*in module salt.modules.hashutil*), 1844
hmac_signature() (*in module salt.modules.hashutil*), 1844
hmget() (*in module salt.modules.redismod*), 2430
hmset() (*in module salt.modules.redismod*), 2430
hold() (*in module salt.modules.aptpkg*), 1218
hold() (*in module salt.modules.opkg*), 2281
hold() (*in module salt.modules.pkgng*), 2357
hold() (*in module salt.modules.postfix*), 2372
hold() (*in module salt.modules.yumpkg*), 2866
hold() (*in module salt.modules.zfs*), 2901
hold() (*in module salt.thorium.timer*), 3828
hold_absent() (*in module salt.states.zfs*), 3808
hold_present() (*in module salt.states.zfs*), 3808
holds() (*in module salt.modules.zfs*), 2901
homebrew_packages() (*in module salt.modules.osquery*), 2290
host (*salt.engines.ircbot.PrivEvent attribute*), 1156
host() (*in module salt.grains.napalm*), 1181
host() (*in module salt.proxy.fx2*), 3064
host_cache_configured() (*in module salt.states.esxi*), 3411
host_create() (*in module salt.modules.purefa*), 2397
host_create() (*in module salt.modules.zabbix*), 2879
host_delete() (*in module salt.modules.purefa*), 2398
host_delete() (*in module salt.modules.zabbix*), 2880
host_dns() (*in module salt.grains.napalm*), 1182
host_exists() (*in module salt.modules.zabbix*), 2880
host_get() (*in module salt.modules.zabbix*), 2880
host_info() (*in module salt.runners.virt*), 3158
host_inventory_get() (*in module salt.modules.zabbix*), 2881
host_inventory_set() (*in module salt.modules.zabbix*), 2881
host_keys() (*in module salt.modules.ssh*), 2540
host_list() (*in module salt.cloud.clouds.xen*), 1150
host_list() (*in module salt.modules.zabbix*), 2881
host_status() (*in module salt.modules.nagios_rpc*), 2113
host_update() (*in module salt.modules.purefa*), 2398
host_update() (*in module salt.modules.zabbix*), 2882

hosted_zone_absent() (in module salt.states.boto3_route53), 3247
hosted_zone_absent() (in module salt.states.boto_route53), 3317
hosted_zone_present() (in module salt.states.boto3_route53), 3247
hosted_zone_present() (in module salt.states.boto_route53), 3317
hostgroup_create() (in module salt.modules.zabbix), 2882
hostgroup_delete() (in module salt.modules.zabbix), 2882
hostgroup_exists() (in module salt.modules.zabbix), 2883
hostgroup_get() (in module salt.modules.zabbix), 2883
hostgroup_list() (in module salt.modules.zabbix), 2884
hostgroup_update() (in module salt.modules.zabbix), 2884
hostinterface_create() (in module salt.modules.zabbix), 2884
hostinterface_delete() (in module salt.modules.zabbix), 2885
hostinterface_get() (in module salt.modules.zabbix), 2885
hostinterface_update() (in module salt.modules.zabbix), 2885
hostname() (in module salt.grains.core), 1178
hostname() (in module salt.grains.napalm), 1182
hostname() (in module salt.states.cimc), 3340
hostname() (in module salt.states.win_system), 3782
hosts_append() (in module salt.modules.dnsutil), 1609
hosts_remove() (in module salt.modules.dnsutil), 1609
hpa() (in module salt.modules.disk), 1603
hscan() (in module salt.modules.redismod), 2430
hset() (in module salt.modules.redismod), 2430
hsetnx() (in module salt.modules.redismod), 2430
html_format() (in module salt.modules.out), 2296
http_connect_timeout
    conf/master, 64
    conf/minion, 141
http_event_collector (class in salt.returners.splunk), 352
http_query
    jinja filters, 441
http_request_timeout
    conf/master, 64
    conf/minion, 141
hvals() (in module salt.modules.redismod), 2430
hw_addr() (in module salt.modules.network), 2217
hw_addr() (in module salt.modules.win_network), 2773
hwaddr() (in module salt.modules.network), 2217
hwaddr() (in module salt.modules.win_network), 2773
hwaddr_interfaces() (in module salt.grains.core), 1178

I

iam_profile() (in module salt.cloud.clouds.ec2), 1049
id
    conf/minion, 130
id() (in module salt.proxy.rest_sample), 3085
id_() (in module salt.grains.core), 1178
id_() (in module salt.modules.monit), 2089
id_exists() (in module salt.modules.state), 2552
Idempotent, **5401**
IdleProcessTime (salt.modules.win_status.SYSTEM_PERFORMANCE_I attribute), 2808
idrac_general() (in module salt.modules.dracr), 1667
ids() (in module salt.modules.freebsd_update), 1738
ifacestartswith() (in module salt.modules.network), 2217
ignore() (in module salt.modules.mac_softwareupdate), 2046
IGNORE_FS_TYPES (salt.modules.inspectlib.collector.Inspector attribute), 1874
IGNORE_MOUNTS (salt.modules.inspectlib.collector.Inspector attribute), 1874
ignore_null (salt.pillar.sql_base.SqlBaseExtPillar attribute), 3035
IGNORE_PATHS (salt.modules.inspectlib.collector.Inspector attribute), 1874
IgnoredDir (class in salt.modules.inspectlib.entities), 1876
image_absent() (in module salt.states.smartos), 3716
image_alias_add() (in module salt.modules.lxd), 2012
image_alias_delete() (in module salt.modules.lxd), 2012
image_allocate() (in module salt.cloud.clouds.opennebula), 1093
image_clone() (in module salt.cloud.clouds.opennebula), 1093
image_copy_lxd() (in module salt.modules.lxd), 2013
image_create() (in module salt.modules.glance), 1809
image_create() (in module salt.modules.glanceng), 1810
image_delete() (in module salt.cloud.clouds.opennebula), 1094
image_delete() (in module salt.modules.glance), 1809
image_delete() (in module salt.modules.glanceng), 1810

image_delete() (*in module salt.modules.lxd*), 2013
image_from_file() (*in module salt.modules.lxd*), 2013
image_from_simplestreams() (*in module salt.modules.lxd*), 2014
image_from_url() (*in module salt.modules.lxd*), 2014
image_get() (*in module salt.modules.glanceng*), 1811
image_get() (*in module salt.modules.lxd*), 2015
image_get_by_alias() (*in module salt.modules.lxd*), 2015
image_info() (*in module salt.cloud.clouds.opennebula*), 1094
image_list() (*in module salt.modules.glance*), 1809
image_list() (*in module salt.modules.glanceng*), 1811
image_list() (*in module salt.modules.lxd*), 2015
image_list() (*in module salt.modules.nova*), 2254
image_meta_delete() (*in module salt.modules.nova*), 2254
image_meta_set() (*in module salt.modules.nova*), 2254
image_persistent() (*in module salt.cloud.clouds.opennebula*), 1094
image_present() (*in module salt.states.glance*), 3465
image_present() (*in module salt.states.smartos*), 3716
image_schema() (*in module salt.modules.glance*), 1809
image_search() (*in module salt.modules.glanceng*), 1811
image_show() (*in module salt.modules.glance*), 1809
image_snapshot_delete() (*in module salt.cloud.clouds.opennebula*), 1094
image_snapshot_flatten() (*in module salt.cloud.clouds.opennebula*), 1095
image_snapshot_revert() (*in module salt.cloud.clouds.opennebula*), 1095
image_update() (*in module salt.cloud.clouds.opennebula*), 1095
image_update() (*in module salt.modules.glance*), 1809
image_vacuum() (*in module salt.states.smartos*), 3716
images() (*in module salt.modules.dockermod*), 1640
images() (*in module salt.modules.lxc*), 1993
import_() (*in module salt.modules.dockermod*), 1640
import_() (*in module salt.modules.zonecfg*), 2911
import_() (*in module salt.modules.zpool*), 2919
import_() (*in module salt.states.zone*), 3812
import_cert() (*in module salt.modules.win_pki*), 2786
import_cert() (*in module salt.states.win_pki*), 3774
import_file() (*in module salt.modules.reg*), 2435
import_image() (*in module salt.modules.smartos_imgadm*), 2500
import_json() (*in module salt.modules.jinja*), 1911
import_key() (*in module salt.cloud.clouds.joyent*), 1063
import_key() (*in module salt.modules.boto_ec2*), 1370
import_key() (*in module salt.modules.gpg*), 1820
import_key_pair() (*in module salt.modules.libcloud_compute*), 1963
import_keypair() (*in module salt.cloud.clouds.digitalocean*), 1041
import_keypair() (*in module salt.cloud.clouds.ec2*), 1050
import_status() (*in module salt.modules.solr*), 2530
import_yaml() (*in module salt.modules.jinja*), 1911
in_subnet() (*in module salt.modules.network*), 2217
in_subnet() (*in module salt.modules.win_network*), 2773
include
  conf/master, 117
  conf/minion, 164
increase_stream_retention_period() (*in module salt.modules.boto_kinesis*), 1414
increment() (*in module salt.modules.memcached*), 2077
index() (*in module salt.modules.textfsm_mod*), 2606
index_absent() (*in module salt.states.elasticsearch*), 3401
index_close() (*in module salt.modules.elasticsearch*), 1685
index_create() (*in module salt.modules.elasticsearch*), 1685
index_delete() (*in module salt.modules.elasticsearch*), 1685
index_exists() (*in module salt.modules.elasticsearch*), 1685
index_get() (*in module salt.modules.elasticsearch*), 1686
index_get_settings() (*in module salt.modules.elasticsearch*), 1686
index_open() (*in module salt.modules.elasticsearch*), 1686
index_present() (*in module salt.states.elasticsearch*), 3401
index_put_settings() (*in module salt.modules.elasticsearch*), 1687
index_template_absent() (*in module salt.states.elasticsearch*), 3401
index_template_create() (*in module salt.modules.elasticsearch*), 1687
index_template_delete() (*in module salt.modules.elasticsearch*), 1687
index_template_exists() (*in module salt.modules.elasticsearch*), 1688
index_template_get() (*in module salt.modules.elasticsearch*), 1688

`index_template_present()` (*in module salt.states.elasticsearch*), 3401
`indexes()` (*in module salt.modules.sqlite3*), 2538
`indices()` (*in module salt.modules.sqlite3*), 2538
`info`
    `spm command line option`, 991
`info()` (*in module salt.modules.acme*), 1202
`info()` (*in module salt.modules.aix_group*), 1203
`info()` (*in module salt.modules.bsd_shadow*), 1465
`info()` (*in module salt.modules.btrfs*), 1467
`info()` (*in module salt.modules.cassandra*), 1482
`info()` (*in module salt.modules.cassandra_cql*), 1487
`info()` (*in module salt.modules.cisconso*), 1519
`info()` (*in module salt.modules.dockermod*), 1641
`info()` (*in module salt.modules.dpkg_lowpkg*), 1661
`info()` (*in module salt.modules.elasticsearch*), 1688
`info()` (*in module salt.modules.glusterfs*), 1813
`info()` (*in module salt.modules.groupadd*), 1838
`info()` (*in module salt.modules.logmod*), 1982
`info()` (*in module salt.modules.lxc*), 1993
`info()` (*in module salt.modules.mac_group*), 2027
`info()` (*in module salt.modules.mac_shadow*), 2043
`info()` (*in module salt.modules.mac_user*), 2058
`info()` (*in module salt.modules.marathon*), 2070
`info()` (*in module salt.modules.nspawn*), 2260
`info()` (*in module salt.modules.pw_group*), 2406
`info()` (*in module salt.modules.pw_user*), 2408
`info()` (*in module salt.modules.redismod*), 2431
`info()` (*in module salt.modules.rpm_lowpkg*), 2448
`info()` (*in module salt.modules.smartos_vmadm*), 2505
`info()` (*in module salt.modules.solaris_group*), 2516
`info()` (*in module salt.modules.solaris_shadow*), 2516
`info()` (*in module salt.modules.solaris_user*), 2520
`info()` (*in module salt.modules.svn*), 2570
`info()` (*in module salt.modules.udev*), 2626
`info()` (*in module salt.modules.useradd*), 2633
`info()` (*in module salt.modules.win_groupadd*), 2753
`info()` (*in module salt.modules.win_license*), 2772
`info()` (*in module salt.modules.win_service*), 2798
`info()` (*in module salt.modules.win_shadow*), 2801
`info()` (*in module salt.modules.win_task*), 2829
`info()` (*in module salt.modules.win_useradd*), 2836
`info()` (*in module salt.modules.xfs*), 2860
`info()` (*in module salt.modules.zonecfg*), 2911
`info()` (*in module salt.runners.lxc*), 3119
`info_available()` (*in module salt.modules.zypperpkg*), 2925
`info_installed()` (*in module salt.modules.aptpkg*), 1218
`info_installed()` (*in module salt.modules.mac_brew_pkg*), 2023
`info_installed()` (*in module salt.modules.opkg*), 2281
`info_installed()` (*in module salt.modules.yumpkg*), 2866
`info_installed()` (*in module salt.modules.zypperpkg*), 2925
`inherit()` (*in module salt.modules.zfs*), 2902
`inherit()` (*in module salt.states.win_dacl*), 3758
`init()` (*in module salt.modules.git*), 1772
`init()` (*in module salt.modules.lxc*), 1993
`init()` (*in module salt.modules.lxd*), 2016
`init()` (*in module salt.modules.qemu_nbd*), 2411
`init()` (*in module salt.modules.smartos_virt*), 2503
`init()` (*in module salt.modules.solaris_system*), 2517
`init()` (*in module salt.modules.system*), 2589
`init()` (*in module salt.modules.vagrant*), 2635
`init()` (*in module salt.modules.virt*), 2653
`init()` (*in module salt.modules.win_system*), 2815
`init()` (*in module salt.proxy.arista_pyeapi*), 3050
`init()` (*in module salt.proxy.chronos*), 3052
`init()` (*in module salt.proxy.cimc*), 3052
`init()` (*in module salt.proxy.cisconso*), 3056
`init()` (*in module salt.proxy.dummy*), 3057
`init()` (*in module salt.proxy.esxi*), 3061
`init()` (*in module salt.proxy.fx2*), 3064
`init()` (*in module salt.proxy.junos*), 3065
`init()` (*in module salt.proxy.marathon*), 3066
`init()` (*in module salt.proxy.napalm*), 3069
`init()` (*in module salt.proxy.netmiko_px*), 3073
`init()` (*in module salt.proxy.nxos*), 3075
`init()` (*in module salt.proxy.nxos_api*), 3078
`init()` (*in module salt.proxy.panos*), 3081
`init()` (*in module salt.proxy.philips_hue*), 3084
`init()` (*in module salt.proxy.rest_sample*), 3085
`init()` (*in module salt.proxy.ssh_sample*), 3086
`init()` (*in module salt.runners.lxc*), 3119
`init()` (*in module salt.runners.virt*), 3158
`init_kwargs()` (*in module salt.cache.localfs*), 1028
`init_kwargs()` (*in module salt.cache.redis_cache*), 1031
`initialized()` (*in module salt.proxy.arista_pyeapi*), 3050
`initialized()` (*in module salt.proxy.cimc*), 3052
`initialized()` (*in module salt.proxy.dummy*), 3057
`initialized()` (*in module salt.proxy.junos*), 3065
`initialized()` (*in module salt.proxy.napalm*), 3069
`initialized()` (*in module salt.proxy.netmiko_px*), 3073
`initialized()` (*in module salt.proxy.nxos*), 3075
`initialized()` (*in module salt.proxy.nxos_api*), 3078
`initialized()` (*in module salt.proxy.panos*), 3081
`initialized()` (*in module salt.proxy.rest_sample*), 3085
`initialized()` (*in module salt.proxy.ssh_sample*), 3086
`initialized()` (*in module salt.states.vagrant*), 3745

`inodeusage()` (*in module salt.modules.disk*), 1604
`insert()` (*in module salt.modules.iptables*), 1901
`insert()` (*in module salt.modules.mongodb*), 2087
`insert()` (*in module salt.modules.nftables*), 2246
`insert()` (*in module salt.queues.pgjsonb_queue*), 3087
`insert()` (*in module salt.queues.sqlite_queue*), 3088
`insert()` (*in module salt.runners.queue*), 3140
`insert()` (*in module salt.states.iptables*), 3506
`insert()` (*in module salt.states.nftables*), 3611
`insert_runner()` (*in module salt.runners.queue*), 3140
`inspect()` (*in module salt.modules.dockermod*), 1641
`inspect()` (*in module salt.modules.inspector*), 1879
`inspect_container()` (*in module salt.modules.dockermod*), 1642
`inspect_image()` (*in module salt.modules.dockermod*), 1642
`inspect_network()` (*in module salt.modules.dockermod*), 1642
`inspect_volume()` (*in module salt.modules.dockermod*), 1642
`Inspector` (*class in salt.modules.inspectlib.collector*), 1874
`InspectorKiwiProcessorException`, 1875
`InspectorQueryException`, 1875
`InspectorSnapshotException`, 1875
`install`
    `spm command line option`, 991
`install()` (*in module salt.modules.aixpkg*), 1204
`install()` (*in module salt.modules.alternatives*), 1207
`install()` (*in module salt.modules.apkpkg*), 1213
`install()` (*in module salt.modules.aptpkg*), 1219
`install()` (*in module salt.modules.bower*), 1463
`install()` (*in module salt.modules.cabal*), 1471
`install()` (*in module salt.modules.chocolatey*), 1501
`install()` (*in module salt.modules.composer*), 1555
`install()` (*in module salt.modules.cpan*), 1579
`install()` (*in module salt.modules.cyg*), 1584
`install()` (*in module salt.modules.dummyproxy_pkg*), 1673
`install()` (*in module salt.modules.ebuildpkg*), 1676
`install()` (*in module salt.modules.freebsd_update*), 1739
`install()` (*in module salt.modules.freebsdpkg*), 1743
`install()` (*in module salt.modules.freebsdports*), 1745
`install()` (*in module salt.modules.gem*), 1749
`install()` (*in module salt.modules.mac_assistive*), 2022
`install()` (*in module salt.modules.mac_brew_pkg*), 2023
`install()` (*in module salt.modules.mac_keychain*), 2028
`install()` (*in module salt.modules.mac_pkgutil*), 2031
`install()` (*in module salt.modules.mac_portspkg*), 2032
`install()` (*in module salt.modules.macpackage*), 2030
`install()` (*in module salt.modules.nix*), 2251
`install()` (*in module salt.modules.npm*), 2258
`install()` (*in module salt.modules.openbsdpkg*), 2270
`install()` (*in module salt.modules.opkg*), 2282
`install()` (*in module salt.modules.pacmanpkg*), 2298
`install()` (*in module salt.modules.pecl*), 2333
`install()` (*in module salt.modules.pip*), 2345
`install()` (*in module salt.modules.pkgin*), 2351
`install()` (*in module salt.modules.pkgng*), 2357
`install()` (*in module salt.modules.pkgutil*), 2368
`install()` (*in module salt.modules.pyenv*), 2409
`install()` (*in module salt.modules.rbenv*), 2424
`install()` (*in module salt.modules.rest_pkg*), 2438
`install()` (*in module salt.modules.rvm*), 2457
`install()` (*in module salt.modules.solarisipspkg*), 2521
`install()` (*in module salt.modules.solarispkg*), 2524
`install()` (*in module salt.modules.ssh_pkg*), 2542
`install()` (*in module salt.modules.win_license*), 2772
`install()` (*in module salt.modules.win_pkg*), 2778
`install()` (*in module salt.modules.win_psget*), 2790
`install()` (*in module salt.modules.win_servermanager*), 2792
`install()` (*in module salt.modules.win_wua*), 2843
`install()` (*in module salt.modules.win_wusa*), 2846
`install()` (*in module salt.modules.xbpspkg*), 2858
`install()` (*in module salt.modules.yumpkg*), 2867
`install()` (*in module salt.modules.zoneadm*), 2908
`install()` (*in module salt.modules.zypperpkg*), 2926
`install()` (*in module salt.states.alternatives*), 3184
`install()` (*salt.modules.ansiblegate.AnsibleModuleResolver method*), 1208
`install_agent()` (*in module salt.modules.serverdensity_device*), 2495
`install_antivirus()` (*in module salt.modules.panos*), 2314
`install_app()` (*in module salt.modules.macpackage*), 2030
`install_config()` (*in module salt.modules.junos*), 1915
`install_config()` (*in module salt.states.junos*), 3511
`install_cygwin()` (*in module salt.modules.chocolatey*), 1502
`install_gem()` (*in module salt.modules.chocolatey*), 1502
`install_license()` (*in module salt.modules.panos*), 2315
`install_missing()` (*in module salt.modules.chocolatey*), 1502
`install_os()` (*in module salt.modules.junos*), 1916
`install_os()` (*in module salt.states.junos*), 3512
`install_pyenv()` (*in module salt.states.pyenv*), 3680

install_python() (*in module salt.modules.chocolatey*), 1503
install_python() (*in module salt.modules.pyenv*), 2409
install_rbenv() (*in module salt.states.rbenv*), 3687
install_ruby() (*in module salt.modules.rbenv*), 2425
install_ruby() (*in module salt.modules.rvm*), 2457
install_semod() (*in module salt.modules.selinux*), 2489
install_software() (*in module salt.modules.panos*), 2315
install_webpi() (*in module salt.modules.chocolatey*), 1503
install_windowsfeatures() (*in module salt.modules.chocolatey*), 1503
installed() (*in module salt.modules.dummyproxy_pkg*), 1673
installed() (*in module salt.modules.mac_assistive*), 2022
installed() (*in module salt.modules.rest_pkg*), 2438
installed() (*in module salt.modules.win_license*), 2772
installed() (*in module salt.states.bower*), 3335
installed() (*in module salt.states.cabal*), 3336
installed() (*in module salt.states.chocolatey*), 3338
installed() (*in module salt.states.composer*), 3354
installed() (*in module salt.states.cyg*), 3361
installed() (*in module salt.states.gem*), 3453
installed() (*in module salt.states.mac_assistive*), 3548
installed() (*in module salt.states.mac_keychain*), 3549
installed() (*in module salt.states.macpackage*), 3550
installed() (*in module salt.states.npm*), 3612
installed() (*in module salt.states.pecl*), 3637
installed() (*in module salt.states.pip_state*), 3638
installed() (*in module salt.states.pkg*), 3644
installed() (*in module salt.states.ports*), 3665
installed() (*in module salt.states.pyenv*), 3680
installed() (*in module salt.states.rbenv*), 3687
installed() (*in module salt.states.rvm*), 3697
installed() (*in module salt.states.win_servermanager*), 3776
installed() (*in module salt.states.win_wua*), 3784
installed() (*in module salt.states.win_wusa*), 3787
installed() (*in module salt.states.zcbuildout*), 3804
installed() (*in module salt.states.zone*), 3812
installed_capabilities() (*in module salt.modules.win_dism*), 2726
installed_extensions() (*in module salt.modules.postgres*), 2378
installed_features() (*in module salt.modules.win_dism*), 2726
installed_packages() (*in module salt.modules.win_dism*), 2726
installed_pkgs() (*in module salt.modules.macpackage*), 2030
instance_absent() (*in module salt.states.boto_ec2*), 3275
instance_present() (*in module salt.states.boto_ec2*), 3275
instance_profile_exists() (*in module salt.modules.boto_iam*), 1402
interface
  conf/master, 51
interface() (*in module salt.modules.network*), 2217
interface_addresses() (*in module salt.modules.osquery*), 2290
interface_details() (*in module salt.modules.osquery*), 2290
interface_get_options() (*in module salt.modules.openvswitch*), 2278
interface_get_type() (*in module salt.modules.openvswitch*), 2279
interface_ip() (*in module salt.modules.network*), 2218
interfaces() (*in module salt.grains.napalm*), 1182
interfaces() (*in module salt.modules.bridge*), 1464
interfaces() (*in module salt.modules.napalm_network*), 2162
interfaces() (*in module salt.modules.network*), 2218
interfaces() (*in module salt.modules.sysfs*), 2576
interfaces() (*in module salt.modules.win_network*), 2773
interfaces() (*in module salt.runners.net*), 3133
interfaces_names() (*in module salt.modules.win_network*), 2773
internet_gateway_absent() (*in module salt.states.boto_vpc*), 3330
internet_gateway_present() (*in module salt.states.boto_vpc*), 3330
intersect
  jinja filters, 433
InvalidConfigError, 3999
InvalidEntityError, 3999
inventory() (*in module salt.modules.dracr*), 1668
inventory() (*in module salt.modules.xfs*), 2860
iokit_devicetree() (*in module salt.modules.osquery*), 2291
iokit_registry() (*in module salt.modules.osquery*), 2291
IoOtherOperationCount (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATIC attribute*), 2808
IoOtherTransferCount (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATIC attribute*), 2808
IoReadOperationCount

(*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATION attribute*), 2808
IoReadTransferCount (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATION attribute*), 2809
iostat() (*in module salt.modules.disk*), 1604
iostat() (*in module salt.modules.zpool*), 2920
IoWriteOperationCount (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATION attribute*), 2809
IoWriteTransferCount (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATION attribute*), 2809
ip4_interfaces() (*in module salt.grains.core*), 1178
ip6_interfaces() (*in module salt.grains.core*), 1178
ip_addrs() (*in module salt.modules.network*), 2218
ip_addrs() (*in module salt.modules.win_network*), 2774
ip_addrs6() (*in module salt.modules.network*), 2218
ip_addrs6() (*in module salt.modules.win_network*), 2774
ip_fqdn() (*in module salt.grains.core*), 1178
ip_in_subnet() (*in module salt.modules.network*), 2218
ip_interfaces() (*in module salt.grains.core*), 1178
ipaddr
    jinja filters, 444
ipaddrs() (*in module salt.modules.napalm_network*), 2163
ipaddrs() (*in module salt.modules.network*), 2218
ipaddrs() (*in module salt.modules.win_network*), 2774
ipaddrs6() (*in module salt.modules.network*), 2218
ipaddrs6() (*in module salt.modules.win_network*), 2774
ipc_mode
    conf/master, 63
    conf/minion, 140
ipcidr() (*in module salt.modules.match*), 2072
iphexval() (*in module salt.modules.network*), 2219
ipv4
    jinja filters, 444
ipv6
    conf/master, 51
    conf/minion, 124
    jinja filters, 445
IRCClient (*class in salt.engines.ircbot*), 1156
is_alive() (*in module salt.modules.inspectlib.collector*), 1875
is_auth() (*in module salt.modules.pcs*), 2329
is_available_extension() (*in module salt.modules.postgres*), 2378
is_binary_file
    jinja filters, 447
is_blkdev() (*in module salt.modules.file*), 1717
is_cached() (*in module salt.modules.cp*), 1577
is_changed_uses() (*in module salt.modules.portage_config*), 2372
is_chrdev() (*in module salt.modules.file*), 1717
is_closed() (*salt.modules.inspectlib.fsdb.CsvDB method*), 1877
is_connected_via_vcenter() (*in module salt.proxy.esxi*), 3061
is_connection_usable() (*in module salt.auth.django*), 999
is_disabled() (*in module salt.modules.win_ip*), 2765
is_empty
    jinja filters, 447
is_enabled() (*in module salt.modules.freebsdjail*), 1739
is_enabled() (*in module salt.modules.schedule*), 2482
is_enabled() (*in module salt.modules.win_ip*), 2765
is_encrypted() (*in module salt.modules.archive*), 1228
is_fifo() (*in module salt.modules.file*), 1717
is_fuse_exec() (*in module salt.modules.mount*), 2092
is_hardlink() (*in module salt.modules.file*), 1717
is_hex
    jinja filters, 434
is_hyper() (*in module salt.modules.virt*), 2656
is_hyper() (*in module salt.modules.xapi_virt*), 2854
is_installed() (*in module salt.modules.mac_pkgutil*), 2031
is_installed() (*in module salt.modules.pip*), 2347
is_installed() (*in module salt.modules.pyenv*), 2410
is_installed() (*in module salt.modules.rbenv*), 2425
is_installed() (*in module salt.modules.rvm*), 2457
is_installed() (*in module salt.modules.solarisipspkg*), 2521
is_installed() (*in module salt.modules.win_wusa*), 2846
is_installed_extension() (*in module salt.modules.postgres*), 2378
is_ip
    jinja filters, 443
is_ipaddr_in_ipfunc_range() (*in module salt.modules.infoblox*), 1871
is_ipv4
    jinja filters, 444
is_ipv6
    jinja filters, 444
is_iter
    jinja filters, 432
is_jail() (*in module salt.modules.poudriere*), 2386
is_kvm_hyper() (*in module salt.modules.virt*), 2656
is_link() (*in module salt.modules.file*), 1717
is_link() (*in module salt.modules.win_file*), 2738

is_list
    jinja filters, 431
is_loaded() (*in module salt.modules.freebsdkmod*), 1741
is_loaded() (*in module salt.modules.kmod*), 1945
is_loopback() (*in module salt.modules.network*), 2219
is_mounted() (*in module salt.modules.mount*), 2092
is_peering_connection_pending() (*in module salt.modules.boto_vpc*), 1459
is_present() (*in module salt.modules.portage_config*), 2372
is_present() (*in module salt.states.zabbix_template*), 3798
is_private() (*in module salt.modules.network*), 2219
is_private() (*in module salt.modules.win_network*), 2775
is_replication_enabled() (*in module salt.modules.solr*), 2530
is_required_version() (*in module salt.proxy.panos*), 3081
is_running() (*in module salt.modules.salt_proxy*), 2462
is_running() (*in module salt.modules.saltutil*), 2471
is_running() (*in module salt.runners.smartos_vmadm*), 3149
is_sorted
    jinja filters, 434
is_team_member() (*in module salt.modules.github*), 1805
is_text_file
    jinja filters, 447
is_worktree() (*in module salt.modules.git*), 1773
is_xen_hyper() (*in module salt.modules.virt*), 2656
iscsi_grains
    conf/minion, 135
issue_closed() (*in module salt.modules.jira_mod*), 1913
item() (*in module salt.modules.environ*), 1693
item() (*in module salt.modules.grains*), 1835
item() (*in module salt.modules.pillar*), 2342
item_create() (*in module salt.modules.pcs*), 2330
item_show() (*in module salt.modules.pcs*), 2330
items() (*in module salt.modules.config*), 1559
items() (*in module salt.modules.data*), 1587
items() (*in module salt.modules.environ*), 1693
items() (*in module salt.modules.grains*), 1835
items() (*in module salt.modules.pillar*), 2342

J

Jinja, **5401**
jinja filters
    append_dict_key_value, 438
    avg, 432
    base64_decode, 441
    base64_encode, 440
    calling-salt-functions, 451
    check_whitelist_blacklist, 435
    compare_dicts, 434
    compare_lists, 434
    contains_whitespace, 435
    custom-execution-modules, 452
    custom-jinja-filters, 452
    date_format, 436
    debugging, 451
    difference, 433
    dns_check, 446
    equalto, 449
    escaping-jinja, 451
    exactly_n_true, 430
    exactly_one_true, 430
    extend_dict_key_value, 439
    file_hashsum, 448
    gen_mac, 446
    hmac, 441
    http_query, 441
    intersect, 433
    ipaddr, 444
    ipv4, 444
    ipv6, 445
    is_binary_file, 447
    is_empty_file, 447
    is_hex, 434
    is_ip, 443
    is_ipv4, 444
    is_ipv6, 444
    is_iter, 432
    is_list, 431
    is_sorted, 434
    is_text_file, 447
    json_decode_dict, 437
    json_decode_list, 437
    json_encode_dict, 437
    json_encode_list, 437
    json_query, 442
    list_files, 448
    logs, 452
    mac_str_to_bytes, 446
    match, 449
    max, 432
    md5, 440
    min, 432
    network_hosts, 445
    network_size, 445
    path_join, 448
    quote, 430
    random_hash, 438
    regex_escape, 450

- regex_match, 431
- regex_replace, 431
- regex_search, 430
- sequence, 428
- set_dict_key_value, 438
- sha256, 440
- sha512, 440
- strftime, 428
- substring_in_list, 435
- symmetric_difference, 433
- to_bool, 429
- to_bytes, 436
- to_camelcase, 443
- to_num, 436
- to_snake_case, 443
- tojson, 438
- traverse, 442
- union, 433
- unique, 450
- update_dict_key_value, 439
- uuid, 431
- which, 448
- yaml_dquote, 429
- yaml_encode, 428
- yaml_squote, 429

jinja_env
- conf/master, 77

jinja_lstrip_blocks
- conf/master, 79

jinja_sls_env
- conf/master, 78

jinja_trim_blocks
- conf/master, 79

Job, **5402**

Job Cache, **5402**

Job ID, **5402**

Job Management, 234

job() (*in module salt.modules.chronos*), 1506

job_cache
- conf/master, 57

job_cache_store_endtime
- conf/master, 60

job_exists() (*in module salt.modules.jenkinsmod*), 1910

job_status() (*in module salt.modules.jenkinsmod*), 1910

jobcheck() (*in module salt.modules.at*), 1243

jobcheck() (*in module salt.modules.at_solaris*), 1244

Jobs (*class in salt.netapi.rest_cherrypy.app*), 2948

jobs() (*in module salt.modules.chronos*), 1506

JobsSaltAPIHandler (*class in salt.netapi.rest_tornado.saltnado*), 2972

join() (*in module salt.modules.file*), 1717

join() (*in module salt.states.rabbitmq_cluster*), 3681

join_channel() (*salt.engines.ircbot.IRCClient method*), 1156

join_cluster() (*in module salt.modules.rabbitmq*), 2414

join_domain() (*in module salt.modules.win_system*), 2815

join_domain() (*in module salt.states.win_system*), 3782

joined() (*in module salt.runners.manage*), 3123

joined() (*in module salt.states.rabbitmq_cluster*), 3682

joinswarm() (*in module salt.modules.swarm*), 2571

joyent_node_state() (*in module salt.cloud.clouds.joyent*), 1063

json_decode_dict
- jinja filters, 437

json_decode_list
- jinja filters, 437

json_encode_dict
- jinja filters, 437

json_encode_list
- jinja filters, 437

json_query
- jinja filters, 442

junos_call() (*in module salt.modules.napalm_mod*), 2130

junos_cli() (*in module salt.modules.napalm_mod*), 2130

junos_commit() (*in module salt.modules.napalm_mod*), 2131

junos_copy_file() (*in module salt.modules.napalm_mod*), 2131

junos_facts() (*in module salt.modules.napalm_mod*), 2131

junos_install_os() (*in module salt.modules.napalm_mod*), 2131

junos_rpc() (*in module salt.modules.napalm_mod*), 2132

K

keep_acl_in_token
- conf/master, 71

keep_jobs
- conf/master, 55

keepvol_on_destroy() (*in module salt.cloud.clouds.ec2*), 1050

kernel() (*in module salt.grains.chronos*), 1177

kernel() (*in module salt.grains.esxi*), 1179

kernel() (*in module salt.grains.fx2*), 1180

kernel() (*in module salt.grains.marathon*), 1180

kernel() (*in module salt.grains.philips_hue*), 1185

kernel() (*in module salt.grains.rest_sample*), 1185

kernel_extensions() (*in module salt.modules.osquery*), 2291

kernel_info() (*in module salt.modules.osquery*), 2291
kernel_integrity() (*in module salt.modules.osquery*), 2291
kernel_modules() (*in module salt.modules.osquery*), 2291
key_absent() (*in module salt.states.boto_ec2*), 3277
key_absent() (*in module salt.states.reg*), 3690
key_exists() (*in module salt.modules.boto_kms*), 1417
key_exists() (*in module salt.modules.reg*), 2435
key_is_encrypted() (*in module salt.modules.ssh*), 2540
key_json() (*in module salt.pillar.redismod*), 3030
key_list() (*in module salt.cloud.clouds.joyent*), 1063
key_present() (*in module salt.states.boto_ec2*), 3277
key_present() (*in module salt.states.boto_kms*), 3305
key_regen() (*in module salt.runners.manage*), 3123
key_str() (*in module salt.wheel.key*), 3841
key_type() (*in module salt.modules.redismod*), 2431
key_value() (*in module salt.pillar.redismod*), 3030
key_value_to_tree() (*in module salt.pillar.pepa*), 3026
keychain_items() (*in module salt.modules.osquery*), 2291
keygen() (*in module salt.modules.nacl*), 2110
keygen() (*in module salt.runners.nacl*), 3129
keyname() (*in module salt.cloud.clouds.ec2*), 1050
keypair_add() (*in module salt.modules.nova*), 2254
keypair_delete() (*in module salt.modules.nova*), 2255
keypair_list() (*in module salt.modules.nova*), 2255
keyring_auth_add() (*in module salt.modules.ceph*), 1491
keyring_auth_del() (*in module salt.modules.ceph*), 1491
keyring_auth_list() (*in module salt.modules.ceph*), 1492
keyring_create() (*in module salt.modules.ceph*), 1492
keyring_present() (*in module salt.modules.ceph*), 1492
keyring_purge() (*in module salt.modules.ceph*), 1493
keyring_save() (*in module salt.modules.ceph*), 1493
Keys (*class in salt.netapi.rest_cherrypy.app*), 2956
keys() (*in module salt.modules.data*), 1587
keys() (*in module salt.modules.pillar*), 2343
keys() (*in module salt.modules.redismod*), 2431
keys() (*in module salt.states.virt*), 3748
keys_absent() (*in module salt.states.boto_iam*), 3295
keys_present() (*in module salt.states.boto_iam*), 3296
keysize
  conf/master, 68
  conf/minion, 157
keyspace_exists() (*in module salt.modules.cassandra_cql*), 1487
keyspaces() (*in module salt.modules.cassandra*), 1482
kill() (*in module salt.modules.dockercompose*), 1612
kill() (*in module salt.modules.dockermod*), 1643
kill() (*in module salt.modules.minion*), 2080
kill_all_jobs() (*in module salt.modules.saltutil*), 2471
kill_job() (*in module salt.modules.saltutil*), 2471
kill_pid() (*in module salt.modules.ps*), 2391
KiwiExporter (*class in salt.modules.inspectlib.kiwiproc*), 1878
kwarg() (*in module salt.modules.test*), 2602

L

label_absent() (*in module salt.modules.k8s*), 1921
label_folder_absent() (*in module salt.modules.k8s*), 1921
label_present() (*in module salt.modules.k8s*), 1921
labelclear() (*in module salt.modules.zpool*), 2920
LAMP_OFF (*salt.proxy.philips_hue.Const attribute*), 3082
LAMP_ON (*salt.proxy.philips_hue.Const attribute*), 3082
language_create() (*in module salt.modules.postgres*), 2378
language_exists() (*in module salt.modules.postgres*), 2379
language_list() (*in module salt.modules.postgres*), 2379
language_remove() (*in module salt.modules.postgres*), 2379
last() (*in module salt.modules.osquery*), 2291
last_run() (*in module salt.runners.jobs*), 3117
lastsave() (*in module salt.modules.redismod*), 2431
latest() (*in module salt.states.boto3_elasticsearch*), 3243
latest() (*in module salt.states.git*), 3458
latest() (*in module salt.states.hg*), 3485
latest() (*in module salt.states.pkg*), 3653
latest() (*in module salt.states.svn*), 3730
latest_active() (*in module salt.states.kernelpkg*), 3516
latest_available() (*in module salt.modules.kernelpkg_linux_apt*), 1925
latest_available() (*in module salt.modules.kernelpkg_linux_yum*), 1926
latest_installed() (*in module salt.modules.kernelpkg_linux_apt*), 1925
latest_installed() (*in module salt.modules.kernelpkg_linux_yum*), 1927
latest_installed() (*in module salt.states.kernelpkg*), 3516

`latest_version()` (*in module salt.modules.aixpkg*), 1205
`latest_version()` (*in module salt.modules.apkpkg*), 1214
`latest_version()` (*in module salt.modules.aptpkg*), 1220
`latest_version()` (*in module salt.modules.ebuildpkg*), 1677
`latest_version()` (*in module salt.modules.freebsdpkg*), 1743
`latest_version()` (*in module salt.modules.mac_brew_pkg*), 2024
`latest_version()` (*in module salt.modules.mac_portspkg*), 2033
`latest_version()` (*in module salt.modules.openbsdpkg*), 2271
`latest_version()` (*in module salt.modules.opkg*), 2283
`latest_version()` (*in module salt.modules.pacmanpkg*), 2299
`latest_version()` (*in module salt.modules.pkgin*), 2352
`latest_version()` (*in module salt.modules.pkgng*), 2359
`latest_version()` (*in module salt.modules.pkgutil*), 2368
`latest_version()` (*in module salt.modules.solarisipspkg*), 2522
`latest_version()` (*in module salt.modules.solarispkg*), 2526
`latest_version()` (*in module salt.modules.win_pkg*), 2780
`latest_version()` (*in module salt.modules.xbpspkg*), 2858
`latest_version()` (*in module salt.modules.yumpkg*), 2869
`latest_version()` (*in module salt.modules.zypperpkg*), 2927
`latest_wait()` (*in module salt.states.kernelpkg*), 3516
`launch_configuration_exists()` (*in module salt.modules.boto_asg*), 1344
`launchctl()` (*in module salt.modules.mac_service*), 2039
`launchd()` (*in module salt.modules.osquery*), 2292
`layers()` (*in module salt.modules.dockermod*), 1643
`lb_edit()` (*in module salt.modules.modjk*), 2084
`lbmethods()` (*salt.runners.f5.F5Mgmt method*), 3110
`lchown()` (*in module salt.modules.file*), 1717
`lchown()` (*in module salt.modules.win_file*), 2738
`LDAPError`, 1953
`ldd_deps()` (*in module salt.modules.genesis*), 1753
`leaks()` (*in module salt.modules.tomcat*), 2620
`lease_storage_container()` (*in module salt.cloud.clouds.msazure*), 1078
`leave_swarm()` (*in module salt.modules.swarm*), 2571
`len_eq()` (*in module salt.thorium.check*), 3822
`len_gt()` (*in module salt.thorium.check*), 3822
`len_gte()` (*in module salt.thorium.check*), 3822
`len_lt()` (*in module salt.thorium.check*), 3823
`len_lte()` (*in module salt.thorium.check*), 3823
`len_ne()` (*in module salt.thorium.check*), 3823
`less_than()` (*in module salt.modules.salt_version*), 2463
`license_absent()` (*in module salt.states.powerpath*), 3676
`license_present()` (*in module salt.states.powerpath*), 3676
`licensed()` (*in module salt.modules.win_license*), 2772
`line` (*salt.engines.ircbot.Event attribute*), 1156
`line` (*salt.engines.ircbot.PrivEvent attribute*), 1156
`line()` (*in module salt.modules.file*), 1718
`line()` (*in module salt.states.file*), 3430
`link()` (*in module salt.modules.file*), 1719
`links()` (*in module salt.modules.udev*), 2626
`linux_distribution()` (*in module salt.grains.core*), 1178
`list()` (*in module salt.modules.ansiblegate*), 1208
`list()` (*in module salt.modules.boto_cloudtrail*), 1349
`list()` (*in module salt.modules.boto_s3_bucket*), 1437
`list()` (*in module salt.modules.keystore*), 1944
`list()` (*in module salt.modules.win_timezone*), 2832
`list()` (*in module salt.modules.win_wua*), 2843
`list()` (*in module salt.modules.win_wusa*), 2846
`list()` (*in module salt.runners.virt*), 3159
`list()` (*salt.modules.inspectlib.fsdb.CsvDB method*), 1877
`list_()` (*in module salt.cache.consul*), 1026
`list_()` (*in module salt.cache.localfs*), 1028
`list_()` (*in module salt.cache.redis_cache*), 1031
`list_()` (*in module salt.modules.archive*), 1229
`list_()` (*in module salt.modules.beacons*), 1284
`list_()` (*in module salt.modules.boto_sqs*), 1446
`list_()` (*in module salt.modules.bower*), 1463
`list_()` (*in module salt.modules.bridge*), 1464
`list_()` (*in module salt.modules.cabal*), 1472
`list_()` (*in module salt.modules.chocolatey*), 1503
`list_()` (*in module salt.modules.consul*), 1570
`list_()` (*in module salt.modules.cpan*), 1579
`list_()` (*in module salt.modules.cyg*), 1584
`list_()` (*in module salt.modules.dummyproxy_service*), 1674
`list_()` (*in module salt.modules.gem*), 1750
`list_()` (*in module salt.modules.lvs*), 1985
`list_()` (*in module salt.modules.lxc*), 1995
`list_()` (*in module salt.modules.mac_pkgutil*), 2032
`list_()` (*in module salt.modules.mac_service*), 2040
`list_()` (*in module salt.modules.mac_xattr*), 2059
`list_()` (*in module salt.modules.match*), 2072

`list_()` (*in module salt.modules.mdadm_raid*), 2075
`list_()` (*in module salt.modules.mdata*), 2076
`list_()` (*in module salt.modules.minion*), 2081
`list_()` (*in module salt.modules.nova*), 2255
`list_()` (*in module salt.modules.npm*), 2259
`list_()` (*in module salt.modules.parted_partition*), 2326
`list_()` (*in module salt.modules.pecl*), 2333
`list_()` (*in module salt.modules.pip*), 2347
`list_()` (*in module salt.modules.pyenv*), 2410
`list_()` (*in module salt.modules.rbenv*), 2425
`list_()` (*in module salt.modules.rest_service*), 2439
`list_()` (*in module salt.modules.rvm*), 2457
`list_()` (*in module salt.modules.schedule*), 2482
`list_()` (*in module salt.modules.splunk_search*), 2537
`list_()` (*in module salt.modules.ssh_service*), 2542
`list_()` (*in module salt.modules.tuned*), 2625
`list_()` (*in module salt.modules.win_autoruns*), 2720
`list_()` (*in module salt.modules.zfs*), 2902
`list_()` (*in module salt.modules.zpool*), 2920
`list_()` (*in module salt.runners.cache*), 3102
`list_()` (*in module salt.runners.lxc*), 3120
`list_()` (*in module salt.runners.reactor*), 3142
`list_()` (*in module salt.thorium.reg*), 3826
`list_()` (*in module salt.wheel.key*), 3841
`list_absent()` (*in module salt.states.grains*), 3480
`list_absent()` (*in module salt.states.linux_acl*), 3539
`list_actions()` (*in module salt.modules.win_task*), 2829
`list_active_vms()` (*in module salt.modules.smartos_virt*), 2503
`list_active_vms()` (*in module salt.modules.vagrant*), 2636
`list_active_vms()` (*in module salt.modules.virt*), 2656
`list_affinity_groups()` (*in module salt.cloud.clouds.msazure*), 1078
`list_agents()` (*in module salt.modules.neutron*), 2229
`list_aliases()` (*in module salt.modules.aliases*), 1206
`list_all()` (*in module salt.modules.firewalld*), 1734
`list_all()` (*in module salt.modules.freebsdports*), 1745
`list_all()` (*in module salt.modules.layman*), 1953
`list_all()` (*in module salt.modules.nspawn*), 2261
`list_all()` (*in module salt.modules.splunk_search*), 2537
`list_all()` (*in module salt.wheel.key*), 3841
`list_all_versions()` (*in module salt.modules.pip*), 2348
`list_all_zones_by_id()` (*in module salt.modules.boto_route53*), 1433
`list_all_zones_by_name()` (*in module salt.modules.boto_route53*), 1434
`list_apppools()` (*in module salt.modules.win_iis*), 2758
`list_apps()` (*in module salt.modules.win_iis*), 2758
`list_assigned_licenses()` (*in module salt.modules.vsphere*), 2699
`list_attached_group_policies()` (*in module salt.modules.boto_iam*), 1402
`list_attached_role_policies()` (*in module salt.modules.boto_iam*), 1403
`list_attached_tags()` (*in module salt.modules.vsphere*), 2699
`list_attached_user_policies()` (*in module salt.modules.boto_iam*), 1403
`list_avail()` (*in module salt.modules.localemod*), 1980
`list_availability_zones()` (*in module salt.cloud.clouds.aliyun*), 1033
`list_availability_zones()` (*in module salt.cloud.clouds.tencentcloud*), 1131
`list_available()` (*in module salt.modules.beacons*), 1284
`list_available()` (*in module salt.modules.mac_softwareupdate*), 2046
`list_available()` (*in module salt.modules.win_pkg*), 2781
`list_available()` (*in module salt.modules.win_servermanager*), 2793
`list_available_plugins()` (*in module salt.modules.rabbitmq*), 2414
`list_backends()` (*in module salt.modules.haproxyconn*), 1841
`list_backups()` (*in module salt.modules.file*), 1719
`list_backups()` (*in module salt.modules.win_iis*), 2758
`list_backups_dir()` (*in module salt.modules.file*), 1719
`list_balancer_members()` (*in module salt.modules.libcloud_loadbalancer*), 1970
`list_balancers()` (*in module salt.modules.libcloud_loadbalancer*), 1971
`list_bindings()` (*in module salt.modules.win_iis*), 2758
`list_blobs()` (*in module salt.cloud.clouds.azurearm*), 1037
`list_blobs()` (*in module salt.cloud.clouds.msazure*), 1079
`list_branches()` (*in module salt.modules.git*), 1773
`list_cache_subnet_groups()` (*in module salt.modules.boto3_elasticache*), 1306
`list_cache_subnet_groups()` (*in module salt.modules.boto_elasticache*), 1382
`list_capability_definitions()` (*in module salt.modules.vsphere*), 2699
`list_cert_bindings()` (*in module*

salt.modules.win_iis), 2759
`list_certs()` (*in module salt.modules.mac_keychain*), 2029
`list_cidr_ips()` (*in module salt.modules.netaddress*), 2196
`list_cidr_ips_ipv6()` (*in module salt.modules.netaddress*), 2196
`list_cluster()` (*in module salt.modules.vsphere*), 2700
`list_clusters()` (*in module salt.cloud.clouds.opennebula*), 1096
`list_clusters()` (*in module salt.cloud.clouds.vmware*), 1141
`list_clusters()` (*in module salt.modules.vsphere*), 2700
`list_clusters_by_datacenter()` (*in module salt.cloud.clouds.vmware*), 1141
`list_column_families()` (*in module salt.modules.cassandra_cql*), 1488
`list_common_lookups()` (*in module salt.cloud.clouds.gogrid*), 1060
`list_conf()` (*in module salt.modules.logadm*), 1981
`list_configs()` (*in module salt.modules.snapper*), 2545
`list_configured_members()` (*in module salt.modules.modjk*), 2085
`list_container_objects()` (*in module salt.modules.libcloud_storage*), 1974
`list_containers()` (*in module salt.modules.dockermod*), 1643
`list_containers()` (*in module salt.modules.libcloud_storage*), 1974
`list_custom_images()` (*in module salt.cloud.clouds.softlayer*), 1128
`list_custom_images()` (*in module salt.cloud.clouds.tencentcloud*), 1131
`list_datacenters()` (*in module salt.cloud.clouds.profitbricks*), 1118
`list_datacenters()` (*in module salt.cloud.clouds.vmware*), 1141
`list_datacenters()` (*in module salt.modules.vsphere*), 2700
`list_datacenters_via_proxy()` (*in module salt.modules.vsphere*), 2700
`list_datastore_clusters()` (*in module salt.cloud.clouds.vmware*), 1141
`list_datastore_clusters()` (*in module salt.modules.vsphere*), 2701
`list_datastores()` (*in module salt.cloud.clouds.opennebula*), 1096
`list_datastores()` (*in module salt.cloud.clouds.vmware*), 1141
`list_datastores()` (*in module salt.modules.vsphere*), 2701
`list_datastores_via_proxy()` (*in module salt.modules.vsphere*), 2701
`list_dbs()` (*in module salt.modules.influxdbmod*), 1860
`list_default_storage_policy_of_datastore()` (*in module salt.modules.vsphere*), 2701
`list_default_vsan_policy()` (*in module salt.modules.vsphere*), 2702
`list_deployments()` (*in module salt.modules.jboss7*), 1904
`list_disabled()` (*in module salt.modules.state*), 2552
`list_disk_partitions()` (*in module salt.modules.vsphere*), 2702
`list_diskgroups()` (*in module salt.modules.vsphere*), 2702
`list_disks()` (*in module salt.cloud.clouds.msazure*), 1079
`list_disks()` (*in module salt.modules.vsphere*), 2702
`list_domain_cache()` (*in module salt.runners.digicertapi*), 3105
`list_domain_cache()` (*in module salt.runners.venafiapi*), 3158
`list_domain_names()` (*in module salt.modules.boto3_elasticsearch*), 1317
`list_domains()` (*in module salt.modules.smartos_virt*), 2503
`list_domains()` (*in module salt.modules.vagrant*), 2636
`list_domains()` (*in module salt.modules.virt*), 2657
`list_domains()` (*in module salt.modules.xapi_virt*), 2854
`list_domains()` (*in module salt.runners.digicertapi*), 3106
`list_downloaded()` (*in module salt.modules.aptpkg*), 1220
`list_downloaded()` (*in module salt.modules.yumpkg*), 2869
`list_downloaded()` (*in module salt.modules.zypperpkg*), 2927
`list_downloads()` (*in module salt.modules.mac_softwareupdate*), 2046
`list_dvportgroups()` (*in module salt.modules.vsphere*), 2702
`list_dvs()` (*in module salt.cloud.clouds.vmware*), 1141
`list_dvs()` (*in module salt.modules.vsphere*), 2703
`list_dvss()` (*in module salt.modules.vsphere*), 2703
`list_elasticsearch_instance_types()` (*in module salt.modules.boto3_elasticsearch*), 1317
`list_elasticsearch_versions()` (*in module salt.modules.boto3_elasticsearch*), 1317
`list_elbs()` (*in module salt.modules.boto_elb*), 1389
`list_employees()` (*in module salt.modules.bamboohr*), 1279

`list_enabled_plugins()` (*in module salt.modules.rabbitmq*), 2414
`list_entities_for_policy()` (*in module salt.modules.boto_iam*), 1403
`list_env()` (*in module salt.wheel.file_roots*), 3838
`list_env()` (*in module salt.wheel.pillar_roots*), 3842
`list_escalation_policies()` (*in module salt.modules.pagerduty*), 2302
`list_escalation_policies()` (*in module salt.runners.pagerduty*), 3137
`list_exports()` (*in module salt.modules.nfs3*), 2243
`list_extensions()` (*in module salt.modules.neutron*), 2230
`list_extmods()` (*in module salt.modules.saltutil*), 2471
`list_files`
  `jinja filters`, 448
`list_firewall_rules()` (*in module salt.modules.neutron*), 2230
`list_firewalls()` (*in module salt.modules.neutron*), 2230
`list_floating_ips()` (*in module salt.cloud.clouds.digitalocean*), 1041
`list_floatingips()` (*in module salt.modules.neutron*), 2230
`list_folders()` (*in module salt.cloud.clouds.vmware*), 1142
`list_folders()` (*in module salt.modules.vsphere*), 2703
`list_folders()` (*in module salt.modules.win_task*), 2829
`list_frontends()` (*in module salt.modules.haproxyconn*), 1841
`list_function_versions()` (*in module salt.modules.boto_lambda*), 1422
`list_functions()` (*in module salt.modules.boto_lambda*), 1423
`list_functions()` (*in module salt.modules.sysmod*), 2583
`list_grants()` (*in module salt.modules.boto_kms*), 1417
`list_groups()` (*in module salt.modules.boto_asg*), 1344
`list_groups()` (*in module salt.modules.mac_user*), 2058
`list_groups()` (*in module salt.modules.pw_user*), 2408
`list_groups()` (*in module salt.modules.solaris_user*), 2520
`list_groups()` (*in module salt.modules.useradd*), 2633
`list_groups()` (*in module salt.modules.win_groupadd*), 2754
`list_groups()` (*in module salt.modules.win_useradd*), 2836
`list_hbas()` (*in module salt.cloud.clouds.vmware*), 1142
`list_holds()` (*in module salt.modules.yumpkg*), 2869
`list_hosted_services()` (*in module salt.cloud.clouds.msazure*), 1079
`list_hosted_zones()` (*in module salt.modules.boto3_route53*), 1328
`list_hosts()` (*in module salt.cloud.clouds.opennebula*), 1096
`list_hosts()` (*in module salt.cloud.clouds.vmware*), 1142
`list_hosts()` (*in module salt.modules.hosts*), 1850
`list_hosts()` (*in module salt.modules.vsphere*), 2703
`list_hosts_by_cluster()` (*in module salt.cloud.clouds.vmware*), 1142
`list_hosts_by_datacenter()` (*in module salt.cloud.clouds.vmware*), 1143
`list_hosts_via_proxy()` (*in module salt.modules.vsphere*), 2703
`list_icmp_block()` (*in module salt.modules.firewalld*), 1734
`list_ignored()` (*in module salt.modules.mac_softwareupdate*), 2046
`list_ikepolicies()` (*in module salt.modules.neutron*), 2230
`list_images()` (*in module salt.cloud.clouds.profitbricks*), 1118
`list_images()` (*in module salt.modules.cloud*), 1520
`list_images()` (*in module salt.modules.libcloud_compute*), 1963
`list_images()` (*in module salt.runners.cloud*), 3103
`list_images()` (*salt.cloud.CloudClient method*), 3852
`list_inactive_vms()` (*in module salt.modules.smartos_virt*), 2503
`list_inactive_vms()` (*in module salt.modules.vagrant*), 2636
`list_inactive_vms()` (*in module salt.modules.virt*), 2657
`list_incidents()` (*in module salt.modules.pagerduty*), 2302
`list_incidents()` (*in module salt.runners.pagerduty*), 3137
`list_input_endpoints()` (*in module salt.cloud.clouds.msazure*), 1079
`list_installed()` (*in module salt.modules.kernelpkg_linux_apt*), 1925
`list_installed()` (*in module salt.modules.kernelpkg_linux_yum*), 1927
`list_installed()` (*in module salt.modules.smartos_imgadm*), 2501
`list_installed()` (*in module salt.modules.win_servermanager*), 2793
`list_installed_patches()` (*in module*

salt.modules.yumpkg), 2870
`list_installed_patches()` (*in module salt.modules.zypperpkg*), 2928
`list_installed_patterns()` (*in module salt.modules.zypperpkg*), 2928
`list_instance_profiles()` (*in module salt.modules.boto_iam*), 1403
`list_interfaces()` (*in module salt.modules.iwtools*), 1902
`list_ipsec_site_connections()` (*in module salt.modules.neutron*), 2230
`list_ipsecpolicies()` (*in module salt.modules.neutron*), 2231
`list_items()` (*in module salt.modules.rallydev*), 2418
`list_items()` (*in module salt.modules.vboxmanage*), 2643
`list_items()` (*in module salt.queues.pgjsonb_queue*), 3087
`list_items()` (*in module salt.queues.sqlite_queue*), 3088
`list_items()` (*in module salt.runners.queue*), 3140
`list_jails()` (*in module salt.modules.poudriere*), 2386
`list_job()` (*in module salt.runners.jobs*), 3117
`list_jobs()` (*in module salt.runners.jobs*), 3117
`list_jobs_filter()` (*in module salt.runners.jobs*), 3118
`list_key_pairs()` (*in module salt.modules.libcloud_compute*), 1964
`list_key_policies()` (*in module salt.modules.boto_kms*), 1417
`list_keypairs()` (*in module salt.cloud.clouds.digitalocean*), 1041
`list_keys()` (*in module salt.cloud.clouds.joyent*), 1063
`list_keys()` (*in module salt.modules.gpg*), 1820
`list_keys()` (*in module salt.modules.reg*), 2435
`list_keyspaces()` (*in module salt.modules.cassandra_cql*), 1488
`list_l3_agent_hosting_routers()` (*in module salt.modules.neutron*), 2231
`list_launch_configurations()` (*in module salt.modules.boto_asg*), 1344
`list_length()` (*in module salt.queues.pgjsonb_queue*), 3088
`list_length()` (*in module salt.queues.sqlite_queue*), 3088
`list_length()` (*in module salt.runners.queue*), 3140
`list_licenses()` (*in module salt.modules.powerpath*), 2387
`list_licenses()` (*in module salt.modules.vsphere*), 2704
`list_loadbalancers()` (*in module salt.cloud.clouds.profitbricks*), 1118
`list_local()` (*in module salt.modules.layman*), 1953
`list_locations()` (*in module salt.modules.cloud*), 1520
`list_locations()` (*in module salt.modules.libcloud_compute*), 1964
`list_locations()` (*in module salt.runners.cloud*), 3103
`list_locations()` (*salt.cloud.CloudClient method*), 3852
`list_locked()` (*in module salt.modules.pkgng*), 2359
`list_locks()` (*in module salt.modules.zypperpkg*), 2928
`list_maintenance_windows()` (*in module salt.modules.pagerduty*), 2302
`list_maintenance_windows()` (*in module salt.runners.pagerduty*), 3137
`list_management_certificates()` (*in module salt.cloud.clouds.msazure*), 1080
`list_master()` (*in module salt.modules.cp*), 1577
`list_master_dirs()` (*in module salt.modules.cp*), 1577
`list_master_symlinks()` (*in module salt.modules.cp*), 1577
`list_members_without_mfa()` (*in module salt.modules.github*), 1805
`list_meta_fields()` (*in module salt.modules.bamboohr*), 1279
`list_minion()` (*in module salt.modules.cp*), 1577
`list_modules()` (*in module salt.modules.sysmod*), 2583
`list_modules()` (*in module salt.modules.win_psget*), 2790
`list_monitor()` (*in module salt.modules.bigip*), 1292
`list_monitor()` (*in module salt.states.bigip*), 3226
`list_monitor_data()` (*in module salt.cloud.clouds.aliyun*), 1033
`list_mount()` (*in module salt.modules.zfs*), 2902
`list_networks()` (*in module salt.cloud.clouds.openstack*), 1113
`list_networks()` (*in module salt.cloud.clouds.vmware*), 1143
`list_networks()` (*in module salt.modules.neutron*), 2231
`list_networks()` (*in module salt.modules.neutronng*), 2239
`list_networks()` (*in module salt.modules.virt*), 2657
`list_networks()` (*in module salt.modules.vsphere*), 2704
`list_nictags()` (*in module salt.modules.smartos_nictagadm*), 2502
`list_node()` (*in module salt.modules.bigip*), 1292
`list_node()` (*in module salt.states.bigip*), 3226
`list_nodes()` (*in module salt.cloud.clouds.aliyun*), 1033
`list_nodes()` (*in module salt.cloud.clouds.azurearm*),

1037
`list_nodes()` (*in module salt.cloud.clouds.cloudstack*), 1039
`list_nodes()` (*in module salt.cloud.clouds.digitalocean*), 1041
`list_nodes()` (*in module salt.cloud.clouds.dimensiondata*), 1044
`list_nodes()` (*in module salt.cloud.clouds.ec2*), 1050
`list_nodes()` (*in module salt.cloud.clouds.gce*), 1057
`list_nodes()` (*in module salt.cloud.clouds.gogrid*), 1060
`list_nodes()` (*in module salt.cloud.clouds.joyent*), 1063
`list_nodes()` (*in module salt.cloud.clouds.linode*), 1068
`list_nodes()` (*in module salt.cloud.clouds.lxc*), 1070
`list_nodes()` (*in module salt.cloud.clouds.msazure*), 1080
`list_nodes()` (*in module salt.cloud.clouds.oneandone*), 1089
`list_nodes()` (*in module salt.cloud.clouds.opennebula*), 1096
`list_nodes()` (*in module salt.cloud.clouds.openstack*), 1113
`list_nodes()` (*in module salt.cloud.clouds.parallels*), 1114
`list_nodes()` (*in module salt.cloud.clouds.profitbricks*), 1118
`list_nodes()` (*in module salt.cloud.clouds.proxmox*), 1120
`list_nodes()` (*in module salt.cloud.clouds.qingcloud*), 1123
`list_nodes()` (*in module salt.cloud.clouds.saltify*), 1125
`list_nodes()` (*in module salt.cloud.clouds.scaleway*), 1127
`list_nodes()` (*in module salt.cloud.clouds.softlayer*), 1128
`list_nodes()` (*in module salt.cloud.clouds.softlayer_hw*), 1129
`list_nodes()` (*in module salt.cloud.clouds.tencentcloud*), 1131
`list_nodes()` (*in module salt.cloud.clouds.vagrant*), 1134
`list_nodes()` (*in module salt.cloud.clouds.virtualbox*), 1135
`list_nodes()` (*in module salt.cloud.clouds.vmware*), 1143
`list_nodes()` (*in module salt.cloud.clouds.vultrpy*), 1148
`list_nodes()` (*in module salt.cloud.clouds.xen*), 1150
`list_nodes()` (*in module salt.modules.libcloud_compute*), 1964
`list_nodes()` (*in module salt.modules.vboxmanage*), 2643
`list_nodes_full()` (*in module salt.cloud.clouds.aliyun*), 1033
`list_nodes_full()` (*in module salt.cloud.clouds.azurearm*), 1037
`list_nodes_full()` (*in module salt.cloud.clouds.cloudstack*), 1039
`list_nodes_full()` (*in module salt.cloud.clouds.digitalocean*), 1041
`list_nodes_full()` (*in module salt.cloud.clouds.dimensiondata*), 1044
`list_nodes_full()` (*in module salt.cloud.clouds.ec2*), 1050
`list_nodes_full()` (*in module salt.cloud.clouds.gce*), 1057
`list_nodes_full()` (*in module salt.cloud.clouds.gogrid*), 1060
`list_nodes_full()` (*in module salt.cloud.clouds.joyent*), 1063
`list_nodes_full()` (*in module salt.cloud.clouds.linode*), 1069
`list_nodes_full()` (*in module salt.cloud.clouds.lxc*), 1071
`list_nodes_full()` (*in module salt.cloud.clouds.msazure*), 1080
`list_nodes_full()` (*in module salt.cloud.clouds.oneandone*), 1089
`list_nodes_full()` (*in module salt.cloud.clouds.opennebula*), 1096
`list_nodes_full()` (*in module salt.cloud.clouds.openstack*), 1113
`list_nodes_full()` (*in module salt.cloud.clouds.parallels*), 1114
`list_nodes_full()` (*in module salt.cloud.clouds.profitbricks*), 1118
`list_nodes_full()` (*in module salt.cloud.clouds.proxmox*), 1120
`list_nodes_full()` (*in module salt.cloud.clouds.qingcloud*), 1123
`list_nodes_full()` (*in module salt.cloud.clouds.saltify*), 1125
`list_nodes_full()` (*in module salt.cloud.clouds.scaleway*), 1127
`list_nodes_full()` (*in module salt.cloud.clouds.softlayer*), 1128
`list_nodes_full()` (*in module salt.cloud.clouds.softlayer_hw*), 1129
`list_nodes_full()` (*in module salt.cloud.clouds.tencentcloud*), 1132
`list_nodes_full()` (*in module salt.cloud.clouds.vagrant*), 1134
`list_nodes_full()` (*in module salt.cloud.clouds.virtualbox*), 1135
`list_nodes_full()` (*in module*

salt.cloud.clouds.vmware), 1143
`list_nodes_full()` (*in module salt.cloud.clouds.vultrpy*), 1148
`list_nodes_full()` (*in module salt.cloud.clouds.xen*), 1151
`list_nodes_full()` (*in module salt.modules.vboxmanage*), 2643
`list_nodes_min()` (*in module salt.cloud.clouds.aliyun*), 1033
`list_nodes_min()` (*in module salt.cloud.clouds.ec2*), 1050
`list_nodes_min()` (*in module salt.cloud.clouds.linode*), 1069
`list_nodes_min()` (*in module salt.cloud.clouds.openstack*), 1113
`list_nodes_min()` (*in module salt.cloud.clouds.qingcloud*), 1123
`list_nodes_min()` (*in module salt.cloud.clouds.tencentcloud*), 1132
`list_nodes_min()` (*in module salt.cloud.clouds.vmware*), 1143
`list_nodes_min()` (*in module salt.modules.vboxmanage*), 2644
`list_nodes_select()` (*in module salt.cloud.clouds.aliyun*), 1033
`list_nodes_select()` (*in module salt.cloud.clouds.cloudstack*), 1039
`list_nodes_select()` (*in module salt.cloud.clouds.digitalocean*), 1041
`list_nodes_select()` (*in module salt.cloud.clouds.dimensiondata*), 1044
`list_nodes_select()` (*in module salt.cloud.clouds.ec2*), 1050
`list_nodes_select()` (*in module salt.cloud.clouds.gce*), 1057
`list_nodes_select()` (*in module salt.cloud.clouds.gogrid*), 1060
`list_nodes_select()` (*in module salt.cloud.clouds.joyent*), 1064
`list_nodes_select()` (*in module salt.cloud.clouds.linode*), 1069
`list_nodes_select()` (*in module salt.cloud.clouds.lxc*), 1071
`list_nodes_select()` (*in module salt.cloud.clouds.msazure*), 1080
`list_nodes_select()` (*in module salt.cloud.clouds.oneandone*), 1089
`list_nodes_select()` (*in module salt.cloud.clouds.opennebula*), 1096
`list_nodes_select()` (*in module salt.cloud.clouds.openstack*), 1113
`list_nodes_select()` (*in module salt.cloud.clouds.parallels*), 1114
`list_nodes_select()` (*in module salt.cloud.clouds.proxmox*), 1120
`list_nodes_select()` (*in module salt.cloud.clouds.qingcloud*), 1123
`list_nodes_select()` (*in module salt.cloud.clouds.saltify*), 1126
`list_nodes_select()` (*in module salt.cloud.clouds.scaleway*), 1127
`list_nodes_select()` (*in module salt.cloud.clouds.softlayer*), 1128
`list_nodes_select()` (*in module salt.cloud.clouds.softlayer_hw*), 1129
`list_nodes_select()` (*in module salt.cloud.clouds.tencentcloud*), 1132
`list_nodes_select()` (*in module salt.cloud.clouds.vagrant*), 1134
`list_nodes_select()` (*in module salt.cloud.clouds.virtualbox*), 1135
`list_nodes_select()` (*in module salt.cloud.clouds.vmware*), 1144
`list_nodes_select()` (*in module salt.cloud.clouds.vultrpy*), 1148
`list_nodes_select()` (*in module salt.cloud.clouds.xen*), 1151
`list_non_ssds()` (*in module salt.modules.vsphere*), 2704
`list_not_state()` (*in module salt.runners.manage*), 3123
`list_object_versions()` (*in module salt.modules.boto_s3_bucket*), 1438
`list_objects()` (*in module salt.modules.boto_s3_bucket*), 1438
`list_orders()` (*in module salt.runners.digicertapi*), 3106
`list_organizations()` (*in module salt.runners.digicertapi*), 3106
`list_ostypes()` (*in module salt.modules.vboxmanage*), 2644
`list_passwords()` (*in module salt.cloud.clouds.gogrid*), 1060
`list_patches()` (*in module salt.modules.yumpkg*), 2870
`list_patches()` (*in module salt.modules.zypperpkg*), 2928
`list_patterns()` (*in module salt.modules.zypperpkg*), 2928
`list_permissions()` (*in module salt.modules.cassandra_cql*), 1488
`list_permissions()` (*in module salt.modules.rabbitmq*), 2414
`list_pipelines()` (*in module salt.modules.boto_datapipeline*), 1359
`list_pkgs()` (*in module salt.modules.aixpkg*), 1205
`list_pkgs()` (*in module salt.modules.apkpkg*), 1214
`list_pkgs()` (*in module salt.modules.aptpkg*), 1221

`list_pkgs()` (*in module salt.modules.dpkg_lowpkg*), 1661
`list_pkgs()` (*in module salt.modules.dummyproxy_pkg*), 1673
`list_pkgs()` (*in module salt.modules.ebuildpkg*), 1677
`list_pkgs()` (*in module salt.modules.freebsdpkg*), 1743
`list_pkgs()` (*in module salt.modules.mac_brew_pkg*), 2024
`list_pkgs()` (*in module salt.modules.mac_portspkg*), 2033
`list_pkgs()` (*in module salt.modules.nix*), 2252
`list_pkgs()` (*in module salt.modules.openbsdpkg*), 2271
`list_pkgs()` (*in module salt.modules.opkg*), 2283
`list_pkgs()` (*in module salt.modules.pacmanpkg*), 2299
`list_pkgs()` (*in module salt.modules.pkgin*), 2352
`list_pkgs()` (*in module salt.modules.pkgng*), 2359
`list_pkgs()` (*in module salt.modules.pkgutil*), 2369
`list_pkgs()` (*in module salt.modules.rest_pkg*), 2439
`list_pkgs()` (*in module salt.modules.rpm_lowpkg*), 2448
`list_pkgs()` (*in module salt.modules.solarisipspkg*), 2522
`list_pkgs()` (*in module salt.modules.solarispkg*), 2526
`list_pkgs()` (*in module salt.modules.ssh_pkg*), 2542
`list_pkgs()` (*in module salt.modules.win_pkg*), 2781
`list_pkgs()` (*in module salt.modules.xbpspkg*), 2858
`list_pkgs()` (*in module salt.modules.yumpkg*), 2870
`list_pkgs()` (*in module salt.modules.zypperpkg*), 2928
`list_platform_sets()` (*in module salt.runners.asam*), 3096
`list_platforms()` (*in module salt.runners.asam*), 3096
`list_plugins()` (*in module salt.modules.munin*), 2098
`list_plugins()` (*in module salt.modules.nagios*), 2111
`list_policies()` (*in module salt.modules.boto_iam*), 1403
`list_policies()` (*in module salt.modules.boto_iot*), 1410
`list_policies()` (*in module salt.modules.kerberos*), 1924
`list_policies()` (*in module salt.modules.pagerduty*), 2302
`list_policies()` (*in module salt.modules.rabbitmq*), 2415
`list_policies()` (*in module salt.runners.pagerduty*), 3137
`list_policy_versions()` (*in module salt.modules.boto_iam*), 1403
`list_policy_versions()` (*in module salt.modules.boto_iot*), 1410
`list_pool()` (*in module salt.modules.bigip*), 1293
`list_pool()` (*in module salt.states.bigip*), 3226
`list_pools()` (*in module salt.modules.virt*), 2657
`list_port_fwd()` (*in module salt.modules.firewalld*), 1735
`list_portgroups()` (*in module salt.cloud.clouds.vmware*), 1144
`list_ports()` (*in module salt.modules.firewalld*), 1735
`list_ports()` (*in module salt.modules.neutron*), 2231
`list_ports()` (*in module salt.modules.poudriere*), 2386
`list_present()` (*in module salt.states.grains*), 3481
`list_present()` (*in module salt.states.linux_acl*), 3539
`list_principal_policies()` (*in module salt.modules.boto_iot*), 1410
`list_principals()` (*in module salt.modules.kerberos*), 1924
`list_private_repos()` (*in module salt.modules.github*), 1805
`list_privileges()` (*in module salt.modules.influxdbmod*), 1860
`list_products()` (*in module salt.modules.zypperpkg*), 2929
`list_profile()` (*in module salt.modules.bigip*), 1293
`list_profile()` (*in module salt.states.bigip*), 3226
`list_protocols()` (*in module salt.modules.libcloud_loadbalancer*), 1971
`list_provides()` (*in module salt.modules.zypperpkg*), 2929
`list_public_ips()` (*in module salt.cloud.clouds.gogrid*), 1060
`list_public_repos()` (*in module salt.modules.github*), 1806
`list_queues()` (*in module salt.modules.aws_sqs*), 1247
`list_queues()` (*in module salt.modules.rabbitmq*), 2415
`list_queues()` (*in module salt.queues.pgjsonb_queue*), 3088
`list_queues()` (*in module salt.queues.sqlite_queue*), 3088
`list_queues()` (*in module salt.runners.queue*), 3141
`list_queues_vhost()` (*in module salt.modules.rabbitmq*), 2415
`list_quota_volume()` (*in module salt.modules.glusterfs*), 1813
`list_quotas()` (*in module salt.modules.neutron*), 2231
`list_record_types()` (*in module salt.modules.libcloud_dns*), 1967
`list_records()` (*in module salt.modules.libcloud_dns*), 1967

`list_records()` (*in module salt.modules.solaris_fmadm*), 2514
`list_renderers()` (*in module salt.modules.sysmod*), 2583
`list_repo_pkgs()` (*in module salt.modules.aptpkg*), 1221
`list_repo_pkgs()` (*in module salt.modules.pacmanpkg*), 2299
`list_repo_pkgs()` (*in module salt.modules.yumpkg*), 2870
`list_repo_pkgs()` (*in module salt.modules.zypperpkg*), 2929
`list_repos()` (*in module salt.modules.aptpkg*), 1221
`list_repos()` (*in module salt.modules.github*), 1806
`list_repos()` (*in module salt.modules.opkg*), 2283
`list_repos()` (*in module salt.modules.xbpspkg*), 2858
`list_repos()` (*in module salt.modules.yumpkg*), 2872
`list_repos()` (*in module salt.modules.zypperpkg*), 2930
`list_requests()` (*in module salt.runners.digicertapi*), 3106
`list_resource_groups()` (*in module salt.cloud.clouds.azurearm*), 1037
`list_resourcepools()` (*in module salt.cloud.clouds.vmware*), 1144
`list_resourcepools()` (*in module salt.modules.vsphere*), 2705
`list_returner_functions()` (*in module salt.modules.sysmod*), 2583
`list_returners()` (*in module salt.modules.sysmod*), 2584
`list_role_policies()` (*in module salt.modules.boto_iam*), 1404
`list_rooms()` (*in module salt.modules.slack_notify*), 2497
`list_roots()` (*in module salt.wheel.file_roots*), 3838
`list_roots()` (*in module salt.wheel.pillar_roots*), 3842
`list_routers()` (*in module salt.modules.neutron*), 2231
`list_rules()` (*in module salt.modules.boto_cloudwatch_event*), 1354
`list_runner_functions()` (*in module salt.modules.sysmod*), 2584
`list_runners()` (*in module salt.modules.sysmod*), 2584
`list_running()` (*in module salt.modules.nspawn*), 2261
`list_saml_providers()` (*in module salt.modules.boto_iam*), 1404
`list_schedules()` (*in module salt.modules.pagerduty*), 2302
`list_schedules()` (*in module salt.runners.pagerduty*), 3137
`list_scripts()` (*in module salt.cloud.clouds.vultrpy*), 1148
`list_sebool()` (*in module salt.modules.selinux*), 2490
`list_secret_keys()` (*in module salt.modules.gpg*), 1821
`list_secrets()` (*in module salt.modules.vault*), 2640
`list_security_group_rules()` (*in module salt.modules.neutron*), 2232
`list_security_groups()` (*in module salt.cloud.clouds.opennebula*), 1096
`list_security_groups()` (*in module salt.modules.neutron*), 2232
`list_securitygroup()` (*in module salt.cloud.clouds.aliyun*), 1033
`list_securitygroups()` (*in module salt.cloud.clouds.tencentcloud*), 1132
`list_semod()` (*in module salt.modules.selinux*), 2490
`list_servers()` (*in module salt.modules.haproxyconn*), 1841
`list_service_certificates()` (*in module salt.cloud.clouds.msazure*), 1080
`list_services()` (*in module salt.cloud.clouds.msazure*), 1080
`list_services()` (*in module salt.modules.firewalld*), 1735
`list_services()` (*in module salt.modules.pagerduty*), 2303
`list_services()` (*in module salt.runners.pagerduty*), 3138
`list_sessions()` (*in module salt.modules.rdp*), 2426
`list_sets()` (*in module salt.modules.ipset*), 1897
`list_shared_folders_users()` (*in module salt.modules.vbox_guest*), 2642
`list_sites()` (*in module salt.modules.win_iis*), 2759
`list_sizes()` (*in module salt.modules.cloud*), 1521
`list_sizes()` (*in module salt.modules.libcloud_compute*), 1964
`list_sizes()` (*in module salt.runners.cloud*), 3103
`list_sizes()` (*salt.cloud.CloudClient method*), 3853
`list_slotnames()` (*in module salt.modules.dracr*), 1668
`list_snapshots()` (*in module salt.cloud.clouds.vmware*), 1144
`list_snapshots()` (*in module salt.modules.parallels*), 2321
`list_snapshots()` (*in module salt.modules.snapper*), 2545
`list_snapshots()` (*in module salt.modules.virt*), 2658
`list_ssds()` (*in module salt.modules.vsphere*), 2705
`list_stack()` (*in module salt.modules.heat*), 1846
`list_startup_disks()` (*in module salt.modules.mac_system*), 2050
`list_state()` (*in module salt.runners.manage*), 3123
`list_state_functions()` (*in module*

salt.modules.sysmod), 2584
list_state_modules() (in module salt.modules.sysmod), 2585
list_states() (in module salt.modules.cp), 1578
list_stopped() (in module salt.modules.nspawn), 2261
list_storage() (in module salt.cloud.clouds.msazure), 1080
list_storage_accounts() (in module salt.cloud.clouds.azurearm), 1037
list_storage_containers() (in module salt.cloud.clouds.msazure), 1080
list_storage_policies() (in module salt.modules.vsphere), 2705
list_storage_services() (in module salt.cloud.clouds.msazure), 1081
list_streams() (in module salt.modules.boto_kinesis), 1414
list_subnets() (in module salt.cloud.clouds.azurearm), 1037
list_subnets() (in module salt.cloud.clouds.openstack), 1113
list_subnets() (in module salt.modules.neutron), 2232
list_subnets() (in module salt.modules.neutronng), 2239
list_supported_algorithms() (in module salt.modules.libcloud_loadbalancer), 1971
list_tab() (in module salt.modules.cron), 1580
list_tab() (in module salt.modules.incron), 1856
list_tables() (in module salt.modules.nftables), 2247
list_tables() (salt.modules.inspectlib.fsdb.CsvDB method), 1877
list_tag_categories() (in module salt.modules.vsphere), 2705
list_tags() (in module salt.modules.boto3_elasticsearch), 1318
list_tags() (in module salt.modules.boto_cloudtrail), 1350
list_tags() (in module salt.modules.boto_elasticsearch_domain), 1384
list_tags() (in module salt.modules.dockermod), 1643
list_tags() (in module salt.modules.git), 1774
list_tags() (in module salt.modules.vsphere), 2706
list_tags_for_resource() (in module salt.modules.boto3_elasticache), 1306
list_targets() (in module salt.modules.boto_cloudwatch_event), 1354
list_tasks() (in module salt.modules.win_task), 2829
list_team_members() (in module salt.modules.github), 1806
list_team_repos() (in module salt.modules.github), 1806
list_teams() (in module salt.modules.github), 1806
list_templates() (in module salt.cloud.clouds.opennebula), 1097
list_templates() (in module salt.cloud.clouds.vmware), 1144
list_tokens() (in module salt.tokens.localfs), 3828
list_tokens() (in module salt.tokens.rediscluster), 3829
list_topic_rules() (in module salt.modules.boto_iot), 1411
list_transaction() (in module salt.modules.bigip), 1293
list_triggers() (in module salt.modules.win_task), 2830
list_unix() (salt.modules.win_timezone.TzMapper method), 2831
list_upgrades() (in module salt.modules.apkpkg), 1214
list_upgrades() (in module salt.modules.aptpkg), 1221
list_upgrades() (in module salt.modules.ebuildpkg), 1677
list_upgrades() (in module salt.modules.gem), 1750
list_upgrades() (in module salt.modules.mac_brew_pkg), 2024
list_upgrades() (in module salt.modules.mac_portspkg), 2033
list_upgrades() (in module salt.modules.opkg), 2283
list_upgrades() (in module salt.modules.pacmanpkg), 2300
list_upgrades() (in module salt.modules.pip), 2348
list_upgrades() (in module salt.modules.pkgin), 2352
list_upgrades() (in module salt.modules.pkgng), 2360
list_upgrades() (in module salt.modules.pkgutil), 2369
list_upgrades() (in module salt.modules.solarisipspkg), 2522
list_upgrades() (in module salt.modules.win_pkg), 2781
list_upgrades() (in module salt.modules.xbpspkg), 2859
list_upgrades() (in module salt.modules.yumpkg), 2872
list_upgrades() (in module salt.modules.zypperpkg), 2930
list_upgrades() (in module salt.runners.pkg), 3139
list_uplink_dvportgroup() (in module salt.modules.vsphere), 2706

`list_upstreams()` (*in module salt.modules.rabbitmq*), 2415
`list_user_permissions()` (*in module salt.modules.rabbitmq*), 2415
`list_users()` (*in module salt.modules.bamboohr*), 1279
`list_users()` (*in module salt.modules.cassandra_cql*), 1489
`list_users()` (*in module salt.modules.drac*), 1663
`list_users()` (*in module salt.modules.dracr*), 1668
`list_users()` (*in module salt.modules.github*), 1807
`list_users()` (*in module salt.modules.ilo*), 1854
`list_users()` (*in module salt.modules.influxdbmod*), 1860
`list_users()` (*in module salt.modules.mac_user*), 2058
`list_users()` (*in module salt.modules.pagerduty*), 2303
`list_users()` (*in module salt.modules.pdbedit*), 2332
`list_users()` (*in module salt.modules.pw_user*), 2409
`list_users()` (*in module salt.modules.rabbitmq*), 2415
`list_users()` (*in module salt.modules.rallydev*), 2418
`list_users()` (*in module salt.modules.slack_notify*), 2497
`list_users()` (*in module salt.modules.solaris_user*), 2520
`list_users()` (*in module salt.modules.splunk*), 2536
`list_users()` (*in module salt.modules.useradd*), 2633
`list_users()` (*in module salt.modules.win_useradd*), 2836
`list_users()` (*in module salt.runners.pagerduty*), 3138
`list_users_info()` (*in module salt.modules.ilo*), 1854
`list_values()` (*in module salt.modules.reg*), 2436
`list_vapps()` (*in module salt.cloud.clouds.vmware*), 1144
`list_vapps()` (*in module salt.modules.vsphere*), 2706
`list_vdirs()` (*in module salt.modules.win_iis*), 2759
`list_vhosts()` (*in module salt.modules.rabbitmq*), 2415
`list_virtual()` (*in module salt.modules.bigip*), 1293
`list_virtual()` (*in module salt.states.bigip*), 3226
`list_virtual_machine_scale_set_network_interfaces()` (*in module salt.modules.azurearm_network*), 1256
`list_virtual_machine_scale_set_vm_network_interfaces()` (*in module salt.modules.azurearm_network*), 1256
`list_virtual_networks()` (*in module salt.cloud.clouds.azurearm*), 1037
`list_virtual_networks()` (*in module salt.cloud.clouds.msazure*), 1081
`list_vlans()` (*in module salt.cloud.clouds.softlayer*), 1128
`list_vlans()` (*in module salt.cloud.clouds.softlayer_hw*), 1129
`list_vms()` (*in module salt.modules.parallels*), 2322
`list_vms()` (*in module salt.modules.smartos_vmadm*), 2506
`list_vms()` (*in module salt.modules.vsphere*), 2706
`list_vms()` (*in module salt.runners.smartos_vmadm*), 3150
`list_vns()` (*in module salt.cloud.clouds.opennebula*), 1097
`list_volume_snapshots()` (*in module salt.modules.libcloud_compute*), 1964
`list_volumes()` (*in module salt.modules.glusterfs*), 1813
`list_volumes()` (*in module salt.modules.libcloud_compute*), 1965
`list_vpnservices()` (*in module salt.modules.neutron*), 2232
`list_webpi()` (*in module salt.modules.chocolatey*), 1504
`list_win()` (*salt.modules.win_timezone.TzMapper method*), 2831
`list_windows()` (*in module salt.modules.pagerduty*), 2303
`list_windows()` (*in module salt.runners.pagerduty*), 3138
`list_windowsfeatures()` (*in module salt.modules.chocolatey*), 1504
`list_worker_processes()` (*in module salt.modules.win_iis*), 2759
`list_worktrees()` (*in module salt.modules.git*), 1775
`list_zones()` (*in module salt.modules.firewalld*), 1735
`list_zones()` (*in module salt.modules.libcloud_dns*), 1968
`list_zones()` (*in module salt.modules.mac_timezone*), 2055
`list_zones()` (*in module salt.modules.zoneadm*), 2908
`Listener` (*class in salt.engines.redis_sentinel*), 1164
`listener_dict_to_tuple()` (*in module salt.modules.boto_elb*), 1389
`listening_ports()` (*in module salt.modules.osquery*), 2292
`lldp()` (*in module salt.modules.napalm_network*), 2164
`lldp()` (*in module salt.runners.net*), 3133
`llen()` (*in module salt.modules.redismod*), 2431
`load()` (*in module salt.modules.data*), 1587
`load()` (*in module salt.modules.dockermod*), 1643
`load()` (*in module salt.modules.freebsdkmod*), 1741
`load()` (*in module salt.modules.junos*), 1916
`load()` (*in module salt.modules.kmod*), 1945
`load()` (*in module salt.modules.pf*), 2337
`load()` (*in module salt.modules.solaris_fmadm*), 2514

`load()` (*in module salt.modules.vmctl*), 2678
`load()` (*in module salt.states.junos*), 3512
`load()` (*salt.modules.inspectlib.kiwiproc.KiwiExporter method*), 1878
`load_balancer_absent()` (*in module salt.states.azurearm_network*), 3204
`load_balancer_create_or_update()` (*in module salt.modules.azurearm_network*), 1256
`load_balancer_delete()` (*in module salt.modules.azurearm_network*), 1257
`load_balancer_get()` (*in module salt.modules.azurearm_network*), 1257
`load_balancer_present()` (*in module salt.states.azurearm_network*), 3204
`load_balancers_list()` (*in module salt.modules.azurearm_network*), 1257
`load_balancers_list_all()` (*in module salt.modules.azurearm_network*), 1257
`load_config()` (*in module salt.modules.napalm_network*), 2164
`load_config()` (*in module salt.modules.napalm_yang_mod*), 2190
`load_filter_config()` (*in module salt.modules.napalm_netacl*), 2139
`load_map()` (*in module salt.modules.jinja*), 1911
`load_module()` (*salt.modules.ansiblegate.AnsibleModuleResolver method*), 1208
`load_policy_config()` (*in module salt.modules.napalm_netacl*), 2142
`load_public_key()` (*in module salt.cloud.clouds.oneandone*), 1089
`load_reg()` (*in module salt.returners.local_cache*), 325
`load_states()` (*in module salt.renderers.pyobjects*), 380
`load_template()` (*in module salt.modules.napalm_network*), 2167
`load_term_config()` (*in module salt.modules.napalm_netacl*), 2144
`loadavg()` (*in module salt.modules.status*), 2559
`loadavg()` (*in module salt.modules.win_status*), 2810
`loadavg()` (*in module salt.states.status*), 3727
`loaddata()` (*in module salt.modules.djangomod*), 1605
`LoaderError`, 3999
`local`
    `spm command line option`, 991
`local()` (*salt.netapi.NetapiClient method*), 3854
`local_async()` (*salt.netapi.NetapiClient method*), 3854
`local_subset()` (*salt.netapi.NetapiClient method*), 3855
`LocalClient` (*class in salt.client*), 3845
`locale_info()` (*in module salt.grains.core*), 1178
`locate()` (*in module salt.modules.locate*), 1980
`location()` (*in module salt.grains.fx2*), 1180
`location()` (*in module salt.grains.rest_sample*), 1185
`lock()` (*in module salt.modules.junos*), 1917
`lock()` (*in module salt.modules.mac_desktop*), 2026
`lock()` (*in module salt.modules.nova*), 2255
`lock()` (*in module salt.modules.pkgng*), 2360
`lock()` (*in module salt.modules.win_system*), 2815
`lock()` (*in module salt.modules.zk_concurrency*), 2905
`lock()` (*in module salt.runners.fileserver*), 3114
`lock()` (*in module salt.states.junos*), 3513
`lock()` (*in module salt.states.zk_concurrency*), 3806
`lock_holders()` (*in module salt.modules.zk_concurrency*), 2906
`lock_saltenv`
    `conf/minion`, 151
`locked()` (*in module salt.modules.pkgng*), 2360
`log` (*in module salt.modules.solrcloud*), 2535
`log_datefmt`
    `conf/logging`, 247
    `conf/master`, 115
    `conf/minion`, 161
`log_datefmt_logfile`
    `conf/logging`, 247
    `conf/master`, 115
    `conf/minion`, 161
`log_file`
    `conf/logging`, 246
    `conf/master`, 114
    `conf/minion`, 161
`log_fmt_console`
    `conf/logging`, 247
    `conf/master`, 115
    `conf/minion`, 161
`log_fmt_jid`
    `conf/logging`, 248
`log_fmt_logfile`
    `conf/logging`, 247
    `conf/master`, 115
    `conf/minion`, 162
`log_granular_levels`
    `conf/logging`, 248
    `conf/master`, 115
    `conf/minion`, 162
`log_level`
    `conf/logging`, 246
    `conf/master`, 114
    `conf/minion`, 161
`log_level_logfile`
    `conf/logging`, 246
    `conf/master`, 114
    `conf/minion`, 161
`log_levels`
    `conf/logging`, 245
`log_rotate_backup_count`
    `conf/master`, 116

conf/minion, 162
log_rotate_max_bytes
conf/master, 116
conf/minion, 162
logged_in_users() (*in module salt.modules.osquery*), 2292
logging_levels() (*in module salt.states.cimc*), 3340
LoggingRuntimeError, 3999
Login (*class in salt.netapi.rest_cherrypy.app*), 2945
login() (*in module salt.modules.dockermod*), 1644
login_create() (*in module salt.modules.mssql*), 2096
login_exists() (*in module salt.modules.mssql*), 2096
login_remove() (*in module salt.modules.mssql*), 2096
login_test() (*in module salt.modules.influxdb08mod*), 1863
loglevel() (*in module salt.modules.pf*), 2337
logoff_session() (*in module salt.modules.rdp*), 2427
logon() (*in module salt.proxy.cimc*), 3052
Logout (*class in salt.netapi.rest_cherrypy.app*), 2947
logout() (*in module salt.proxy.cimc*), 3052
logs
jinja filters, 452
logs() (*in module salt.modules.dockermod*), 1644
long_int() (*in module salt.modules.boto_kinesis*), 1414
long_range() (*in module salt.modules.ipset*), 1897
lookup() (*in module salt.modules.smartos_vmadm*), 2506
lookup_jid() (*in module salt.runners.jobs*), 3118
loop_interval
conf/master, 55
conf/minion, 139
Low State, **5402**
low() (*in module salt.modules.state*), 2552
low() (*salt.cloud.CloudClient method*), 3853
low_light() (*in module salt.modules.sensehat*), 2493
LowDataAdapter (*class in salt.netapi.rest_cherrypy.app*), 2944
lowstate, **2961**
lrange() (*in module salt.modules.redismod*), 2431
ls() (*in module salt.cache.etcd_cache*), 1027
ls() (*in module salt.cache.mysql_cache*), 1028
ls() (*in module salt.modules.augeas_cfg*), 1245
ls() (*in module salt.modules.cron*), 1580
ls() (*in module salt.modules.grains*), 1835
ls() (*in module salt.modules.incron*), 1856
ls() (*in module salt.modules.pillar*), 2343
ls() (*in module salt.modules.serverdensity_device*), 2495
ls() (*in module salt.modules.tomcat*), 2620
ls_() (*in module salt.modules.etcd_mod*), 1697
ls_() (*in module salt.modules.lxc*), 1995
ls_() (*in module salt.modules.scsi*), 2485
ls_remote() (*in module salt.modules.git*), 1776
lsattr() (*in module salt.modules.file*), 1719
lsmod() (*in module salt.modules.freebsdkmod*), 1741
lsmod() (*in module salt.modules.kmod*), 1945
lsof() (*in module salt.modules.ps*), 2391
lstat() (*in module salt.modules.file*), 1720
lt() (*in module salt.thorium.check*), 3823
lte() (*in module salt.thorium.check*), 3824
lucene_version() (*in module salt.modules.solr*), 2530
lv_absent() (*in module salt.states.lvm*), 3543
lv_present() (*in module salt.states.lvm*), 3543
lvcreate() (*in module salt.modules.linux_lvm*), 1977
lvdisplay() (*in module salt.modules.linux_lvm*), 1977
lvremove() (*in module salt.modules.linux_lvm*), 1977
lvresize() (*in module salt.modules.linux_lvm*), 1977

M

mac() (*in module salt.modules.napalm_network*), 2171
mac_str_to_bytes
jinja filters, 446
main() (*in module salt.modules.inspectlib.collector*), 1875
make_blob_url() (*in module salt.cloud.clouds.msazure*), 1081
make_digest_acl() (*in module salt.modules.zookeeper*), 2915
make_hashable() (*in module salt.states.grains*), 3481
make_image() (*in module salt.modules.qemu_img*), 2411
make_interface_child() (*in module salt.modules.netbox*), 2201
make_interface_lag() (*in module salt.modules.netbox*), 2201
make_permanent() (*in module salt.modules.firewalld*), 1735
make_pkgng_aware() (*in module salt.modules.poudriere*), 2386
make_repo() (*in module salt.modules.debuild_pkgbuild*), 1593
make_repo() (*in module salt.modules.rpmbuild_pkgbuild*), 2449
make_src_pkg() (*in module salt.modules.debuild_pkgbuild*), 1594
make_src_pkg() (*in module salt.modules.rpmbuild_pkgbuild*), 2451
makedirs_() (*in module salt.modules.file*), 1720
makedirs_() (*in module salt.modules.win_file*), 2739
makedirs_perms() (*in module salt.modules.file*), 1720
makedirs_perms() (*in module salt.modules.win_file*), 2740
makeopts_contains() (*in module salt.modules.makeconf*), 2064
manage() (*in module salt.states.ssh_auth*), 3724

manage_file() (*in module salt.modules.file*), 1720
manage_mode() (*in module salt.modules.config*), 1559
manage_monitor() (*in module salt.states.bigip*), 3226
manage_node() (*in module salt.states.bigip*), 3226
manage_pool() (*in module salt.states.bigip*), 3227
manage_pool_members() (*in module salt.states.bigip*), 3228
manage_profile() (*in module salt.states.bigip*), 3228
manage_virtual() (*in module salt.states.bigip*), 3228
managed() (*in module salt.states.file*), 3431
managed() (*in module salt.states.keystore*), 3526
managed() (*in module salt.states.ldap*), 3531
managed() (*in module salt.states.memcached*), 3553
managed() (*in module salt.states.net_napalm_yang*), 3603
managed() (*in module salt.states.netacl*), 3573
managed() (*in module salt.states.netconfig*), 3582
managed() (*in module salt.states.netntp*), 3590
managed() (*in module salt.states.netsnmp*), 3591
managed() (*in module salt.states.netusers*), 3592
managed() (*in module salt.states.network*), 3601
managed() (*in module salt.states.ntp*), 3613
managed() (*in module salt.states.pdbedit*), 3638
managed() (*in module salt.states.pkgrepo*), 3662
managed() (*in module salt.states.probes*), 3676
managed() (*in module salt.states.proxy*), 3678
managed() (*in module salt.states.rbac_solaris*), 3686
managed() (*in module salt.states.statuspage*), 3728
managed() (*in module salt.states.sysrc*), 3732
managed() (*in module salt.states.virtualenv_mod*), 3755
managed() (*in module salt.states.win_network*), 3773
Map (*class in salt.utils.aggregation*), 3997
map_clonemode() (*in module salt.cloud.clouds.virtualbox*), 1135
map_run() (*in module salt.modules.cloud*), 1521
map_run() (*in module salt.runners.cloud*), 3103
map_run() (*salt.cloud.CloudClient method*), 3853
MappedPagesWriteCount (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATION attribute*), 2809
MappedWriteIoCount (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATION attribute*), 2809
mapping_create() (*in module salt.modules.elasticsearch*), 1688
mapping_delete() (*in module salt.modules.elasticsearch*), 1688
mapping_get() (*in module salt.modules.elasticsearch*), 1688
marathon() (*in module salt.grains.marathon*), 1180
mask() (*in module salt.modules.systemd_service*), 2594
masked() (*in module salt.modules.systemd_service*), 2595
masked() (*in module salt.states.service*), 3711
Master, **5402**
master
    conf/minion, 123
Master Tops, **5402**
master() (*in module salt.modules.status*), 2560
master() (*in module salt.modules.win_status*), 2810
master_alive_interval
    conf/minion, 126
master_failback
    conf/minion, 126
master_failback_interval
    conf/minion, 126
master_finger
    conf/minion, 157
master_id
    conf/master, 52
master_job_cache
    conf/master, 60
master_port
    conf/minion, 128
master_pubkey_signature
    conf/master, 72
master_roots
    conf/master, 84
master_shuffle
    conf/minion, 126
master_sign_key_name
    conf/master, 72
    conf/minion, 158
master_sign_pubkey
    conf/master, 72
master_stats
    conf/master, 62
master_stats_event_iter
    conf/master, 62
master_tops
    conf/master, 77
master_tops_first
    conf/minion, 125
master_tries
    conf/minion, 137
master_type
    conf/minion, 125
master_uri_format
    conf/minion, 124
master_use_pubkey_signature
    conf/master, 72
MasterExit, 3999
Masterless, **5402**
match
    jinja filters, 449
match() (*in module salt.modules.augeas_cfg*), 1245
match_config() (*in module salt.modules.trafficserver*), 2623

match_index_versions() (*in module salt.modules.solr*), 2530
match_metric() (*in module salt.modules.trafficserver*), 2623
max
    jinja filters, 432
max_event_size
    conf/master, 60
    conf/minion, 125
max_minions
    conf/master, 61
max_op_version() (*in module salt.states.glusterfs*), 3466
max_open_files
    conf/master, 74
maybe_fix_ssl_version() (*in module salt.modules.tls*), 2617
md5
    jinja filters, 440
md5_digest() (*in module salt.modules.hashutil*), 1844
mdadm() (*in module salt.grains.mdadm*), 1180
mds_create() (*in module salt.modules.ceph*), 1493
mds_destroy() (*in module salt.modules.ceph*), 1493
mean() (*in module salt.thorium.calc*), 3819
mean() (*in module salt.thorium.reg*), 3826
median() (*in module salt.thorium.calc*), 3819
median_grouped() (*in module salt.thorium.calc*), 3819
median_high() (*in module salt.thorium.calc*), 3820
median_low() (*in module salt.thorium.calc*), 3820
mediatype_create() (*in module salt.modules.zabbix*), 2886
mediatype_delete() (*in module salt.modules.zabbix*), 2886
mediatype_get() (*in module salt.modules.zabbix*), 2887
mediatype_update() (*in module salt.modules.zabbix*), 2887
member_absent() (*in module salt.states.libcloud_loadbalancer*), 3536
member_present() (*in module salt.states.libcloud_loadbalancer*), 3536
member_status() (*in module salt.modules.riak*), 2446
members() (*in module salt.modules.aix_group*), 1204
members() (*in module salt.modules.groupadd*), 1838
members() (*in module salt.modules.mac_group*), 2027
members() (*in module salt.modules.pw_group*), 2406
members() (*in module salt.modules.win_groupadd*), 2754
memcache_debug
    conf/master, 59
memcache_expire_seconds
    conf/master, 58
memcache_full_cleanup
    conf/master, 58
memcache_max_items
    conf/master, 58
meminfo() (*in module salt.modules.status*), 2560
meminfo() (*in module salt.modules.win_status*), 2811
memory() (*in module salt.modules.sysbench*), 2575
memory_map() (*in module salt.modules.osquery*), 2292
merge() (*in module salt.modules.config*), 1559
merge() (*in module salt.modules.defaults*), 1600
merge() (*in module salt.modules.git*), 1777
merge() (*in module salt.modules.slsutil*), 2498
merge_all() (*in module salt.modules.slsutil*), 2498
merge_base() (*in module salt.modules.git*), 1778
merge_diff() (*in module salt.modules.iosconfig*), 1882
merge_text() (*in module salt.modules.iosconfig*), 1882
merge_tree() (*in module salt.modules.git*), 1780
merge_tree() (*in module salt.modules.iosconfig*), 1883
message_text() (*salt.engines.slack.SlackClient method*), 1168
metadata() (*in module salt.grains.metadata*), 1181
metadata_server_grains
    conf/minion, 134
metasyntactic() (*in module salt.runners.test*), 3156
migrate() (*in module salt.modules.djangomod*), 1606
migrate() (*in module salt.modules.virt*), 2658
migrate() (*in module salt.modules.xapi_virt*), 2854
migrate() (*in module salt.runners.virt*), 3159
migrate_non_shared() (*in module salt.modules.virt*), 2658
migrate_non_shared_inc() (*in module salt.modules.virt*), 2658
min
    jinja filters, 432
min_party() (*in module salt.states.zk_concurrency*), 3806
min_query() (*salt.cloud.CloudClient method*), 3853
Mine, **5402**
Mine, **417**
mine() (*in module salt.runners.cache*), 3102
mine_enabled
    conf/minion, 135
mine_interval
    conf/minion, 136
mine_return_job
    conf/minion, 135
Minion, **5402**
Minion ID, **5402**
Minion proc System, 234
minion_config() (*in module salt.config*), 3843
minion_data_cache
    conf/master, 58
minion_data_cache_events
    conf/master, 64
minion_id_caching

conf/minion, 131
minion_id_lowercase
conf/minion, 131
minion_id_remove_domain
conf/minion, 131
minion_mods() (*in module salt.loader*), 3844
minion_pillar_cache
conf/minion, 156
MinionError, 3999
minionfs_blacklist
conf/master, 97
minionfs_env
conf/master, 96
minionfs_mountpoint
conf/master, 97
minionfs_update_interval
conf/master, 97
minionfs_whitelist
conf/master, 97
Minions (*class in salt.netapi.rest_cherrypy.app*), 2947
MinionSaltAPIHandler (*class in salt.netapi.rest_tornado.saltnado*), 2970
missing() (*in module salt.modules.daemontools*), 1585
missing() (*in module salt.modules.debian_service*), 1598
missing() (*in module salt.modules.freebsdservice*), 1748
missing() (*in module salt.modules.gentoo_service*), 1755
missing() (*in module salt.modules.launchctl_service*), 1952
missing() (*in module salt.modules.mac_service*), 2040
missing() (*in module salt.modules.netbsdservice*), 2204
missing() (*in module salt.modules.openbsdrcctl_service*), 2273
missing() (*in module salt.modules.openbsdservice*), 2275
missing() (*in module salt.modules.rh_service*), 2444
missing() (*in module salt.modules.runit*), 2454
missing() (*in module salt.modules.s6*), 2461
missing() (*in module salt.modules.smf_service*), 2511
missing() (*in module salt.modules.systemd_service*), 2595
missing() (*in module salt.modules.upstart_service*), 2628
missing() (*in module salt.modules.win_service*), 2798
missing() (*in module salt.states.file*), 3438
missing_func() (*in module salt.modules.test*), 2602
MissingSmb, 3999
mk_token() (*in module salt.runners.auth*), 3096
mk_token() (*in module salt.tokens.localfs*), 3828
mk_token() (*in module salt.tokens.rediscluster*), 3829
mkconfig() (*in module salt.modules.seed*), 2486
mkdir() (*in module salt.modules.file*), 1721
mkdir() (*in module salt.modules.win_file*), 2741
mkfs() (*in module salt.modules.btrfs*), 1467
mkfs() (*in module salt.modules.extfs*), 1701
mkfs() (*in module salt.modules.parted_partition*), 2326
mkfs() (*in module salt.modules.xfs*), 2861
mklabel() (*in module salt.modules.parted_partition*), 2326
mknod() (*in module salt.modules.file*), 1721
mknod() (*in module salt.states.file*), 3438
mknod_blkdev() (*in module salt.modules.file*), 1721
mknod_chrdev() (*in module salt.modules.file*), 1722
mknod_fifo() (*in module salt.modules.file*), 1722
mkpart() (*in module salt.modules.parted_partition*), 2326
mkpartfs() (*in module salt.modules.parted_partition*), 2326
mksls() (*in module salt.modules.genesis*), 1753
mmodule() (*in module salt.modules.saltutil*), 2471
mod_aggregate() (*in module salt.states.iptables*), 3506
mod_aggregate() (*in module salt.states.pkg*), 3654
mod_bufsize() (*in module salt.modules.network*), 2219
mod_hostname() (*in module salt.modules.network*), 2219
mod_list() (*in module salt.modules.freebsdkmod*), 1741
mod_list() (*in module salt.modules.kmod*), 1945
mod_repo() (*in module salt.modules.aptpkg*), 1222
mod_repo() (*in module salt.modules.opkg*), 2283
mod_repo() (*in module salt.modules.yumpkg*), 2872
mod_repo() (*in module salt.modules.zypperpkg*), 2930
mod_run_check() (*in module salt.states.cmd*), 3348
mod_run_check() (*in module salt.states.docker_container*), 3370
mod_run_check() (*in module salt.states.git*), 3462
mod_run_check_cmd() (*in module salt.states.file*), 3439
mod_watch() (*in module salt.states.at*), 3196
mod_watch() (*in module salt.states.cmd*), 3348
mod_watch() (*in module salt.states.docker_container*), 3370
mod_watch() (*in module salt.states.docker_image*), 3391
mod_watch() (*in module salt.states.etcd_mod*), 3406
mod_watch() (*in module salt.states.kernelpkg*), 3517
mod_watch() (*in module salt.states.mount*), 3563
mod_watch() (*in module salt.states.pkg*), 3654
mod_watch() (*in module salt.states.service*), 3711
mod_watch() (*in module salt.states.supervisord*), 3729
mod_watch() (*in module salt.states.test*), 3735
mod_watch() (*in module salt.states.tomcat*), 3737
MODE (*salt.modules.inspectlib.collector.Inspector attribute*), 1874

mode() (*in module salt.states.quota*), 3681
mode() (*in module salt.states.selinux*), 3708
mode() (*in module salt.thorium.calc*), 3820
model() (*in module salt.grains.napalm*), 1183
modfacl() (*in module salt.modules.linux_acl*), 1975
modified() (*in module salt.modules.rpm_lowpkg*), 2448
modified() (*in module salt.modules.yumpkg*), 2872
modified() (*in module salt.modules.zypperpkg*), 2930
modify() (*in module salt.modules.beacons*), 1285
modify() (*in module salt.modules.ldap3*), 1957
modify() (*in module salt.modules.pdbedit*), 2332
modify() (*in module salt.modules.schedule*), 2482
modify() (*in module salt.modules.sqlite3*), 2538
modify() (*in module salt.modules.win_service*), 2798
modify() (*in module salt.modules.xfs*), 2861
modify_binding() (*in module salt.modules.win_iis*), 2759
modify_cache_cluster() (*in module salt.modules.boto3_elasticache*), 1306
modify_cache_subnet_group() (*in module salt.modules.boto3_elasticache*), 1307
modify_db_instance() (*in module salt.modules.boto_rds*), 1428
modify_monitor() (*in module salt.modules.bigip*), 1293
modify_monitor() (*in module salt.states.bigip*), 3229
modify_network_interface_attribute() (*in module salt.modules.boto_ec2*), 1371
modify_node() (*in module salt.modules.bigip*), 1294
modify_node() (*in module salt.states.bigip*), 3229
modify_pool() (*in module salt.modules.bigip*), 1294
modify_pool() (*in module salt.states.bigip*), 3230
modify_pool_member() (*in module salt.modules.bigip*), 1296
modify_pool_member() (*in module salt.states.bigip*), 3231
modify_profile() (*in module salt.modules.bigip*), 1296
modify_profile() (*in module salt.states.bigip*), 3231
modify_replication_group() (*in module salt.modules.boto3_elasticache*), 1307
modify_site() (*in module salt.modules.win_iis*), 2760
modify_snapshot() (*in module salt.modules.snapper*), 2546
modify_virtual() (*in module salt.modules.bigip*), 1297
modify_virtual() (*in module salt.states.bigip*), 3231
module() (*in module salt.states.selinux*), 3708
module_dirs
    conf/master, 54
    conf/minion, 145
module_install() (*in module salt.states.selinux*), 3708
module_remove() (*in module salt.states.selinux*), 3708
module_report() (*in module salt.modules.test*), 2602
modules() (*in module salt.modules.apache*), 1210
modules() (*in module salt.modules.syslog_ng*), 2580
modules_max_memory
    conf/minion, 146
mon_active() (*in module salt.modules.ceph*), 1494
mon_create() (*in module salt.modules.ceph*), 1494
mon_is() (*in module salt.modules.ceph*), 1494
mon_quorum() (*in module salt.modules.ceph*), 1494
mon_status() (*in module salt.modules.ceph*), 1495
monitor() (*in module salt.modules.monit*), 2089
monitor() (*in module salt.states.monit*), 3562
monitored() (*in module salt.states.serverdensity_device*), 3709
monitored() (*in module salt.states.uptime*), 3742
monitored() (*in module salt.states.zenoss*), 3805
mount() (*in module salt.modules.guestfs*), 1839
mount() (*in module salt.modules.macpackage*), 2030
mount() (*in module salt.modules.mount*), 2092
mount() (*in module salt.modules.qemu_nbd*), 2411
mount() (*in module salt.modules.zfs*), 2902
mount_share() (*in module salt.modules.cimc*), 1510
mounted() (*in module salt.states.mount*), 3563
mounts() (*in module salt.modules.moosefs*), 2091
mounts() (*in module salt.modules.osquery*), 2292
move() (*in module salt.modules.file*), 1722
move() (*in module salt.modules.schedule*), 2482
move() (*in module salt.modules.zoneadm*), 2909
move_config() (*in module salt.states.panos*), 3625
msg (*salt.auth.pam.PamMessage attribute*), 1003
msg_style (*salt.auth.pam.PamMessage attribute*), 1003
mul() (*in module salt.thorium.calc*), 3820
Multi-Master, **5402**
multi_call() (*in module salt.modules.netmiko_mod*), 2209
multi_find() (*in module salt.runners.net*), 3135
multipath_flush() (*in module salt.modules.devmap*), 1601
multipath_list() (*in module salt.modules.devmap*), 1601
multiprocessing
    conf/minion, 160
mutex() (*in module salt.modules.sysbench*), 2575
MX() (*in module salt.modules.dig*), 1601
MX() (*in module salt.modules.dnsutil*), 1608
MySQLExtPillar (*class in salt.pillar.mysql*), 3019

N

name() (*in module salt.modules.parted_partition*), 2327
name() (*in module salt.modules.udev*), 2626
name_match() (*in module salt.wheel.key*), 3841

`NamedStatement` (*class in salt.modules.syslog_ng*), 2578
`namenode_format()` (*in module salt.modules.hadoop*), 1839
`nameservers()` (*in module salt.modules.drac*), 1663
`nameservers()` (*in module salt.modules.dracr*), 1668
`namespace_absent()` (*in module salt.states.kubernetes*), 3529
`namespace_present()` (*in module salt.states.kubernetes*), 3529
`namespaces()` (*in module salt.modules.kubernetesmod*), 1948
`nat_gateway_absent()` (*in module salt.states.boto_vpc*), 3331
`nat_gateway_exists()` (*in module salt.modules.boto_vpc*), 1459
`nat_gateway_present()` (*in module salt.states.boto_vpc*), 3331
`ne()` (*in module salt.thorium.check*), 3824
`needs_reboot()` (*in module salt.modules.kernelpkg_linux_apt*), 1925
`needs_reboot()` (*in module salt.modules.kernelpkg_linux_yum*), 1927
`needs_renewal()` (*in module salt.modules.acme*), 1202
`neighbors()` (*in module salt.modules.napalm_bgp*), 2150
`neighbors()` (*in module salt.runners.bgp*), 3098
`NestDisplay` (*class in salt.output.nested*), 2982
`NestDisplay` (*class in salt.output.no_return*), 2984
`NetapiClient` (*class in salt.netapi*), 3854
`netdev()` (*in module salt.modules.status*), 2560
`netmiko_args()` (*in module salt.modules.napalm_mod*), 2132
`netmiko_call()` (*in module salt.modules.napalm_mod*), 2132
`netmiko_commands()` (*in module salt.modules.napalm_mod*), 2133
`netmiko_config()` (*in module salt.modules.napalm_mod*), 2133
`netmiko_conn()` (*in module salt.modules.napalm_mod*), 2134
`netmiko_fun()` (*in module salt.modules.napalm_mod*), 2134
`netmiko_multi_call()` (*in module salt.modules.napalm_mod*), 2134
`netstat()` (*in module salt.modules.network*), 2219
`netstat()` (*in module salt.modules.ps*), 2391
`netstat()` (*in module salt.modules.win_network*), 2775
`netstats()` (*in module salt.modules.cassandra*), 1482
`netstats()` (*in module salt.modules.status*), 2560
`network()` (*in module salt.modules.ilo*), 1854
`network()` (*in module salt.states.drac*), 3400
`network_acl_exists()` (*in module salt.modules.boto_vpc*), 1460
`network_create()` (*in module salt.modules.cloud*), 1521
`network_create()` (*in module salt.modules.neutronng*), 2239
`network_define()` (*in module salt.modules.virt*), 2659
`network_delete()` (*in module salt.modules.neutronng*), 2239
`network_get()` (*in module salt.modules.neutronng*), 2240
`network_get_xml()` (*in module salt.modules.virt*), 2659
`network_hosts`
  `jinja filters`, 445
`network_info()` (*in module salt.modules.drac*), 1663
`network_info()` (*in module salt.modules.dracr*), 1668
`network_info()` (*in module salt.modules.virt*), 2660
`network_interface_absent()` (*in module salt.states.azurearm_network*), 3207
`network_interface_create_or_update()` (*in module salt.modules.azurearm_network*), 1257
`network_interface_delete()` (*in module salt.modules.azurearm_network*), 1258
`network_interface_get()` (*in module salt.modules.azurearm_network*), 1258
`network_interface_get_effective_route_table()` (*in module salt.modules.azurearm_network*), 1258
`network_interface_list_effective_network_security_g` (*in module salt.modules.azurearm_network*), 1259
`network_interface_present()` (*in module salt.states.azurearm_network*), 3207
`network_interfaces_list()` (*in module salt.modules.azurearm_network*), 1259
`network_interfaces_list_all()` (*in module salt.modules.azurearm_network*), 1259
`network_io_counters()` (*in module salt.modules.ps*), 2391
`network_list()` (*in module salt.modules.cloud*), 1521
`network_running()` (*in module salt.states.virt*), 3749
`network_security_group_absent()` (*in module salt.states.azurearm_network*), 3209
`network_security_group_create_or_update()` (*in module salt.modules.azurearm_network*), 1259
`network_security_group_delete()` (*in module salt.modules.azurearm_network*), 1260
`network_security_group_get()` (*in module salt.modules.azurearm_network*), 1260
`network_security_group_present()` (*in module salt.states.azurearm_network*), 3209
`network_security_groups_list()` (*in module*

salt.modules.azurearm_network), 1260

`network_security_groups_list_all()` (*in module salt.modules.azurearm_network*), 1260

`network_set_autostart()` (*in module salt.modules.virt*), 2660

`network_size`
- `jinja filters`, 445

`network_start()` (*in module salt.modules.virt*), 2660

`network_stop()` (*in module salt.modules.virt*), 2660

`network_undefine()` (*in module salt.modules.virt*), 2661

`networks()` (*in module salt.modules.dockermod*), 1645

`new()` (*salt.modules.inspectlib.fsdb.CsvDB method*), 1877

`new_chain()` (*in module salt.modules.iptables*), 1901

`new_chain()` (*in module salt.modules.nftables*), 2247

`new_service()` (*in module salt.modules.firewalld*), 1735

`new_set()` (*in module salt.modules.ipset*), 1897

`new_table()` (*in module salt.modules.nftables*), 2247

`new_zone()` (*in module salt.modules.firewalld*), 1736

`next_host()` (*in module salt.runners.virt*), 3159

`nfs_shares()` (*in module salt.modules.osquery*), 2292

`nick` (*salt.engines.ircbot.PrivEvent attribute*), 1156

`nics_skip()` (*in module salt.states.csf*), 3359

`nics_skipped()` (*in module salt.states.csf*), 3359

`no_proxy`
- `conf/minion`, 142

`Node Group`, **5402**

`node()` (*in module salt.modules.kubernetesmod*), 1948

`node_add_label()` (*in module salt.modules.kubernetesmod*), 1948

`node_info()` (*in module salt.modules.elasticsearch*), 1689

`node_info()` (*in module salt.modules.virt*), 2661

`node_info()` (*in module salt.modules.xapi_virt*), 2854

`node_label_absent()` (*in module salt.states.kubernetes*), 3529

`node_label_folder_absent()` (*in module salt.states.kubernetes*), 3529

`node_label_present()` (*in module salt.states.kubernetes*), 3529

`node_labels()` (*in module salt.modules.kubernetesmod*), 1949

`node_ls()` (*in module salt.modules.swarm*), 2572

`node_remove_label()` (*in module salt.modules.kubernetesmod*), 1949

`node_setup()` (*in module salt.modules.icinga2*), 1855

`node_setup()` (*in module salt.states.icinga2*), 3488

`nodegroups`
- `conf/master`, 116

`nodes()` (*in module salt.modules.kubernetesmod*), 1949

`nodes()` (*in module salt.runners.smartos_vmadm*), 3150

`non_structured_query()` (*in module salt.modules.servicenow*), 2496

`NonPagedPoolAllocs` (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATIO attribute*), 2809

`NonPagedPoolFrees` (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATIO attribute*), 2809

`NonPagedPoolLookasideHits` (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATIO attribute*), 2809

`NonPagedPoolPages` (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATIO attribute*), 2809

`noop()` (*in module salt.modules.puppet*), 2395

`nop()` (*in module salt.states.test*), 3735

`normalize_input_values()` (*in module salt.modules.lxd*), 2016

`normalize_name()` (*in module salt.modules.aptpkg*), 1222

`normalize_name()` (*in module salt.modules.solarisipspkg*), 2522

`normalize_name()` (*in module salt.modules.yumpkg*), 2873

`normalize_name()` (*in module salt.modules.zypperpkg*), 2931

`normpath()` (*in module salt.modules.file*), 1722

`noscan()` (*in module salt.modules.bluez_bluetooth*), 1300

`not_alived()` (*in module salt.runners.manage*), 3124

`not_allowed()` (*in module salt.runners.manage*), 3124

`not_cached()` (*in module salt.states.file*), 3439

`not_joined()` (*in module salt.runners.manage*), 3124

`not_loaded()` (*in module salt.modules.test*), 2602

`not_present()` (*in module salt.runners.manage*), 3124

`not_reaped()` (*in module salt.runners.manage*), 3124

`NotImplemented`, 3999

`nproc()` (*in module salt.modules.status*), 2560

`NS()` (*in module salt.modules.dig*), 1601

`NS()` (*in module salt.modules.dnsutil*), 1608

`nslookup()` (*in module salt.modules.win_network*), 2775

`ntp()` (*in module salt.states.cimc*), 3341

`ntp_configured()` (*in module salt.states.esxi*), 3412

`num_cpus()` (*in module salt.modules.ps*), 2391

`num_fields` (*salt.pillar.sql_base.SqlBaseExtPillar attribute*), 3035

`nvme_grains`
- `conf/minion`, 135

`nvram()` (*in module salt.modules.osquery*), 2292

`nxos_api_config()` (*in module salt.modules.napalm_mod*), 2134

`nxos_api_rpc()` (*in module salt.modules.napalm_mod*), 2135

`nxos_api_show()` (*in module salt.modules.napalm_mod*), 2135

O

obfuscate() (*in module salt.modules.pillar*), 2343
object_absent() (*in module salt.states.libcloud_storage*), 3538
object_present() (*in module salt.states.libcloud_storage*), 3538
off() (*in module salt.modules.quota*), 2412
off() (*in module salt.modules.tuned*), 2625
off() (*in module salt.states.tuned*), 3741
offline() (*in module salt.modules.trafficserver*), 2623
offline() (*in module salt.modules.zpool*), 2921
offline() (*in module salt.states.trafficserver*), 3740
offload() (*in module salt.states.ethtool*), 3408
on() (*in module salt.modules.quota*), 2412
on_closed() (*salt.engines.ircbot.IRCClient method*), 1156
on_connect() (*salt.engines.ircbot.IRCClient method*), 1156
on_demand_ext_pillar
    conf/master, 99
    conf/minion, 154
online() (*in module salt.modules.zpool*), 2921
only() (*in module salt.states.host*), 3486
op_version() (*in module salt.states.glusterfs*), 3467
open() (*salt.modules.inspectlib.dbhandle.DBHandleBase method*), 1875
open() (*salt.modules.inspectlib.fsdb.CsvDB method*), 1877
open_connection() (*salt.engines.logentries.PlainTextSocketAppender method*), 1159
open_connection() (*salt.engines.logentries.TLSSocketAppender method*), 1159
open_files() (*in module salt.modules.file*), 1722
open_mode
    conf/master, 68
    conf/minion, 157
openconfig_interfaces() (*in module salt.modules.netbox*), 2201
openconfig_lacp() (*in module salt.modules.netbox*), 2201
OPT_TIMEOUT_KEY (*salt.modules.ansiblegate.AnsibleModuleCaller attribute*), 1208
optics() (*in module salt.modules.napalm_network*), 2172
optimization_order
    conf/master, 73
    conf/minion, 143
optimize() (*in module salt.modules.solr*), 2531
optimize_providers() (*in module salt.cloud.clouds.ec2*), 1050
Option (*class in salt.modules.syslog_ng*), 2578
option() (*in module salt.modules.config*), 1559
option_group_exists() (*in module salt.modules.boto_rds*), 1428
option_present() (*in module salt.states.csf*), 3359
optional_args() (*in module salt.grains.napalm*), 1183
options() (*in module salt.modules.supervisord*), 2566
options_absent() (*in module salt.states.ini_manage*), 3497
options_present() (*in module salt.states.ini_manage*), 3497
opts() (*in module salt.grains.opts*), 1185
opts_pkg() (*in module salt.modules.test*), 2602
orchestrate() (*in module salt.modules.state*), 2552
orchestrate() (*in module salt.runners.state*), 3153
orchestrate_high() (*in module salt.runners.state*), 3154
orchestrate_show_sls() (*in module salt.runners.state*), 3154
orchestrate_single() (*in module salt.runners.state*), 3154
order_certificate() (*in module salt.runners.digicertapi*), 3106
order_masters
    conf/master, 111
os() (*in module salt.grains.chronos*), 1177
os() (*in module salt.grains.esxi*), 1179
os() (*in module salt.grains.marathon*), 1180
os() (*in module salt.grains.philips_hue*), 1185
os() (*in module salt.grains.rest_sample*), 1185
os_data() (*in module salt.grains.chronos*), 1177
os_data() (*in module salt.grains.core*), 1178
os_data() (*in module salt.grains.fx2*), 1180
os_data() (*in module salt.grains.marathon*), 1180
os_data() (*in module salt.grains.rest_sample*), 1185
os_family() (*in module salt.grains.chronos*), 1177
os_family() (*in module salt.grains.esxi*), 1179
os_family() (*in module salt.grains.fx2*), 1180
os_family() (*in module salt.grains.junos*), 1180
os_family() (*in module salt.grains.marathon*), 1180
os_family() (*in module salt.grains.philips_hue*), 1185
os_family() (*in module salt.grains.rest_sample*), 1185
os_version() (*in module salt.modules.osquery*), 2292
osd_activate() (*in module salt.modules.ceph*), 1495
osd_discover() (*in module salt.modules.ceph*), 1495
osd_prepare() (*in module salt.modules.ceph*), 1495
osquery_extensions() (*in module salt.modules.osquery*), 2292
osquery_flags() (*in module salt.modules.osquery*), 2293
osquery_info() (*in module salt.modules.osquery*), 2293
osquery_registry() (*in module salt.modules.osquery*), 2293
out_format() (*in module salt.modules.out*), 2296

`output`
    `conf/master`, 55
`output()` (*in module salt.output.highstate*), 2981
`output()` (*in module salt.output.json_out*), 2982
`output()` (*in module salt.output.key*), 2982
`output()` (*in module salt.output.nested*), 2983
`output()` (*in module salt.output.newline_values_only*), 2984
`output()` (*in module salt.output.no_out_quiet*), 2984
`output()` (*in module salt.output.no_return*), 2984
`output()` (*in module salt.output.overstatestage*), 2985
`output()` (*in module salt.output.pony*), 2985
`output()` (*in module salt.output.pprint_out*), 2986
`output()` (*in module salt.output.profile*), 2986
`output()` (*in module salt.output.progress*), 2986
`output()` (*in module salt.output.raw*), 2987
`output()` (*in module salt.output.table_out*), 2988
`output()` (*in module salt.output.txt*), 2989
`output()` (*in module salt.output.virt_query*), 2989
`output()` (*in module salt.output.yaml_out*), 2989
`output_file`
    `conf/master`, 56
`Outputter`, **5402**
`outputter()` (*in module salt.modules.test*), 2602
`outputter_dirs`
    `conf/master`, 56
    `conf/minion`, 136
`overview()` (*in module salt.modules.drbd*), 1673
`overwrite_api_stage_variables()` (*in module salt.modules.boto_apigateway*), 1340
`owner()` (*in module salt.modules.apkpkg*), 1214
`owner()` (*in module salt.modules.aptpkg*), 1223
`owner()` (*in module salt.modules.opkg*), 2283
`owner()` (*in module salt.modules.pacmanpkg*), 2300
`owner()` (*in module salt.modules.rpm_lowpkg*), 2448
`owner()` (*in module salt.modules.yumpkg*), 2873
`owner()` (*in module salt.modules.zypperpkg*), 2931
`owner_to()` (*in module salt.modules.postgres*), 2379

P

`pack()` (*in module salt.modules.genesis*), 1754
`pack()` (*salt.exceptions.SaltException method*), 4000
`pack_sources()` (*in module salt.modules.pkg_resource*), 2350
`Package` (*class in salt.modules.inspectlib.entities*), 1876
`package_info()` (*in module salt.modules.win_dism*), 2727
`package_install()` (*in module salt.proxy.dummy*), 3057
`package_install()` (*in module salt.proxy.rest_sample*), 3085
`package_install()` (*in module salt.proxy.ssh_sample*), 3086
`package_installed()` (*in module salt.states.win_dism*), 3760
`package_list()` (*in module salt.proxy.dummy*), 3057
`package_list()` (*in module salt.proxy.rest_sample*), 3085
`package_list()` (*in module salt.proxy.ssh_sample*), 3086
`package_remove()` (*in module salt.proxy.dummy*), 3057
`package_remove()` (*in module salt.proxy.rest_sample*), 3085
`package_remove()` (*in module salt.proxy.ssh_sample*), 3086
`package_removed()` (*in module salt.states.win_dism*), 3760
`package_status()` (*in module salt.proxy.dummy*), 3057
`package_status()` (*in module salt.proxy.rest_sample*), 3085
`PackageCfgFile` (*class in salt.modules.inspectlib.entities*), 1876
`PagedPoolAllocs` (*salt.modules.win_status.SYSTEM_PERFORMANCE_I attribute*), 2809
`PagedPoolFrees` (*salt.modules.win_status.SYSTEM_PERFORMANCE_IN attribute*), 2809
`PagedPoolLookasideHits` (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATIO attribute*), 2809
`PagedPoolPages` (*salt.modules.win_status.SYSTEM_PERFORMANCE_IN attribute*), 2809
`PageFaultCount` (*salt.modules.win_status.SYSTEM_PERFORMANCE_IN attribute*), 2809
`PageReadCount` (*salt.modules.win_status.SYSTEM_PERFORMANCE_INF attribute*), 2809
`PageReadIoCount` (*salt.modules.win_status.SYSTEM_PERFORMANCE_I attribute*), 2809
`pair()` (*in module salt.modules.bluez_bluetooth*), 1300
`PamConv` (*class in salt.auth.pam*), 1002
`PamHandle` (*class in salt.auth.pam*), 1003
`PamMessage` (*class in salt.auth.pam*), 1003
`PamResponse` (*class in salt.auth.pam*), 1003
`panos()` (*in module salt.grains.panos*), 1185
`parallel_runners()` (*in module salt.states.saltmod*), 3698
`param_set()` (*in module salt.modules.varnish*), 2638
`param_show()` (*in module salt.modules.varnish*), 2638
`Parameter` (*class in salt.modules.syslog_ng*), 2578
`parameter_group_exists()` (*in module salt.modules.boto_rds*), 1428
`parameter_present()` (*in module salt.states.boto_rds*), 3313
`ParameterValue` (*class in salt.modules.syslog_ng*), 2578
`pardir()` (*in module salt.modules.file*), 1722

`parse()` (*in module salt.modules.napalm_yang_mod*), 2192
`parse()` (*in module salt.proxy.ssh_sample*), 3086
`parse_arch()` (*in module salt.modules.aptpkg*), 1223
`parse_arch()` (*in module salt.modules.yumpkg*), 2873
`parse_arch()` (*in module salt.modules.zypperpkg*), 2931
`parse_args_and_kwargs()` (*salt.engines.slack.SlackClient method*), 1168
`parse_config()` (*in module salt.modules.pkgng*), 2361
`parse_config()` (*in module salt.modules.poudriere*), 2386
`parse_csr()` (*in module salt.modules.namecheap_ssl*), 2121
`parse_hosts()` (*in module salt.modules.dnsutil*), 1609
`parse_ssh_config()` (*in module salt.roster.sshconfig*), 3093
`parse_targets()` (*in module salt.modules.pkg_resource*), 2350
`parse_zone()` (*in module salt.modules.dnsutil*), 1609
`partition_is()` (*in module salt.modules.ceph*), 1496
`partition_list()` (*in module salt.modules.ceph*), 1496
`partition_list_journal()` (*in module salt.modules.ceph*), 1496
`partition_list_osd()` (*in module salt.modules.ceph*), 1496
`party_members()` (*in module salt.modules.zk_concurrency*), 2906
`pass_to_ext_pillars`
  `conf/minion`, 156
`passwd()` (*in module salt.modules.tomcat*), 2620
`passwd_changes()` (*in module salt.modules.osquery*), 2293
`password_present()` (*in module salt.states.esxi*), 3412
`patch()` (*in module salt.modules.file*), 1723
`patch()` (*in module salt.modules.napalm_network*), 2173
`patch()` (*in module salt.states.file*), 3439
`patch_downloaded()` (*in module salt.states.pkg*), 3655
`patch_installed()` (*in module salt.states.pkg*), 3655
`path()` (*in module salt.grains.core*), 1178
`path()` (*in module salt.modules.udev*), 2626
`path_exists_glob()` (*in module salt.modules.file*), 1723
`path_join`
  `jinja filters`, 448
`pause()` (*in module salt.cloud.clouds.xen*), 1151
`pause()` (*in module salt.modules.dockercompose*), 1612
`pause()` (*in module salt.modules.dockermod*), 1645
`pause()` (*in module salt.modules.state*), 2553
`pause()` (*in module salt.modules.vagrant*), 2636
`pause()` (*in module salt.modules.virt*), 2661
`pause()` (*in module salt.modules.xapi_virt*), 2854
`pause()` (*in module salt.runners.state*), 3154
`pause()` (*in module salt.runners.virt*), 3159
`paused()` (*in module salt.states.vagrant*), 3745
`PayloadFile` (*class in salt.modules.inspectlib.entities*), 1876
`pci_devices()` (*in module salt.modules.osquery*), 2293
`pcre()` (*in module salt.modules.match*), 2072
`PeakCommitment` (*salt.modules.win_status.SYSTEM_PERFORMANCE_IN attribute*), 2809
`peer`
  `conf/master`, 113
`Peer Communication`, **5402**
`peer()` (*in module salt.modules.glusterfs*), 1813
`peer_run`
  `conf/master`, 114
`peer_status()` (*in module salt.modules.glusterfs*), 1814
`peered()` (*in module salt.states.glusterfs*), 3467
`peering_connection_pending_from_vpc()` (*in module salt.modules.boto_vpc*), 1460
`peers()` (*in module salt.modules.napalm_ntp*), 2176
`pem_managed()` (*in module salt.states.x509*), 3792
`percent()` (*in module salt.modules.disk*), 1604
`permissive_pki_access`
  `conf/master`, 69
  `conf/minion`, 157
`persist()` (*in module salt.modules.freebsd_sysctl*), 1738
`persist()` (*in module salt.modules.linux_sysctl*), 1979
`persist()` (*in module salt.modules.mac_sysctl*), 2048
`persist()` (*in module salt.modules.netbsd_sysctl*), 2202
`persist()` (*in module salt.modules.openbsd_sysctl*), 2270
`pg_create()` (*in module salt.modules.purefa*), 2398
`pg_delete()` (*in module salt.modules.purefa*), 2398
`pg_eradicate()` (*in module salt.modules.purefa*), 2399
`pg_remove()` (*in module salt.modules.purefa*), 2399
`pg_update()` (*in module salt.modules.purefa*), 2399
`pgrep()` (*in module salt.modules.ps*), 2391
`pid()` (*in module salt.modules.dockermod*), 1646
`pid()` (*in module salt.modules.nspawn*), 2261
`pid()` (*in module salt.modules.status*), 2560
`PID_FILE` (*salt.modules.inspectlib.EnvLoader attribute*), 1876
`pidfile`
  `conf/master`, 52
`pif_list()` (*in module salt.cloud.clouds.xen*), 1151
`Pillar`, **5402**
`pillar()` (*in module salt.modules.match*), 2072

`pillar()` (*in module salt.runners.cache*), 3102
`pillar_cache`
    `conf/master`, 109
`pillar_cache_backend`
    `conf/master`, 109
`pillar_cache_ttl`
    `conf/master`, 109
`pillar_dir()` (*salt.pillar.svn_pillar.SvnPillar method*), 3044
`pillar_format()` (*in module salt.pillar.consul_pillar*), 2994
`pillar_includes_override_sls`
    `conf/master`, 109
`pillar_merge_lists`
    `conf/master`, 108
`pillar_opts`
    `conf/master`, 100
`pillar_pcre()` (*in module salt.modules.match*), 2073
`pillar_raise_on_missing`
    `conf/master`, 102
    `conf/minion`, 156
`pillar_roots`
    `conf/master`, 98
    `conf/minion`, 154
`pillar_safe_render_error`
    `conf/master`, 100
`pillar_source_merging_strategy`
    `conf/master`, 107
`pillarenv`
    `conf/minion`, 155
`pillarenv_from_saltenv`
    `conf/minion`, 101, 155
`ping()` (*in module salt.modules.elasticsearch*), 1689
`ping()` (*in module salt.modules.gnomedesktop*), 1816
`ping()` (*in module salt.modules.junos*), 1917
`ping()` (*in module salt.modules.kubernetesmod*), 1949
`ping()` (*in module salt.modules.napalm_network*), 2173
`ping()` (*in module salt.modules.network*), 2220
`ping()` (*in module salt.modules.redismod*), 2431
`ping()` (*in module salt.modules.solr*), 2531
`ping()` (*in module salt.modules.sysbench*), 2575
`ping()` (*in module salt.modules.test*), 2602
`ping()` (*in module salt.modules.test_virtual*), 2605
`ping()` (*in module salt.modules.win_network*), 2775
`ping()` (*in module salt.proxy.arista_pyeapi*), 3050
`ping()` (*in module salt.proxy.chronos*), 3052
`ping()` (*in module salt.proxy.cimc*), 3052
`ping()` (*in module salt.proxy.cisconso*), 3056
`ping()` (*in module salt.proxy.dummy*), 3057
`ping()` (*in module salt.proxy.esxi*), 3061
`ping()` (*in module salt.proxy.fx2*), 3064
`ping()` (*in module salt.proxy.junos*), 3065
`ping()` (*in module salt.proxy.marathon*), 3066
`ping()` (*in module salt.proxy.napalm*), 3069
`ping()` (*in module salt.proxy.netmiko_px*), 3073
`ping()` (*in module salt.proxy.nxos*), 3075
`ping()` (*in module salt.proxy.nxos_api*), 3078
`ping()` (*in module salt.proxy.panos*), 3081
`ping()` (*in module salt.proxy.philips_hue*), 3085
`ping()` (*in module salt.proxy.rest_sample*), 3085
`ping()` (*in module salt.proxy.ssh_sample*), 3086
`ping_interval`
    `conf/minion`, 138
`ping_master()` (*in module salt.modules.status*), 2561
`ping_on_rotate`
    `conf/master`, 61
`pip_has_exceptions_mod()` (*in module salt.states.pip_state*), 3642
`pip_has_internal_exceptions_mod()` (*in module salt.states.pip_state*), 3642
`pipeline_absent()` (*in module salt.states.elasticsearch*), 3402
`pipeline_create()` (*in module salt.modules.elasticsearch*), 1689
`pipeline_delete()` (*in module salt.modules.elasticsearch*), 1689
`pipeline_get()` (*in module salt.modules.elasticsearch*), 1689
`pipeline_id_from_name()` (*in module salt.modules.boto_datapipeline*), 1359
`pipeline_present()` (*in module salt.states.elasticsearch*), 3402
`pipeline_simulate()` (*in module salt.modules.elasticsearch*), 1690
`pkg()` (*in module salt.modules.state*), 2553
`PkgParseError`, 3999
`pki_dir`
    `conf/master`, 53
    `conf/minion`, 130
`pkill()` (*in module salt.modules.ps*), 2392
`PlainTextSocketAppender` (*class in salt.engines.logentries*), 1159
`playbooks()` (*in module salt.modules.ansiblegate*), 1208
`playbooks()` (*in module salt.states.ansiblegate*), 3185
`plugin_add()` (*in module salt.modules.mysql*), 2103
`plugin_installed()` (*in module salt.modules.jenkinsmod*), 1910
`plugin_is_enabled()` (*in module salt.modules.rabbitmq*), 2415
`plugin_remove()` (*in module salt.modules.mysql*), 2103
`plugin_status()` (*in module salt.modules.mysql*), 2103
`plugin_sync()` (*in module salt.modules.puppet*), 2395
`plugins_list()` (*in module salt.modules.mysql*), 2103
`pod_absent()` (*in module salt.states.kubernetes*), 3529

pod_present() (*in module salt.states.kubernetes*), 3529
pods() (*in module salt.modules.kubernetesmod*), 1949
policy_absent() (*in module salt.states.boto_iam*), 3296
policy_absent() (*in module salt.states.boto_iot*), 3301
policy_assignment_absent() (*in module salt.states.azurearm_resource*), 3217
policy_assignment_create() (*in module salt.modules.azurearm_resource*), 1275
policy_assignment_delete() (*in module salt.modules.azurearm_resource*), 1275
policy_assignment_get() (*in module salt.modules.azurearm_resource*), 1276
policy_assignment_present() (*in module salt.states.azurearm_resource*), 3218
policy_assignments_list() (*in module salt.modules.azurearm_resource*), 1276
policy_assignments_list_for_resource_group() (*in module salt.modules.azurearm_resource*), 1276
policy_attached() (*in module salt.states.boto_iot*), 3301
policy_definition_absent() (*in module salt.states.azurearm_resource*), 3218
policy_definition_create_or_update() (*in module salt.modules.azurearm_resource*), 1276
policy_definition_delete() (*in module salt.modules.azurearm_resource*), 1276
policy_definition_get() (*in module salt.modules.azurearm_resource*), 1277
policy_definition_present() (*in module salt.states.azurearm_resource*), 3218
policy_definitions_list() (*in module salt.modules.azurearm_resource*), 1277
policy_detached() (*in module salt.states.boto_iot*), 3302
policy_exists() (*in module salt.modules.boto_iam*), 1404
policy_exists() (*in module salt.modules.boto_iot*), 1411
policy_exists() (*in module salt.modules.rabbitmq*), 2416
policy_present() (*in module salt.states.boto_iam*), 3296
policy_present() (*in module salt.states.boto_iot*), 3302
policy_present() (*in module salt.states.vault*), 3747
policy_version_exists() (*in module salt.modules.boto_iam*), 1404
policy_version_exists() (*in module salt.modules.boto_iot*), 1411
pool_absent() (*in module salt.states.boto_cognitoidentity*), 3268
pool_add() (*in module salt.modules.ceph*), 1496
pool_build() (*in module salt.modules.virt*), 2661
pool_capabilities() (*in module salt.modules.virt*), 2662
pool_define() (*in module salt.modules.virt*), 2662
pool_del() (*in module salt.modules.ceph*), 1496
pool_delete() (*in module salt.modules.virt*), 2664
pool_deleted() (*in module salt.states.virt*), 3750
pool_get_xml() (*in module salt.modules.virt*), 2664
pool_info() (*in module salt.modules.virt*), 2664
pool_list() (*in module salt.cloud.clouds.xen*), 1151
pool_list() (*in module salt.modules.ceph*), 1497
pool_list_volumes() (*in module salt.modules.virt*), 2664
pool_present() (*in module salt.states.boto_cognitoidentity*), 3268
pool_refresh() (*in module salt.modules.virt*), 2665
pool_running() (*in module salt.states.virt*), 3750
pool_set_autostart() (*in module salt.modules.virt*), 2665
pool_start() (*in module salt.modules.virt*), 2665
pool_stop() (*in module salt.modules.virt*), 2665
pool_undefine() (*in module salt.modules.virt*), 2666
pool_update() (*in module salt.modules.virt*), 2666
pop() (*in module salt.modules.data*), 1587
pop() (*in module salt.queues.pgjsonb_queue*), 3088
pop() (*in module salt.queues.sqlite_queue*), 3088
pop() (*in module salt.runners.queue*), 3141
port() (*in module salt.modules.dockermod*), 1646
port_add() (*in module salt.modules.openvswitch*), 2279
port_add_policy() (*in module salt.modules.selinux*), 2490
port_create_gre() (*in module salt.modules.openvswitch*), 2279
port_create_vlan() (*in module salt.modules.openvswitch*), 2279
port_create_vxlan() (*in module salt.modules.openvswitch*), 2279
port_delete_policy() (*in module salt.modules.selinux*), 2490
port_get_policy() (*in module salt.modules.selinux*), 2490
port_get_tag() (*in module salt.modules.openvswitch*), 2280
port_list() (*in module salt.modules.openvswitch*), 2280
port_policy_absent() (*in module salt.states.selinux*), 3708
port_policy_present() (*in module salt.states.selinux*), 3708
port_remove() (*in module salt.modules.openvswitch*), 2280

`ports_open()` (*in module salt.states.csf*), 3359
`porttree_matches()` (*in module salt.modules.ebuildpkg*), 1678
`POST` (*salt.netapi.rest_cherrypy.app.LowDataAdapter attribute*), 2945
`POST()` (*salt.netapi.rest_cherrypy.app.Keys method*), 2957
`POST()` (*salt.netapi.rest_cherrypy.app.Login method*), 2945
`POST()` (*salt.netapi.rest_cherrypy.app.Logout method*), 2947
`POST()` (*salt.netapi.rest_cherrypy.app.Minions method*), 2947
`POST()` (*salt.netapi.rest_cherrypy.app.Run method*), 2950
`POST()` (*salt.netapi.rest_cherrypy.app.Webhook method*), 2954
`post()` (*salt.netapi.rest_tornado.saltnado.MinionSaltAPIHandler method*), 2971
`post()` (*salt.netapi.rest_tornado.saltnado.RunSaltAPIHandler method*), 2973
`post()` (*salt.netapi.rest_tornado.saltnado.SaltAPIHandler method*), 2967
`post()` (*salt.netapi.rest_tornado.saltnado.SaltAuthHandler method*), 2969
`post()` (*salt.netapi.rest_tornado.saltnado.WebhookSaltAPIHandler method*), 2976
`post_card()` (*in module salt.modules.msteams*), 2098
`post_card()` (*in module salt.states.msteams*), 3565
`post_data()` (*in module salt.modules.opsgenie*), 2286
`post_dns_record()` (*in module salt.cloud.clouds.digitalocean*), 1041
`post_event()` (*in module salt.modules.datadog_api*), 1588
`post_event()` (*in module salt.runners.mattermost*), 3126
`post_message()` (*in module salt.modules.mattermost*), 2073
`post_message()` (*in module salt.modules.pushover_notify*), 2405
`post_message()` (*in module salt.modules.slack_notify*), 2497
`post_message()` (*in module salt.modules.telegram*), 2598
`post_message()` (*in module salt.returners.mattermost_returner*), 326
`post_message()` (*in module salt.runners.mattermost*), 3126
`post_message()` (*in module salt.states.pushover*), 3679
`post_message()` (*in module salt.states.slack*), 3713
`POSTGRESExtPillar` (*class in salt.pillar.postgres*), 3028
`postpone_job()` (*in module salt.modules.schedule*), 2482
`power()` (*in module salt.modules.bluez_bluetooth*), 1300
`power()` (*in module salt.states.ipmi*), 3499
`power_configuration()` (*in module salt.states.cimc*), 3341
`power_off_vm()` (*in module salt.modules.vsphere*), 2707
`power_on_vm()` (*in module salt.modules.vsphere*), 2707
`powered_off()` (*in module salt.states.vagrant*), 3745
`powered_off()` (*in module salt.states.virt*), 3751
`poweroff()` (*in module salt.modules.nspawn*), 2261
`poweroff()` (*in module salt.modules.solaris_system*), 2517
`poweroff()` (*in module salt.modules.system*), 2590
`poweroff()` (*in module salt.modules.win_system*), 2816
`poweroff()` (*in module salt.runners.drac*), 3107
`poweron()` (*in module salt.runners.drac*), 3108
`powershell()` (*in module salt.modules.cmdmod*), 1523
`powershell_all()` (*in module salt.modules.cmdmod*), 1526
`preferences()` (*in module salt.modules.osquery*), 2293
`preferred_ip()` (*in module salt.cloud.clouds.dimensiondata*), 1044
`preferred_ip()` (*in module salt.cloud.clouds.openstack*), 1113
`prep_bootstrap()` (*in module salt.modules.seed*), 2487
`prep_jid()` (*in module salt.returners.carbon_return*), 313
`prep_jid()` (*in module salt.returners.cassandra_cql_return*), 315
`prep_jid()` (*in module salt.returners.cassandra_return*), 316
`prep_jid()` (*in module salt.returners.couchbase_return*), 317
`prep_jid()` (*in module salt.returners.couchdb_return*), 318
`prep_jid()` (*in module salt.returners.django_return*), 318
`prep_jid()` (*in module salt.returners.elasticsearch_return*), 320
`prep_jid()` (*in module salt.returners.etcd_return*), 321
`prep_jid()` (*in module salt.returners.influxdb_return*), 323
`prep_jid()` (*in module salt.returners.local_cache*), 325
`prep_jid()` (*in module salt.returners.memcache_return*), 327
`prep_jid()` (*in module salt.returners.mongo_future_return*), 329
`prep_jid()` (*in module salt.returners.mongo_return*), 330
`prep_jid()` (*in module salt.returners.multi_returner*), 330

`prep_jid()` (*in module salt.returners.mysql*), 333
`prep_jid()` (*in module salt.returners.odbc*), 336
`prep_jid()` (*in module salt.returners.pgjsonb*), 340
`prep_jid()` (*in module salt.returners.postgres*), 342
`prep_jid()` (*in module salt.returners.postgres_local_cache*), 344
`prep_jid()` (*in module salt.returners.redis_return*), 347
`prep_jid()` (*in module salt.returners.sentry_return*), 348
`prep_jid()` (*in module salt.returners.smtp_return*), 352
`prep_jid()` (*in module salt.returners.sqlite3_return*), 354
`prep_jid()` (*in module salt.returners.syslog_return*), 355
`prepare_return()` (*in module salt.proxy.cimc*), 3052
`prepare_rows()` (*salt.output.table_out.TableDisplay method*), 2988
`prepend()` (*in module salt.modules.file*), 1723
`prepend()` (*in module salt.states.file*), 3440
`presence_events`
  `conf/master`, 61
`present()` (*in module salt.runners.manage*), 3125
`present()` (*in module salt.states.alias*), 3183
`present()` (*in module salt.states.at*), 3196
`present()` (*in module salt.states.beacon*), 3222
`present()` (*in module salt.states.boto3_elasticsearch*), 3243
`present()` (*in module salt.states.boto_apigateway*), 3251
`present()` (*in module salt.states.boto_asg*), 3258
`present()` (*in module salt.states.boto_cfn*), 3261
`present()` (*in module salt.states.boto_cloudfront*), 3263
`present()` (*in module salt.states.boto_cloudtrail*), 3264
`present()` (*in module salt.states.boto_cloudwatch_alarm*), 3266
`present()` (*in module salt.states.boto_cloudwatch_event*), 3267
`present()` (*in module salt.states.boto_datapipeline*), 3269
`present()` (*in module salt.states.boto_dynamodb*), 3273
`present()` (*in module salt.states.boto_elasticache*), 3281
`present()` (*in module salt.states.boto_elasticsearch_domain*), 3284
`present()` (*in module salt.states.boto_elb*), 3289
`present()` (*in module salt.states.boto_iam_role*), 3299
`present()` (*in module salt.states.boto_kinesis*), 3304
`present()` (*in module salt.states.boto_lc*), 3311
`present()` (*in module salt.states.boto_rds*), 3313
`present()` (*in module salt.states.boto_route53*), 3318
`present()` (*in module salt.states.boto_s3_bucket*), 3321
`present()` (*in module salt.states.boto_secgroup*), 3323
`present()` (*in module salt.states.boto_sns*), 3325
`present()` (*in module salt.states.boto_sqs*), 3326
`present()` (*in module salt.states.boto_vpc*), 3331
`present()` (*in module salt.states.cloud*), 3343
`present()` (*in module salt.states.cron*), 3358
`present()` (*in module salt.states.ddns*), 3362
`present()` (*in module salt.states.docker_image*), 3391
`present()` (*in module salt.states.docker_network*), 3394
`present()` (*in module salt.states.docker_volume*), 3399
`present()` (*in module salt.states.drac*), 3400
`present()` (*in module salt.states.elasticsearch_index*), 3403
`present()` (*in module salt.states.elasticsearch_index_template*), 3403
`present()` (*in module salt.states.firewalld*), 3453
`present()` (*in module salt.states.git*), 3462
`present()` (*in module salt.states.github*), 3463
`present()` (*in module salt.states.glance_image*), 3466
`present()` (*in module salt.states.gpg*), 3470
`present()` (*in module salt.states.grafana4_dashboard*), 3474
`present()` (*in module salt.states.grafana4_datasource*), 3475
`present()` (*in module salt.states.grafana4_org*), 3477
`present()` (*in module salt.states.grafana4_user*), 3478
`present()` (*in module salt.states.grafana_dashboard*), 3478
`present()` (*in module salt.states.grafana_datasource*), 3479
`present()` (*in module salt.states.grains*), 3481
`present()` (*in module salt.states.group*), 3483
`present()` (*in module salt.states.host*), 3486
`present()` (*in module salt.states.incron*), 3490
`present()` (*in module salt.states.influxdb08_database*), 3490
`present()` (*in module salt.states.influxdb08_user*), 3491
`present()` (*in module salt.states.influxdb_continuous_query*), 3491
`present()` (*in module salt.states.influxdb_database*), 3491
`present()` (*in module salt.states.influxdb_retention_policy*), 3492
`present()` (*in module salt.states.influxdb_user*), 3492
`present()` (*in module salt.states.infoblox_a*), 3493
`present()` (*in module salt.states.infoblox_cname*), 3493
`present()` (*in module salt.states.infoblox_host_record*), 3494
`present()` (*in module salt.states.infoblox_range*), 3495
`present()` (*in module salt.states.ipset*), 3501
`present()` (*in module salt.states.jenkins*), 3509
`present()` (*in module salt.states.keystone_domain*), 3521
`present()` (*in module salt.states.keystone_endpoint*), 3521
`present()` (*in module salt.states.keystone_group*), 3522

`present()` (*in module salt.states.keystone_project*), 3523
`present()` (*in module salt.states.keystone_role*), 3523
`present()` (*in module salt.states.keystone_role_grant*), 3524
`present()` (*in module salt.states.keystone_service*), 3525
`present()` (*in module salt.states.keystone_user*), 3525
`present()` (*in module salt.states.kmod*), 3527
`present()` (*in module salt.states.layman*), 3530
`present()` (*in module salt.states.linux_acl*), 3539
`present()` (*in module salt.states.locale*), 3540
`present()` (*in module salt.states.lvs_server*), 3544
`present()` (*in module salt.states.lvs_service*), 3544
`present()` (*in module salt.states.lxc*), 3546
`present()` (*in module salt.states.makeconf*), 3551
`present()` (*in module salt.states.mdadm_raid*), 3552
`present()` (*in module salt.states.mongodb_user*), 3561
`present()` (*in module salt.states.mysql_database*), 3566
`present()` (*in module salt.states.mysql_grants*), 3567
`present()` (*in module salt.states.mysql_user*), 3569
`present()` (*in module salt.states.neutron_network*), 3604
`present()` (*in module salt.states.neutron_secgroup*), 3605
`present()` (*in module salt.states.neutron_secgroup_rule*), 3606
`present()` (*in module salt.states.neutron_subnet*), 3607
`present()` (*in module salt.states.nfs_export*), 3608
`present()` (*in module salt.states.openstack_config*), 3615
`present()` (*in module salt.states.openvswitch_bridge*), 3615
`present()` (*in module salt.states.openvswitch_port*), 3615
`present()` (*in module salt.states.pagerduty_escalation_policy*), 3618
`present()` (*in module salt.states.pagerduty_schedule*), 3619
`present()` (*in module salt.states.pagerduty_service*), 3619
`present()` (*in module salt.states.pagerduty_user*), 3620
`present()` (*in module salt.states.pdbedit*), 3638
`present()` (*in module salt.states.postgres_cluster*), 3666
`present()` (*in module salt.states.postgres_database*), 3666
`present()` (*in module salt.states.postgres_extension*), 3667
`present()` (*in module salt.states.postgres_group*), 3668
`present()` (*in module salt.states.postgres_initdb*), 3669
`present()` (*in module salt.states.postgres_language*), 3670
`present()` (*in module salt.states.postgres_privileges*), 3672
`present()` (*in module salt.states.postgres_schema*), 3673
`present()` (*in module salt.states.postgres_tablespace*), 3674
`present()` (*in module salt.states.postgres_user*), 3675
`present()` (*in module salt.states.pyrax_queues*), 3681
`present()` (*in module salt.states.rabbitmq_policy*), 3683
`present()` (*in module salt.states.rabbitmq_upstream*), 3683
`present()` (*in module salt.states.rabbitmq_user*), 3685
`present()` (*in module salt.states.rabbitmq_vhost*), 3685
`present()` (*in module salt.states.reg*), 3690
`present()` (*in module salt.states.schedule*), 3706
`present()` (*in module salt.states.splunk*), 3720
`present()` (*in module salt.states.splunk_search*), 3721
`present()` (*in module salt.states.ssh_auth*), 3725
`present()` (*in module salt.states.ssh_known_hosts*), 3726
`present()` (*in module salt.states.sysctl*), 3731
`present()` (*in module salt.states.telemetry_alert*), 3733
`present()` (*in module salt.states.user*), 3742
`present()` (*in module salt.states.win_dacl*), 3758
`present()` (*in module salt.states.zabbix_action*), 3794
`present()` (*in module salt.states.zabbix_host*), 3795
`present()` (*in module salt.states.zabbix_hostgroup*), 3796
`present()` (*in module salt.states.zabbix_mediatype*), 3797
`present()` (*in module salt.states.zabbix_template*), 3798
`present()` (*in module salt.states.zabbix_user*), 3801
`present()` (*in module salt.states.zabbix_usergroup*), 3803
`present()` (*in module salt.states.zabbix_valuemap*), 3803
`present()` (*in module salt.states.zone*), 3812
`present()` (*in module salt.states.zookeeper*), 3815
`present()` (*in module salt.states.zpool*), 3817
`preserve_minion_cache`
  `conf/master`, 73
`primary_group()` (*in module salt.modules.mac_user*), 2059
`primary_group()` (*in module salt.modules.useradd*), 2633
`primary_suffix()` (*in module salt.states.win_dns_client*), 3761
`print_job()` (*in module salt.runners.jobs*), 3118
`private_ips_absent()` (*in module salt.states.boto_ec2*), 3277
`private_ips_present()` (*in module salt.states.boto_ec2*), 3277
`private_key_managed()` (*in module salt.states.x509*), 3792

PrivEvent (*class in salt.engines.ircbot*), 1156
privileges_grant() (*in module salt.modules.postgres*), 2380
privileges_list() (*in module salt.modules.postgres*), 2380
privileges_revoke() (*in module salt.modules.postgres*), 2381
prlctl() (*in module salt.modules.parallels*), 2322
prlsrvctl() (*in module salt.modules.parallels*), 2322
probe() (*in module salt.modules.parted_partition*), 2327
proc_info() (*in module salt.modules.ps*), 2392
process() (*in module salt.states.status*), 3727
process_acl() (*in module salt.auth.ldap*), 1001
process_count_max
  conf/minion, 160
process_envs() (*in module salt.modules.osquery*), 2293
process_fields() (*salt.pillar.sql_base.SqlBaseExtPillar method*), 3035
process_memory_map() (*in module salt.modules.osquery*), 2293
process_open_files() (*in module salt.modules.osquery*), 2294
process_open_sockets() (*in module salt.modules.osquery*), 2294
process_queue() (*in module salt.runners.queue*), 3141
process_results() (*salt.pillar.sql_base.SqlBaseExtPillar method*), 3035
process_runner() (*in module salt.runners.queue*), 3141
processes() (*in module salt.modules.osquery*), 2294
processlist() (*in module salt.modules.mysql*), 2103
processPath() (*salt.modules.win_dacl.daclConstants method*), 2722
procs() (*in module salt.modules.status*), 2561
procs() (*in module salt.modules.win_status*), 2811
product() (*in module salt.grains.philips_hue*), 1185
profile() (*in module salt.modules.tuned*), 2625
profile() (*in module salt.runners.cloud*), 3103
profile() (*in module salt.states.cloud*), 3343
profile() (*in module salt.states.tuned*), 3741
profile() (*salt.cloud.CloudClient method*), 3853
profile_() (*in module salt.modules.cloud*), 1521
profile_add() (*in module salt.modules.rbac_solaris*), 2423
profile_associated() (*in module salt.modules.boto_iam*), 1404
profile_config_delete() (*in module salt.modules.lxd*), 2016
profile_config_get() (*in module salt.modules.lxd*), 2017
profile_config_set() (*in module salt.modules.lxd*), 2017
profile_create() (*in module salt.modules.lxd*), 2017
profile_delete() (*in module salt.modules.lxd*), 2018
profile_device_delete() (*in module salt.modules.lxd*), 2018
profile_device_get() (*in module salt.modules.lxd*), 2018
profile_device_set() (*in module salt.modules.lxd*), 2019
profile_get() (*in module salt.modules.lxd*), 2019
profile_get() (*in module salt.modules.rbac_solaris*), 2423
profile_list() (*in module salt.modules.lxd*), 2019
profile_list() (*in module salt.modules.rbac_solaris*), 2423
profile_rm() (*in module salt.modules.rbac_solaris*), 2423
progress_iter() (*in module salt.output.progress*), 2987
project_absent() (*in module salt.states.keystone*), 3519
project_create() (*in module salt.modules.keystone*), 1931
project_create() (*in module salt.modules.keystoneng*), 1939
project_delete() (*in module salt.modules.keystone*), 1931
project_delete() (*in module salt.modules.keystoneng*), 1940
project_get() (*in module salt.modules.keystone*), 1932
project_get() (*in module salt.modules.keystoneng*), 1940
project_list() (*in module salt.modules.keystone*), 1932
project_list() (*in module salt.modules.keystoneng*), 1940
project_present() (*in module salt.states.keystone*), 3519
project_search() (*in module salt.modules.keystoneng*), 1940
project_update() (*in module salt.modules.keystone*), 1932
project_update() (*in module salt.modules.keystoneng*), 1940
promote() (*in module salt.modules.zfs*), 2903
promoted() (*in module salt.states.zfs*), 3808
prop_has_value() (*in module salt.states.pcs*), 3635
prop_set() (*in module salt.modules.pcs*), 2330
prop_show() (*in module salt.modules.pcs*), 2330
properties() (*in module salt.modules.btrfs*), 1468
property_absent() (*in module salt.states.zone*),

3813
property_present() (*in module salt.states.zone*), 3813
provider() (*in module salt.modules.test*), 2602
providers
    conf/minion, 146
providers() (*in module salt.modules.test*), 2602
Proxy Minion, **5402**
proxy_always_alive
    conf/proxy, 169
proxy_functions() (*in module salt.grains.rest_sample*), 1185
proxy_host
    conf/minion, 141
proxy_keep_alive
    conf/proxy, 169
proxy_keep_alive_interval
    conf/proxy, 169
proxy_merge_grains_in_module
    conf/proxy, 168
proxy_password
    conf/minion, 142
proxy_port
    conf/minion, 142
proxy_reconnect() (*in module salt.modules.status*), 2561
proxy_username
    conf/minion, 142
ProxyCaller (*class in salt.client*), 3849
proxytype() (*in module salt.proxy.junos*), 3065
prune() (*in module salt.modules.bower*), 1463
prune() (*in module salt.modules.dockermod*), 1646
prune_dump() (*in module salt.modules.xfs*), 2861
pruned() (*in module salt.states.bower*), 3335
ps() (*in module salt.modules.dockercompose*), 1612
ps_() (*in module salt.modules.dockermod*), 1647
psaux() (*in module salt.modules.ps*), 2392
psed() (*in module salt.modules.file*), 1724
psql_query() (*in module salt.modules.postgres*), 2382
pub_hwm
    conf/master, 74
pub_ret
    conf/minion, 139
public_ip_address_absent() (*in module salt.states.azurearm_network*), 3210
public_ip_address_create_or_update() (*in module salt.modules.azurearm_network*), 1261
public_ip_address_delete() (*in module salt.modules.azurearm_network*), 1261
public_ip_address_get() (*in module salt.modules.azurearm_network*), 1261
public_ip_address_present() (*in module salt.states.azurearm_network*), 3210
public_ip_addresses_list() (*in module salt.modules.azurearm_network*), 1261
public_ip_addresses_list_all() (*in module salt.modules.azurearm_network*), 1261
publish() (*in module salt.modules.publish*), 2394
publish_port
    conf/master, 51
    conf/minion, 128
publish_session
    conf/master, 73
publisher_acl
    conf/master, 70
publisher_acl_blacklist
    conf/master, 70
PublishError, 3999
pull() (*in module salt.modules.dockercompose*), 1613
pull() (*in module salt.modules.dockermod*), 1647
pull() (*in module salt.modules.git*), 1780
pull() (*in module salt.modules.hg*), 1848
pull() (*salt.pillar.hg_pillar.Repo method*), 3009
pull_dkr() (*in module salt.modules.nspawn*), 2261
pull_raw() (*in module salt.modules.nspawn*), 2262
pull_tar() (*in module salt.modules.nspawn*), 2262
purchase_reserved_elasticsearch_instance_offering(
    (*in module salt.modules.boto3_elasticsearch*), 1318
purge() (*in module salt.modules.apkpkg*), 1215
purge() (*in module salt.modules.aptpkg*), 1223
purge() (*in module salt.modules.ceph*), 1497
purge() (*in module salt.modules.ebuildpkg*), 1678
purge() (*in module salt.modules.openbsdpkg*), 2271
purge() (*in module salt.modules.opkg*), 2284
purge() (*in module salt.modules.pacmanpkg*), 2300
purge() (*in module salt.modules.pkgin*), 2352
purge() (*in module salt.modules.pkgutil*), 2369
purge() (*in module salt.modules.schedule*), 2483
purge() (*in module salt.modules.solarisipspkg*), 2522
purge() (*in module salt.modules.solarispkg*), 2526
purge() (*in module salt.modules.varnish*), 2638
purge() (*in module salt.modules.virt*), 2667
purge() (*in module salt.modules.win_pkg*), 2782
purge() (*in module salt.modules.yumpkg*), 2873
purge() (*in module salt.modules.zypperpkg*), 2931
purge() (*in module salt.runners.lxc*), 3120
purge() (*in module salt.runners.virt*), 3159
purge() (*salt.modules.inspectlib.dbhandle.DBHandleBase method*), 1875
purge() (*salt.modules.inspectlib.fsdb.CsvDB method*), 1877
purge_pip() (*in module salt.states.pip_state*), 3642
purged() (*in module salt.states.pkg*), 3655
push() (*in module salt.modules.cp*), 1578
push() (*in module salt.modules.dockermod*), 1648
push() (*in module salt.modules.git*), 1781

push_dir() (*in module salt.modules.cp*), 1578
push_note() (*in module salt.modules.pushbullet*), 2404
put() (*in module salt.modules.consul*), 1571
put() (*in module salt.modules.s3*), 2460
put() (*in module salt.modules.scp_mod*), 2484
put() (*in module salt.modules.swift*), 2574
put() (*salt.engines.logentries.PlainTextSocketAppender method*), 1159
put_() (*in module salt.modules.mdata*), 2076
put_acl() (*in module salt.modules.boto_s3_bucket*), 1438
put_blob() (*in module salt.cloud.clouds.msazure*), 1081
put_cors() (*in module salt.modules.boto_s3_bucket*), 1438
put_group_policy() (*in module salt.modules.boto_iam*), 1404
put_key_policy() (*in module salt.modules.boto_kms*), 1418
put_lifecycle_configuration() (*in module salt.modules.boto_s3_bucket*), 1439
put_logging() (*in module salt.modules.boto_s3_bucket*), 1439
put_notification_configuration() (*in module salt.modules.boto_s3_bucket*), 1439
put_pipeline_definition() (*in module salt.modules.boto_datapipeline*), 1359
put_policy() (*in module salt.modules.boto_s3_bucket*), 1439
put_replication() (*in module salt.modules.boto_s3_bucket*), 1440
put_request_payment() (*in module salt.modules.boto_s3_bucket*), 1440
put_tagging() (*in module salt.modules.boto_s3_bucket*), 1440
put_targets() (*in module salt.modules.boto_cloudwatch_event*), 1354
put_user_policy() (*in module salt.modules.boto_iam*), 1404
put_versioning() (*in module salt.modules.boto_s3_bucket*), 1440
put_website() (*in module salt.modules.boto_s3_bucket*), 1440
pv_absent() (*in module salt.states.lvm*), 3543
pv_present() (*in module salt.states.lvm*), 3543
pvcreate() (*in module salt.modules.linux_lvm*), 1977
pvdisplay() (*in module salt.modules.linux_lvm*), 1978
pvremove() (*in module salt.modules.linux_lvm*), 1978
pxe() (*in module salt.runners.drac*), 3108
PyDSL, **5402**
pyeapi_call() (*in module salt.modules.napalm_mod*), 2135
pyeapi_config() (*in module salt.modules.napalm_mod*), 2136
pyeapi_conn() (*in module salt.modules.napalm_mod*), 2136
pyeapi_nxos_api_args() (*in module salt.modules.napalm_mod*), 2136
pyeapi_run_commands() (*in module salt.modules.napalm_mod*), 2137
pylxd_client_get() (*in module salt.modules.lxd*), 2020
pylxd_save_object() (*in module salt.modules.lxd*), 2020
pylxd_version() (*in module salt.modules.lxd*), 2020
PyobjectsModule (*class in salt.renderers.pyobjects*), 379
pythonexecutable() (*in module salt.grains.core*), 1179
pythonpath() (*in module salt.grains.core*), 1179
pythonversion() (*in module salt.grains.core*), 1179

Q

quarantine() (*in module salt.modules.osquery*), 2294
Query (*class in salt.modules.inspectlib.query*), 1876
query() (*in module salt.cloud.clouds.aliyun*), 1033
query() (*in module salt.cloud.clouds.digitalocean*), 1041
query() (*in module salt.cloud.clouds.ec2*), 1050
query() (*in module salt.cloud.clouds.joyent*), 1064
query() (*in module salt.cloud.clouds.msazure*), 1082
query() (*in module salt.cloud.clouds.parallels*), 1114
query() (*in module salt.cloud.clouds.proxmox*), 1120
query() (*in module salt.cloud.clouds.qingcloud*), 1123
query() (*in module salt.cloud.clouds.scaleway*), 1127
query() (*in module salt.modules.cloud*), 1521
query() (*in module salt.modules.http*), 1850
query() (*in module salt.modules.influxdb08mod*), 1863
query() (*in module salt.modules.influxdbmod*), 1860
query() (*in module salt.modules.inspector*), 1879
query() (*in module salt.modules.mysql*), 2103
query() (*in module salt.modules.osquery*), 2294
query() (*in module salt.runners.cloud*), 3103
query() (*in module salt.runners.http*), 3116
query() (*in module salt.runners.virt*), 3159
query() (*in module salt.sdb.rest*), 3168
query() (*in module salt.states.http*), 3486
query() (*in module salt.utils.http*), 1850
query() (*salt.cloud.CloudClient method*), 3853
query_instance() (*in module salt.cloud.clouds.ec2*), 1050
query_instance() (*in module salt.cloud.clouds.joyent*), 1064
query_item() (*in module salt.modules.rallydev*), 2419
query_user() (*in module salt.modules.rallydev*), 2419
queue_exists() (*in module salt.modules.aws_sqs*), 1247
queue_instances() (*in module salt.cloud.clouds.ec2*), 1050

queues_create() (*in module salt.cloud.clouds.pyrax*), 1121
queues_delete() (*in module salt.cloud.clouds.pyrax*), 1121
queues_exists() (*in module salt.cloud.clouds.pyrax*), 1121
queues_show() (*in module salt.cloud.clouds.pyrax*), 1121
quorum() (*in module salt.states.ceph*), 3336
quote
    jinja filters, 430
quote_identifier() (*in module salt.modules.mysql*), 2104

R

raise_exception() (*in module salt.modules.test*), 2603
rand_int() (*in module salt.modules.mod_random*), 2082
rand_sleep() (*in module salt.modules.test*), 2603
random_hash
    jinja filters, 438
random_hash() (*in module salt.modules.test*), 2603
random_master
    conf/minion, 127
random_reauth_delay
    conf/minion, 137
random_startup_delay
    conf/minion, 139
range_server
    conf/master, 116
rar() (*in module salt.modules.archive*), 1230
raw() (*in module salt.modules.pillar*), 2343
raw_arg() (*in module salt.runners.test*), 3156
raw_command() (*in module salt.modules.ipmi*), 1889
raw_cron() (*in module salt.modules.cron*), 1580
raw_incron() (*in module salt.modules.incron*), 1856
raw_interface_configs() (*in module salt.modules.win_ip*), 2765
raw_mod() (*in module salt.loader*), 3844
raw_system_incron() (*in module salt.modules.incron*), 1856
re_encrypt() (*in module salt.modules.boto_kms*), 1418
reactivate() (*in module salt.modules.namecheap_domains*), 2116
Reactor, **5402**
Reactor, *see* Event, **684**
reactor
    conf/master, 110
    conf/minion, 159
reactor_refresh_interval
    conf/master, 110
    conf/minion, 160
reactor_worker_hwm
    conf/master, 110
    conf/minion, 160
reactor_worker_threads
    conf/master, 110
    conf/minion, 160
read() (*in module salt.modules.file*), 1724
read() (*in module salt.modules.mac_xattr*), 2060
read() (*in module salt.modules.macdefaults*), 2025
read() (*in module salt.modules.sysfs*), 2577
read() (*in module salt.wheel.file_roots*), 3838
read() (*in module salt.wheel.pillar_roots*), 3842
read_certificate() (*in module salt.modules.x509*), 2851
read_certificates() (*in module salt.modules.x509*), 2851
read_conf() (*in module salt.modules.lxc*), 1995
read_config() (*in module salt.modules.trafficserver*), 2623
read_crl() (*in module salt.modules.x509*), 2851
read_csr() (*in module salt.modules.x509*), 2851
read_datasource() (*in module salt.modules.jboss7*), 1904
read_file() (*in module salt.modules.pam*), 2305
read_messages() (*salt.engines.ircbot.IRCClient method*), 1156
read_metric() (*in module salt.modules.trafficserver*), 2623
read_mount_cache() (*in module salt.modules.mount*), 2092
read_secret() (*in module salt.modules.vault*), 2640
read_simple_binding() (*in module salt.modules.jboss7*), 1904
read_value() (*in module salt.modules.reg*), 2436
readdir() (*in module salt.modules.file*), 1724
readlink() (*in module salt.modules.file*), 1725
readlink() (*in module salt.modules.win_file*), 2742
ready() (*in module salt.modules.zoneadm*), 2909
reaped() (*in module salt.runners.manage*), 3125
rebase() (*in module salt.modules.git*), 1783
reboot() (*in module salt.cloud.clouds.aliyun*), 1033
reboot() (*in module salt.cloud.clouds.digitalocean*), 1041
reboot() (*in module salt.cloud.clouds.dimensiondata*), 1044
reboot() (*in module salt.cloud.clouds.ec2*), 1050
reboot() (*in module salt.cloud.clouds.gce*), 1057
reboot() (*in module salt.cloud.clouds.gogrid*), 1060
reboot() (*in module salt.cloud.clouds.joyent*), 1064
reboot() (*in module salt.cloud.clouds.linode*), 1069
reboot() (*in module salt.cloud.clouds.oneandone*), 1089
reboot() (*in module salt.cloud.clouds.opennebula*), 1097

reboot() (*in module salt.cloud.clouds.profitbricks*), 1118
reboot() (*in module salt.cloud.clouds.qingcloud*), 1123
reboot() (*in module salt.cloud.clouds.saltify*), 1126
reboot() (*in module salt.cloud.clouds.tencentcloud*), 1132
reboot() (*in module salt.cloud.clouds.vagrant*), 1134
reboot() (*in module salt.cloud.clouds.xen*), 1151
reboot() (*in module salt.modules.cimc*), 1511
reboot() (*in module salt.modules.lxc*), 1996
reboot() (*in module salt.modules.nspawn*), 2262
reboot() (*in module salt.modules.panos*), 2315
reboot() (*in module salt.modules.smartos_virt*), 2503
reboot() (*in module salt.modules.smartos_vmadm*), 2506
reboot() (*in module salt.modules.solaris_system*), 2518
reboot() (*in module salt.modules.system*), 2590
reboot() (*in module salt.modules.vagrant*), 2636
reboot() (*in module salt.modules.virt*), 2668
reboot() (*in module salt.modules.win_system*), 2816
reboot() (*in module salt.modules.xapi_virt*), 2855
reboot() (*in module salt.modules.zoneadm*), 2909
reboot() (*in module salt.runners.drac*), 3108
reboot() (*in module salt.runners.smartos_vmadm*), 3150
reboot() (*in module salt.states.win_system*), 3782
reboot_host() (*in module salt.cloud.clouds.vmware*), 1144
reboot_node() (*in module salt.modules.libcloud_compute*), 1965
rebooted() (*in module salt.states.vagrant*), 3746
rebooted() (*in module salt.states.virt*), 3751
receipts() (*in module salt.modules.system_profiler*), 2592
receive() (*in module salt.modules.smartos_vmadm*), 2506
receive_keys() (*in module salt.modules.gpg*), 1821
receive_message() (*in module salt.modules.aws_sqs*), 1248
recon_default
  conf/minion, 138
recon_max
  conf/minion, 139
recon_randomize
  conf/minion, 139
reconfigure() (*in module salt.modules.lxc*), 1996
reconnect() (*in module salt.modules.napalm_mod*), 2137
record_absent() (*in module salt.states.libcloud_dns*), 3534
record_present() (*in module salt.states.libcloud_dns*), 3535
records() (*in module salt.modules.smbios*), 2509
recover_all() (*in module salt.modules.modjk*), 2085
recurse() (*in module salt.states.file*), 3442
recv() (*in module salt.modules.cp*), 1578
recv_chunked() (*in module salt.modules.cp*), 1578
recv_known_host_entries() (*in module salt.modules.ssh*), 2540
reformat_node() (*in module salt.cloud.clouds.joyent*), 1064
refresh() (*in module salt.modules.apf*), 1212
refresh() (*in module salt.modules.trafficserver*), 2623
refresh() (*in module salt.states.trafficserver*), 3740
refresh_beacons() (*in module salt.modules.saltutil*), 2471
refresh_db() (*in module salt.modules.apkpkg*), 1215
refresh_db() (*in module salt.modules.aptpkg*), 1223
refresh_db() (*in module salt.modules.ebuildpkg*), 1678
refresh_db() (*in module salt.modules.freebsdpkg*), 1744
refresh_db() (*in module salt.modules.mac_brew_pkg*), 2024
refresh_db() (*in module salt.modules.mac_portspkg*), 2033
refresh_db() (*in module salt.modules.opkg*), 2284
refresh_db() (*in module salt.modules.pacmanpkg*), 2300
refresh_db() (*in module salt.modules.pkgin*), 2352
refresh_db() (*in module salt.modules.pkgng*), 2361
refresh_db() (*in module salt.modules.pkgutil*), 2369
refresh_db() (*in module salt.modules.solarisipspkg*), 2523
refresh_db() (*in module salt.modules.win_pkg*), 2782
refresh_db() (*in module salt.modules.xbpspkg*), 2859
refresh_db() (*in module salt.modules.yumpkg*), 2874
refresh_db() (*in module salt.modules.zypperpkg*), 2932
refresh_fqdn_cache() (*in module salt.modules.panos*), 2315
refresh_grains() (*in module salt.modules.saltutil*), 2471
refresh_matchers() (*in module salt.modules.saltutil*), 2472
refresh_modules() (*in module salt.modules.saltutil*), 2472
refresh_pillar() (*in module salt.modules.saltutil*), 2472
reg() (*in module salt.thorium.status*), 3827
regen_keys() (*in module salt.modules.saltutil*), 2472
regenerate_storage_keys() (*in module salt.cloud.clouds.msazure*), 1082
regex_escape
  jinja filters, 450
regex_match
  jinja filters, 431
regex_replace

jinja filters, 431
regex_search
jinja filters, 430
register() (*in module salt.modules.vboxmanage*), 2644
register_image() (*in module salt.cloud.clouds.ec2*), 1050
register_instances() (*in module salt.modules.boto_elb*), 1389
register_instances() (*in module salt.states.boto_elb*), 3290
register_repository() (*in module salt.modules.win_psget*), 2790
register_targets() (*in module salt.modules.boto_elbv2*), 1393
register_vm() (*in module salt.modules.vsphere*), 2707
reguid() (*in module salt.modules.zpool*), 2921
rehash() (*in module salt.modules.pyenv*), 2410
rehash() (*in module salt.modules.rbenv*), 2425
rehash() (*in module salt.modules.win_path*), 2777
rehashconf() (*in module salt.modules.znc*), 2907
reinstall_ruby() (*in module salt.modules.rvm*), 2458
reissue() (*in module salt.modules.namecheap_ssl*), 2122
reject() (*in module salt.wheel.key*), 3841
reject_dict() (*in module salt.wheel.key*), 3841
rejected_retry
conf/minion, 137
relay_ip_list() (*in module salt.states.win_smtp_server*), 3779
release() (*in module salt.modules.zfs*), 2903
release_eip_address() (*in module salt.modules.boto_ec2*), 1371
reload() (*in module salt.modules.apf*), 1213
reload() (*in module salt.modules.csf*), 1583
reload() (*in module salt.modules.vmctl*), 2678
reload_() (*in module salt.modules.daemontools*), 1585
reload_() (*in module salt.modules.debian_service*), 1599
reload_() (*in module salt.modules.freebsdservice*), 1748
reload_() (*in module salt.modules.gentoo_service*), 1755
reload_() (*in module salt.modules.jboss7*), 1905
reload_() (*in module salt.modules.monit*), 2089
reload_() (*in module salt.modules.netbsdservice*), 2204
reload_() (*in module salt.modules.openbsdrcctl_service*), 2273
reload_() (*in module salt.modules.openbsdservice*), 2275
reload_() (*in module salt.modules.rh_service*), 2444
reload_() (*in module salt.modules.runit*), 2454
reload_() (*in module salt.modules.s6*), 2461
reload_() (*in module salt.modules.schedule*), 2483
reload_() (*in module salt.modules.smf_service*), 2511
reload_() (*in module salt.modules.syslog_ng*), 2580
reload_() (*in module salt.modules.systemd_service*), 2595
reload_() (*in module salt.modules.tomcat*), 2620
reload_() (*in module salt.modules.upstart_service*), 2629
reload_core() (*in module salt.modules.solr*), 2531
reload_exports() (*in module salt.modules.nfs3*), 2243
reload_import_config() (*in module salt.modules.solr*), 2532
reload_modules() (*in module salt.modules.sysmod*), 2585
reload_rules() (*in module salt.modules.firewalld*), 1736
reloaded() (*in module salt.states.jboss7*), 3509
reloaded() (*in module salt.states.syslog_ng*), 3732
remote_get() (*in module salt.modules.git*), 1784
remote_refs() (*in module salt.modules.git*), 1784
remote_set() (*in module salt.modules.git*), 1786
remotes() (*in module salt.modules.git*), 1787
remount() (*in module salt.modules.mount*), 2092
remove
spm command line option, 991
remove() (*in module salt.modules.aixpkg*), 1205
remove() (*in module salt.modules.alternatives*), 1207
remove() (*in module salt.modules.apf*), 1213
remove() (*in module salt.modules.apkpkg*), 1215
remove() (*in module salt.modules.aptpkg*), 1224
remove() (*in module salt.modules.augeas_cfg*), 1245
remove() (*in module salt.modules.cpan*), 1579
remove() (*in module salt.modules.dummyproxy_pkg*), 1673
remove() (*in module salt.modules.ebuildpkg*), 1679
remove() (*in module salt.modules.file*), 1725
remove() (*in module salt.modules.freebsdkmod*), 1741
remove() (*in module salt.modules.freebsdpkg*), 1744
remove() (*in module salt.modules.grains*), 1835
remove() (*in module salt.modules.kernelpkg_linux_apt*), 1925
remove() (*in module salt.modules.kernelpkg_linux_yum*), 1927
remove() (*in module salt.modules.keystore*), 1944
remove() (*in module salt.modules.kmod*), 1945
remove() (*in module salt.modules.logadm*), 1981
remove() (*in module salt.modules.mac_assistive*), 2022
remove() (*in module salt.modules.mac_brew_pkg*), 2024
remove() (*in module salt.modules.mac_portspkg*), 2034
remove() (*in module salt.modules.mongodb*), 2087
remove() (*in module salt.modules.nspawn*), 2263

remove() (*in module salt.modules.openbsdpkg*), 2271
remove() (*in module salt.modules.opkg*), 2284
remove() (*in module salt.modules.pacmanpkg*), 2300
remove() (*in module salt.modules.pkgin*), 2353
remove() (*in module salt.modules.pkgng*), 2361
remove() (*in module salt.modules.pkgutil*), 2369
remove() (*in module salt.modules.rest_pkg*), 2439
remove() (*in module salt.modules.runit*), 2455
remove() (*in module salt.modules.solarisipspkg*), 2523
remove() (*in module salt.modules.solarispkg*), 2526
remove() (*in module salt.modules.ssh_pkg*), 2542
remove() (*in module salt.modules.supervisord*), 2566
remove() (*in module salt.modules.svn*), 2570
remove() (*in module salt.modules.sysrc*), 2588
remove() (*in module salt.modules.win_file*), 2742
remove() (*in module salt.modules.win_path*), 2777
remove() (*in module salt.modules.win_pkg*), 2783
remove() (*in module salt.modules.win_psget*), 2790
remove() (*in module salt.modules.win_servermanager*), 2793
remove() (*in module salt.modules.xbpspkg*), 2859
remove() (*in module salt.modules.yumpkg*), 2874
remove() (*in module salt.modules.zypperpkg*), 2932
remove() (*in module salt.states.alternatives*), 3184
remove() (*in module salt.states.logadm*), 3540
remove() (*salt.modules.win_timezone.TzMapper method*), 2831
remove_all_snapshots() (*in module salt.cloud.clouds.vmware*), 1145
remove_app() (*in module salt.modules.win_iis*), 2760
remove_app() (*in module salt.states.win_iis*), 3766
remove_apppool() (*in module salt.modules.win_iis*), 2760
remove_apppool() (*in module salt.states.win_iis*), 3766
remove_backup() (*in module salt.modules.win_iis*), 2760
remove_binding() (*in module salt.modules.win_iis*), 2761
remove_binding() (*in module salt.states.win_iis*), 3767
remove_capability() (*in module salt.modules.win_dism*), 2727
remove_capacity_from_diskgroup() (*in module salt.modules.vsphere*), 2707
remove_cert() (*in module salt.modules.win_pki*), 2786
remove_cert() (*in module salt.states.win_pki*), 3775
remove_cert_binding() (*in module salt.modules.win_iis*), 2761
remove_cert_binding() (*in module salt.states.win_iis*), 3767
remove_config() (*in module salt.modules.napalm_snmp*), 2185
remove_config() (*in module salt.modules.win_dsc*), 2731
remove_config_lock() (*in module salt.modules.panos*), 2315
remove_config_lock() (*in module salt.states.panos*), 3626
remove_datasource() (*in module salt.modules.jboss7*), 1905
remove_datastore() (*in module salt.modules.vsphere*), 2707
remove_diskgroup() (*in module salt.modules.vsphere*), 2707
remove_dvportgroup() (*in module salt.modules.vsphere*), 2708
remove_feature() (*in module salt.modules.win_dism*), 2727
remove_gateway_router() (*in module salt.modules.neutron*), 2232
remove_host() (*in module salt.cloud.clouds.vmware*), 1145
remove_interface() (*in module salt.modules.firewalld*), 1736
remove_interface_router() (*in module salt.modules.neutron*), 2233
remove_key() (*in module salt.cloud.clouds.digitalocean*), 1042
remove_license() (*in module salt.modules.powerpath*), 2387
remove_lock() (*in module salt.modules.zypperpkg*), 2932
remove_masquerade() (*in module salt.modules.firewalld*), 1736
remove_network() (*in module salt.modules.dockermod*), 1649
remove_node() (*in module salt.modules.swarm*), 2572
remove_option() (*in module salt.modules.ini_manage*), 1873
remove_package() (*in module salt.modules.win_dism*), 2727
remove_permission() (*in module salt.modules.boto_lambda*), 1423
remove_platform() (*in module salt.runners.asam*), 3096
remove_port() (*in module salt.modules.firewalld*), 1736
remove_port_fwd() (*in module salt.modules.firewalld*), 1736
remove_repo() (*in module salt.modules.github*), 1807
remove_resource() (*in module salt.modules.zonecfg*), 2911
remove_rich_rule() (*in module salt.modules.firewalld*), 1737
remove_rule() (*in module salt.modules.csf*), 1583
remove_section() (*in module*

salt.modules.ini_manage), 1873
remove_semod() (*in module salt.modules.selinux*), 2491
remove_service() (*in module salt.modules.firewalld*), 1737
remove_service() (*in module salt.modules.swarm*), 2572
remove_service_port() (*in module salt.modules.firewalld*), 1737
remove_service_protocol() (*in module salt.modules.firewalld*), 1737
remove_site() (*in module salt.modules.win_iis*), 2761
remove_site() (*in module salt.states.win_iis*), 3767
remove_snapshot() (*in module salt.cloud.clouds.vmware*), 1145
remove_source() (*in module salt.modules.firewalld*), 1737
remove_tags() (*in module salt.modules.boto3_elasticsearch*), 1318
remove_tags() (*in module salt.modules.boto_cloudtrail*), 1350
remove_tags() (*in module salt.modules.boto_elasticsearch_domain*), 1385
remove_tags_from_resource() (*in module salt.modules.boto3_elasticache*), 1307
remove_targets() (*in module salt.modules.boto_cloudwatch_event*), 1354
remove_team() (*in module salt.modules.github*), 1807
remove_team_member() (*in module salt.modules.github*), 1807
remove_team_repo() (*in module salt.modules.github*), 1808
remove_temp_rule() (*in module salt.modules.csf*), 1583
remove_user() (*in module salt.modules.github*), 1808
remove_user() (*in module salt.modules.influxdbmod*), 1860
remove_user() (*in module salt.proxy.nxos*), 3075
remove_user_from_group() (*in module salt.modules.boto_iam*), 1405
remove_var() (*in module salt.modules.makeconf*), 2064
remove_vdir() (*in module salt.modules.win_iis*), 2761
remove_vdir() (*in module salt.states.win_iis*), 3768
remove_volume() (*in module salt.modules.dockermod*), 1649
removed (*salt.modules.file.AttrChanges attribute*), 1703
removed() (*in module salt.states.bower*), 3336
removed() (*in module salt.states.cabal*), 3336
removed() (*in module salt.states.cyg*), 3361
removed() (*in module salt.states.gem*), 3454
removed() (*in module salt.states.npm*), 3612
removed() (*in module salt.states.pecl*), 3637
removed() (*in module salt.states.pip_state*), 3642
removed() (*in module salt.states.pkg*), 3656
removed() (*in module salt.states.win_servermanager*), 3778
removed() (*in module salt.states.win_wua*), 3785
removed() (*salt.states.cyg.DictDiffer method*), 3360
removegroup() (*in module salt.modules.win_useradd*), 2837
rename() (*in module salt.cloud.clouds.ec2*), 1051
rename() (*in module salt.modules.dockermod*), 1649
rename() (*in module salt.modules.file*), 1725
rename() (*in module salt.modules.mac_user*), 2059
rename() (*in module salt.modules.pw_user*), 2409
rename() (*in module salt.modules.solaris_user*), 2520
rename() (*in module salt.modules.useradd*), 2633
rename() (*in module salt.modules.win_useradd*), 2837
rename() (*in module salt.modules.zfs*), 2903
rename() (*in module salt.states.file*), 3444
rename_config() (*in module salt.states.panos*), 3626
rename_datastore() (*in module salt.modules.vsphere*), 2708
rename_set() (*in module salt.modules.ipset*), 1897
Render Pipe, **5402**
render() (*in module salt.renderers.cheetah*), 361
render() (*in module salt.renderers.dson*), 361
render() (*in module salt.renderers.genshi*), 362
render() (*in module salt.renderers.gpg*), 365
render() (*in module salt.renderers.hjson*), 365
render() (*in module salt.renderers.jinja*), 365
render() (*in module salt.renderers.json*), 368
render() (*in module salt.renderers.json5*), 368
render() (*in module salt.renderers.mako*), 368
render() (*in module salt.renderers.msgpack*), 368
render() (*in module salt.renderers.pass*), 369
render() (*in module salt.renderers.py*), 371
render() (*in module salt.renderers.pydsl*), 375
render() (*in module salt.renderers.pyobjects*), 380
render() (*in module salt.renderers.wempy*), 384
render() (*in module salt.renderers.yaml*), 385
render() (*in module salt.renderers.yamlex*), 386
render_dirs
  conf/minion, 145
render_field() (*in module salt.modules.napalm_formula*), 2153
render_fields() (*in module salt.modules.napalm_formula*), 2154
Renderer, **5402**
renderer
  conf/master, 77
  conf/minion, 150
renderer() (*in module salt.modules.slsutil*), 2499
renderer_doc() (*in module salt.modules.sysmod*), 2585

renew() (*in module salt.modules.namecheap_domains*), 2116
renew() (*in module salt.modules.namecheap_ssl*), 2123
renew() (*in module salt.runners.venafiapi*), 3158
renew_by() (*in module salt.modules.acme*), 1202
reopen() (*in module salt.modules.zpool*), 2922
reopen_connection() (*salt.engines.logentries.PlainTextSocketAppender method*), 1159
repaired() (*in module salt.modules.solaris_fmadm*), 2514
replace() (*in module salt.modules.file*), 1725
replace() (*in module salt.modules.memcached*), 2077
replace() (*in module salt.modules.zpool*), 2922
replace() (*in module salt.proxy.nxos*), 3075
replace() (*in module salt.states.file*), 3444
replace() (*in module salt.states.nxos*), 3613
replace_configmap() (*in module salt.modules.kubernetesmod*), 1949
replace_deployment() (*in module salt.modules.kubernetesmod*), 1949
replace_network_acl_entry() (*in module salt.modules.boto_vpc*), 1460
replace_pattern() (*in module salt.modules.napalm_network*), 2173
replace_pattern() (*in module salt.states.netconfig*), 3587
replace_pool_members() (*in module salt.modules.bigip*), 1299
replace_route() (*in module salt.modules.boto_vpc*), 1460
replace_route_table_association() (*in module salt.modules.boto_vpc*), 1461
replace_secret() (*in module salt.modules.kubernetesmod*), 1949
replace_service() (*in module salt.modules.kubernetesmod*), 1950
replace_topic_rule() (*in module salt.modules.boto_iot*), 1411
replaced() (*in module salt.modules.solaris_fmadm*), 2514
replica_present() (*in module salt.states.boto_rds*), 3315
replication_details() (*in module salt.modules.solr*), 2532
replication_group_absent() (*in module salt.states.boto3_elasticache*), 3239
replication_group_absent() (*in module salt.states.boto_elasticache*), 3282
replication_group_exists() (*in module salt.modules.boto3_elasticache*), 1307
replication_group_present() (*in module salt.states.boto3_elasticache*), 3239
replication_group_present() (*in module salt.states.boto_elasticache*), 3282
Repo (*class in salt.pillar.hg_pillar*), 3009
repo() (*in module salt.states.pkgbuild*), 3659
repo_absent() (*in module salt.states.github*), 3463
repo_present() (*in module salt.states.github*), 3464
report() (*in module salt.modules.quota*), 2412
report_highstate_tests() (*in module salt.modules.saltcheck*), 2468
repository_create() (*in module salt.modules.elasticsearch*), 1690
repository_delete() (*in module salt.modules.elasticsearch*), 1690
repository_get() (*in module salt.modules.elasticsearch*), 1690
repository_verify() (*in module salt.modules.elasticsearch*), 1691
reprovision() (*in module salt.modules.smartos_vmadm*), 2506
request() (*in module salt.modules.state*), 2553
request() (*in module salt.runners.venafiapi*), 3158
request_cert() (*in module salt.modules.icinga2*), 1855
request_cert() (*in module salt.states.icinga2*), 3488
request_instance() (*in module salt.cloud.clouds.azurearm*), 1037
request_instance() (*in module salt.cloud.clouds.ec2*), 1051
request_instance() (*in module salt.cloud.clouds.gce*), 1057
request_instance() (*in module salt.cloud.clouds.openstack*), 1113
request_snapshot() (*salt.modules.inspectlib.collector.Inspector method*), 1875
request_vpc_peering_connection() (*in module salt.modules.boto_vpc*), 1461
request_vpc_peering_connection() (*in module salt.states.boto_vpc*), 3332
requeue() (*in module salt.modules.postfix*), 2372
require_password_change() (*in module salt.modules.win_shadow*), 2801
reread() (*in module salt.modules.supervisord*), 2566
rescan_all() (*in module salt.modules.scsi*), 2485
rescan_hba() (*in module salt.cloud.clouds.vmware*), 1145
rescue() (*in module salt.modules.parted_partition*), 2327
reserve_ipblock() (*in module salt.cloud.clouds.profitbricks*), 1118
reset() (*in module salt.cloud.clouds.vmware*), 1145
reset() (*in module salt.modules.beacons*), 1285
reset() (*in module salt.modules.git*), 1787
reset() (*in module salt.modules.parallels*), 2322
reset() (*in module salt.modules.rabbitmq*), 2416

`reset()` (*in module salt.modules.solaris_fmadm*), 2515
`reset()` (*in module salt.modules.virt*), 2668
`reset()` (*in module salt.modules.vmctl*), 2678
`reset()` (*in module salt.modules.xapi_virt*), 2855
`reset()` (*in module salt.runners.virt*), 3159
`reset_catalog()` (*in module salt.modules.mac_softwareupdate*), 2046
`reset_ignored()` (*in module salt.modules.mac_softwareupdate*), 2046
`reset_stats()` (*in module salt.modules.modjk*), 2085
`reset_syslog_config()` (*in module salt.modules.vsphere*), 2708
`reshard()` (*in module salt.modules.boto_kinesis*), 1414
`ResidentAvailablePages` (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATION attribute*), 2809
`ResidentPagedPoolPage` (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATION attribute*), 2809
`ResidentSystemCachePage` (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATION attribute*), 2809
`ResidentSystemCodePage` (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATION attribute*), 2809
`ResidentSystemDriverPage` (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATION attribute*), 2810
`resize()` (*in module salt.modules.btrfs*), 1468
`resize()` (*in module salt.modules.parted_partition*), 2327
`resize2fs()` (*in module salt.modules.disk*), 1604
`resolve()` (*salt.modules.ansiblegate.AnsibleModuleResolver method*), 1208
`resolve_address()` (*in module salt.modules.panos*), 2315
`resolve_capabilities()` (*in module salt.modules.zypperpkg*), 2932
`resolve_image_id()` (*in module salt.modules.dockermod*), 1649
`resolve_tag()` (*in module salt.modules.dockermod*), 1649
`resource_absent()` (*in module salt.modules.pagerduty_util*), 2304
`resource_absent()` (*in module salt.states.zone*), 3813
`resource_create()` (*in module salt.modules.pcs*), 2330
`resource_defaults_to()` (*in module salt.states.pcs*), 3635
`resource_exists()` (*in module salt.modules.boto_vpc*), 1462
`resource_group_absent()` (*in module salt.states.azurearm_resource*), 3220
`resource_group_check_existence()` (*in module salt.modules.azurearm_resource*), 1277
`resource_group_create_or_update()` (*in module salt.modules.azurearm_resource*), 1277
`resource_group_delete()` (*in module salt.modules.azurearm_resource*), 1277
`resource_group_get()` (*in module salt.modules.azurearm_resource*), 1278
`resource_group_present()` (*in module salt.states.azurearm_resource*), 3220
`resource_groups_list()` (*in module salt.modules.azurearm_resource*), 1278
`resource_op_defaults_to()` (*in module salt.states.pcs*), 3635
`resource_present()` (*in module salt.modules.pagerduty_util*), 2304
`resource_present()` (*in module salt.states.pcs*), 3636
`resource_present()` (*in module salt.states.zone*), 3813
`resource_show()` (*in module salt.modules.pcs*), 2331
`resp` (*salt.auth.pam.PamResponse attribute*), 1003
`resp_retcode` (*salt.auth.pam.PamResponse attribute*), 1003
`rest_auth_setup()` (*in module salt.auth.rest*), 1004
`rest_timeout`
`conf/master`, 111
`restart()` (*in module salt.modules.daemontools*), 1586
`restart()` (*in module salt.modules.debian_service*), 1599
`restart()` (*in module salt.modules.dockercompose*), 1613
`restart()` (*in module salt.modules.dockermod*), 1650
`restart()` (*in module salt.modules.dummyproxy_service*), 1674
`restart()` (*in module salt.modules.freebsdjail*), 1739
`restart()` (*in module salt.modules.freebsdservice*), 1748
`restart()` (*in module salt.modules.gentoo_service*), 1755
`restart()` (*in module salt.modules.launchctl_service*), 1952
`restart()` (*in module salt.modules.lxc*), 1996
`restart()` (*in module salt.modules.mac_service*), 2040
`restart()` (*in module salt.modules.mac_system*), 2050
`restart()` (*in module salt.modules.minion*), 2081
`restart()` (*in module salt.modules.monit*), 2090
`restart()` (*in module salt.modules.netbsdservice*), 2204
`restart()` (*in module salt.modules.openbsdrcctl_service*), 2274
`restart()` (*in module salt.modules.openbsdservice*), 2275
`restart()` (*in module salt.modules.parallels*), 2323
`restart()` (*in module salt.modules.rest_service*), 2439

restart() (*in module salt.modules.rh_service*), 2444
restart() (*in module salt.modules.runit*), 2455
restart() (*in module salt.modules.s6*), 2461
restart() (*in module salt.modules.smf_service*), 2512
restart() (*in module salt.modules.ssh_service*), 2543
restart() (*in module salt.modules.supervisord*), 2567
restart() (*in module salt.modules.systemd_service*), 2596
restart() (*in module salt.modules.upstart_service*), 2629
restart() (*in module salt.modules.win_service*), 2799
restart_app() (*in module salt.modules.marathon*), 2070
restart_apppool() (*in module salt.modules.win_iis*), 2762
restart_cluster() (*in module salt.modules.trafficserver*), 2623
restart_cluster() (*in module salt.states.trafficserver*), 3740
restart_local() (*in module salt.modules.trafficserver*), 2623
restart_local() (*in module salt.states.trafficserver*), 3740
restart_service() (*in module salt.modules.openstack_mng*), 2277
restart_site() (*in module salt.modules.win_iis*), 2762
restartcheck() (*in module salt.modules.restartcheck*), 2440
restore() (*in module salt.modules.pkgng*), 2362
restore_backup() (*in module salt.modules.file*), 1727
restore_config() (*in module salt.modules.win_dsc*), 2731
restorecon() (*in module salt.modules.file*), 1727
result (*salt.pillar.sql_base.SqlBaseExtPillar attribute*), 3035
resultdecorator() (*in module salt.states.junos*), 3513
results() (*in module salt.modules.napalm_probes*), 2180
resume() (*in module salt.cloud.clouds.xen*), 1151
resume() (*in module salt.modules.nova*), 2255
resume() (*in module salt.modules.state*), 2553
resume() (*in module salt.modules.virt*), 2668
resume() (*in module salt.modules.xapi_virt*), 2855
resume() (*in module salt.runners.state*), 3154
resume() (*in module salt.runners.virt*), 3159
ret_glob_minions() (*salt.roster.sshconfig.RosterMatcher method*), 3093
ret_port
  conf/master, 52
retcode() (*in module salt.modules.cmdmod*), 1529
retcode() (*in module salt.modules.dockermod*), 1650
retcode() (*in module salt.modules.lxc*), 1997
retcode() (*in module salt.modules.nagios*), 2111
retcode() (*in module salt.modules.nspawn*), 2263
retcode() (*in module salt.modules.test*), 2603
retcode_pillar() (*in module salt.modules.nagios*), 2111
retention_policy_add() (*in module salt.modules.influxdb08mod*), 1864
retention_policy_alter() (*in module salt.modules.influxdb08mod*), 1864
retention_policy_exists() (*in module salt.modules.influxdb08mod*), 1864
retention_policy_exists() (*in module salt.modules.influxdbmod*), 1861
retention_policy_get() (*in module salt.modules.influxdb08mod*), 1865
retention_schedule() (*in module salt.states.file*), 3446
retrieve() (*in module salt.modules.statuspage*), 2563
retry_dns
  conf/minion, 127
retry_dns_count
  conf/minion, 127
return_retry_timer
  conf/minion, 140
return_retry_timer_max
  conf/minion, 140
Returner, **5402**
returner() (*in module salt.returners.carbon_return*), 313
returner() (*in module salt.returners.cassandra_cql_return*), 315
returner() (*in module salt.returners.cassandra_return*), 316
returner() (*in module salt.returners.couchbase_return*), 317
returner() (*in module salt.returners.couchdb_return*), 318
returner() (*in module salt.returners.django_return*), 318
returner() (*in module salt.returners.elasticsearch_return*), 320
returner() (*in module salt.returners.etcd_return*), 321
returner() (*in module salt.returners.highstate_return*), 323
returner() (*in module salt.returners.influxdb_return*), 324
returner() (*in module salt.returners.kafka_return*), 324
returner() (*in module salt.returners.librato_return*), 324
returner() (*in module salt.returners.local*), 325
returner() (*in module salt.returners.local_cache*), 325

`returner()` (*in module salt.returners.mattermost_returner*), 327
`returner()` (*in module salt.returners.memcache_return*), 327
`returner()` (*in module salt.returners.mongo_future_return*), 329
`returner()` (*in module salt.returners.mongo_return*), 330
`returner()` (*in module salt.returners.multi_returner*), 331
`returner()` (*in module salt.returners.mysql*), 333
`returner()` (*in module salt.returners.nagios_nrdp_return*), 334
`returner()` (*in module salt.returners.odbc*), 337
`returner()` (*in module salt.returners.pgjsonb*), 340
`returner()` (*in module salt.returners.postgres*), 342
`returner()` (*in module salt.returners.postgres_local_cache*), 344
`returner()` (*in module salt.returners.pushover_returner*), 346
`returner()` (*in module salt.returners.rawfile_json*), 346
`returner()` (*in module salt.returners.redis_return*), 348
`returner()` (*in module salt.returners.sentry_return*), 348
`returner()` (*in module salt.returners.slack_returner*), 350
`returner()` (*in module salt.returners.sms_return*), 350
`returner()` (*in module salt.returners.smtp_return*), 352
`returner()` (*in module salt.returners.splunk*), 352
`returner()` (*in module salt.returners.sqlite3_return*), 354
`returner()` (*in module salt.returners.syslog_return*), 355
`returner()` (*in module salt.returners.telegram_return*), 355
`returner()` (*in module salt.returners.xmpp_return*), 357
`returner()` (*in module salt.returners.zabbix_return*), 357
`returner_argspec()` (*in module salt.modules.sysmod*), 2586
`returner_dirs`
  `conf/minion`, 145
`returner_doc()` (*in module salt.modules.sysmod*), 2586
`reuse_snapshot()` (*salt.modules.inspectlib.collector.Inspector method*), 1875
`rev_parse()` (*in module salt.modules.git*), 1788
`revdep_rebuild()` (*in module salt.modules.gentoolkitmod*), 1757
`reverse_ip()` (*in module salt.modules.network*), 2220
`revert_snapshot()` (*in module salt.modules.parallels*), 2323
`revert_snapshot()` (*in module salt.modules.virt*), 2669
`revert_to_snapshot()` (*in module salt.cloud.clouds.vmware*), 1146
`reverted()` (*in module salt.states.virt*), 3751
`revision()` (*in module salt.modules.git*), 1789
`revision()` (*in module salt.modules.hg*), 1849
`revoke()` (*in module salt.modules.boto_secgroup*), 1443
`revoke_admin_privileges()` (*in module salt.modules.influxdbmod*), 1861
`revoke_auth()` (*in module salt.modules.saltutil*), 2472
`revoke_cache_security_group_ingress()` (*in module salt.modules.boto3_elasticache*), 1308
`revoke_cache_security_group_ingress()` (*in module salt.modules.boto_elasticache*), 1382
`revoke_cert()` (*in module salt.modules.tls*), 2617
`revoke_grant()` (*in module salt.modules.boto_kms*), 1418
`revoke_privilege()` (*in module salt.modules.influxdbmod*), 1861
`rgw_create()` (*in module salt.modules.ceph*), 1497
`rgw_destroy()` (*in module salt.modules.ceph*), 1497
`rgw_pools_create()` (*in module salt.modules.ceph*), 1498
`rgw_pools_missing()` (*in module salt.modules.ceph*), 1498
`ring()` (*in module salt.modules.cassandra*), 1482
`ring()` (*in module salt.states.ethtool*), 3409
`rm()` (*in module salt.modules.cron*), 1580
`rm()` (*in module salt.modules.dockercompose*), 1613
`rm()` (*in module salt.modules.incron*), 1856
`rm()` (*in module salt.modules.parted_partition*), 2327
`rm()` (*in module salt.states.etcd_mod*), 3407
`rm_()` (*in module salt.modules.dockermod*), 1650
`rm_()` (*in module salt.modules.etcd_mod*), 1697
`rm_()` (*in module salt.modules.git*), 1790
`rm_ace()` (*in module salt.modules.win_dacl*), 2723
`rm_alias()` (*in module salt.modules.aliases*), 1206
`rm_app()` (*in module salt.modules.marathon*), 2070
`rm_auth_key()` (*in module salt.modules.ssh*), 2540
`rm_auth_key_from_file()` (*in module salt.modules.ssh*), 2541
`rm_automaster()` (*in module salt.modules.mount*), 2093
`rm_dns()` (*in module salt.modules.win_dns_client*), 2729
`rm_env()` (*in module salt.modules.cron*), 1580
`rm_filesystems()` (*in module salt.modules.mount*), 2093
`rm_fstab()` (*in module salt.modules.mount*), 2093
`rm_host()` (*in module salt.modules.hosts*), 1850
`rm_job()` (*in module salt.modules.chronos*), 1506
`rm_job()` (*in module salt.modules.cron*), 1580
`rm_job()` (*in module salt.modules.incron*), 1856

`rm_known_host()` (*in module salt.modules.ssh*), 2541
`rm_pause()` (*in module salt.runners.state*), 3154
`rm_special()` (*in module salt.modules.cron*), 1580
`rm_token()` (*in module salt.tokens.localfs*), 3829
`rm_token()` (*in module salt.tokens.rediscluster*), 3829
`rm_vfstab()` (*in module salt.modules.mount*), 2093
`rmconfig()` (*in module salt.modules.freebsdports*), 1745
`rmdir()` (*in module salt.modules.file*), 1727
`rmi()` (*in module salt.modules.dockermod*), 1651
`role_absent()` (*in module salt.states.keystone*), 3519
`role_add()` (*in module salt.modules.rbac_solaris*), 2423
`role_assignment_list()` (*in module salt.modules.keystoneng*), 1940
`role_create()` (*in module salt.modules.keystone*), 1933
`role_create()` (*in module salt.modules.keystoneng*), 1941
`role_create()` (*in module salt.modules.mssql*), 2097
`role_delete()` (*in module salt.modules.keystone*), 1933
`role_delete()` (*in module salt.modules.keystoneng*), 1941
`role_exists()` (*in module salt.modules.boto_iam*), 1405
`role_exists()` (*in module salt.modules.mssql*), 2097
`role_get()` (*in module salt.modules.keystone*), 1933
`role_get()` (*in module salt.modules.keystoneng*), 1941
`role_get()` (*in module salt.modules.postgres*), 2382
`role_get()` (*in module salt.modules.rbac_solaris*), 2423
`role_grant()` (*in module salt.modules.keystoneng*), 1941
`role_list()` (*in module salt.modules.keystone*), 1933
`role_list()` (*in module salt.modules.keystoneng*), 1941
`role_list()` (*in module salt.modules.mssql*), 2097
`role_list()` (*in module salt.modules.rbac_solaris*), 2424
`role_present()` (*in module salt.states.keystone*), 3519
`role_remove()` (*in module salt.modules.mssql*), 2097
`role_revoke()` (*in module salt.modules.keystoneng*), 1941
`role_rm()` (*in module salt.modules.rbac_solaris*), 2424
`role_search()` (*in module salt.modules.keystoneng*), 1941
`role_update()` (*in module salt.modules.keystoneng*), 1942
`rollback()` (*in module salt.modules.freebsd_update*), 1739
`rollback()` (*in module salt.modules.junos*), 1917
`rollback()` (*in module salt.modules.napalm_network*), 2174
`rollback()` (*in module salt.modules.zfs*), 2904
`rollback()` (*in module salt.states.junos*), 3513
`rollback_snapshot()` (*in module salt.modules.smartos_vmadm*), 2507
`root_dir`
    `conf/master`, 53
    `conf/minion`, 130
`roots_update_interval`
    `conf/master`, 84
`Roster`, **5402**
`roster`
    `conf/master`, 65
`roster_defaults`
    `conf/master`, 65
`roster_file`
    `conf/master`, 65
`RosterMatcher` (*class in salt.roster.scan*), 3093
`RosterMatcher` (*class in salt.roster.sshconfig*), 3093
`rosters`
    `conf/master`, 65
`rotate()` (*in module salt.modules.logadm*), 1981
`rotate()` (*in module salt.states.logadm*), 3540
`rotate_aes_key`
    `conf/master`, 72
`route_absent()` (*in module salt.states.azurearm_network*), 3211
`route_create_or_update()` (*in module salt.modules.azurearm_network*), 1262
`route_delete()` (*in module salt.modules.azurearm_network*), 1262
`route_exists()` (*in module salt.modules.boto_vpc*), 1462
`route_filter_create_or_update()` (*in module salt.modules.azurearm_network*), 1262
`route_filter_delete()` (*in module salt.modules.azurearm_network*), 1263
`route_filter_get()` (*in module salt.modules.azurearm_network*), 1263
`route_filter_rule_create_or_update()` (*in module salt.modules.azurearm_network*), 1263
`route_filter_rule_delete()` (*in module salt.modules.azurearm_network*), 1263
`route_filter_rule_get()` (*in module salt.modules.azurearm_network*), 1264
`route_filter_rules_list()` (*in module salt.modules.azurearm_network*), 1264
`route_filters_list()` (*in module salt.modules.azurearm_network*), 1264
`route_filters_list_all()` (*in module salt.modules.azurearm_network*), 1264
`route_get()` (*in module salt.modules.azurearm_network*), 1265
`route_present()` (*in module salt.states.azurearm_network*), 3211
`route_table_absent()` (*in module salt.states.azurearm_network*), 3212

`route_table_absent()` (*in module salt.states.boto_vpc*), 3332
`route_table_create_or_update()` (*in module salt.modules.azurearm_network*), 1265
`route_table_delete()` (*in module salt.modules.azurearm_network*), 1265
`route_table_exists()` (*in module salt.modules.boto_vpc*), 1462
`route_table_get()` (*in module salt.modules.azurearm_network*), 1265
`route_table_present()` (*in module salt.states.azurearm_network*), 3212
`route_table_present()` (*in module salt.states.boto_vpc*), 3332
`route_tables_list()` (*in module salt.modules.azurearm_network*), 1266
`route_tables_list_all()` (*in module salt.modules.azurearm_network*), 1266
`routes()` (*in module salt.modules.network*), 2220
`routes()` (*in module salt.modules.osquery*), 2294
`routes()` (*in module salt.states.network*), 3601
`routes_list()` (*in module salt.modules.azurearm_network*), 1266
`row_absent()` (*in module salt.states.sqlite3*), 3722
`row_present()` (*in module salt.states.sqlite3*), 3723
`rpc()` (*in module salt.modules.junos*), 1918
`rpc()` (*in module salt.modules.napalm_mod*), 2137
`rpc()` (*in module salt.modules.nxos_api*), 2268
`rpc()` (*in module salt.proxy.nxos_api*), 3078
`rpc()` (*in module salt.states.junos*), 3513
`rpm_packages()` (*in module salt.modules.osquery*), 2294
`rr_absent()` (*in module salt.states.boto3_route53*), 3247
`rr_absent()` (*in module salt.states.boto_route53*), 3318
`rr_present()` (*in module salt.states.boto3_route53*), 3248
`rr_present()` (*in module salt.states.boto_route53*), 3318
`rsync()` (*in module salt.modules.rsync*), 2451
`rubygems()` (*in module salt.modules.rvm*), 2458
`rule_absent()` (*in module salt.states.csf*), 3359
`rule_exists()` (*in module salt.modules.win_firewall*), 2749
`rule_present()` (*in module salt.states.csf*), 3360
Run (*class in salt.netapi.rest_cherrypy.app*), 2949
`run()` (*in module salt.modules.boto_ec2*), 1371
`run()` (*in module salt.modules.cmdmod*), 1532
`run()` (*in module salt.modules.container_resource*), 1573
`run()` (*in module salt.modules.dockermod*), 1651
`run()` (*in module salt.modules.jenkinsmod*), 1910
`run()` (*in module salt.modules.lxc*), 1997
`run()` (*in module salt.modules.munin*), 2098
`run()` (*in module salt.modules.nagios*), 2111
`run()` (*in module salt.modules.nspawn*), 2263
`run()` (*in module salt.modules.puppet*), 2396
`run()` (*in module salt.modules.snapper*), 2546
`run()` (*in module salt.modules.win_task*), 2830
`run()` (*in module salt.states.cmd*), 3348
`run()` (*in module salt.states.docker_container*), 3370
`run()` (*in module salt.states.module*), 3559
`run()` (*in module salt.states.mysql_query*), 3568
`run()` (*in module salt.utils.extend*), 3929
`run()` (*salt.engines.redis_sentinel.Listener method*), 1164
`run_all()` (*in module salt.modules.cmdmod*), 1534
`run_all()` (*in module salt.modules.dockermod*), 1652
`run_all()` (*in module salt.modules.lxc*), 1998
`run_all()` (*in module salt.modules.munin*), 2099
`run_all()` (*in module salt.modules.nagios*), 2112
`run_all()` (*in module salt.modules.nspawn*), 2264
`run_all_pillar()` (*in module salt.modules.nagios*), 2112
`run_bg()` (*in module salt.modules.cmdmod*), 1537
`run_buildout()` (*in module salt.modules.zcbuildout*), 2897
`run_chroot()` (*in module salt.modules.cmdmod*), 1539
`run_command()` (*in module salt.modules.jboss7_cli*), 1907
`run_command_async()` (*salt.engines.slack.SlackClient method*), 1168
`run_commands()` (*in module salt.modules.arista_pyeapi*), 1238
`run_commands_from_slack_async()` (*salt.engines.slack.SlackClient method*), 1168
`run_config()` (*in module salt.modules.win_dsc*), 2731
`run_container()` (*in module salt.modules.dockermod*), 1652
`run_file()` (*in module salt.states.mysql_query*), 3568
`run_highstate_tests()` (*in module salt.modules.saltcheck*), 2468
`run_job()` (*in module salt.modules.schedule*), 2483
`run_job()` (*salt.client.LocalClient method*), 3848
`run_operation()` (*in module salt.modules.jboss7_cli*), 1907
`run_pillar()` (*in module salt.modules.nagios*), 2112
`run_query()` (*in module salt.cache.mysql_cache*), 1029
`run_query()` (*in module salt.modules.oracle*), 2287
`run_query()` (*in module salt.modules.zabbix*), 2887
`run_request()` (*in module salt.modules.state*), 2553
`run_state_tests()` (*in module salt.modules.saltcheck*), 2468
`run_state_tests_ssh()` (*in module salt.modules.saltcheck*), 2468
`run_stderr()` (*in module salt.modules.cmdmod*), 1541
`run_stderr()` (*in module salt.modules.dockermod*), 1653
`run_stderr()` (*in module salt.modules.lxc*), 1999
`run_stderr()` (*in module salt.modules.nspawn*), 2264

run_stdout() (*in module salt.modules.cmdmod*), 1544
run_stdout() (*in module salt.modules.dockermod*), 1653
run_stdout() (*in module salt.modules.lxc*), 1999
run_stdout() (*in module salt.modules.nspawn*), 2264
run_task() (*in module salt.modules.celery*), 1490
run_test() (*in module salt.modules.saltcheck*), 2468
run_wait() (*in module salt.modules.win_task*), 2830
Runner Function, **5402**
Runner Module, **5402**
runner() (*in module salt.modules.publish*), 2394
runner() (*in module salt.modules.saltutil*), 2472
runner() (*in module salt.runners.doc*), 3107
runner() (*in module salt.states.saltmod*), 3699
runner() (*salt.netapi.NetapiClient method*), 3855
runner_argspec() (*in module salt.modules.sysmod*), 2586
runner_async() (*salt.netapi.NetapiClient method*), 3855
runner_dirs
  conf/master, 75
runner_doc() (*in module salt.modules.sysmod*), 2586
RunnerClient (*class in salt.runner*), 3850
running() (*in module salt.modules.apf*), 1213
running() (*in module salt.modules.csf*), 1583
running() (*in module salt.modules.dummyproxy_service*), 1674
running() (*in module salt.modules.rest_service*), 2440
running() (*in module salt.modules.saltutil*), 2473
running() (*in module salt.modules.ssh_service*), 2543
running() (*in module salt.modules.state*), 2554
running() (*in module salt.states.docker_container*), 3371
running() (*in module salt.states.lxc*), 3547
running() (*in module salt.states.marathon_app*), 3551
running() (*in module salt.states.service*), 3712
running() (*in module salt.states.supervisord*), 3729
running() (*in module salt.states.vagrant*), 3746
running() (*in module salt.states.virt*), 3752
running_service_owners() (*in module salt.modules.introspect*), 1880
running_systemd() (*in module salt.modules.lxc*), 2000
RunSaltAPIHandler (*class in salt.netapi.rest_tornado.saltnado*), 2973

S

S3Credentials (*class in salt.pillar.s3*), 3031
s3fs_update_interval
  conf/master, 98
safari_extensions() (*in module salt.modules.osquery*), 2294
safe_accept() (*in module salt.runners.manage*), 3125
Salt Cloud, **5402**
salt command line option
  --args-separator=ARGS_SEPARATOR, 971
  --async, 970
  --force-color, 972
  --grain-pcre, 971
  --hide-timeout, 970
  --log-file-level=LOG_LEVEL_LOGFILE, 971
  --log-file=LOG_FILE, 971
  --no-color, 972
  --out, 972
  --out-file-append, --output-file-append, 972
  --out-file=OUTPUT_FILE, --output-file=OUTPUT_FILE, 972
  --out-indent OUTPUT_INDENT, --output-indent OUTPUT_INDENT, 972
  --return=RETURNER, 970
  --state-output=STATE_OUTPUT, --state_output=STATE_OUTPUT, 972
  --state-verbose=STATE_VERBOSE, --state_verbose=STATE_VERBOSE, 972
  --subset=SUBSET, 970
  --version, 970
  --versions-report, 970
  -C, --compound, 972
  -E, --pcre, 971
  -G, --grain, 971
  -I, --pillar, 972
  -L, --list, 971
  -N, --nodegroup, 971
  -R, --range, 971
  -S, --ipcidr, 972
  -T, --make-token, 970
  -a EAUTH, --auth=EAUTH, 970
  -b BATCH, --batch-size=BATCH, 970
  -c CONFIG_DIR, --config-dir=CONFIG_dir, 970
  -d, --doc, --documentation, 971
  -h, --help, 970
  -l LOG_LEVEL, --log-level=LOG_LEVEL, 971
  -s, --static, 970
  -t TIMEOUT, --timeout=TIMEOUT, 970
  -v VERBOSE, --verbose, 970
Salt Mine, 417
Salt Reactor, 684
Salt SSH, **5403**
Salt Thin, **5403**
Salt Virt, **5403**
salt-api command line option

- --log-file-level=LOG_LEVEL_LOGFILE, 990
- --log-file=LOG_FILE, 990
- --pid-file=PIDFILE, 990
- --version, 990
- --versions-report, 990
- -c CONFIG_DIR, --config-dir=CONFIG_dir, 990
- -d, --daemon, 990
- -h, --help, 990
- -l LOG_LEVEL, --log-level=LOG_LEVEL, 990

salt-call command line option
- --file-root=FILE_ROOT, 968
- --force-color, 969
- --hard-crash, 967
- --id=ID, 968
- --local, 968
- --log-file-level=LOG_LEVEL_LOGFILE, 968
- --log-file=LOG_FILE, 968
- --master=MASTER, 968
- --metadata, 968
- --no-color, 969
- --out, 969
- --out-file-append, --output-file-append, 969
- --out-file=OUTPUT_FILE, --output-file=OUTPUT_FILE, 969
- --out-indent OUTPUT_INDENT, --output-indent OUTPUT_INDENT, 969
- --pillar-root=PILLAR_ROOT, 968
- --refresh-grains-cache, 968
- --retcode-passthrough, 968
- --return RETURNER, 968
- --skip-grains, 968
- --state-output=STATE_OUTPUT, --state_output=STATE_OUTPUT, 969
- --state-verbose=STATE_VERBOSE, --state_verbose=STATE_VERBOSE, 969
- --version, 967
- --versions-report, 967
- -c CONFIG_DIR, --config-dir=CONFIG_dir, 967
- -d, --doc, --documentation, 968
- -g, --grains, 967
- -h, --help, 967
- -l LOG_LEVEL, --log-level=LOG_LEVEL, 968
- -m MODULE_DIRS, --module-dirs=MODULE_DIRS, 967

salt-cloud command line option
- --force-color, 725
- --list-images=LIST_IMAGES, 724
- --list-locations=LIST_LOCATIONS, 724
- --list-profiles, 724
- --list-providers, 724
- --list-sizes=LIST_SIZES, 724
- --no-color, 725
- --out, 724
- --out-file-append, --output-file-append, 724
- --out-file=OUTPUT_FILE, --output-file=OUTPUT_FILE, 724
- --out-indent OUTPUT_INDENT, --output-indent OUTPUT_INDENT, 724
- --script-args=SCRIPT_ARGS, 723
- --set-password=<USERNAME> <PROVIDER>, 724
- --show-deploy-args, 723
- --state-output=STATE_OUTPUT, --state_output=STATE_OUTPUT, 725
- --state-verbose=STATE_VERBOSE, --state_verbose=STATE_VERBOSE, 725
- --version, 722
- --versions-report, 722
- -F, --full-query, 723
- -H, --hard, 723
- -L LOCATION, --location=LOCATION, 723
- -P, --parallel, 723
- -Q, --query, 723
- -S, --select-query, 724
- -a ACTION, --action=ACTION, 723
- -c CONFIG_DIR, --config-dir=CONFIG_dir, 722
- -d, --destroy, 723
- -f <FUNC-NAME> <PROVIDER>, --function=<FUNC-NAME> <PROVIDER>, 723
- -h, --help, 722
- -k, --keep-tmp, 723
- -m MAP, --map=MAP, 723
- -p PROFILE, --profile=PROFILE, 723
- -u, --update-bootstrap, 723
- -y, --assume-yes, 723

salt-cp command line option
- --grain-pcre, 974
- --log-file-level=LOG_LEVEL_LOGFILE, 974
- --log-file=LOG_FILE, 974
- --version, 974
- --versions-report, 974

- -C, --chunked, 975
- -E, --pcre, 974
- -G, --grain, 974
- -L, --list, 974
- -N, --nodegroup, 974
- -R, --range, 975
- -c CONFIG_DIR, --config-dir=CONFIG_dir, 974
- -h, --help, 974
- -l LOG_LEVEL, --log-level=LOG_LEVEL, 974
- -n, --no-compression, 975
- -t TIMEOUT, --timeout=TIMEOUT, 974

salt-extend command line option
- --debug, 976
- --description, -d, 976
- --extension, -e, 976
- --name, -n, 976
- --no-merge, 976
- --salt-directory, -o, 976

salt-key command line option
- --auto-create, 979
- --force-color, 978
- --gen-keys-dir=GEN_KEYS_DIR, 979
- --gen-keys=GEN_KEYS, 979
- --gen-signature, 979
- --hard-crash, 977
- --include-all, 978
- --keysize=KEYSIZE, 979
- --log-file-level=LOG_LEVEL_LOGFILE, 977
- --log-file=LOG_FILE, 977
- --no-color, 978
- --out, 977
- --out-file-append, --output-file-append, 978
- --out-file=OUTPUT_FILE, --output-file=OUTPUT_FILE, 977
- --out-indent OUTPUT_INDENT, --output-indent OUTPUT_INDENT, 977
- --priv=PRIV, 979
- --pub=PUB, 979
- --rotate-aes-key=ROTATE_AES_KEY, 977
- --signature-path=SIGNATURE_PATH, 979
- --state-output=STATE_OUTPUT, --state_output=STATE_OUTPUT, 978
- --state-verbose=STATE_VERBOSE, --state_verbose=STATE_VERBOSE, 978
- --version, 977
- --versions-report, 977
- -A, --accept-all, 978
- -D, --delete-all, 978
- -F, --finger-all, 979
- -L, --list-all, 978
- -P, --print-all, 978
- -R, --reject-all, 978
- -a ACCEPT, --accept=ACCEPT, 978
- -c CONFIG_DIR, --config-dir=CONFIG_dir, 977
- -d DELETE, --delete=DELETE, 978
- -f FINGER, --finger=FINGER, 979
- -h, --help, 977
- -l ARG, --list=ARG, 978
- -p PRINT, --print=PRINT, 978
- -q, --quiet, 977
- -r REJECT, --reject=REJECT, 978
- -u USER, --user=USER, 977
- -y, --yes, 977

salt-master command line option
- --log-file-level=LOG_LEVEL_LOGFILE, 980
- --log-file=LOG_FILE, 980
- --pid-file PIDFILE, 980
- --version, 980
- --versions-report, 980
- -c CONFIG_DIR, --config-dir=CONFIG_dir, 980
- -d, --daemon, 980
- -h, --help, 980
- -l LOG_LEVEL, --log-level=LOG_LEVEL, 980
- -u USER, --user=USER, 980

salt-minion command line option
- --log-file-level=LOG_LEVEL_LOGFILE, 981
- --log-file=LOG_FILE, 981
- --pid-file PIDFILE, 981
- --version, 981
- --versions-report, 981
- -c CONFIG_DIR, --config-dir=CONFIG_dir, 981
- -d, --daemon, 981
- -h, --help, 981
- -l LOG_LEVEL, --log-level=LOG_LEVEL, 981
- -u USER, --user=USER, 981

salt-proxy command line option
- --log-file-level=LOG_LEVEL_LOGFILE, 982
- --log-file=LOG_FILE, 982
- --pid-file PIDFILE, 982
- --proxyid, 982
- --version, 982
- --versions-report, 982

- -c CONFIG_DIR, --config-dir=CONFIG_dir, 982
- -d, --daemon, 982
- -h, --help, 982
- -l LOG_LEVEL, --log-level=LOG_LEVEL, 982
- -u USER, --user=USER, 982

salt-run command line option
- --hard-crash, 983
- --log-file-level=LOG_LEVEL_LOGFILE, 984
- --log-file=LOG_FILE, 984
- --version, 983
- --versions-report, 983
- -c CONFIG_DIR, --config-dir=CONFIG_dir, 983
- -d, --doc, --documentation, 983
- -h, --help, 983
- -l LOG_LEVEL, --log-level=LOG_LEVEL, 984
- -t TIMEOUT, --timeout=TIMEOUT, 983

salt-ssh command line option
- --askpass, 986
- --extra-filerefs=EXTRA_FILEREFS, 985
- --force-color, 987
- --hard-crash, 984
- --identities-only, 986
- --jid=JID, 985
- --key-deploy, 986
- --log-file-level=LOG_LEVEL_LOGFILE, 986
- --log-file=LOG_FILE, 986
- --max-procs, 985
- --min-extra-modules=MIN_EXTRA_MODS, 985
- --no-color, 987
- --no-host-keys, 986
- --out, 987
- --out-file-append, --output-file-append, 987
- --out-file=OUTPUT_FILE, --output-file=OUTPUT_FILE, 987
- --out-indent OUTPUT_INDENT, --output-indent OUTPUT_INDENT, 987
- --passwd, 986
- --priv-passwd=SSH_PRIV_PASSWD, 985
- --priv=SSH_PRIV, 985
- --python2-bin=PYTHON2_BIN, 985
- --python3-bin=PYTHON3_BIN, 985
- --refresh, --refresh-cache, 985
- --roster, 984
- --roster-file, 985
- --scan-ports=SSH_SCAN_PORTS, 986
- --scan-timeout=SSH_SCAN_TIMEOUT, 986
- --state-output=STATE_OUTPUT, --state_output=STATE_OUTPUT, 987
- --state-verbose=STATE_VERBOSE, --state_verbose=STATE_VERBOSE, 987
- --sudo, 986
- --thin-extra-modules=THIN_EXTRA_MODS, 985
- --user=SSH_USER, 986
- --version, 984
- --versions-report, 984
- -E, --pcre, 986
- -W, --rand-thin-dir, 985
- -c CONFIG_DIR, --config-dir=CONFIG_dir, 984
- -h, --help, 984
- -i, --ignore-host-keys, 985
- -l LOG_LEVEL, --log-level=LOG_LEVEL, 986
- -r, --raw, --raw-shell, 984
- -s, --static, 985
- -t, --regen-thin, --thin, 985
- -v, --verbose, 985
- -w, --wipe, 985

salt-syndic command line option
- --log-file-level=LOG_LEVEL_LOGFILE, 988
- --log-file=LOG_FILE, 988
- --pid-file PIDFILE, 988
- --version, 988
- --versions-report, 988
- -c CONFIG_DIR, --config-dir=CONFIG_dir, 988
- -d, --daemon, 988
- -h, --help, 988
- -l LOG_LEVEL, --log-level=LOG_LEVEL, 988
- -u USER, --user=USER, 988

salt.auth.auto (*module*), 997

salt.auth.django (*module*), 997

salt.auth.file (*module*), 999

salt.auth.keystone (*module*), 1000

salt.auth.ldap (*module*), 1001

salt.auth.mysql (*module*), 1001

salt.auth.pam (*module*), 1002

salt.auth.pki (*module*), 1003

salt.auth.rest (*module*), 1004

salt.auth.sharedsecret (*module*), 1004

salt.auth.yubico (*module*), 1005

salt.beacons.adb (*module*), 1006

salt.beacons.avahi_announce (*module*), 1007

salt.beacons.bonjour_announce (*module*), 1008
salt.beacons.btmp (*module*), 1009
salt.beacons.diskusage (*module*), 1010
salt.beacons.glxinfo (*module*), 1011
salt.beacons.haproxy (*module*), 1011
salt.beacons.inotify (*module*), 1012
salt.beacons.journald (*module*), 1013
salt.beacons.load (*module*), 1013
salt.beacons.log (*module*), 1014
salt.beacons.memusage (*module*), 1014
salt.beacons.napalm_beacon (*module*), 1014
salt.beacons.network_info (*module*), 1017
salt.beacons.network_settings (*module*), 1018
salt.beacons.pkg (*module*), 1018
salt.beacons.proxy_example (*module*), 1019
salt.beacons.ps (*module*), 1019
salt.beacons.salt_proxy (*module*), 1019
salt.beacons.sensehat (*module*), 1020
salt.beacons.service (*module*), 1020
salt.beacons.sh (*module*), 1021
salt.beacons.status (*module*), 1021
salt.beacons.telegram_bot_msg (*module*), 1023
salt.beacons.twilio_txt_msg (*module*), 1023
salt.beacons.wtmp (*module*), 1024
salt.cache.consul (*module*), 1026
salt.cache.etcd_cache (*module*), 1026
salt.cache.localfs (*module*), 1027
salt.cache.mysql_cache (*module*), 1028
salt.cache.redis_cache (*module*), 1029
salt.cloud.clouds.aliyun (*module*), 1032
salt.cloud.clouds.azurearm (*module*), 1034
salt.cloud.clouds.cloudstack (*module*), 1038
salt.cloud.clouds.digitalocean (*module*), 1039
salt.cloud.clouds.dimensiondata (*module*), 1043
salt.cloud.clouds.ec2 (*module*), 1044
salt.cloud.clouds.gce (*module*), 1053
salt.cloud.clouds.gogrid (*module*), 1059
salt.cloud.clouds.joyent (*module*), 1061
salt.cloud.clouds.linode (*module*), 1065
salt.cloud.clouds.lxc (*module*), 1070
salt.cloud.clouds.msazure (*module*), 1071
salt.cloud.clouds.oneandone (*module*), 1087
salt.cloud.clouds.opennebula (*module*), 1090
salt.cloud.clouds.openstack (*module*), 843, 1109
salt.cloud.clouds.parallels (*module*), 1114
salt.cloud.clouds.profitbricks (*module*), 1115
salt.cloud.clouds.proxmox (*module*), 1119
salt.cloud.clouds.pyrax (*module*), 1121
salt.cloud.clouds.qingcloud (*module*), 1122
salt.cloud.clouds.saltify (*module*), 1124
salt.cloud.clouds.scaleway (*module*), 1126
salt.cloud.clouds.softlayer (*module*), 1127
salt.cloud.clouds.softlayer_hw (*module*), 1128
salt.cloud.clouds.tencentcloud (*module*), 1130
salt.cloud.clouds.vagrant (*module*), 1133
salt.cloud.clouds.virtualbox (*module*), 1134
salt.cloud.clouds.vmware (*module*), 1136
salt.cloud.clouds.vultrpy (*module*), 1147
salt.cloud.clouds.xen (*module*), 1149
salt.engines.docker_events (*module*), 1153
salt.engines.fluent (*module*), 1154
salt.engines.http_logstash (*module*), 1154
salt.engines.ircbot (*module*), 1155
salt.engines.junos_syslog (*module*), 1157
salt.engines.logentries (*module*), 1159
salt.engines.logstash_engine (*module*), 1160
salt.engines.napalm_syslog (*module*), 1160
salt.engines.reactor (*module*), 1163
salt.engines.redis_sentinel (*module*), 1163
salt.engines.slack (*module*), 1164
salt.engines.sqs_events (*module*), 1168
salt.engines.stalekey (*module*), 1169
salt.engines.test (*module*), 1170
salt.engines.thorium (*module*), 1170
salt.engines.webhook (*module*), 1170
salt.exceptions (*module*), 3998
salt.executors.direct_call (*module*), 1171
salt.executors.docker (*module*), 1171
salt.executors.splay (*module*), 1171
salt.executors.sudo (*module*), 1172
salt.fileserver.azurefs (*module*), 1172
salt.fileserver.gitfs (*module*), 1173
salt.fileserver.hgfs (*module*), 1174
salt.fileserver.minionfs (*module*), 1174
salt.fileserver.roots (*module*), 1174
salt.fileserver.s3fs (*module*), 1175
salt.fileserver.svnfs (*module*), 1176
salt.grains.chronos (*module*), 1177
salt.grains.cimc (*module*), 1177
salt.grains.core (*module*), 1177
salt.grains.disks (*module*), 1179
salt.grains.esxi (*module*), 1179
salt.grains.extra (*module*), 1179
salt.grains.fx2 (*module*), 1180
salt.grains.junos (*module*), 1180
salt.grains.marathon (*module*), 1180
salt.grains.mdadm (*module*), 1180
salt.grains.metadata (*module*), 1180
salt.grains.napalm (*module*), 1181

salt.grains.opts (*module*), 1185
salt.grains.panos (*module*), 1185
salt.grains.philips_hue (*module*), 1185
salt.grains.rest_sample (*module*), 1185
salt.log.handlers.fluent_mod (*module*), 248
salt.log.handlers.log4mongo_mod (*module*), 249
salt.log.handlers.logstash_mod (*module*), 250
salt.log.handlers.sentry_mod (*module*), 252
salt.modules.acme (*module*), 1200
salt.modules.aix_group (*module*), 1203
salt.modules.aixpkg (*module*), 1204
salt.modules.aliases (*module*), 1206
salt.modules.alternatives (*module*), 1206
salt.modules.ansiblegate (*module*), 1208
salt.modules.apache (*module*), 1209
salt.modules.apcups (*module*), 1211
salt.modules.apf (*module*), 1212
salt.modules.apkpkg (*module*), 1213
salt.modules.aptpkg (*module*), 1216
salt.modules.archive (*module*), 1226
salt.modules.arista_pyeapi (*module*), 1234
salt.modules.artifactory (*module*), 1240
salt.modules.at (*module*), 1242
salt.modules.at_solaris (*module*), 1243
salt.modules.augeas_cfg (*module*), 1244
salt.modules.aws_sqs (*module*), 1246
salt.modules.azurearm_compute (*module*), 1248
salt.modules.azurearm_network (*module*), 1253
salt.modules.azurearm_resource (*module*), 1271
salt.modules.bamboohr (*module*), 1279
salt.modules.bcache (*module*), 1280
salt.modules.beacons (*module*), 1283
salt.modules.bigip (*module*), 1285
salt.modules.bluez_bluetooth (*module*), 1299
salt.modules.boto3_elasticache (*module*), 1301
salt.modules.boto3_elasticsearch (*module*), 1308
salt.modules.boto3_route53 (*module*), 1322
salt.modules.boto_apigateway (*module*), 1329
salt.modules.boto_asg (*module*), 1341
salt.modules.boto_cfn (*module*), 1344
salt.modules.boto_cloudfront (*module*), 1346
salt.modules.boto_cloudtrail (*module*), 1348
salt.modules.boto_cloudwatch (*module*), 1351
salt.modules.boto_cloudwatch_event (*module*), 1353
salt.modules.boto_cognitoidentity (*module*), 1355
salt.modules.boto_datapipeline (*module*), 1358
salt.modules.boto_dynamodb (*module*), 1359
salt.modules.boto_ec2 (*module*), 1361
salt.modules.boto_efs (*module*), 1375
salt.modules.boto_elasticache (*module*), 1378
salt.modules.boto_elasticsearch_domain (*module*), 1382
salt.modules.boto_elb (*module*), 1386
salt.modules.boto_elbv2 (*module*), 1391
salt.modules.boto_iam (*module*), 1393
salt.modules.boto_iot (*module*), 1406
salt.modules.boto_kinesis (*module*), 1412
salt.modules.boto_kms (*module*), 1415
salt.modules.boto_lambda (*module*), 1418
salt.modules.boto_rds (*module*), 1424
salt.modules.boto_route53 (*module*), 1429
salt.modules.boto_s3_bucket (*module*), 1435
salt.modules.boto_secgroup (*module*), 1441
salt.modules.boto_sns (*module*), 1444
salt.modules.boto_sqs (*module*), 1445
salt.modules.boto_vpc (*module*), 1447
salt.modules.bower (*module*), 1462
salt.modules.bridge (*module*), 1464
salt.modules.bsd_shadow (*module*), 1465
salt.modules.btrfs (*module*), 1466
salt.modules.cabal (*module*), 1471
salt.modules.capirca_acl (*module*), 1472
salt.modules.cassandra (*module*), 1481
salt.modules.cassandra_cql (*module*), 1482
salt.modules.celery (*module*), 1490
salt.modules.ceph (*module*), 1491
salt.modules.chassis (*module*), 1498
salt.modules.chef (*module*), 1499
salt.modules.chocolatey (*module*), 1500
salt.modules.chronos (*module*), 1506
salt.modules.cimc (*module*), 1507
salt.modules.ciscoconfparse_mod (*module*), 1514
salt.modules.cisconso (*module*), 1518
salt.modules.cloud (*module*), 1519
salt.modules.cmdmod (*module*), 1523
salt.modules.composer (*module*), 1554
salt.modules.config (*module*), 1556
salt.modules.consul (*module*), 1560
salt.modules.container_resource (*module*), 1573
salt.modules.cp (*module*), 1574
salt.modules.cpan (*module*), 1579
salt.modules.cron (*module*), 1579
salt.modules.csf (*module*), 1581
salt.modules.cyg (*module*), 1583
salt.modules.daemontools (*module*), 1585

salt.modules.data (*module*), 1586
salt.modules.datadog_api (*module*), 1588
salt.modules.ddns (*module*), 1589
salt.modules.deb_apache (*module*), 1590
salt.modules.deb_postgres (*module*), 1592
salt.modules.debconfmod (*module*), 1595
salt.modules.debian_ip (*module*), 1596
salt.modules.debian_service (*module*), 1597
salt.modules.debuild_pkgbuild (*module*), 1593
salt.modules.defaults (*module*), 1599
salt.modules.devmap (*module*), 1601
salt.modules.dig (*module*), 1601
salt.modules.disk (*module*), 1602
salt.modules.djangomod (*module*), 1605
salt.modules.dnsmasq (*module*), 1607
salt.modules.dnsutil (*module*), 1608
salt.modules.dockercompose (*module*), 1609
salt.modules.dockermod (*module*), 1615
salt.modules.dpkg_lowpkg (*module*), 1661
salt.modules.drac (*module*), 1662
salt.modules.dracr (*module*), 1665
salt.modules.drbd (*module*), 1673
salt.modules.dummyproxy_pkg (*module*), 1673
salt.modules.dummyproxy_service (*module*), 1674
salt.modules.ebuildpkg (*module*), 1675
salt.modules.eix (*module*), 1680
salt.modules.elasticsearch (*module*), 1681
salt.modules.environ (*module*), 1693
salt.modules.eselect (*module*), 1694
salt.modules.esxi (*module*), 1696
salt.modules.etcd_mod (*module*), 1696
salt.modules.ethtool (*module*), 1698
salt.modules.event (*module*), 1700
salt.modules.extfs (*module*), 1701
salt.modules.file (*module*), 1703
salt.modules.firewalld (*module*), 1731
salt.modules.freebsd_sysctl (*module*), 1738
salt.modules.freebsd_update (*module*), 1738
salt.modules.freebsdjail (*module*), 1739
salt.modules.freebsdkmod (*module*), 1740
salt.modules.freebsdpkg (*module*), 1741
salt.modules.freebsdports (*module*), 1744
salt.modules.freebsdservice (*module*), 1746
salt.modules.gem (*module*), 1749
salt.modules.genesis (*module*), 1752
salt.modules.gentoo_service (*module*), 1754
salt.modules.gentoolkitmod (*module*), 1756
salt.modules.git (*module*), 1757
salt.modules.github (*module*), 1799
salt.modules.glance (*module*), 1808
salt.modules.glanceng (*module*), 1810
salt.modules.glusterfs (*module*), 1811
salt.modules.gnomedesktop (*module*), 1815
salt.modules.google_chat (*module*), 1817
salt.modules.gpg (*module*), 1817
salt.modules.grafana4 (*module*), 1823
salt.modules.grains (*module*), 1831
salt.modules.group (*module*), 1186
salt.modules.groupadd (*module*), 1836
salt.modules.grub_legacy (*module*), 1838
salt.modules.guestfs (*module*), 1838
salt.modules.hadoop (*module*), 1839
salt.modules.haproxyconn (*module*), 1840
salt.modules.hashutil (*module*), 1842
salt.modules.heat (*module*), 1845
salt.modules.hg (*module*), 1847
salt.modules.hosts (*module*), 1849
salt.modules.http (*module*), 1850
salt.modules.icinga2 (*module*), 1855
salt.modules.ifttt (*module*), 1852
salt.modules.ilo (*module*), 1852
salt.modules.incron (*module*), 1856
salt.modules.influxdb08mod (*module*), 1862
salt.modules.influxdbmod (*module*), 1857
salt.modules.infoblox (*module*), 1867
salt.modules.ini_manage (*module*), 1872
salt.modules.inspectlib (*module*), 1876
salt.modules.inspectlib.collector (*module*), 1874
salt.modules.inspectlib.dbhandle (*module*), 1875
salt.modules.inspectlib.entities (*module*), 1876
salt.modules.inspectlib.exceptions (*module*), 1875
salt.modules.inspectlib.fsdb (*module*), 1876
salt.modules.inspectlib.kiwiproc (*module*), 1878
salt.modules.inspectlib.query (*module*), 1876
salt.modules.inspector (*module*), 1878
salt.modules.introspect (*module*), 1880
salt.modules.iosconfig (*module*), 1881
salt.modules.ipmi (*module*), 1883
salt.modules.ipset (*module*), 1896
salt.modules.iptables (*module*), 1898
salt.modules.iwtools (*module*), 1902
salt.modules.jboss7 (*module*), 1903
salt.modules.jboss7_cli (*module*), 1907
salt.modules.jenkinsmod (*module*), 1908
salt.modules.jinja (*module*), 1911
salt.modules.jira_mod (*module*), 1911
salt.modules.junos (*module*), 1913
salt.modules.k8s (*module*), 1919
salt.modules.kapacitor (*module*), 1922
salt.modules.kerberos (*module*), 1923

`salt.modules.kernelpkg` (*module*), 1186
`salt.modules.kernelpkg_linux_apt` (*module*), 1925
`salt.modules.kernelpkg_linux_yum` (*module*), 1926
`salt.modules.key` (*module*), 1928
`salt.modules.keyboard` (*module*), 1928
`salt.modules.keystone` (*module*), 1929
`salt.modules.keystoneng` (*module*), 1936
`salt.modules.keystore` (*module*), 1944
`salt.modules.kmod` (*module*), 1945
`salt.modules.kubernetesmod` (*module*), 1946
`salt.modules.launchctl_service` (*module*), 1951
`salt.modules.layman` (*module*), 1952
`salt.modules.ldap3` (*module*), 1953
`salt.modules.ldapmod` (*module*), 1958
`salt.modules.libcloud_compute` (*module*), 1959
`salt.modules.libcloud_dns` (*module*), 1965
`salt.modules.libcloud_loadbalancer` (*module*), 1968
`salt.modules.libcloud_storage` (*module*), 1971
`salt.modules.linux_acl` (*module*), 1975
`salt.modules.linux_ip` (*module*), 1976
`salt.modules.linux_lvm` (*module*), 1976
`salt.modules.linux_sysctl` (*module*), 1979
`salt.modules.localemod` (*module*), 1979
`salt.modules.locate` (*module*), 1980
`salt.modules.logadm` (*module*), 1981
`salt.modules.logmod` (*module*), 1982
`salt.modules.logrotate` (*module*), 1983
`salt.modules.lvs` (*module*), 1984
`salt.modules.lxc` (*module*), 1986
`salt.modules.lxd` (*module*), 2003
`salt.modules.mac_assistive` (*module*), 2022
`salt.modules.mac_brew_pkg` (*module*), 2023
`salt.modules.mac_desktop` (*module*), 2026
`salt.modules.mac_group` (*module*), 2027
`salt.modules.mac_keychain` (*module*), 2028
`salt.modules.mac_pkgutil` (*module*), 2031
`salt.modules.mac_portspkg` (*module*), 2032
`salt.modules.mac_power` (*module*), 2034
`salt.modules.mac_service` (*module*), 2038
`salt.modules.mac_shadow` (*module*), 2041
`salt.modules.mac_softwareupdate` (*module*), 2045
`salt.modules.mac_sysctl` (*module*), 2048
`salt.modules.mac_system` (*module*), 2049
`salt.modules.mac_timezone` (*module*), 2054
`salt.modules.mac_user` (*module*), 2056
`salt.modules.mac_xattr` (*module*), 2059
`salt.modules.macdefaults` (*module*), 2025
`salt.modules.macpackage` (*module*), 2029
`salt.modules.makeconf` (*module*), 2060
`salt.modules.mandrill` (*module*), 2068
`salt.modules.marathon` (*module*), 2069
`salt.modules.match` (*module*), 2070
`salt.modules.mattermost` (*module*), 2073
`salt.modules.mdadm_raid` (*module*), 2074
`salt.modules.mdata` (*module*), 2076
`salt.modules.memcached` (*module*), 2077
`salt.modules.mine` (*module*), 2078
`salt.modules.minion` (*module*), 2080
`salt.modules.mod_random` (*module*), 2082
`salt.modules.modjk` (*module*), 2083
`salt.modules.mongodb` (*module*), 2086
`salt.modules.monit` (*module*), 2089
`salt.modules.moosefs` (*module*), 2091
`salt.modules.mount` (*module*), 2091
`salt.modules.mssql` (*module*), 2095
`salt.modules.msteams` (*module*), 2098
`salt.modules.munin` (*module*), 2098
`salt.modules.mysql` (*module*), 2099
`salt.modules.nacl` (*module*), 2107
`salt.modules.nagios` (*module*), 2111
`salt.modules.nagios_rpc` (*module*), 2112
`salt.modules.namecheap_domains` (*module*), 2115
`salt.modules.namecheap_domains_dns` (*module*), 2113
`salt.modules.namecheap_domains_ns` (*module*), 2116
`salt.modules.namecheap_ssl` (*module*), 2118
`salt.modules.namecheap_users` (*module*), 2124
`salt.modules.napalm_bgp` (*module*), 2148
`salt.modules.napalm_formula` (*module*), 2152
`salt.modules.napalm_mod` (*module*), 2125
`salt.modules.napalm_netacl` (*module*), 2139
`salt.modules.napalm_network` (*module*), 2155
`salt.modules.napalm_ntp` (*module*), 2175
`salt.modules.napalm_probes` (*module*), 2178
`salt.modules.napalm_route` (*module*), 2183
`salt.modules.napalm_snmp` (*module*), 2184
`salt.modules.napalm_users` (*module*), 2186
`salt.modules.napalm_yang_mod` (*module*), 2188
`salt.modules.netaddress` (*module*), 2195
`salt.modules.netbox` (*module*), 2196
`salt.modules.netbsd_sysctl` (*module*), 2202
`salt.modules.netbsdservice` (*module*), 2202
`salt.modules.netmiko_mod` (*module*), 2205
`salt.modules.netscaler` (*module*), 2211
`salt.modules.network` (*module*), 2215
`salt.modules.neutron` (*module*), 2221
`salt.modules.neutronng` (*module*), 2238
`salt.modules.nfs3` (*module*), 2243
`salt.modules.nftables` (*module*), 2243

`salt.modules.nginx` (*module*), 2248
`salt.modules.nilrt_ip` (*module*), 2248
`salt.modules.nix` (*module*), 2251
`salt.modules.nova` (*module*), 2252
`salt.modules.npm` (*module*), 2257
`salt.modules.nspawn` (*module*), 2259
`salt.modules.nxos` (*module*), 2265
`salt.modules.nxos_api` (*module*), 2266
`salt.modules.omapi` (*module*), 2269
`salt.modules.openbsd_sysctl` (*module*), 2270
`salt.modules.openbsdpkg` (*module*), 2270
`salt.modules.openbsdrcctl_service` (*module*), 2272
`salt.modules.openbsdservice` (*module*), 2274
`salt.modules.openscap` (*module*), 2276
`salt.modules.openstack_config` (*module*), 2276
`salt.modules.openstack_mng` (*module*), 2277
`salt.modules.openvswitch` (*module*), 2278
`salt.modules.opkg` (*module*), 2280
`salt.modules.opsgenie` (*module*), 2286
`salt.modules.oracle` (*module*), 2286
`salt.modules.osquery` (*module*), 2288
`salt.modules.out` (*module*), 2296
`salt.modules.pacmanpkg` (*module*), 2297
`salt.modules.pagerduty` (*module*), 2302
`salt.modules.pagerduty_util` (*module*), 2303
`salt.modules.pam` (*module*), 2305
`salt.modules.panos` (*module*), 2305
`salt.modules.parallels` (*module*), 2320
`salt.modules.parted_partition` (*module*), 2324
`salt.modules.pcs` (*module*), 2328
`salt.modules.pdbedit` (*module*), 2331
`salt.modules.pecl` (*module*), 2333
`salt.modules.peeringdb` (*module*), 2333
`salt.modules.pf` (*module*), 2336
`salt.modules.philips_hue` (*module*), 2338
`salt.modules.pillar` (*module*), 2338
`salt.modules.pip` (*module*), 2344
`salt.modules.pkg` (*module*), 1186
`salt.modules.pkg_resource` (*module*), 2349
`salt.modules.pkgin` (*module*), 2351
`salt.modules.pkgng` (*module*), 2354
`salt.modules.pkgutil` (*module*), 2368
`salt.modules.portage_config` (*module*), 2370
`salt.modules.postfix` (*module*), 2372
`salt.modules.postgres` (*module*), 2374
`salt.modules.poudriere` (*module*), 2385
`salt.modules.powerpath` (*module*), 2387
`salt.modules.proxy` (*module*), 2387
`salt.modules.ps` (*module*), 2389
`salt.modules.publish` (*module*), 2393
`salt.modules.puppet` (*module*), 2395
`salt.modules.purefa` (*module*), 2396
`salt.modules.purefb` (*module*), 2402
`salt.modules.pushbullet` (*module*), 2404
`salt.modules.pushover_notify` (*module*), 2405
`salt.modules.pw_group` (*module*), 2405
`salt.modules.pw_user` (*module*), 2406
`salt.modules.pyenv` (*module*), 2409
`salt.modules.qemu_img` (*module*), 2410
`salt.modules.qemu_nbd` (*module*), 2411
`salt.modules.quota` (*module*), 2411
`salt.modules.rabbitmq` (*module*), 2412
`salt.modules.rallydev` (*module*), 2418
`salt.modules.random_org` (*module*), 2420
`salt.modules.rbac_solaris` (*module*), 2422
`salt.modules.rbenv` (*module*), 2424
`salt.modules.rdp` (*module*), 2426
`salt.modules.redismod` (*module*), 2427
`salt.modules.reg` (*module*), 2433
`salt.modules.rest_pkg` (*module*), 2438
`salt.modules.rest_sample_utils` (*module*), 2439
`salt.modules.rest_service` (*module*), 2439
`salt.modules.restartcheck` (*module*), 2440
`salt.modules.ret` (*module*), 2441
`salt.modules.rh_ip` (*module*), 2441
`salt.modules.rh_service` (*module*), 2443
`salt.modules.riak` (*module*), 2445
`salt.modules.rpm_lowpkg` (*module*), 2447
`salt.modules.rpmbuild_pkgbuild` (*module*), 2449
`salt.modules.rsync` (*module*), 2451
`salt.modules.runit` (*module*), 2452
`salt.modules.rvm` (*module*), 2456
`salt.modules.s3` (*module*), 2458
`salt.modules.s6` (*module*), 2461
`salt.modules.salt_proxy` (*module*), 2462
`salt.modules.salt_version` (*module*), 2462
`salt.modules.saltcheck` (*module*), 2463
`salt.modules.saltcloudmod` (*module*), 2469
`salt.modules.saltutil` (*module*), 2469
`salt.modules.schedule` (*module*), 2481
`salt.modules.scp_mod` (*module*), 2484
`salt.modules.scsi` (*module*), 2485
`salt.modules.sdb` (*module*), 2485
`salt.modules.seed` (*module*), 2486
`salt.modules.selinux` (*module*), 2487
`salt.modules.sensehat` (*module*), 2492
`salt.modules.sensors` (*module*), 2494
`salt.modules.serverdensity_device` (*module*), 2494
`salt.modules.service` (*module*), 1186
`salt.modules.servicenow` (*module*), 2495
`salt.modules.shadow` (*module*), 1187
`salt.modules.slack_notify` (*module*), 2496

salt.modules.slsutil (*module*), 2498
salt.modules.smartos_imgadm (*module*), 2500
salt.modules.smartos_nictagadm (*module*), 2502
salt.modules.smartos_virt (*module*), 2503
salt.modules.smartos_vmadm (*module*), 2504
salt.modules.smbios (*module*), 2508
salt.modules.smf_service (*module*), 2510
salt.modules.smtp (*module*), 2512
salt.modules.snapper (*module*), 2543
salt.modules.solaris_fmadm (*module*), 2513
salt.modules.solaris_group (*module*), 2515
salt.modules.solaris_shadow (*module*), 2516
salt.modules.solaris_system (*module*), 2517
salt.modules.solaris_user (*module*), 2518
salt.modules.solarisipspkg (*module*), 2520
salt.modules.solarispkg (*module*), 2524
salt.modules.solr (*module*), 2527
salt.modules.solrcloud (*module*), 2533
salt.modules.splunk (*module*), 2536
salt.modules.splunk_search (*module*), 2536
salt.modules.sqlite3 (*module*), 2538
salt.modules.ssh (*module*), 2539
salt.modules.ssh_pkg (*module*), 2542
salt.modules.ssh_service (*module*), 2542
salt.modules.state (*module*), 2547
salt.modules.status (*module*), 2558
salt.modules.statuspage (*module*), 2562
salt.modules.supervisord (*module*), 2566
salt.modules.suse_apache (*module*), 2568
salt.modules.svn (*module*), 2568
salt.modules.swarm (*module*), 2571
salt.modules.swift (*module*), 2573
salt.modules.sysbench (*module*), 2575
salt.modules.sysfs (*module*), 2575
salt.modules.syslog_ng (*module*), 2577
salt.modules.sysmod (*module*), 2582
salt.modules.sysrc (*module*), 2588
salt.modules.system (*module*), 2588
salt.modules.system_profiler (*module*), 2591
salt.modules.systemd_service (*module*), 2592
salt.modules.telegram (*module*), 2598
salt.modules.telemetry (*module*), 2598
salt.modules.temp (*module*), 2599
salt.modules.test (*module*), 2600
salt.modules.test_virtual (*module*), 2605
salt.modules.testinframod (*module*), 2604
salt.modules.textfsm_mod (*module*), 2605
salt.modules.timezone (*module*), 2608
salt.modules.tls (*module*), 2610
salt.modules.tomcat (*module*), 2618
salt.modules.trafficserver (*module*), 2622
salt.modules.travisci (*module*), 2624
salt.modules.tuned (*module*), 2625
salt.modules.twilio_notify (*module*), 2625
salt.modules.udev (*module*), 2626
salt.modules.upstart_service (*module*), 2627
salt.modules.uptime (*module*), 2629
salt.modules.user (*module*), 1187
salt.modules.useradd (*module*), 2630
salt.modules.uwsgi (*module*), 2634
salt.modules.vagrant (*module*), 2634
salt.modules.varnish (*module*), 2637
salt.modules.vault (*module*), 2638
salt.modules.vbox_guest (*module*), 2641
salt.modules.vboxmanage (*module*), 2642
salt.modules.victorops (*module*), 2644
salt.modules.virt (*module*), 2645
salt.modules.virtualenv_mod (*module*), 2676
salt.modules.vmctl (*module*), 2678
salt.modules.vsphere (*module*), 2679
salt.modules.webutil (*module*), 2719
salt.modules.win_autoruns (*module*), 2720
salt.modules.win_certutil (*module*), 2720
salt.modules.win_dacl (*module*), 2721
salt.modules.win_disk (*module*), 2723
salt.modules.win_dism (*module*), 2724
salt.modules.win_dns_client (*module*), 2728
salt.modules.win_dsc (*module*), 2729
salt.modules.win_file (*module*), 2733
salt.modules.win_firewall (*module*), 2745
salt.modules.win_groupadd (*module*), 2752
salt.modules.win_iis (*module*), 2754
salt.modules.win_ip (*module*), 2764
salt.modules.win_lgpo (*module*), 2766
salt.modules.win_license (*module*), 2772
salt.modules.win_network (*module*), 2772
salt.modules.win_ntp (*module*), 2776
salt.modules.win_path (*module*), 2776
salt.modules.win_pkg (*module*), 2777
salt.modules.win_pki (*module*), 2785
salt.modules.win_powercfg (*module*), 2787
salt.modules.win_psget (*module*), 2789
salt.modules.win_servermanager (*module*), 2792
salt.modules.win_service (*module*), 2794
salt.modules.win_shadow (*module*), 2801
salt.modules.win_smtp_server (*module*), 2802
salt.modules.win_snmp (*module*), 2804
salt.modules.win_status (*module*), 2806
salt.modules.win_system (*module*), 2812
salt.modules.win_task (*module*), 2820
salt.modules.win_timezone (*module*), 2831
salt.modules.win_useradd (*module*), 2833
salt.modules.win_wua (*module*), 2838
salt.modules.win_wusa (*module*), 2846
salt.modules.winrepo (*module*), 2791
salt.modules.x509 (*module*), 2847

`salt.modules.xapi_virt` (*module*), 2853
`salt.modules.xbpspkg` (*module*), 2857
`salt.modules.xfs` (*module*), 2860
`salt.modules.xml` (*module*), 2861
`salt.modules.xmpp` (*module*), 2862
`salt.modules.yumpkg` (*module*), 2863
`salt.modules.zabbix` (*module*), 2878
`salt.modules.zcbuildout` (*module*), 2896
`salt.modules.zenoss` (*module*), 2897
`salt.modules.zfs` (*module*), 2898
`salt.modules.zk_concurrency` (*module*), 2905
`salt.modules.znc` (*module*), 2907
`salt.modules.zoneadm` (*module*), 2907
`salt.modules.zonecfg` (*module*), 2910
`salt.modules.zookeeper` (*module*), 2912
`salt.modules.zpool` (*module*), 2916
`salt.modules.zypperpkg` (*module*), 2924
`salt.netapi.rest_cherrypy.app` (*module*), 2935
`salt.netapi.rest_cherrypy.wsgi` (*module*), 2943
`salt.netapi.rest_tornado.saltnado` (*module*), 2960
`salt.netapi.rest_tornado.saltnado_websockets` (*module*), 2962
`salt.netapi.rest_wsgi` (*module*), 2977
`salt.output.highstate` (*module*), 2980
`salt.output.json_out` (*module*), 2981
`salt.output.key` (*module*), 2982
`salt.output.nested` (*module*), 2982
`salt.output.newline_values_only` (*module*), 2983
`salt.output.no_out_quiet` (*module*), 2984
`salt.output.no_return` (*module*), 2984
`salt.output.overstatestage` (*module*), 2985
`salt.output.pony` (*module*), 2985
`salt.output.pprint_out` (*module*), 2986
`salt.output.profile` (*module*), 2986
`salt.output.progress` (*module*), 2986
`salt.output.raw` (*module*), 2987
`salt.output.table_out` (*module*), 2987
`salt.output.txt` (*module*), 2988
`salt.output.virt_query` (*module*), 2989
`salt.output.yaml_out` (*module*), 2989
`salt.pillar.cmd_json` (*module*), 2991
`salt.pillar.cmd_yaml` (*module*), 2991
`salt.pillar.cmd_yamlex` (*module*), 2991
`salt.pillar.cobbler` (*module*), 2991
`salt.pillar.confidant` (*module*), 2992
`salt.pillar.consul_pillar` (*module*), 2992
`salt.pillar.csvpillar` (*module*), 2994
`salt.pillar.digicert` (*module*), 2995
`salt.pillar.django_orm` (*module*), 2995
`salt.pillar.ec2_pillar` (*module*), 2997
`salt.pillar.etcd_pillar` (*module*), 2998
`salt.pillar.file_tree` (*module*), 2999
`salt.pillar.foreman` (*module*), 3003
`salt.pillar.git_pillar` (*module*), 3003
`salt.pillar.gpg` (*module*), 3008
`salt.pillar.hg_pillar` (*module*), 3009
`salt.pillar.hiera` (*module*), 3009
`salt.pillar.http_json` (*module*), 3009
`salt.pillar.http_yaml` (*module*), 3010
`salt.pillar.libvirt` (*module*), 3011
`salt.pillar.makostack` (*module*), 3011
`salt.pillar.mongo` (*module*), 3017
`salt.pillar.mysql` (*module*), 3018
`salt.pillar.netbox` (*module*), 3019
`salt.pillar.neutron` (*module*), 3020
`salt.pillar.nodegroups` (*module*), 3021
`salt.pillar.pepa` (*module*), 3021
`salt.pillar.pillar_ldap` (*module*), 3026
`salt.pillar.postgres` (*module*), 3028
`salt.pillar.puppet` (*module*), 3029
`salt.pillar.reclass_adapter` (*module*), 3029
`salt.pillar.redismod` (*module*), 3030
`salt.pillar.s3` (*module*), 3030
`salt.pillar.saltclass` (*module*), 3032
`salt.pillar.sql_base` (*module*), 3032
`salt.pillar.sqlcipher` (*module*), 3035
`salt.pillar.sqlite3` (*module*), 3036
`salt.pillar.stack` (*module*), 3037
`salt.pillar.svn_pillar` (*module*), 3043
`salt.pillar.varstack_pillar` (*module*), 3044
`salt.pillar.vault` (*module*), 3044
`salt.pillar.venafi` (*module*), 3046
`salt.pillar.virtkey` (*module*), 3046
`salt.pillar.vmware_pillar` (*module*), 3046
`salt.proxy.arista_pyeapi` (*module*), 3049
`salt.proxy.chronos` (*module*), 3052
`salt.proxy.cimc` (*module*), 3050
`salt.proxy.cisconso` (*module*), 3053
`salt.proxy.dummy` (*module*), 3056
`salt.proxy.esxi` (*module*), 3057
`salt.proxy.fx2` (*module*), 3062
`salt.proxy.junos` (*module*), 3065
`salt.proxy.marathon` (*module*), 3065
`salt.proxy.napalm` (*module*), 3066
`salt.proxy.netmiko_px` (*module*), 3069
`salt.proxy.nxos` (*module*), 3073
`salt.proxy.nxos_api` (*module*), 3076
`salt.proxy.panos` (*module*), 3078
`salt.proxy.philips_hue` (*module*), 3082
`salt.proxy.rest_sample` (*module*), 3085
`salt.proxy.ssh_sample` (*module*), 3086
`salt.queues.pgjsonb_queue` (*module*), 3087
`salt.queues.sqlite_queue` (*module*), 3088
`salt.renderers.cheetah` (*module*), 361

salt.renderers.dson (*module*), 361
salt.renderers.genshi (*module*), 362
salt.renderers.gpg (*module*), 362
salt.renderers.hjson (*module*), 365
salt.renderers.jinja (*module*), 365
salt.renderers.json (*module*), 368
salt.renderers.json5 (*module*), 368
salt.renderers.mako (*module*), 368
salt.renderers.msgpack (*module*), 368
salt.renderers.pass (*module*), 368
salt.renderers.py (*module*), 369
salt.renderers.pydsl (*module*), 371
salt.renderers.pyobjects (*module*), 375
salt.renderers.stateconf (*module*), 380
salt.renderers.wempy (*module*), 384
salt.renderers.yaml (*module*), 385
salt.renderers.yamlex (*module*), 386
salt.returners.carbon_return (*module*), 312
salt.returners.cassandra_cql_return (*module*), 313
salt.returners.cassandra_return (*module*), 316
salt.returners.couchbase_return (*module*), 316
salt.returners.couchdb_return (*module*), 317
salt.returners.django_return (*module*), 318
salt.returners.elasticsearch_return (*module*), 319
salt.returners.etcd_return (*module*), 320
salt.returners.highstate_return (*module*), 322
salt.returners.influxdb_return (*module*), 323
salt.returners.kafka_return (*module*), 324
salt.returners.librato_return (*module*), 324
salt.returners.local (*module*), 325
salt.returners.local_cache (*module*), 325
salt.returners.mattermost_returner (*module*), 326
salt.returners.memcache_return (*module*), 327
salt.returners.mongo_future_return (*module*), 328
salt.returners.mongo_return (*module*), 329
salt.returners.multi_returner (*module*), 330
salt.returners.mysql (*module*), 331
salt.returners.nagios_nrdp_return (*module*), 333
salt.returners.odbc (*module*), 334
salt.returners.pgjsonb (*module*), 337
salt.returners.postgres (*module*), 340
salt.returners.postgres_local_cache (*module*), 342
salt.returners.pushover_returner (*module*), 344
salt.returners.rawfile_json (*module*), 346
salt.returners.redis_return (*module*), 346
salt.returners.sentry_return (*module*), 348
salt.returners.slack_returner (*module*), 348
salt.returners.sms_return (*module*), 350
salt.returners.smtp_return (*module*), 350
salt.returners.splunk (*module*), 352
salt.returners.sqlite3_return (*module*), 352
salt.returners.syslog_return (*module*), 354
salt.returners.telegram_return (*module*), 355
salt.returners.xmpp_return (*module*), 356
salt.returners.zabbix_return (*module*), 357
salt.roster.ansible (*module*), 3089
salt.roster.cache (*module*), 3090
salt.roster.cloud (*module*), 3092
salt.roster.clustershell (*module*), 3092
salt.roster.flat (*module*), 3092
salt.roster.range (*module*), 3092
salt.roster.scan (*module*), 3093
salt.roster.sshconfig (*module*), 3093
salt.roster.terraform (*module*), 3093
salt.runners.asam (*module*), 3095
salt.runners.auth (*module*), 3096
salt.runners.bgp (*module*), 3097
salt.runners.cache (*module*), 3099
salt.runners.cloud (*module*), 3102
salt.runners.ddns (*module*), 3103
salt.runners.digicertapi (*module*), 3104
salt.runners.doc (*module*), 3107
salt.runners.drac (*module*), 3107
salt.runners.error (*module*), 3108
salt.runners.event (*module*), 3108
salt.runners.f5 (*module*), 3109
salt.runners.fileserver (*module*), 3111
salt.runners.git_pillar (*module*), 3115
salt.runners.http (*module*), 3116
salt.runners.jobs (*module*), 3116
salt.runners.launchd (*module*), 3118
salt.runners.lxc (*module*), 3119
salt.runners.manage (*module*), 3121
salt.runners.mattermost (*module*), 3126
salt.runners.mine (*module*), 3127
salt.runners.nacl (*module*), 3127
salt.runners.net (*module*), 3130
salt.runners.network (*module*), 3136
salt.runners.pagerduty (*module*), 3137
salt.runners.pillar (*module*), 3138
salt.runners.pkg (*module*), 3139
salt.runners.queue (*module*), 3139
salt.runners.reactor (*module*), 3141
salt.runners.salt (*module*), 3142

`salt.runners.saltutil` (*module*), 3143
`salt.runners.sdb` (*module*), 3148
`salt.runners.smartos_vmadm` (*module*), 3149
`salt.runners.spacewalk` (*module*), 3151
`salt.runners.ssh` (*module*), 3152
`salt.runners.state` (*module*), 3152
`salt.runners.survey` (*module*), 3155
`salt.runners.test` (*module*), 3156
`salt.runners.thin` (*module*), 3156
`salt.runners.vault` (*module*), 3157
`salt.runners.venafiapi` (*module*), 3157
`salt.runners.virt` (*module*), 3158
`salt.runners.vistara` (*module*), 3160
`salt.runners.winrepo` (*module*), 3160
`salt.sdb.cache` (*module*), 3161
`salt.sdb.confidant` (*module*), 3162
`salt.sdb.consul` (*module*), 3163
`salt.sdb.couchdb` (*module*), 3163
`salt.sdb.env` (*module*), 3164
`salt.sdb.etcd_db` (*module*), 3165
`salt.sdb.keyring_db` (*module*), 3165
`salt.sdb.memcached` (*module*), 3166
`salt.sdb.redis_sdb` (*module*), 3167
`salt.sdb.rest` (*module*), 3167
`salt.sdb.sqlite3` (*module*), 3168
`salt.sdb.tism` (*module*), 3169
`salt.sdb.vault` (*module*), 3170
`salt.sdb.yaml` (*module*), 3170
`salt.serializers.configparser` (*module*), 3171
`salt.serializers.json` (*module*), 3172
`salt.serializers.msgpack` (*module*), 3172
`salt.serializers.python` (*module*), 3172
`salt.serializers.toml` (*module*), 3173
`salt.serializers.yaml` (*module*), 3173
`salt.serializers.yamlex` (*module*), 3173
`salt.states.acme` (*module*), 3182
`salt.states.alias` (*module*), 3183
`salt.states.alternatives` (*module*), 3184
`salt.states.ansiblegate` (*module*), 3184
`salt.states.apache` (*module*), 3185
`salt.states.apache_conf` (*module*), 3187
`salt.states.apache_module` (*module*), 3187
`salt.states.apache_site` (*module*), 3188
`salt.states.aptpkg` (*module*), 3188
`salt.states.archive` (*module*), 3188
`salt.states.artifactory` (*module*), 3195
`salt.states.at` (*module*), 3195
`salt.states.augeas` (*module*), 3197
`salt.states.aws_sqs` (*module*), 3199
`salt.states.azurearm_compute` (*module*), 3200
`salt.states.azurearm_network` (*module*), 3202
`salt.states.azurearm_resource` (*module*), 3216
`salt.states.beacon` (*module*), 3220
`salt.states.bigip` (*module*), 3222
`salt.states.blockdev` (*module*), 3233
`salt.states.boto3_elasticache` (*module*), 3233
`salt.states.boto3_elasticsearch` (*module*), 3242
`salt.states.boto3_route53` (*module*), 3246
`salt.states.boto_apigateway` (*module*), 3250
`salt.states.boto_asg` (*module*), 3255
`salt.states.boto_cfn` (*module*), 3260
`salt.states.boto_cloudfront` (*module*), 3262
`salt.states.boto_cloudtrail` (*module*), 3263
`salt.states.boto_cloudwatch_alarm` (*module*), 3265
`salt.states.boto_cloudwatch_event` (*module*), 3266
`salt.states.boto_cognitoidentity` (*module*), 3267
`salt.states.boto_datapipeline` (*module*), 3269
`salt.states.boto_dynamodb` (*module*), 3270
`salt.states.boto_ec2` (*module*), 3273
`salt.states.boto_elasticache` (*module*), 3280
`salt.states.boto_elasticsearch_domain` (*module*), 3283
`salt.states.boto_elb` (*module*), 3285
`salt.states.boto_elbv2` (*module*), 3290
`salt.states.boto_iam` (*module*), 3292
`salt.states.boto_iam_role` (*module*), 3298
`salt.states.boto_iot` (*module*), 3300
`salt.states.boto_kinesis` (*module*), 3303
`salt.states.boto_kms` (*module*), 3304
`salt.states.boto_lambda` (*module*), 3306
`salt.states.boto_lc` (*module*), 3309
`salt.states.boto_rds` (*module*), 3312
`salt.states.boto_route53` (*module*), 3316
`salt.states.boto_s3_bucket` (*module*), 3318
`salt.states.boto_secgroup` (*module*), 3321
`salt.states.boto_sns` (*module*), 3324
`salt.states.boto_sqs` (*module*), 3325
`salt.states.boto_vpc` (*module*), 3326
`salt.states.bower` (*module*), 3335
`salt.states.cabal` (*module*), 3336
`salt.states.ceph` (*module*), 3336
`salt.states.chef` (*module*), 3337
`salt.states.chocolatey` (*module*), 3338
`salt.states.chronos_job` (*module*), 3339
`salt.states.cimc` (*module*), 3340
`salt.states.cisconso` (*module*), 3342
`salt.states.cloud` (*module*), 3343
`salt.states.cmd` (*module*), 3344
`salt.states.composer` (*module*), 3354
`salt.states.cron` (*module*), 3355

`salt.states.csf` (*module*), 3359
`salt.states.cyg` (*module*), 3360
`salt.states.ddns` (*module*), 3361
`salt.states.debconfmod` (*module*), 3362
`salt.states.dellchassis` (*module*), 3364
`salt.states.disk` (*module*), 3368
`salt.states.docker_container` (*module*), 3369
`salt.states.docker_image` (*module*), 3390
`salt.states.docker_network` (*module*), 3393
`salt.states.docker_volume` (*module*), 3399
`salt.states.drac` (*module*), 3400
`salt.states.elasticsearch` (*module*), 3401
`salt.states.elasticsearch_index` (*module*), 3402
`salt.states.elasticsearch_index_template` (*module*), 3403
`salt.states.environ` (*module*), 3404
`salt.states.eselect` (*module*), 3404
`salt.states.esxi` (*module*), 3409
`salt.states.etcd_mod` (*module*), 3405
`salt.states.ethtool` (*module*), 3407
`salt.states.event` (*module*), 3415
`salt.states.file` (*module*), 3415
`salt.states.firewall` (*module*), 3451
`salt.states.firewalld` (*module*), 3451
`salt.states.gem` (*module*), 3453
`salt.states.git` (*module*), 3454
`salt.states.github` (*module*), 3463
`salt.states.glance` (*module*), 3465
`salt.states.glance_image` (*module*), 3465
`salt.states.glusterfs` (*module*), 3466
`salt.states.gnomedesktop` (*module*), 3468
`salt.states.gpg` (*module*), 3470
`salt.states.grafana` (*module*), 3470
`salt.states.grafana4_dashboard` (*module*), 3474
`salt.states.grafana4_datasource` (*module*), 3475
`salt.states.grafana4_org` (*module*), 3476
`salt.states.grafana4_user` (*module*), 3477
`salt.states.grafana_dashboard` (*module*), 3478
`salt.states.grafana_datasource` (*module*), 3479
`salt.states.grains` (*module*), 3479
`salt.states.group` (*module*), 3482
`salt.states.heat` (*module*), 3483
`salt.states.hg` (*module*), 3484
`salt.states.host` (*module*), 3485
`salt.states.http` (*module*), 3486
`salt.states.icinga2` (*module*), 3487
`salt.states.ifttt` (*module*), 3488
`salt.states.incron` (*module*), 3489
`salt.states.influxdb08_database` (*module*), 3490
`salt.states.influxdb08_user` (*module*), 3490
`salt.states.influxdb_continuous_query` (*module*), 3491
`salt.states.influxdb_database` (*module*), 3491
`salt.states.influxdb_retention_policy` (*module*), 3492
`salt.states.influxdb_user` (*module*), 3492
`salt.states.infoblox_a` (*module*), 3493
`salt.states.infoblox_cname` (*module*), 3493
`salt.states.infoblox_host_record` (*module*), 3494
`salt.states.infoblox_range` (*module*), 3495
`salt.states.ini_manage` (*module*), 3496
`salt.states.ipmi` (*module*), 3498
`salt.states.ipset` (*module*), 3500
`salt.states.iptables` (*module*), 3501
`salt.states.jboss7` (*module*), 3507
`salt.states.jenkins` (*module*), 3509
`salt.states.junos` (*module*), 3509
`salt.states.kapacitor` (*module*), 3514
`salt.states.kernelpkg` (*module*), 3515
`salt.states.keyboard` (*module*), 3517
`salt.states.keystone` (*module*), 3517
`salt.states.keystone_domain` (*module*), 3520
`salt.states.keystone_endpoint` (*module*), 3521
`salt.states.keystone_group` (*module*), 3522
`salt.states.keystone_project` (*module*), 3522
`salt.states.keystone_role` (*module*), 3523
`salt.states.keystone_role_grant` (*module*), 3524
`salt.states.keystone_service` (*module*), 3524
`salt.states.keystone_user` (*module*), 3525
`salt.states.keystore` (*module*), 3526
`salt.states.kmod` (*module*), 3526
`salt.states.kubernetes` (*module*), 3527
`salt.states.layman` (*module*), 3530
`salt.states.ldap` (*module*), 3531
`salt.states.libcloud_dns` (*module*), 3534
`salt.states.libcloud_loadbalancer` (*module*), 3535
`salt.states.libcloud_storage` (*module*), 3536
`salt.states.linux_acl` (*module*), 3538
`salt.states.locale` (*module*), 3540
`salt.states.logadm` (*module*), 3540
`salt.states.logrotate` (*module*), 3541
`salt.states.loop` (*module*), 3541
`salt.states.lvm` (*module*), 3543
`salt.states.lvs_server` (*module*), 3544
`salt.states.lvs_service` (*module*), 3544
`salt.states.lxc` (*module*), 3545

salt.states.mac_assistive (*module*), 3548
salt.states.mac_keychain (*module*), 3549
salt.states.mac_xattr (*module*), 3550
salt.states.macdefaults (*module*), 3548
salt.states.macpackage (*module*), 3549
salt.states.makeconf (*module*), 3551
salt.states.marathon_app (*module*), 3551
salt.states.mdadm_raid (*module*), 3552
salt.states.memcached (*module*), 3553
salt.states.modjk (*module*), 3553
salt.states.modjk_worker (*module*), 3554
salt.states.module (*module*), 3555
salt.states.mongodb_database (*module*), 3560
salt.states.mongodb_user (*module*), 3560
salt.states.monit (*module*), 3561
salt.states.mount (*module*), 3562
salt.states.msteams (*module*), 3565
salt.states.mysql_database (*module*), 3566
salt.states.mysql_grants (*module*), 3566
salt.states.mysql_query (*module*), 3568
salt.states.mysql_user (*module*), 3569
salt.states.net_napalm_yang (*module*), 3602
salt.states.netacl (*module*), 3570
salt.states.netconfig (*module*), 3581
salt.states.netntp (*module*), 3590
salt.states.netsnmp (*module*), 3591
salt.states.netusers (*module*), 3592
salt.states.network (*module*), 3596
salt.states.neutron_network (*module*), 3603
salt.states.neutron_secgroup (*module*), 3604
salt.states.neutron_secgroup_rule (*module*), 3605
salt.states.neutron_subnet (*module*), 3606
salt.states.nfs_export (*module*), 3607
salt.states.nftables (*module*), 3608
salt.states.npm (*module*), 3611
salt.states.ntp (*module*), 3612
salt.states.nxos (*module*), 3613
salt.states.openstack_config (*module*), 3615
salt.states.openvswitch_bridge (*module*), 3615
salt.states.openvswitch_port (*module*), 3615
salt.states.opsgenie (*module*), 3616
salt.states.pagerduty (*module*), 3617
salt.states.pagerduty_escalation_policy (*module*), 3617
salt.states.pagerduty_schedule (*module*), 3618
salt.states.pagerduty_service (*module*), 3619
salt.states.pagerduty_user (*module*), 3620
salt.states.panos (*module*), 3621
salt.states.pcs (*module*), 3630
salt.states.pdbedit (*module*), 3637
salt.states.pecl (*module*), 3637
salt.states.pip_state (*module*), 3638
salt.states.pkg (*module*), 3642
salt.states.pkgbuild (*module*), 3658
salt.states.pkgng (*module*), 3660
salt.states.pkgrepo (*module*), 3661
salt.states.portage_config (*module*), 3664
salt.states.ports (*module*), 3665
salt.states.postgres_cluster (*module*), 3665
salt.states.postgres_database (*module*), 3666
salt.states.postgres_extension (*module*), 3667
salt.states.postgres_group (*module*), 3668
salt.states.postgres_initdb (*module*), 3669
salt.states.postgres_language (*module*), 3670
salt.states.postgres_privileges (*module*), 3670
salt.states.postgres_schema (*module*), 3673
salt.states.postgres_tablespace (*module*), 3674
salt.states.postgres_user (*module*), 3675
salt.states.powerpath (*module*), 3676
salt.states.probes (*module*), 3676
salt.states.process (*module*), 3678
salt.states.proxy (*module*), 3678
salt.states.pushover (*module*), 3678
salt.states.pyenv (*module*), 3679
salt.states.pyrax_queues (*module*), 3680
salt.states.quota (*module*), 3681
salt.states.rabbitmq_cluster (*module*), 3681
salt.states.rabbitmq_plugin (*module*), 3682
salt.states.rabbitmq_policy (*module*), 3682
salt.states.rabbitmq_upstream (*module*), 3683
salt.states.rabbitmq_user (*module*), 3684
salt.states.rabbitmq_vhost (*module*), 3685
salt.states.rbac_solaris (*module*), 3686
salt.states.rbenv (*module*), 3686
salt.states.rdp (*module*), 3688
salt.states.redismod (*module*), 3688
salt.states.reg (*module*), 3689
salt.states.rsync (*module*), 3694
salt.states.rvm (*module*), 3695
salt.states.salt_proxy (*module*), 3697
salt.states.saltmod (*module*), 3698
salt.states.saltutil (*module*), 3701
salt.states.schedule (*module*), 3704
salt.states.selinux (*module*), 3707
salt.states.serverdensity_device (*module*), 3709
salt.states.service (*module*), 3710
salt.states.slack (*module*), 3713

`salt.states.smartos` (*module*), 3714
`salt.states.smtp` (*module*), 3717
`salt.states.snapper` (*module*), 3718
`salt.states.solrcloud` (*module*), 3720
`salt.states.splunk` (*module*), 3720
`salt.states.splunk_search` (*module*), 3721
`salt.states.sqlite3` (*module*), 3721
`salt.states.ssh_auth` (*module*), 3723
`salt.states.ssh_known_hosts` (*module*), 3725
`salt.states.stateconf` (*module*), 3726
`salt.states.status` (*module*), 3727
`salt.states.statuspage` (*module*), 3727
`salt.states.supervisord` (*module*), 3729
`salt.states.svn` (*module*), 3729
`salt.states.sysctl` (*module*), 3730
`salt.states.syslog_ng` (*module*), 3731
`salt.states.sysrc` (*module*), 3732
`salt.states.telemetry_alert` (*module*), 3732
`salt.states.test` (*module*), 3733
`salt.states.testinframod` (*module*), 3736
`salt.states.timezone` (*module*), 3736
`salt.states.tls` (*module*), 3736
`salt.states.tomcat` (*module*), 3736
`salt.states.trafficserver` (*module*), 3739
`salt.states.tuned` (*module*), 3741
`salt.states.uptime` (*module*), 3741
`salt.states.user` (*module*), 3742
`salt.states.vagrant` (*module*), 3744
`salt.states.vault` (*module*), 3746
`salt.states.vbox_guest` (*module*), 3747
`salt.states.victorops` (*module*), 3747
`salt.states.virt` (*module*), 3748
`salt.states.virtualenv_mod` (*module*), 3755
`salt.states.webutil` (*module*), 3756
`salt.states.win_certutil` (*module*), 3756
`salt.states.win_dacl` (*module*), 3757
`salt.states.win_dism` (*module*), 3758
`salt.states.win_dns_client` (*module*), 3761
`salt.states.win_firewall` (*module*), 3761
`salt.states.win_iis` (*module*), 3763
`salt.states.win_lgpo` (*module*), 3769
`salt.states.win_license` (*module*), 3772
`salt.states.win_network` (*module*), 3772
`salt.states.win_path` (*module*), 3774
`salt.states.win_pki` (*module*), 3774
`salt.states.win_powercfg` (*module*), 3775
`salt.states.win_servermanager` (*module*), 3776
`salt.states.win_smtp_server` (*module*), 3779
`salt.states.win_snmp` (*module*), 3781
`salt.states.win_system` (*module*), 3782
`salt.states.win_wua` (*module*), 3784
`salt.states.win_wusa` (*module*), 3787
`salt.states.winrepo` (*module*), 3787
`salt.states.x509` (*module*), 3788
`salt.states.xml` (*module*), 3792
`salt.states.xmpp` (*module*), 3793
`salt.states.zabbix_action` (*module*), 3793
`salt.states.zabbix_host` (*module*), 3795
`salt.states.zabbix_hostgroup` (*module*), 3796
`salt.states.zabbix_mediatype` (*module*), 3797
`salt.states.zabbix_template` (*module*), 3798
`salt.states.zabbix_user` (*module*), 3800
`salt.states.zabbix_usergroup` (*module*), 3802
`salt.states.zabbix_valuemap` (*module*), 3803
`salt.states.zcbuildout` (*module*), 3804
`salt.states.zenoss` (*module*), 3805
`salt.states.zfs` (*module*), 3806
`salt.states.zk_concurrency` (*module*), 3805
`salt.states.zone` (*module*), 3809
`salt.states.zookeeper` (*module*), 3814
`salt.states.zpool` (*module*), 3816
`salt.thorium.calc` (*module*), 3819
`salt.thorium.check` (*module*), 3820
`salt.thorium.file` (*module*), 3824
`salt.thorium.key` (*module*), 3825
`salt.thorium.local` (*module*), 3825
`salt.thorium.reg` (*module*), 3826
`salt.thorium.runner` (*module*), 3827
`salt.thorium.status` (*module*), 3827
`salt.thorium.timer` (*module*), 3828
`salt.thorium.wheel` (*module*), 3828
`salt.tokens.localfs` (*module*), 3828
`salt.tokens.rediscluster` (*module*), 3829
`salt.tops.cobbler` (*module*), 3830
`salt.tops.ext_nodes` (*module*), 3830
`salt.tops.mongo` (*module*), 3831
`salt.tops.reclass_adapter` (*module*), 3832
`salt.tops.saltclass` (*module*), 3832
`salt.tops.varstack_top` (*module*), 3836
`salt.utils.aggregation` (*module*), 3996
`salt.utils.extend` (*module*), 3928
`salt.wheel.config` (*module*), 3837
`salt.wheel.error` (*module*), 3838
`salt.wheel.file_roots` (*module*), 3838
`salt.wheel.key` (*module*), 3838
`salt.wheel.minions` (*module*), 3842
`salt.wheel.pillar_roots` (*module*), 3842
`SaltAPIHandler` (*class in salt.netapi.rest_tornado.saltnado*), 2967
`SaltAuthHandler` (*class in salt.netapi.rest_tornado.saltnado*), 2969
`SaltCacheError`, 3999
`SaltClientError`, 3999
`SaltClientTimeout`, 3999
`SaltCloudConfigError`, 3999
`SaltCloudException`, 3999
`SaltCloudExecutionFailure`, 3999

SaltCloudExecutionTimeout, 3999
SaltCloudNotFound, 3999
SaltCloudPasswordError, 3999
SaltCloudSystemExit, 3999
SaltConfigurationError, 4000
SaltDaemonNotRunning, 4000
SaltDeserializationError, 4000
saltenv
  conf/minion, 151
SaltException, 4000
SaltInvocationError, 4000
SaltMasterError, 4000
SaltMasterUnresolvableError, 4000
saltmem() (*in module salt.modules.win_status*), 2811
SaltNoMinionsFound, 4000
saltpath() (*in module salt.grains.core*), 1179
SaltRenderError, 4000
SaltReqTimeoutError, 4000
SaltRunnerError, 4000
SaltSyndicMasterError, 4000
SaltSystemExit, 4000
saltversion() (*in module salt.grains.core*), 1179
saltversioninfo() (*in module salt.grains.core*), 1179
SaltWheelError, 4000
same() (*salt.states.cyg.DictDiffer method*), 3361
saml_provider_absent() (*in module salt.states.boto_iam*), 3296
saml_provider_present() (*in module salt.states.boto_iam*), 3297
save() (*in module salt.modules.beacons*), 1285
save() (*in module salt.modules.dockermod*), 1654
save() (*in module salt.modules.iptables*), 1902
save() (*in module salt.modules.nftables*), 2247
save() (*in module salt.modules.redismod*), 2431
save() (*in module salt.modules.schedule*), 2483
save() (*in module salt.thorium.file*), 3825
save_cert() (*in module salt.modules.icinga2*), 1855
save_cert() (*in module salt.states.icinga2*), 3488
save_config() (*in module salt.modules.mdadm_raid*), 2075
save_config() (*in module salt.modules.napalm_network*), 2175
save_device_config() (*in module salt.modules.panos*), 2315
save_device_state() (*in module salt.modules.panos*), 2316
save_load() (*in module salt.returners.cassandra_cql_return*), 315
save_load() (*in module salt.returners.couchbase_return*), 317
save_load() (*in module salt.returners.django_return*), 319
save_load() (*in module salt.returners.elasticsearch_return*), 320
save_load() (*in module salt.returners.etcd_return*), 321
save_load() (*in module salt.returners.influxdb_return*), 324
save_load() (*in module salt.returners.local_cache*), 325
save_load() (*in module salt.returners.memcache_return*), 328
save_load() (*in module salt.returners.mongo_future_return*), 329
save_load() (*in module salt.returners.multi_returner*), 331
save_load() (*in module salt.returners.mysql*), 333
save_load() (*in module salt.returners.odbc*), 337
save_load() (*in module salt.returners.pgjsonb*), 340
save_load() (*in module salt.returners.postgres*), 342
save_load() (*in module salt.returners.postgres_local_cache*), 344
save_load() (*in module salt.returners.redis_return*), 348
save_load() (*in module salt.returners.sqlite3_return*), 354
save_load() (*in module salt.returners.zabbix_return*), 357
save_minions() (*in module salt.returners.couchbase_return*), 317
save_minions() (*in module salt.returners.local_cache*), 326
save_reg() (*in module salt.returners.local_cache*), 326
saved() (*in module salt.states.netconfig*), 3588
saved() (*in module salt.states.virt*), 3754
say() (*in module salt.modules.mac_desktop*), 2026
Scalar() (*in module salt.utils.aggregation*), 3998
scan() (*in module salt.modules.bluez_bluetooth*), 1300
scan() (*in module salt.modules.iwtools*), 1902
schedule_downtime() (*in module salt.modules.datadog_api*), 1589
schedule_enable() (*in module salt.modules.mac_softwareupdate*), 2047
schedule_enabled() (*in module salt.modules.mac_softwareupdate*), 2047
schedule_probes() (*in module salt.modules.napalm_probes*), 2181
scheduled_snapshot() (*in module salt.states.zfs*), 3808
schema_create() (*in module salt.modules.postgres*), 2382
schema_exists() (*in module salt.modules.postgres*), 2382
schema_get() (*in module salt.modules.glance*), 1810
schema_get() (*in module salt.modules.postgres*), 2383
schema_list() (*in module salt.modules.postgres*),

2383
`schema_remove()` (*in module salt.modules.postgres*), 2383
`SCOPES` (*salt.modules.inspectlib.query.Query attribute*), 1876
`scp_get()` (*in module salt.modules.napalm_mod*), 2137
`scp_put()` (*in module salt.modules.napalm_mod*), 2138
`screensaver()` (*in module salt.modules.mac_desktop*), 2026
`script()` (*in module salt.cloud.clouds.aliyun*), 1034
`script()` (*in module salt.cloud.clouds.cloudstack*), 1039
`script()` (*in module salt.cloud.clouds.digitalocean*), 1042
`script()` (*in module salt.cloud.clouds.dimensiondata*), 1044
`script()` (*in module salt.cloud.clouds.ec2*), 1051
`script()` (*in module salt.cloud.clouds.gce*), 1057
`script()` (*in module salt.cloud.clouds.msazure*), 1082
`script()` (*in module salt.cloud.clouds.oneandone*), 1089
`script()` (*in module salt.cloud.clouds.parallels*), 1114
`script()` (*in module salt.cloud.clouds.proxmox*), 1120
`script()` (*in module salt.cloud.clouds.qingcloud*), 1123
`script()` (*in module salt.cloud.clouds.scaleway*), 1127
`script()` (*in module salt.cloud.clouds.softlayer*), 1128
`script()` (*in module salt.cloud.clouds.softlayer_hw*), 1129
`script()` (*in module salt.cloud.clouds.tencentcloud*), 1132
`script()` (*in module salt.modules.cmdmod*), 1546
`script()` (*in module salt.modules.dockermod*), 1655
`script()` (*in module salt.states.cmd*), 3350
`script_retcode()` (*in module salt.modules.cmdmod*), 1548
`script_retcode()` (*in module salt.modules.dockermod*), 1655
`scrub()` (*in module salt.modules.zpool*), 2922
`sealedbox_decrypt()` (*in module salt.modules.nacl*), 2110
`sealedbox_decrypt()` (*in module salt.runners.nacl*), 3129
`sealedbox_encrypt()` (*in module salt.modules.nacl*), 2110
`sealedbox_encrypt()` (*in module salt.runners.nacl*), 3130
`search()` (*in module salt.modules.dockermod*), 1656
`search()` (*in module salt.modules.file*), 1727
`search()` (*in module salt.modules.freebsdports*), 1746
`search()` (*in module salt.modules.ldap3*), 1957
`search()` (*in module salt.modules.ldapmod*), 1958
`search()` (*in module salt.modules.pkgin*), 2353
`search()` (*in module salt.modules.pkgng*), 2363
`search()` (*in module salt.modules.solarisipspkg*), 2523
`search()` (*in module salt.modules.zypperpkg*), 2933
`search_by()` (*in module salt.modules.match*), 2073
`search_keys()` (*in module salt.modules.gpg*), 1821
`search_lxc_bridge()` (*in module salt.modules.lxc*), 2000
`search_lxc_bridges()` (*in module salt.modules.lxc*), 2000
`search_template_absent()` (*in module salt.states.elasticsearch*), 3402
`search_template_create()` (*in module salt.modules.elasticsearch*), 1691
`search_template_delete()` (*in module salt.modules.elasticsearch*), 1691
`search_template_get()` (*in module salt.modules.elasticsearch*), 1691
`search_template_present()` (*in module salt.states.elasticsearch*), 3402
`secgroup_allocate()` (*in module salt.cloud.clouds.opennebula*), 1097
`secgroup_clone()` (*in module salt.cloud.clouds.opennebula*), 1097
`secgroup_create()` (*in module salt.modules.nova*), 2255
`secgroup_delete()` (*in module salt.cloud.clouds.opennebula*), 1098
`secgroup_delete()` (*in module salt.modules.nova*), 2255
`secgroup_info()` (*in module salt.cloud.clouds.opennebula*), 1098
`secgroup_list()` (*in module salt.modules.nova*), 2255
`secgroup_update()` (*in module salt.cloud.clouds.opennebula*), 1098
`SecondLevelTbFills` (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATIO attribute*), 2810
`secret_absent()` (*in module salt.states.kubernetes*), 3530
`secret_present()` (*in module salt.states.kubernetes*), 3530
`secretbox_decrypt()` (*in module salt.modules.nacl*), 2110
`secretbox_decrypt()` (*in module salt.runners.nacl*), 3130
`secretbox_encrypt()` (*in module salt.modules.nacl*), 2111
`secretbox_encrypt()` (*in module salt.runners.nacl*), 3130
`secrets()` (*in module salt.modules.kubernetesmod*), 1950
`section()` (*in module salt.modules.arista_pyeapi*), 1239
`sections_absent()` (*in module salt.states.ini_manage*), 3497
`sections_present()` (*in module salt.states.ini_manage*), 3497
`security_group_create()` (*in module*

salt.modules.neutronng), 2240
security_group_delete() (*in module salt.modules.neutronng*), 2240
security_group_get() (*in module salt.modules.neutronng*), 2240
security_group_rule_create() (*in module salt.modules.neutronng*), 2240
security_group_rule_delete() (*in module salt.modules.neutronng*), 2241
security_group_update() (*in module salt.modules.neutronng*), 2241
security_rule_absent() (*in module salt.states.azurearm_network*), 3213
security_rule_create_or_update() (*in module salt.modules.azurearm_network*), 1266
security_rule_delete() (*in module salt.modules.azurearm_network*), 1267
security_rule_exists() (*in module salt.states.panos*), 3626
security_rule_get() (*in module salt.modules.azurearm_network*), 1268
security_rule_present() (*in module salt.states.azurearm_network*), 3213
security_rules_list() (*in module salt.modules.azurearm_network*), 1268
securitygroup() (*in module salt.cloud.clouds.ec2*), 1051
securitygroupid() (*in module salt.cloud.clouds.ec2*), 1051
sed() (*in module salt.modules.file*), 1727
sed_contains() (*in module salt.modules.file*), 1728
seed() (*in module salt.modules.mod_random*), 2083
seed_non_shared_migrate() (*in module salt.modules.virt*), 2669
seek_read() (*in module salt.modules.file*), 1728
seek_write() (*in module salt.modules.file*), 1728
select_query() (*in module salt.modules.cloud*), 1522
select_query() (*in module salt.runners.cloud*), 3103
select_query() (*salt.cloud.CloudClient method*), 3853
selfupdate() (*in module salt.modules.composer*), 1555
selinux_fs_path() (*in module salt.modules.selinux*), 2491
send() (*in module salt.modules.event*), 1700
send() (*in module salt.modules.mandrill*), 2068
send() (*in module salt.modules.mine*), 2079
send() (*in module salt.modules.smartos_vmadm*), 2507
send() (*in module salt.runners.event*), 3108
send() (*in module salt.states.event*), 3415
send_command() (*in module salt.modules.netmiko_mod*), 2209
send_command_timing() (*in module salt.modules.netmiko_mod*), 2210
send_config() (*in module salt.modules.netmiko_mod*), 2210
send_message() (*in module salt.modules.google_chat*), 1817
send_message() (*salt.engines.ircbot.IRCClient method*), 1156
send_msg() (*in module salt.modules.smtp*), 2513
send_msg() (*in module salt.modules.xmpp*), 2863
send_msg() (*in module salt.states.smtp*), 3717
send_msg() (*in module salt.states.xmpp*), 3793
send_msg_multi() (*in module salt.modules.xmpp*), 2863
send_msg_multi() (*in module salt.states.xmpp*), 3793
send_sms() (*in module salt.modules.twilio_notify*), 2626
sendEvent() (*salt.returners.splunk.http_event_collector method*), 352
sendline() (*in module salt.proxy.nxos*), 3075
SendMsgBot (*class in salt.modules.xmpp*), 2862
SendMsgBot (*class in salt.returners.xmpp_return*), 357
sense() (*in module salt.modules.sensors*), 2494
sentinel_get_master_ip() (*in module salt.modules.redismod*), 2431
sequence
  jinja filters, 428
Sequence (*class in salt.utils.aggregation*), 3998
serial() (*in module salt.grains.napalm*), 1183
serial() (*in module salt.modules.dnsutil*), 1609
serialize() (*in module salt.modules.slsutil*), 2499
serialize() (*in module salt.serializers.configparser*), 3171
serialize() (*in module salt.serializers.json*), 3172
serialize() (*in module salt.serializers.msgpack*), 3172
serialize() (*in module salt.serializers.python*), 3172
serialize() (*in module salt.serializers.toml*), 3173
serialize() (*in module salt.serializers.yaml*), 3173
serialize() (*in module salt.serializers.yamlex*), 3175
serialize() (*in module salt.states.file*), 3447
SerializerExtension (*class in salt.utils.jinja*), 365
server_add() (*in module salt.modules.netscaler*), 2211
server_by_name() (*in module salt.modules.nova*), 2256
server_cert_absent() (*in module salt.states.boto_iam*), 3297
server_cert_present() (*in module salt.states.boto_iam*), 3297
server_delete() (*in module salt.modules.netscaler*), 2211
server_disable() (*in module salt.modules.netscaler*), 2212
server_enable() (*in module salt.modules.netscaler*), 2212
server_enabled() (*in module salt.modules.netscaler*), 2212

`server_exists()` (*in module salt.modules.netscaler*), 2212

`server_hardreset()` (*in module salt.modules.drac*), 1663

`server_hardreset()` (*in module salt.modules.dracr*), 1668

`server_list()` (*in module salt.modules.nova*), 2256

`server_list_detailed()` (*in module salt.modules.nova*), 2256

`server_power()` (*in module salt.modules.dracr*), 1669

`server_poweroff()` (*in module salt.modules.drac*), 1663

`server_poweroff()` (*in module salt.modules.dracr*), 1669

`server_poweron()` (*in module salt.modules.drac*), 1663

`server_poweron()` (*in module salt.modules.dracr*), 1669

`server_powerstatus()` (*in module salt.modules.dracr*), 1670

`server_pxe()` (*in module salt.modules.drac*), 1663

`server_pxe()` (*in module salt.modules.dracr*), 1670

`server_reboot()` (*in module salt.modules.drac*), 1664

`server_reboot()` (*in module salt.modules.dracr*), 1670

`server_setting()` (*in module salt.states.win_smtp_server*), 3780

`server_show()` (*in module salt.modules.nova*), 2256

`server_status()` (*in module salt.modules.apache*), 1210

`server_update()` (*in module salt.modules.netscaler*), 2212

`serverinfo()` (*in module salt.modules.tomcat*), 2620

`servermods()` (*in module salt.modules.apache*), 1210

`servers()` (*in module salt.modules.napalm_ntp*), 2177

`service()` (*in module salt.states.firewalld*), 3453

`service_absent()` (*in module salt.states.keystone*), 3519

`service_absent()` (*in module salt.states.kubernetes*), 3530

`service_create()` (*in module salt.modules.dockercompose*), 1613

`service_create()` (*in module salt.modules.keystone*), 1933

`service_create()` (*in module salt.modules.keystoneng*), 1942

`service_create()` (*in module salt.modules.swarm*), 2572

`service_delete()` (*in module salt.modules.keystone*), 1933

`service_delete()` (*in module salt.modules.keystoneng*), 1942

`service_disable()` (*in module salt.modules.netscaler*), 2212

`service_enable()` (*in module salt.modules.netscaler*), 2212

`service_exists()` (*in module salt.modules.netscaler*), 2212

`service_exists()` (*in module salt.states.panos*), 3629

`service_get()` (*in module salt.modules.keystone*), 1933

`service_get()` (*in module salt.modules.keystoneng*), 1942

`service_group_exists()` (*in module salt.states.panos*), 3629

`service_highstate()` (*in module salt.modules.introspect*), 1880

`service_list()` (*in module salt.modules.keystone*), 1934

`service_list()` (*in module salt.modules.keystoneng*), 1942

`service_list()` (*in module salt.proxy.dummy*), 3057

`service_list()` (*in module salt.proxy.rest_sample*), 3085

`service_list()` (*in module salt.proxy.ssh_sample*), 3086

`service_present()` (*in module salt.states.keystone*), 3519

`service_present()` (*in module salt.states.kubernetes*), 3530

`service_remove()` (*in module salt.modules.dockercompose*), 1613

`service_restart()` (*in module salt.modules.vsphere*), 2708

`service_restart()` (*in module salt.proxy.dummy*), 3057

`service_restart()` (*in module salt.proxy.rest_sample*), 3085

`service_restart()` (*in module salt.proxy.ssh_sample*), 3086

`service_search()` (*in module salt.modules.keystoneng*), 1942

`service_set_tag()` (*in module salt.modules.dockercompose*), 1614

`service_start()` (*in module salt.modules.vsphere*), 2709

`service_start()` (*in module salt.proxy.dummy*), 3057

`service_start()` (*in module salt.proxy.rest_sample*), 3085

`service_start()` (*in module salt.proxy.ssh_sample*), 3086

`service_status()` (*in module salt.modules.nagios_rpc*), 2113

`service_status()` (*in module salt.proxy.dummy*), 3057

`service_status()` (*in module*

salt.proxy.rest_sample), 3085
service_stop() (in module salt.modules.vsphere), 2710
service_stop() (in module salt.proxy.dummy), 3057
service_stop() (in module salt.proxy.rest_sample), 3085
service_stop() (in module salt.proxy.ssh_sample), 3087
service_up() (in module salt.modules.netscaler), 2213
service_update() (in module salt.modules.keystoneng), 1942
service_upsert() (in module salt.modules.dockercompose), 1614
servicegroup_add() (in module salt.modules.netscaler), 2213
servicegroup_delete() (in module salt.modules.netscaler), 2213
servicegroup_exists() (in module salt.modules.netscaler), 2213
servicegroup_server_add() (in module salt.modules.netscaler), 2213
servicegroup_server_delete() (in module salt.modules.netscaler), 2213
servicegroup_server_disable() (in module salt.modules.netscaler), 2213
servicegroup_server_enable() (in module salt.modules.netscaler), 2213
servicegroup_server_exists() (in module salt.modules.netscaler), 2214
servicegroup_server_up() (in module salt.modules.netscaler), 2214
services() (in module salt.modules.kubernetesmod), 1950
services() (in module salt.modules.riak), 2446
session_create() (in module salt.modules.consul), 1571
session_destroy() (in module salt.modules.consul), 1572
session_info() (in module salt.modules.consul), 1572
session_list() (in module salt.modules.consul), 1572
sessions() (in module salt.modules.tomcat), 2620
set() (in module salt.modules.grains), 1836
set() (in module salt.modules.zfs), 2904
set() (in module salt.modules.zookeeper), 2915
set() (in module salt.modules.zpool), 2922
set() (in module salt.states.debconfmod), 3363
set() (in module salt.states.stateconf), 3727
set_() (in module salt.modules.alternatives), 1207
set_() (in module salt.modules.debconfmod), 1595
set_() (in module salt.modules.etcd_mod), 1697
set_() (in module salt.modules.gnomedesktop), 1816
set_() (in module salt.modules.logrotate), 1983
set_() (in module salt.modules.memcached), 2077
set_() (in module salt.modules.openstack_config), 2277
set_() (in module salt.modules.parted_partition), 2327
set_() (in module salt.modules.quota), 2412
set_() (in module salt.modules.sdb), 2486
set_() (in module salt.modules.sysrc), 2588
set_() (in module salt.modules.win_lgpo), 2770
set_() (in module salt.runners.sdb), 3149
set_() (in module salt.sdb.cache), 3162
set_() (in module salt.sdb.consul), 3163
set_() (in module salt.sdb.couchdb), 3164
set_() (in module salt.sdb.env), 3165
set_() (in module salt.sdb.etcd_db), 3165
set_() (in module salt.sdb.keyring_db), 3166
set_() (in module salt.sdb.memcached), 3167
set_() (in module salt.sdb.redis_sdb), 3167
set_() (in module salt.sdb.rest), 3168
set_() (in module salt.sdb.sqlite3), 3169
set_() (in module salt.sdb.vault), 3170
set_() (in module salt.sdb.yaml), 3171
set_() (in module salt.states.alternatives), 3184
set_() (in module salt.states.eselect), 3404
set_() (in module salt.states.etcd_mod), 3407
set_() (in module salt.states.logrotate), 3541
set_() (in module salt.states.win_lgpo), 3771
set_() (in module salt.thorium.reg), 3827
set_absent() (in module salt.states.ipset), 3501
set_acls() (in module salt.modules.zookeeper), 2916
set_advanced_configs() (in module salt.modules.vsphere), 2711
set_agent_settings() (in module salt.modules.win_snmp), 2805
set_app() (in module salt.states.win_iis), 3768
set_attribute() (in module salt.modules.boto_ec2), 1373
set_attribute() (in module salt.modules.xml), 2862
set_attributes() (in module salt.modules.boto_elb), 1389
set_attributes() (in module salt.modules.boto_sqs), 1446
set_attributes() (in module salt.modules.win_file), 2742
set_auth_key() (in module salt.modules.ssh), 2541
set_auth_key_from_file() (in module salt.modules.ssh), 2541
set_auth_traps_enabled() (in module salt.modules.win_snmp), 2805
set_authentication_profile() (in module salt.modules.panos), 2316
set_automaster() (in module salt.modules.mount), 2093
set_autostart() (in module salt.modules.virt), 2669
set_backend_policy() (in module salt.modules.boto_elb), 1390

`set_binary_path()` (*in module salt.modules.syslog_ng*), 2581
`set_blob_properties()` (*in module salt.cloud.clouds.msazure*), 1082
`set_blob_service_properties()` (*in module salt.cloud.clouds.msazure*), 1083
`set_boot_arch()` (*in module salt.modules.mac_system*), 2051
`set_bootdev()` (*in module salt.modules.ipmi*), 1890
`set_ca_path()` (*in module salt.modules.tls*), 2618
`set_catalog()` (*in module salt.modules.mac_softwareupdate*), 2047
`set_cflags()` (*in module salt.modules.makeconf*), 2064
`set_change()` (*in module salt.modules.bsd_shadow*), 1465
`set_change()` (*in module salt.modules.mac_shadow*), 2043
`set_change_request_state()` (*in module salt.modules.servicenow*), 2496
`set_channel_access()` (*in module salt.modules.ipmi*), 1891
`set_chassis_datacenter()` (*in module salt.modules.dracr*), 1670
`set_chassis_location()` (*in module salt.modules.dracr*), 1670
`set_chassis_name()` (*in module salt.modules.dracr*), 1671
`set_chost()` (*in module salt.modules.makeconf*), 2064
`set_coalesce()` (*in module salt.modules.ethtool*), 1699
`set_community_names()` (*in module salt.modules.win_snmp*), 2806
`set_computer_desc()` (*in module salt.modules.system*), 2590
`set_computer_desc()` (*in module salt.modules.win_system*), 2816
`set_computer_name()` (*in module salt.modules.mac_system*), 2051
`set_computer_name()` (*in module salt.modules.system*), 2590
`set_computer_name()` (*in module salt.modules.win_system*), 2817
`set_computer_policy()` (*in module salt.modules.win_lgpo*), 2771
`set_computer_sleep()` (*in module salt.modules.mac_power*), 2036
`set_config()` (*in module salt.modules.dnsmasq*), 1607
`set_config()` (*in module salt.modules.snapper*), 2546
`set_config()` (*in module salt.modules.trafficserver*), 2624
`set_config()` (*in module salt.states.panos*), 3630
`set_config_file()` (*in module salt.modules.syslog_ng*), 2581
`set_config_modify()` (*in module salt.proxy.cimc*), 3052
`set_connection_ip_list()` (*in module salt.modules.win_smtp_server*), 2803
`set_container_setting()` (*in module salt.modules.win_iis*), 2762
`set_coredump_network_config()` (*in module salt.modules.vsphere*), 2711
`set_custom()` (*in module salt.modules.namecheap_domains_dns*), 2114
`set_cxxflags()` (*in module salt.modules.makeconf*), 2065
`set_data_value()` (*in module salt.modules.cisconso*), 1519
`set_data_value()` (*in module salt.proxy.cisconso*), 3056
`set_date()` (*in module salt.modules.mac_timezone*), 2055
`set_default()` (*in module salt.modules.namecheap_domains_dns*), 2114
`set_default()` (*in module salt.modules.rvm*), 2458
`set_default_keychain()` (*in module salt.modules.mac_keychain*), 2029
`set_default_policy_version()` (*in module salt.modules.boto_iam*), 1405
`set_default_policy_version()` (*in module salt.modules.boto_iot*), 1411
`set_default_zone()` (*in module salt.modules.firewalld*), 1737
`set_dhcp_all()` (*in module salt.modules.win_ip*), 2765
`set_dhcp_dns()` (*in module salt.modules.win_ip*), 2765
`set_dhcp_ip()` (*in module salt.modules.win_ip*), 2765
`set_dhcp_linklocal_all()` (*in module salt.modules.nilrt_ip*), 2250
`set_dhcp_only_all()` (*in module salt.modules.nilrt_ip*), 2250
`set_dict_key_value`
  `jinja filters`, 438
`set_disable_keyboard_on_lock()` (*in module salt.modules.mac_system*), 2051
`set_disk_timeout()` (*in module salt.modules.win_powercfg*), 2788
`set_display_sleep()` (*in module salt.modules.mac_power*), 2036
`set_dns()` (*in module salt.modules.lxc*), 2000
`set_dns_dracname()` (*in module salt.modules.dracr*), 1671
`set_emerge_default_opts()` (*in module salt.modules.makeconf*), 2065
`set_env()` (*in module salt.modules.cron*), 1581
`set_ethercat()` (*in module salt.modules.nilrt_ip*), 2250

`set_expire()` (*in module salt.modules.bsd_shadow*), 1466
`set_expire()` (*in module salt.modules.mac_shadow*), 2043
`set_expire()` (*in module salt.modules.win_shadow*), 2801
`set_file()` (*in module salt.modules.debconfmod*), 1595
`set_file()` (*in module salt.states.debconfmod*), 3364
`set_filesystems()` (*in module salt.modules.mount*), 2093
`set_firewall_settings()` (*in module salt.modules.win_firewall*), 2749
`set_fstab()` (*in module salt.modules.mount*), 2094
`set_ftp_proxy()` (*in module salt.modules.proxy*), 2388
`set_general()` (*in module salt.modules.dracr*), 1671
`set_gentoo_mirrors()` (*in module salt.modules.makeconf*), 2065
`set_harddisk_sleep()` (*in module salt.modules.mac_power*), 2036
`set_health_check()` (*in module salt.modules.boto_elb*), 1390
`set_hibernate_timeout()` (*in module salt.modules.win_powercfg*), 2788
`set_host()` (*in module salt.modules.hosts*), 1850
`set_hostname()` (*in module salt.modules.cimc*), 1511
`set_hostname()` (*in module salt.modules.junos*), 1918
`set_hostname()` (*in module salt.modules.panos*), 2316
`set_hostname()` (*in module salt.modules.win_system*), 2817
`set_hostname()` (*in module salt.states.junos*), 3514
`set_hosts()` (*in module salt.modules.namecheap_domains_dns*), 2114
`set_http_port()` (*in module salt.modules.ilo*), 1854
`set_http_proxy()` (*in module salt.modules.proxy*), 2388
`set_https_port()` (*in module salt.modules.ilo*), 1854
`set_https_proxy()` (*in module salt.modules.proxy*), 2388
`set_hwclock()` (*in module salt.modules.mac_timezone*), 2055
`set_hwclock()` (*in module salt.modules.timezone*), 2609
`set_hwclock()` (*in module salt.modules.win_timezone*), 2832
`set_id()` (*in module salt.modules.parted_partition*), 2328
`set_identify()` (*in module salt.modules.ipmi*), 1892
`set_identity_pool_roles()` (*in module salt.modules.boto_cognitoidentity*), 1357
`set_inactdays()` (*in module salt.modules.mac_shadow*), 2044
`set_instances()` (*in module salt.modules.boto_elb*), 1390
`set_is_polling()` (*in module salt.modules.solr*), 2532
`set_job()` (*in module salt.modules.cron*), 1581
`set_job()` (*in module salt.modules.incron*), 1856
`set_key()` (*in module salt.modules.redismod*), 2432
`set_known_host()` (*in module salt.modules.ssh*), 2541
`set_lcm_config()` (*in module salt.modules.win_dsc*), 2732
`set_linklocal_only_all()` (*in module salt.modules.nilrt_ip*), 2250
`set_listener_policy()` (*in module salt.modules.boto_elb*), 1390
`set_locale()` (*in module salt.modules.localemod*), 1980
`set_log_format()` (*in module salt.modules.win_smtp_server*), 2803
`set_logging_levels()` (*in module salt.modules.cimc*), 1511
`set_logging_settings()` (*in module salt.modules.win_firewall*), 2750
`set_main()` (*in module salt.modules.postfix*), 2373
`set_makeopts()` (*in module salt.modules.makeconf*), 2065
`set_management_http()` (*in module salt.modules.panos*), 2316
`set_management_https()` (*in module salt.modules.panos*), 2316
`set_management_icmp()` (*in module salt.modules.panos*), 2317
`set_management_ocsp()` (*in module salt.modules.panos*), 2317
`set_management_snmp()` (*in module salt.modules.panos*), 2317
`set_management_ssh()` (*in module salt.modules.panos*), 2317
`set_management_telnet()` (*in module salt.modules.panos*), 2317
`set_master()` (*in module salt.modules.postfix*), 2373
`set_maxdays()` (*in module salt.modules.mac_shadow*), 2044
`set_maxdays()` (*in module salt.modules.solaris_shadow*), 2517
`set_mindays()` (*in module salt.modules.mac_shadow*), 2044
`set_mindays()` (*in module salt.modules.solaris_shadow*), 2517
`set_mode()` (*in module salt.modules.file*), 1728
`set_mode()` (*in module salt.modules.iwtools*), 1902
`set_mode()` (*in module salt.modules.win_file*), 2742
`set_monitor_timeout()` (*in module salt.modules.win_powercfg*), 2789
`set_network()` (*in module salt.modules.drac*), 1664

set_network() (*in module salt.modules.dracr*), 1671
set_niccfg() (*in module salt.modules.dracr*), 1671
set_nicvlan() (*in module salt.modules.dracr*), 1671
set_ntp_authentication() (*in module salt.modules.panos*), 2318
set_ntp_config() (*in module salt.modules.vsphere*), 2711
set_ntp_server() (*in module salt.modules.cimc*), 1512
set_ntp_servers() (*in module salt.modules.panos*), 2318
set_offload() (*in module salt.modules.ethtool*), 1699
set_op_version() (*in module salt.modules.glusterfs*), 1814
set_option() (*in module salt.modules.csf*), 1583
set_option() (*in module salt.modules.ini_manage*), 1874
set_output_volume() (*in module salt.modules.mac_desktop*), 2026
set_parameter() (*in module salt.modules.lxc*), 2000
set_parameters() (*in module salt.modules.syslog_ng*), 2581
set_pass() (*in module salt.states.lxc*), 3547
set_password() (*in module salt.modules.bsd_shadow*), 1466
set_password() (*in module salt.modules.lxc*), 2001
set_password() (*in module salt.modules.mac_shadow*), 2044
set_password() (*in module salt.modules.solaris_shadow*), 2517
set_password() (*in module salt.modules.win_shadow*), 2801
set_password() (*in module salt.proxy.nxos*), 3076
set_pause() (*in module salt.runners.state*), 3154
set_peers() (*in module salt.modules.napalm_ntp*), 2177
set_permissions() (*in module salt.modules.drac*), 1664
set_permissions() (*in module salt.modules.dracr*), 1671
set_permissions() (*in module salt.modules.rabbitmq*), 2416
set_permitted_ip() (*in module salt.modules.panos*), 2318
set_perms() (*in module salt.modules.win_file*), 2743
set_pixel() (*in module salt.modules.sensehat*), 2493
set_pixels() (*in module salt.modules.sensehat*), 2493
set_policy() (*in module salt.modules.iptables*), 1902
set_policy() (*in module salt.modules.rabbitmq*), 2416
set_policy() (*in module salt.states.iptables*), 3507
set_power() (*in module salt.modules.ipmi*), 1893
set_power_configuration() (*in module salt.modules.cimc*), 1512
set_present() (*in module salt.states.ipset*), 3501
set_probes() (*in module salt.modules.napalm_probes*), 2181
set_prod_state() (*in module salt.modules.zenoss*), 2898
set_property() (*in module salt.modules.zonecfg*), 2912
set_proxy_bypass() (*in module salt.modules.proxy*), 2389
set_proxy_win() (*in module salt.modules.proxy*), 2389
set_public_lan() (*in module salt.cloud.clouds.profitbricks*), 1118
set_pv_args() (*in module salt.cloud.clouds.xen*), 1151
set_quota_volume() (*in module salt.modules.glusterfs*), 1814
set_reboot_required_witnessed() (*in module salt.modules.system*), 2590
set_reboot_required_witnessed() (*in module salt.modules.win_system*), 2817
set_relay_ip_list() (*in module salt.modules.win_smtp_server*), 2803
set_remote_events() (*in module salt.modules.mac_system*), 2052
set_remote_login() (*in module salt.modules.mac_system*), 2052
set_replication_enabled() (*in module salt.modules.solr*), 2532
set_restart_delay() (*in module salt.modules.mac_system*), 2052
set_restart_freeze() (*in module salt.modules.mac_power*), 2036
set_restart_power_failure() (*in module salt.modules.mac_power*), 2037
set_ring() (*in module salt.modules.ethtool*), 1699
set_role() (*in module salt.proxy.nxos*), 3076
set_salt_view() (*in module salt.returners.couchdb_return*), 318
set_security_groups() (*in module salt.modules.boto_efs*), 1378
set_selections() (*in module salt.modules.aptpkg*), 1224
set_selinux_context() (*in module salt.modules.file*), 1729
set_server_setting() (*in module salt.modules.win_smtp_server*), 2804
set_servers() (*in module salt.modules.napalm_ntp*), 2177
set_servers() (*in module salt.modules.win_ntp*), 2776
set_service_policy() (*in module salt.modules.vsphere*), 2712
set_settings() (*in module salt.modules.win_firewall*), 2751

`set_sleep()` (*in module salt.modules.mac_power*), 2037
`set_sleep_on_power_button()` (*in module salt.modules.mac_power*), 2037
`set_slotname()` (*in module salt.modules.dracr*), 1672
`set_snmp()` (*in module salt.modules.drac*), 1664
`set_snmp()` (*in module salt.modules.dracr*), 1672
`set_special()` (*in module salt.modules.cron*), 1581
`set_ssh_key()` (*in module salt.modules.ilo*), 1854
`set_ssh_port()` (*in module salt.modules.ilo*), 1854
`set_standby_timeout()` (*in module salt.modules.win_powercfg*), 2789
`set_startup_disk()` (*in module salt.modules.mac_system*), 2052
`set_state()` (*in module salt.modules.haproxyconn*), 1841
`set_state()` (*in module salt.modules.win_firewall*), 2752
`set_static_all()` (*in module salt.modules.nilrt_ip*), 2250
`set_static_dns()` (*in module salt.modules.win_ip*), 2765
`set_static_ip()` (*in module salt.modules.win_ip*), 2765
`set_storage_container_acl()` (*in module salt.cloud.clouds.msazure*), 1083
`set_storage_container_metadata()` (*in module salt.cloud.clouds.msazure*), 1083
`set_subnet_name()` (*in module salt.modules.mac_system*), 2052
`set_sync()` (*in module salt.modules.makeconf*), 2066
`set_sys()` (*in module salt.modules.keyboard*), 1928
`set_syslog_config()` (*in module salt.modules.vsphere*), 2713
`set_syslog_server()` (*in module salt.modules.cimc*), 1513
`set_system_date()` (*in module salt.modules.system*), 2590
`set_system_date()` (*in module salt.modules.win_system*), 2817
`set_system_date_time()` (*in module salt.modules.system*), 2590
`set_system_date_time()` (*in module salt.modules.win_system*), 2817
`set_system_time()` (*in module salt.modules.system*), 2591
`set_system_time()` (*in module salt.modules.win_system*), 2818
`set_tags()` (*in module salt.cloud.clouds.ec2*), 1051
`set_tags()` (*in module salt.modules.boto_elb*), 1390
`set_tags()` (*in module salt.modules.boto_secgroup*), 1443
`set_target()` (*in module salt.modules.aliases*), 1206
`set_target()` (*in module salt.modules.eselect*), 1695
`set_template()` (*in module salt.modules.debconfmod*), 1595
`set_time()` (*in module salt.modules.mac_timezone*), 2055
`set_time_server()` (*in module salt.modules.mac_timezone*), 2055
`set_timeout()` (*in module salt.states.win_powercfg*), 3776
`set_timezone()` (*in module salt.modules.panos*), 2319
`set_upstream()` (*in module salt.modules.rabbitmq*), 2416
`set_user()` (*in module salt.modules.cimc*), 1513
`set_user_access()` (*in module salt.modules.ipmi*), 1893
`set_user_name()` (*in module salt.modules.ipmi*), 1894
`set_user_password()` (*in module salt.modules.influxdbmod*), 1861
`set_user_password()` (*in module salt.modules.ipmi*), 1894
`set_user_policy()` (*in module salt.modules.win_lgpo*), 2771
`set_user_tags()` (*in module salt.modules.rabbitmq*), 2417
`set_users()` (*in module salt.modules.napalm_users*), 2188
`set_using_network_time()` (*in module salt.modules.mac_timezone*), 2056
`set_value()` (*in module salt.modules.reg*), 2437
`set_value()` (*in module salt.modules.xml*), 2862
`set_var()` (*in module salt.modules.makeconf*), 2066
`set_vfstab()` (*in module salt.modules.mount*), 2094
`set_vm_ip()` (*in module salt.cloud.clouds.xen*), 1151
`set_vm_status()` (*in module salt.cloud.clouds.proxmox*), 1121
`set_volumes_tags()` (*in module salt.modules.boto_ec2*), 1374
`set_wake_on_modem()` (*in module salt.modules.mac_power*), 2037
`set_wake_on_network()` (*in module salt.modules.mac_power*), 2037
`set_warndays()` (*in module salt.modules.mac_shadow*), 2045
`set_warndays()` (*in module salt.modules.solaris_shadow*), 2517
`set_webapp_settings()` (*in module salt.modules.win_iis*), 2762
`set_webconfiguration_settings()` (*in module salt.modules.win_iis*), 2763
`set_weight()` (*in module salt.modules.haproxyconn*), 1842
`set_wu_settings()` (*in module salt.modules.win_wua*), 2845
`set_x()` (*in module salt.modules.keyboard*), 1928

set_zone() (*in module salt.modules.mac_timezone*), 2056
set_zone() (*in module salt.modules.timezone*), 2609
set_zone() (*in module salt.modules.win_timezone*), 2832
setClockFormat() (*in module salt.modules.gnomedesktop*), 1816
setClockShowDate() (*in module salt.modules.gnomedesktop*), 1816
setenforce() (*in module salt.modules.selinux*), 2491
setenv() (*in module salt.modules.environ*), 1693
setenv() (*in module salt.states.environ*), 3404
setIdleActivation() (*in module salt.modules.gnomedesktop*), 1816
setIdleDelay() (*in module salt.modules.gnomedesktop*), 1816
setmem() (*in module salt.modules.smartos_virt*), 2504
setmem() (*in module salt.modules.virt*), 2670
setmem() (*in module salt.modules.xapi_virt*), 2855
setpassword() (*in module salt.modules.win_useradd*), 2837
setsebool() (*in module salt.modules.selinux*), 2491
setsebools() (*in module salt.modules.selinux*), 2491
setsemod() (*in module salt.modules.selinux*), 2491
setup_clouds() (*in module salt.modules.glanceng*), 1811
setup_clouds() (*in module salt.modules.keystoneng*), 1943
setup_clouds() (*in module salt.modules.neutronng*), 2241
setval() (*in module salt.modules.environ*), 1694
setval() (*in module salt.modules.grains*), 1836
setval() (*in module salt.modules.napalm_formula*), 2154
setvals() (*in module salt.modules.grains*), 1836
setvalue() (*in module salt.modules.augeas_cfg*), 1246
setvcpus() (*in module salt.modules.virt*), 2670
setvcpus() (*in module salt.modules.xapi_virt*), 2855
sha256
  jinja filters, 440
sha256_digest() (*in module salt.modules.hashutil*), 1845
sha512
  jinja filters, 440
sha512_digest() (*in module salt.modules.hashutil*), 1845
shadow_hash() (*in module salt.modules.mod_random*), 2083
shared_memory() (*in module salt.modules.osquery*), 2294
SharedCommittedPages (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATION attribute*), 2810
shell() (*in module salt.grains.extra*), 1179
shell() (*in module salt.modules.cmdmod*), 1550
shell_history() (*in module salt.modules.osquery*), 2295
shell_info() (*in module salt.modules.cmdmod*), 1553
shells() (*in module salt.modules.cmdmod*), 1554
shortcut() (*in module salt.states.file*), 3449
show() (*in module salt.modules.aptpkg*), 1225
show() (*in module salt.modules.bridge*), 1465
show() (*in module salt.modules.cpan*), 1579
show() (*in module salt.modules.debconfmod*), 1596
show() (*in module salt.modules.freebsd_sysctl*), 1738
show() (*in module salt.modules.linux_sysctl*), 1979
show() (*in module salt.modules.mac_service*), 2040
show() (*in module salt.modules.mac_sysctl*), 2048
show() (*in module salt.modules.napalm_route*), 2183
show() (*in module salt.modules.netbsd_sysctl*), 2202
show() (*in module salt.modules.nova*), 2256
show() (*in module salt.modules.nxos_api*), 2269
show() (*in module salt.modules.openbsd_sysctl*), 2270
show() (*in module salt.modules.pf*), 2337
show() (*in module salt.modules.runit*), 2455
show() (*in module salt.modules.smartos_imgadm*), 2501
show() (*in module salt.modules.solaris_fmadm*), 2515
show() (*in module salt.modules.systemd_service*), 2596
show_address() (*in module salt.cloud.clouds.gce*), 1057
show_affinity_group() (*in module salt.cloud.clouds.msazure*), 1083
show_all_categories() (*in module salt.cloud.clouds.softlayer_hw*), 1129
show_all_prices() (*in module salt.cloud.clouds.softlayer_hw*), 1129
show_args() (*in module salt.modules.logadm*), 1982
show_artifact() (*in module salt.modules.rallydev*), 2419
show_backends() (*in module salt.modules.haproxyconn*), 1842
show_blob_properties() (*in module salt.cloud.clouds.msazure*), 1084
show_blob_service_properties() (*in module salt.cloud.clouds.msazure*), 1084
show_cert() (*in module salt.runners.venafiapi*), 3158
show_coalesce() (*in module salt.modules.ethtool*), 1699
show_conf() (*in module salt.modules.logadm*), 1982
show_conf() (*in module salt.modules.logrotate*), 1984
show_config() (*in module salt.modules.cpan*), 1579
show_config() (*in module salt.modules.freebsdjail*), 1740
show_configmap() (*in module salt.modules.kubernetesmod*), 1950
show_csrs() (*in module salt.runners.digicertapi*), 3106
show_current() (*in module salt.modules.alternatives*), 1207

show_dbs() (*in module salt.modules.oracle*), 2287
show_delvol_on_destroy() (*in module salt.cloud.clouds.ec2*), 1051
show_deployment() (*in module salt.cloud.clouds.msazure*), 1084
show_deployment() (*in module salt.modules.kubernetesmod*), 1950
show_detailed_monitoring() (*in module salt.cloud.clouds.ec2*), 1051
show_disk() (*in module salt.cloud.clouds.aliyun*), 1034
show_disk() (*in module salt.cloud.clouds.gce*), 1057
show_disk() (*in module salt.cloud.clouds.msazure*), 1084
show_disk() (*in module salt.cloud.clouds.tencentcloud*), 1132
show_driver() (*in module salt.modules.ethtool*), 1699
show_employee() (*in module salt.modules.bamboohr*), 1279
show_env() (*in module salt.modules.oracle*), 2287
show_firewall() (*in module salt.modules.neutron*), 2233
show_firewall_rule() (*in module salt.modules.neutron*), 2233
show_floating_ip() (*in module salt.cloud.clouds.digitalocean*), 1042
show_floatingip() (*in module salt.modules.neutron*), 2233
show_frontends() (*in module salt.modules.haproxyconn*), 1842
show_fwrule() (*in module salt.cloud.clouds.gce*), 1057
show_hc() (*in module salt.cloud.clouds.gce*), 1058
show_highstate() (*in module salt.modules.state*), 2554
show_ikepolicy() (*in module salt.modules.neutron*), 2233
show_image() (*in module salt.cloud.clouds.aliyun*), 1034
show_image() (*in module salt.cloud.clouds.ec2*), 1051
show_image() (*in module salt.cloud.clouds.parallels*), 1115
show_image() (*in module salt.cloud.clouds.qingcloud*), 1123
show_image() (*in module salt.cloud.clouds.tencentcloud*), 1132
show_image() (*in module salt.cloud.clouds.virtualbox*), 1135
show_image() (*in module salt.modules.sensehat*), 2493
show_input_endpoint() (*in module salt.cloud.clouds.msazure*), 1084
show_instance() (*in module salt.cloud.clouds.aliyun*), 1034
show_instance() (*in module salt.cloud.clouds.azurearm*), 1037
show_instance() (*in module salt.cloud.clouds.cloudstack*), 1039
show_instance() (*in module salt.cloud.clouds.digitalocean*), 1042
show_instance() (*in module salt.cloud.clouds.dimensiondata*), 1044
show_instance() (*in module salt.cloud.clouds.ec2*), 1051
show_instance() (*in module salt.cloud.clouds.gce*), 1058
show_instance() (*in module salt.cloud.clouds.gogrid*), 1061
show_instance() (*in module salt.cloud.clouds.joyent*), 1064
show_instance() (*in module salt.cloud.clouds.linode*), 1069
show_instance() (*in module salt.cloud.clouds.lxc*), 1071
show_instance() (*in module salt.cloud.clouds.msazure*), 1084
show_instance() (*in module salt.cloud.clouds.oneandone*), 1089
show_instance() (*in module salt.cloud.clouds.opennebula*), 1098
show_instance() (*in module salt.cloud.clouds.openstack*), 1113
show_instance() (*in module salt.cloud.clouds.parallels*), 1115
show_instance() (*in module salt.cloud.clouds.profitbricks*), 1118
show_instance() (*in module salt.cloud.clouds.proxmox*), 1121
show_instance() (*in module salt.cloud.clouds.qingcloud*), 1123
show_instance() (*in module salt.cloud.clouds.saltify*), 1126
show_instance() (*in module salt.cloud.clouds.scaleway*), 1127
show_instance() (*in module salt.cloud.clouds.softlayer*), 1128
show_instance() (*in module salt.cloud.clouds.softlayer_hw*), 1130
show_instance() (*in module salt.cloud.clouds.tencentcloud*), 1132
show_instance() (*in module salt.cloud.clouds.vagrant*), 1134
show_instance() (*in module salt.cloud.clouds.vmware*), 1146
show_instance() (*in module salt.cloud.clouds.vultrpy*), 1148
show_instance() (*in module salt.cloud.clouds.xen*), 1151
show_ipsec_site_connection() (*in module salt.modules.neutron*), 2234
show_ipsecpolicy() (*in module*

salt.modules.neutron), 2234
show_item() (*in module salt.modules.rallydev*), 2419
show_jid
  conf/master, 56
show_key() (*in module salt.cloud.clouds.joyent*), 1064
show_keypair() (*in module salt.cloud.clouds.digitalocean*), 1042
show_keypair() (*in module salt.cloud.clouds.ec2*), 1052
show_lb() (*in module salt.cloud.clouds.gce*), 1058
show_letter() (*in module salt.modules.sensehat*), 2493
show_link() (*in module salt.modules.alternatives*), 1207
show_low_sls() (*in module salt.modules.state*), 2554
show_lowstate() (*in module salt.modules.state*), 2554
show_main() (*in module salt.modules.postfix*), 2373
show_management_certificate() (*in module salt.cloud.clouds.msazure*), 1085
show_master() (*in module salt.modules.postfix*), 2373
show_message() (*in module salt.modules.sensehat*), 2493
show_namespace() (*in module salt.modules.kubernetesmod*), 1950
show_network() (*in module salt.cloud.clouds.gce*), 1058
show_network() (*in module salt.modules.neutron*), 2234
show_next_fire_time() (*in module salt.modules.schedule*), 2483
show_notification() (*in module salt.states.test*), 3735
show_offload() (*in module salt.modules.ethtool*), 1699
show_organization() (*in module salt.runners.digicertapi*), 3107
show_pillar() (*in module salt.modules.oracle*), 2287
show_pillar() (*in module salt.runners.pillar*), 3138
show_pod() (*in module salt.modules.kubernetesmod*), 1950
show_policies() (*in module salt.runners.vault*), 3157
show_port() (*in module salt.modules.neutron*), 2234
show_pricing() (*in module salt.cloud.clouds.digitalocean*), 1042
show_pricing() (*in module salt.cloud.clouds.ec2*), 1052
show_pricing() (*in module salt.cloud.clouds.gce*), 1058
show_pricing() (*in module salt.cloud.clouds.linode*), 1070
show_pricing() (*in module salt.cloud.clouds.softlayer_hw*), 1130
show_queue() (*in module salt.modules.postfix*), 2373
show_quota() (*in module salt.modules.neutron*), 2234
show_ring() (*in module salt.modules.ethtool*), 1699
show_router() (*in module salt.modules.neutron*), 2235
show_rsa() (*in module salt.runners.digicertapi*), 3107
show_run() (*in module salt.proxy.nxos*), 3076
show_secret() (*in module salt.modules.kubernetesmod*), 1951
show_security_group() (*in module salt.modules.neutron*), 2235
show_security_group_rule() (*in module salt.modules.neutron*), 2235
show_service() (*in module salt.cloud.clouds.msazure*), 1085
show_service() (*in module salt.modules.kubernetesmod*), 1951
show_service_certificate() (*in module salt.cloud.clouds.msazure*), 1085
show_sls() (*in module salt.modules.state*), 2554
show_sls() (*in module salt.modules.winrepo*), 2791
show_snapshot() (*in module salt.cloud.clouds.gce*), 1058
show_stack() (*in module salt.modules.heat*), 1847
show_state_usage() (*in module salt.modules.state*), 2555
show_states() (*in module salt.modules.state*), 2555
show_storage() (*in module salt.cloud.clouds.msazure*), 1085
show_storage_container() (*in module salt.cloud.clouds.msazure*), 1085
show_storage_container_acl() (*in module salt.cloud.clouds.msazure*), 1085
show_storage_container_metadata() (*in module salt.cloud.clouds.msazure*), 1086
show_storage_keys() (*in module salt.cloud.clouds.msazure*), 1086
show_subnet() (*in module salt.modules.neutron*), 2235
show_subnetwork() (*in module salt.cloud.clouds.gce*), 1058
show_term_protect() (*in module salt.cloud.clouds.ec2*), 1052
show_timeout
  conf/master, 56
show_top() (*in module salt.modules.state*), 2555
show_top() (*in module salt.runners.pillar*), 3138
show_user() (*in module salt.modules.rallydev*), 2419
show_ver() (*in module salt.proxy.nxos*), 3076
show_volume() (*in module salt.cloud.clouds.ec2*), 1052
show_vpnservice() (*in module salt.modules.neutron*), 2235
showconfig() (*in module salt.modules.freebsdports*), 1746

showglobal() (*in module salt.modules.mysql*), 2104
showvariables() (*in module salt.modules.mysql*), 2105
shutdown() (*in module salt.cloud.clouds.proxmox*), 1121
shutdown() (*in module salt.cloud.clouds.xen*), 1151
shutdown() (*in module salt.modules.junos*), 1918
shutdown() (*in module salt.modules.mac_system*), 2053
shutdown() (*in module salt.modules.panos*), 2319
shutdown() (*in module salt.modules.redismod*), 2432
shutdown() (*in module salt.modules.smartos_virt*), 2504
shutdown() (*in module salt.modules.solaris_system*), 2518
shutdown() (*in module salt.modules.system*), 2591
shutdown() (*in module salt.modules.trafficserver*), 2624
shutdown() (*in module salt.modules.vagrant*), 2636
shutdown() (*in module salt.modules.virt*), 2670
shutdown() (*in module salt.modules.win_system*), 2818
shutdown() (*in module salt.modules.xapi_virt*), 2855
shutdown() (*in module salt.modules.zoneadm*), 2909
shutdown() (*in module salt.proxy.arista_pyeapi*), 3050
shutdown() (*in module salt.proxy.chronos*), 3052
shutdown() (*in module salt.proxy.cimc*), 3052
shutdown() (*in module salt.proxy.cisconso*), 3056
shutdown() (*in module salt.proxy.dummy*), 3057
shutdown() (*in module salt.proxy.esxi*), 3062
shutdown() (*in module salt.proxy.fx2*), 3064
shutdown() (*in module salt.proxy.junos*), 3065
shutdown() (*in module salt.proxy.marathon*), 3066
shutdown() (*in module salt.proxy.napalm*), 3069
shutdown() (*in module salt.proxy.netmiko_px*), 3073
shutdown() (*in module salt.proxy.nxos*), 3076
shutdown() (*in module salt.proxy.nxos_api*), 3078
shutdown() (*in module salt.proxy.panos*), 3082
shutdown() (*in module salt.proxy.philips_hue*), 3085
shutdown() (*in module salt.proxy.rest_sample*), 3086
shutdown() (*in module salt.proxy.ssh_sample*), 3087
shutdown() (*in module salt.states.junos*), 3514
shutdown() (*in module salt.states.trafficserver*), 3740
shutdown() (*in module salt.states.win_system*), 3783
shutdown_abort() (*in module salt.modules.win_system*), 2819
shutdown_hard() (*in module salt.modules.win_system*), 2819
SIException, 1875
sign() (*in module salt.cloud.clouds.ec2*), 1052
sign() (*in module salt.modules.gpg*), 1821
sign_remote_certificate() (*in module salt.modules.x509*), 2852
signal() (*in module salt.modules.apache*), 1211
signal() (*in module salt.modules.nginx*), 2248
signal() (*in module salt.modules.solr*), 2533
signal() (*in module salt.modules.tomcat*), 2621
signal_() (*in module salt.modules.dockermod*), 1656
signal_event() (*in module salt.cloud.clouds.profitbricks*), 1118
signal_job() (*in module salt.modules.saltutil*), 2473
SimpleParameter (*class in salt.modules.syslog_ng*), 2578
SimpleParameterValue (*class in salt.modules.syslog_ng*), 2579
single() (*in module salt.modules.state*), 2555
skip_job() (*in module salt.modules.schedule*), 2483
skip_nic() (*in module salt.modules.csf*), 1583
skip_nics() (*in module salt.modules.csf*), 1583
SlackClient (*class in salt.engines.slack*), 1166
slave_lag() (*in module salt.modules.mysql*), 2105
slaveof() (*in module salt.modules.redismod*), 2432
slaveof() (*in module salt.states.redismod*), 3688
SleekXMPPMUC (*class in salt.modules.xmpp*), 2863
sleep() (*in module salt.modules.mac_system*), 2053
sleep() (*in module salt.modules.test*), 2603
sleep() (*in module salt.runners.test*), 3156
SLS Module, **5403**
sls() (*in module salt.modules.dockermod*), 1657
sls() (*in module salt.modules.state*), 2555
sls_build() (*in module salt.modules.dockermod*), 1657
sls_exists() (*in module salt.modules.state*), 2557
sls_id() (*in module salt.modules.state*), 2557
sls_list
    conf/minion, 149
slugify() (*in module salt.modules.netbox*), 2201
smart_attributes() (*in module salt.modules.disk*), 1604
smbios_tables() (*in module salt.modules.osquery*), 2295
smembers() (*in module salt.modules.redismod*), 2432
snap_create() (*in module salt.modules.purefa*), 2399
snap_create() (*in module salt.modules.purefb*), 2403
snap_delete() (*in module salt.modules.purefa*), 2400
snap_delete() (*in module salt.modules.purefb*), 2404
snap_eradicate() (*in module salt.modules.purefa*), 2400
snap_eradicate() (*in module salt.modules.purefb*), 2404
snap_volume_create() (*in module salt.modules.purefa*), 2400
snapper_states
    conf/minion, 151
snapper_states_config
    conf/minion, 152
snapshot() (*in module salt.modules.parallels*), 2323
snapshot() (*in module salt.modules.virt*), 2671
snapshot() (*in module salt.modules.zfs*), 2904
snapshot() (*in module salt.states.virt*), 3754

snapshot() (*salt.modules.inspectlib.collector.Inspector method*), 1875
snapshot_absent() (*in module salt.states.zfs*), 3809
snapshot_create() (*in module salt.modules.elasticsearch*), 1691
snapshot_created() (*in module salt.states.boto_ec2*), 3278
snapshot_delete() (*in module salt.modules.elasticsearch*), 1692
snapshot_get() (*in module salt.modules.elasticsearch*), 1692
snapshot_id_to_name() (*in module salt.modules.parallels*), 2323
snapshot_name_to_id() (*in module salt.modules.parallels*), 2324
snapshot_present() (*in module salt.states.zfs*), 3809
snapshot_restore() (*in module salt.modules.elasticsearch*), 1692
snapshot_status() (*in module salt.modules.elasticsearch*), 1692
snapshots() (*in module salt.modules.inspector*), 1880
snapshots_all() (*in module salt.modules.lxd*), 2020
snapshots_create() (*in module salt.modules.lxd*), 2020
snapshots_delete() (*in module salt.modules.lxd*), 2021
snapshots_get() (*in module salt.modules.lxd*), 2021
sock_dir
    conf/master, 57
    conf/minion, 136
sock_pool_size
    conf/master, 62
SocketAppender (*in module salt.engines.logentries*), 1159
soft_kill() (*in module salt.modules.state*), 2557
soft_kill() (*in module salt.runners.state*), 3155
solo() (*in module salt.modules.chef*), 1499
solo() (*in module salt.states.chef*), 3337
sort_pkglist() (*in module salt.modules.pkg_resource*), 2350
source (*salt.engines.ircbot.Event attribute*), 1156
source (*salt.engines.ircbot.PrivEvent attribute*), 1156
source_absent() (*in module salt.states.smartos*), 3716
source_add() (*in module salt.modules.smartos_imgadm*), 2501
source_address
    conf/minion, 128
source_delete() (*in module salt.modules.smartos_imgadm*), 2501
source_interface_name
    conf/minion, 128
source_list() (*in module salt.modules.file*), 1729
source_present() (*in module salt.states.smartos*), 3716
source_publish_port
    conf/minion, 129
source_ret_port
    conf/minion, 129
sources() (*in module salt.modules.smartos_imgadm*), 2501
sources_add() (*in module salt.modules.gem*), 1750
sources_add() (*in module salt.states.gem*), 3454
sources_list() (*in module salt.modules.gem*), 1751
sources_remove() (*in module salt.modules.gem*), 1751
sources_remove() (*in module salt.states.gem*), 3454
SPF() (*in module salt.modules.dig*), 1602
SPF() (*in module salt.modules.dnsutil*), 1608
split() (*in module salt.modules.zpool*), 2923
split_option() (*in module salt.modules.csf*), 1583
spm command line option
    --log-file-level=LOG_LEVEL_LOGFILE, 991
    --log-file=LOG_FILE, 991
    -f, --force, 991
    -l LOG_LEVEL, --log-level=LOG_LEVEL, 991
    -y, --assume-yes, 991
    build, 991
    create_repo, 991
    files, 991
    info, 991
    install, 991
    local, 991
    remove, 991
    update_repo, 991
SqlBaseExtPillar (*class in salt.pillar.sql_base*), 3034
SQLCipherExtPillar (*class in salt.pillar.sqlcipher*), 3036
SQLite3ExtPillar (*class in salt.pillar.sqlite3*), 3037
sqlite_version() (*in module salt.modules.sqlite3*), 2538
sr_list() (*in module salt.cloud.clouds.xen*), 1152
ss() (*in module salt.modules.ps*), 2392
ssh() (*salt.netapi.NetapiClient method*), 3855
ssh_configured() (*in module salt.states.esxi*), 3413
ssh_identities_only
    conf/master, 67
ssh_interface() (*in module salt.cloud.clouds.dimensiondata*), 1044
ssh_interface() (*in module salt.cloud.clouds.ec2*), 1052
ssh_interface() (*in module salt.cloud.clouds.joyent*), 1064
ssh_interface() (*in module salt.cloud.clouds.openstack*), 1114

`ssh_interface()` (*in module salt.cloud.clouds.profitbricks*), 1118
`ssh_list_nodegroups`
  `conf/master`, 67
`ssh_log_file`
  `conf/master`, 67
`ssh_minion_opts`
  `conf/master`, 67
`ssh_passwd`
  `conf/master`, 65
`ssh_port`
  `conf/master`, 66
`ssh_priv_passwd`
  `conf/master`, 66
`ssh_scan_ports`
  `conf/master`, 66
`ssh_scan_timeout`
  `conf/master`, 66
`ssh_sudo`
  `conf/master`, 66
`ssh_timeout`
  `conf/master`, 66
`ssh_use_home_key`
  `conf/master`, 67
`ssh_user`
  `conf/master`, 66
`SSHClient` (*class in salt.client.ssh.client*), 3853
`ssl`
  `conf/master`, 73
  `conf/minion`, 159
`ssm_create_association()` (*in module salt.cloud.clouds.ec2*), 1052
`ssm_describe_association()` (*in module salt.cloud.clouds.ec2*), 1052
`stack()` (*in module salt.modules.test*), 2603
`start()` (*in module salt.cloud.clouds.aliyun*), 1034
`start()` (*in module salt.cloud.clouds.azurearm*), 1037
`start()` (*in module salt.cloud.clouds.digitalocean*), 1042
`start()` (*in module salt.cloud.clouds.dimensiondata*), 1044
`start()` (*in module salt.cloud.clouds.ec2*), 1052
`start()` (*in module salt.cloud.clouds.gce*), 1058
`start()` (*in module salt.cloud.clouds.gogrid*), 1061
`start()` (*in module salt.cloud.clouds.joyent*), 1064
`start()` (*in module salt.cloud.clouds.linode*), 1070
`start()` (*in module salt.cloud.clouds.oneandone*), 1089
`start()` (*in module salt.cloud.clouds.opennebula*), 1099
`start()` (*in module salt.cloud.clouds.parallels*), 1115
`start()` (*in module salt.cloud.clouds.profitbricks*), 1118
`start()` (*in module salt.cloud.clouds.proxmox*), 1121
`start()` (*in module salt.cloud.clouds.qingcloud*), 1124
`start()` (*in module salt.cloud.clouds.tencentcloud*), 1133
`start()` (*in module salt.cloud.clouds.virtualbox*), 1135
`start()` (*in module salt.cloud.clouds.vmware*), 1146
`start()` (*in module salt.cloud.clouds.vultrpy*), 1148
`start()` (*in module salt.cloud.clouds.xen*), 1152
`start()` (*in module salt.engines.docker_events*), 1153
`start()` (*in module salt.engines.fluent*), 1154
`start()` (*in module salt.engines.http_logstash*), 1155
`start()` (*in module salt.engines.ircbot*), 1156
`start()` (*in module salt.engines.junos_syslog*), 1158
`start()` (*in module salt.engines.logentries*), 1159
`start()` (*in module salt.engines.logstash_engine*), 1160
`start()` (*in module salt.engines.napalm_syslog*), 1162
`start()` (*in module salt.engines.reactor*), 1163
`start()` (*in module salt.engines.redis_sentinel*), 1164
`start()` (*in module salt.engines.slack*), 1168
`start()` (*in module salt.engines.sqs_events*), 1169
`start()` (*in module salt.engines.stalekey*), 1170
`start()` (*in module salt.engines.test*), 1170
`start()` (*in module salt.engines.thorium*), 1170
`start()` (*in module salt.engines.webhook*), 1170
`start()` (*in module salt.modules.bcache*), 1282
`start()` (*in module salt.modules.bluez_bluetooth*), 1300
`start()` (*in module salt.modules.daemontools*), 1586
`start()` (*in module salt.modules.debian_service*), 1599
`start()` (*in module salt.modules.dockercompose*), 1614
`start()` (*in module salt.modules.dummyproxy_service*), 1674
`start()` (*in module salt.modules.freebsdjail*), 1740
`start()` (*in module salt.modules.freebsdservice*), 1748
`start()` (*in module salt.modules.gentoo_service*), 1755
`start()` (*in module salt.modules.launchctl_service*), 1952
`start()` (*in module salt.modules.lxc*), 2001
`start()` (*in module salt.modules.mac_service*), 2040
`start()` (*in module salt.modules.monit*), 2090
`start()` (*in module salt.modules.netbsdservice*), 2204
`start()` (*in module salt.modules.nspawn*), 2265
`start()` (*in module salt.modules.openbsdrcctl_service*), 2274
`start()` (*in module salt.modules.openbsdservice*), 2276
`start()` (*in module salt.modules.parallels*), 2324
`start()` (*in module salt.modules.rest_service*), 2440
`start()` (*in module salt.modules.rh_service*), 2445
`start()` (*in module salt.modules.riak*), 2446
`start()` (*in module salt.modules.runit*), 2455
`start()` (*in module salt.modules.s6*), 2461
`start()` (*in module salt.modules.smartos_virt*), 2504
`start()` (*in module salt.modules.smartos_vmadm*), 2507
`start()` (*in module salt.modules.smf_service*), 2512
`start()` (*in module salt.modules.ssh_service*), 2543
`start()` (*in module salt.modules.supervisord*), 2567
`start()` (*in module salt.modules.syslog_ng*), 2581
`start()` (*in module salt.modules.systemd_service*), 2596
`start()` (*in module salt.modules.tomcat*), 2621
`start()` (*in module salt.modules.upstart_service*), 2629
`start()` (*in module salt.modules.vagrant*), 2637

- start() (*in module salt.modules.vboxmanage*), 2644
- start() (*in module salt.modules.virt*), 2671
- start() (*in module salt.modules.vmctl*), 2679
- start() (*in module salt.modules.win_service*), 2800
- start() (*in module salt.modules.xapi_virt*), 2855
- start() (*in module salt.runners.lxc*), 3120
- start() (*in module salt.runners.smartos_vmadm*), 3150
- start() (*in module salt.runners.virt*), 3159
- start() (*salt.modules.xmpp.SendMsgBot method*), 2863
- start() (*salt.returners.xmpp_return.SendMsgBot method*), 357
- start_() (*in module salt.modules.dockermod*), 1658
- start_app() (*in module salt.modules.rabbitmq*), 2417
- start_apppool() (*in module salt.modules.win_iis*), 2763
- start_elasticsearch_service_software_update() (*in module salt.modules.boto3_elasticsearch*), 1319
- start_event_grains
  - conf/minion, 149
- start_logging() (*in module salt.modules.boto_cloudtrail*), 1350
- start_service() (*in module salt.modules.openstack_mng*), 2277
- start_site() (*in module salt.modules.win_iis*), 2763
- start_time_service() (*in module salt.modules.win_system*), 2819
- start_transaction() (*in module salt.modules.bigip*), 1299
- start_volume() (*in module salt.modules.glusterfs*), 1814
- started() (*in module salt.states.glusterfs*), 3467
- started() (*in module salt.states.syslog_ng*), 3732
- startup() (*in module salt.modules.trafficserver*), 2624
- startup() (*in module salt.states.trafficserver*), 3740
- startup_items() (*in module salt.modules.osquery*), 2295
- startup_states
  - conf/minion, 149
- stash() (*in module salt.modules.git*), 1791
- stat_file() (*in module salt.modules.cp*), 1578
- State Compiler, **5403**
- State Declaration, **5403**
- State Function, **5403**
- State Module, **5403**
- State Run, **5403**
- state() (*in module salt.modules.dockermod*), 1658
- state() (*in module salt.modules.lxc*), 2001
- state() (*in module salt.modules.nspawn*), 2265
- state() (*in module salt.states.saltmod*), 3699
- state_aggregate
  - conf/master, 80
- state_apply() (*in module salt.modules.saltcheck*), 2469
- state_argspec() (*in module salt.modules.sysmod*), 2587
- state_doc() (*in module salt.modules.sysmod*), 2587
- state_events
  - conf/master, 80
- state_output
  - conf/master, 80
  - conf/minion, 150
- state_output_diff
  - conf/master, 80
  - conf/minion, 150
- state_result() (*in module salt.states.libcloud_dns*), 3535
- state_result() (*in module salt.states.libcloud_loadbalancer*), 3536
- state_result() (*in module salt.states.libcloud_storage*), 3538
- state_schema() (*in module salt.modules.sysmod*), 2587
- state_top
  - conf/master, 75
  - conf/minion, 147
- state_top_saltenv
  - conf/master, 76
  - conf/minion, 147
- state_verbose
  - conf/master, 79
  - conf/minion, 150
- Statement (*class in salt.modules.syslog_ng*), 2579
- states() (*in module salt.loader*), 3845
- states_dirs
  - conf/minion, 145
- Stats (*class in salt.netapi.rest_cherrypy.app*), 2959
- stats() (*in module salt.modules.file*), 1729
- stats() (*in module salt.modules.locate*), 1981
- stats() (*in module salt.modules.napalm_ntp*), 2177
- stats() (*in module salt.modules.pkgng*), 2364
- stats() (*in module salt.modules.quota*), 2412
- stats() (*in module salt.modules.syslog_ng*), 2581
- stats() (*in module salt.modules.uwsgi*), 2634
- stats() (*in module salt.modules.win_file*), 2744
- status() (*in module salt.modules.apcups*), 1211
- status() (*in module salt.modules.bcache*), 1282
- status() (*in module salt.modules.boto_cloudtrail*), 1350
- status() (*in module salt.modules.boto_elasticsearch_domain*), 1385
- status() (*in module salt.modules.daemontools*), 1586
- status() (*in module salt.modules.debian_service*), 1599
- status() (*in module salt.modules.dummyproxy_service*), 1674
- status() (*in module salt.modules.freebsdjail*), 1740
- status() (*in module salt.modules.freebsdservice*), 1749

status() (*in module salt.modules.gentoo_service*), 1755
status() (*in module salt.modules.git*), 1792
status() (*in module salt.modules.glusterfs*), 1815
status() (*in module salt.modules.hg*), 1849
status() (*in module salt.modules.jboss7*), 1905
status() (*in module salt.modules.launchctl_service*), 1952
status() (*in module salt.modules.mac_service*), 2041
status() (*in module salt.modules.memcached*), 2078
status() (*in module salt.modules.monit*), 2090
status() (*in module salt.modules.mysql*), 2105
status() (*in module salt.modules.netbsdservice*), 2204
status() (*in module salt.modules.nginx*), 2248
status() (*in module salt.modules.openbsdrcctl_service*), 2274
status() (*in module salt.modules.openbsdservice*), 2276
status() (*in module salt.modules.parallels*), 2324
status() (*in module salt.modules.puppet*), 2396
status() (*in module salt.modules.rabbitmq*), 2417
status() (*in module salt.modules.rdp*), 2427
status() (*in module salt.modules.rest_service*), 2440
status() (*in module salt.modules.rh_service*), 2445
status() (*in module salt.modules.riak*), 2446
status() (*in module salt.modules.runit*), 2455
status() (*in module salt.modules.s6*), 2462
status() (*in module salt.modules.smf_service*), 2512
status() (*in module salt.modules.snapper*), 2546
status() (*in module salt.modules.ssh_service*), 2543
status() (*in module salt.modules.supervisord*), 2567
status() (*in module salt.modules.svn*), 2570
status() (*in module salt.modules.systemd_service*), 2597
status() (*in module salt.modules.tomcat*), 2621
status() (*in module salt.modules.trafficserver*), 2624
status() (*in module salt.modules.upstart_service*), 2629
status() (*in module salt.modules.vmctl*), 2679
status() (*in module salt.modules.win_service*), 2800
status() (*in module salt.modules.win_task*), 2830
status() (*in module salt.modules.zpool*), 2923
status() (*in module salt.runners.manage*), 3125
status() (*in module salt.states.disk*), 3369
status_autostart() (*in module salt.modules.runit*), 2455
status_battery() (*in module salt.modules.apcups*), 1212
status_charge() (*in module salt.modules.apcups*), 1212
status_leader() (*in module salt.modules.consul*), 1572
status_load() (*in module salt.modules.apcups*), 1212
status_peers() (*in module salt.modules.consul*), 1573
status_raw() (*in module salt.modules.supervisord*), 2567
status_to_string() (*in module salt.modules.snapper*), 2546
status_webapp() (*in module salt.modules.tomcat*), 2621
statvfs() (*in module salt.modules.file*), 1729
stdout_print() (*in module salt.runners.test*), 3156
stonith_create() (*in module salt.modules.pcs*), 2331
stonith_present() (*in module salt.states.pcs*), 3636
stonith_show() (*in module salt.modules.pcs*), 2331
stop() (*in module salt.cloud.clouds.aliyun*), 1034
stop() (*in module salt.cloud.clouds.azurearm*), 1038
stop() (*in module salt.cloud.clouds.digitalocean*), 1042
stop() (*in module salt.cloud.clouds.dimensiondata*), 1044
stop() (*in module salt.cloud.clouds.ec2*), 1052
stop() (*in module salt.cloud.clouds.gce*), 1059
stop() (*in module salt.cloud.clouds.gogrid*), 1061
stop() (*in module salt.cloud.clouds.joyent*), 1064
stop() (*in module salt.cloud.clouds.linode*), 1070
stop() (*in module salt.cloud.clouds.oneandone*), 1090
stop() (*in module salt.cloud.clouds.opennebula*), 1099
stop() (*in module salt.cloud.clouds.parallels*), 1115
stop() (*in module salt.cloud.clouds.profitbricks*), 1119
stop() (*in module salt.cloud.clouds.proxmox*), 1121
stop() (*in module salt.cloud.clouds.qingcloud*), 1124
stop() (*in module salt.cloud.clouds.tencentcloud*), 1133
stop() (*in module salt.cloud.clouds.virtualbox*), 1136
stop() (*in module salt.cloud.clouds.vmware*), 1146
stop() (*in module salt.cloud.clouds.vultrpy*), 1148
stop() (*in module salt.cloud.clouds.xen*), 1152
stop() (*in module salt.modules.bcache*), 1282
stop() (*in module salt.modules.bluez_bluetooth*), 1301
stop() (*in module salt.modules.daemontools*), 1586
stop() (*in module salt.modules.debian_service*), 1599
stop() (*in module salt.modules.dockercompose*), 1614
stop() (*in module salt.modules.dockermod*), 1658
stop() (*in module salt.modules.dummyproxy_service*), 1675
stop() (*in module salt.modules.freebsdjail*), 1740
stop() (*in module salt.modules.freebsdservice*), 1749
stop() (*in module salt.modules.gentoo_service*), 1756
stop() (*in module salt.modules.launchctl_service*), 1952
stop() (*in module salt.modules.lxc*), 2001
stop() (*in module salt.modules.mac_service*), 2041
stop() (*in module salt.modules.mdadm_raid*), 2076
stop() (*in module salt.modules.monit*), 2090
stop() (*in module salt.modules.netbsdservice*), 2204
stop() (*in module salt.modules.openbsdrcctl_service*), 2274
stop() (*in module salt.modules.openbsdservice*), 2276
stop() (*in module salt.modules.parallels*), 2324
stop() (*in module salt.modules.rest_service*), 2440
stop() (*in module salt.modules.rh_service*), 2445

stop() (in module salt.modules.riak), 2446
stop() (in module salt.modules.runit), 2455
stop() (in module salt.modules.s6), 2462
stop() (in module salt.modules.smartos_virt), 2504
stop() (in module salt.modules.smartos_vmadm), 2507
stop() (in module salt.modules.smf_service), 2512
stop() (in module salt.modules.ssh_service), 2543
stop() (in module salt.modules.supervisord), 2567
stop() (in module salt.modules.syslog_ng), 2581
stop() (in module salt.modules.systemd_service), 2597
stop() (in module salt.modules.tomcat), 2621
stop() (in module salt.modules.upstart_service), 2629
stop() (in module salt.modules.vagrant), 2637
stop() (in module salt.modules.vboxmanage), 2644
stop() (in module salt.modules.virt), 2671
stop() (in module salt.modules.vmctl), 2679
stop() (in module salt.modules.win_service), 2800
stop() (in module salt.modules.win_task), 2831
stop() (in module salt.modules.xapi_virt), 2855
stop() (in module salt.runners.lxc), 3120
stop() (in module salt.runners.smartos_vmadm), 3150
stop() (in module salt.states.modjk_worker), 3555
stop_app() (in module salt.modules.rabbitmq), 2418
stop_apppool() (in module salt.modules.win_iis), 2763
stop_logging() (in module salt.modules.boto_cloudtrail), 1350
stop_server() (in module salt.modules.jboss7), 1905
stop_service() (in module salt.modules.openstack_mng), 2277
stop_site() (in module salt.modules.win_iis), 2764
stop_time_service() (in module salt.modules.win_system), 2819
stop_volume() (in module salt.modules.glusterfs), 1815
stopped() (in module salt.states.docker_container), 3390
stopped() (in module salt.states.lxc), 3548
stopped() (in module salt.states.syslog_ng), 3732
stopped() (in module salt.states.vagrant), 3746
stopped() (in module salt.states.virt), 3754
store() (in module salt.cache.consul), 1026
store() (in module salt.cache.etcd_cache), 1027
store() (in module salt.cache.localfs), 1028
store() (in module salt.cache.mysql_cache), 1029
store() (in module salt.cache.redis_cache), 1031
store() (in module salt.runners.cache), 3102
store() (salt.modules.inspectlib.fsdb.CsvDB method), 1878
stp() (in module salt.modules.bridge), 1465
str_encode() (in module salt.modules.mod_random), 2083
stream() (in module salt.runners.test), 3156
strftime
    jinja filters, 428
string() (in module salt.states.redismod), 3688
string_format() (in module salt.modules.out), 2297
stringify() (in module salt.modules.pkg_resource), 2350
submodule() (in module salt.modules.git), 1792
subnet_absent() (in module salt.states.azurearm_network), 3214
subnet_absent() (in module salt.states.boto_vpc), 3333
subnet_create() (in module salt.modules.neutronng), 2241
subnet_create_or_update() (in module salt.modules.azurearm_network), 1268
subnet_delete() (in module salt.modules.azurearm_network), 1268
subnet_delete() (in module salt.modules.neutronng), 2242
subnet_exists() (in module salt.modules.boto_vpc), 1462
subnet_get() (in module salt.modules.azurearm_network), 1269
subnet_get() (in module salt.modules.neutronng), 2242
subnet_group_absent() (in module salt.states.boto_elasticache), 3282
subnet_group_absent() (in module salt.states.boto_rds), 3315
subnet_group_exists() (in module salt.modules.boto_elasticache), 1382
subnet_group_exists() (in module salt.modules.boto_rds), 1428
subnet_group_present() (in module salt.states.boto_elasticache), 3282
subnet_group_present() (in module salt.states.boto_rds), 3315
subnet_present() (in module salt.states.azurearm_network), 3214
subnet_present() (in module salt.states.boto_vpc), 3333
subnet_update() (in module salt.modules.neutronng), 2242
subnets() (in module salt.modules.network), 2220
subnets() (in module salt.modules.win_network), 2775
subnets6() (in module salt.modules.network), 2220
subnets_list() (in module salt.modules.azurearm_network), 1269
subscribe() (in module salt.modules.boto_sns), 1445
subscription_get() (in module salt.modules.azurearm_resource), 1278
subscriptions_list() (in module salt.modules.azurearm_resource), 1278
subscriptions_list_locations() (in module salt.modules.azurearm_resource), 1278

substitute_params() (*in module salt.modules.zabbix*), 2888
substring_in_list
  jinja filters, 435
subvolume_create() (*in module salt.modules.btrfs*), 1468
subvolume_delete() (*in module salt.modules.btrfs*), 1469
subvolume_exists() (*in module salt.modules.btrfs*), 1469
subvolume_find_new() (*in module salt.modules.btrfs*), 1469
subvolume_get_default() (*in module salt.modules.btrfs*), 1469
subvolume_list() (*in module salt.modules.btrfs*), 1470
subvolume_set_default() (*in module salt.modules.btrfs*), 1470
subvolume_show() (*in module salt.modules.btrfs*), 1470
subvolume_snapshot() (*in module salt.modules.btrfs*), 1471
subvolume_sync() (*in module salt.modules.btrfs*), 1471
succeed_with_changes() (*in module salt.states.test*), 3735
succeed_without_changes() (*in module salt.states.test*), 3735
sudo_acl
  conf/master, 70
sudo_user
  conf/minion, 129
suid_bin() (*in module salt.modules.osquery*), 2295
summary() (*in module salt.modules.monit*), 2090
summary() (*in module salt.modules.puppet*), 2396
super_() (*in module salt.modules.bcache*), 1283
supports_proxies() (*in module salt.modules.vsphere*), 2713
suspend() (*in module salt.cloud.clouds.vmware*), 1146
suspend() (*in module salt.cloud.clouds.xen*), 1152
suspend() (*in module salt.modules.nova*), 2256
svnfs_branches
  conf/master, 95
svnfs_mountpoint
  conf/master, 94
svnfs_remotes
  conf/master, 94
svnfs_root
  conf/master, 94
svnfs_saltenv_blacklist
  conf/master, 96
svnfs_saltenv_whitelist
  conf/master, 95
svnfs_tags
  conf/master, 95
svnfs_trunk
  conf/master, 95
svnfs_update_interval
  conf/master, 96
SvnPillar (*class in salt.pillar.svn_pillar*), 3044
swap() (*in module salt.states.mount*), 3565
swap_memory() (*in module salt.modules.ps*), 2392
swapoff() (*in module salt.modules.mount*), 2094
swapon() (*in module salt.modules.mount*), 2094
swaps() (*in module salt.modules.mount*), 2094
swarm_init() (*in module salt.modules.swarm*), 2572
swarm_service_info() (*in module salt.modules.swarm*), 2572
swarm_tokens() (*in module salt.modules.swarm*), 2573
switch() (*in module salt.modules.svn*), 2570
switch() (*in module salt.states.dellchassis*), 3368
switch_org() (*in module salt.modules.grafana4*), 1828
symbolic_ref() (*in module salt.modules.git*), 1794
symlink() (*in module salt.modules.file*), 1729
symlink() (*in module salt.modules.win_file*), 2744
symlink() (*in module salt.states.file*), 3449
symlink_list() (*in module salt.runners.fileserver*), 3114
symmetric_difference
  jinja filters, 433
sync() (*in module salt.modules.eix*), 1680
sync() (*in module salt.modules.layman*), 1953
sync_all() (*in module salt.modules.saltutil*), 2473
sync_all() (*in module salt.runners.saltutil*), 3143
sync_all() (*in module salt.states.saltutil*), 3701
sync_beacons() (*in module salt.modules.saltutil*), 2473
sync_beacons() (*in module salt.states.saltutil*), 3701
sync_cache() (*in module salt.runners.saltutil*), 3143
sync_clouds() (*in module salt.modules.saltutil*), 2474
sync_clouds() (*in module salt.runners.saltutil*), 3144
sync_clouds() (*in module salt.states.saltutil*), 3701
sync_config_devices() (*in module salt.modules.lxd*), 2021
sync_contains() (*in module salt.modules.makeconf*), 2066
sync_eauth_tokens() (*in module salt.runners.saltutil*), 3144
sync_engines() (*in module salt.modules.saltutil*), 2474
sync_engines() (*in module salt.runners.saltutil*), 3144
sync_engines() (*in module salt.states.saltutil*), 3701
sync_executors() (*in module salt.modules.saltutil*), 2474
sync_executors() (*in module salt.runners.saltutil*), 3144

sync_executors() (*in module salt.states.saltutil*), 3702
sync_fileserver() (*in module salt.runners.saltutil*), 3144
sync_grains() (*in module salt.modules.saltutil*), 2475
sync_grains() (*in module salt.runners.saltutil*), 3145
sync_grains() (*in module salt.states.saltutil*), 3702
sync_log_handlers() (*in module salt.modules.saltutil*), 2475
sync_log_handlers() (*in module salt.states.saltutil*), 3702
sync_matchers() (*in module salt.modules.saltutil*), 2475
sync_matchers() (*in module salt.states.saltutil*), 3702
sync_modules() (*in module salt.modules.saltutil*), 2476
sync_modules() (*in module salt.runners.saltutil*), 3145
sync_modules() (*in module salt.states.saltutil*), 3702
sync_output() (*in module salt.modules.saltutil*), 2476
sync_output() (*in module salt.runners.saltutil*), 3145
sync_output() (*in module salt.states.saltutil*), 3702
sync_outputters() (*in module salt.modules.saltutil*), 2477
sync_outputters() (*in module salt.states.saltutil*), 3702
sync_pillar() (*in module salt.modules.saltutil*), 2477
sync_pillar() (*in module salt.runners.saltutil*), 3145
sync_pillar() (*in module salt.states.saltutil*), 3703
sync_proxymodules() (*in module salt.modules.saltutil*), 2477
sync_proxymodules() (*in module salt.runners.saltutil*), 3145
sync_proxymodules() (*in module salt.states.saltutil*), 3703
sync_queues() (*in module salt.runners.saltutil*), 3146
sync_renderers() (*in module salt.modules.saltutil*), 2478
sync_renderers() (*in module salt.runners.saltutil*), 3146
sync_renderers() (*in module salt.states.saltutil*), 3703
sync_returners() (*in module salt.modules.saltutil*), 2478
sync_returners() (*in module salt.runners.saltutil*), 3146
sync_returners() (*in module salt.states.saltutil*), 3703
sync_roster() (*in module salt.runners.saltutil*), 3146
sync_runners() (*in module salt.runners.saltutil*), 3146
sync_sdb() (*in module salt.modules.saltutil*), 2478
sync_sdb() (*in module salt.runners.saltutil*), 3147
sync_sdb() (*in module salt.states.saltutil*), 3703
sync_serializers() (*in module salt.modules.saltutil*), 2479
sync_serializers() (*in module salt.runners.saltutil*), 3147
sync_serializers() (*in module salt.states.saltutil*), 3703
sync_states() (*in module salt.modules.saltutil*), 2479
sync_states() (*in module salt.runners.saltutil*), 3147
sync_states() (*in module salt.states.saltutil*), 3703
sync_thorium() (*in module salt.modules.saltutil*), 2479
sync_thorium() (*in module salt.runners.saltutil*), 3147
sync_thorium() (*in module salt.states.saltutil*), 3704
sync_tops() (*in module salt.runners.saltutil*), 3148
sync_utils() (*in module salt.modules.saltutil*), 2480
sync_utils() (*in module salt.runners.saltutil*), 3148
sync_utils() (*in module salt.states.saltutil*), 3704
sync_wheel() (*in module salt.runners.saltutil*), 3148
syncdb() (*in module salt.modules.djangomod*), 1606
synchronized() (*in module salt.states.rsync*), 3694
Syndic, **5403**
syndic_failover
  conf/master, 113
syndic_finger
  conf/minion, 141
syndic_forward_all_events
  conf/master, 113
syndic_log_file
  conf/master, 112
syndic_master
  conf/master, 112
syndic_master_port
  conf/master, 112
syndic_pidfile
  conf/master, 112
syndic_wait
  conf/master, 113
sysctl() (*in module salt.modules.freebsdjail*), 1740
SysInfo (*class in salt.modules.inspectlib.query*), 1876
syslog() (*in module salt.modules.drac*), 1664
syslog() (*in module salt.modules.dracr*), 1672
syslog() (*in module salt.states.cimc*), 3342
syslog_configured() (*in module salt.states.esxi*), 3413
syslog_service_reload() (*in module salt.modules.vsphere*), 2713
SyslogNgError, 2579
sysrq() (*in module salt.modules.smartos_vmadm*), 2507
system() (*in module salt.states.keyboard*), 3517
system() (*in module salt.states.locale*), 3540
system() (*in module salt.states.network*), 3601
system() (*in module salt.states.timezone*), 3736

`system_controls()` (*in module salt.modules.osquery*), 2295
`system_info()` (*in module salt.modules.drac*), 1665
`system_info()` (*in module salt.modules.dracr*), 1672
`system_info()` (*in module salt.modules.nxos*), 2266
`system_info()` (*in module salt.modules.vsphere*), 2714
`system_info()` (*in module salt.proxy.nxos*), 3076
`SYSTEM_PERFORMANCE_INFORMATION` (*class in salt.modules.win_status*), 2806
`system_types()` (*in module salt.modules.parted_partition*), 2328
`SystemCalls` (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATION attribute*), 2810
`systemctl_reload()` (*in module salt.modules.systemd_service*), 2597
`systemd_running_state()` (*in module salt.modules.lxc*), 2002

T

`table()` (*in module salt.modules.pf*), 2337
`table_absent()` (*in module salt.states.sqlite3*), 3723
`table_present()` (*in module salt.states.sqlite3*), 3723
`TableDisplay` (*class in salt.output.table_out*), 2988
`tables()` (*in module salt.modules.sqlite3*), 2538
`tablespace_alter()` (*in module salt.modules.postgres*), 2383
`tablespace_create()` (*in module salt.modules.postgres*), 2384
`tablespace_exists()` (*in module salt.modules.postgres*), 2384
`tablespace_list()` (*in module salt.modules.postgres*), 2384
`tablespace_remove()` (*in module salt.modules.postgres*), 2384
`tag()` (*in module salt.modules.git*), 1795
`tag_()` (*in module salt.modules.dockermod*), 1659
`tagged()` (*in module salt.states.boto3_elasticsearch*), 3245
`take_action()` (*in module salt.cloud.clouds.joyent*), 1065
`tar()` (*in module salt.modules.archive*), 1231
`Target`, **5403**
`target()` (*in module salt.modules.sysfs*), 2577
`target_glob()` (*in module salt.roster.range*), 3092
`target_group_exists()` (*in module salt.modules.boto_elbv2*), 1393
`target_range()` (*in module salt.roster.range*), 3092
`targets()` (*in module salt.roster.ansible*), 3090
`targets()` (*in module salt.roster.cache*), 3091
`targets()` (*in module salt.roster.cloud*), 3092
`targets()` (*in module salt.roster.clustershell*), 3092
`targets()` (*in module salt.roster.flat*), 3092
`targets()` (*in module salt.roster.range*), 3092
`targets()` (*in module salt.roster.scan*), 3093
`targets()` (*in module salt.roster.sshconfig*), 3093
`targets()` (*in module salt.roster.terraform*), 3094
`targets()` (*salt.roster.scan.RosterMatcher method*), 3093
`targets()` (*salt.roster.sshconfig.RosterMatcher method*), 3093
`targets_deregistered()` (*in module salt.states.boto_elbv2*), 3292
`targets_registered()` (*in module salt.states.boto_elbv2*), 3292
`task_absent()` (*in module salt.states.kapacitor*), 3515
`task_present()` (*in module salt.states.kapacitor*), 3515
`tcp_keepalive`
  `conf/master`, 117
  `conf/minion`, 164
`tcp_keepalive_cnt`
  `conf/master`, 118
  `conf/minion`, 164
`tcp_keepalive_idle`
  `conf/master`, 118
  `conf/minion`, 164
`tcp_keepalive_intvl`
  `conf/master`, 118
  `conf/minion`, 165
`tcp_master_pub_port`
  `conf/master`, 63
`tcp_master_publish_pull`
  `conf/master`, 63
`tcp_master_pull_port`
  `conf/master`, 63
`tcp_master_workers`
  `conf/master`, 63
`tcp_pub_port`
  `conf/minion`, 140
`tcp_pull_port`
  `conf/minion`, 141
`team_absent()` (*in module salt.states.github*), 3464
`team_present()` (*in module salt.states.github*), 3464
`tempallow()` (*in module salt.modules.csf*), 1583
`tempdeny()` (*in module salt.modules.csf*), 1583
`template()` (*in module salt.modules.state*), 2558
`template_allocate()` (*in module salt.cloud.clouds.opennebula*), 1099
`template_clone()` (*in module salt.cloud.clouds.opennebula*), 1099
`template_delete()` (*in module salt.cloud.clouds.opennebula*), 1100
`template_get()` (*in module salt.modules.zabbix*), 2888
`template_instantiate()` (*in module salt.cloud.clouds.opennebula*), 1100

`template_list()` (*in module salt.cloud.clouds.xen*), 1152
`template_stack()` (*in module salt.modules.heat*), 1847
`template_str()` (*in module salt.modules.state*), 2558
`template_update()` (*in module salt.cloud.clouds.opennebula*), 1100
`TemplateError`, 4000
`templates()` (*in module salt.modules.lxc*), 2002
`tenant_absent()` (*in module salt.states.keystone*), 3519
`tenant_create()` (*in module salt.modules.keystone*), 1934
`tenant_delete()` (*in module salt.modules.keystone*), 1934
`tenant_get()` (*in module salt.modules.keystone*), 1934
`tenant_list()` (*in module salt.modules.keystone*), 1934
`tenant_present()` (*in module salt.states.keystone*), 3519
`tenant_update()` (*in module salt.modules.keystone*), 1934
`tenants_list()` (*in module salt.modules.azurearm_resource*), 1278
`term()` (*in module salt.modules.daemontools*), 1586
`term()` (*in module salt.modules.s6*), 2462
`term()` (*in module salt.states.netacl*), 3577
`term_all_jobs()` (*in module salt.modules.saltutil*), 2480
`term_job()` (*in module salt.modules.saltutil*), 2480
`terminate()` (*in module salt.cloud.clouds.vmware*), 1146
`terminate()` (*in module salt.modules.boto_ec2*), 1374
`terminate()` (*in module salt.modules.nspawn*), 2265
`test`
  `conf/minion`, 150
`test()` (*in module salt.modules.ipset*), 1897
`test()` (*in module salt.modules.riak*), 2446
`test_bare_started_state()` (*in module salt.modules.lxc*), 2002
`test_cert()` (*in module salt.modules.win_pki*), 2786
`test_config()` (*in module salt.modules.win_dsc*), 2733
`test_fib_route()` (*in module salt.modules.panos*), 2319
`test_from_state()` (*in module salt.proxy.dummy*), 3057
`test_from_state()` (*in module salt.proxy.rest_sample*), 3086
`test_sd_started_state()` (*in module salt.modules.lxc*), 2002
`test_security_policy()` (*in module salt.modules.panos*), 2319
`test_vcenter_connection()` (*in module salt.cloud.clouds.vmware*), 1147
`test_vcenter_connection()` (*in module salt.modules.vsphere*), 2714
`testing_off()` (*in module salt.states.csf*), 3360
`testing_on()` (*in module salt.states.csf*), 3360
`tftp_update_bios()` (*in module salt.modules.cimc*), 1513
`tftp_update_cimc()` (*in module salt.modules.cimc*), 1513
`thin_extra_mods`
  `conf/master`, 68
`thing_type_absent()` (*in module salt.states.boto_iot*), 3302
`thing_type_exists()` (*in module salt.modules.boto_iot*), 1412
`thing_type_present()` (*in module salt.states.boto_iot*), 3302
`threads()` (*in module salt.modules.sysbench*), 2575
`tidied()` (*in module salt.states.file*), 3450
`time()` (*in module salt.modules.redismod*), 2432
`time_()` (*in module salt.modules.osquery*), 2295
`time_()` (*in module salt.modules.status*), 2561
`TimedProcTimeoutError`, 4000
`timeout`
  `conf/master`, 55
`timeout()` (*in module salt.thorium.key*), 3825
`timeoutDecorator()` (*in module salt.modules.junos*), 1919
`TimeoutError`, 4000
`TLSSocketAppender` (*class in salt.engines.logentries*), 1159
`to_bool`
  `jinja filters`, 429
`to_bytes`
  `jinja filters`, 436
`to_camelcase`
  `jinja filters`, 443
`to_num`
  `jinja filters`, 436
`to_snake_case`
  `jinja filters`, 443
`todict()` (*salt.states.firewalld.ForwardingMapping method*), 3453
`toggle()` (*in module salt.modules.parted_partition*), 2328
`tojson`
  `jinja filters`, 438
`token_expire`
  `conf/master`, 71
`token_expire_user_override`
  `conf/master`, 71
`token_get()` (*in module salt.modules.keystone*), 1934
`TokenAuthenticationError`, 4000

`tokenize_grant()` (*in module salt.modules.mysql*), 2105
`Top File`, **5403**
`top()` (*in module salt.modules.dockermod*), 1659
`top()` (*in module salt.modules.ps*), 2393
`top()` (*in module salt.modules.state*), 2558
`top()` (*in module salt.tops.cobbler*), 3830
`top()` (*in module salt.tops.ext_nodes*), 3831
`top()` (*in module salt.tops.mongo*), 3831
`top()` (*in module salt.tops.reclass_adapter*), 3832
`top()` (*in module salt.tops.saltclass*), 3836
`top()` (*in module salt.tops.varstack_top*), 3837
`top_file`
    `conf/minion`, 149
`top_file_merging_strategy`
    `conf/master`, 76
    `conf/minion`, 148
`topic_rule_absent()` (*in module salt.states.boto_iot*), 3302
`topic_rule_exists()` (*in module salt.modules.boto_iot*), 1412
`topic_rule_present()` (*in module salt.states.boto_iot*), 3303
`total_physical_memory()` (*in module salt.modules.ps*), 2393
`TotalSystemCodePages` (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATION attribute*), 2810
`TotalSystemDriverPages` (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATION attribute*), 2810
`touch()` (*in module salt.modules.file*), 1729
`touch()` (*in module salt.states.file*), 3450
`tpstats()` (*in module salt.modules.cassandra*), 1482
`traceroute()` (*in module salt.modules.napalm_network*), 2175
`traceroute()` (*in module salt.modules.network*), 2221
`traceroute()` (*in module salt.modules.win_network*), 2775
`TransitionCount` (*salt.modules.win_status.SYSTEM_PERFORMANCE_INFORMATION attribute*), 2810
`transport`
    `conf/master`, 62
    `conf/minion`, 141
`transport_opts`
    `conf/master`, 62
`traverse`
    `jinja filters`, 442
`traverse()` (*in module salt.modules.napalm_formula*), 2155
`tree()` (*in module salt.modules.augeas_cfg*), 1246
`tree()` (*in module salt.modules.etcd_mod*), 1697
`tree()` (*in module salt.modules.iosconfig*), 1883
`trigger_event()` (*in module salt.modules.ifttt*), 1852
`trigger_event()` (*in module salt.states.ifttt*), 3489
`trim_cflags()` (*in module salt.modules.makeconf*), 2066
`trim_cxxflags()` (*in module salt.modules.makeconf*), 2066
`trim_emerge_default_opts()` (*in module salt.modules.makeconf*), 2066
`trim_features()` (*in module salt.modules.makeconf*), 2067
`trim_gentoo_mirrors()` (*in module salt.modules.makeconf*), 2067
`trim_makeopts()` (*in module salt.modules.makeconf*), 2067
`trim_var()` (*in module salt.modules.makeconf*), 2067
`true_()` (*in module salt.modules.test*), 2603
`truncate()` (*in module salt.modules.file*), 1729
`trust_key()` (*in module salt.modules.gpg*), 1822
`try_()` (*in module salt.modules.test*), 2604
`tsql_query()` (*in module salt.modules.mssql*), 2097
`tty()` (*in module salt.modules.cmdmod*), 1554
`tty()` (*in module salt.modules.test*), 2604
`tune()` (*in module salt.modules.disk*), 1604
`tune()` (*in module salt.modules.extfs*), 1702
`tuned()` (*in module salt.states.blockdev*), 3233
`TXT()` (*in module salt.modules.dig*), 1602
`TypedParameter` (*class in salt.modules.syslog_ng*), 2579
`TypedParameterValue` (*class in salt.modules.syslog_ng*), 2579
`TzMapper` (*class in salt.modules.win_timezone*), 2831

U

`uid_to_user()` (*in module salt.modules.file*), 1730
`uid_to_user()` (*in module salt.modules.win_file*), 2744
`umount()` (*in module salt.modules.mount*), 2095
`unallow()` (*in module salt.modules.csf*), 1583
`unassign_floating_ip()` (*in module salt.cloud.clouds.digitalocean*), 1042
`unassign_private_ip_addresses()` (*in module salt.modules.boto_ec2*), 1374
`unblock()` (*in module salt.modules.bluez_bluetooth*), 1301
`unchanged()` (*salt.states.cyg.DictDiffer method*), 3361
`uncomment()` (*in module salt.modules.file*), 1730
`uncomment()` (*in module salt.states.file*), 3451
`undefine()` (*in module salt.modules.virt*), 2671
`undeny()` (*in module salt.modules.csf*), 1583
`undeploy()` (*in module salt.modules.jboss7*), 1906
`undeploy()` (*in module salt.modules.tomcat*), 2621
`undeployed()` (*in module salt.states.tomcat*), 3737
`undo()` (*in module salt.modules.snapper*), 2547
`undo_jid()` (*in module salt.modules.snapper*), 2547
`unfreeze()` (*in module salt.modules.lxc*), 2002

unfreeze() (in module salt.runners.lxc), 3121
unhold() (in module salt.modules.aptpkg), 1225
unhold() (in module salt.modules.opkg), 2285
unhold() (in module salt.modules.pkgng), 2365
unhold() (in module salt.modules.postfix), 2373
unhold() (in module salt.modules.yumpkg), 2875
uninstall() (in module salt.modules.bower), 1463
uninstall() (in module salt.modules.cabal), 1472
uninstall() (in module salt.modules.chocolatey), 1504
uninstall() (in module salt.modules.cyg), 1584
uninstall() (in module salt.modules.gem), 1751
uninstall() (in module salt.modules.mac_keychain), 2029
uninstall() (in module salt.modules.nix), 2252
uninstall() (in module salt.modules.npm), 2259
uninstall() (in module salt.modules.pecl), 2333
uninstall() (in module salt.modules.pip), 2348
uninstall() (in module salt.modules.win_license), 2772
uninstall() (in module salt.modules.win_wua), 2845
uninstall() (in module salt.modules.win_wusa), 2846
uninstall() (in module salt.modules.zoneadm), 2909
uninstall_app() (in module salt.modules.macpackage), 2030
uninstall_python() (in module salt.modules.pyenv), 2410
uninstall_ruby() (in module salt.modules.rbenv), 2425
uninstalled() (in module salt.states.chocolatey), 3338
uninstalled() (in module salt.states.mac_keychain), 3549
uninstalled() (in module salt.states.win_wusa), 3787
uninstalled() (in module salt.states.zone), 3813
union
    jinja filters, 433
unique
    jinja filters, 450
unjoin_domain() (in module salt.modules.win_system), 2819
unload() (in module salt.modules.solaris_fmadm), 2515
unlock() (in module salt.modules.junos), 1919
unlock() (in module salt.modules.pkgng), 2365
unlock() (in module salt.modules.zk_concurrency), 2906
unlock() (in module salt.states.junos), 3514
unlock() (in module salt.states.zk_concurrency), 3806
unlock_account() (in module salt.modules.win_shadow), 2801
unlock_admin() (in module salt.modules.panos), 2320
unlock_keychain() (in module salt.modules.mac_keychain), 2029
unmask_() (in module salt.modules.systemd_service), 2597
unmasked() (in module salt.states.service), 3712
unmonitor() (in module salt.modules.monit), 2090
unmonitor() (in module salt.states.monit), 3562
unmount() (in module salt.modules.macpackage), 2031
unmount() (in module salt.modules.zfs), 2905
unmounted() (in module salt.states.mount), 3565
UnnamedStatement (class in salt.modules.syslog_ng), 2580
unpack() (in module salt.modules.genesis), 1754
unpair() (in module salt.modules.bluez_bluetooth), 1301
unpause() (in module salt.cloud.clouds.xen), 1152
unpause() (in module salt.modules.dockercompose), 1615
unpause() (in module salt.modules.dockermod), 1659
unpowered() (in module salt.states.virt), 3755
unpurge() (in module salt.modules.dpkg_lowpkg), 1662
unrar() (in module salt.modules.archive), 1232
unregister() (in module salt.modules.vboxmanage), 2644
unregister() (in module salt.runners.spacewalk), 3152
unregister_vm() (in module salt.modules.vsphere), 2714
unseal() (in module salt.runners.vault), 3157
unset_quota_volume() (in module salt.modules.glusterfs), 1815
unset_role() (in module salt.proxy.nxos), 3076
unsubscribe() (in module salt.modules.boto_sns), 1445
until() (in module salt.states.loop), 3542
until_no_eval() (in module salt.states.loop), 3542
unzip() (in module salt.modules.archive), 1232
up() (in module salt.modules.debian_ip), 1597
up() (in module salt.modules.dockercompose), 1615
up() (in module salt.modules.linux_ip), 1976
up() (in module salt.modules.nilrt_ip), 2251
up() (in module salt.modules.rh_ip), 2443
up() (in module salt.runners.manage), 3126
update() (in module salt.modules.boto_asg), 1344
update() (in module salt.modules.boto_cloudtrail), 1351
update() (in module salt.modules.boto_dynamodb), 1361
update() (in module salt.modules.boto_elasticsearch_domain), 1385
update() (in module salt.modules.cabal), 1472
update() (in module salt.modules.chocolatey), 1505
update() (in module salt.modules.composer), 1556
update() (in module salt.modules.cyg), 1584
update() (in module salt.modules.data), 1588
update() (in module salt.modules.ddns), 1590
update() (in module salt.modules.defaults), 1600

`update()` (*in module salt.modules.ebuildpkg*), 1679
`update()` (*in module salt.modules.eix*), 1681
`update()` (*in module salt.modules.etcd_mod*), 1697
`update()` (*in module salt.modules.freebsd_update*), 1739
`update()` (*in module salt.modules.freebsdports*), 1746
`update()` (*in module salt.modules.gem*), 1751
`update()` (*in module salt.modules.hg*), 1849
`update()` (*in module salt.modules.mac_softwareupdate*), 2047
`update()` (*in module salt.modules.mine*), 2079
`update()` (*in module salt.modules.namecheap_domains_ns*), 2117
`update()` (*in module salt.modules.pecl*), 2333
`update()` (*in module salt.modules.pyenv*), 2410
`update()` (*in module salt.modules.rbenv*), 2425
`update()` (*in module salt.modules.saltutil*), 2480
`update()` (*in module salt.modules.serverdensity_device*), 2495
`update()` (*in module salt.modules.slsutil*), 2500
`update()` (*in module salt.modules.smartos_nictagadm*), 2503
`update()` (*in module salt.modules.smartos_vmadm*), 2508
`update()` (*in module salt.modules.splunk_search*), 2538
`update()` (*in module salt.modules.statuspage*), 2565
`update()` (*in module salt.modules.supervisord*), 2567
`update()` (*in module salt.modules.svn*), 2571
`update()` (*in module salt.modules.virt*), 2672
`update()` (*in module salt.modules.win_psget*), 2791
`update()` (*in module salt.modules.win_useradd*), 2837
`update()` (*in module salt.modules.yumpkg*), 2875
`update()` (*in module salt.pillar.hg_pillar*), 3009
`update()` (*in module salt.runners.ddns*), 3104
`update()` (*in module salt.runners.fileserver*), 3115
`update()` (*in module salt.runners.git_pillar*), 3115
`update()` (*in module salt.runners.mine*), 3127
`update()` (*in module salt.states.composer*), 3355
`update()` (*in module salt.states.statuspage*), 3728
`update()` (*salt.modules.inspectlib.fsdb.CsvDB method*), 1878
`update()` (*salt.pillar.hg_pillar.Repo method*), 3009
`update()` (*salt.pillar.svn_pillar.SvnPillar method*), 3044
`update_account_password_policy()` (*in module salt.modules.boto_iam*), 1405
`update_affinity_group()` (*in module salt.cloud.clouds.msazure*), 1086
`update_alarm()` (*in module salt.modules.telemetry*), 2599
`update_alias()` (*in module salt.modules.boto_lambda*), 1423
`update_all()` (*in module salt.modules.mac_softwareupdate*), 2047
`update_api_key_description()` (*in module salt.modules.boto_apigateway*), 1340
`update_api_model_schema()` (*in module salt.modules.boto_apigateway*), 1340
`update_app()` (*in module salt.modules.marathon*), 2070
`update_assume_role_policy()` (*in module salt.modules.boto_iam*), 1405
`update_available()` (*in module salt.modules.mac_softwareupdate*), 2048
`update_ca_bundle()` (*in module salt.modules.http*), 1851
`update_ca_bundle()` (*in module salt.runners.http*), 3116
`update_cluster()` (*in module salt.modules.vsphere*), 2714
`update_cname()` (*in module salt.modules.infoblox*), 1872
`update_config()` (*in module salt.modules.napalm_snmp*), 2185
`update_config()` (*in module salt.wheel.config*), 3837
`update_datasource()` (*in module salt.modules.grafana4*), 1828
`update_datasource()` (*in module salt.modules.jboss7*), 1906
`update_device()` (*in module salt.modules.netbox*), 2201
`update_dict_key_value`
  `jinja filters`, 439
`update_disk()` (*in module salt.cloud.clouds.msazure*), 1086
`update_distribution()` (*in module salt.modules.boto_cloudfront*), 1347
`update_dvportgroup()` (*in module salt.modules.vsphere*), 2715
`update_dvs()` (*in module salt.modules.vsphere*), 2715
`update_elasticsearch_domain_config()` (*in module salt.modules.boto3_elasticsearch*), 1319
`update_employee()` (*in module salt.modules.bamboohr*), 1280
`update_endtime()` (*in module salt.returners.local_cache*), 326
`update_event_source_mapping()` (*in module salt.modules.boto_lambda*), 1423
`update_firewall_rule()` (*in module salt.modules.neutron*), 2236
`update_firmware()` (*in module salt.modules.dracr*), 1672
`update_firmware_nfs_or_cifs()` (*in module salt.modules.dracr*), 1673
`update_floatingip()` (*in module salt.modules.neutron*), 2236
`update_function_code()` (*in module salt.modules.boto_lambda*), 1423

update_function_config() (*in module salt.modules.boto_lambda*), 1424
update_git_repos() (*in module salt.modules.winrepo*), 2792
update_git_repos() (*in module salt.runners.winrepo*), 3160
update_global_secondary_index() (*in module salt.modules.boto_dynamodb*), 1361
update_host() (*in module salt.modules.infoblox*), 1872
update_host_datetime() (*in module salt.modules.vsphere*), 2715
update_host_password() (*in module salt.modules.vsphere*), 2715
update_hosted_zone_comment() (*in module salt.modules.boto3_route53*), 1329
update_identity_pool() (*in module salt.modules.boto_cognitoidentity*), 1357
update_image_properties() (*in module salt.modules.glanceng*), 1811
update_input_endpoint() (*in module salt.cloud.clouds.msazure*), 1086
update_installed() (*in module salt.modules.smartos_imgadm*), 2501
update_interface() (*in module salt.modules.netbox*), 2202
update_item() (*in module salt.modules.rallydev*), 2419
update_jail() (*in module salt.modules.poudriere*), 2387
update_job() (*in module salt.modules.chronos*), 1506
update_job() (*in module salt.modules.jenkinsmod*), 1910
update_key_description() (*in module salt.modules.boto_kms*), 1418
update_linode() (*in module salt.cloud.clouds.linode*), 1070
update_lxc_conf() (*in module salt.modules.lxc*), 2002
update_network() (*in module salt.modules.neutron*), 2236
update_node() (*in module salt.modules.swarm*), 2573
update_object() (*in module salt.modules.infoblox*), 1872
update_one() (*in module salt.modules.mongodb*), 2087
update_org() (*in module salt.modules.grafana4*), 1829
update_org_address() (*in module salt.modules.grafana4*), 1829
update_org_prefs() (*in module salt.modules.grafana4*), 1829
update_org_user() (*in module salt.modules.grafana4*), 1829
update_package_site() (*in module salt.modules.pkgng*), 2365
update_packaging_site() (*in module salt.states.pkgng*), 3661
update_parameter_group() (*in module salt.modules.boto_rds*), 1429
update_port() (*in module salt.modules.neutron*), 2237
update_ports_tree() (*in module salt.modules.poudriere*), 2387
update_pricing() (*in module salt.cloud.clouds.ec2*), 1053
update_pricing() (*in module salt.cloud.clouds.gce*), 1059
update_quota() (*in module salt.modules.neutron*), 2237
update_record() (*in module salt.modules.boto_route53*), 1434
update_record_field() (*in module salt.modules.servicenow*), 2496
update_repo
    spm command line option, 991
update_resource() (*in module salt.modules.zonecfg*), 2912
update_restart_services
    conf/minion, 165
update_router() (*in module salt.modules.neutron*), 2237
update_saml_provider() (*in module salt.modules.boto_iam*), 1406
update_secret() (*in module salt.modules.k8s*), 1921
update_security_group() (*in module salt.modules.neutron*), 2238
update_simple_binding() (*in module salt.modules.jboss7*), 1906
update_stack() (*in module salt.modules.boto_cfn*), 1345
update_stack() (*in module salt.modules.heat*), 1847
update_storage() (*in module salt.cloud.clouds.msazure*), 1087
update_storage_policy() (*in module salt.modules.vsphere*), 2716
update_subnet() (*in module salt.modules.neutron*), 2238
update_system() (*in module salt.modules.gem*), 1752
update_url
    conf/minion, 165
update_usage_plan() (*in module salt.modules.boto_apigateway*), 1340
update_user() (*in module salt.modules.grafana4*), 1830
update_user() (*in module salt.modules.rallydev*), 2419
update_user() (*in module salt.modules.splunk*), 2536
update_user_password() (*in module*

salt.modules.grafana4), 1830
update_user_permissions() (*in module salt.modules.grafana4*), 1830
update_vm() (*in module salt.modules.vsphere*), 2716
update_vpnservice() (*in module salt.modules.neutron*), 2238
update_zone() (*in module salt.modules.libcloud_dns*), 1968
updated() (*in module salt.cache.localfs*), 1028
updated() (*in module salt.states.cyg*), 3361
updatedb() (*in module salt.modules.locate*), 1981
updating() (*in module salt.modules.pkgng*), 2366
upgrade() (*in module salt.modules.apkpkg*), 1215
upgrade() (*in module salt.modules.aptpkg*), 1225
upgrade() (*in module salt.modules.chocolatey*), 1505
upgrade() (*in module salt.modules.dummyproxy_pkg*), 1673
upgrade() (*in module salt.modules.ebuildpkg*), 1679
upgrade() (*in module salt.modules.freebsd_update*), 1739
upgrade() (*in module salt.modules.kernelpkg_linux_apt*), 1926
upgrade() (*in module salt.modules.kernelpkg_linux_yum*), 1927
upgrade() (*in module salt.modules.mac_brew_pkg*), 2024
upgrade() (*in module salt.modules.mac_portspkg*), 2034
upgrade() (*in module salt.modules.nix*), 2252
upgrade() (*in module salt.modules.openbsdpkg*), 2272
upgrade() (*in module salt.modules.opkg*), 2285
upgrade() (*in module salt.modules.pacmanpkg*), 2301
upgrade() (*in module salt.modules.pip*), 2349
upgrade() (*in module salt.modules.pkgin*), 2353
upgrade() (*in module salt.modules.pkgng*), 2366
upgrade() (*in module salt.modules.pkgutil*), 2369
upgrade() (*in module salt.modules.rest_pkg*), 2439
upgrade() (*in module salt.modules.solarisipspkg*), 2523
upgrade() (*in module salt.modules.win_pkg*), 2784
upgrade() (*in module salt.modules.xbpspkg*), 2859
upgrade() (*in module salt.modules.yumpkg*), 2875
upgrade() (*in module salt.modules.zpool*), 2923
upgrade() (*in module salt.modules.zypperpkg*), 2933
upgrade() (*in module salt.proxy.dummy*), 3057
upgrade_available() (*in module salt.modules.aixpkg*), 1205
upgrade_available() (*in module salt.modules.aptpkg*), 1226
upgrade_available() (*in module salt.modules.ebuildpkg*), 1680
upgrade_available() (*in module salt.modules.kernelpkg_linux_apt*), 1926
upgrade_available() (*in module salt.modules.kernelpkg_linux_yum*), 1927
upgrade_available() (*in module salt.modules.mac_brew_pkg*), 2025
upgrade_available() (*in module salt.modules.mac_portspkg*), 2034
upgrade_available() (*in module salt.modules.openbsdpkg*), 2272
upgrade_available() (*in module salt.modules.opkg*), 2285
upgrade_available() (*in module salt.modules.pacmanpkg*), 2301
upgrade_available() (*in module salt.modules.pip*), 2349
upgrade_available() (*in module salt.modules.pkgutil*), 2370
upgrade_available() (*in module salt.modules.solarisipspkg*), 2523
upgrade_available() (*in module salt.modules.solarispkg*), 2527
upgrade_available() (*in module salt.modules.win_pkg*), 2784
upgrade_available() (*in module salt.modules.xbpspkg*), 2859
upgrade_available() (*in module salt.modules.yumpkg*), 2877
upgrade_available() (*in module salt.modules.zypperpkg*), 2934
upgrade_bootstrap() (*in module salt.modules.zcbuildout*), 2897
upgrade_elasticsearch_domain() (*in module salt.modules.boto3_elasticsearch*), 1321
upgrade_tools() (*in module salt.cloud.clouds.vmware*), 1147
upgrade_tools_all() (*in module salt.cloud.clouds.vmware*), 1147
upgraded() (*in module salt.states.boto3_elasticsearch*), 3245
upgraded() (*in module salt.states.chocolatey*), 3339
upload_object() (*in module salt.modules.libcloud_storage*), 1974
upload_server_cert() (*in module salt.modules.boto_iam*), 1406
upload_ssh_key() (*in module salt.modules.vsphere*), 2716
upstream_exists() (*in module salt.modules.rabbitmq*), 2418
uptime() (*in module salt.grains.napalm*), 1184
uptime() (*in module salt.modules.status*), 2561
uptime() (*in module salt.modules.win_status*), 2811
uptodate() (*in module salt.proxy.dummy*), 3057
uptodate() (*in module salt.proxy.rest_sample*), 3086
uptodate() (*in module salt.states.pip_state*), 3642
uptodate() (*in module salt.states.pkg*), 3657
uptodate() (*in module salt.states.win_wua*), 3786
usage() (*in module salt.modules.btrfs*), 1471

`usage()` (*in module salt.modules.disk*), 1605
`usage()` (*in module salt.modules.win_disk*), 2723
`usage_plan_absent()` (*in module salt.states.boto_apigateway*), 3253
`usage_plan_association_absent()` (*in module salt.states.boto_apigateway*), 3253
`usage_plan_association_present()` (*in module salt.states.boto_apigateway*), 3254
`usage_plan_present()` (*in module salt.states.boto_apigateway*), 3254
`usages_list()` (*in module salt.modules.azurearm_network*), 1269
`usb_devices()` (*in module salt.modules.osquery*), 2295
`use_master_when_local`
  `conf/minion`, 152
`user`
  `conf/master`, 52
  `conf/minion`, 129
`user` (*salt.engines.ircbot.PrivEvent attribute*), 1156
`user()` (*in module salt.states.cimc*), 3342
`user_absent()` (*in module salt.states.boto_iam*), 3297
`user_absent()` (*in module salt.states.ipmi*), 3499
`user_absent()` (*in module salt.states.keystone*), 3519
`user_absent()` (*in module salt.states.nxos*), 3614
`user_absent()` (*in module salt.states.webutil*), 3756
`user_addmedia()` (*in module salt.modules.zabbix*), 2888
`user_chpass()` (*in module salt.modules.influxdb08mod*), 1865
`user_chpass()` (*in module salt.modules.mysql*), 2105
`user_create()` (*in module salt.modules.influxdb08mod*), 1865
`user_create()` (*in module salt.modules.keystone*), 1935
`user_create()` (*in module salt.modules.keystoneng*), 1943
`user_create()` (*in module salt.modules.mongodb*), 2087
`user_create()` (*in module salt.modules.mssql*), 2097
`user_create()` (*in module salt.modules.mysql*), 2106
`user_create()` (*in module salt.modules.postgres*), 2384
`user_create()` (*in module salt.modules.zabbix*), 2889
`user_delete()` (*in module salt.modules.ipmi*), 1895
`user_delete()` (*in module salt.modules.keystone*), 1935
`user_delete()` (*in module salt.modules.keystoneng*), 1943
`user_delete()` (*in module salt.modules.zabbix*), 2889
`user_deletemedia()` (*in module salt.modules.zabbix*), 2889
`user_exists()` (*in module salt.modules.influxdb08mod*), 1866
`user_exists()` (*in module salt.modules.influxdbmod*), 1861
`user_exists()` (*in module salt.modules.mongodb*), 2088
`user_exists()` (*in module salt.modules.mssql*), 2097
`user_exists()` (*in module salt.modules.mysql*), 2106
`user_exists()` (*in module salt.modules.postgres*), 2385
`user_exists()` (*in module salt.modules.rabbitmq*), 2418
`user_exists()` (*in module salt.modules.zabbix*), 2890
`user_exists()` (*in module salt.states.webutil*), 3756
`user_exists_in_group()` (*in module salt.modules.boto_iam*), 1406
`user_find()` (*in module salt.modules.mongodb*), 2088
`user_get()` (*in module salt.modules.keystone*), 1935
`user_get()` (*in module salt.modules.keystoneng*), 1943
`user_get()` (*in module salt.modules.zabbix*), 2890
`user_getmedia()` (*in module salt.modules.zabbix*), 2890
`user_grant_roles()` (*in module salt.modules.mongodb*), 2088
`user_grants()` (*in module salt.modules.mysql*), 2106
`user_info()` (*in module salt.modules.influxdbmod*), 1861
`user_info()` (*in module salt.modules.mysql*), 2106
`user_keys()` (*in module salt.modules.ssh*), 2542
`user_list()` (*in module salt.modules.influxdb08mod*), 1866
`user_list()` (*in module salt.modules.keystone*), 1935
`user_list()` (*in module salt.modules.keystoneng*), 1943
`user_list()` (*in module salt.modules.mongodb*), 2088
`user_list()` (*in module salt.modules.mssql*), 2097
`user_list()` (*in module salt.modules.mysql*), 2107
`user_list()` (*in module salt.modules.postgres*), 2385
`user_list()` (*in module salt.modules.zabbix*), 2891
`user_password_update()` (*in module salt.modules.keystone*), 1935
`user_present()` (*in module salt.states.boto_iam*), 3298
`user_present()` (*in module salt.states.ipmi*), 3499
`user_present()` (*in module salt.states.keystone*), 3520
`user_present()` (*in module salt.states.nxos*), 3614
`user_remove()` (*in module salt.modules.influxdb08mod*), 1866
`user_remove()` (*in module salt.modules.mongodb*), 2088
`user_remove()` (*in module salt.modules.mssql*), 2098
`user_remove()` (*in module salt.modules.mysql*), 2107
`user_remove()` (*in module salt.modules.postgres*), 2385
`user_revoke_roles()` (*in module salt.modules.mongodb*), 2088

user_role_add() (*in module salt.modules.keystone*), 1935
user_role_list() (*in module salt.modules.keystone*), 1936
user_role_remove() (*in module salt.modules.keystone*), 1936
user_roles_exists() (*in module salt.modules.mongodb*), 2089
user_search() (*in module salt.modules.keystoneng*), 1943
user_to_uid() (*in module salt.modules.file*), 1730
user_to_uid() (*in module salt.modules.win_file*), 2744
user_update() (*in module salt.modules.keystone*), 1936
user_update() (*in module salt.modules.keystoneng*), 1943
user_update() (*in module salt.modules.postgres*), 2385
user_update() (*in module salt.modules.zabbix*), 2891
user_verify_password() (*in module salt.modules.keystone*), 1936
useradd() (*in module salt.modules.apache*), 1211
useradd() (*in module salt.modules.webutil*), 2719
userdata_template
    conf/master, 77
userdel() (*in module salt.modules.apache*), 1211
userdel() (*in module salt.modules.webutil*), 2719
usergroup_create() (*in module salt.modules.zabbix*), 2891
usergroup_delete() (*in module salt.modules.zabbix*), 2892
usergroup_exists() (*in module salt.modules.zabbix*), 2892
usergroup_get() (*in module salt.modules.zabbix*), 2892
usergroup_list() (*in module salt.modules.zabbix*), 2893
usergroup_update() (*in module salt.modules.zabbix*), 2893
usermacro_create() (*in module salt.modules.zabbix*), 2893
usermacro_createglobal() (*in module salt.modules.zabbix*), 2894
usermacro_delete() (*in module salt.modules.zabbix*), 2894
usermacro_deleteglobal() (*in module salt.modules.zabbix*), 2894
usermacro_get() (*in module salt.modules.zabbix*), 2895
usermacro_update() (*in module salt.modules.zabbix*), 2895
usermacro_updateglobal() (*in module salt.modules.zabbix*), 2895
username() (*in module salt.grains.napalm*), 1184
users() (*in module salt.modules.osquery*), 2295
ustring() (*salt.output.nested.NestDisplay method*), 2982
ustring() (*salt.output.table_out.TableDisplay method*), 2988
utils_dirs
    conf/master, 75
    conf/minion, 145
uuid
    jinja filters, 431
uuid() (*in module salt.modules.bcache*), 1283

V

vacuum() (*in module salt.modules.smartos_imgadm*), 2502
valid() (*in module salt.modules.mine*), 2080
valid_certificate() (*in module salt.states.tls*), 3736
valid_fileproto() (*in module salt.modules.config*), 1560
validate() (*in module salt.beacons.adb*), 1006
validate() (*in module salt.beacons.avahi_announce*), 1007
validate() (*in module salt.beacons.bonjour_announce*), 1008
validate() (*in module salt.beacons.btmp*), 1010
validate() (*in module salt.beacons.diskusage*), 1011
validate() (*in module salt.beacons.glxinfo*), 1011
validate() (*in module salt.beacons.haproxy*), 1011
validate() (*in module salt.beacons.inotify*), 1013
validate() (*in module salt.beacons.journald*), 1013
validate() (*in module salt.beacons.load*), 1014
validate() (*in module salt.beacons.memusage*), 1014
validate() (*in module salt.beacons.napalm_beacon*), 1017
validate() (*in module salt.beacons.network_info*), 1017
validate() (*in module salt.beacons.network_settings*), 1018
validate() (*in module salt.beacons.pkg*), 1019
validate() (*in module salt.beacons.proxy_example*), 1019
validate() (*in module salt.beacons.ps*), 1019
validate() (*in module salt.beacons.salt_proxy*), 1020
validate() (*in module salt.beacons.sensehat*), 1020
validate() (*in module salt.beacons.service*), 1021
validate() (*in module salt.beacons.sh*), 1021
validate() (*in module salt.beacons.status*), 1023
validate() (*in module salt.beacons.telegram_bot_msg*), 1023
validate() (*in module salt.beacons.twilio_txt_msg*), 1023
validate() (*in module salt.beacons.wtmp*), 1025

`validate()` (*in module salt.modules.monit*), 2090
`validate()` (*in module salt.pillar.pepa*), 3026
`validate_template()` (*in module salt.modules.boto_cfn*), 1346
`value_exists()` (*in module salt.modules.reg*), 2438
`value_present()` (*in module salt.states.cisconso*), 3342
`value_present()` (*in module salt.states.xml*), 3792
`values()` (*in module salt.modules.data*), 1588
`values()` (*in module salt.wheel.config*), 3838
`var_contains()` (*in module salt.modules.makeconf*), 2068
`vbd_list()` (*in module salt.cloud.clouds.xen*), 1152
`vboxcmd()` (*in module salt.modules.vboxmanage*), 2644
`vcpu_pin()` (*in module salt.modules.xapi_virt*), 2856
`vdi_list()` (*in module salt.cloud.clouds.xen*), 1152
`vendor()` (*in module salt.grains.napalm*), 1184
`vendor()` (*in module salt.grains.philips_hue*), 1185
`verify()` (*in module salt.modules.gpg*), 1822
`verify()` (*in module salt.modules.rpm_lowpkg*), 2448
`verify()` (*in module salt.modules.webutil*), 2720
`verify()` (*in module salt.modules.yumpkg*), 2877
`verify()` (*in module salt.modules.zoneadm*), 2910
`verify()` (*in module salt.modules.zypperpkg*), 2934
`verify_crl()` (*in module salt.modules.x509*), 2852
`verify_env`
    `conf/master`, 55
`verify_login()` (*in module salt.modules.mysql*), 2107
`verify_master_pubkey_sign`
    `conf/minion`, 157
`verify_private_key()` (*in module salt.modules.x509*), 2852
`verify_signature()` (*in module salt.modules.x509*), 2852
`verify_webhook()` (*in module salt.modules.travisci*), 2624
`version()` (*in module salt.grains.napalm*), 1184
`version()` (*in module salt.modules.aixpkg*), 1205
`version()` (*in module salt.modules.apache*), 1211
`version()` (*in module salt.modules.apkpkg*), 1215
`version()` (*in module salt.modules.aptpkg*), 1226
`version()` (*in module salt.modules.bluez_bluetooth*), 1301
`version()` (*in module salt.modules.btrfs*), 1471
`version()` (*in module salt.modules.cassandra*), 1482
`version()` (*in module salt.modules.cassandra_cql*), 1489
`version()` (*in module salt.modules.chocolatey*), 1505
`version()` (*in module salt.modules.dnsmasq*), 1607
`version()` (*in module salt.modules.dockermod*), 1660
`version()` (*in module salt.modules.dummyproxy_pkg*), 1673
`version()` (*in module salt.modules.ebuildpkg*), 1680
`version()` (*in module salt.modules.firewalld*), 1737
`version()` (*in module salt.modules.freebsdpkg*), 1744
`version()` (*in module salt.modules.gem*), 1752
`version()` (*in module salt.modules.git*), 1796
`version()` (*in module salt.modules.grub_legacy*), 1838
`version()` (*in module salt.modules.hadoop*), 1840
`version()` (*in module salt.modules.ipset*), 1897
`version()` (*in module salt.modules.iptables*), 1902
`version()` (*in module salt.modules.kapacitor*), 1923
`version()` (*in module salt.modules.linux_acl*), 1976
`version()` (*in module salt.modules.linux_lvm*), 1978
`version()` (*in module salt.modules.locate*), 1981
`version()` (*in module salt.modules.lxc*), 2003
`version()` (*in module salt.modules.lxd*), 2021
`version()` (*in module salt.modules.mac_brew_pkg*), 2025
`version()` (*in module salt.modules.mac_portspkg*), 2034
`version()` (*in module salt.modules.modjk*), 2085
`version()` (*in module salt.modules.mongodb*), 2089
`version()` (*in module salt.modules.monit*), 2090
`version()` (*in module salt.modules.mssql*), 2098
`version()` (*in module salt.modules.mysql*), 2107
`version()` (*in module salt.modules.nftables*), 2247
`version()` (*in module salt.modules.nginx*), 2248
`version()` (*in module salt.modules.openbsdpkg*), 2272
`version()` (*in module salt.modules.opkg*), 2285
`version()` (*in module salt.modules.oracle*), 2287
`version()` (*in module salt.modules.osquery*), 2296
`version()` (*in module salt.modules.pacmanpkg*), 2301
`version()` (*in module salt.modules.pip*), 2349
`version()` (*in module salt.modules.pkg_resource*), 2350
`version()` (*in module salt.modules.pkgin*), 2353
`version()` (*in module salt.modules.pkgng*), 2367
`version()` (*in module salt.modules.pkgutil*), 2370
`version()` (*in module salt.modules.postgres*), 2385
`version()` (*in module salt.modules.poudriere*), 2387
`version()` (*in module salt.modules.rest_pkg*), 2439
`version()` (*in module salt.modules.rsync*), 2452
`version()` (*in module salt.modules.smartos_imgadm*), 2502
`version()` (*in module salt.modules.solarisipspkg*), 2524
`version()` (*in module salt.modules.solarispkg*), 2527
`version()` (*in module salt.modules.solr*), 2533
`version()` (*in module salt.modules.sqlite3*), 2538
`version()` (*in module salt.modules.status*), 2561
`version()` (*in module salt.modules.syslog_ng*), 2582
`version()` (*in module salt.modules.test*), 2604
`version()` (*in module salt.modules.tomcat*), 2622
`version()` (*in module salt.modules.vagrant*), 2637
`version()` (*in module salt.modules.varnish*), 2638
`version()` (*in module salt.modules.win_pkg*), 2784
`version()` (*in module salt.modules.xbpspkg*), 2859
`version()` (*in module salt.modules.yumpkg*), 2877

version() (*in module salt.modules.znc*), 2907
version() (*in module salt.modules.zypperpkg*), 2934
version() (*in module salt.runners.drac*), 3108
version_clean() (*in module salt.modules.ebuildpkg*), 1680
version_clean() (*in module salt.modules.opkg*), 2285
version_clean() (*in module salt.modules.pkg_resource*), 2350
version_cmp() (*in module salt.modules.aptpkg*), 1226
version_cmp() (*in module salt.modules.ebuildpkg*), 1680
version_cmp() (*in module salt.modules.opkg*), 2285
version_cmp() (*in module salt.modules.pkgng*), 2367
version_cmp() (*in module salt.modules.rpm_lowpkg*), 2449
version_cmp() (*in module salt.modules.yumpkg*), 2877
version_cmp() (*in module salt.modules.zypperpkg*), 2934
version_compatible() (*in module salt.cloud.clouds.profitbricks*), 1119
versions() (*in module salt.modules.pyenv*), 2410
versions() (*in module salt.modules.rbenv*), 2426
versions() (*in module salt.modules.test*), 2604
versions() (*in module salt.runners.manage*), 3126
versions_information() (*in module salt.modules.test*), 2604
versions_report() (*in module salt.modules.test*), 2604
vfstab() (*in module salt.modules.mount*), 2095
vg_absent() (*in module salt.states.lvm*), 3543
vg_present() (*in module salt.states.lvm*), 3544
vgcreate() (*in module salt.modules.linux_lvm*), 1978
vgdisplay() (*in module salt.modules.linux_lvm*), 1978
vgextend() (*in module salt.modules.linux_lvm*), 1978
vgremove() (*in module salt.modules.linux_lvm*), 1978
vhost_exists() (*in module salt.modules.rabbitmq*), 2418
vhosts() (*in module salt.modules.apache*), 1211
vif_list() (*in module salt.cloud.clouds.xen*), 1152
virt_type() (*in module salt.modules.virt*), 2673
virtual_interface_create() (*in module salt.modules.cloud*), 1522
virtual_interface_list() (*in module salt.modules.cloud*), 1522
virtual_machine_capture() (*in module salt.modules.azurearm_compute*), 1250
virtual_machine_convert_to_managed_disks() (*in module salt.modules.azurearm_compute*), 1251
virtual_machine_deallocate() (*in module salt.modules.azurearm_compute*), 1251
virtual_machine_generalize() (*in module salt.modules.azurearm_compute*), 1251
virtual_machine_get() (*in module salt.modules.azurearm_compute*), 1251
virtual_machine_power_off() (*in module salt.modules.azurearm_compute*), 1252
virtual_machine_redeploy() (*in module salt.modules.azurearm_compute*), 1252
virtual_machine_restart() (*in module salt.modules.azurearm_compute*), 1252
virtual_machine_start() (*in module salt.modules.azurearm_compute*), 1252
virtual_machines_list() (*in module salt.modules.azurearm_compute*), 1253
virtual_machines_list_all() (*in module salt.modules.azurearm_compute*), 1253
virtual_machines_list_available_sizes() (*in module salt.modules.azurearm_compute*), 1253
virtual_memory() (*in module salt.modules.ps*), 2393
virtual_network_absent() (*in module salt.states.azurearm_network*), 3215
virtual_network_create_or_update() (*in module salt.modules.azurearm_network*), 1269
virtual_network_delete() (*in module salt.modules.azurearm_network*), 1270
virtual_network_get() (*in module salt.modules.azurearm_network*), 1270
virtual_network_present() (*in module salt.states.azurearm_network*), 3215
virtual_networks_list() (*in module salt.modules.azurearm_network*), 1270
virtual_networks_list_all() (*in module salt.modules.azurearm_network*), 1270
virtualenv_ver() (*in module salt.modules.virtualenv_mod*), 2678
vm_absent() (*in module salt.states.smartos*), 3716
vm_action() (*in module salt.cloud.clouds.opennebula*), 1101
vm_allocate() (*in module salt.cloud.clouds.opennebula*), 1101
vm_attach() (*in module salt.cloud.clouds.opennebula*), 1102
vm_attach_nic() (*in module salt.cloud.clouds.opennebula*), 1102
vm_cputime() (*in module salt.modules.virt*), 2673
vm_cputime() (*in module salt.modules.xapi_virt*), 2856
vm_deploy() (*in module salt.cloud.clouds.opennebula*), 1102
vm_detach() (*in module salt.cloud.clouds.opennebula*), 1103
vm_detach_nic() (*in module salt.cloud.clouds.opennebula*), 1103
vm_disk_save() (*in module*

salt.cloud.clouds.opennebula), 1103
`vm_disk_snapshot_create()` (*in module salt.cloud.clouds.opennebula*), 1104
`vm_disk_snapshot_delete()` (*in module salt.cloud.clouds.opennebula*), 1104
`vm_disk_snapshot_revert()` (*in module salt.cloud.clouds.opennebula*), 1104
`vm_diskstats()` (*in module salt.modules.virt*), 2673
`vm_diskstats()` (*in module salt.modules.xapi_virt*), 2856
`vm_info()` (*in module salt.cloud.clouds.opennebula*), 1104
`vm_info()` (*in module salt.modules.smartos_virt*), 2504
`vm_info()` (*in module salt.modules.virt*), 2674
`vm_info()` (*in module salt.modules.xapi_virt*), 2856
`vm_info()` (*in module salt.runners.virt*), 3159
`vm_migrate()` (*in module salt.cloud.clouds.opennebula*), 1104
`vm_monitoring()` (*in module salt.cloud.clouds.opennebula*), 1105
`vm_netstats()` (*in module salt.modules.virt*), 2674
`vm_netstats()` (*in module salt.modules.xapi_virt*), 2856
`vm_present()` (*in module salt.states.smartos*), 3716
`vm_resize()` (*in module salt.cloud.clouds.opennebula*), 1105
`vm_running()` (*in module salt.states.smartos*), 3717
`vm_snapshot_create()` (*in module salt.cloud.clouds.opennebula*), 1106
`vm_snapshot_delete()` (*in module salt.cloud.clouds.opennebula*), 1106
`vm_snapshot_revert()` (*in module salt.cloud.clouds.opennebula*), 1106
`vm_state()` (*in module salt.modules.vagrant*), 2637
`vm_state()` (*in module salt.modules.virt*), 2675
`vm_state()` (*in module salt.modules.xapi_virt*), 2857
`vm_stopped()` (*in module salt.states.smartos*), 3717
`vm_update()` (*in module salt.cloud.clouds.opennebula*), 1106
`vm_virt_type()` (*in module salt.modules.smartos_virt*), 2504
`vmotion_configured()` (*in module salt.states.esxi*), 3414
`vmotion_disable()` (*in module salt.modules.vsphere*), 2717
`vmotion_enable()` (*in module salt.modules.vsphere*), 2717
`vms()` (*in module salt.modules.smartos_nictagadm*), 2503
`vmstats()` (*in module salt.modules.status*), 2562
`vmstats()` (*in module salt.modules.win_status*), 2811
`VMwareApiError`, 4000
`VMwareConnectionError`, 4001
`VMwareFileNotFoundError`, 4001
`VMwareMultipleObjectsError`, 4001
`VMwareNotFoundError`, 4001
`VMwareObjectExistsError`, 4001
`VMwareObjectNotFoundError`, 4001
`VMwareObjectRetrievalError`, 4001
`VMwarePowerOnError`, 4001
`VMwareRuntimeError`, 4001
`VMwareSaltError`, 4001
`VMwareSystemError`, 4001
`VMwareVmCreationError`, 4001
`VMwareVmRegisterError`, 4001
`vn_add_ar()` (*in module salt.cloud.clouds.opennebula*), 1106
`vn_allocate()` (*in module salt.cloud.clouds.opennebula*), 1107
`vn_delete()` (*in module salt.cloud.clouds.opennebula*), 1107
`vn_free_ar()` (*in module salt.cloud.clouds.opennebula*), 1107
`vn_hold()` (*in module salt.cloud.clouds.opennebula*), 1108
`vn_info()` (*in module salt.cloud.clouds.opennebula*), 1108
`vn_release()` (*in module salt.cloud.clouds.opennebula*), 1108
`vn_reserve()` (*in module salt.cloud.clouds.opennebula*), 1108
`volume_absent()` (*in module salt.states.boto_ec2*), 3278
`volume_absent()` (*in module salt.states.cloud*), 3344
`volume_absent()` (*in module salt.states.zfs*), 3809
`volume_attach()` (*in module salt.modules.cloud*), 1522
`volume_attach()` (*in module salt.modules.nova*), 2256
`volume_attach()` (*in module salt.modules.purefa*), 2400
`volume_attached()` (*in module salt.states.cloud*), 3344
`volume_clone()` (*in module salt.modules.purefa*), 2400
`volume_create()` (*in module salt.cloud.clouds.ec2*), 1053
`volume_create()` (*in module salt.modules.cloud*), 1522
`volume_create()` (*in module salt.modules.nova*), 2256
`volume_create()` (*in module salt.modules.purefa*), 2401
`volume_delete()` (*in module salt.modules.cloud*), 1522
`volume_delete()` (*in module salt.modules.nova*), 2257
`volume_delete()` (*in module salt.modules.purefa*), 2401

volume_delete() (*in module salt.modules.virt*), 2675

volume_detach() (*in module salt.modules.cloud*), 1522

volume_detach() (*in module salt.modules.nova*), 2257

volume_detach() (*in module salt.modules.purefa*), 2401

volume_detached() (*in module salt.states.cloud*), 3344

volume_eradicate() (*in module salt.modules.purefa*), 2401

volume_extend() (*in module salt.modules.purefa*), 2401

volume_infos() (*in module salt.modules.virt*), 2676

volume_list() (*in module salt.cloud.clouds.ec2*), 1053

volume_list() (*in module salt.modules.cloud*), 1522

volume_list() (*in module salt.modules.nova*), 2257

volume_present() (*in module salt.states.boto_ec2*), 3278

volume_present() (*in module salt.states.cloud*), 3344

volume_present() (*in module salt.states.glusterfs*), 3467

volume_present() (*in module salt.states.zfs*), 3809

volume_show() (*in module salt.modules.nova*), 2257

volumes() (*in module salt.modules.dockermod*), 1660

volumes_tagged() (*in module salt.states.boto_ec2*), 3279

vpc_peering_connection_absent() (*in module salt.states.boto_vpc*), 3334

vpc_peering_connection_present() (*in module salt.states.boto_vpc*), 3334

vsan_add_disks() (*in module salt.modules.vsphere*), 2718

vsan_configured() (*in module salt.states.esxi*), 3414

vsan_disable() (*in module salt.modules.vsphere*), 2718

vsan_enable() (*in module salt.modules.vsphere*), 2718

vserver_add() (*in module salt.modules.netscaler*), 2214

vserver_delete() (*in module salt.modules.netscaler*), 2214

vserver_exists() (*in module salt.modules.netscaler*), 2214

vserver_servicegroup_add() (*in module salt.modules.netscaler*), 2214

vserver_servicegroup_delete() (*in module salt.modules.netscaler*), 2214

vserver_servicegroup_exists() (*in module salt.modules.netscaler*), 2215

vserver_sslcert_add() (*in module salt.modules.netscaler*), 2215

vserver_sslcert_delete() (*in module salt.modules.netscaler*), 2215

vserver_sslcert_exists() (*in module salt.modules.netscaler*), 2215

W

w() (*in module salt.modules.status*), 2562

wait() (*in module salt.modules.dockermod*), 1660

wait() (*in module salt.states.cmd*), 3351

wait() (*in module salt.states.event*), 3415

wait() (*in module salt.states.module*), 3560

wait() (*in module salt.states.tomcat*), 3737

wait_call() (*in module salt.states.cmd*), 3352

wait_for_created() (*in module salt.cloud.clouds.proxmox*), 1121

wait_for_event() (*in module salt.states.saltmod*), 3700

wait_for_instance() (*in module salt.cloud.clouds.ec2*), 1053

wait_for_state() (*in module salt.cloud.clouds.proxmox*), 1121

wait_for_successful_query() (*in module salt.modules.http*), 1851

wait_for_successful_query() (*in module salt.states.http*), 3487

wait_for_upgrade() (*in module salt.modules.boto3_elasticsearch*), 1322

wait_rm() (*in module salt.states.etcd_mod*), 3407

wait_script() (*in module salt.states.cmd*), 3352

wait_set() (*in module salt.states.etcd_mod*), 3407

wait_started() (*in module salt.modules.lxc*), 2003

wait_state() (*in module salt.modules.haproxyconn*), 1842

wait_until() (*in module salt.cloud.clouds.parallels*), 1115

war_deployed() (*in module salt.states.tomcat*), 3738

warn() (*in module salt.modules.quota*), 2412

warning() (*in module salt.modules.logmod*), 1983

watch() (*in module salt.modules.etcd_mod*), 1698

watch() (*in module salt.states.at*), 3197

webconfiguration_settings() (*in module salt.states.win_iis*), 3769

Webhook (*class in salt.netapi.rest_cherrypy.app*), 2954

WebhookSaltAPIHandler (*class in salt.netapi.rest_tornado.saltnado*), 2975

WebsocketEndpoint (*class in salt.netapi.rest_cherrypy.app*), 2958

wheel() (*in module salt.modules.saltutil*), 2480

wheel() (*in module salt.runners.doc*), 3107

wheel() (*in module salt.states.saltmod*), 3701

wheel() (*salt.netapi.NetapiClient method*), 3855

wheel_async() (*salt.netapi.NetapiClient method*), 3855

WheelClient (*class in salt.wheel*), 3851

which
- jinja filters, 448

`which()` (*in module salt.modules.cmdmod*), 1554
`which()` (*in module salt.modules.pkgng*), 2367
`which_bin()` (*in module salt.modules.cmdmod*), 1554
`Wildcard` (*class in salt.modules.zypperpkg*), 2924
`will_expire()` (*in module salt.modules.x509*), 2852
`win_gitrepos`
    `conf/master`, 119
    `conf/minion`, 167
`win_repo`
    `conf/master`, 118
    `conf/minion`, 166
`win_repo_cachefile`
    `conf/minion`, 166
`win_repo_mastercachefile`
    `conf/master`, 119
`WindowsError`, 2733
`winrepo_branch`
    `conf/master`, 120
`winrepo_cache_expire_max`
    `conf/minion`, 165
`winrepo_cache_expire_min`
    `conf/minion`, 165
`winrepo_cachefile`
    `conf/master`, 119
    `conf/minion`, 166
`winrepo_dir`
    `conf/master`, 118
    `conf/minion`, 166
`winrepo_dir_ng`
    `conf/master`, 118
    `conf/minion`, 166
`winrepo_insecure_auth`
    `conf/master`, 121
`winrepo_passphrase`
    `conf/master`, 121
`winrepo_password`
    `conf/master`, 120
`winrepo_privkey`
    `conf/master`, 121
`winrepo_provider`
    `conf/master`, 118
`winrepo_pubkey`
    `conf/master`, 121
`winrepo_refspecs`
    `conf/master`, 121
`winrepo_remotes`
    `conf/master`, 119
    `conf/minion`, 167
`winrepo_remotes_ng`
    `conf/master`, 119
    `conf/minion`, 167
`winrepo_source_dir`
    `conf/minion`, 166
`winrepo_ssl_verify`
    `conf/master`, 120
`winrepo_user`
    `conf/master`, 120
`wipe()` (*in module salt.modules.disk*), 1605
`wipefacls()` (*in module salt.modules.linux_acl*), 1976
`with_lists` (*salt.pillar.sql_base.SqlBaseExtPillar attribute*), 3035
`wm_preferences()` (*in module salt.states.gnomedesktop*), 3469
`wol()` (*in module salt.modules.network*), 2221
`wol()` (*in module salt.runners.network*), 3136
`wollist()` (*in module salt.runners.network*), 3136
`wolmatch()` (*in module salt.runners.network*), 3136
`work()` (*salt.engines.redis_sentinel.Listener method*), 1164
`Worker`, **5403**
`worker_activate()` (*in module salt.modules.modjk*), 2085
`worker_activated()` (*in module salt.states.modjk*), 3553
`worker_disable()` (*in module salt.modules.modjk*), 2085
`worker_disabled()` (*in module salt.states.modjk*), 3553
`worker_edit()` (*in module salt.modules.modjk*), 2085
`worker_recover()` (*in module salt.modules.modjk*), 2086
`worker_recover()` (*in module salt.states.modjk*), 3553
`worker_status()` (*in module salt.modules.modjk*), 2086
`worker_stop()` (*in module salt.modules.modjk*), 2086
`worker_stopped()` (*in module salt.states.modjk*), 3554
`worker_threads`
    `conf/master`, 74
`workers()` (*in module salt.modules.modjk*), 2086
`worktree_add()` (*in module salt.modules.git*), 1796
`worktree_prune()` (*in module salt.modules.git*), 1797
`worktree_rm()` (*in module salt.modules.git*), 1798
`wrap_onspace()` (*salt.output.table_out.TableDisplay method*), 2988
`wrapper()` (*in module salt.modules.rvm*), 2458
`write()` (*in module salt.modules.file*), 1730
`write()` (*in module salt.modules.mac_xattr*), 2060
`write()` (*in module salt.modules.macdefaults*), 2025
`write()` (*in module salt.modules.sysfs*), 2577
`write()` (*in module salt.states.macdefaults*), 3548
`write()` (*in module salt.wheel.file_roots*), 3838
`write()` (*in module salt.wheel.pillar_roots*), 3842
`write_conf()` (*in module salt.modules.lxc*), 2003
`write_config()` (*in module salt.modules.syslog_ng*), 2582

write_cron_file() (*in module salt.modules.cron*), 1581
write_cron_file_verbose() (*in module salt.modules.cron*), 1581
write_incron_file() (*in module salt.modules.incron*), 1856
write_incron_file_verbose() (*in module salt.modules.incron*), 1857
write_launchd_plist() (*in module salt.runners.launchd*), 3118
write_mount_cache() (*in module salt.modules.mount*), 2095
write_pem() (*in module salt.modules.x509*), 2853
write_raw() (*in module salt.modules.vault*), 2640
write_secret() (*in module salt.modules.vault*), 2641
write_version() (*in module salt.modules.syslog_ng*), 2582

X

xattr_where_from() (*in module salt.modules.osquery*), 2296
xccdf() (*in module salt.modules.openscap*), 2276
xorg() (*in module salt.states.keyboard*), 3517
xprotect_entries() (*in module salt.modules.osquery*), 2296
xprotect_reports() (*in module salt.modules.osquery*), 2296

Y

yaml_dquote
  jinja filters, 429
yaml_encode
  jinja filters, 428
yaml_squote
  jinja filters, 429
yaml_utf8
  conf/master, 81

Z

Z_OP (*salt.modules.zypperpkg.Wildcard attribute*), 2924
zabbix_send() (*in module salt.returners.zabbix_return*), 357
zap() (*in module salt.modules.ceph*), 1498
zap() (*in module salt.modules.gentoo_service*), 1756
zbx() (*in module salt.returners.zabbix_return*), 357
zcard() (*in module salt.modules.redismod*), 2432
zero() (*in module salt.modules.lvs*), 1986
zero_cluster() (*in module salt.modules.trafficserver*), 2624
zero_cluster() (*in module salt.states.trafficserver*), 3740
zero_node() (*in module salt.modules.trafficserver*), 2624
zero_node() (*in module salt.states.trafficserver*), 3741
zeroize() (*in module salt.modules.junos*), 1919
zeroize() (*in module salt.states.junos*), 3514
zip_() (*in module salt.modules.archive*), 1233
zmq_backlog
  conf/master, 75
zmq_monitor
  conf/minion, 162
zmqversion() (*in module salt.grains.core*), 1179
zone_absent() (*in module salt.states.libcloud_dns*), 3535
zone_compare() (*in module salt.modules.mac_timezone*), 2056
zone_compare() (*in module salt.modules.timezone*), 2609
zone_compare() (*in module salt.modules.win_timezone*), 2833
zone_exists() (*in module salt.modules.boto_route53*), 1434
zone_present() (*in module salt.states.libcloud_dns*), 3535
zrange() (*in module salt.modules.redismod*), 2432